ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2014-ENST-0018

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Dinh-Phong VO

le 31 mars 2014

Inférence transductive pour l'interpretation et la recherche d'images

Directeur de thèse : **Hichem SAHBI**

Jury

Mme. Jenny BENOIS-PINEAU, Professeur, LaBRI, Université Bordeaux 1	Rapporteur
M. Chaabane DJERABA, Professeur, Université Lille 1	Rapporteur
M. Matthieu CORD, Professeur, LIP6, Université Pierre et Marie Curie	Examinateur
M. Frederic JURIE, Professeur, Université de Caen	Examinateur
M. Jean-Marc OGIER, Professeur, Université de la Rochelle	Examinateur

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

Résume de Thèse
Inférence transductive pour l'interprétation et la recherche d'images

Vo Dinh Phong

March 3, 2015

A partir de 1950, les chercheurs ont essayé d'inventer des machines intelligent. A cette époque, Alan Turing a introduit le test de Turing. Ce test vérifie la capacité des machines pour effectuer des comportements intelligents et le raisonnement tels qu'ils sont indiscernables de ceux d'un être humain. Le test devient bientôt un concept essentiel dans l'intelligence artificielle (AI). Beaucoup de chercheurs a l'époque étaient optimistes quant a la perspective de AI que les machines pouvaient passer le test dans les 20 ans. Jusqu'à présent, après plus de 60 ans, le rêve de machines pensantes est toujours insaisissable. Bien que les réalisations de AI sont juste a petits pas dans la reproduction de l'intelligence humaine, ils ont ouvert une nouvelle ère de la technologie de l'information.

Être un sous-domaine de AI, l'apprentissage automatique a pour but de concevoir des algorithmes qui améliorent automatiquement leurs comportements par l'expérience; l'apprentissage automatique se définit comme un ensemble de techniques statistiques spécialisées pour les données haut dimensionnelles. De 1990, des outils d'apprentissage automatique statistiques sont très populaires dans la résolution des problèmes spécifiques de AI. C'est en effet une étape remarquable dans le développement de AI. Bien que le rôle des approches statistiques a été controversée, ils ont apporté beaucoup de succès récemment. Diverses applications de l'intelligence artificielle ont été inventé et développé pour un usage quotidien tels que la traduction multi-langue de Google, la reconnaissance vocale dans Siri d'Apple, la reconnaissance des gestes dans Kinect de Microsoft. L'apprentissage automatique est l'un des facteurs clés de ces histoires de succès.

Parmi les problèmes d'apprentissage automatique, nous sommes intéressés a la vision de machine. Vision de machine vise a reproduire la capacité de la cognition humaine étonnante. En raison de l'augmentation rapide de contenu multimédia sur Internet et interactions profondes entre l'homme et les technologies information, les applications de vision de machine sont a chaque coin de la vie moderne. Dans cette thèse, nous présentons nos études sur de nouvelles méthodes d'apprentissage automatique dans la résolution de deux problèmes classiques de la vision de machine : image interprétation et de recherche.

Dans nos études, les algorithmes sont conçus pour apprendre représentations de données visuelles qui favorisent l'interprétation des images et la recherche même avec une quantité insuffisante de données. Ce chapitre d'introduction est consacrée aux discussions sur les contextes, les motivations ainsi que les contributions de notre recherche. Dans la section 1 nous discutons davantage le développement chronologique, des défis fondamentaux, ainsi que les réalisations de la vision de machine. Dans la section 2 nous révisons le rôle de l'apprentissage de la machine dans la résolution de certains problèmes de vision de machine. Notre motivation ainsi que les contributions sont introduits dans la section 3. La dernière section 4 explique l'organisation de la thèse.

1 Introduction à la reconnaissance d'objets

Créé au début des années 1960, la vision de machine vise a créer des algorithmes qui reproduisent la capacité humaine a percevoir et reconnaître le monde visuel. Des objectifs ambitieux de la vision de machine comprennent la détection, la reconnaissance et l'interprétation des objets visuels en images. Dans les premiers temps, il était populaire pour voir problème d'interprétation d'images comme un processus inverse de l'ordinateur de rendu graphique. En particulier, un pipeline graphique d'ordinateur de rendu est constitué de trois dimensions (3D) d'objets dans le monde de coordonnées et en les recouvrant avec des matériaux et illumination ; ces objets sont enfin projetés sur un écran a deux dimensions (2D). Objet interprétation, en revanche, récupéré des objets a partir d'images 2D. Cette perspective d'un problème de vision avait été le courant principal dans les premières années de son développement. Basé sur les conclusions sur l'organisation du cortex visuel humain [Hubel 1988], un problème de cognition visuelle est présumé comme un processus en trois étapes : première vision, vision a mi-niveau, et la vision de haut niveau. Le premier stade comprend des techniques de filtrage utilisées pour détecter les primitives visuelles telles que des bords, des couleurs, des textures. L'étape a mi-niveau traite de plus grandes entités telles que des correctifs et des régions d'image. Le stade de haut niveau en déduit sémantique de fonctionnalités de niveau intermédiaire tels que l'image entière forme un sens cohérent vu par la vision humaine. Alors que la phase de vision précoce a été partiellement exploré par les neuro-scientifiques et les psychologues de vision [Marr 1982, Livingstone 2008, Boden 2006, Biederman 1982], ils n'ont pas encore compris stades supérieurs.

1.1 Problèmes du vision de machine

En dépit des efforts de la communauté de la recherche, un algorithme qui reconnaît les objets visuels et génériques reste hors de portée. Même si nous mettons de côté les limites des infrastructures informatiques, les algorithmes de vision actuels sont encore contestées par de nombreuses difficultés (voir les exemples dans la figure 1) :

– Le premier défi est due a des conditions d'acquisition. Par exemple, les

conditions d'éclairage au moment de l'acquisition brûlent soit détails de l'image (environnement trop lumineux) ou les couvrir par l'ombre.
- Le deuxième défi est de comprendre le contexte. Objets dans le monde réel ne se produisent jamais dans l'isolement, mais coexistent avec d'autres objets et par rapport a des contextes particuliers. En fait, la vision humaine revient a être tolérant bruit et très imaginative a anticiper le sens d'une scène, même si les objets constituant ne sont pas complètes [Biederman 1982]. En outre, la littérature en psychologie souligne que la vision humaine voit toute la scène avant de reconnaître des objets individuels (lois de la Gestalt [Metzger 2006]).
- Apparence variabilité est un autre grand défi. Dans la figure 1(d), il y a quatre cas avec différentes apparences d'un seul concept *chaise*, bien que regardant différemment, tous ces cas ont une fonctionnalité commune d'une chaise. Jusqu'à présent, les psychologues n'ont pas découvert une assez bonne théorie qui explique universellement différentes formes de concepts humains [Murphy 2004].

1.2 Les applications de vision de machine

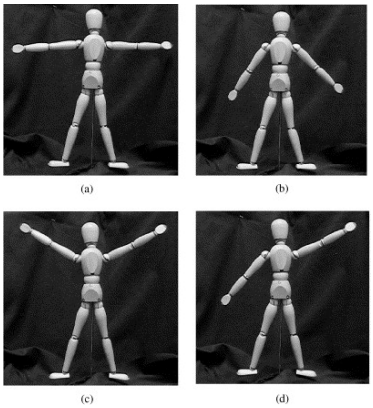
Malgré ces défis, la vision de machine de nos jours ont dépassé ce que les gens imaginaient il y a des décennies. Différentes approches et méthodologies ont contribué a des succès importants de la vision de machine dans la vie réelle. Nous énumérons ci-dessous quelques exemples de réussite (illustrés sur la figure 2.) Dans le secteur manufacturier, le commerce de détail, la conduite autonome, la sécurité, le divertissement et multimédia sur Internet.

- robots industriels avec capacité de reconnaissance d'objets peuvent remplacer l'homme dans les opérations qui nécessitent des manipulations très précises ou traitement de masse, c'est a dire, l'inspection automatique des circuits intégrés, prétraitement des aliments, la fabrication métallique.
- intelligents caméras installées sur des véhicules en charge la conduite sécuritaire en détectant les piétons qui traversent, en gardant une distance de sécurité des voitures a proximité.
- voitures autonomes tels que Google Car [1] ou ceux de la DARPA Urban Challenge [2] peut se conduire en toute sécurité sur des dizaines de miles dans les rues urbaines.
- détaillants obtiennent également de bénéficier de technologies de vision. En installant le long de voies de caisse caméras intelligentes [3], les articles dans les paniers des clients sont automatiquement détectés et reconnus ; la caissière n'a pas a déplacer les éléments de la corbeille pour codes a barres lecture, mais reçoit des informations de facturation a partir de caméras intelligentes.

1. `http://www.google.com/about/jobs/lifeatgoogle/self-driving-car-test-steve-mahan.html`
2. `http://www.torcrobotics.com/case-studies/darpa-urban-challenge`
3. `http://www.evoretail.com/`

(a) conditions d'éclairage faible.

(b) L'ensemble unifiée est différente de la somme des parties.



(c) Objets non rigides.

(d) Variantes intra-classe.

FIGURE 1 – Certains défis en vision de machine. En (a) est quelques captures inégale-exposition de visages ; ces images provoquent des difficultés de reconnaissance de visage depuis de nombreux détails sont perdus a cause de l'ombre. En outre, estompé les conditions d'éclairage et une surexposition aussi causer des difficultés similaires. Un autre défi est l'incapacité des algorithmes de vision industrielle dans la capture globale compréhension de l'image. L'exemple en (b) montre un chien avec le déplacement des jambes sur le terrain, mais même les meilleurs détecteurs d'objets ne peuvent pas reconnaître que nous faisons. Pour eux, cette image n'est pas plus que les segments noirs sur un fond blanc. Montré dans (c) sont le défi causés par des objets articulés. Des parties d'un tel objet sont mobiles de sorte qu'ils forment un bon nombre de positions relatives. Chaque objet articulé doit être suivi par un modèle spécialisé. La multiplicité des apparences en ce qui concerne le concept est illustré dans (d), bien que partageant les mêmes fonctionnalités, les quatre chaises propriétaire de tous les différents aspects, a savoir, la forme, le matériau, la structure et la couleur. Méthodes d'apprentissage actuelles sont encore loin d' abstraction conceptuelle.

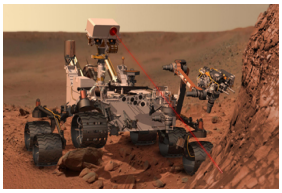
- vision de machine crée également des révolutions dans la guerre et l'exploration spatiale moderne. Drones (Unmanned Aerial Vehicle) remplacent les combattants humains contrôlée par des patrouilles et la recherche de tâches, les techniques de vision de machine aident le robot Curiosity naviguer Mars.
- La biométrie est très utile pour le contrôle des frontières, car il peut identifier avec précision des centaines de millions d'identités par des visages et des empreintes digitales correspondant ; cette technique d'adaptation est mis en ouvre par des algorithmes de vision de machine.
- Placer des caméras intelligentes sur les espaces publics tels que les aéroports et les gares aide a détecter les activités anormales ou bagages abandonnés, la surveillance automatique sur les autoroutes aider la régulation du trafic.
- Appareils photo numériques compacts offrent une meilleure qualité de photo en localisant les visages et détection des visages souriants, stations de jeux comme Kinect Xbox offre des jeux interactifs basés sur la localisation et la reconnaissance des parties du corps en utilisant des algorithmes efficaces de vision de machine.

2 Apprentissage automatique pour vision de machine

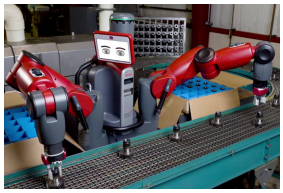
Au cours des années 1950 et 1960, les objets visuels sont souvent modélisés comme des primitives géométriques [Mundy 2006]. La popularité de cette modélisation pourrait être due a i) l'idée que la vision de machine est un processus inverse de l'infographie, et ii) la hausse de la logique formelle et l'intelligence artificielle. Dans l'ouvre de Robert [Roberts 1963], il propose de décrire les objets dans le monde réel comme des blocs simplifiées. En particulier, les objets ont été limités a des formes polyédriques sur un fond uniforme. Cette approche a ensuite été abandonné en raison de l'incapacité des primitives géométriques pour caractériser les objets dont les apparences et formes sont compliquées.

Apprentissage statistique devient une alternative. La méthode la plus largement utilisée est réseaux de neurones artificiels (ANN) [McCulloch 1943] et ses applications sont vastes, par exemple la reconnaissance optique de caractères [4], la reconnaissance du visage, et la reconnaissance de plaque [Rowley 1996, Draghici 1997, Lawrence 1997]. Le cœur de ANN est un réseau multicouche d'unités de neurones et de l'utilisation de l'algorithme de rétro-propagation afin d'apprendre les poids de connexion. Cependant, l'algorithme d'optimisation de rétro-propagation utilisé pour ANN s'avère inefficace s'il n'y a plus de trois couches d'un réseau. Les progrès récents de [Krizhevsky 2012, Bengio 2009] dans les méthodes d'optimisation ont relancé ANN et maintenant les structures de réseau profondes peut être optimisé de manière efficace avec plus de trois couches.

4. reconnaissance optique de caractères (OCR) `http://yann.lecun.com/exdb/Lenet/`



(a) Curiosité sur Mars



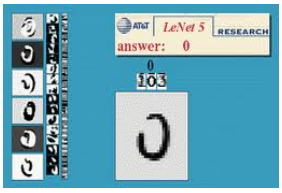
(b) robot autonome

(c) détection de visage

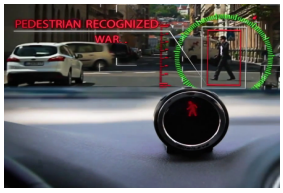
(d) Vérification des empreintes digitales

(e) Articulation suivi du corps

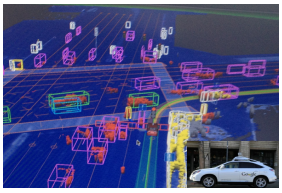
(f) La reconnaissance d'objet



(g) Reconnaissance optique de caractères



(h) Détection de piétons



(i) Driveless voiture

FIGURE 2 – Certaines applications commerciales de la vision de machine :(a) techniques de vision de machine ont été utilisés dans l'exploration de Mars Rover a lancé en 2003 et Curiosity Rover a lancé en 2011 `http://mars.nasa.gov/msl/` ; (b) Baxter robot peut apprendre a faire des missions simples telles que le transfert des pièces d'une ligne a, l'emballage des produits en boîtes, en inspectant les articles défectueux `http://www.rethinkrobotics.com/` ; (c) la plupart des caméras compactes peuvent détecter le visage humain et le sourire ; (d) les empreintes digitales scanner pour la biométrie identification `http://www.fulcrumbiometrics.com/` ; (e) Kinect détecte et suit les parties du joueur afin qu'il / elle peut interagir pleinement avec le jeu basé sur le mouvement du corps `http://www.xbox.com/kinect` ; (f) la caméra détecte automatiquement LaneHawk articles dans le panier entrant et facturer en conséquence a leur identification `http://www.evoretail.com/` ; (g) la reconnaissance optique de caracteres est l'un des premiers problèmes de vision de machine `http://yann.lecun.com/exdb/lenet/` ; (h) la technologie d'évitement de collision de Mobileye est capable de " interpréter " une scene en temps réel et de fournir aux conducteurs une évaluation immédiate `http://www.mobileye.com/`.

Comme ANN, le modèle d'apprentissage de Poggio [Riesenhuber 1999] est également inspiré du mécanisme de travail de cortex visuel humain. Leur approche, nommé *système biologiquement inspiré*, suppose que la chaîne de traitement dans notre cortex visuel peut être modélisée comme une hiérarchie de représentations plus en plus sophistiqués. L'intérieur de chaque cellule, au niveau le plus bas est la convolution (l'opérateur de SUM) entre les filtres (par exemple les filtres de Gabor directionnel) et l'image d'entrée. A un niveau supérieur, l'opérateur non linéaire MAX cherche pour la réponse la plus forte parmi les cellules dans le niveau inférieur. En alternant les deux mécanismes, le système réalise a la fois la spécificité de motif et l'invariance de la traduction et de mise a l'échelle.

L'apprentissage basé sur l'énergie - [Bakir 2007] a été un cadre efficace pour les méthodes paramétriques tels que les champs aléatoires conditionnels et les réseaux de Markov de marge maximale [Sutton 2012, Wallach 2004, Kindermann 1980]. Ce modele tient compte des dépendances entre les variables en associant une énergie scalaire a chaque observation des variables. Termes d'énergie sont conçus de telle sorte que leurs valeurs sont abaissées dans la mesure ou des prédictions plus corrects sont obtenus. Un grand avantage de ce modele est la flexibilité dans la conception termes d'énergie qui représentent diverses fonctions de bas niveau (couleur, texture, forme) ainsi que les relations sémantiques. L'apprentissage basé sur l'énergie - a été appliquée avec succès a la compréhension de l'image, la segmentation de la classe de l'objet et de l'image analyse [Tighe 2013, Chen 2011, Eigen 2012].

Parmi les modèles d'apprentissage statistique, l'une des méthodes les plus efficaces est Support Vector Machine (SVM) [Cortes 1995]. Il ya plusieurs raisons pour SVM a être largement utilisés dans les troubles de la vision : i) la méthode est pour que les données est conditionnée a une distribution gratuite de distribution ; ii) une bien bonne classificateur peut être obtenue avec une quantité limitée de données de formation, iii) la condition max marge garantit la généralisation du classificateur ; iv) noyau-SVM peut traiter les données non linéaire ; v) formule SVM est convexe. En outre, la propriété de la modularité de SVM, permet de devenir une technique prédéfini pour pratiques de classification de modèle.

3 Motivations et Contributions

Dans les petites et moyennes échelles, des problèmes de vision de machine sont généralement bien traités en utilisant des techniques d'apprentissage supervisé mentionnés ci-dessus. Une condition essentielle est que les données de formation doivent être suffisantes pour les algorithmes d'apprendre de bonnes classificateurs. Mais dans le long terme, les algorithmes d'apprentissage devraient être en mesure de faire face a l'absence de données de formation et d'exploiter les données non marqué qui est abondante. Ce mouvement est du a des changements importants de la façon dont les gens utilisent les technologies de nos jours. Par exemple, le Web est en train de changer de 1.0 à 2.0 et de personnes dans

le monde sont de plus en plus impliqués dans les réseaux sociaux. Ils ont été la création, ajout, et partager des images et des vidéos plus que jamais. Flickr [5], un site de photo d'hébergement et de partage, reçoit environ 60 millions de photos téléchargées par mois ; dans les trois ans, le partage photo en ligne Instagram [6] a enregistré plus de 130 millions d'utilisateurs qui ont téléchargé 40 millions de photos par jour.

Pour les algorithmes de vision pour attraper ces tendances, la recherche sur les systèmes de vision a grande échelle a été menée. Torralba et ses collègues ont recueilli une base de données de 80 millions d'images minuscules [7] dans 75 milliers noun mentions figurant dans la base de données lexicale Wordnet. Cette base de données vaste et diverse est une bonne source pour les algorithmes d'apprentissage machine a généraliser. Un autre exemple de base de données a grande échelle est ImageNet [8]. Cette base de données est organisée selon la hiérarchie de WordNet et contient 21 841 synsets et 14 millions d'images dans les totaux. ImageCLEF [9] est une autre récupération d'image a grande échelle et le défi d'annotation, qui se concentre sur plusieurs domaines d'image tels que des images médicales, des photos de consommateurs, photos des plantes, et la vision robotique.

Cependant, cette vague de grand données soulève une question plus importante :comment ces algorithmes peuvent encore bien performer avec moins de quantité de données d'apprentissage. Ce problème est pratique parce que le coût d'annotation est cher pour les bases de données a grande échelle. Comme les méthodes d'apprentissage supervisées ne sont plus appropriées, il existe dans l'apprentissage machine l'approche semi-supervisé, celui qui utilise des données a la fois marqués et non marqués pour l'apprentissage. Toujours a l'aide de données étiquetées comme l'apprentissage supervisé ne, apprentissage semi-supervisé utilise également les données non marquée qui peut présenter des informations sur la distribution de densité de données. Parmi les techniques d'apprentissage semi- supervisé, les inductifs et transductives sont les deux grandes approches. Les anciens induit a partir des données de formation d'une règle de décision qui s'applique a toutes les données de test invisibles. Ce dernier en déduit un étiquetage des données d'essai sur la base a la fois sur la formation et les données de test, puisque cet étiquetage n'est pas disponible ailleurs, sauf les données d'essai données, les classificateurs de formation n'est pas nécessaire. En outre, l'apprentissage transductive peut donner un étiquetage plus approprié par rapport a des données de test. Sur la base de cette approche, notre étude prend en compte : i) la façon d'améliorer l'inférence transductive lorsqu'il est combiné avec l'apprentissage du noyau, et ii) d'étendre son utilisation afin d'apprendre représentation sémantique et de faible dimension des bases de données d'image.

Pour le premier but, deux contributions sont faites. Dans la section 3.1, une

5. www.flickr.com
6. http://instagram.com
7. http://groups.csail.mit.edu/vision/TinyImages/
8. http://www.image-net.org/
9. http://imageclef.org/

nouvelle formulation de l'apprentissage du noyau transductive est proposé qui est représenté a au moins compétitive face a des méthodes d'apprentissage classiques. Dans la section 3.2, la méthode proposée est étendu a de nombreuses applications dans lesquelles leurs spécificités sont utilisés pour adapter la formule originale. Pour le deuxième objectif, la troisième contribution est faite, un nouvel algorithme d'apprentissage sous-espace sémantique est introduit dans la section 3.3 comme le noyau du modèle de recherche mentale introduit dans la même section.

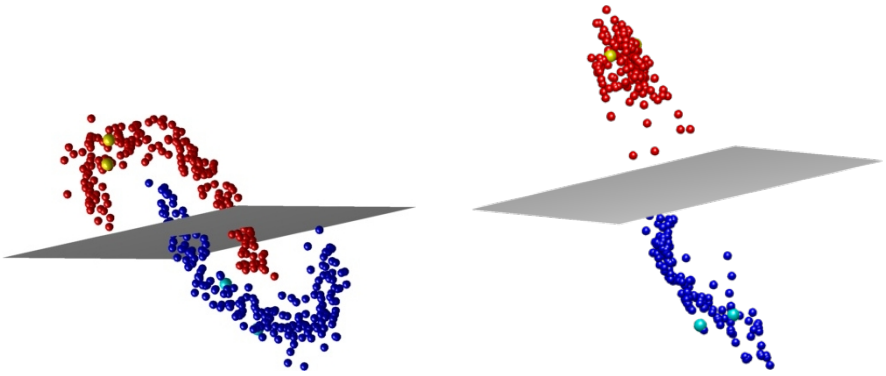
3.1 Apprentissage du noyau transductive

Notre première contribution est une méthode de transductive roman pour la carte de noyau apprentissage. Lorsque les données marqué est rare ou cher, l'apprentissage d'une bonne classificateur devient difficile. Transductive apprentissage est particulièrement adapté a de telles situations. L' objectif de l'apprentissage est transductive de déduire les étiquettes de données de test et de ne pas apprendre une règle de décision générale pour les données invisibles sans l'aide de données de test. Par conséquent, les approches transductives considèrent a la fois les données marquées et non marquées dans l'inférence. Sur la base de cette approche, notre carte de noyau algorithme d'apprentissage peut mieux exploiter la structure topologique des données, qui permet de diffuser des informations de l'étiquette des données étiquetées a celui non marqué.

En fait les techniques d'inférence transductives deviennent la norme dans l'apprentissage de la machine en raison de leur succès relatif dans la résolution de nombreuses applications dans le monde réel [Joachims 1999, Duchenne 2008, Liu 2009, Joachims 2003]. Parmi eux, les méthodes a noyaux sont particulièrement intéressants, mais leur succès reste très dépendante du choix de noyaux. Ce dernier est généralement fabriqué a la main ou conçu afin de mieux saisir la similitude dans les données de formation. L'aspect novateur de notre méthode comprend un nouvel algorithme d'apprentissage de transductive pour la conception du noyau et de la classification.

Différente de la connexes travaux [Maji 2008, Joachims 2002, Asa 2008, Bach 2004, Wu 2006, Rakotomamonjy 2008, Bach 2008, Sonnenburg 2006], nous n'adoptons pas des tours du noyau [Scholkopf 2001] mais apprenons une carte explicite de noyau basé sur la structure topologique de données étiquetées et non étiquetées. Notre approche est donc *sans modèle* ce qui signifie pas restreinte aux noyaux prédéfini. Elle conduit aussi a de meilleures performances de généralisation.

Mathématiquement, notre approche est basée sur la minimisation d'une fonction d' énergie de mélange i) un terme de reconstruction $E_{\text{données}}(\mathbf{X}, \mathbf{B\Phi})$ qui factorise une matrice de données d'entrée $\mathbf{X}$ en tant que produit d'un savant dictionnaire $\mathbf{B}$ et un noyau carte appris $\mathbf{\Phi}$, ii) un terme de fidélité $E_{\text{label}}(\mathbf{Y}, f(\mathbf{\Phi}))$ qui assure cohérence des prévisions de l'étiquette $f(\mathbf{\Phi})$ avec celles prévues dans un rez-de-vérité $\mathbf{Y}$, iii) un terme de régularité $\sum_{(i,j)\in\mathcal{E}} E_{\text{lisser}}(f(\mathbf{\Phi}_i), f(\mathbf{\Phi}_j))$ qui garantit des étiquettes similaires pour les données voisin qui est prise a partir du bord mis en $\mathcal{E}$ de la k - plus proche graphe de voisinage $\{\mathcal{V}, \mathcal{E}\}$ des données,



(a) L'ensemble de données de jouet-dessus n'est pas séparable de $\mathbb{R}^2$. Compte tenu de la formation (points jaunes) et le test (points de données cyan), un hyperplan ne peut pas séparer les deux classes.

(b) La carte de noyau linéarisé appris par notre méthode. La répartition des données permet de diffuser des informations de l'étiquette de la marqué a celles non marquées (points rouges et bleus).

FIGURE 3 – Illustration of learning the transductive kernel map with toy data.

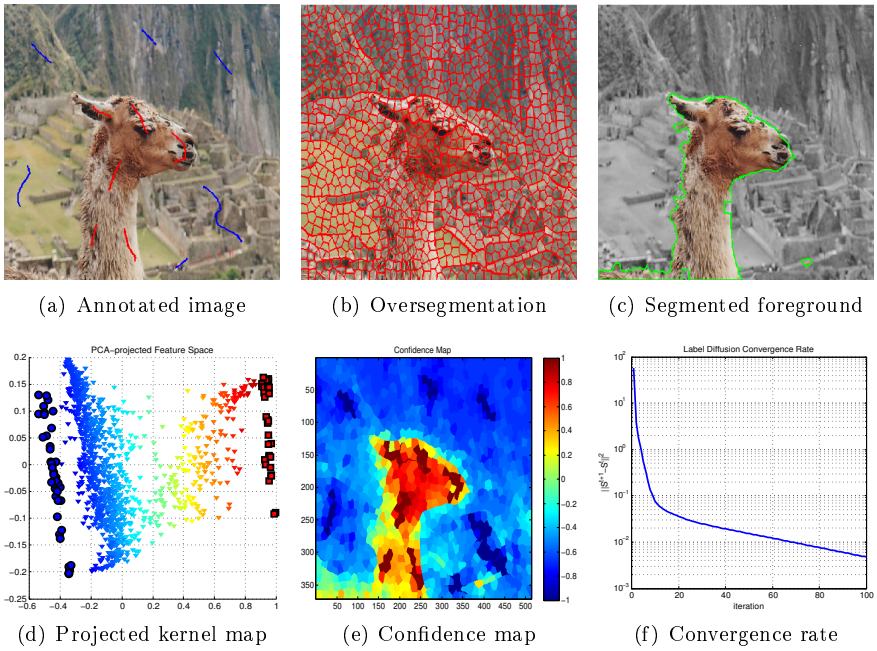
et iv) regularizers $\psi(f, \mathbf{\Phi})$ qui limitent la complexité de classificateur et le rang de la carte de noyau. Le problème de minimisation décrit ci-dessus admet la forme générique suivante et sera précisé dans les chapitres suivants :

$$\min_{f,\mathbf{\Phi},\mathbf{B}} \left\{ E_{\text{data}}(\mathbf{X}, \mathbf{B\Phi}) + E_{\text{label}}(\mathbf{Y}, f(\mathbf{\Phi})) + \sum_{(i,j)\in\mathcal{E}} E_{\text{smooth}}(f(\mathbf{\Phi}_i), f(\mathbf{\Phi}_j)) + \psi(f, \mathbf{\Phi}) \right\}. \tag{1}$$

La résolution de ce problème de minimisation, il est possible d'apprendre a la fois un critère de décision et une carte de noyau qui garantit la séparabilité linéaire dans un espace de grande dimension et de bonnes performances de généralisation. Des expériences menées sur des classes d'objets segmentation (Fig. 4) montrent des améliorations par rapport a la ligne de base ainsi que des travaux connexes sur la base de données de VOC difficile [Everingham 2010].

3.2 Apprentissage du noyau transductive régularisé

Comme la deuxième contribution, nous étendons le cadre du plan du noyau apprentissage a l'annotation d'image et de la scène l'interprétation des images. Pour la première demande, une formule multi- classe est dérivée basée sur le modèle de classification binaire, qui est mentionné dans la section ci-dessus. Depuis étiquette multiplicité mene a une feuille de noyau partagé entre classificateurs, regularizers supplémentaires peuvent être ajoutés a la formule de base afin d'amplifier les dépendances entre les classes. Dans le cas de l'annotation d'image, nous concevons un régularisateur qui applique co-occurrence d'étiquettes dans chaque image sur la base des statistiques de co-occurrence de données de formation. Par conséquent, a la fois la douceur et l'étiquette de



(a) Annotated image (b) Oversegmentation (c) Segmented foreground

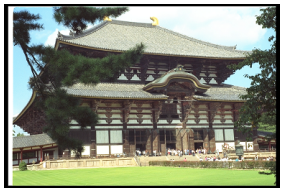
(d) Projected kernel map (e) Confidence map (f) Convergence rate

FIGURE 4 – Le pipeline de traitement de l'objet interactif segmentation. Initialement l'utilisateur annote quelques régions représentatives de premier plan de & arrière-plan dans l'image (a) ; l'image est annoté sur-segmentée en superpixels (b) ; sur la base des données dont ceux marqués sont les superpixels annotés et ceux non marqués sont les superpixels annotées , notre apprentissage du noyau transductive déduit l' objet de premier plan complet comme indiqué dans (c). En (d) est la visualisation de la carte de noyau appris ; la carte du noyau est projeté dans l'espace 2D utilisant PCA qui complètent points bleus indiquent fond données étiquetées et les points rouges carrés indiquer les données de premier plan étiquetés. Données non marqués (points de triangle) sont affectées de couleur par rapport à leurs distances relatives par rapport aux données étiquetées positives et négatives. Fig.(e) montre la carte de prédiction dans lequel les couleurs chaudes ou froides correspondent à des prévisions plus sûres de la classe positive ou négative respectivement. Le taux de ce processus d'inférence de convergence est représenté en (f).

dépendance termes de diffuser des informations de l'étiquette non seulement entre les images climatisées sur les étiquettes individuelles, mais aussi entre les étiquettes dans chaque image. En d'autres termes, le terme de finesse soutient inter-images similitude visuelle et le terme étiquette - dépendance prend en charge intra-image dépendances statistiques entre les étiquettes.

Pour cette dernière application, qui est l'interprétation des images, nous utilisons l'apprentissage du noyau transductive pour segmenter et reconnaître

(a) **Avant** : ocean, coral, reefs, people, water ; **Apres** : coral, ocean, reefs, *fish*, fan.



(b) **Avant** : tree, water, buildings, sky, mountain ; **Apres** : tree, roofs, water, arch, *temple*.

(c) **Avant** : tree, sky, sunset, kauai, train ; **Apres** : kauai, locomotive, train, railroad, sunset.

FIGURE 5 – Des exemples de l'ensemble de données qui démontrent comment Corel5K étiquette dépendance pourrait être utilisé pour améliorer la tâche d'annotation. Chaque exemple donne les résultats d'annotation avant et après l'addition du marqueur de dépendance ; mots-clés soulignées sont les bonnes étiquettes en italique mots-clés sont les bonnes découvertes en raison du modèle étiquette-dépendance. Dans la (a), l'apparition de *coral*, *ocean*, et *reefs* signifie une forte probabilité que *fish* est également présent dans la scène. Dans la (b), la co-occurrence de labels tels que *roof* et *arch* conduit à la présence de *temple*. Dans la (c), l'apparition de deux fausses étiquettes *kauai* et *train* favorise la présence d'autres fausses étiquettes *locomotive*, *train* et *railrooad* trop.

des objets visuels dans une scène. Nous réutilisons la formule multi- classe afin d' attribuer des étiquettes appropriées pour chaque superpixel image test dans lesquelles une étiquette de l'information est tirée de superpixels marquées ayant l'aspect visuel similaire. Depuis la variabilité visuelle peut provoquer une fausse étiquette, nous ajoutons une nouvelle régularisateur qui exploite significations référentielles de superpixels. Ces caractéristiques contextuelles non seulement aident a reconnaître superpixels dont les apparences ne sont pas assez discriminant mais également l'étiquetage prévu plus cohérente a l'égard de règles contextuelles implicites par la vision humaine.. L'application est représenté sur la Fig 5 et 6.

3.3 Sémantique subspatial apprentissage

La troisième contribution est de concevoir un nouvel algorithme d'apprentissage sous-espace dédié a la recherche mentale. Les moteurs de recherche visuelle actuels utilisent essentiellement des requêtes basées sur des textes a la recherche de contenu visuel. Comme les données multimédia sur Internet est en pleine explosion (c'est a dire, YouTube, Flickrs, Instagram, Facebook), il est presque impossible de donner annotation a chaque image ou vidéo. Ainsi la recherche sémantique de données visuelle est la solution la plus probable pour la recherche multimédia évolutive. Nous utilisons la visualisation de bases de données [Heesch 2008, Schaefer 2010] comme une alternative, qui s'appuie sur les données cartographiques de haute aux espaces de faibles dimensions ou les

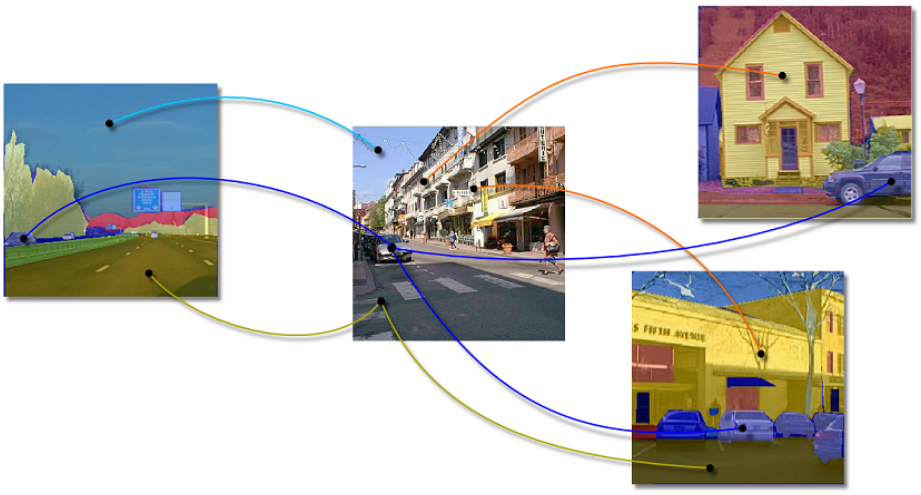


FIGURE 6 – Compréhension de la scène le probleme de la localisation et de classification des objets visuels dans une image de la scène dans les catégories connues. Étant donné une image de requête, nous mettons a K images les plus similaires de la base de données étiquetées. Nous oversegment alors ces $(K + 1)$ images et obtient superpixels. En connectant superpixels en fonction de leurs similitudes visuelles, les informations de l'étiquette peut être transféré de la marqué a celles non marquées. En raison de l'étape d'extraction, objets inconnus dans l'image de test sont censés trouver leurs exemples similaires dans les images d'apprentissage, mais ambiguïtés visuelles sont inévitables. Afin de réduire ces erreurs, relations contextuelles sont prises en compte. En termes de statistiques, ces relations sont les corrélations de l'étiquette ou de probabilités a priori sur la disposition spatiale des images de la scène, par exemple, les voitures sont dans les rues, les fenêtres ne peuvent pas apparaître sans bâtiments, le ciel est toujours au-dessus des routes, etc

données peuvent être facilement repérés et explorés par l'utilisateur. En dépit de l'extension des techniques non linéaires de réduction de la dimensionnalité [Tenenbaum 2000, Roweis 2000, Belkin 2001], leur succès dans la base de données la visualisation des images est limitée a des ensembles de données avec une sémantique bien contrôlées, comme des poses de visages ou des distorsions de chiffres ; que ces techniques sont totalement sans surveillance, leur application aux bases de données génériques [Schaefer 2010, Rubner 2001] produit dimensions sémantique moins qui sont difficiles a interpréter et a explorer (voir la figure 8).

Notre contribution est de présenter un algorithme de recherche mentale roman basé sur l'apprentissage sémantique de sous-espace. Ce dernier est conçu en utilisant un nouveau principe, que décompose K sémantique de données d'image $\mathbf{X} \in \mathbb{R}^{n \times m}$ et les cartes a partir d'un espace ambiant initiale $\mathbb{R}^n$ (liée a des caractéristiques visuelles de bas niveau, y compris la texture, la couleur et

la forme) a un sous-espace de sortie de $\mathbb{R}^K$ engendré par K bien défini bases sémantiques. Nous rejetons ce problème que l'optimisation de la programmation quadratique convexe, contraint a un simplex engendré par quelques (purs) endmembers sémantiques, c'est-a-

$$\begin{array}{lll} \min\limits_{\mathbf{\Phi}\geq 0} & \sum_{(i,j)\in\mathcal{E}} E_{\text{smooth}}(\mathbf{\Phi}_i, \mathbf{\Phi}_j) & \\ \text{s.t} & \mathbf{\Phi}_i = \mathbf{Y}_i & i = 1, \ldots, \ell \\ & \sum_{k=1}^{K} \mathbf{\Phi}_{ki} = 1 & i = \ell+1, \ldots, m \end{array} \tag{2}$$

Supposons que l'échantillon de données donnée est prise en charge par un collecteur, le problème d'optimisation (2) préserve la topologie des données quand elle est mappée de l'espace ambiant dans l'espace sémantique. La sémantique représentation $\mathbf{\Phi}_i$ de chaque échantillon d'entrée $\mathbf{X}_i$ peut être considérée comme le vecteur d'adhésion par rapport a K sémantique définies. Les premiers ℓ contraintes d'égalité, dans lequel ℓ est le nombre de données étiquetées, état que la nouvelle représentation $\mathbf{\Phi}_i$'s d'échantillons marqués est égal étiquette vecteurs $\mathbf{Y}_i$ s. Nous appelons ces ℓ samples *endmembers* parce que chaque échantillon représente un must uniques sémantique ; données non marquées, en revanche, peuvent être endmembers ou des mélanges de plusieurs sémantique (voir figure 7).

L'avantage de l'approche proposée est double : d'une part, il permet de réduire de manière significative la dimension de l'espace d'entrée (ce qui est difficile a explorer ou a visualiser), et d'autre part, il apprend caractéristiques qui sont sémantiquement interprétable, a savoir, leur valeurs sont fortement corrélées avec la sémantique définie. Ainsi, la recherche d'une cible mentale réduit simplement a la numérisation et les données de ciblage en fonction de leurs coordonnées dans le sous-espace sémantique appris.

4 Aperçu de la thèse

Le reste de la thèse est organisé comme suit.

- Chapitre 2 présente les concepts de base de la théorie de l'apprentissage et revue de la littérature d'une sélection de techniques d'apprentissage automatique qui sont pertinents pour notre travail.
- Chapitre 3 présente notre première contribution qui est une méthode d'apprentissage roman carte de noyau. Le chapitre décrit les étapes pour construire la formulation complète, des procédures d'optimisation, les garanties théoriques et expériences. Interactive segmentation de l'objet est utilisé pour démontrer l'idée.
- Chapitre 4 et 5 présenter notre deuxième contribution, qui est le prolongement de la formule binaire proposé au chapitre 3 des problèmes de multiclasse. Les nouvelles formules sont appliquées a l'image annotation et l'interprétation de scène. Pour chacun des problèmes, regularizers spécifiques sont conçus pour exploiter la connaissance préalable de son domaine de données.

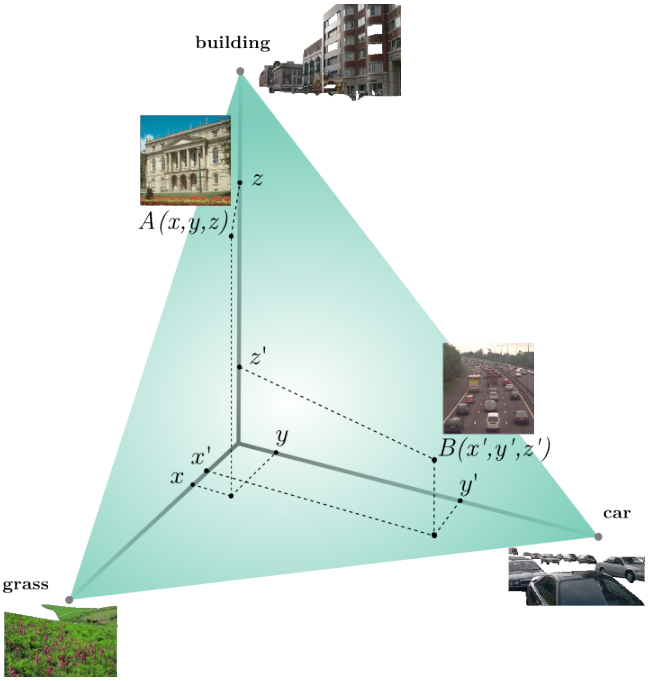


FIGURE 7 – Notre méthode apprend une représentation basée simplex donné une base de données de l'image et K sémantique prédéfinis. Étant donné que $K = 3$ et *building*, *car*, et *plant* sont ces sémantique, nous apprenons une nouvelle représentation des données dans le $(K-1)$ sous-espace de dimension. Cette nouvelle représentation cartes données étiquetées, également appelés endmembers, dans les sommets de l'unité $(K-1)$ - simplex tandis que les données non étiquetées sont mappés a différents endroits dans la surface simplex, en fait leurs coordonnées sont déterminées sur la base de la similitude de leur sémantique sont contenus par rapport a chacun de la sémantique prédéfinies. Par exemple, a l'emplacement $A(x, y, z)$, l'image contient seulement *building* et *plant*. Par conséquent, il est mappé a proximité de la *building* sommet. A l'emplacement $B(x', y', z')$, l'image contient beaucoup de voitures, donc il est certainement mappé a proximité de la *car* sommet. La représentation apprise permet a l' utilisateur de rechercher une cible mental juste un pointeur en la glissant le long de la surface de telle sorte que son simplex valeurs de coordonnées a peu près égales aux valeurs d'appartenance de la cible mental. Images autour de ce pointeur sont susceptibles de contenir d'image (s) d'intérêt. Comparez notre modèle simplex avec les autres modèles de visualisation représentées sur la figure 8.

(a) Corel dataset

(b) Navigateur d'image hiérarchique Google Swirl

FIGURE 8 – Les techniques de visualisation de base de données. En (a) est un nuage image visualisée dans l'espace tridimensionnel [Rubner 2001] ; la visualisation arborescente de Google remous en (b) est une alternative.

- Chapitre 6 est dédié a notre troisième contribution. Le chapitre contient discussion sur les limites actuelles de la recherche visuelle, la formulation de notre méthode, les algorithmes d'optimisation pour les ensembles de données a grande échelle, et des expériences de visualisation de données, classement de l'image, et le retour de pertinence.
- Chapitre 7 conclut la thèse d'une révision des contributions et des discussions sur les perspectives de thèse.

Références

[Asa 2008] Asa, Ong Cheng Soon, Sonnenburg Sören, Schölkopf Bernhard and Rätsch Gunnar Ben-Hur. *Support Vector Machines and Kernels for Computational Biology.* PLoS Comput Biol, vol. 4, no. 10, 10 2008.

[Bach 2004] Francis R. Bach, Gert R. G. Lanckriet and Michael I. Jordan. *Multiple kernel learning, conic duality, and the SMO algorithm.* In ICML, 2004.

[Bach 2008] Francis R. Bach. *Consistency of the Group Lasso and Multiple Kernel Learning.* Journal of Machine Learning Research, vol. 9, pages 1179–1225, 2008.

[Bakir 2007] Gükhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar and S. V. N. Vishwanathan. Predicting structured data (neural information processing). The MIT Press, 2007.

[Belkin 2001] M. Belkin and P. Niyogi. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering.* In NIPS, pages 585–591, 2001.

[Bengio 2009] Yoshua Bengio. *Learning Deep Architectures for AI.* Foundations and Trends in Machine Learning, vol. 2, no. 1, pages 1–127, 2009.

[Biederman 1982] Irving Biederman. *Scene perception : detecting and judging objects undergoing relational violations.* Cognitive Psychology, vol. 14, pages 143–177, 1982.

[Boden 2006] Margeret A. Boden. Mind as machine : A history of cognitive science. Oxford University Press, Oxford, England, 2006.

[Chen 2011] Xi Chen, Arpit Jain, Abhinav Gupta and Larry S. Davis. *Piecing together the segmentation jigsaw using context.* In CVPR, pages 2001–2008, 2011.

[Cortes 1995] Corinna Cortes and Vladimir Vapnik. *Support-vector networks.* Machine Learning, vol. 20, no. 3, pages 273–297, 1995.

[Draghici 1997] Sorin Draghici. *A neural network based artificial vision system for licence plate recognition.* International Journal of Neural Systems, vol. 8, no. 01, pages 113–126, 1997.

[Duchenne 2008] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce and F. Segonne. *Segmentation by transduction.* In IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[Eigen 2012] David Eigen and Rob Fergus. *Nonparametric image parsing using adaptive neighbor sets.* In CVPR, pages 2799–2806, 2012.

[Everingham 2010] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The Pascal Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, June 2010.

[Heesch 2008] Daniel Heesch. *A survey of browsing models for content based image retrieval.* Multimedia Tools Appl., vol. 40, no. 2, pages 261–284, 2008.

[Hubel 1988] David H. Hubel. Eye, brain, and vision. W H Freeman & Co, NewYork, 1988.

[Joachims 1999] T. Joachims. *Transductive Inference for Text Classification using Support Vector Machines.* In ICML, pages 200–209, 1999.

[Joachims 2002] T. Joachims. Learning to classify text using support vector machines – methods, theory, and algorithms. Kluwer/Springer, 2002.

[Joachims 2003] Thorsten Joachims. *Transductive Learning via Spectral Graph Partitioning.* In In ICML, pages 290–297, 2003.

[Kindermann 1980] Ross Kindermann, James Laurie Snell*et al.* Markov random fields and their applications, volume 1. American Mathematical Society Providence, RI, 1980.

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks.* In NIPS, pages 1106–1114, 2012.

[Lawrence 1997] Steve Lawrence, C Lee Giles, Ah Chung Tsoi and Andrew D Back. *Face recognition : A convolutional neural-network approach.* Neural Networks, IEEE Transactions on, vol. 8, no. 1, pages 98–113, 1997.

[Liu 2009] Wei Liu and Shih-Fu Chang. *Robust multi-class transductive learning with graphs.* In CVPR, pages 381–388. IEEE, 2009.

[Livingstone 2008] Margaret S. Livingstone. Vision and Art : The Biology of Seeing. Abrams, 2008.

[Maji 2008] S. Maji, A-C. Berg and J. Malik. *Classification using intersection kernel support vector machines is efficient.* In CVPR, 2008.

[Marr 1982] David Marr. Vision : A computational investigation into the human representation and processing of visual information. Henry Holt and Co., Inc., New York, NY, USA, 1982.

[McCulloch 1943] WarrenS. McCulloch and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115–133, 1943.

[Metzger 2006] Wolfgang Metzger. Laws of seeing. The MIT Press, September 2006.

[Mundy 2006] Joseph L. Mundy. *Object Recognition in the Geometric Era : A Retrospective.* In Toward Category-Level Object Recognition, pages 3–28, 2006.

[Murphy 2004] Gregory Murphy. The big book of concepts. The MIT Press, January 2004.

[Rakotomamonjy 2008] Alain Rakotomamonjy and Francis R Bach. *SimpleMKL.* Journal of Machine Learning Research, pages 1–34, 2008.

[Riesenhuber 1999] Maximilian Riesenhuber and Tomaso Poggio. *Hierarchical models of object recognition in cortex.* Nature Neuroscience, 1999.

[Roberts 1963] Lawrence G. Roberts. Machine perception of three-dimensional solids. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963.

[Roweis 2000] S-T. Roweis and L-K. Saul. *Nonlinear Dimensionality Reduction by Locally Linear Embedding.* Science, vol. 290, pages 2323–2326, 2000.

[Rowley 1996] Henry A. Rowley, Shumeet Baluja and Takeo Kanade. *Neural Network-Based Face Detection.* In CVPR, pages 203–208. IEEE Computer Society, 1996.

[Rubner 2001] Y. Rubner and C. Tomasi. Perceptual Metrics for Image Database Navigation. Springer, 2001.

[Schaefer 2010] G. Schaefer. *A next generation browsing environment for large image repositories.* Multimedia Tools and Applications, vol. 47, pages 105–120, 2010.

[Scholkopf 2001] Bernard Scholkopf and Alexander J. Smola. Learning with kernels. The MIT Press, 2001.

[Sonnenburg 2006] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer and Bernhard Schölkopf. *Large Scale Multiple Kernel Learning.* Journal of Machine Learning Research, vol. 7, pages 1531–1565, 2006.

[Sutton 2012] Charles A. Sutton and Andrew McCallum. *An Introduction to Conditional Random Fields.* Foundations and Trends in Machine Learning, vol. 4, no. 4, pages 267–373, 2012.

[Tenenbaum 2000] J-B. Tenenbaum, V. de Silva and J-C. Langford. *A Global Geometric Framework for Nonlinear Dimensionality Reduction.* Science, vol. 290, no. 5500, pages 2319–2323, 2000.

[Tighe 2013] Joseph Tighe and Svetlana Lazebnik. *Superparsing - Scalable Nonparametric Image Parsing with Superpixels.* International Journal of Computer Vision, vol. 101, no. 2, pages 329–349, 2013.

[Wallach 2004] Hanna M Wallach. *Conditional random fields : An introduction.* Technical Reports (CIS), page 22, 2004.

[Wu 2006] Mingrui Wu, Bernhard Schölkopf and Gokhan Bakir. *A Direct Method for Building Sparse Kernel Learning Algorithms.* Journal of Machine Learning Research, vol. 7, pages 603–624, 2006.

THÈSE DE DOCTORAT DE
TELECOM PARISTECH

Spécialité

Signal et Images

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Dinh-Phong VO

Pour obtenir le grade de

DOCTEUR de TELECOM PARISTECH

Sujet de la thèse :

Inférence transductive pour l'interprétation et la recherche d'images

devant le jury composé de :

M. Hichem SAHBI	Directeur de thèse
Mme. Jenny BENOIS-PINEAU	Rapporteur
M. Chaabane DJERABA	Rapporteur
M. Matthieu CORD	Examinateur
M. Frederic JURIE	Examinateur
M. Jean-Marc OGIER	Examinateur

Abstract

Dans cette thèse, on s'intéresse à l'apprentissage automatique pour traiter deux problèmes fondamentaux en vision par ordinateurs. Le premier concerne l'interprétation d'images qui consiste à classer des images ou des objets en catégories. Les techniques classiques sont généralement inductives et exigent des données d'apprentissage étiquetées afin d'apprendre explicitement des classifieurs. Dans certaines applications, les données d'apprentissage étiquetées sont rares ce qui affecte les capacités de généralisation des classifieurs sous-jacents. Dans cette thèse on s'intéresse à l'apprentissage transductif qui vise à estimer la réponse d'un classifieur implicite sur un ensemble fini incluant à la fois les données d'apprentissage et de test.

On présente d'abord un nouveau cadre d'apprentissage transductif des noyaux pour l'interprétation des images. Cette méthode, contrairement aux noyaux classiques, apprend une projection explicite des noyaux, en exploitant la topologie des données d'apprentissage et de test. Le problème d'optimisation sous-jacent vise à minimiser une énergie mélangeant i) un terme de reconstruction, qui décompose une matrice des données en un produit impliquant un dictionnaire et une nouvelle représentation liée au noyau appris, ii) un terme d'attache aux données qui assure la consistance des étiquettes inférées par rapport à celles des données d'apprentissage et iii) un terme de régularisation qui garantit des étiquettes similaires pour des données semblables. La représentation du noyau et le critère de décision obtenus garantissent la séparabilité linéaire des données et de bonnes performances de généralisation. En partant de cette formulation, on propose une extension qui permet d'exploiter les dépendances contextuelles et les liens sémantiques entre les catégories d'images afin d'améliorer encore plus les performances de notre méthode d'annotation et d'interprétation des images. Cette extension a été motivée par des expériences en psychologie, qui montrent que les informations contextuelles sont essentielles et permettent de faciliter la reconnaissance d'objets chez les humains.

Le deuxième problème abordé dans la thèse concerne la recherche mentale dans les bases d'images. Au départ, on rappelle les limites des paradigmes de recherche classiques (basés sur les mots clés, exemples visuels et requêtes par croquis) dans l'interprétation des requêtes mentales des utilisateurs ; notamment lorsque les cibles mentales des utilisateurs sont difficiles à exprimer avec des mots clés ou lorsque les exemples des requêtes ne sont pas disponibles. La solution alternative proposée construit une représentation qui préserve la topologie globale des données en les projetant dans un espace Euclidien exprimé à travers une base sémantique. L'avantage de la méthode est double ; d'une part elle permet de réduire significativement la dimension des données, et d'autre part, la méthode permet de définir une nouvelle représentation des données qui est plus facile à exploiter par l'utilisateur afin de retrouver sa cible. Ainsi, retrouver une cible mentale revient simplement à scanner et pointer les données selon leurs coordonnées dans l'espace sémantique appris. Les expériences effectuées en visualisation, ordonnancement et recherche d'images avec

contrôle de pertinence, sur des bases génériques, montrent que l'approche proposée est effective.

Abstract

In this thesis, we use machine learning in order to tackle two fundamental problems of computer vision. The first one is image interpretation which consists in classifying images and objects into categories. Conventional inductive learning models require some training data from which classifiers are learned. If training data is scarce, classifiers hardly generalize well to test data. We are interested in transductive learning - the approach that aims to estimate the response of an implicit classifier at particular test points using both training and test data.

We first introduce a new transductive kernel learning framework for image interpretation. Our method, in contrast to many usual kernels, learns an explicit kernel map based on topological structure of both training and test data. The underlying optimization problem minimizes an energy function mixing i) a reconstruction term that decomposes a matrix of input data as a product of a learned dictionary and a kernel map ii) a fidelity term that ensures consistent label predictions with respect to those provided by training data and iii) a smoothness term which guarantees similar labels for neighboring data. The resulting decision criterion and the new kernel map guarantee the linear separability of training data and good generalization performance. Based on this formulation, we also study how to harness contextual dependencies between categories into images and how to use their semantic relationships during inference in order to further improve image annotation and scene understanding performances. This extension was motivated by experiments in psychology, which have shown that contextual information includes important cues for human vision in order to recognize objects effortlessly.

The second fundamental problem is mental search ; we address the limitation of current multimedia search paradigms (based on keywords, image examples, and sketches) in interpreting mental targets of users, especially if those targets are difficult to express verbally or visual examples are not ready to hand. We introduce a novel alternative solution which builds a mapping that preserves the global topology of the input data while associating them into an Euclidean subspace spanned by well defined semantics. The advantage of the method is twofold. On the one hand, it significantly reduces the dimensionality of the data ; on the other hand, it defines a new data representation which is more friendly and easy to use. Thereby, searching for a mental target simply reduces to scanning and targeting data according to their coordinates in the learned semantic subspace. Quantitative evaluations in data visualization, image ranking and retrieval with relevance feedback, using generic image databases, show that the proposed method is effective.

Acknowledgments

First and foremost, I enthusiastically thank my advisor Hichem Sahbi for his invaluable support to my research and for patient correction of my work. From my first days until these past three years, I have been learning a lot from him not only knowledge but also profession in research. He is knowledgeable, diligent, and enthusiastic ; his advices and instructions have been so valuable for me to accomplish my thesis, and for my future scientific career as well. I think Hichem is an advisor that any student wants to work with and it is my luck to be advised by him.

I would like to appreciate professors Shin'ichi Satoh, Duong Nguyen-Vu, Le Hoai-Bac, and Le Dinh-Duy who supported me to be a doctorant.

I would like to give thanks to professors : Jenny Benois-Pineau, Chaabane Djeraba, Matthieu Cord, Frederic Jurie, and Jean-Marc Ogier for accepting to be members of my PhD jury. I also appreciate professor Michel Roux and professor Hugues Talbot for being juries in my mid-term evaluation.

Working here in TSI department has been always one of the most memorable experiences in my life. I would like to express my most sincere gratitude to all members of the department for a comfortable working atmosphere with lots of support they have daily shared with me. I am happy with the time we worked together in 46 rue Barrault : Pierre, Nicolas, Ling, Eric, Kevin, Geoffroy, and Ana.

A special thank and much gratitude to my parents, who now are living in Viet Nam. The PhD time is tough ; nevertheless they share joyful moments with me and encourage me during frustrating times. Sharing happy moments with me in Paris are good friends such as Nga, Giang, T.Trung, N.Trung, Duc, Van and many more.

Lastly, to all of those acknowledged here, friends, teachers and colleagues, thank you for being you and letting me be a part of your lives.

Table des matières

Chapitre 1

Introduction

From 1950's, researchers have tried inventing thinking machines. At that time, Alan Turing introduced the Turing test. The test checks the ability of machines to perform intelligent behaviors and reasoning such that they are indistinguishable from those of a human. The test soon becomes an essential concept in artificial intelligence (AI). Many of AI founders at that time were optimistic about the prospect of AI that machines could pass the test within 20 years. Until now, after more than 60 years, the dream about thinking machines is still elusive. Though achievements of AI are just at baby steps in reproducing human intelligence, they have opened a new era for information technology.

Being a subfield of AI, machine learning aims to design algorithms that automatically improve their behaviors through experience ; technically, machine learning is defined as a set of statistical techniques specialized for high dimensional and massive data. From 1990's, statistical machine learning tools are popular in solving specific problems of AI. This is indeed a remarkable milestone in the development of AI. Although the role of statistical approaches has been controversial, they have brought many successes recently. Various applications of artificial intelligence have been invented and developed for daily use such as Google's multi-language translation, speech recognition in Apple's Siri, gesture recognition in Microsoft's Kinect. Machine learning is one of key factors in those successful stories.

Among problems of machine learning, we are interested in computer vision. Computer vision aims to reproduce the astonishing cognition ability of human. Due to the rapid increase of multimedia content on the Internet and deeper interactions between human and information technologies, applications of computer vision come to every corner of the modern life. In this thesis we introduce our studies about novel machine learning methods in solving two classical problems of computer vision : image interpretation and search.

In our studies, algorithms are designed in order to learn representations of visual data that promote image interpretation and search even with insufficient amount of training data. This introductory chapter is dedicated for discussions about contexts, motivations as well as contributions of our research. In Section 1.1 we discuss further the chronological development, fundamental challenges, as well as achievements of computer vision. In Section 1.2 we revise the role of machine learning in solving some computer vision problems. Our motivation as well as contributions are introduced in Section 1.3. The final Section 1.4 explains the organization of the thesis.

1.1 Introduction to Object Recognition

Established in the early 1960s, computer vision aims at creating algorithms that replicate human ability in perceiving and recognizing the visual world. Ambitious objectives of computer vision include detecting, recognizing and interpreting visual objects in images. In the early time, it was popular to see image interpretation problem as an inverse process of computer graphics rendering. In particular, a computer graphics pipeline consists of rendering three dimensional (3D) objects in the world coordinate and covering them with materials and illumination ; these objects are finally projected to a two dimensional (2D) screen. Object interpretation, by contrast, recovers objects from 2D images. This perspective of a vision problem had been the main stream in the early years of its development. Based on findings about the organization of the human visual cortex [Hubel 1988], a visual cognition problem is presumed as a three-stage process : early vision, mid-level vision, and high-level vision. The early stage comprises filtering techniques used to detect visual primitives such as edges, colors, textures. The mid-level stage processes bigger entities such as image patches and regions. The high-level stage infers semantics from mid-level features such that the whole image forms a coherent meaning as seen by human vision. While the early vision stage has been partly explored by neuro-scientists and vision psychologists [Marr 1982, Livingstone 2008, Boden 2006, Biederman 1982a], they have not figured out yet higher stages.

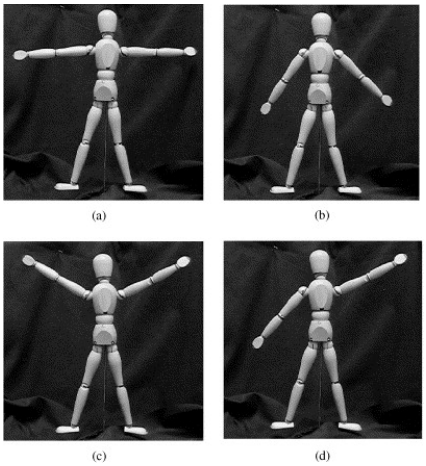
1.1.1 Issues in Computer Vision

In spite of efforts from the research community, an algorithm that recognizes well generic visual objects remains out of reach. Even if we put aside limitations of computational infrastructures, current vision algorithms are still challenged by many difficulties (see examples in Fig. 1.1) :

- The first challenge is due to acquisition conditions. For example, the lighting conditions at the time of acquisition either burn image details (too bright environment) or cover them by shadow.
- The second challenge is context understanding. Objects in the real world never occur in isolation but co-occur with other objects and with respect to particular contexts. In fact, human vision amounts to be noise tolerant and highly imaginative in anticipating the meaning of a scene even if constituting objects are not complete [Biederman 1982a]. Additionally, literature in psychology points out that human vision sees the whole scene before recognizing individual objects (Gestalt laws [Metzger 2006]).
- Appearance variability is another big challenge. In Fig. 1.1(d), there are four instances with different appearances of a single concept *chair* ; although looking differently, all of these instances have a common functionality of a chair. So far psychologists have not discovered a good enough theory that universally explains different forms of human concepts [Murphy 2004].

(a) Ill lighting conditions.

(b) The unified whole is different from the sum of the parts.



(c) Non-rigid objects.

(d) Intra-class variants.

FIGURE 1.1 – Some challenges in computer vision. In (a) is some uneven-exposure captures of faces ; such images cause difficulties for face recognizers since many details are lost due to shadow. Besides, dimmed lighting condition and over-exposure also cause similar difficulties. Another challenge is the inability of machine vision algorithms in capturing holistic image understanding. The example in (b) shows a dog with moving legs on the ground ; however, even the best object detectors cannot recognize as we do. For them, that picture is no more than black segments on a white background. Shown in (c) are the challenge caused by articulated objects. Parts of such an object are movable so that they form lots of relative positions. Every articulated object must be tracked by a specialized model. The multiplicity of appearances with respect to concept is shown in (d) ; although sharing the same functionality, the four chairs own every different appearances, i.e., the shape, material, structure, and color. Current learning methods are still far from conceptual abstraction.

1.1.2 Applications of Computer Vision

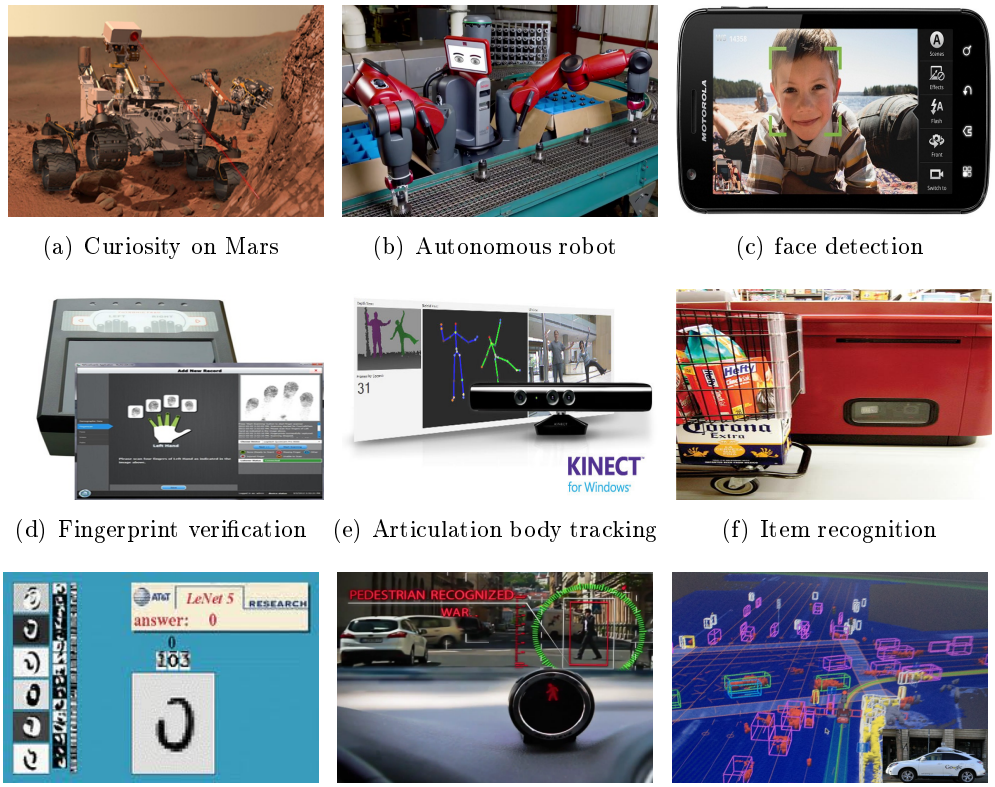
Despite those challenges, computer vision nowadays have moved beyond what people imagined decades ago. Various approaches and methodologies have been contributing to important successes of computer vision in real life. We list below some successful examples (illustrated in Fig. 1.2) in manufacturing, retailing, autonomous driving, security, entertainment, and Internet multimedia.

- Industrial robots with object recognition ability can replace human in operations that require very precise manipulations or mass processing, i.e., automatic inspection of integrated circuits, food preprocessing, metal fabrication.
- Smart cameras installed on vehicles support safe driving by detecting crossing pedestrians, keeping safe distances from nearby cars.
- Autonomous cars such as Google Car [1] or those of the DARPA Urban Challenge [2] can drive themselves safely over dozens of miles in urban streets.
- Retailers also get benefit from vision technologies. By installing along checkout lanes smart cameras [3], items in customers' baskets are automatically detected and recognized ; the cashier does not have to move items out of the basket for barcode reading but receives billing information from smart cameras.
- Computer vision also creates revolutions in modern warfare and space exploration. Drones (Unmanned Aerial Vehicle) are replacing human-controlled fighters in patrolling and seeking tasks ; computer vision techniques help robot Curiosity navigate the Mars.
- Biometrics is very useful for border control because it can accurately identify hundreds of millions of identities by matching faces and fingerprints ; this matching technique is implemented by computer vision algorithms.
- Placing smart cameras on public areas such as airports and stations helps detecting abnormal activities or abandoned baggages ; automatic surveillance on highways help regulating traffic.
- Compact cameras provide better photo quality by localizing faces and detecting smiling faces ; game stations such as Xbox's Kinect provides interactive games based on localizing and recognizing body parts using efficient computer vision algorithms.

1.2 Machine Learning for Computer Vision

During 1950s and 1960s, visual objects were often modeled as geometric primitives [Mundy 2006]. The popularity of this modeling might be due to i) the thought that computer vision is an inverse process of computer graphics, and ii) the rise of formal logic and artificial intelligence. In Robert's work [Roberts 1963], he proposed to describe objects in the real world as simplified blocks. In particular, objects were

1. http://www.google.com/about/jobs/lifeatgoogle/self-driving-car-test-steve-mahan.html
2. http://www.torcrobotics.com/case-studies/darpa-urban-challenge
3. http://www.evoretail.com/



(a) Curiosity on Mars (b) Autonomous robot (c) face detection

(d) Fingerprint verification (e) Articulation body tracking (f) Item recognition

(g) Optical character recognition (h) Pedestrian detection (i) Driveless car

FIGURE 1.2 – Some commercial applications of computer vision : (a) Computer vision techniques have been used in Mars Exploration Rover launched in 2003 and Curiosity Rover launched in 2011 http://mars.nasa.gov/msl/ ; (b) Baxter robot can be taught to do simple missions such as transferring parts from line to line, packing products into boxes, inspecting defective items http://www.rethinkrobotics.com/ ; (c) mostly compact cameras can detect human face and smile ; (d)Fingerprints scanner for biometrics identification http://www.fulcrumbiometrics.com/ ; (e) Kinect detects and tracks parts of the player so that he/she can fully interact with the game based on body movement http://www.xbox.com/kinect ; (f) The LaneHawk camera automatically detect items in the incoming basket and bill them accordingly to their identification http://www.evoretail.com/ ; (g) Optical character recognition is one of the early problems of computer vision http://yann.lecun.com/exdb/lenet/ ; (h) Mobileye's collision avoidance technology is able to "interpret" a scene in real-time and provide drivers with an immediate evaluation http://www.mobileye.com/.

restricted to polyhedral shapes on a uniform background. This approach was then abandoned due to the inability of geometric primitives to characterize objects whose appearances and shapes are complicated.

Statistical learning becomes an alternative. The most widely-used method is artificial neural networks (ANN) [McCulloch 1943] and its applications are vast, for example optical character recognition [4], face recognition, and plate recognition [Rowley 1996, Draghici 1997, Lawrence 1997]. The heart of ANN is a multilayer network of neuron units and the use of back-propagation algorithm in order to learn connection weights. However, the back-propagation optimization algorithm used for ANN turns out to be inefficient if there are more than three layers in a network. Recent advances [Krizhevsky 2012, Bengio 2009] in optimization methods have revived ANN and now deep network structures can be optimized efficiently with more than three layers.

Like ANN, the learning model of Poggio [Riesenhuber 1999] is also inspired from the working mechanism of human visual cortex. Their approach, named as *biologically inspired system*, assumes that the processing chain in our visual cortex can be modeled as a hierarchy of increasingly sophisticated representations. Inside every cell at the lowest level is the convolution (the SUM operator) between filters (for example directional Gabor filters) and the input image. At a higher level, the nonlinear MAX operator seeks for the strongest response among the cells in the lower level. By alternating the two mechanisms, the system achieves both pattern specificity and invariance to translation and scaling.

Energy-based learning [Bakir 2007] has been an effective framework for parametric methods such as conditional random fields and maximum margin Markov networks [Sutton 2012, Wallach 2004, Kindermann 1980]. This model captures dependencies between variables by associating a scalar energy to each observation of the variables. Energy terms are designed such that their values are lowered as long as more correct predictions are achieved. A big advantage of this model is the flexibility in designing energy terms which account for various low-level features (color, texture, shape) as well as semantic relationships. Energy-based learning has been successfully applied to image understanding, object class segmentation, and image parsing [Tighe 2013, Chen 2011a, Eigen 2012b].

Among statistical learning models, one of the most successful methods is Support Vector Machine (SVM) [Cortes 1995]. There are several reasons for SVM to be widely used in vision problems : i) the method is distribution-free so that data is conditioned on any distribution ; ii) a considerably good classifier may be obtained with a limited amount of training data ; iii) the max-margin condition guarantees the generalization of the classifier ; iv) kernelized SVM can handle nonlinear data ; v) SVM formula is convex. Additionally, the modularity property of SVM, where feature mapping (or kernelization [Shawe-Taylor 2004, Scholkopf 2001b]) and classification steps are separated, makes it becoming an off-the-shelf technique for pattern classification practices.

4. optical character recognition (OCR) http://yann.lecun.com/exdb/lenet/

1.3 Motivation and Contributions

At small and medium scales, computer vision problems are usually well handled using supervised learning techniques mentioned above. An essential condition is that training data must be sufficient for those algorithms to learn good classifiers. But in the long run, learning algorithms should be able to cope with the lack of training data and to exploit unlabeled data which is abundant. This movement is due to substantial changes of how people use technologies nowadays. For instance, Web is changing from 1.0 to 2.0 and people in the world are more and more involved in social networks. They have been creating, uploading, and sharing images and videos more than ever. Flickr [5], a website of photo hosting and sharing, gets about 60 millions photos uploaded per month ; within three years, the online photo sharing Instagram [6] has registered over 130 million users who have uploaded 40 million photos per day.

In order for vision algorithms to catch these trends, research on large-scale vision systems has been conducted. Torralba and his colleagues have collected a database of 80 million tiny images [7] across 75 thousands noun entries listed in Wordnet lexical database. Such a vast and diversed database is a good source for machine learning algorithms to generalize. Another example of large-scale database is ImageNet [8]. This database is organized according to the WordNet hierarchy and contains 21841 synsets and 14 million images in totals. ImageClef [9] is another large scale image retrieval and annotation challenge, which focuses on multiple image domains such as medical images, consumer photos, plant photos, and robot vision.

However, this wave of big data raises a more important question : how those algorithms can still perform well with less amount of training data. This problem is practical because annotation cost is expensive for large-scale databases. As supervised learning methods are no longer appropriate, there exists in machine learning the semi-supervised approach, the one that uses both labeled and unlabeled data for learning. Still using labeled data as supervised learning does, semi-supervised learning also uses unlabeled data which may exhibit some information about density distribution of data. Among semi-supervised learning techniques, inductive and transductive ones are the two major approaches. The former induces from training data a decision rule which is applicable to any unseen test data. The latter infers a labeling for test data based on both the training and test data ; since this labeling is not available elsewhere except the given test data, training classifiers is not necessary. Moreover, transductive learning may give a more suitable labeling with respect to test data. Based on this approach, our study considers : i) how to improve transductive inference when combined with kernel learning, and ii) to extend its use in order to learn semantic and low-dimensional representation of image databases.

For the first goal, two contributions are made. In Section 1.3.1, a new transduc-

5. www.flickr.com
6. http://instagram.com
7. http://groups.csail.mit.edu/vision/TinyImages/
8. http://www.image-net.org/
9. http://imageclef.org/

tive kernel learning formulation is proposed which is shown to be at least competitive against conventional learning approaches. In Section 1.3.2, the proposed method is expanded to many applications in which their specificities are used to adapt the original formula. For the second goal, the third contribution is made ; a new semantic subspace learning algorithm is introduced in Section 1.3.3 as the core of the mental search model introduced in the same section.

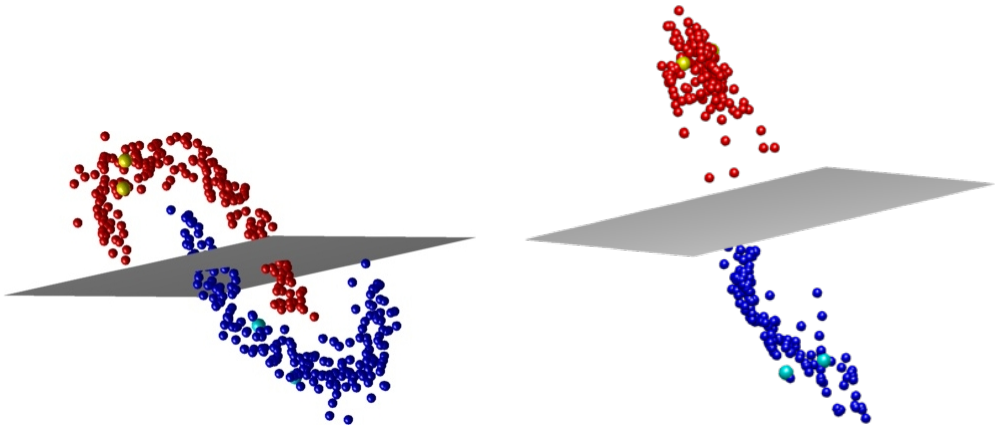
1.3.1 Transductive Kernel Learning

Our first contribution is a novel transductive method for kernel map learning. When labeled data is scarce or expensive, learning a good classifier becomes difficult. Transductive learning is particularly suitable for such situations. The goal of transductive learning is to infer labels of test data and not to learn a general decision rule for unseen data without using test data. Hence, transductive approaches consider both labeled and unlabeled data in inference. Based on this approach, our kernel map learning algorithm can exploit better the topological structure of the data, which helps diffusing label information from the labeled data to the unlabeled one.

In fact transductive inference techniques are becoming standard in machine learning due to their relative success in solving many real-world applications [Joachims 1999, Duchenne 2008, Liu 2009b, Joachims 2003]. Among them, kernel-based methods are particularly interesting but their success remains highly dependent on the choice of kernels. The latter are usually handcrafted or designed in order to capture better similarity in training data. The novel aspect in our method includes a new transductive learning algorithm for kernel design and classification.

Different from the related works [Maji 2008, Joachims 2002a, Asa 2008, Bach 2004, Wu 2006, Rakotomamonjy 2008, Bach 2008b, Sonnenburg 2006], we do not adopt kernel tricks [Scholkopf 2001b] but learn an explicit kernel map based on the topological structure of labeled and unlabeled data. Our approach is, therefore, *model-free* which means not restricted to off-the-shelf kernels. It also leads to better generalization performances.

Mathematically, our approach is based on the minimization of an energy function mixing i) a reconstruction term $E_{\text{data}}(\mathbf{X}, \mathbf{B}\mathbf{\Phi})$ that factorizes a matrix of input data $\mathbf{X}$ as a product of a learned dictionary $\mathbf{B}$ and a learned kernel map $\mathbf{\Phi}$, ii) a fidelity term $E_{\text{label}}(\mathbf{Y}, f(\mathbf{\Phi}))$ that ensures consistent label predictions $f(\mathbf{\Phi})$ with those provided in a ground-truth $\mathbf{Y}$, iii) a smoothness term $\sum_{(i,j)\in\mathcal{E}} E_{\text{smooth}}(f(\mathbf{\Phi}_i), f(\mathbf{\Phi}_j))$ that guarantees similar labels for neighboring data which is taken from the edge set $\mathcal{E}$ of the k-nearest neighbor graph $\{\mathcal{V}, \mathcal{E}\}$ of the data (see Fig. 2.6), and iv) regularizers $\psi(f, \mathbf{\Phi})$ which restrict classifier complexity and kernel map's rank. The minimization problem described above admits the following generic form and will



(a) The toy dataset above is not separable in $\mathbb{R}^2$. Given training (yellow dots) and test (cyan dots) data, a hyperplane cannot separate the two classes.

(b) The linearized kernel map learned by our method. The distribution of the data helps diffusing label information from the labeled to the unlabeled ones (red and blue dots).

FIGURE 1.3 – Illustration of learning the transductive kernel map with toy data.

be clarified in subsequent chapters :

$$\min_{f,\mathbf{\Phi},\mathbf{B}} \left\{ E_{\text{data}}\left(\mathbf{X}, \mathbf{B}\mathbf{\Phi}\right) + E_{\text{label}}\left(\mathbf{Y}, f(\mathbf{\Phi})\right) + \sum_{(i,j)\in\mathcal{E}} E_{\text{smooth}}\left(f(\mathbf{\Phi}_i), f(\mathbf{\Phi}_j)\right) + \psi\left(f, \mathbf{\Phi}\right) \right\}. \tag{1.1}$$

Solving this minimization problem makes it possible to learn both a decision criterion and a kernel map that guarantee linear separability in a high dimensional space and good generalization performance (see Fig. 1.3). Experiments conducted on object class segmentation show improvements with respect to baseline as well as related works on the challenging VOC database [Everingham 2010].

1.3.2 Regularized Transductive Kernel Learning

As the second contribution, we extend the framework of kernel map learning to image annotation and scene image interpretation. For the former application, a multi-class formula is derived based on the binary classification model, which is mentioned in the section above. Since label multiplicity leads to a shared kernel map between classifiers, additional regularizers can be added to the basic formula in order to amplify dependencies between classes. In the case of image annotation, we design a regularizer that enforces co-occurrence of labels within every image based on co-occurrence statistics from training data. Consequently, both smoothness and label-dependency terms diffuse label information not only between images conditioned on individual labels but also between labels within every image. In other word, the smoothness term supports inter-image visual similarity and the label-dependency term supports intra-image statistical dependencies between labels.

For the latter application, which is image interpretation, we use transductive

FIGURE 1.4 – Scene understanding is the problem of localizing and classifying visual objects in a scene image into known categories. Given a query image, we retrieve K most similar images from the labeled database. We then oversegment those $(K+1)$ images and obtain superpixels. By connecting superpixels based on their visual similarities, label information can be transfered from the labeled to the unlabeled ones. Due to the retrieval step, unknown objects in the test image are expected to find their similar instances in the training images ; however, visual ambiguities are unavoidable. In order to reduce such errors, contextual relationships are taken into account. In terms of statistics, these relationships are label correlations or prior probabilities about spatial layout of scene images ; for example, cars are on streets, windows cannot appear without buildings, sky is always on top of roads, etc.

kernel learning in order to segment and recognize visual objects in a scene. We reuse the multi-class formula in order to assign appropriate labels for every superpixel in test image in which label information is taken from labeled superpixels having similar visual appearance. Since visual variability may cause false labeling, we add a new regularizer that exploits referential meanings of superpixels. These contextual features not only help recognize superpixels whose appearances are not discriminative enough but also make the predicted labeling more coherent with respect to contextual rules implied by human vision. The application is depicted in Fig. 1.4.

1.3.3 Semantic Subspace Learning

The third contribution is about designing a new subspace learning algorithm dedicated to mental search. Current visual search engines basically use text-based queries to search for visual content. As multimedia data on the Internet is exploding (i.e., YouTube, Flickrs, Instagram, Facebook), it is almost infeasible to give

annotation to every image or video. Thus semantic search of visual data is the most probable solution for scalable multimedia retrieval. We use database visualization [Heesch 2008, Schaefer 2010] as an alternative, that relies on mapping data from high to low dimensional spaces where data can be easily spotted and explored by the user. In spite of the extension of nonlinear dimensionality reduction techniques [Tenenbaum 2000, Roweis 2000, Belkin 2001], their success in image database visualization is limited to datasets with well controlled semantics such as poses of faces or distortions of digits ; as these techniques are totally unsupervised, their application to generic databases [Schaefer 2010, Rubner 2001] produces "semantic-less" dimensions which are difficult to interpret and explore (see Fig. 1.6).

Our contribution is to introduce a novel mental search algorithm based on semantic subspace learning. The latter is designed using a novel principle, that unmixes K semantics from image data $\mathbf{X} \in \mathbb{R}^{n \times m}$ and maps them from an initial ambient space $\mathbb{R}^n$ (related to low level visual features including texture, color and shape) to an output subspace of $\mathbb{R}^K$ spanned by K well defined semantic bases. We cast this problem as convex quadratic programming optimization, constrained in a simplex spanned by few (pure) semantic endmembers, i.e.,

$$\begin{array}{lll} \min\limits_{\mathbf{\Phi} \geq 0} & \sum_{(i,j)\in\mathcal{E}} E_{\text{smooth}}(\mathbf{\Phi}_i, \mathbf{\Phi}_j) & \\ \text{s.t} & \mathbf{\Phi}_i = \mathbf{Y}_i & i = 1, \ldots, \ell \\ & \sum_{k=1}^{K} \mathbf{\Phi}_{ki} = 1 & i = \ell + 1, \ldots, m \end{array} \tag{1.2}$$

Assume that the given data sample is supported by some manifold, the optimization problem (1.2) preserves the topology of the data when it is mapped from the ambient space into the semantic space. The semantic representation $\mathbf{\Phi}_i$ of every input sample $\mathbf{X}_i$ can be seen as the membership vector with respect to K defined semantics. The first ℓ equality constraints, in which ℓ is the number of labeled data, state that the new representation $\mathbf{\Phi}_i$'s of labeled samples equals label vectors $\mathbf{Y}_i$'s. We call those ℓ samples *endmembers* because every sample must represents a unique semantic ; unlabeled data, in contrast, can be endmembers or mixtures of several semantics (see Fig. 1.5).

The advantage of the proposed approach is twofold ; on the one hand, it significantly reduces the dimensionality of the input space (which is difficult to explore or visualize), and on the other hand, it learns features which are semantically interpretable, i.e., their values are highly correlated with the defined semantics. Thereby, searching for a mental target simply reduces to scanning and targeting data according to their coordinates in the learned semantic subspace.

1.4 Outline of the Thesis

The rest of the thesis is organized as follows.

– Chapter 2 presents basic concepts of learning theory and literature review of a selection of machine learning techniques that are relevant to our work.

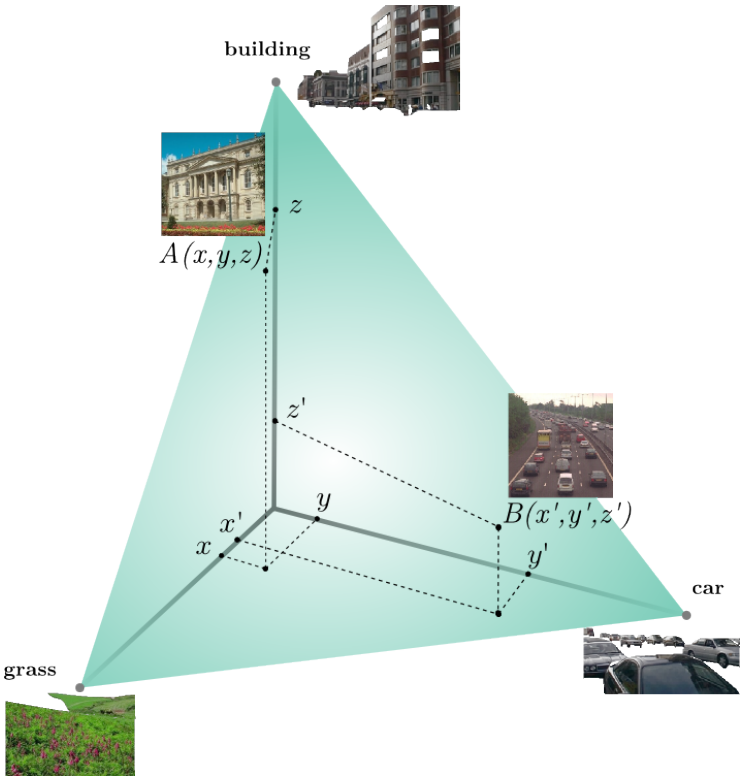


FIGURE 1.5 – Our method learns a simplex-based representation given an image database and K predefined semantics. Given that $K = 3$ and *building*, *car*, and *grass* are those semantics, we learn a new representation of the data in the $(K - 1)$ dimensional subspace. This new representation maps labeled data, also called endmembers, into vertices of the unit $(K - 1)$-simplex while unlabeled data are mapped to different locations in the simplex surface ; in fact their coordinates are determined based on how similar their semantic contents are with respect to each of the predefined semantics. For example, at the location $A(x, y, z)$, the image just contains building and plant. Therefore it is mapped near to the *building* vertex. At the location $B(x', y', z')$, the image contains a lot of cars, thus it is certainly mapped near to the *car* vertex. The learned representation allows the user to search for a mental target just by sliding a pointer along the simplex surface such that its coordinate values approximately equal to membership values of the mental target. Images surrounding this pointer are likely to contain image(s) of interest. Compare our simplex model with other visualization models shown in Fig. 1.6.

(a) Corel dataset (b) Hierarchical image browser Google Swirl

FIGURE 1.6 – Techniques for database visualization. In (a) is an image cloud visualized in three dimensional space [Rubner 2001] ; the tree-based visualization of Google Swirl in (b) is an alternative.

- Chapter 3 introduces our first contribution which is a novel kernel map learning method. The chapter describes steps to build the complete formulation, optimization procedures, theoretical guarantees, and experiments. Interactive object segmentation is used to demonstrate the idea.
- Chapter 4 and 5 present our second contribution, which is the extension of the binary formula proposed in Chapter 3 to multi-class problems. The new formulas are applied to image annotation and scene interpretation. For each of the problems, specific regularizers are designed in order to exploit prior knowledge from its data domain.
- Chapter 6 is dedicated to our third contribution. The chapter contains discussion about current limitations of visual search, the formulation of our method, optimization algorithms for large-scale datasets, and experiments of data visualization, image ranking, and relevance feedback.
- Chapter 7 concludes the thesis with a revision of contributions and discussions about the thesis perspectives.

Publications

1. Phong Vo, Hichem Sahbi, *Transductive Inference & Kernel Design for Ojbect Class Segmentation*, IEEE ICIP, USA 2012.
2. Phong Vo, Hichem Sahbi, *Transductive Kernel Map Learning and Its Applications to Image Annotation*, BMVC, UK, 2012.
3. Phong Vo, Hichem Sahbi, *Semantic Subspace Learning for Mental Search in Satellite Image*, IGARSS, Australia, 2013.
4. Phong Vo, Hichem Sahbi, *Spacious : An Interactive Mental Search Interface*, ACM SIGIR, Ireland, 2013.

Chapitre 2

Background

A revision of machine learning background is necessary before getting into the details of the thesis. Section 2.1 presents an overview about statistical learning theory, asymptotic behaviors of supervised learning with respect to the limit of training data, and the choice of function. This section also introduces regularization framework, which will be investigated in subsequent sections. Section 2.2 introduces Support Vector Machine (SVM), the basic building block of our method. Section 2.3 discusses recent advances in kernel methods and how kernelization helps SVMs tackle nonlinear data. Section 2.4 introduces semi-supervised learning where unlabeled data is used in order to improve classification results. The discussion amounts to transductive learning and necessary assumptions on which these learning algorithms are based. The final Section 2.5 reviews literature about dimensionality reduction and manifold learning techniques.

2.1 Overview on Statistical Learning

There have been two conventional paradigms in machine learning [Bishop 2006, von Luxburg 2008]. The first one is *unsupervised learning* whose goal is to find latent patterns in the data. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathcal{X}$ be a finite sample of size m drawn from an unknown distribution P. Then unsupervised learning seeks to estimate P that generates $\mathcal{X}$. Unsupervised learning consists of various problems such as clustering, density estimation, outlier detection, and dimensionality reduction.

The second paradigm is *supervised learning*. The principle is to associate an input $\mathbf{x}$ with an output y in which $y \in \mathcal{Y}$. The association should be done analogously to the way a finite sample of known inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ are associated with known outputs $\{y_1, \ldots, y_m\}$. Let us denote $\{(\mathbf{x}_i, y_i)\}$ as the training data, the algorithm that uses this data to learn generalized rule(s) is called *learning with supervision*, or simply *supervised learning*. If $\mathcal{Y}$ is continuous, for example $\mathcal{Y} \subseteq [0, 1]$, then we have a regression problem whose goal is to explain the relationship between inputs and outputs by a regressor. If $\mathcal{Y}$ is categorical, for example $\mathcal{Y} = \{-1, +1\}$, then we have a classification problem whose goal is to predict categorical target y for every input $\mathbf{x}$.

For supervised learning, the labeled data $\{(\mathbf{x}_i, y_i)\}$ are drawn from a joint distribution, which is again written as P for short. For an arbitrary test point $\mathbf{x}$ whose label is unknown, one needs to estimate the posterior probability $P(y|\mathbf{x})$ and to choose the target y^* that gives the highest $P(y^*|\mathbf{x})$, i.e.,

$$y^* \leftarrow \arg\max_y P(y|\mathbf{x}) \tag{2.1}$$

There are more than one way to solve (2.1). According to [Bishop 2006], supervised learning methods are classified into two families depending on the methodology of estimating the conditional probability $P(y|\mathbf{x})$. In particular, *generative algorithms* compute the posterior probability $P(y|\mathbf{x})$ via Bayes theorem

$$P(y|\mathbf{x}) \propto P(\mathbf{x}|y)P(y). \tag{2.2}$$

Once $P(\mathbf{x}|y)$ is known, $P(y|\mathbf{x})$ can be computed. The terminology "generative" comes from the fact that knowing $P(\mathbf{x}|y)$ allows us to generate the probability at any value of $\mathbf{x}$.

Unfortunately, estimating $P(\mathbf{x}|y)$ based on a finite sample of training data is difficult, not to mention that is it impossible if the quantity of training data is too small. Another family of supervised learning is *discriminative* ; it does not use Bayes theorem but directly estimates $P(y|\mathbf{x})$. Since $P(y|\mathbf{x})$ expresses the certainty of assigning label y to the test point $\mathbf{x}$, then there may exist a function f of $\mathbf{x}$ that approximates $P(y|\mathbf{x})$ well. We call f a decision rule, or a classifier, for the data generated by P. Training data $\{(\mathbf{x}_i, y_i)\}$ is used to learn this function f.

Formally, given m training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, the learned classifier f maps a feature vector $\mathbf{x}$ from the feature space $\mathcal{X}$ into a target value y in the target space

$\mathcal{Y}$, i.e.,

$$f: \begin{array}{ccc} \mathcal{X} & \longrightarrow & \mathcal{Y} \\ \mathbf{x} & \mapsto & y \end{array}. \tag{2.3}$$

The optimal classifier, given the underlying distribution P, is the Bayes classifier defined as

$$f_{\text{Bayes}}(\mathbf{x}) = \begin{cases} +1 & \text{if } P(y=1|\mathbf{x}) \geq 0.5 \\ -1 & \text{otherwise} \end{cases} \tag{2.4}$$

The Bayes classifier f_{Bayes} fires 1 if the probability of assigning $y = 1$ given $\mathbf{x}$ is equal or greater then 0.5 ; otherwise f_{Bayes} fires -1. The Bayes classifier does not exist in practice because we do not know P.

As our knowledge about P provided by the training data $\{(\mathbf{x}_i, y_i)\}$ is far from sufficient to recover P, we have to pick a function f from some function space $\mathcal{F}$ which maps $\mathcal{X}$ to $\mathcal{Y}$ (for example, $\mathcal{F}$ can be all possible functions from $\mathcal{X}$ to $\mathcal{Y}$, i.e., $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$) such that f approximates best f_{Bayes}. In other word, the function f must be selected such that it makes as least as possible incorrect decisions compared with f_{Bayes}. In order to measure how good a classifier f is, the following 0-1 *loss* function (see Fig. 2.4) tells us how precise f classifies inputs $\{\mathbf{x}_i\}$ as labels $\{y_i\}$, i.e.,

$$\ell(f(\mathbf{x}), y) = \begin{cases} 1 & \text{if } f(\mathbf{x}) \neq y \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Equipped with the loss function, the *(true) risk* of a classifier f is defined as the expected loss of f at all points $\mathbf{x}_i \in \mathcal{X}$, where $\mathbf{x}_i$'s are drawn independently from the same distribution P, i.e., $R(f) = \mathbb{E}\left[\ell\left(f(\mathbf{x}), y\right)\right]$. Therefore our goal is to choose f such that $R(f)$ is as small as possible. If we knew the true risk $R(f)$, then the following classifier $f_{\mathcal{F}}$ would be the best approximation to f_{Bayes} with respect to the function space $\mathcal{F}$:

$$f_{\mathcal{F}} \leftarrow \arg\min_{f \in \mathcal{F}} R(f). \tag{2.6}$$

Since we do not know the true risk $R(f)$, *empirical risk* $R_{\text{emp}}(f)$ is used to estimate classification quality of an arbitrary function f with respect to the finite training sample $\{(\mathbf{x}_i, y_i)\}$:

$$R_{\text{emp}}(f) = \frac{1}{m} \sum_{i=1}^{m} \ell\left(f\left(\mathbf{x}_i\right), y_i\right). \tag{2.7}$$

Let us assume that some machine learning algorithm produces a classifier f_m ; intuitively, f_m should obtain minimal empirical risk (2.7), which means

$$f_m \leftarrow \arg\min_{f \in \mathcal{F}} R_{\text{emp}}(f). \tag{2.8}$$

We call (2.8) the *empirical risk minimization* (ERM) principle [Vapnik 1998b, von Luxburg 2008]. Despite of the fact that f_m is learned from a limited amount of training data, one expects that f_m explains well not only the training data but also test data, which is not guaranteed by (2.8). However, if it is the case, then the

classifier f_m is called to generalize well. We now examine under which conditions the generalization of a classifier f_m is guaranteed, and going further how f_m approaches the best classifier $f_{\mathcal{F}}$.

Since $R_{\text{emp}}(f_m)$ is a biased estimation of $R(f_m)$, then there is no guarantee that f_m will make few errors on unseen data despite that f_m may perform very well on the training data. Instead, a good generalization performance of f_m is obtained if the difference $|R_{\text{emp}}(f_m) - R(f_m)|$ is small. Notice that this does not mention that the empirical risk $R_{\text{emp}}(f_m)$ must be small. In the worst case, f_m is overfitted to the training data, which means $R_{\text{emp}}(f_m)$ is very small but the difference is large.

As opposed to the notion of generalization which is a property of an individual classifier, *consistency* property concerns about the convergence of a function class, for example $\mathcal{F}$, as infinitely many training points are introduced. As $f_{\mathcal{F}}$ is the best classifier in $\mathcal{F}$, the consistency is how close a learned classifier f_m is with respect to the optimal solution $f_{\mathcal{F}}$. However, given a finite sample of training data, it is difficult for the ERM principle (2.8) to learn f_m with good generalization. In the one hand, there exists functions that perfectly predict on the training data but miserably fail to predict on test data. For example, it is a function that returns exact output values for training points and returns random guesses for test points ; such a function clearly obtains zero empirical risk but fails to generalize beyond training data. In the other hand, there may exist very good functions that predict correctly on all possible test data without being based on training data. As a result, the consistency is when f converges to $f_{\mathcal{F}}$ for all $f \in \mathcal{F}$, even with the worst choice of f. Studies in [Cortes 1995, Vapnik 1998b] figure out that the learning algorithm achieves the consistency if the following *uniform convergence* condition is satisfied

$$P\left(\sup_{f\in\mathcal{F}} |R(f) - R_{\text{emp}}(f)| > \varepsilon\right) \to 0 \quad \text{as} \quad m \to \infty \ . \tag{2.9}$$

The uniform convergence is a necessary and sufficient condition for the consistency of ERM principle with respect to $\mathcal{F}$. This condition states that the probability for the supremum of estimation error, with respect to $\mathcal{F}$, to be larger than ε (where ε can be any value) will vanish when the number of training points goes to infinity. Equivalently, the consistency is attained if the empirical risk converges with high probability to the true risk as the amount of training data reaches infinity. Uniform convergence condition gives us a theoretical guarantee for the consistency of f_m. The condition, however, does not give any recipe for practitioners.

The uniform convergence condition (2.9) addresses an important problem in which the capacity of the function space must be restricted such that (2.9) is feasible to be achieved (which is explained in the moment). Recall that the goal is to learn a classifier f_m which is consistent with not only $f_{\mathcal{F}}$ but also f_{Bayes} (we call this Bayes-consistency) ; this goal means that the true risk $R(f_m)$ must converge to the Bayes risk $R(f_{\text{Bayes}})$ [von Luxburg 2008]. However, this depends entirely on the capacity of the function space $\mathcal{F}$. If we are lucky enough then f_{Bayes} already belongs to the chosen function space $\mathcal{F}$ but generally we assume that initially f_{Bayes} is not

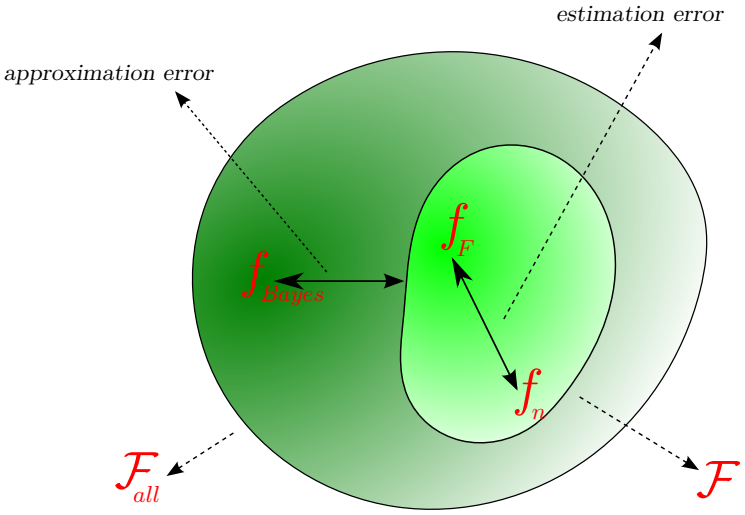


FIGURE 2.1 – Illustration of approximation error and estimation error in the task of learning a classifier f_m from training data. Estimation error must be minimized so that f_m approaches the best function $f_{\mathcal{F}}$. The smaller the function space $\mathcal{F}$ is, the easier the estimation but more difficult the approximation error to be minimized ; as a result, f_m has less chances to be Bayes-consistency. If $\mathcal{F}$ is too large, there will be more chances for f_m to be Bayes-consistency but it is more difficult for estimation error to be minimized.

included in $\mathcal{F}$. Then by incrementally increasing the capacity of $\mathcal{F}$ in some way, we may include f_{Bayes} into $\mathcal{F}$ so that ERM principle finally learns f_m with good generalization. Here it seems that expanding the function space $\mathcal{F}$ leads ERM to accomplish the goal.

Suppose that we select $\mathcal{F}$ as the universe space of all functions, i.e., $\{f : \mathcal{X} \to \mathcal{Y}\}$, in the following we will see that if we optimize over a too large function space $\mathcal{F}$, it will lead to inconsistency. Taking into account the true risk $R(f_{\mathcal{F}})$, we rewrite the gap $R(f_m) - R(f_{\text{Bayes}})$ as

$$R(f_m) - R(f_{\text{Bayes}}) = \underbrace{(R(f_m) - R(f_{\mathcal{F}}))}_{\text{estimation error}} + \underbrace{(R(f_{\mathcal{F}}) - R(f_{\text{Bayes}}))}_{\text{approximation error}}. \tag{2.10}$$

In the above expression, the *estimation error* is due to the randomness of the training data ; it measures how well the learned classifier f_m performs in relation to the best $f_{\mathcal{F}}$. The *approximation error* measures the loss incurred by setting $\mathcal{F}$ to be small. There is a dilemma on the capacity of $\mathcal{F}$: setting $\mathcal{F}$ large leads to the decrease of the approximation error but at the cost of large estimation error ; setting $\mathcal{F}$ to be very small leads to the decrease of the estimation error and the increase of the approximation error because $\mathcal{F}$ may not contain certain functions being able to explain the data. This dilemma is depicted in Fig. 2.1.

In trying to learn the best classifier f_m that obtains the Bayes-consistency property, both the approximation and estimation errors must vanish when m reaches

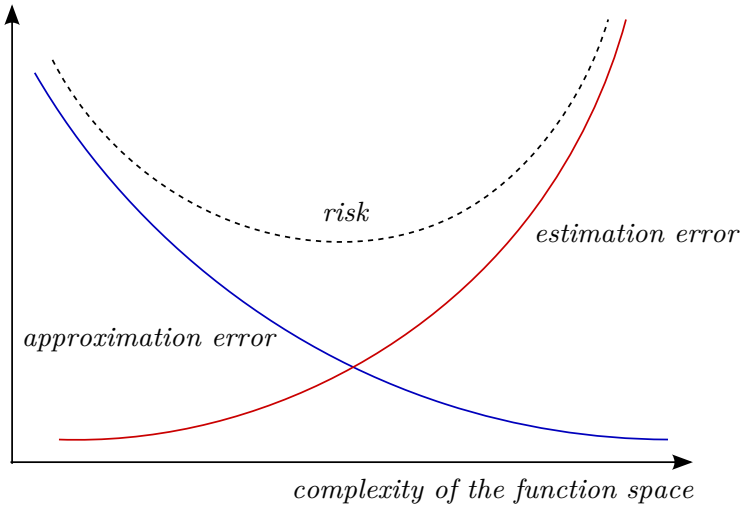


FIGURE 2.2 – The tradeoff between estimation error and approximation error in terms of the capacity of the function space $\mathcal{F}$. If prior knowledge about f are known, we can choose a function space $\mathcal{F}^*$ that minimizes the total risk.

infinity. It is not about to increase or decrease of the capacity of $\mathcal{F}$, it is about choosing the right $\mathcal{F}$. The following generation bound [Vapnik 1998b] explains the idea that an appropriate choice of $\mathcal{F}$ results into a consistent f_m. For any function $f \in \mathcal{F}$, with a probability at least $(1-\delta)$, the following inequality holds

$$R(f) \leq R_{\text{emp}}(f) + \sqrt{\frac{4}{m}\left(\log\left(2\mathcal{N}(\mathcal{F},m)\right) - \log(\delta)\right)}, \tag{2.11}$$

where $\mathcal{N}(\mathcal{F},m)$ is referred to as the *shattering coefficient* of function space $\mathcal{F}$ with respect to sample size m. It is interpreted as the maximal number of ways to give a dataset $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ their labels $\{y_1,\ldots,y_m\}$ in which $y \in \{-1,+1\}$. In other words, $\mathcal{N}(\mathcal{F},m)$ equals to the maximal number of ways that $\mathcal{F}$ can partition a dataset of size m into two partitions. Of course $\mathcal{N}(\mathcal{F},m)$ is bounded by 2^m.

The bound (2.11) states that if both $R_{\text{emp}}(f)$ and the square root term are small simultaneously, then $R(f)$ will be small as well. We can again explain the mechanism of ERM principle, but using the notion of shattering coefficient. If the capacity of $\mathcal{F}$ is large so that it is able to explain the data, the shattering coefficient is likely to be high and the square root term will grow as a result. This leads $R(f)$ to be more different from $R_{\text{emp}}(f)$, which implies inconsistency. If the capacity of $\mathcal{F}$ is small enough but contains function(s) which explain well the data, then the square root term is small so that $R(f) \approx R_{\text{emp}}(f)$ and f_m is more likely to be consistent. Finally, the complexity of the learning task entirely depends on whether $\mathcal{F}$ is suitably chosen, which is partly determined by our prior knowledge about the task. The *structural risk minimization* (SRM) [Vapnik 1998b] reveals more about the principle on how to choose an appropriate capacity level of $\mathcal{F}$ by forming a nested structure of $\mathcal{F}$. Starting from $\mathcal{F}_{\text{init}}$ with considerably small capacity, $\mathcal{F}$ is steadily expanded until

the optimal trade-off between the complexity of the solution (the capacity of $\mathcal{F}$) and the quality of fitting to training data (small empirical risk) is found.

In practice, SRM is rarely used for model selection and regularization approaches [Scholkopf 2001b] propose more efficient ways to learn classifiers with good generalization. The heart of this approach is the regularizer term $C(f)$ in the following objective function

$$f_m \leftarrow \arg\min_{f\in\mathcal{F}} \{R_{\text{emp}}(f) + C(f)\}\,. \tag{2.12}$$

which is similar to ERM in the sense that the empirical risk $R_{\text{emp}}(f)$ is minimized. However, the additional regularizer $C(\cdot)$ is responsible for f_m to generalize well to test data ; it enforces f_m to behave accordingly to some prior(s) induced inside $C(f)$. For instance, $C(f)$ can penalize any high fluctuation of f in explaining the data ; that means f must not be too complex to fit to training data, otherwise f is overfitted. Other priors include the inter-dependency between data points [Bakir 2007], max-margin separation [Cortes 1995], sparsity [Olshausen 1997, Subrahmanya 2010], manifold [Belkin 2006], etc. The basic difference between SRM and the regularization approach is that the former regulates the complexity of function space $\mathcal{F}$ while the latter regulates the complexity of an individual function f.

2.2 Support Vector Machines

In Section 2.1 we briefly introduced the statistical learning theory with some basic concepts such as uniform convergence and generalization bounds. These knowledges are necessary to understand under which circumstances learning is feasible. From now on, we review particular methods which are building blocks of our contributions. In this section we introduce support vector machines (SVMs) – a successful and popular supervised learning algorithm – under the light of statistical learning theory and regularization framework.

Prior to introduce SVM, it is essential to review early linear classification models – the background on which SVM is built. A typical example of linear classification model is perceptron [Rosenblatt 1958], which is the basis for other methods such as logistic regression, support vector machine, and conditional random fields [Bishop 2006]. The binary classifier perceptron has its input a feature vector $\phi(\mathbf{x})$ of signal $\mathbf{x}$ transformed by a nonlinear mapping function $\phi(\cdot)$; its generalized model is of the form

$$y(\mathbf{x}) = g\left(f\left(\phi(\mathbf{x})\right)\right) \tag{2.13}$$

in which $f(\cdot)$ is a linear function $f(\phi(\mathbf{x})) = \mathbf{w}'\phi(\mathbf{x}) + b$, which is a line in 2D space, a plane in 3D space, and hyperplane if the dimensionality more than three. Rosenbatt's model aims to use the hyperplane $\mathbf{w}'\phi(\mathbf{x}) + b = 0$ in order to separate the data into two halves with respect to their labels. The notation $\mathbf{w}'$ denotes the transpose of vector $\mathbf{w}$. The intercept b is often incorporated into $\mathbf{w}$ by increasing by one the dimensionality of $\mathbf{w}$ and putting 1 at the end of vector $\mathbf{x}$, i.e., $f(\phi(\mathbf{x})) =$

$[\mathbf{w}' \ b] \cdot [\phi(\mathbf{x})' \ 1]'$. The nonlinear activation function $g(\cdot)$ converts continuous values given by $f(\cdot)$ into discrete values

$$g(u) = \begin{cases} +1, & u \geq 0 \\ -1, & u < 0 \end{cases}. \tag{2.14}$$

The activation function $g(\cdot)$ is very similar to Bayes classifier (2.4) in which u is related to an estimation of conditional probability $P(y = +1|\mathbf{x})$. A common design for the empirical risk of perceptron is to count the total number of misclassified points, i.e., $L = -\sum_{i \in \mathcal{M}} y_i g(f(\mathbf{x}_i))$ in which $\mathcal{M}$ denotes the set of misclassified points. The loss returns 1 iff $g(f(\mathbf{x}_i))$ and y_i have opposite signs. Since such a non-differentiable quantity is difficult to optimize ($g(u)$ is discontinuous at $u = 0$), (2.14) is relaxed by omitting the thresholding function $g(\cdot)$; thus we get the new empirical risk

$$L(\mathbf{w}) = -\sum_{i \in \mathcal{M}} y_i \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle . \tag{2.15}$$

According to [Bishop 2006], stochastic gradient descent is used to find the update rule for the weight vector $\mathbf{w}$ with respect to every random choice of $\mathbf{x}_i$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta y_i \phi(\mathbf{x}_i) \tag{2.16}$$

in which η is the learning rate. At every update cycle, the weight $\mathbf{w}$ is increased or decreased by a quantity of $\eta y_i \phi(\mathbf{x}_i)$, which depends on the sign of y_i. The idea of the update rule is to adjust the hyperplane $f_{\mathbf{w}} = 0$ such that the misclassified point is correctly classified after every iteration. The optimization converges to a solution (a perfect separation of the training data) after a finite number of steps if such a solution exists with respect to training data [Rosenblatt 1958, Marvin 1969, Hertz 1991]. In practice, it is difficult to know whether the algorithm cannot converge (because of the inseparability of the data) or converges slowly. Furthermore, there are more than one solution for separable data and the final solution depends on the initialization of parameters.

A breakthrough was made when support vector machine (SVM) [Cortes 1995] was proposed. Different from perceptron, SVM provides a convex quadratic form which guarantees global solution. Similarly to perceptrons, SVM seeks for a hyperplane that separates positive and negative training points ; but different from perceptrons, that separation must be maximal. Given that the data is separable, the separation ρ is defined as the largest value such that

$$y_i \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \geq \rho, \forall i = 1, \ldots, m \tag{2.17}$$

where $2\rho \geq 0$ is called the margin of the classifier $f_{\mathbf{w}}$. According to the illustration in Fig. 2.3, this margin is twice the distance from the hyperplane $f_{\mathbf{w}} = 0$ to its nearest training point(s) ; $\mathbf{w}$ and b must be sought in order for ρ to be optimal. As explained in Appendix D, it is equivalent to the following quadratic optimization problem

$$\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w}} \tfrac{1}{2} \|\mathbf{w}\|_2^2 \quad \text{s.t} \quad y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 1, \quad i = 1, \ldots, m \tag{2.18}$$

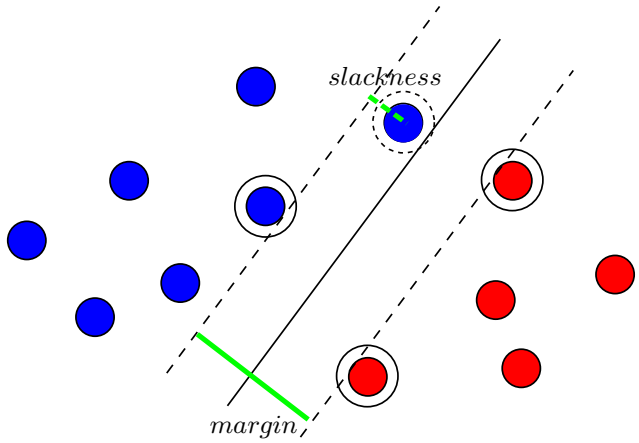


FIGURE 2.3 – Based on the margin concept, SVM finds a separating hyperplane such that the margin (the green solid line), which is twice the distance from the hyperplane to the nearest training point(s), is maximized. Maximal margin gives us a sense that the classification is done with the least uncertainty. Sometimes, it is unavoidable to let some training points violate the margin condition, that is the distance from them to the hyperplane are less than the margin. The margins of those points becomes $1 - \xi_i$ where ξ_i is the slack variable of $\mathbf{x}_i$.

where $\langle \cdot, \cdot \rangle$ denotes dot-product of two vectors in some high (possibly infinite) dimensional feature space, $\|\cdot\|_2$ is the L_2 norm of a vector. Notice that the margin ρ is vanished in (2.18) because $\mathbf{w}$ and b is rescaled such that ρ always equals 1.

In some cases, it is impossible for $f_{\mathbf{w}}$ to perfectly partition the training data. This may be due to dimensionality imposed by mapping $\phi(\cdot)$ when it is not high enough for $\{\phi(\mathbf{x}_i)\}$ to be separable. Even if those mappings are separable, it is not recommended in doing so because $f_{\mathbf{w}}$ may be over-fitted. If $f_{\mathbf{w}}$ has to make some miss-classifications, then some of inequalities in (2.17) are not satisfied. Since we do not know which data points should be misclassified, it is better to replace hard constraints in (2.18) by a soft *hinge loss* $(\cdot)_+ = \max[0, \cdot]$ and let the optimization algorithm decides by itself. The soft-margin SVM is defined as follows

$$\min_{\mathbf{w}} \ \frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{C}{m} \sum_{i=1}^{m} \left(1 - y_i \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle\right)_+ . \tag{2.19}$$

With the presence of the hinge loss, training points can violate the margin but with trade-off. The regularization coefficient C controls this trade-off between complexity of the decision rule and proportion of inseparable points [Cortes 1995, Cherkassky 2002]. If C is very large, the training error is highly penalized so that less inseparable points are made on the training data. This may cause $f_{\mathbf{w}}$ to be overfitted to the training data. If C is very small, more inseparable points are allowed so that the decision function $f_{\mathbf{w}}$ may not explain well the distribution of the

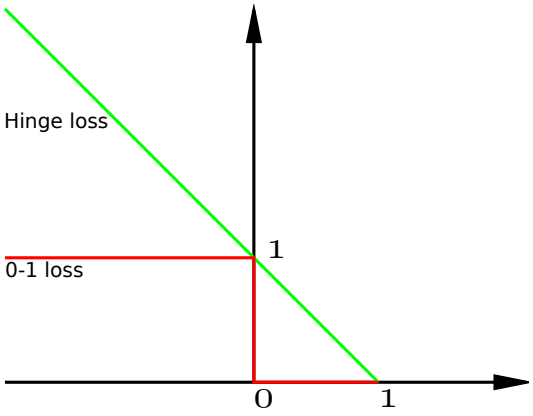


FIGURE 2.4 – Two loss functions used in classification problems : shown in red is the $0-1$ loss which equals 1 iff $f(x) = y$ and 0 if $f(x) \neq y$; shown in green is the hinge loss $(1 - yf(x))_+ = \max[0, 1 - yf(x)]$ which is tightly associated with the margin concept of SVMs.

training data, which is called under-fitting. Appropriate choices of C are owed to model selection methods, which are out of this scope.

The introduction of soft-margin SVM raise an optimization issue because hinge loss $(1-u)_+$ is non-differentiable at $u = 1$. An alternative is to replace hinge loss with slack variables ξ_i's, i.e.,

$$\begin{array}{lll} \min\limits_{\mathbf{w}} & \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i & \\ \text{s.t} & y_i\left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b\right) \geq 1 - \xi_i, & i = 1, \ldots, m \\ & \xi_i \geq 0 & i = 1, \ldots, m \end{array} , \tag{2.20}$$

because treating with bound constraints $\xi_i \geq 0$ is easier. Solving (2.18) can be done in either primal form or dual form. While there are few works [Chapelle 2007] that solve SVM in the primal form, machine learning community rather prefer the dual form. This form can be solved by first applying Lagrange multipliers to inequalities of SVM and then using Karush-Kunn-Tucker conditions [Bertsekas 1999, Fletcher 1987, Scholkopf 2001b]. Solving SVM in dual form also gives us a chance to understand the meaning of support vectors (see Appendix D).

Since SVM is a non-parametric model, then choosing the primal or dual form largely depends on the problem's scale. For small and medium scale problems, the dual form SVM can be solved efficiently using kernel tricks (see section below). For larger scale problems, the dual form exposes its disadvantage because computational cost of kernel methods grows rapidly with respect to the size of training data.

2.3 Feature Mapping

In this section we will discuss the *feature map* $\phi(\cdot)$ that appears in (2.18). The idea of $\phi(\cdot)$ is to convert nonlinear relations between data points into linear ones. In the following discussions, we highlight solutions for this problem.

2.3.1 Implicit Feature Map

In this approach, the feature space $\mathcal{H}$ is not explicitly defined. So one neither needs to define an analytic form of the mapping $\phi(\cdot)$ nor compute $\phi(\mathbf{x}_i)$'s. Instead, there is a family of special functions called *kernels* that compute the dot-product $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ between two feature maps without explicitly computing individual maps. Formally, a kernel is a function κ that takes inputs $\mathbf{x}, \mathbf{z} \in \mathcal{X}$ and outputs $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ in which ϕ is a mapping from $\mathcal{X}$ to a feature space $\mathcal{H}$, i.e.,

$$\begin{aligned} \phi : \quad & \mathcal{X} \to \mathcal{H} \\ & \mathbf{x} \mapsto \phi(\mathbf{x}) \end{aligned} . \tag{2.21}$$

Before studying properties of kernels and how they guarantee an implicit feature map, it is interesting to see an example. The purpose of this example is to show how efficient a polynomial kernel can replace more expensive dot-products. The toy data is shown in Fig. 2.5(a) where "white" points are encircled by "red" points. SVM clearly cannot separate this data by using a linear map $\phi(\mathbf{x}) = \mathbf{x}$. Thus we attempt to lift the data into $\mathcal{H} \equiv \mathbb{R}^3$ space using the following polynomial map

$$\phi : (x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2} x_1 x_2). \tag{2.22}$$

Via the mapping ϕ, the new representation is separable as shown in Fig. 2.5(b). Furthermore, the dot-product in this new space can be computed by the following polynomial kernel

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 z_1 x_2 z_2 = (x_1 z_1 + x_2 z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 . \tag{2.23}$$

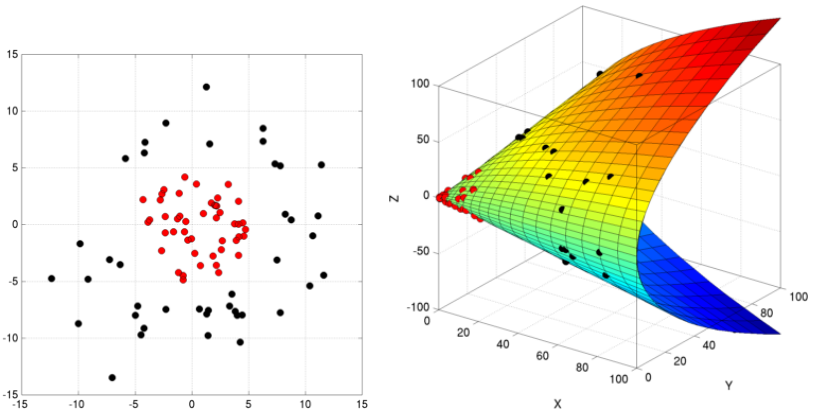
In other word, by squaring the dot-product of $\mathbf{x}$ and $\mathbf{z}$ in the input space $\mathcal{X}$, we obtain the equivalent result when taking the dot-product between the $\phi(\mathbf{z})$ and $\phi(\mathbf{z})$. The rightmost term can be rewritten as the value of a kernel function $\kappa(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z} + b)^d$ in which $b = 0$ and $d = 2$ in the case above.

From the example above, we see that if the SVM formula can be expressed in terms of dot-products, then implicit feature map can be applied; this can save much of computational cost because dot-products in high dimensional space is much expensive than using kernels.

2.3.1.1 Kernelized SVM

This section explains how to exploit implicit kernel map in solving SVMs. Since the goal of SVM is to find the optimal weight vector $\mathbf{w}$, which is the normal vector of the hyperplane $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = 0$ separating the data $\{(\phi(\mathbf{x}_i), y_i)\}$, $\mathbf{w}$ must belong to the feature space $\mathcal{H}$ and it admits a functional representation $\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \phi(\mathbf{x}_i)$ (the Representer theorem [Scholkopf 2001b, Shawe-Taylor 2004] or Appendix E). Since the feature map ϕ is associated to a kernel function $\kappa(\cdot, \cdot)$ (see Appendix E), then L_2 norm of $\mathbf{w}$ in (2.18) therefore becomes

$$\|\mathbf{w}\|_2^2 = \langle \mathbf{w}, \mathbf{w} \rangle = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle \kappa(\cdot, \mathbf{x}_i), \kappa(\cdot, \mathbf{x}_j) \rangle = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j), \tag{2.24}$$



(a) Input data (b) Mapped data in feature space

FIGURE 2.5 – The example of how a mapping can covert nonlinear relations of the data in the input space to the linear relations in the feature space.In this example, the mapping $(x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ lifts the data from $\mathbb{R}^2$ to $\mathbb{R}^3$. It is clear that a plane can separate the red points from black points.

in which the transformations from the left hand side terms to the rightmost hand side term is due to the *reproducing kernel property* (E.9). The kernelized SVM is redefined as

$$\begin{array}{lll} \arg\min\limits_{\boldsymbol{\alpha},b,\boldsymbol{\xi}} & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + \frac{C}{m}\sum_{i=1}^{m}\xi_i & \\ \text{s.t} & y_i\left(\sum_{j=1}^{m}\alpha_j y_j \kappa(\mathbf{x}_i,\mathbf{x}_j) + b\right) \geq 1-\xi_i & i=1,\ldots,m \\ & \xi_i \geq 0 & i=1,\ldots,m \\ & \alpha_i \geq 0 & i=1,\ldots,m \end{array} . \tag{2.25}$$

where kernel values $\kappa(\mathbf{x}_i, \mathbf{x}_j)$'s are computed based on the choice of kernel function $\kappa(\cdot,\cdot)$. There are numerous kernel functions to use, for example polynomial kernel $\kappa(\mathbf{x},\mathbf{y}) = (\mathbf{x}'\mathbf{y}+c)^d$, Gaussian RBF kernels $\kappa(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2\right)$. While we know the dimensionality of the feature space induced by d^{th} degree polynomial kernels, the dimensions of Gaussian RBF kernel is known to be infinite [Scholkopf 2001b, Cristianini 2010, Shawe-Taylor 2004]. Nevertheless, it does not matter since the complexity of (2.25) does not depend on data dimensionality but size of training data.

2.3.1.2 Kernel Learning

In order to choose an appropriate kernel for a given training data, one must choose among off-the-shelf kernels the one that performs best. This can only be done by testing every kernel and tuning its parameters accordingly. Model selection techniques such as cross validation and grid search [Chang 2011] can be used for that purpose. Since parameter space rapidly grows as the number of parameters increases, model selection is efficient only if the number of parameters does not exceed two.

Let us take an example with RBF kernel, a generic smooth kernel. Using RBF kernel with SVM requires tunning for two parameters C and kernel bandwidth σ. In order to select the best model for SVM with RBF kernel, we have to search for the optimal pair (C^*, σ^*) using grid search [Chapelle 2002]. For every attempt, the training data is partitioned into K folds : $K-1$ folds are used for training and the rest for test. The optimal pair (C^*, σ^*) is the one that achieves the highest classification accuracy.

A generic kernel like RBF, however, is not suitable for every problem. For instance, studies of [Lazebnik 2006a, Maji 2013] have revealed that additive kernels such as histogram intersection and χ^2 kernels outperform RBF kernel in vision problems that use histogram data. Therefore, it may be better if kernels can be learned from the data, which means that they are more suitable for particular cases of data. Multiple Kernel Learning (MKL) [Rakotomamonjy 2008, Bach 2004, Bach 2008b, Subrahmanya 2010, Picard 2010] is one of early efforts to learn a weighted linear combination of basic kernels, i.e.,

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sum_{p=1}^{P} \gamma_p^2 \kappa_p(\mathbf{x}_1, \mathbf{x}_2) = \sum_{p=1}^{P} \langle \gamma_p \phi_p(\mathbf{x}_1), \gamma_p \phi_p(\mathbf{x}_2) \rangle = \langle \psi(\mathbf{x}_1), \psi(\mathbf{x}_2) \rangle \,, \quad (2.26)$$

in which we denote the compositional mapping $\psi(\cdot) = (\gamma_p \phi_p(\cdot)')'_{p=1,\dots,P}$. Since the dimensionality of $\psi(\cdot)$ equals the sum of the basic mapping $\phi_p(\cdot)$'s, then the normal vector $\mathbf{w}$ of the separating hyperplane is the concatenation of the basic $\mathbf{w}_p$'s too, i.e., $\mathbf{w} = \left(\gamma_p \mathbf{w}_p'\right)'_{p=1,\dots,P}$. Substituting these new expressions of $\psi(\cdot)$ and $\mathbf{w}$ into the primal form SVM (2.18), we obtain the standard formula of MKL-SVM

$$\begin{array}{ll} \min\limits_{\mathbf{w}_p, b, \boldsymbol{\beta} \geq 0} & \frac{1}{2} \sum_{p=1}^{P} \beta_p \left\| \mathbf{w}_p \right\|^2 + C \sum_{i=1}^{m} \xi_i + \theta(\boldsymbol{\beta}) \\ \text{s.t} & y_i \left(\sum_{p=1}^{P} \beta_p \left\langle \mathbf{w}_p, \phi_p(\mathbf{x}_i) \right\rangle + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, m \\ & \beta_p \geq 0, \qquad\qquad\qquad\qquad\quad p = 1, \dots, P \end{array} \,, \quad (2.27)$$

where $\beta_p = \gamma_p^2$ and $\kappa_p(\cdot, \cdot)$'s are basic kernels, for example RBF with various bandwidth choices, histogram intersection kernel, χ^2 kernel, polynomial kernel. The regularizer term $\theta(\boldsymbol{\beta})$ prevents the combination weight vector $\boldsymbol{\beta}$ from growing to infinity. Two popular choices of $\theta(\boldsymbol{\beta})$ are L_1 norm $\|\boldsymbol{\beta}\|_1 = \sum_p |\beta_p|$ and L_2 norm $\|\boldsymbol{\beta}\|_2 = \sum_p \beta_p^2$. While the former promotes sparse solutions of kernel combination (a.k.a kernel selection), the latter uses all P kernels. Lately, [Gehler 2008] proposes infinite kernel learning, [Cortes 2009b] proposes polynomial combination of basic kernels, [Bach 2008a] learns a hierarchical kernel, etc.

2.3.2 Explicit Feature Map

As mentioned above, the feature map $\phi(\cdot)$ is rarely computed. However, Nystrom's approximation [Williams 2000] can explicitly construct a data-dependent approximation $\hat{\phi}(\mathbf{x}) \in \mathbb{R}^d$ for an implicit feature map $\phi(\mathbf{x}) \in \mathcal{H}$ in which d is considerably small compared with the training size. Let us assume that this approximation is a span of ℓ eigenfunctions $\psi_i(\cdot)$'s associated with ℓ eigenvalues λ_i's and a kernel

$\kappa(\cdot,\cdot)$ such that $\kappa(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{N} \lambda_i \psi_i(\mathbf{x})\psi_i(\mathbf{z})$ where N may be infinite. The Nystrom approximation states that the image in the explicit space $\{\psi_1,\ldots,\psi_\ell\}$ of a vector $\phi(\mathbf{z})$ is $\hat{\phi}(\mathbf{z}) = [\psi_1(\mathbf{z}) \ \ldots \ \psi_\ell(\mathbf{z})]'$ in which entries are approximated with respect to a finite sample $\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$, i.e.,

$$\psi_i(\mathbf{z}) = \frac{\sqrt{\ell}}{\lambda_i} \sum_{k=1}^{\ell} \kappa(\mathbf{z},\mathbf{x}_k)\mathbf{U}_{ki}, \tag{2.28}$$

so that

$$\hat{\phi}(\mathbf{x}) = \sqrt{\ell}\boldsymbol{\Sigma}^{-1}\mathbf{U}' \left[\kappa(\mathbf{z},\mathbf{x}_1) \ \ldots \ \kappa(\mathbf{z},\mathbf{x}_\ell)\right]'. \tag{2.29}$$

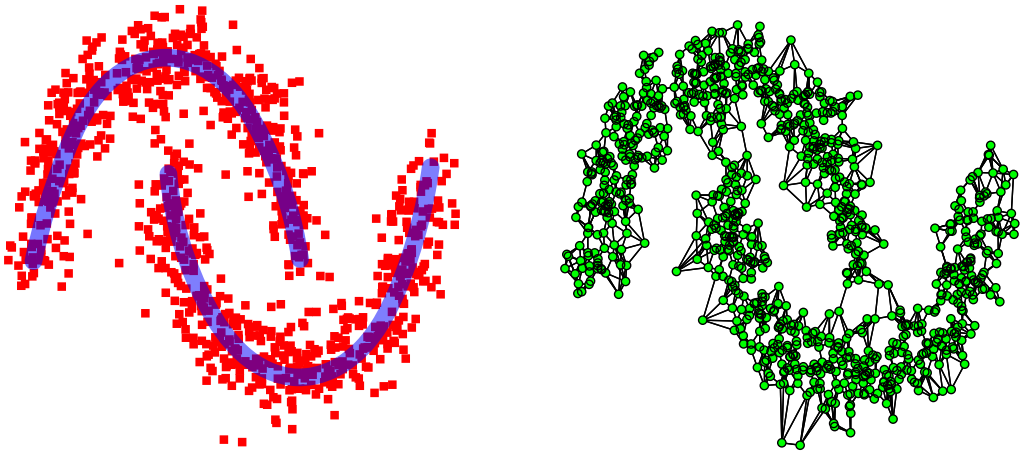
In the formula above, $\mathbf{U}$ is the matrix of eigenvectors resulting from the singular value decomposition (SVD) $\mathbf{K} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}'$ of kernel matrix $\mathbf{K} \in \mathbb{R}^{\ell\times\ell}$ and $\boldsymbol{\Sigma} = \operatorname{diag}(\lambda_1,\ldots,\lambda_\ell)$. Since ℓ is small, the computation of the kernel matrix $\mathbf{K} \in \mathbb{R}^{\ell\times\ell}$ is cheap and so does its SVD decomposition. With a good sampling $\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$, Nystrom approximation is a simple way to compute explicit feature maps. Notice that the approximation is not unique because the mapping depends on data sampling.

Explicit feature map was not noticed in the past because kernel methods can tackle inexpensively medium scale problems. Nowadays one needs to train nonlinear SVMs on tens of millions training points, then solving kernelized SVMs is impractical. In particular, training kernelized SVMs (2.25) has $\mathcal{O}\left(m^3\right)$ time and $\mathcal{O}\left(m^2\right)$ space complexities. It is thus computationally infeasible to train kernelized SVMs on very large data sets. The solution is to approximate nonlinear SVM by linear SVM with suitable feature map. There have been interesting works exploring low-cost explicit mapping methods. For instance, [Maji 2013] approximates intersection kernel $\kappa(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} \min[x_i, y_i]$, [Vedaldi 2012] unifies analyses of a large family of additive kernels such as intersection kernel, χ^2 kernel and Hellinger's kernel.

2.4 Semi-supervised and Transductive Learning

The uniform convergence condition (2.9) states that the generalization of a classifier f depends on function class $\mathcal{F}$ as well as the training sample size m. If m is small, then the learned classifier f may not generalize well, i.e., perform poorly on test data. This situation may happen when labeled data is expensive or scarce to obtain ; for example the labeling task requires skilled annotators or the labeled data is obtained via physical experiments. In contrast unlabeled data is always abundant and inexpensive. Let us denote $\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell,\mathbf{x}_{\ell+1},\ldots,\mathbf{x}_m\}$ a finite sample of both labeled and unlabeled samples in which the first ℓ points are labeled, then our question is how to learn a classifier with good generalization, especially when ℓ is much smaller than m.

Semi-supervised learning is a class of machine learning techniques that uses unlabeled data to improve learning outcome. A semi-supervised algorithm can be *inductive* or *transductive*. If it is inductive, then the learning goal is to infer from



(a) A finite sample of a two arcs manifold (b) A k-nearest neighbor (or ε-ball) graph

FIGURE 2.6 – Without knowing the data distribution P of the joint space $\mathcal{X} \times \mathcal{Y}$, we can still grasp prior knowledge of the marginal distribution $P(\mathbf{x}) = \int_{\mathcal{Y}} P(\mathbf{x}, y)dy$ with sufficiently large amount of unlabeled data. The marginal $P(\mathbf{x})$ gives us information about the geometrical structure of the unknown distribution P, which is supposed to lie on a manifold $\mathcal{M}$. Graph-based learning methods usually construct an adjacency graph from the data. If the amount of data points is sufficiently high, the graph is qualified as an approximation of the unknown $\mathcal{M}$.

data a general decision rule which is usable for any unseen data drawn from the same distribution, the one that generates the training data. SVM is such a typical example of inductive learning. If it is transductive, then the learning goal is to give correct labels, and not to to derive a decision function, for the unlabeled data. Firstly introduced in [Vapnik 1977], transductive learning seeks to transfer label information from labeled to unlabeled data without learning any explicit decision function. As a consequence, in transductive setting the unlabeled data $\{\mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_m\}$ is also the test data and prediction is not made in the whole space $\mathcal{X}$ but only those $(m - \ell)$ test points. The philosophy of transductive learning is based on Vapnik's conjecture (see [Chapelle 2006b], chapter 24) : "When trying to solve some problem (labeling test data), one should not solve a more difficult problem as an intermediate step (learning a decision function)."

Being either inductive or transductive, learning methods with semi-supervision need to uncover patterns hidden in unlabeled data and exploit them. In Section 2.1 we assumed that training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ are drawn from the joint probability distribution P in $\mathcal{X} \times \mathcal{Y}$ space. It also means that unlabeled data $\{\mathbf{x}_i\}_i$ are drawn from the marginal distribution $P(\mathbf{x}) = \int_{y \in \mathcal{Y}} P(\mathbf{x}, y)dy$. With sufficient amount of unlabeled data, we have prior knowledge about $P(\mathbf{x})$. The following assumptions [Chapelle 2006b] provides necessary conditions for semi-supervised learning to relate the posterior probability $P(y|\mathbf{x})$ and the marginal $P(\mathbf{x})$ (see Fig. 2.6).

– *Cluster assumption :* If $\mathbf{x}$ and $\mathbf{z}$ are in the same cluster, which means they are

drawn from the same density region characterized by the marginal probability distribution $P(\mathbf{x})$, then they are likely to be from the same class, i.e., $P(y|\mathbf{x}) \approx P(y|\mathbf{z})$. If $\mathbf{x}$ and $\mathbf{z}$ are separated by a low-density region (for example, $\mathbf{x}$ and $\mathbf{z}$ belong to different clusters), then they are less likely to be from the same class.

– *Smoothness assumption :* Considering $P(\mathbf{x})$ in a low-dimensional manifold and $\mathbf{x}, \mathbf{z}$ drawn from $P(\mathbf{x})$, the smoothness assumption states that if $\mathbf{x}$ and $\mathbf{z}$ are close, then $P(y|\mathbf{x})$ and $P(y|\mathbf{z})$ should be close too. It is easy to see that the cluster assumption is a special case of the smoothness one.

In the following sections, we will examine two typical methods based on SVM formulation, which are based on these assumptions.

2.4.1 Transductive SVM

As two classes are unlikely to be from the same cluster, the decision boundary should lie in a low-density area in between high-density regions. This is the core idea of Transductive SVM(TSVM) [Vapnik 1977], an extension of SVM for semi-supervised learning. Different from SVM, TSVM can access to unlabeled data which contains prior knowledge about the relationship between the feature space $\mathcal{X}$ and the labeling space $\mathcal{Y}$. Given ℓ labeled pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ and $(m-\ell)$ unlabeled samples $\{\mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_m\}$, TSVM can try at maximum 2^m functions (belonging to the family of hyperplanes in Euclidean space) that may separate data into two classes. The learning result is the decision boundary $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = 0$ that maximizes the margin with respect to the data labeled by the optimal labeling choice. The cluster assumption considers that the optimal hyperplane should lie in a low density region because such a placement is more likely to obtain a large margin and few misclassifications. If it is the case, then TSVM admits the following objective function

$$\begin{array}{ll} \min\limits_{\mathbf{w}, b, \hat{y}_{\ell+1}, \ldots, \hat{y}_m} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t} & y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b\right) \geq 1 \quad i = 1, \ldots, \ell \\ & \hat{y}_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b\right) \geq 1 \quad i = \ell+1, \ldots, m \\ & \hat{y}_i \in \{-1, +1\} \qquad\qquad i = 1, \ldots, m \end{array} . \tag{2.30}$$

The presence of predicted labels $\hat{y}_i$'s of test points implies that (2.30) is related to an iterative process. While optimal solvers are just capable of processing less than 100 examples, SVM Light [Joachims 1999], another TSVM implementation, produces approximated solutions for hundred thousands examples in reasonable time.

2.4.2 Laplacian SVM

Laplacian SVM [Belkin 2006] is an extension of SVM for semi-supervised learning that uses the smoothness assumption. Let us assume that the supporting structure is a manifold [1] $\mathcal{M}$, then a smooth variation along $\mathcal{M}$ is equivalent to keeping

1. According to [Lee 2007], a manifold $\mathcal{M}$ is a topological space that is locally Euclidean, meaning that around every point of $\mathcal{M}$ is a neighborhood that is topologically the same as the

small gradient changes $\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \, dP(x)$ with respect to probability density P; here $\nabla_{\mathcal{M}} f$ is the gradient of a function f with respect to $\mathcal{M}$. This idea is illustrated in Fig. 2.7. It is shown in [Lafon 2004, Belkin 2004, Belkin 2006] that this continuous integral can be approximated by a discrete sum which is the operation of graph Laplacian on an appropriate adjacency matrix built from a finite training data sampled from $\mathcal{M}$,

$$\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \, dP(x) = \int_{x\in\mathcal{M}} f\mathcal{L}_{P_{\mathcal{M}}}(f) dP(x) = \langle f, \mathcal{L}_P \rangle_{\mathcal{M}} \triangleq \mathbf{f}'\mathbf{L}\mathbf{f} \tag{2.31}$$

in which $\mathcal{L}_{P_{\mathcal{M}}}(\cdot)$ is the weighted Laplace-Beltrami operator [Belkin 2004] associated to $\mathcal{X}$ and $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ the dot-product defined on $\mathcal{M}$. On the rightmost side is the labeling vector $\mathbf{f} = [f(\phi(\mathbf{x}_1)), \ldots, f(\phi(\mathbf{x}_m))]'$ and the graph Laplacian $\mathbf{L}$ is the approximation of $\mathcal{L}_{P_{\mathcal{M}}}(f)$ on graph $G = \{\mathcal{V}, \mathcal{E}\}$ (see Fig. 2.6) whose vertex set $\mathcal{V} = \{v_i\}_{i=1}^m$ are the input data and the edge set $\mathcal{E} = \{e_{ij} | j \in \mathcal{N}(v_i) \wedge i \in \mathcal{N}(v_j)\}$ consists of connections between close data points. The notation $\mathcal{N}(v)$ denotes the neighborhood system of vertex v. In order to compute $\mathbf{L}$, one first needs to compute the weighted adjacency matrix $\mathbf{A}$ whose element $\mathbf{A}_{ij}$ equals to the similarity score of the edge e_{ij}; then $\mathbf{L} = \mathrm{diag}(\mathbf{A}\mathbf{1}_m) - \mathbf{A}$ [Chung 1997] in which $\mathbf{1}_m$ is the column vector with all 1's. A more intuitive form that explains (2.31) can be written as follows

$$\mathbf{f}'\mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \left(f(\phi(\mathbf{x}_i)) - f(\phi(\mathbf{x}_j))\right)^2 \mathbf{A}_{ij} = \frac{1}{2}\sum_{i,j=1}^{m} \left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - \langle \mathbf{w}, \phi(\mathbf{x}_j)\rangle\right)^2 \mathbf{A}_{ij} \tag{2.32}$$

The larger the weight $\mathbf{A}_{ij}$ is, the more similar the labeling results $f(\phi(\mathbf{x}_i))$ and $f(\phi(\mathbf{x}_j))$ are, and the closer the mappings $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ are. Through this regularizer, label information is diffused from labeled data to the unlabeled data. The formal construction of Laplacian SVM add the manifold regularization term into the primal form of SVM

$$\begin{array}{ll} \min\limits_{\mathbf{w}} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{2}\sum_{i,j=\ell+1}^{m} \left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - \langle \mathbf{w}, \phi(\mathbf{x}_j)\rangle\right)^2 \mathbf{A}_{ij} \\ \text{s.t} & y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b\right) \geq 1, \qquad\qquad i = 1, \ldots, \ell \end{array} . \tag{2.33}$$

The coefficient γ controls the degree of smoothness of the labeling. For further optimization procedure and empirical results see [Belkin 2004, Belkin 2006].

2.5 Subspace Methods

In this last section, we discuss subspace methods aiming to explain the data by revealing latent structures of the unknown distribution of data assumed earlier in previous sections of this chapter.

Subspace methods study mapping techniques that transform data from a high to a low-dimensional space while retaining their key characteristics. Their main

open unit ball in $\mathbb{R}^D$. For example, the Earth is spherical but looks flat at the human scale.

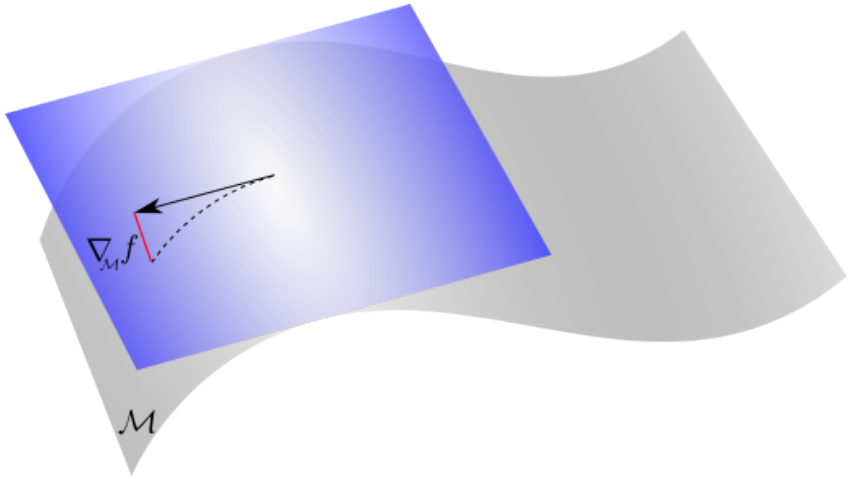


FIGURE 2.7 – This figure illustrates the fact that if a manifold $\mathcal{M}$ is smooth, then its magnitude of gradient $\|\nabla_{\mathcal{M}} f\|$ is small at any point in $\mathcal{M}$.

hypothesis is that an underlying but unknown distribution P generates the observed data and this could be related to some physical process or geometrical structure. Due to our interpretation of data, such a structure should be low-dimensional. Subspace methods, henceforth, correspond to the inverse process that eliminates irrelevant dimensions and produces a more meaningful and compact representation of the data.

This section gives a brief revision about the concepts and techniques of subspace methods. Linear techniques are presented in Section 2.5.1 and nonlinear ones are presented in Section 2.5.2 as well as 2.5.3.

2.5.1 Linear Techniques

Principal component analysis (PCA) [Jolliffe 1986] is perhaps one of the oldest and most well known methods. The goal of PCA is to find a transformation that decorrelates dimensions of the data $\mathbf{X} \in \mathbb{R}^{D \times m}$ and present them into a new subspace $\mathcal{S} \subseteq \mathbb{R}^d$ such that $d \ll D$. Assume that d bases of the new representation are columns of the matrix $\mathbf{B} \in \mathbb{R}^{D \times d}$, then any $\mathbf{x} \in \mathbb{R}^D$ is linearly approximated in terms of $\mathbf{B}$ as follows

$$\mathbf{x} \approx \mathbf{B}\boldsymbol{\alpha}, \tag{2.34}$$

where $\boldsymbol{\alpha} \in \mathcal{S}$ is the new representation of $\mathbf{x}$. Since the Euclidean subspace $\mathcal{S}$ is a span of column vectors $\{\mathbf{b}_i\}$, then these vectors are orthonormal, which means dot-products $\mathbf{b}_i'\mathbf{b}_j = \delta_{ij}$ where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise ; additionally, column vectors $\mathbf{b}_i$'s satisfy $\|\mathbf{b}_i\|_2 = 1$. Since the construction of $\mathbf{B}$ must take into account information loss minimization of the approximation (2.34), the optimal $\mathbf{B}^*$ is the solution of the following optimization problem of PCA :

$$\mathbf{B}^* \leftarrow \arg\min_{\mathbf{B}} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 \quad \text{s.t} \quad \mathbf{B}'\mathbf{B} = \mathbf{I}. \tag{2.35}$$

By substituting $\boldsymbol{\alpha} = \mathbf{B}^{-1}\mathbf{x} = \mathbf{B}'\mathbf{x}$ into (2.35), the new formula of PCA becomes

$$\min_{\mathbf{B}} \sum_{i=1}^{m} \left\| \mathbf{x}_i - \mathbf{B}\mathbf{B}'\mathbf{x}_i \right\|_2^2 \quad \text{s.t} \quad \mathbf{B}'\mathbf{B} = \mathbf{I}. \tag{2.36}$$

After expanding the $\|\cdot\|_2$ norm and eliminating irrelevant terms, we obtain m eigenvalue problems

$$\max_{\|\mathbf{B}_i\|=1, \mathbf{B}_i \perp \mathbf{B}_j} \left(\frac{\mathbf{B}_i'(\mathbf{X}'\mathbf{X})\mathbf{B}_i}{\mathbf{B}_i'\mathbf{B}_i} \right), \quad i, j = 1, \ldots, m, \tag{2.37}$$

whose objective function is the Rayleigh quotient [Horn 1990a]. Based on the Courant-Fisher theorem, for instance [Shawe-Taylor 2004], the optimal $\mathbf{B}^*$ is the matrix whose columns are d eigenvectors corresponding to d largest eigenvalues of the covariance matrix $\mathbf{X}'\mathbf{X}$. As covariance matrix is a generalization of the variance concept from two to multiple dimensions, then the eigenvectors of the top d eigenvalues of $\mathbf{X}'\mathbf{X}$ determine the top d directions where the data are scattered most. In order to reduce the dimensionality of data $\mathbf{X}$ while keeping information loss negligible, those d eigenvectors are chosen as the bases of the subspace $\mathcal{S}$.

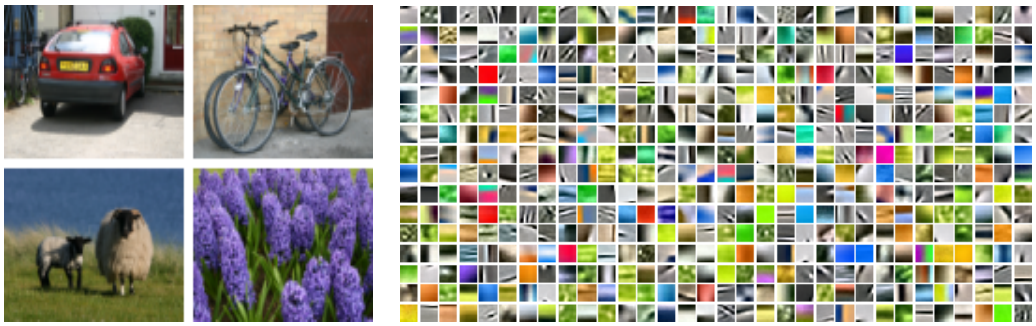
2.5.2 Sparse Coding and Dictionary Learning

In the construction of PCA, a data point $\mathbf{x}$ is approximated by d eigenvectors of d largest eigenvalues of covariance matrix $\mathbf{X}'\mathbf{X}$. Such a covariance matrix is unable to account for data nonlinearity, then PCA fails to learn subspaces for nonlinear data.

Sparse coding [Olshausen 1997] is a family of encoding algorithms that harnesses data nonlinearity via sparsity, i.e., letting the number of basis significantly larger than the dimension of the input data ($d \gg D$) and enforcing a parsimonious representation to the data. The advantage of sparsity lies in the over-completeness of the basis, which means that the basis set is diverse enough to characterize an individual data point by few vectors of the basis $\mathbf{B}$. A general formulation of sparse coding is very similar to (2.35), i.e.,

$$\arg\min_{\mathbf{B}, \boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda\psi(\boldsymbol{\alpha}_i), \tag{2.38}$$

except the regularizer $\psi(\cdot)$ which controls the sparsity of the composition (2.34). If $\lambda = 0$, sparse coding returns to PCA ; the larger λ is, the sparser the $\boldsymbol{\alpha}$ is. Popular choices for $\psi(\cdot)$ are L_0 and L_1 norms. The former counts the number of non-zero entries ; optimizing with this discrete quantity is difficult. The latter $\|\boldsymbol{\alpha}\|_1 = \sum_i |\alpha_i|$ is easier to be solved and numerically proved of having a similar effect as of the L_0 norm. There are various ways to solve (2.38), however, the general idea is to alternate between $\mathbf{B}$ and $\boldsymbol{\alpha}$ in an optimization algorithm. Shown in Fig. 2.8(b) is an illustration of a learned dictionary from image data shown in Fig. 2.8(b).



(a) Some training images (b) Over-complete basis

FIGURE 2.8 – Using sparse coding to learn an overcomplete basis from images. Based on m image patches extracted from the set of images (a), a basis $\mathbf{B}$ is learned by (2.38). As shown in (b), the basis $\mathbf{B}$ consists of visual primitives with different colors (single and gradient colors), textures (less and more ridges), and their combinations.

Sparse coding is a powerful representation for image-related tasks such as denoising, inpainting, and data compression. Fig. 2.8 shows a basis learned from natural images. Sparse representation is also used in feature extraction [Zheng 2011, Bar 2010] and classification [Ramírez 2010, Mairal 2008, Gao 2010].

2.5.3 Manifold Learning

Manifold learning is a family of subspace techniques based on *manifold assumption* which states that high-dimensional data is supported by (low-dimensional) manifolds. The goal of manifold learning is to re-embed [2] a manifold (from a high dimensional space) to a lower dimensional space such that topology structure [3] of the data is preserved.

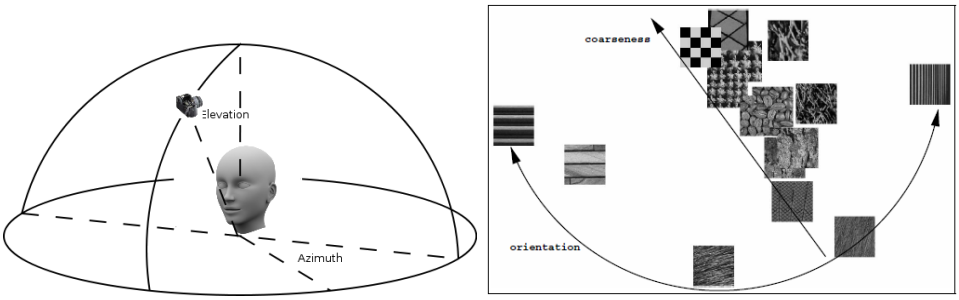
In certain situations, it is quite clear that the data are supported on a low-dimensional manifold. This is especially true for the data generated by some physical process, i.e., see examples in Fig. 2.9. By studying a finite sample of $\mathcal{M}$, which is the observed data, we expect to capture the intrinsic structures of $\mathcal{M}$. In the following, we will show that manifold learning methods achieve this goal in various ways.

2.5.3.1 Distance Preservation

This approach assumes that by an isometric mapping into the embedding space, global structure of the underlying manifold $\mathcal{M}$ is preserved. Such a method is called *isometric*, which is a mathematical concept used to denote distance-preserving mappings.

2. An embedding is a representation of a topological object (a manifold, a graph) in a certain space such that its topological properties are preserved [Lee 2007].

3. Topological structure of an object are attributes which are unchanged against deformation, twisting, and stretching. In terms of topological structure of point cloud, it denotes the intrinsic connectivity between points [Lee 2007].



(a) A face manifold. (b) A texture manifold.

FIGURE 2.9 – Manifold naturally emerges from semantic content of the data. In (a) is the manifold of faces taken with fixed distance but from various angles. In (b) is the manifold of texture patches which differ in terms of *coarseness* and *orientation*.

Multi-Dimensional Scaling. MDS is one of early manifold learning methods which are based on the Euclidean distance. Given a finite sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathcal{X}$ in which $\mathcal{X}$ is the embedded space, MDS finds a low-dimensional embedding $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ such that pairwise distances computed between $\mathbf{x}_i$'s are preserved in $\mathbf{y}_i$'s respectively. The input of MDS is the square distance matrix $\mathbf{S}$ defined as

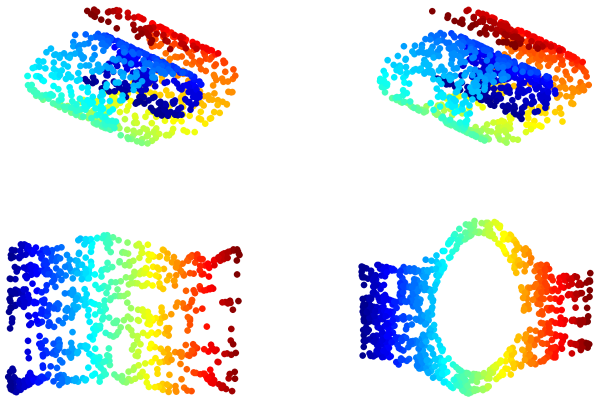
$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \ldots & s_{1n} \\ s_{21} & s_{22} & \ldots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \ldots & s_{nn} \end{pmatrix}, \tag{2.39}$$

where $s_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. It turns out that the exact coordinates $\{\mathbf{x}_i\}$ are not necessarily known and only the dissimilarity matrix $\mathbf{S}$ is necessary for MDS to work. MDS learns a low-dimensional embedding $\{\mathbf{y}_i\}$ such that the pairwise distances in the embedding space must approximate those of the embedded space. MDS's objective function, therefore, is to minimize the following problem

$$\min_{\{\mathbf{y}_i\}} \quad \sum_{i,j=1}^{m} \left(\|\mathbf{y}_i - \mathbf{y}_j\| - s_{ij}\right)^2 , \tag{2.40}$$

whose global solution is obtained by applying singular value decomposition, which is quite similar to that of PCA.

Isomap. Since the data is assumed to be nonlinear, then the underlying manifold $\mathcal{M}$ has non-zero curvature. Euclidean distance, however, cannot account for curvature. Isomap [Tenenbaum 2000] avoids the disadvantage of Euclidean distance by using geodesic distance as an alternative. Geodesic distance used in geography measures the shortest walking distance between two geographical locations on Earth. In mathematics, the notion of geodesic distance is used to indicate the shortest path between two points on a manifold. Based on the data $\{\mathbf{x}_i\}$ sampled from $\mathcal{M}$, if one can find a good approximation of the geodesic measure with respect to $\mathcal{M}$, then the topological structure of $\mathcal{M}$ is preserved.



(a) Swiss roll (b) Swiss roll with hole

FIGURE 2.10 – *Top row :* the input data ; *Bottom row :* the learned embeddings of Isomap. According to (b), Isomap is not suitable for nonconvex data as it wrongly estimates true distances between points surrounding the hole (compare the holes in the input data and in the resulting embedding).

Isomap approximates $\mathcal{M}$ by constructing an adjacency graph $\{\mathcal{V}, \mathcal{E}\}$ whose vertex set $\mathcal{V}$ comprises the data $\{\mathbf{x}_i\}$ while the edge set $\mathcal{E}$ comprises connections between vertices with similar appearances. Edge weights represent distances between data points in the embedded space. Based on the defined graph, geodesic distances between vertex pairs are computed using shortest path algorithms such as Kruskal. Storing these distances dissimilarity matrix $\mathbf{S}$, Isomap uses the algorithm of MDS in order to compute the embedding.

2.5.3.2 Topology Preservation

Recall that MDS and Isomap are based on isometry property in which the learned embedding preserves exact pairwise distances of the data. This condition is rather strict. Although Isomap accounts for the curvature of $\mathcal{M}$, geodesic distance could not be approximated well if the given data is not uniformly sampled or $\mathcal{M}$ contains holes (see Fig. 2.10). Nevertheless, an alternative is to approximate $\mathcal{M}$ using an adjacency graph and via preserving the connectivity of the graph, the topology of $\mathcal{M}$ is preserved.

Locally linear embedding. LLE [Roweis 2000] seeks to preserve *local isometry*. It means that isometry property at local areas of manifold $\mathcal{M}$ must be preserved in the embedding space. In term of graph connectivity, the distances of the edges connecting the vertex of interest and its neighborhoods must be preserved. This is a mild condition compared with the "global" isometry of MDS or Isomap.

Given an arbitrary vertex v_i and its k adjacent vertices $\{v_j | e_{ij} > 0\}$, LLE finds the best reconstruction $\hat{\mathbf{x}}_i$ of $\mathbf{x}_i$ based on a weighted linear combination of the neighborhoods $\mathbf{x}_j$, i.e., $\hat{\mathbf{x}}_i \approx \sum_j \delta(e_{ij}) w_{ij} \mathbf{x}_j$, in which $\delta(e_{ij}) = 1$ if $e_{ij} > 0$ and $\delta(e_{ij}) = 0$ otherwise. The following objective function is minimized with the optimizer $\mathbf{W}^*$:

$$\mathbf{W}^* \leftarrow \arg\min_{\mathbf{W}} \frac{1}{2} \sum_{i=1}^{m} \left\| \mathbf{x}_i - \sum_{j=1}^{m} \delta(e_{ij}) w_{ij} \mathbf{x}_j \right\|^2 \tag{2.41}$$

in which w_{ij}'s are entries of $\mathbf{W}$ at row i and column j. Assume that the local isometry holds, then $\mathbf{W}$ is reusable for reconstructions $\mathbf{y}_i \approx \sum_j w_{ij} \mathbf{y}_j$ of the embedding $\{\mathbf{y}_i\}$. This embedding is found as the optimizer of the following quadratic program

$$\{\mathbf{y}_i^*\} \leftarrow \arg\min_{\{\mathbf{y}_i\}} \frac{1}{2} \sum_{i=1}^{m} \left\| \mathbf{y}_i - \sum_{j=1}^{m} \delta(e_{ij}) w_{ij} \mathbf{y}_j \right\|^2 . \tag{2.42}$$

Laplacian Eigenmaps. LE [Belkin 2001] aims to preserve graph connectivity, which is associations between vertices and their neighborhoods. In other word, LE brings the neighborhood system of the graph from the embedded into the embedding space. Technically, LE's objective function penalizes any non-smooth variation of the embedding's coordinates between adjacent vertices. The neighborhood relation is determined as pairwise similarity $w_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2\right)$ between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. Since w_{ij} decreases exponentially according to the dissimilarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, then w_{ij} is positive within a small range with respect to bandwidth σ. If w_{ij} is large, then $\mathbf{y}_i$ and $\mathbf{y}_j$ must be close to each other ; we do it by forcing the pairwise distances $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ to be minimized, i.e.,

$$\begin{array}{ll} \min\limits_{\mathbf{Y}} & \frac{1}{2}\text{trace}\left(\mathbf{YLY}'\right) \\ \text{s.t} & \mathbf{YDY}' = \mathbf{I} \end{array} , \tag{2.43}$$

in which $\mathbf{Y} \in \mathbb{R}^{d \times m}$ is the low-dimensional embedding and $\text{trace}\left(\mathbf{YLY}'\right) = \sum_{ij} w_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2$; the equality constraint $\mathbf{YDY}' = \mathbf{I} \Leftrightarrow \mathbf{D}_{ii} \mathbf{y}_i' \mathbf{y}_i = \mathbf{I}_{ii}, \forall i$, where the diagonal matrix $\mathbf{D}$ has its diagonal entries $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. This constraint prevents the embedding to have any arbitrary scaling. In the simplest case where $d = 1$, let us denote $\mathbf{z} \in \mathbb{R}^m$ being the first row of $\mathbf{Y}$, then (2.43) becomes

$$\begin{array}{ll} \min\limits_{\mathbf{z}} & \frac{1}{2}\mathbf{z}'\mathbf{L}\mathbf{z} \\ \text{s.t} & \mathbf{z}'\mathbf{D}\mathbf{z} = 1 \end{array} . \tag{2.44}$$

The problem (2.44) is equivalent to the minimization of the generalized Rayleigh quotient [Horn 1990a] :

$$\min_{\mathbf{z}} \frac{\mathbf{z}'\mathbf{L}\mathbf{z}}{\mathbf{z}'\mathbf{D}\mathbf{z}}. \tag{2.45}$$

The solution of (2.45) is the eigenvector of the second smallest eigenvalue of the generalized eigen problem $\mathbf{Lz} = \lambda \mathbf{Dz}$. If $d > 1$, then the 2^{nd} dimension of the

embedding will get the eigenvector of the $(2+1)^{\text{rd}}$ smallest eigenvalue, and so on and so forth. If the Laplacian matrix $\mathbf{L}$ is normalized $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ then (2.45) becomes similar to the canonical Rayleigh quotient (similar to the case of PCA). Notice that $\mathbf{L}$ is always positive semidefinite because for all $\mathbf{z}$, then

$$\mathbf{z}'\mathbf{L}\mathbf{z} = \sum_{ij} W_{ij}(z_i - z_j)^2 \geq 0., \tag{2.46}$$

The only difference in solving PCA and LE is that the use of the linear covariance matrix $\mathbf{X}'\mathbf{X}$ versus the graph Laplacian $\mathbf{L}$.

2.6 Summary

In this chapter we have introduced necessary theoretical background for contributions to be presented later. This chapter consists of two parts ; the classification task is presented from Sections 2.1 to 2.4 and the data representation task is presented from Section 2.5 to the end. The former discusses classification techniques which map data into feature spaces such that the data are best categorized ; the latter discusses unsupervised embedding techniques which learn low-dimensional feature space and preserve topology of data.

Chapitre 3

Transductive Kernel Learning

In the context of classification, we tackle kernel learning problem from transduction perspective. Kernel-based methods are conventionally known to be efficient to classify data in a high (possibly infinite) feature space in which the mapping function associated with the kernel is not necessarily known, which leads to reduced computational effort. This method, however, encounters limitations in tunning parameters of predefined kernel functions in order to fit to the data ; furthermore, the computational cost of the kernel matrices becomes expensive for large-scale databases. We instead propose to learn an explicit kernel map with bounded dimensionality. The finite dimensionality of a kernel map may provide better generalization. Additionally, by adopting the transductive approach (see Section 2.4), our method can exploit unlabeled data when labeled data are scarce. We investigate our method in three vision problems : object recognition, image interpretation, and image annotation. The formulation in this chapter is built on background knowledge presented in Chapter 2. The formulation is followed by an efficient optimization algorithm which helps us to solve the interactive object segmentation problem. Benchmarked on the VOC Pascal 2011 dataset, quantitative results demonstrate that our method is at least comparable with state of the art.

This work was published in the following paper :

1. Phong Vo, Hichem Sahbi, *Transductive Inference & Kernel Design for Ojbect Class Segmentation*, IEEE ICIP, USA 2012.

3.1 Introduction

According to Section 2.1, existing machine inference techniques may be categorized into *inductive* and *transductive*. The former consists in finding a decision function from a labeled training set, and uses that function in order to generalize across unlabeled data. Among popular inductive techniques support vector machines (SVMs) [Vapnik 1998a, Schölkopf 2001a] are well studied and proved to be performant in many real-world applications including object recognition, text analysis, and bioinformatics [Maji 2008, Joachims 2002a, Asa 2008]. The success of SVMs is highly dependent on the choice of kernels ; existing ones include the linear, the gaussian and the histogram intersection.

Usual kernels may not be appropriate in order to capture the actual and the "semantic" similarity between data for some specific tasks, so appropriate kernels require considerable work on model selection and parameters tuning. For instance [Ong 2005, Cristianini 2001, Chapelle 2002] adapt the parameters of existing kernels (such as the order of the polynomial kernel or the scale of the gaussian) using quality assessment functions and generalization bounds. Other approaches, consider kernel design as a feature selection problem [Grandvalet 2002] or distance learning [Chatpatanasiri 2010, Jain 2009, Kulis 2010]. Alternatives including hyperkernels [Ong 2005] and multiple kernel learning (MKL) [Bach 2004, Wu 2006, Rakotomamonjy 2008, Bach 2008b, Sonnenburg 2006] directly learn kernel (gram) matrices from training data. MKLs are particularly successful and their principle consists in finding linear combinations of standard kernels using the L_1, L_2 or mixed norms [Cortes 2009a, Rakotomamonjy 2011, Aflalo 2011]. Going beyond linear combinations of kernels, [Varma 2009] extends traditional MKL to other combinations and even though the underlying optimization problem is no longer convex, the performance is better. Similarly, polynomial combination of basic kernels is also proposed in [Cortes 2009c]. Further discussions can be found in Section 2.3.

Even-though performant, the success of these inductive kernel based methods also depends on cardinality of the labeled data. For some applications labeled data is rare and expensive ; only a very small fraction of training data is labeled and the unlabeled data may not follow the same distribution as the labeled one, so learning kernels using inductive inference techniques is clearly not appropriate (see a toy example in Fig. 3.1). Alternative approaches [Lanckriet 2004b, Belkin 2006, Zhou 2003a] may include the unlabeled data as a part of the learning process and this is known as transductive inference [Vapnik 1977, Vapnik 1998a, Chapelle 2006a] (see Section 2.4). This concept was pioneered by Vapnik (see for instance [Vapnik 1977]). It relates to semi-supervised learning and relies on the i) smoothness assumption which states that close data in a high-density area of the input space, should have similar labels [Zhou 2003a, Belkin 2006] and ii) the cluster assumption which finds decision rules in low density areas of the input space [Narayanan 2006, Chapelle 2005]. In that context, transductive versions of SVMs were also introduced [Joachims 1999] ; they build decision functions by optimizing the parameters of a learning model together with the labels of the unlabeled data. This turned out to be very useful in

order to overcome the limited cardinality of the labeled data w.r.t the number of training parameters.

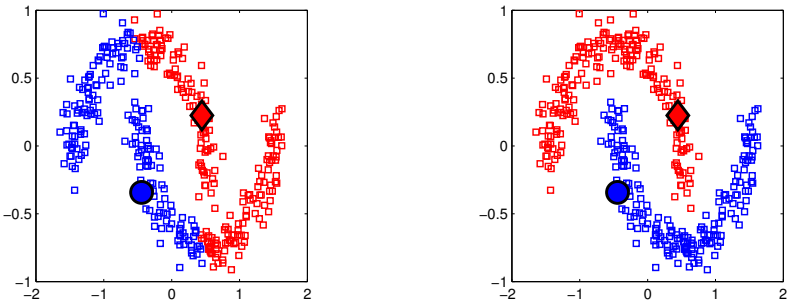
While various kernel learning algorithms have been developed for inductive setting, little work is achieved for transductive kernel learning. Existing related work includes semidefinite programming (SDP) [Lanckriet 2004a], alignment based kernels [Cristianini 2001] and manifold based kernel learning subject to constraints on angles and distances [Weinberger 2004]. Regardless their effectiveness, some of these techniques are expensive and hardly extensible to large scale training problems.

We introduce a novel transductive learning algorithm for kernel design and classification. Our method is based on a constrained matrix factorization which produces *a* kernel map that takes data from the input space into a high dimensional space in order to guarantee their linear separability while maximizing their margin. This margin property, however, and as known [Vapnik 1998a], does not necessarily guarantee good generalization performance on the unlabeled set, if the latter is drawn from a different probability distribution compared to the labeled data. Therefore and beside maximizing the margin, our transductive approach includes a regularization term that enforces smoothness in the resulting kernel map in order to correctly diffuse labels to the unlabeled data. Additionally, the rank of learned kernel map is reduced in order to maintain good generalization. Following our formulation, and in contrast to MKL, our learning model is not restricted to only convex linear combinations of existing kernels ; indeed it is model-free. Furthermore, it also takes advantage from both labeled and unlabeled data and this results into better generalization performances as corroborated by our experiments.

The rest of this chapter is organized as follows. Section 3.2 presents our transductive learning and kernel design approach. Its optimization algorithm is presented Section 3.3 ; experiments with the interactive object segmentation is explained in Section 3.4. We conclude the chapter and discuss about future extensions in Section 3.5.

3.2 Problem Formulation

Define $\mathcal{X} \subseteq \mathbb{R}^n$ as an input space corresponding to all the possible image features and let $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \ldots, \mathbf{x}_m\}$ be a finite subset of $\mathcal{X}$ with an arbitrary order. This order is defined so only the first ℓ labels of $\mathcal{S}$, denoted $\{y_1, \ldots, y_\ell\}$ (with $y_i \in \{-1, +1\}$), are known. In many real-world applications only a few data is labeled (i.e., $\ell \ll m$) and its distribution may be different from the unlabeled data. We can view $\mathcal{S}$ as a matrix $\mathbf{X}$ in which the i^{th} column corresponds to $\mathbf{x}_i$. Our objective is to build both a decision criterion and an optimal *k*ernel map in order to infer the unknown labels $\{y_{\ell+1}, \ldots, y_m\}$.



(a) Inductive (b) Transductive

FIGURE 3.1 – This figure shows classification results on the "two moon" example in [Belkin 2006]. In (a) an inductive method is used for classification ; In (b) a transductive technique is used instead and it exploits the density of the unlabeled data. In this example labeled data are marked with "diamond" and "circle" and correspond to the positive and the negative classes respectively.

3.2.1 Max-margin Inference and Kernel Design

Inductive learning aims to build a decision function f that predicts a label y for any given input data $\mathbf{x}$; this function is trained on $\mathcal{S}' = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ and used in order to infer labels on $\mathcal{S} \backslash \mathcal{S}'$. In the max-margin classification [Vapnik 1998a], we consider ϕ as a mapping of the input data (in $\mathcal{X}$) into a high dimensional space $\mathcal{H}$. The dimension of $\mathcal{H}$ is usually sufficiently large (possibly infinite) in order to guarantee linear separability of data.

Assuming data linearly separable in $\mathcal{H}$, the max-margin inductive learning finds a hyperplane f (with a normal $\mathbf{w}$ and shift b) that separates ℓ training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ while maximizing their margin. The margin is defined as twice the distance between the closest training samples with respect to f and the optimal $(\hat{\mathbf{w}}, \hat{b})$ correspond to

$$\begin{array}{ll} \underset{\mathbf{w},b}{\operatorname{argmin}} & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t} & y_i\left(\langle \mathbf{w}, \phi(x_i)\rangle + b\right) \geq 1, \quad i = 1, \ldots, \ell, \end{array} \tag{3.1}$$

which is the primal form of the hard margin support vector machine [Vapnik 1998a]. Given $\mathbf{x}_i \in \mathcal{S} \backslash \mathcal{S}'$, the class of $\mathbf{x}_i$ in $\{-1, +1\}$ is decided by the sign of $f(\mathbf{x}_i) = \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b$. Following the kernel trick [Vapnik 1998a], one may show that $f(\mathbf{x}_i)$ can also be expressed as $\sum_{j=1}^{\ell} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) + b$, here $(\alpha_1 \ldots \alpha_\ell)'$ is a vector of positive real-valued training parameters and $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$ is a symmetric, continuous, positive (semi-definite) kernel function [Schölkopf 2001a]. The closed form of $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is defined among a collection of existing kernels including linear, gaussian and histogram intersection ; but the underlying mapping $\phi(\mathbf{x}) \in \mathcal{H}$ is usually *i*mplicit, i.e., it does exist but it is not necessarily known and may be infinite dimensional.

We propose in the remainder of this section a new approach that builds explicit and finite dimensional kernel maps. In contrast to usual kernels, such as the Gaussian RBF, the VC-dimension [Vapnik 1998a] – related to a finite dimensional kernel map – is finite [1]. According to Vapnik's VC-theory [Vapnik 1977], the finiteness of the VC-dimension avoids loose generalization bounds and may guarantee better performance.

3.2.2 Enforcing Low Rank Kernels

Now, we turn the problem into finding the hyperplane f as well as a Gram (kernel) matrix $\mathbf{K} = \mathbf{\Phi}'\mathbf{\Phi}$ where each column $\mathbf{\Phi}_i$ corresponds to an explicit mapping of $\mathbf{x}_i$ into a finite dimensional space (i.e., $\phi(\mathbf{x}_i) = \mathbf{\Phi}_i$). This mapping is designed in order to i) guarantee linear separability of data in $\mathcal{S}$, ii) to ensure good generalization performance by maximizing the margin, iii) to approximate the input data, and also iv) to ensure positive definiteness of $\mathbf{K}$ by construction, i.e., without adding further constraints. This results into the following constrained minimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{B},\mathbf{\Phi},\mathbf{w}} & \frac{1}{2}\left\|\mathbf{X} - \mathbf{B}\mathbf{\Phi}\right\|_F^2 + \frac{1}{2}\left\|\mathbf{w}\right\|_2^2 + \frac{\mu}{2}\|\mathbf{\Phi}\|_F^2 \\ \text{s.t} & y_i\mathbf{w}'\mathbf{\Phi}_i \geq 1, \quad i = 1, \ldots, \ell \\ & \|\mathbf{B}_i\|_2^2 = 1, \, \forall i = 1, \ldots, p \end{array} \tag{3.2}$$

here $\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}\mathbf{A}')$ stands for the square of the Frobenius norm and $\mathbf{X} \approx \mathbf{B}\mathbf{\Phi}$ is factorized using an overcomplete basis $\mathbf{B} \in \mathbb{R}^{n \times p}$ (i.e., $p > n$) and a new kernel map $\mathbf{\Phi} \in \mathbb{R}^{p \times m}$. Without a loss of generality b is omitted in the above expression as it can be induced from $\mathbf{w}$ and the mapping $\mathbf{\Phi}$.

As discussed earlier, and according to [Vapnik 1998a], the VC-dimension (related to a family of classifiers) depends also on the dimension of the learned kernel map and this may affect generalization, especially if this dimension is very high. Since the actual (intrinsic) dimension of the learned kernel map $\mathbf{\Phi}$ is unknown, we choose the number of basis p to be sufficiently large such that the first inequality constraint in (3.2) can be satisfied and the left-hand side term tends to zero for an infinite number of solutions.

First, p is overestimated to $\max(\ell, n) + 1$, and this guarantees that the above constrained minimization problem has a solution. Then, the actual (intrinsic) dimension is found by regularizing (3.2) by the Frobenius norm $\|\mathbf{\Phi}\|_F^2$ which has similar effect as the nuclear norm where $\mu \geq 0$ controls the rank of $\mathbf{K}$. Indeed, the squared Frobenius norm is exactly the L_2-norm on the eigenvalues of $\mathbf{K}$ and it is less likely to shrink these eigenvalues into zeros compared to the L_1-norm (which is the nuclear norm). Nevertheless, as will be shown later, it provides a closed form kernel solution and our experiments show that it indeed reduces the rank of the kernel map (see Lemma 3.2.1 below) while allowing us to derive a simple optimization algorithm [2].

1. The VC-dimension is the maximum number of data samples, that can be shattered, whatever their labels.
2. We tried to optimize (3.5) using nuclear norm; details are presented in Appendix B.

Lemma 3.2.1 *For any matrix* $\mathbf{\Phi} \in \mathbb{R}^{p \times m}$*, the following inequality holds*

$$\|\mathbf{\Phi}\|_* \leq \sqrt{r}\|\mathbf{\Phi}\|_F, \tag{3.3}$$

where the Frobenius norm $\|\mathbf{\Phi}\|_F = \sqrt{\sum_{i=1}^{\min\{p,m\}} \sigma_i^2}$*; the nuclear norm* $\|\mathbf{\Phi}\|_* = \sum_{i=1}^{\min\{m,p\}} \sigma_i$*;* $r = rank(\mathbf{\Phi}) = rank(\mathbf{\Phi}'\mathbf{\Phi})$ *and* σ_i*'s are eigenvalues of the Gram matrix* $\mathbf{K} = \mathbf{\Phi}'\mathbf{\Phi}$*.*

Proof. See for instance [Horn 1990b, Golub 1996].

3.2.3 Transduction Setting

For a better conditioning of (3.2), the smoothness assumption is introduced to kernel maps. This makes it possible to design smooth kernel maps and to assign similar predictions to neighboring data for a better generalization on the unlabeled ones (see toy example in Fig. 3.2).

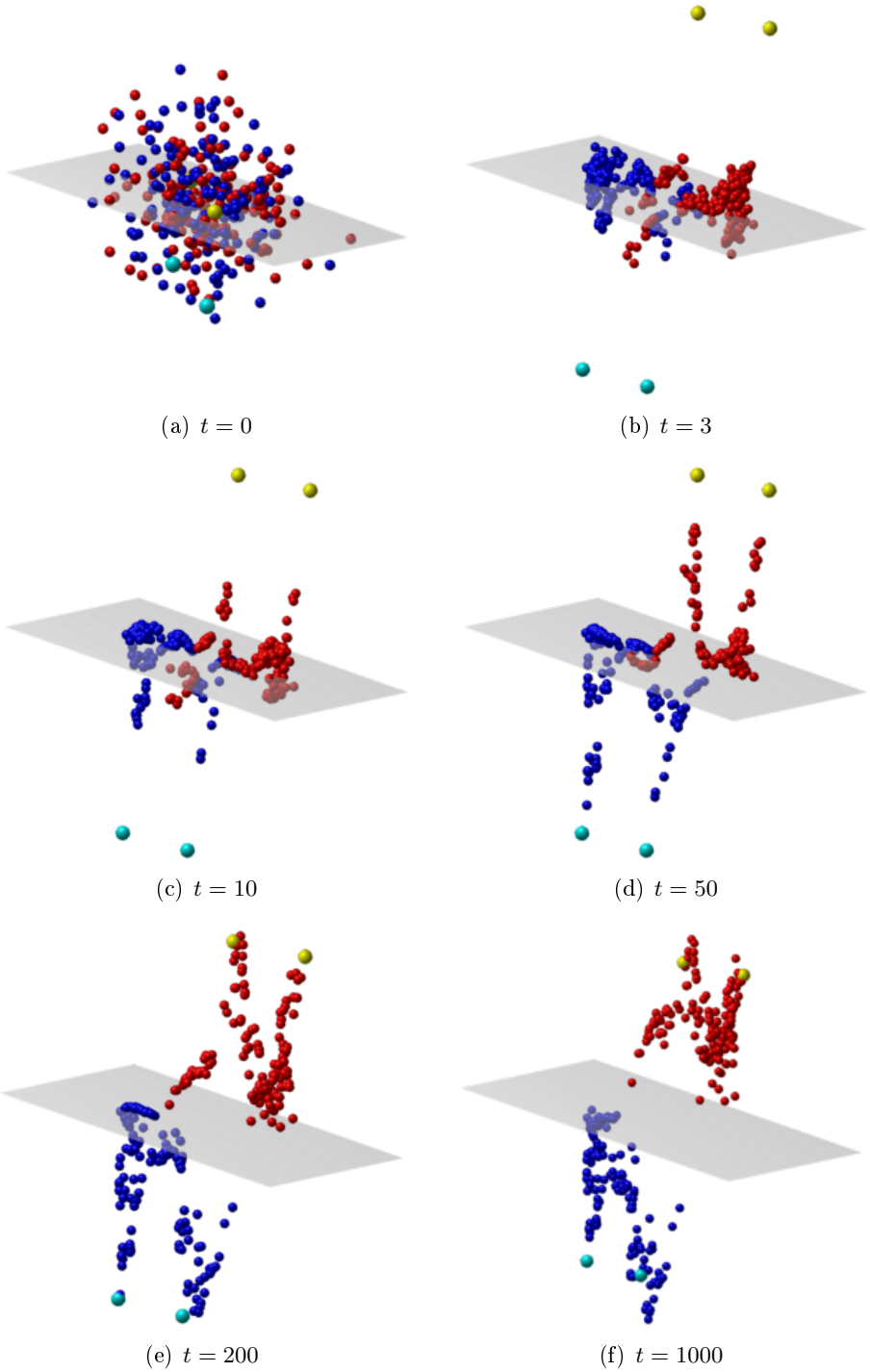
We model the input data $\mathcal{S}$ using an adjacency graph $\{\mathcal{V}, \mathcal{E}\}$ where nodes $\mathcal{V} = \{v_1, \ldots, v_m\}$ correspond to samples $\{\mathbf{x}_i\}$ and edges $\mathcal{E} = \{e_{ij}\}$ are the set of weighted links of the graph. In the above definition, $\mathbf{x}_i \in \mathbb{R}^n$ is a feature vector (color, texture, etc.) while $e_{ij} = (v_i, v_j, \mathbf{A}_{ij})$ defines a connection between v_i, v_j weighted by $\mathbf{A}_{ij}$. The latter is defined as $\mathbf{A}_{ij} = 1_{\{v_j \in \mathcal{N}_k(v_i)\}} \cdot s(\mathbf{x}_i, \mathbf{x}_j)$, here the neighborhood $\mathcal{N}_k(v_i)$ of a given node v_i, includes the set of the k-nearest neighbors of v_i. Notice that the neighborhood system is designed in order to guarantee that $\forall v_i, v_j \in \mathcal{V}$, $v_j \in \mathcal{N}_k(v_i)$ implies $v_i \in \mathcal{N}_k(v_j)$ and vice-versa. The function $s(\cdot, \cdot)$ measures the similarity between two given points $\mathbf{x}_i$ and $\mathbf{x}_j$, and we set it in practice to either the RBF or the histogram intersection functions.

Given that two vertices v_i and v_j are connected and that the weight $\mathbf{A}_{ij}$ of the edge e_{ij} is large, the smoothness constraint requires that v_i and v_j having similar label, which mean $f(x_i) = \mathbf{w}'\mathbf{\Phi}_i$ approximately equals $f(x_j) = \mathbf{w}'\mathbf{\Phi}_j$. The larger the weight $\mathbf{A}_{ij}$ is, the smoother the labeling between v_i and v_j is. At the scope of the whole graph $\{\mathcal{V}, \mathcal{E}\}$, the following energy term must be minimized

$$\frac{\beta}{4} \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\mathbf{w}'\mathbf{\Phi}_i - \mathbf{w}'\mathbf{\Phi}_j\right)^2 \mathbf{A}_{ij} = \frac{\beta}{2} \mathbf{w}'\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\mathbf{w}, \tag{3.4}$$

where $\beta \geq 0$ and $\mathbf{L}$ is the graph Laplacian defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}_m)$ and $\mathbf{1}_m$ is column vector of length m with all its entries equal to one.

When adding this regularizer into the objective function (3.2) and replacing its inequality constraints by the squared loss in order to group with the matrix factorization term, we obtain the complete form of our transductive learning problem



(a) $t = 0$ (b) $t = 3$

(c) $t = 10$ (d) $t = 50$

(e) $t = 200$ (f) $t = 1000$

FIGURE 3.2 – This figure shows the evolution of the learned kernel map through different iterations of our method (see Algorithm 1). This map is found for the popular "two moon" example in [Belkin 2006]. The underlying 2D input data are not linearly separable, while the learned kernel map makes them linearly separable in a 3D space. In these experiments, only $\ell = 4$ samples were labeled (shown in cyan and yellow resp. for the positive and the negative classes).

$$\min_{\mathbf{B},\mathbf{\Phi},\mathbf{w}} \quad \underbrace{\frac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \frac{1}{2}\mathbf{w}'\left(\mathbf{I}_p + \beta\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\right)\mathbf{w}}_{\text{regularization}} + \underbrace{\frac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{1\times p} & \mathbf{w}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2}_{\text{data term}},$$
$$\text{s.t} \quad \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \tag{3.5}$$

with $\mathbf{I}_p$ the $p \times p$ identity matrix, $\mathbf{C}$ is the diagonal $m \times m$ matrix for which the i^{th} diagonal element is fixed to 1 for a labeled sample, and 0 for an unlabeled one, and similarly, $\mathbf{Y} \in \mathbb{R}^{1\times m}$ has its i^{th} element equal to y_i for a labeled data, and 0 for an unlabeled one.

3.3 Optimization

It is clear that the minimization problem (3.5) is not convex jointly with respect to $\mathbf{B}, \mathbf{\Phi}, \mathbf{w}$. We consider an alternating optimization procedure by solving three subproblems : we first maximize the margin $2/\|\mathbf{w}\|$ with respect to $\mathbf{w}$ and we update the basis $\mathbf{B}$, then we minimize the regularization criterion, the rank and the reconstruction error with respect to $\mathbf{\Phi}$. This process is repeated until convergence ; i.e., all the unknowns remain unchanged from one iteration to another. Different steps of the algorithm are shown in Algorithm (1) ; the superscript (t) is added to $\mathbf{w}$, $\mathbf{B}$ and $\mathbf{\Phi}$ in order to show the evolution of their values through different iterations of the learning process.

Algorithm 1 Transductive kernel map learning

Input : labeled $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{m}$
Initialization : compute adjacency matrix $\mathbf{A}$, degree matrix $\mathbf{D}$, graph Laplalcian $\mathbf{L}$, $t \leftarrow 0$, set $\mathbf{\Phi}^{(0)}$ to a full-rank random matrix.
Repeat steps (1+2) **until** convergence OR $t > t_{\max}$

1. Update $\mathbf{w}^{(t+1)}$ and $\mathbf{B}^{(t+1)}$ using (3.6), (3.7) respectively.
2. Update $\mathbf{\Phi}^{(t+1)}$ by taking the limit $\tilde{\mathbf{\Psi}}$ of (3.10), with $\mathbf{\Psi}^{(0)} = \mathbf{\Phi}^{(t)}$.

Output : kernel maps $\{\mathbf{\Phi}_i^{(t+1)}\}_{i=\ell+1}^{m}$ and labels $\{y_i\}_{i=\ell+1}^{m}$ with $y_i = \left(\mathbf{w}^{(t+1)}\right)' \mathbf{\Phi}_i^{(t+1)}$.

3.3.1 Learning Basis and Classifier

Assuming fixed $\mathbf{\Phi}^{(t)}$ (denoted simply as $\mathbf{\Phi}$) and enforcing the gradient of (3.5) to vanish (with respect to $\mathbf{w}$) leads to

$$\mathbf{w}^{(t+1)} = \alpha\left(\mathbf{I}_p + \mathbf{\Phi}\left(\alpha\mathbf{C} + \beta\mathbf{L}\right)\mathbf{\Phi}'\right)^{-1}\mathbf{\Phi}\mathbf{C}\mathbf{Y}'. \tag{3.6}$$

Similarly, we find $\mathbf{B}^{(t+1)}$ as

$$\begin{array}{ll} \underset{\mathbf{B}}{\text{argmin}} & \frac{1}{2}\left\|\mathbf{X}-\mathbf{B}\boldsymbol{\Phi}\right\|_F^2 \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \end{array} \tag{3.7}$$

and its dual function is

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{B}}\left(\frac{1}{2}\left\|\mathbf{X}-\mathbf{B}\boldsymbol{\Phi}\right\|_F^2 + \sum_{i=1}^{p}\lambda_i\left(\|\mathbf{B}_i\|_2^2 - 1\right)\right) \tag{3.8}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ldots \lambda_i \ldots \lambda_p]$ are the Lagrange multipliers associated with p equality constraints in (3.7). The minimizer of the primal problem (3.7) can be obtained by finding $\boldsymbol{\lambda}^*$ that maximizes (3.8) using Newton method as reported in [Lee 2006]. The analytic solution for basis update is therefore $\mathbf{B}^{(t+1)} = \mathbf{X}\boldsymbol{\Phi}'\left(\boldsymbol{\Phi}\boldsymbol{\Phi}' + \text{diag}(\boldsymbol{\lambda}^*)\right)^{-1}$.

When the data scale up or the dimensionality of $\mathbf{X}$ grows, solving the dual problem using Newton method, which requires inverting a $p \times p$ matrix at each Newton iteration, becomes impractical. Alternatives include using gradient descent methods such as block coordinate descent [Mairal 2009] and projected gradient descent [Mazumder 2009].

3.3.2 Learning Kernel Map

Considering fixed $\mathbf{B}^{(t+1)}$ and $\mathbf{w}^{(t+1)}$ (denoted simply as $\mathbf{B}$, $\mathbf{w}$ in the remainder of this section), and the previous kernel map solution $\boldsymbol{\Phi}^{(t)}$, our goal is to find $\boldsymbol{\Phi}^{(t+1)}$ by solving (3.5). Conditions for the existence of this new kernel map solution $\boldsymbol{\Phi}^{(t+1)}$ are given in the following proposition.

Proposition 3.3.1 [3] *Let $\|.\|_1$ denote the entrywise L_1-norm. Provided that the following inequality holds,*

$$\beta < \|\mathbf{w}\mathbf{w}'\|_1^{-1}.\|\mathbf{A}\|_1^{-1}, \tag{3.9}$$

the optimization problem (3.5) admits a unique solution $\boldsymbol{\Phi}^{(t+1)} = \tilde{\boldsymbol{\Psi}}$ as the limit of

$$\boldsymbol{\Psi}^{(\tau+1)} = \psi\left(\boldsymbol{\Psi}^{(\tau)}\right), \tag{3.10}$$

here $\psi : \mathbb{R}^{p\times m} \to \mathbb{R}^{p\times m}$ is defined as $\psi(\boldsymbol{\Psi}) = \left(\psi_1(\boldsymbol{\Psi}) \ldots \psi_m(\boldsymbol{\Psi})\right)$, with each column vector $\psi_i(\boldsymbol{\Psi})$ as

$$\psi_i(\boldsymbol{\Psi}) = \left(\mu\mathbf{I} + \alpha\mathbf{B}'\mathbf{B} + (\alpha\mathbf{C}_{ii} + \beta\mathbf{D}_{ii})\,\mathbf{w}\mathbf{w}'\right)^{-1}.\left[\alpha\left(\mathbf{B}'\mathbf{X} + \mathbf{w}\mathbf{Y}\mathbf{C}\right) + \beta\mathbf{w}\mathbf{w}'\boldsymbol{\Psi}\mathbf{A}\right]_i, \tag{3.11}$$

$[.]_i$ stands for the i^{th} column of a matrix. Furthermore, and for a sufficiently large μ, the kernel maps $\boldsymbol{\Psi}^{(\tau)}$ in (3.10) satisfy the convergence property :

$$\left\|\boldsymbol{\Psi}^{(\tau)} - \tilde{\boldsymbol{\Psi}}\right\|_1 \leq L\left\|\boldsymbol{\Psi}^{(0)} - \tilde{\boldsymbol{\Psi}}\right\|_1, \tag{3.12}$$

with $L = \beta\|\mathbf{w}\mathbf{w}'\|_1.\|\mathbf{A}\|_1$ and $\boldsymbol{\Psi}^{(0)} = \boldsymbol{\Phi}^{(t-1)}$.

3. A more intuitive bound of the smoothness coefficient β, regarding its relations to classification margin ρ and data quantity m, is presented in Appendix C.

Proof. Following (3.5), let us consider the function defined on the set of matrices in $\mathbb{R}^{p\times m}$

$$E : \mathbf{\Psi} \mapsto \frac{\mu}{2}\|\mathbf{\Psi}\|_F + \frac{1}{2}\mathbf{w}'\Big(\mathbf{I}_p + \beta\mathbf{\Psi}\mathbf{L}\mathbf{\Psi}'\Big)\mathbf{w} + \frac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{1\times p} & \mathbf{w}'\end{pmatrix}\begin{pmatrix}\mathbf{\Psi}\\ \mathbf{\Psi C}\end{pmatrix}\right\|_F^2 \tag{3.13}$$

The necessary condition of the fixed-point relation in (3.10) results from $\partial E/\partial\mathbf{\Psi} = 0$ (details about derivative are omitted in this proof). We will now prove that the function ψ is L-Lipschitzian, with $L = \beta\|\mathbf{w}\mathbf{w}'\|_1.\|\mathbf{A}\|_1$.
Let us denote the left-hand side (inverse) matrix in (3.11) simply as $\mathbf{Z}_i$ and introduce $g(\mathbf{\Psi}) = (g_1(\mathbf{\Psi})\dots g_m(\mathbf{\Psi}))$ with $g_i(\mathbf{\Psi}) = \mathbf{Z}_i^{-1}\psi_i(\mathbf{\Psi})$.

Given two matrices $\mathbf{\Psi}^{(1)}$ and $\mathbf{\Psi}^{(2)}$ in $\mathbb{R}^{p\times m}$, we have

$$\begin{aligned}
&\sum_{i=1}^{m}\left\|\mathbf{Z}_i^{-1}\psi_i(\mathbf{\Psi}^{(1)}) - \mathbf{Z}_i^{-1}\psi_i(\mathbf{\Psi}^{(2)})\right\|_1\\
=\;&\sum_{i=1}^{m}\left\|g_i(\mathbf{\Psi}^{(1)}) - g_i(\mathbf{\Psi}^{(2)})\right\|_1\\
=\;&\left\|g(\mathbf{\Psi}^{(1)}) - g(\mathbf{\Psi}^{(2)})\right\|_1\\
=\;&\beta\left\|\mathbf{w}\mathbf{w}'(\mathbf{\Psi}^{(1)} - \mathbf{\Psi}^{(2)})\mathbf{A}\right\|_1\\
\le\;&\beta\left\|\mathbf{w}\mathbf{w}'\right\|_1.\left\|\mathbf{A}\right\|_1.\left\|\mathbf{\Psi}^{(1)} - \mathbf{\Psi}^{(2)}\right\|_1\\
\le\;&L\left\|\mathbf{\Psi}^{(1)} - \mathbf{\Psi}^{(2)}\right\|_1,\ \text{with}\ \ L = \beta\left\|\mathbf{w}\mathbf{w}'\right\|_1.\left\|\mathbf{A}\right\|_1.
\end{aligned} \tag{3.14}$$

By taking the free parameter μ (in $\mathbf{Z_i}$) sufficiently large

$$\begin{aligned}
&\sum_{i=1}^{m}\left\|\mathbf{Z}_i^{-1}\psi_i(\mathbf{\Psi}^{(1)}) - \mathbf{Z}_i^{-1}\psi_i(\mathbf{\Psi}^{(2)})\right\|_1\\
=\;&\sum_{i=1}^{m}\left\|\left[\psi_i(\mathbf{\Psi}^{(1)}) - \psi_i(\mathbf{\Psi}^{(2)})\right].\mathbf{Z}_i^{-1}\right\|_1\\
\ge\;&\sum_{i=1}^{m}\left\|\psi_i(\mathbf{\Psi}^{(1)}) - \psi_i(\mathbf{\Psi}^{(2)})\right\|_1\\
=\;&\left\|\psi(\mathbf{\Psi}^{(1)}) - \psi(\mathbf{\Psi}^{(2)})\right\|_1
\end{aligned} \tag{3.15}$$

Combining (3.14) and (3.15), we get

$$\left\|\psi(\mathbf{\Psi}^{(1)}) - \psi(\mathbf{\Psi}^{(2)})\right\|_1 \le L\left\|\mathbf{\Psi}^{(1)} - \mathbf{\Psi}^{(2)}\right\|_1$$

. □

The process described in (3.10) allows us to recursively diffuse the kernel maps from the labeled to the unlabeled data, through the neighborhood system defined in the graph $\mathcal{G}$. This process is iterative and may require many steps before convergence. The latter is reached when $\|\mathbf{\Psi}^{(\tau)} - \mathbf{\Psi}^{(\tau-1)}\| \leq \varepsilon$; (in practice, $\varepsilon = 10^{-2}$, and convergence usually happens in less than $\tau_{\max} = 100$ iterations, see Fig. 3.3).

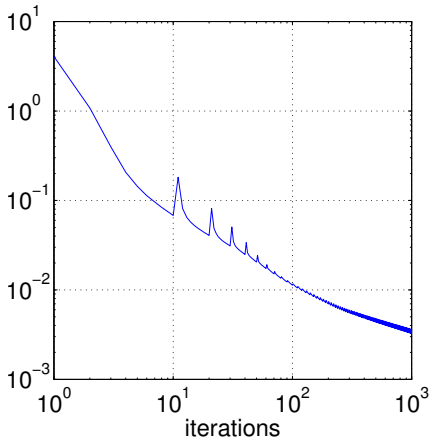


FIGURE 3.3 – This figure illustrates the convergence process on the particular example of Fig. 3.1, i.e., the difference between current and previous estimate of kernel maps through different iterations. The ragged points are resulted from the incompatibility between the kernel map learned at iteration $(t-1)$ (based on $\mathbf{w}^{(t-1)}$ and $\mathbf{B}^{(t-1)}$) and the kernel map learned at iteration t (based on newly updated $\mathbf{w}$ and $\mathbf{B}$).

Optimization Complexity. Based on the optimization procedure introduced above, we test the elapsed time required in order to obtain a baseline performance. We use another toy data example in order to generate from few hundreds to tens of thousands points ; training data are randomly picked at 5 percent of the sampled points. More details about data generation is commented in Fig. 3.4. The experiment is tested on a workstation with quadcore 2Ghz and the parameters of the algorithm are $t_{\max} = 3$, $\tau_{\max} = 50$, $\alpha = \beta = 1$, $\mu = 10^{-8}$, $k = 4$. Shown in Fig. 3.5 are the complexity curve of time required for the algorithm to obtain classification accuracies varying between 60 to 70 percent. With few hundred data points, it takes less than a sencond to finish the optimization ; several thousands of data points requires from several seconds to a minute ; with half a million points, it takes approximately 1.5 hours for our algorithm to finish iterations.

Convergence Speed. We show how the proposed classification algorithm performs on the ORL face dataset (see Fig. 3.6) which contains 10 different face images for every 40 distinct subjects. For every subject, two images are randomly selected as training data. 40 binary problems are evaluated and their results are summarized in Fig. 3.7. As the curves suggest, high classification precision is obtained early, i.e

just after few iterations ; subsequent iterations are mainly about reducing kernel maps dimensionality.

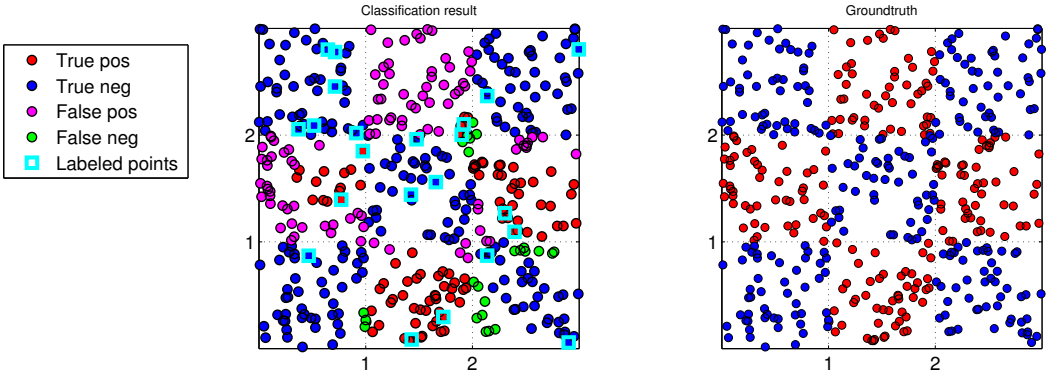


FIGURE 3.4 – The toy data are sampled from a squared chessboard with equal numbers of positive and negative classes ; 5 percent of sampled points are randomly picked as training data. We generate the chessboard data with dimensions ranging from 2^1 to 2^5 cells ; 50 points are sampled at each cell so that data quantity varies from 200 to 51200. Shown above is an example of how the data looks like (right) and the classification result (left).

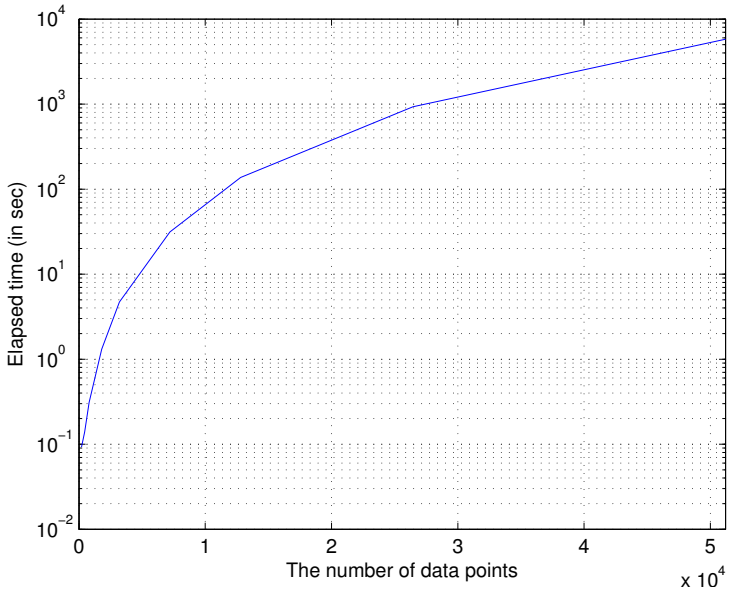


FIGURE 3.5 – The time complexity of our algorithm to obtain baseline accuracy on chessboard data when the number of data points increasing from few hundreds to tens of thousands.

FIGURE 3.6 – Some examples of the ORL face images.

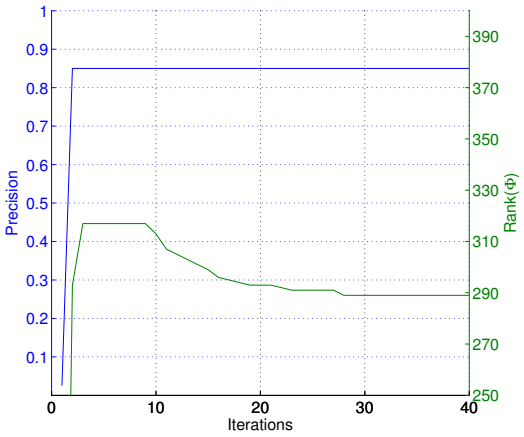


FIGURE 3.7 – The obtained precision and rank of learned kernels maps as they evolve over iterations.

3.4 Experiments

We use the Pascal VOC 2011 dataset [4] in order to evaluate the performance of our transductive inference method on object class segmentation (OCS) whose processing pipeline is shown in Fig. 3.9. For that purpose, we sample from the VOC database 556 images containing object instances of 21 classes (20 object classes and the background class). For every image, object instances belonging to these 21 categories need to be recognized and segmented from background. Similarly to recent works, we approach the segmentation problem at the superpixel-level rather than pixel-level; this approach not only reduces computational resources but also improve segmentation quality.

In our experiments, images are subdivided using an irregular grid (neighborhood system) of 700 superpixels, each one is processed in order to extract four visual descriptor types [Tighe 2010] : position, texture, SIFT, and color. As a result, every superpixel is characterized by its visual appearance and position information. While the former supports connections between visually similar superpixels in the graph,

4. http ://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/index.html

TABLE 3.1 – This table shows the list of features used to describe superpixels. All feature vectors, excepting position, are normalized using L_1-norm prior to their concatenation. See [Tighe 2010, Malisiewicz 2008] for more details.

Type	Description	Dimension
Position	Absolute Mask	$8 \times 8 = 64$
	Top Height	1
	Bottom Height	1
Texture	Interior texton histogram	100
SIFT	Interior SIFT histogram	100
Color	RGB mean	3
	RGB std. dev.	3
	RGB Color Histograms	$11 \times 3 = 33$

the latter supports connections between local superpixels. For more details about the construction of descriptors and their dimensions, see Table. 3.1.

For every image, we turn OCS into a transductive inference problem where only a small fraction of its underlying superpixels is labeled (see Fig. 3.12, third column). Different from the example in Fig. 3.9, test images are not annotated by the human user but simulated by computer. We randomly choose a subset of superpixels of a test image as the labeled data. A transductive classifier is trained for each category and we combine these classifiers using the "winner-take-all" strategy in order to infer the category of a given unlabeled superpixel.

We use the standard protocol of Pascal VOC 2011 in order to evaluate segmentation accuracy. It is defined as the average accuracy across 21 classes. For each of class, the accuracy is computed as the ratio between the number of correctly labeled pixels and the number of pixels in the union area of the segmented result and the ground truth, i.e.,

$$\text{accuracy} = \frac{\text{true pos.}}{\text{true pos.} + \text{false pos.} + \text{false neg.}}. \tag{3.16}$$

In the equation above, true pos. means the number of pixels correctly labeled as the ground truth ; false pos. means the number of pixels incorrectly labeled ; false neg. means the number of missing pixels. Since the difficulty of an OCS problem lies in the ratio between the labeled and unlabeled data, we compare average accuracies at different percentages of annotation.

3.4.1 Settings and Performance

Different settings were experimented for our method including the size of the neighborhood (denoted k) when building the graph $\{\mathcal{V}, \mathcal{E}\}$. The choice of these parameters will be discussed in the remainder of this section.

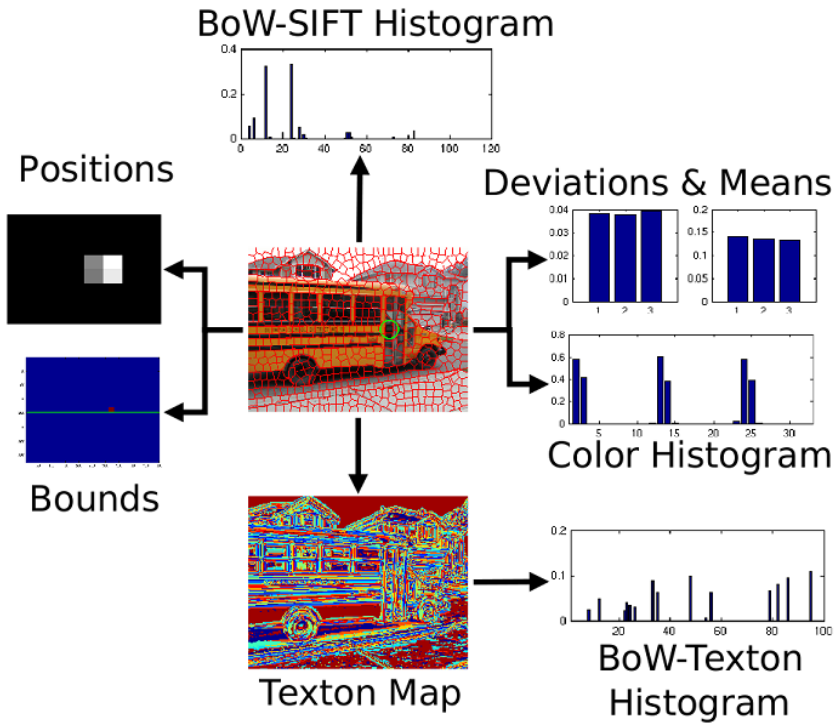
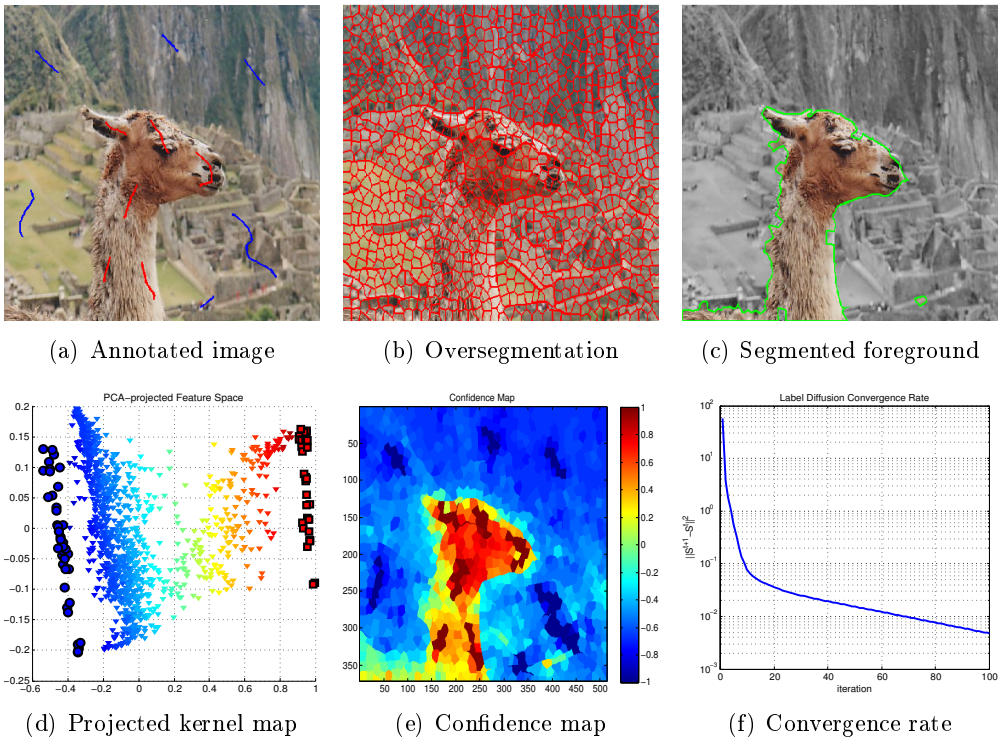


FIGURE 3.8 – For every superpixel of a test image, four descriptors types – three visual and one positional descriptors – are extracted. These descriptors are used in order to construct the affinity graph for our transductive setting.

Graph topology. The neighborhood size k of the graph is very dependent on the topology of the data. An appropriate selection of k should avoid short-cuts (overestimated k) and missing-connections (underestimated k). Shown in Fig. 3.10(a) are the accuracy curves of three values of k in which $k = 3$ and $k = 6$ give the best mean accuracy at any annotation rate. We explain that the optimal neighborhood size $k = 6$ approximately equals to the average number of neighboring superpixels (see Fig. 3.9(b) with zooming). We fix this optimal choice $k = 6$ for subsequent experiments.

Regularization & Rank reduction Shown in Fig. 3.10(b) are the average accuracy curves as increasing functions of α (almost quasi-constant for larger values of α). Given $\beta = 0.1$ fixed, these curves achieve their optimal mean accuracies when $\alpha = 10$ and this satisfies our convergence criterion in Proposition 3.3.1 (see example in Fig. 3.3).

Fixing $\alpha = 0.1$, we analyze the effect of the smoothness regularizer. Fig. 3.10(c) reports average accuracy for different values of the regularization parameter β; note that $\beta = 0$ corresponds to the baseline inductive setting (i.e., no regularization is applied). As increasing β from 0.001 to 0.1, the mean accuracy is increased too and it achieves the best performance when $\beta = 0.1$. However, the increase rate of the

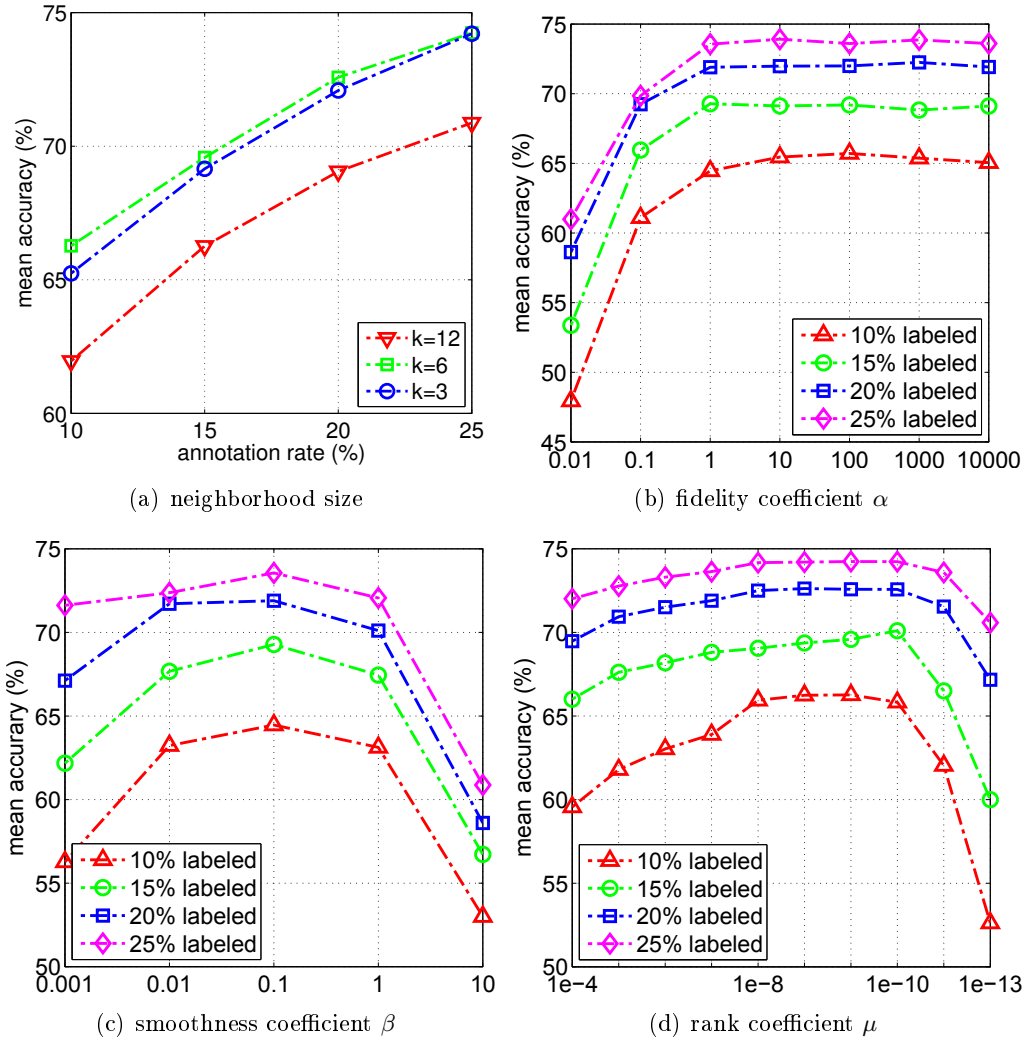


(a) Annotated image (b) Oversegmentation (c) Segmented foreground

(d) Projected kernel map (e) Confidence map (f) Convergence rate

FIGURE 3.9 – The processing pipeline of interactive object segmentation. Initially the user annotates some representative foreground & background regions in the image (a) ; the annotated image is over-segmented into superpixels (b) ; based on the data in which the labeled ones are the annotated superpixels and the unlabeled ones are the unannotated superpixels, our transductive kernel learning infers the complete foreground object as shown in (c). In (d) is the visualization of the learned kernel map ; the kernel map is projected into 2D space using PCA in which round blue points indicate background labeled data and square red points indicate labeled foreground data. Unlabeled data (triangle points) are assigned color with respect to their relative distances with respect to the positive and negative labeled data. Fig. (e) shows the prediction map in which hotter or cooler colors correspond to more confident predictions of the positive or negative class respectively. The convergence rate of this inference process is shown in (f).

curve with 25% labeled data is less than the others. It means that the smoothness regularizer takes a more important role if data is highly insufficient. If the learned kernel map is over-smoothed, i.e., $\beta = 10$, the mean accuracy is quickly degraded. According to these results, an underestimated β results into noisy segmentation while an overestimated β makes the segmentation results very smooth which leads to lose object details since regularization applies to both foreground and background classes ; since background tends to occupy more labeled data than foreground, the smoothness favors the one with larger number of labeled data. Larger values of β result in more superpixels to be labeled as background class. Examples in Fig. 3.12



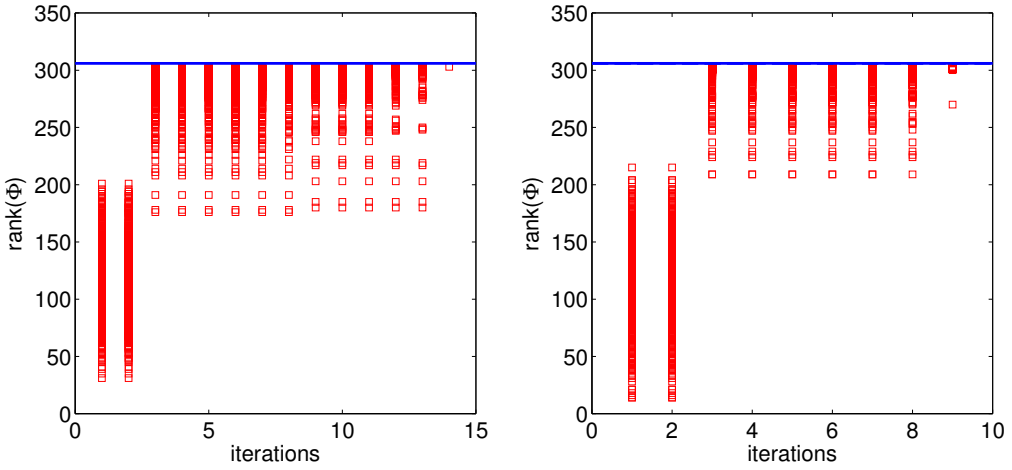
(a) neighborhood size (b) fidelity coefficient α

(c) smoothness coefficient β (d) rank coefficient μ

FIGURE 3.10 – The evolution of the average accuracy with respect to our algorithm's parameters.

show such evolutions of the segmentation results.

Finally, Fig. 3.10(d) shows the evolution of accuracy with respect to the parameter μ. From these figures, it is clear that larger values of μ favor low rank kernels while maintaining high accuracy. If the rank is unbounded as $\mu \to 0$, the kernel map is ill-conditioned and the mean accuracy is dropped. With an appropriate value of μ, rank of the kernel map steadily increases after every iteration of the optimization algorithm and it converges to the upper-bound $\max(\ell, n) + 1$ with respect to the number of iterations t. However, the coefficient μ determines how fast this convergence is. Fig. 3.11 illustrates that fact that the smaller the coefficient μ is, the more rapidly the rank is raised. Rather than considering μ as the coefficient of the rank regularizer, it is also regarded as the regularization on the fluctuation of kernel map ranks. For example. if μ is larger, then the rank regularizer restricts the rep-



(a) $\mu = 10^{-4}$ (b) $\mu = 10^{-8}$

FIGURE 3.11 – Rank evolution of kernel map during an optimization process. Red squares at every iteration t depict rank distribution of kernel maps over 556 segmentation problems. In the first and second steps, i.e., $t = 1$ and $t = 2$, these distributions are scattered and then progressively shrink as t grows. If t is large enough, these distributions reduce to few points which are upperbounded by $\max(\ell, n) + 1$ (blue solid line). Smaller μ requires fewer iterations for the optimization to converge).

resentation to be learned along the direction which is orthogonal to the separating hyperplane (see Fig. 3.2). Therefore, it μ is set too large, then the learned kernel map is less capable of discriminating the data. In our experiments, μ is kept at 10^{-8}.

3.4.2 Comparison

Our method is compared with inductive as well as transductive approaches : SVM [Cortes 1995], MKL-SVM [Rakotomamonjy 2008], TranSVM [Joachims 1999], LapSVM [Belkin 2006]. For all these comparisons, we fix our parameters as $\alpha = 10$, $\beta = 10^{-1}$, $\mu = 10^{-10}$, $\varepsilon = 10^{-2}$, $\tau_{\max} = 20$, and $t_{\max} = 5$. In all these experiments, our proposed method has an average run time of 1.008(s) per image on a quadcore 2Ghz PC while SimpleMKL requires 1.337(s), SVM 0.084(s), Transductive SVM 0.271(s), and Laplacian SVM 0.082(s).

Our method versus Inductive learning. In our experiments, inductive approaches include SVM classifiers [Vapnik 1998a] with four different kernels (linear, RBF, χ^2, and histogram intersection). The Gaussian RBF kernel is optimally tuned with respect to its bandwidth parameter while other kernels are parameter-free. For the implementation part of SVM, we use additive kernel SVMs [Maji 2008]. Additionally, we also compare our method with multiple kernel learning (MKL-SVM) that learns the kernel by linearly combining predefined kernel functions. By associating 4 kernel functions mentioned above with 5 descriptors (Table 3.1), we train

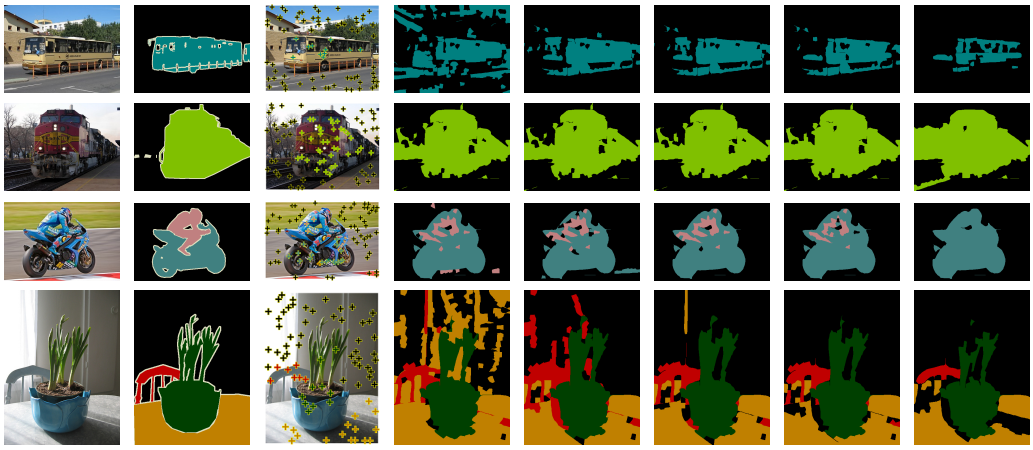
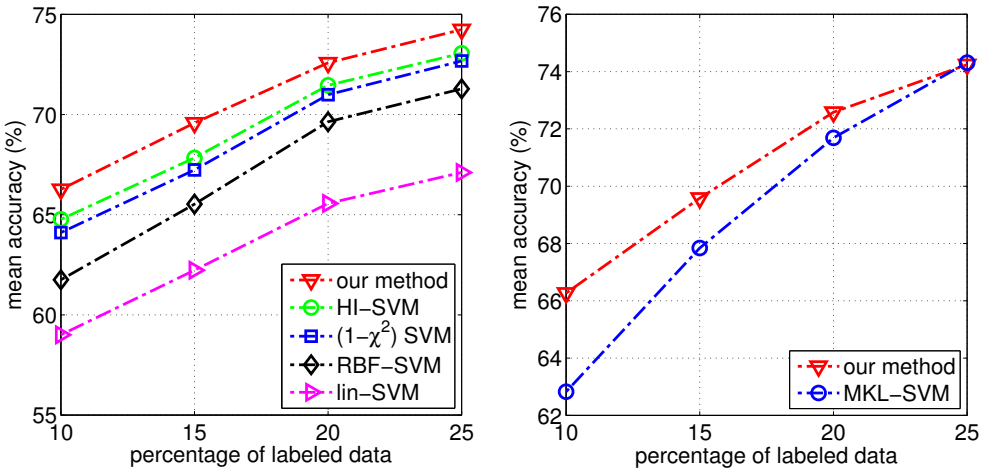


FIGURE 3.12 – The effect of the smoothness regularizer with respect to the coefficient β. $\mathbf{1}^{\text{st}}$ column : test images ; $\mathbf{2}^{\text{nd}}$ column : ground truths ; $\mathbf{3}^{\text{rd}}$ column : annotated superpixels ; $\mathbf{4}-\mathbf{8}^{\text{th}}$ columns : segmentation results with $\beta = 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$. In all the four examples above, the smoothness regularizer favors classes having the highest quantities of labeled data.



(a) compare with SVMs (b) compare with MKL

FIGURE 3.13 – Comparison between our algorithm and state-of-the-arts of inductive learning methods.

MKL via SimpleMKL [5] [Rakotomamonjy 2008] using a pool of 20 Gram matrices.

As shown in Fig. 3.13(a), 3.13(b), our method outperforms the inductive classifiers, with various kernels as well as their combination using MKL-SVM, and the accuracy of the inductive techniques and our method become more and more similar as the percentage of labeled data increases. Our first conclusion is that the proposed method is very suitable to learn a classifier especially when the fraction of labeled

5. http ://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html

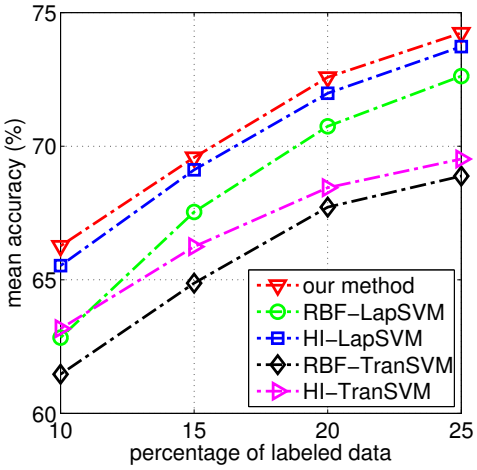


FIGURE 3.14 – Comparison between our algorithm and related transductive methods.

data is very small and the second conclusion is that the learned kernel map is more appropriate for classification than linear combination of kernels.

Our method versus Related transductive methods. Transductive approaches, used for comparison, include Laplacian-SVM[6] and transductive SVM[7] and their implementations can be downloaded from their author's homepages. The optimal parameters (C^*, γ^*) used by SVM are reused by TranSVM. The weights for regularization terms of LapSVM $\gamma_I = 0.79$ and $\beta = 0.01$ produce good classification results. According to the result presented in Fig. 3.14, our method consistently outperforms Transductive SVMs and Laplacian SVMs ; note that the latter also relies on regularization with a setting similar to our, (i.e., the same graph Laplacian and graph construction) but our method has an extra advantage of optimizing the kernel map resulting into a more suitable data representation for classification.

3.5 Summary

We introduced in this chapter, a new transductive learning approach for kernel design and classification. The strength of our contribution resides in the variational framework that allows us to explicitly design an "optimal" kernel map as a part of the learning process. When compared to baseline inductive methods, multiple kernel learning and also related transductive methods, our approach shows competitive performance on the challenging object class segmentation task. As shown in experiments, a smooth segmentation result depends on the coefficient β ; it must be set not too big otherwise the result is over-smoothed. This is due to data imbalance which frequently happens in practice. In subsequent chapters, we investigate extensively

6. http ://www.dii.unisi.it/∼melacci/lapsvmp/
7. http ://svmlight.joachims.org/

how our formulation is tailored in order to exploit image context in problems of image annotation and scene understanding.

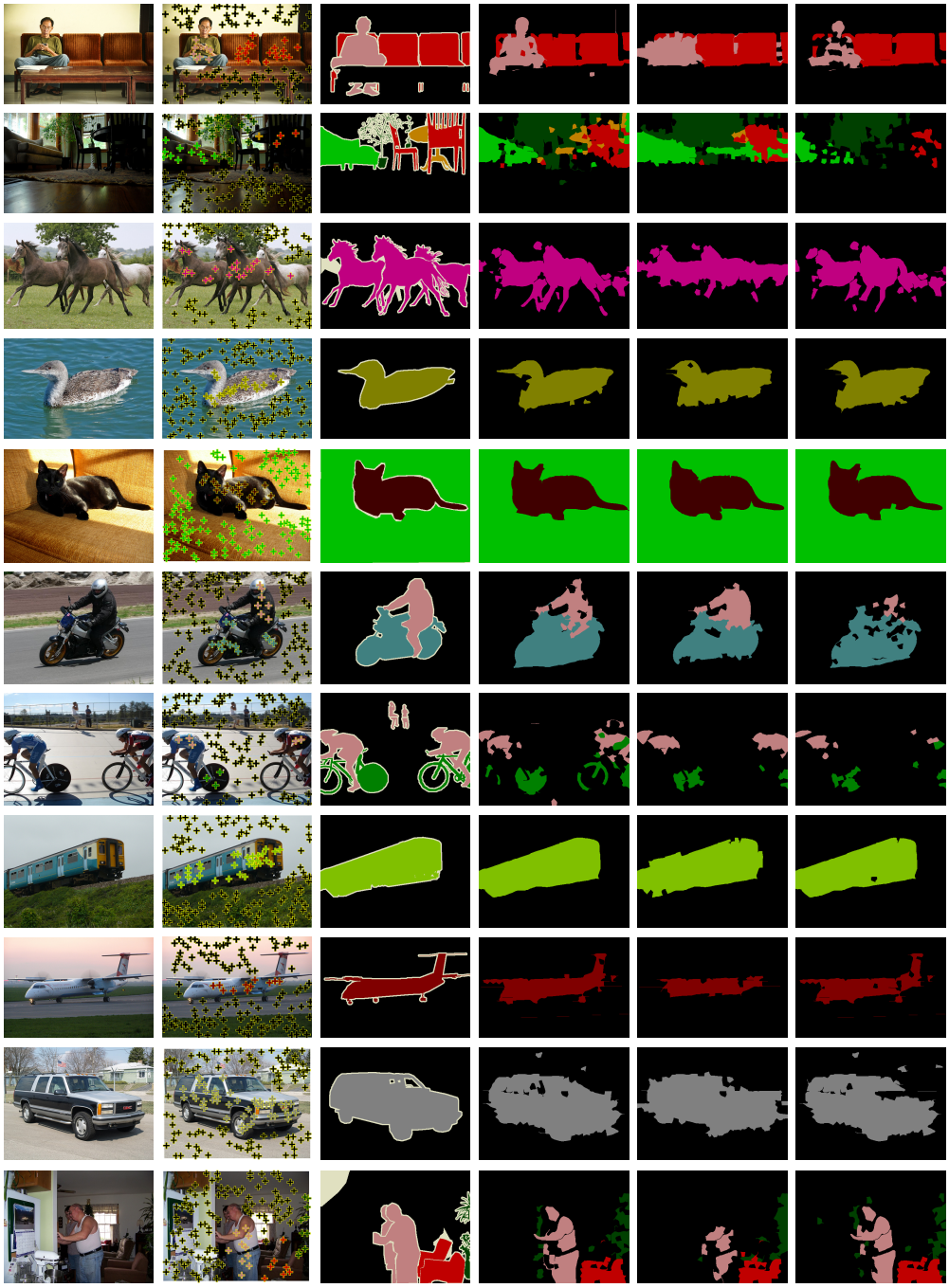


FIGURE 3.15 – **1**st column : test image ; **2**nd column : simulated annotations ; **3**rd column : ground truths ; **4**th column : our method ; **5**th column : MKL-SVM ; **6**th column : LapSVM. Our method outperforms MKL-SVM in segmenting articulated objects (see 3^{rd}, 4^{th}, and 9^{th} rows) and evenly outperforms LapSVM in segmenting complex scenes which contain more than two object classes (see 1^{st}, 2^{nd}, 6^{th}, 7^{th} rows). These examples demonstrates that our learned kernels are more appropriate than conventional kernels in OCS problem.

Chapitre 4

Multi-class Kernel Learning for Image Annotation

As mentioned in the thesis title, our concern is about the creation of machine learning methods in solving problems of image interpretation and search. In the previous chapter, our transductive kernel learning goes beyond the naive use of existing kernels and their restricted combinations ; our method is able to learn in a transductive setting a "model-free" kernel map capable of explaining the training data and generalize well on unseen data. A first application to image interpretation – interactive object class segmentation – was introduced in the previous chapter. We present in this chapter another application of our learning method for image search. In particular, we study the automatic image annotation problem, which is the key part of any image search system that uses keywords in order to index images by semantic concepts. Our contribution of this chapter is twofold : (i) we extend the transductive kernel learning formula for multi-class classification, and (ii) we incorporate regularization by modeling dependency between labels. Experiments conducted on image annotation show that our method achieves at least comparable results with related state of the art methods using the MSRC and the Corel5k databases.

Parts of this work were mentioned in the followings papers :

1. Phong Vo, Hichem Sahbi, *Transductive Kernel Map Learning and Its Applications to Image Annotation*, BMVC, UK, 2012.

4.1 Introduction

With the exponential growth of multimedia sharing spaces, such as social networks, visual contents are nowadays abundant. Searching these large collections requires a preliminary step of image annotation that translates visual contents into labels also known as keywords or concepts (see for instance [Duygulu 2002]). Automatic image annotation is challenging due to the perplexity when assigning many possible labels to images and the difficulty to analyze rich and highly semantic contents. In annotation, image observations are first described using low-level features (color, texture, shape, etc.), and labels are then assigned to images using variety of inference techniques such as hidden Markov models [Ephraim 1989], latent Dirichlet allocation [Blei 2003], probabilistic latent semantic analysis [Hofmann 1999], and support vector machines (SVMs) [Cortes 1995]. These inference techniques are used in order to model the correspondence between low level features and labels and allow us to predict keywords for unlabeled images.

Among existing image annotation approaches, machine learning ones are particularly successful and may be categorized into generative and discriminative. Generative methods model a priori knowledge and dependencies between image observations and their possible labels using for instance graphical models [Lavrenko 2003, Fan 2004, Mensink 2011, Ulges 2011]. In these models, the annotation process is based on maximizing a posterior probability using a variety of network inference techniques. This category of methods even though relatively successful suffers from complexity in modeling and inference especially when labels are taken from a large scale vocabulary. Alternative approaches are discriminative and consider image annotation as a classification problem [Carneiro 2007, Feng 2004, Russakovsky 2010a]. A vocabulary of labels is first defined, and a decision criterion is then learned for each label and used in order to identify images belonging to that label.

The aforementioned categories of machine learning techniques are highly dependent on the learned concepts and may fail when the latter are highly semantic and difficult to model. In order to overcome these issues, recent discriminative approaches consider a priori knowledge and relationships between data and the learned concepts (context, shared features, etc.) [Ulges 2011, Xue 2011, Li 2010, Tsai 2011]. The success of these image annotation methods also depends on cardinality of the labelled data and the choice of the appropriate setting for learning. The inductive setting [Deng 2010, Deng 2011, Farhadi 2009, Russakovsky 2010a] consists in building a decision function for each concept using labelled images, and uses that function in order to generalize across unlabelled images. In these methods, labelled data are usually scarce and expensive ; only a very small fraction of training images is labelled and the unlabelled images may not follow the same distribution as the labelled ones, so learning using inductive techniques is clearly not appropriate.

Alternatives [Belkin 2006, Vapnik 1977] may include the unlabelled data as a part of the learning process and this is known again as transductive inference. The concept of transductive inference, or transduction, was pioneered by Vapnik (see for instance [Vapnik 1977]). It relates to semi-supervised learning [Chapelle 2006a]

and relies on the i) smoothness assumption which states that close data in a high-density area of the input space, should have similar labels [Chapelle 2006a] and ii) the cluster assumption which finds decision rules in low density areas of the input space [Chapelle 2006a]. Learning consists in building decision functions by optimizing the parameters of a learning model together with the labels of the unlabelled data (see for instance [Belkin 2006, Joachims 2002a, Joachims 1999, Melacci 2011]). When applied, these transductive methods turned out to be very useful in order to overcome the limited cardinality of the labelled images in image annotation [Fergus 2009, Ma 2011, Yuan 2011, Wang 2009, Chen 2011b].

Among popular learning techniques support vector machines [Cortes 1995] are well studied and proved to be performant in image annotation [Grangier 2008] ; in SVMs, kernels are used in order to model visual similarity between images, and only images sharing the same concepts are expected to have high kernel values. The success of SVMs is therefore, highly dependent on the choice of kernels and usual ones, such the linear, the Gaussian RBF and the histogram intersection, may not be appropriate in order to capture the actual and the semantic similarity between images for some specific concepts.

Better inductive kernels are obtained by learning metric distance functions [Chatpatanasiri 2010, Kulis 2010, Guillaumin 2009, Makadia 2008, Feng 2013] ; other transductive kernels were designed using semidefinite programming [Lanckriet 2004a]. In order to take extra advantage from different settings, (inductive) multiple kernels (MKL) were also introduced [Bach 2004, Wu 2006, Rakotomamonjy 2008, Bach 2008b, Sonnenburg 2006] and consider convex (and possibly sparse) linear combinations of elementary kernels and proved to be more suitable [Varma 2009]. With the current state of the art, MKL are considered as one of the most effective kernel design and combination techniques. Nevertheless, MKL based design hits at least two major limitations ; On the one hand, and as mentioned earlier, these methods are limited by the cardinality of labelled data and they do not rely on any extra information in order to overcome that limitation, on the other hand they are mainly restricted to linear combinations of existing kernels only.

In the previous chapter we proposed the transductive kernel learning (TKL) algorithm which is based on a constrained matrix factorization which produces a kernel map that takes image data from the input space into a high dimensional space in order to guarantee their linear separability while maximizing their margin. In this respect, transductive kernel learning is not restricted to only convex linear combinations of existing kernels [Rakotomamonjy 2008, Vishwanathan 2010b] ; indeed it is *model-free*. Beside maximizing the margin, its transductive approach includes a regularization term that enforces smoothness and low rankness in the resulting kernel map in order to diffuse label information from training to test data. Due to the availability of both the training and test data, the learning outcome obtains better generalization performance. In this chapter we propose the multi-class transductive kernel learning algorithm and apply it to the problem of image annotation. Compared to the binary formula (3.5) where its use in multi-class problems

requires running many binary classifiers, the multi-class formula allows sharing one kernel map to many classifiers which significantly reduces time and computational resources.

Besides, a multi-class classification model allows us to model thematic relationships between classes (for instance, see [Ulges 2011, Tsai 2011]). Such relationships can be modeled as statistical dependencies between semantically related labels [He 2009]. Based on the multi-class formula, we design a label-dependency model which is built on semantic dependency of labels in an image ; it promotes highly related labels to co-occur in which the relatedness is represented by label co-occurrence frequencies computed based on training data. Simultaneously, both the *relatedness* between labels and *similarity* between images are incorporated into our new formula. As corroborated by our image annotation experiments, empirical results have shown that the novel algorithm performs well on the MSRC and Corel5K datasets while being robust against imbalanced data.

The remainder of this chapter is organized as follows. We update our transductive learning approach and kernel design in Section 4.2 for multiple classes and the implementation of our optimization procedure in Section 4.3. We illustrate in Section 4.4.3 the application of our method to image annotation using two datasets ; MSRC and Corel5K. We conclude the chapter in Section 4.5 while providing a possible extension for a future work.

4.2 Method

4.2.1 Mathematical Notations

Define $\mathcal{X} \subseteq \mathbb{R}^n$ as an input space corresponding to all the possible image features and let $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \ldots, \mathbf{x}_m\}$ be a finite subset of $\mathcal{X}$ with an arbitrary order. This order is defined so only the first ℓ label vectors of $\mathcal{S}$, denoted $\{\mathbf{y}_1, \ldots, \mathbf{y}_\ell\}$ are given ; here $\mathbf{y}_i \in \{-1, +1\}^K$ where K is the number of the labels used to annotate ℓ training points. In many real-world applications only a few data is labeled (i.e., $\ell \ll m$) and its distribution may be different from the unlabeled data. For brevity, the features of input data are represented as the matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ and their labels are represented as the matrix $\mathbf{Y} \in \mathbb{R}^{K \times m}$ where its first ℓ columns are labeled and the next $(m - \ell)$ columns are zero vectors ; $\mathbf{X}_i$ or $[\mathbf{X}]_i$ denotes the column i of $\mathbf{X}$ while $\mathbf{X}_{ij}$ is the entry of $\mathbf{X}$ at row i and column j. Additionally, $\|\cdot\|_F$ is the Frobenius norm, $\mathrm{tr}(\cdot)$ is the trace operator, $\mathbf{X}'$ is the transpose of $\mathbf{X}$, and $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix whose diagonal is vector $\mathbf{v}$.

4.2.2 Multi-class Kernel Learning

Conventional max-margin classification models (for instance SVM [Cortes 1995]) aim to learn linear functions of the form $y = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$ in which $\phi(\cdot)$ maps nonlinear data $\{\mathbf{x}_i\} \in \mathcal{X}$ into linear ones $\{\mathbf{\Phi}_i\} \in \mathcal{H}$ and $\mathbf{w} \in \mathcal{H}$ is the normal vector of the optimal hyperplane separating the training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$ via max-

imizing the margin $2/\|\mathbf{w}\|$; here $\mathcal{H}$ is some Reproducing Kernel Hilbert Space (for instance [Scholkopf 2001b]) equipped with a dot-product $\langle\cdot,\cdot\rangle$. For kernel methods [Shawe-Taylor 2004, Scholkopf 2001b], $\phi(\cdot)$ and $\mathcal{H}$ are not explicitly known ; thus the dimensionality of $\mathcal{H}$ (and the VC-dimension [Vapnik 1998b]) may be infinite, which is bad for generalization performance. Transductive kernel learning (TKL) learns finite-dimensional kernel maps $\{\mathbf{\Phi}_i\} \in \mathcal{H} \subset \mathbb{R}^{p\times m}$ and a basis $\mathbf{B} \in \mathbb{R}^{n\times p}$, which are the elements of the factorization $\mathbf{B\Phi} = \mathbf{X}$, altogether with a classifier $\mathbf{w}$ that guarantees max-margin property. By sharing $\mathbf{\Phi}$ to K classifiers, the multi-class TKL formula can be written as follows

$$\begin{array}{ll} \min\limits_{\mathbf{W},\mathbf{\Phi},\mathbf{B}} & \frac{\alpha}{2}\left\|\mathbf{B\Phi}-\mathbf{X}\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{\Phi}\right\|_F^2 + \frac{1}{2}\left\|\mathbf{W}\right\|_F^2 \\ \text{s.t} & \mathbf{W}'\mathbf{\Phi C} = \mathbf{Y} \\ & \|\mathbf{B}_j\|_2^2 = 1, j = 1,\ldots,p \end{array} , \tag{4.1}$$

where the Frobenius norms of $\mathbf{\Phi}$ and $\mathbf{W}$ control the dimensionality of the kernel map and the complexity of classifiers respectively ; setting μ small enough guarantees both dimensionality finiteness of $\mathbf{\Phi}$ and data fidelity of ℓ equality constraints in (4.1) ; the mask matrix $\mathbf{C} \in \mathbb{R}^{m\times m}$ is diagonal in which $\mathbf{C}_{ii} = 1_{\{1\le i\le l\}}$. The next p equality constraints force the basis to have unit magnitude in order to prevent $\mathbf{\Phi}$ and $\mathbf{B}$ from growing to infinite. The upper bound of p must be at least $\max(n,m)+1$ so that the training data can be shattered [Vapnik 1998b, Scholkopf 2001b].

A new formula is obtained by replacing these ℓ first equality constraints by a squared loss term, i.e.,

$$\begin{array}{ll} \min\limits_{\mathbf{W},\mathbf{\Phi},\mathbf{B}} & \frac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \frac{1}{2}\|\mathbf{W}\|_F^2 + \frac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{K\times p} & \mathbf{W}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2 , \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \end{array} \tag{4.2}$$

in which $\mathbf{0}_{K\times p}$ and $\mathbf{0}_{n\times p}$ are zero matrices of sizes $K\times p$ and $n\times p$ respectively.

4.2.2.1 Smoothness Constraint

For a better conditioning of (4.2) a smoothness term is introduced into the equation. This term makes it possible to design a smooth kernel map and to assign similar predictions to neighboring data. Similar to the construction of transductive kernel map in the previous chapter, we model the input data $\mathcal{S}$ using an adjacency graph $\{\mathcal{V},\mathcal{E}\}$ where nodes $\mathcal{V} = \{v_1,\ldots,v_m\}$ correspond to samples $\{\mathbf{x}_i\}$ and edges $\mathcal{E} = \{e_{ij}\}$ are the set of weighted links of that graph. Considering $\mathbf{y}_i = \mathbf{W}'\mathbf{\Phi}_i$ and $\mathbf{y}_j = \mathbf{W}'\mathbf{\Phi}_j$, we define our regularizer as

$$\frac{\beta}{4}\sum_{i=1}^{m}\sum_{j=1}^{m}\|\mathbf{W}'\mathbf{\Phi}_i - \mathbf{W}'\mathbf{\Phi}_j\|^2\mathbf{A}_{ij} = \frac{\beta}{2}\mathrm{tr}\left(\mathbf{W}'\mathbf{\Phi L\Phi}'\mathbf{W}\right) \tag{4.3}$$

here $\beta \ge 0$ and $\mathbf{L}$ is the graph Laplacian defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_m)$ where $\mathbf{1}_m$ is the all-ones vector of length m.

4.2.2.2 Modeling Label Dependency

Our idea is based on modeling the dependency between labels in an image : if the label c is likely to occur in an image and it highly correlates with the label c' (according to some training data), then c' is also likely to be present in that image. Let us denote $p(c|c')$ as the probability of the occurrence of c given that of c', minimizing the following term encourages labels c and c' to co-occur :

$$\frac{1}{2}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left\|\mathbf{W}'_c\mathbf{\Phi}-\mathbf{W}'_{c'}\mathbf{\Phi}\right\|^2\mathbf{P}_{cc'}=\mathrm{tr}\left(\mathbf{\Phi}'\mathbf{W}\mathbf{Q}\mathbf{W}'\mathbf{\Phi}\right). \tag{4.4}$$

In the above equation, $\mathbf{P}_{cc'}=p(c|c')$ and $\mathbf{Q}=\mathrm{diag}(\mathbf{P}\mathbf{1}_K)-\mathbf{P}$ and $\mathbf{1}_K$ is the all-one vector of size K. Given a training database, the conditional probability $p(c|c')$ is computed as the ratio between the number of images annotated with both labels c, c' and the number of images annotated with c' :

$$p(c|c')=\frac{\sum_{i=1}^{\ell}1_{\{\mathbf{Y}_{ci}\}}1_{\{\mathbf{Y}_{c'i}\}}}{\sum_{i=1}^{\ell}\sum_{c''=1}^{K}1_{\{\mathbf{Y}_{c''i}\}}1_{\{\mathbf{Y}_{c'i}\}}} \tag{4.5}$$

in which the training label $1_{\{\mathbf{Y}_{ci}\}}=1$ if $\mathbf{Y}_{ci}=1$ and $1_{\{\mathbf{Y}_{ci}\}}=0$ if $\mathbf{Y}_{ci}\neq 1$. Since we do not consider the case where $c=c'$, then $p(c|c)$ is undefined, thus $\mathbf{P}_{cc}=0$. In practice, we found that better performance is obtained if $\mathbf{P}$ is symmetric, i.e., $\mathbf{P}_{cc'}=\frac{1}{2}\left(p(c|c')+p(c'|c)\right)$.

4.3 Optimization

Combining (4.2), (4.3) and (4.4) we obtain the complete form of our transductive learning problem

$$\begin{aligned}\min_{\mathbf{W},\mathbf{\Phi},\mathbf{B}}\quad & \tfrac{1}{2}\mathrm{tr}\left(\mathbf{\Phi}'\left(\mu\mathbf{I}+\gamma\mathbf{W}\mathbf{Q}\mathbf{W}'\right)\mathbf{\Phi}\right)+\tfrac{1}{2}\mathrm{tr}\left(\mathbf{W}'\left(\mathbf{I}+\beta\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\right)\mathbf{W}\right)+\\ & +\tfrac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix}-\begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{K\times p} & \mathbf{W}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi}\mathbf{C}\end{pmatrix}\right\|_F^2,\\ \text{s.t}\quad & \|\mathbf{B}_i\|_2^2=1,\ \forall i=1,\ldots,p\end{aligned} \tag{4.6}$$

where β and γ controls how much the smoothness and the dependencies between labels are embedded affect the learning outcome. Similarly to the precedent chapter, an EM-like optimization algorithm is used to solve (4.6). Since the change from binary to multi-class setting just involves the classifier $\mathbf{W}$ and the kernel map $\mathbf{\Phi}$, the update rule (3.7) of the basis $\mathbf{B}$ is reused.

4.3.1 Updating Classifier and Basis

Since classifiers $\mathbf{W}_c$'s depend on each other, an iterative optimization procedure is required for every vector $\mathbf{W}_c$ to converge to a stationary solution. Assuming fixed

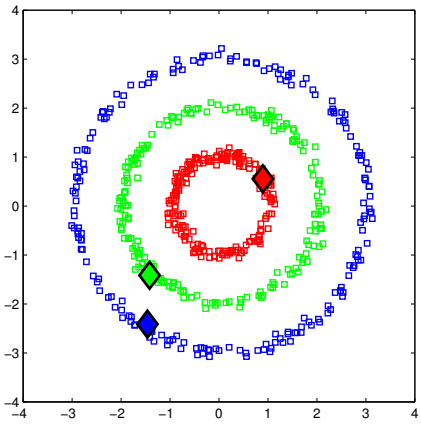


FIGURE 4.1 – The toy data for multi-class problem ; shown in red square dots are label vectors $[1,-1]'$, blue square dots are $[-1,1]'$, and green square dots are $[1,1]'$. There is just one labeled sample for every class, i.e. diamond dots, and the others are unlabeled.

$\mathbf{\Phi}^{(t)}$ (denoted simply as $\mathbf{\Phi}$) and enforcing the gradient of (4.6) to vanish (with respect to $\mathbf{W}$) leads to $\mathbf{W}^{(t)} = \tilde{\mathbf{V}}$ with $\tilde{\mathbf{V}} = \lim_{\varsigma \to \varsigma_{\max}} \mathbf{V}^{(\varsigma)}$ and

$$\mathbf{V}_c^{(\varsigma)} = \left(\mathbf{I} + \mathbf{\Phi}\left(\alpha\mathbf{C} + \beta\mathbf{L} + \gamma\mathbf{M}_{cc}\mathbf{I}\right)\mathbf{\Phi}'\right)^{-1} \cdot \left[\alpha\mathbf{\Phi}\mathbf{C}\mathbf{Y}' + \gamma\mathbf{\Phi}\mathbf{\Phi}'\mathbf{V}^{(\varsigma-1)}\mathbf{P}\right]_c \tag{4.7}$$

In order to find $\mathbf{B}$, let us assume that $\mathbf{\Phi}^{(t)}$ and $\mathbf{W}^{(t)}$ fixed (defined simply as $\mathbf{\Phi}, \mathbf{W}$), then the following optimization problem

$$\min_{\mathbf{B}} \quad \tfrac{1}{2}\left\|\mathbf{X} - \mathbf{B}\mathbf{\Phi}\right\|_F^2 \quad \text{s.t} \quad \left\|\mathbf{B}_i\right\|_2^2 = 1, i = 1, \ldots, p. \tag{4.8}$$

is solved similarly as presented in Section 3.3.1.

4.3.2 Updating Kernel Map

Considering fixed $\mathbf{B}^{(t+1)}$ and $\mathbf{W}^{(t+1)}$ (denoted simply as $\mathbf{B}$, $\mathbf{W}$ in the remainder of this section), and the previous kernel map solution $\mathbf{\Phi}^{(t)}$, our goal is to find $\mathbf{\Phi}^{(t+1)}$ by solving (4.6). The optimization problem (4.6) admits a unique solution $\mathbf{\Phi}^{(t+1)} = \lim_{\tau \to \tau_{\max}} \mathbf{\Psi}^{(\tau)}$ and

$$\begin{aligned}\mathbf{\Psi}_i^{(\tau)} = &\left(\mu\mathbf{I} + \alpha\mathbf{B}'\mathbf{B} + \mathbf{W}\left(\alpha\mathbf{C}_{ii}\mathbf{I} + \beta\mathbf{D}_{ii}\mathbf{I} + \gamma\mathbf{Q}\right)\mathbf{W}'\right)^{-1} \cdot \\ &\left[\alpha\left(\mathbf{B}'\mathbf{X} + \mathbf{W}\mathbf{Y}\mathbf{C}\right) + \beta\mathbf{W}\mathbf{W}'\mathbf{\Psi}^{(\tau-1)}\mathbf{A}\right]_i\end{aligned} \tag{4.9}$$

Proof about this kernel map solution and its convergence to a fixed point are similar to the Proposition 3.3.1. The process (4.9) allows us to recursively diffuse the kernel maps from the labeled to the unlabeled data, through the neighborhood system defined in the graph $\{\mathcal{V}, \mathcal{E}\}$. The algorithm terminates when either $\|\mathbf{\Psi}^{(\tau)} - \mathbf{\Psi}^{(\tau-1)}\| \leq$

ε or the iterative optimization algorithm reaches $t_{\max}$ iterations. The optimization algorithm is summarized in Alg. 2.

Shown in Fig. 4.2 is the experiment results of applying the proposed method to the toy dataset in Fig. 4.1. The dataset consists of two labels. In this example the parameter $\gamma = 0$ since the label-dependency is not modeled by the toy data.

Algorithm 2 Multi-class kernel map learning

Input : labeled $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$ and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{m}$
Initialization : compute the adjacency matrix $\mathbf{A}$, degree matrix $\mathbf{D}$, graph Laplacian $\mathbf{L}$, $t \leftarrow 0$ and set $\mathbf{\Phi}^{(0)}$ to a random full rank matrix.
Repeat steps (1+2) **until** convergence OR $t > t_{\max}$

1. Update $\mathbf{W}^{(t+1)}$ by taking the limit $\tilde{\mathbf{V}}$ of (4.7), with $\mathbf{V}^{(0)} = \mathbf{W}^{(t-1)}$.
2. Update $\mathbf{B}^{(t+1)}$ using (4.8).
3. Update $\mathbf{\Phi}^{(t+1)}$ by taking the limit $\tilde{\mathbf{\Psi}}$ of (4.9), with $\mathbf{\Psi}^{(0)} = \mathbf{\Phi}^{(t)}$.

Output : kernel maps $\{\mathbf{\Phi}_i^{(t+1)}\}_{i=\ell+1}^{m}$ and labels $\{\mathbf{y}_i\}_{i=\ell+1}^{m}$ with $\mathbf{y}_i = \left(\mathbf{W}^{(t+1)}\right)' \mathbf{\Phi}_i^{(t+1)}$.

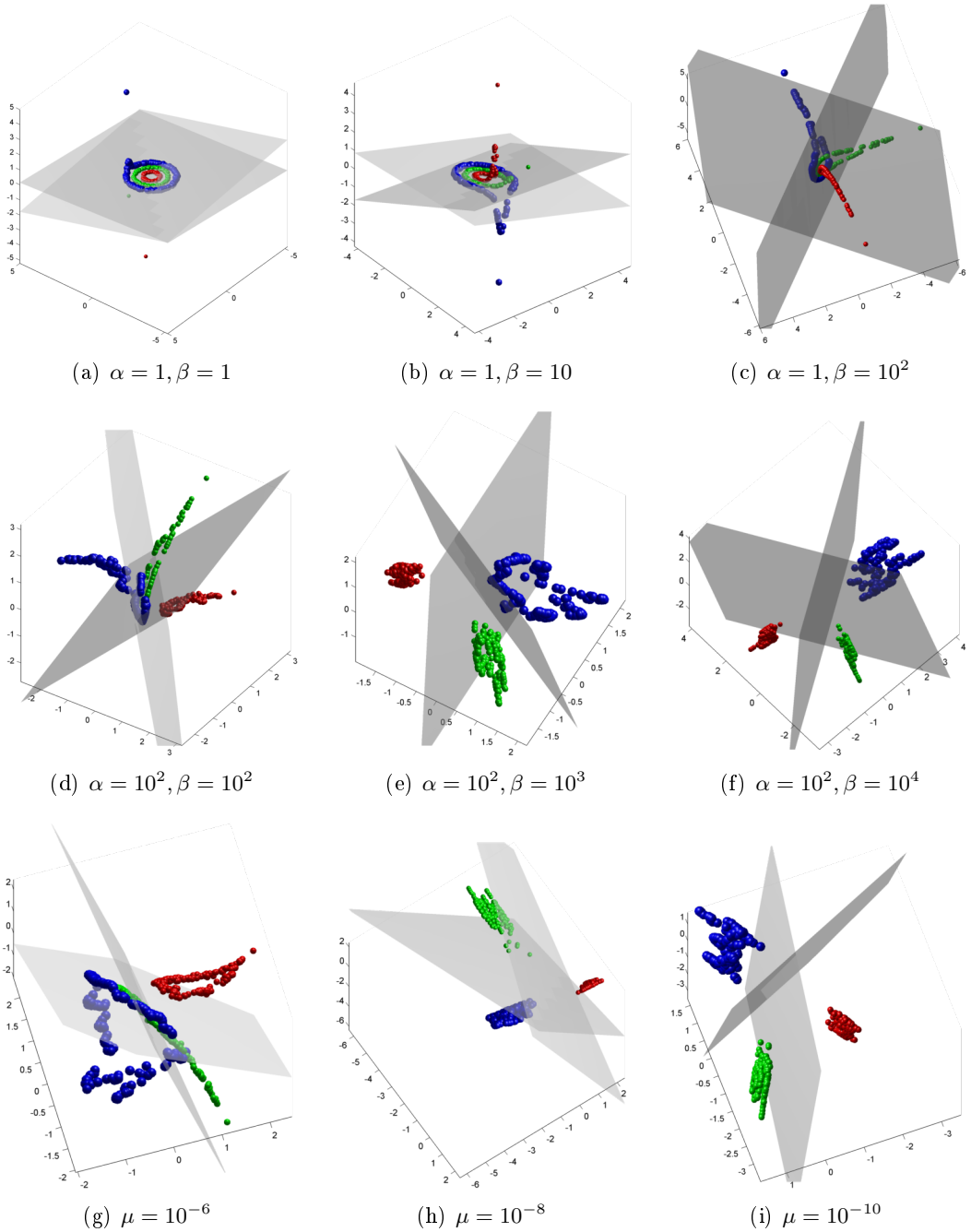
4.4 Experiments

In the remainder of this section we apply our method to the problem of multi-class image annotation using two standard datasets MSRC and Corel5K. The MSRC dataset includes 591 images from 23 categories mixing man-made and natural objects ; the class "horse" is omitted from the evaluation set as it appears only 2 times. Note that the MSRC set was originally used for segmentation, so we adapt it to image annotation by considering label information in every image and ignoring other available information. As in [Liu 2010], the dataset is randomly split into two equal subsets for training and testing. The Corel5K dataset contains 5000 images manually annotated with 260 labels (each image has at least one label and may include up to 5 labels). This dataset is standard and widely used in the related image annotation work ; it includes 4500 images for training and 500 images for testing. The test set was built in order to guarantee that every label is used at least once.

4.4.1 Features and Graph Construction

Every image is divided into blocks using three grids of size 1×1, 2×2, and 1×3 ; every block is represented by a bag-of-word histogram based on 512 visual words. The latter result from the quantization of densely sampled SIFT descriptors extracted from training images. We use six variants of SIFT descriptors [van de Sande 2010] in order to obtain better visual discrimination : SIFT, rgbSIFT, rgSIFT, hsvSIFT, cSIFT, and opponentSIFT. Every image is described with a super-descriptor corresponding to histograms of 8 blocks ; this super-descriptor is normalized in order to guarantee that its L_2-norm is equal to 1.

Once vectorial representations of images are computed, a k-nearest neighbor graph $\{\mathcal{V}, \mathcal{E}\}$ is constructed in which the vertex set $\mathcal{V}$ consists of images and the



(a) $\alpha = 1, \beta = 1$ (b) $\alpha = 1, \beta = 10$ (c) $\alpha = 1, \beta = 10^2$

(d) $\alpha = 10^2, \beta = 10^2$ (e) $\alpha = 10^2, \beta = 10^3$ (f) $\alpha = 10^2, \beta = 10^4$

(g) $\mu = 10^{-6}$ (h) $\mu = 10^{-8}$ (i) $\mu = 10^{-10}$

FIGURE 4.2 – The visualizations of learned kernel maps with different configurations ; the input data of this problem is depicted in Fig. 4.1. (a-c) : keeping α fixed and slowly increasing β, test data are pulled more to their nearest labeled data. (d-f) : Increasing α leads to small values of $\mathbf{\Phi}$ (note the coordinate values on the axes). (g-i) : Decreasing the value of rank regularizer yields clearer separation of the data. With a large value of μ, the data tend to keep their geometrical shape as of the input space.

edge set $\mathcal{E}$ consists of links between similar images. Image similarity between two arbitrary images of indices a and b is computed as

$$\mathbf{A}_{ab} = \exp\left(-\frac{1}{2\sigma^2}\left(\sum_{q=1}^{Q}\frac{D(\mathbf{x}_a^q,\mathbf{x}_b^q)}{Z_q}\right)^2\right) \tag{4.10}$$

where $D(\cdot,\cdot)$ measures the dissimilarity between two feature vectors $\mathbf{x}_a^q$ and $\mathbf{x}_b^q$ of feature type q ; in our experiments $Q = 6$ and $D(\cdot,\cdot)$ is defined based on histogram intersection distance, i.e., $D(\mathbf{x}_a^q,\mathbf{x}_b^q) = \sum_{i=1}^{n}(1 - \min([\mathbf{x}_a^q]_i,[\mathbf{x}_b^q]_i))$. Since every feature type has its own value range, we normalize the dissimilarity scores by the factor Z_q in which $Z_q = \sum_{ab} D(\mathbf{x}_a^q,\mathbf{x}_b^q)/m^2$. If we denote $\overline{D}_{ab} = \sum_{q=1}^{Q}\left(D(\mathbf{x}_a^q,\mathbf{x}_b^q)/Z_q\right)$, the bandwidth σ is empirically estimated as $\sigma = \sum_{\{a,b\}\in\mathcal{E}}\overline{D}_{ab}/(km)$ in which $\{a,b\}\in\mathcal{E}$ denotes all the image pairs $\{a,b\}$ where there exists a graph link between images a and b. Notice that every image is connected to its k most similar images which are taken from m images ($k \ll m$) of the database ; if a is disconnected from b, then $\mathbf{A}_{ab} = 0$. The resulting affinity matrix $\mathbf{A}$ is symetrized by taking $\mathbf{A} \leftarrow (\mathbf{A} + \mathbf{A}')/2$ instead of $\mathbf{A}$.

4.4.2 Evaluation Measures

Different evaluation criteria are used in order to measure the quality of this annotation process including precision (denoted **P**), recall (denoted **R**) and positive recall (denoted **N+**) ; these criteria are defined as

$$\begin{aligned}
\mathbf{P} &= \mathbb{E}_\omega\left(\frac{\text{number of images correctly annotated with a label } \omega}{\text{number of images annotated with } \omega}\right)\\
\mathbf{R} &= \mathbb{E}_\omega\left(\frac{\text{number of images correctly annotated with a label } \omega}{\text{number of images annotated with } \omega \text{ in the ground truth}}\right)\\
\mathbf{N+} &= \textstyle\sum_\omega 1_{\{(\text{number of images correctly annotated with a label } \omega)\,\geq\, 1\}},
\end{aligned}$$

here the expectation $\mathbb{E}_\omega$ is with respect to all possible labels $\{\omega\}$ in our dataset. We further benchmark the quality of label assignment using break-even point (denoted **BEP** [Grangier 2008]), with

$$\mathbf{BEP} = \mathbb{E}_\omega\left(\frac{\text{number of images correctly annotated with a label } \omega \text{ in a sorted list of } N_\omega \text{ images}}{N_\omega}\right)$$

here N_ω is the number of images annotated with ω in the ground truth and the list of N_ω images is sorted by decreasing classification scores. By varying the size of the sorted lists and taking the expectation of precision, with respect to this size, we obtain the mean average precision (denoted **mAP**).

4.4.3 Results and Discussion

Given the training and the test sets, we define our neighborhood system using an adjacency graph where each node corresponds to an image and an edge connects

two images if they are visually similar. Using this neighborhood system, we run our optimization procedure in order to measure the membership of a given test image to different classes ; these memberships correspond to the scores of different classifiers. A label is then assigned to a test image iff its classifier score is among the 5 largest values.

In what follows we denote TKL and wTKL for results produced by (4.6) with $\gamma = 0$ and $\gamma > 0$ respectively. In all these experiments the size of the neighborhood is fixed at $k = 3$. We use the normalized graph Laplacian $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$. The setting of i) the data fitting coefficient is $\alpha = 1$, ii) the smoothness coefficient is $\beta = 1$, iii) the low-rank coefficient is $\mu = 10^{-8}$, $\varepsilon = 10^{-2}$, iv) the maximum number of iterations for (4.7) and (4.9) to converge are $\varsigma_{\max} = 5$ and $\tau_{\max} = 20$ respectively, v) the maximum number of iterations for (4.6) to derive its final solution $t_{\max} = 5$.

4.4.3.1 Does Label-Dependency Help ?

Empirical results demonstrate that modeling dependencies between labels contributes positively in retrieving more rare keywords from the Corel5K dataset. From Table. 4.2 we can observe that this model increases the number of recalled keywords $N+$ from 140 to 165. As a consequence, the average recall rate is significantly improved at the detriment of a small degradation of the precision. Fig. 4.3 describes in details how the label-dependency model affects the annotation performance. According to this figure, a significant improvement of the recall from 35% to 42% when γ is increased from 10^{-2} to 1 with a slight decrease of the precision from 28% to 26% and also BEP. For example, more correct keywords are found in examples of Fig. 4.4(a) and Fig. 4.4(b) due to the use of the label-dependency model. However, continuing to increase γ do not enhance either recall or mAP ; instead, it reduces the precision, i.e., from 26% to 23% when γ is increased from 1 to 10 or 100. In Fig. 4.4(c), the two incorrect labellings *kauai* and *train* – made by the basic formula TKL – provide two other incorrect labels *locomotive* and *railroad*, which are highly correlated to *train* and *kauai*, thus making the labeling wrong.

We also tested the behavior of label-dependency model in the MSRC dataset ; however, its effect is not as clear as in the Corel5K dataset. This is because the number of labels in MSRC dataset is so small and occurrence frequencies of labels are approximately equal. As a result, label-dependency may be useful if the dataset consists of many labels and those labels are highly imbalanced, or some of them are difficult to recognize due to either their less discriminative appearances or their non-visual meaning. More annotation examples are shown in Fig. 4.7.

Remarks Label-dependency modeling is highly related to the problem of imbalanced databases. This is an inherent problem of learning problems where the frequencies of classes – according to the input dataset – are unequal. [He 2009] investigates the nature of imbalance and states that its causes can be intrinsic (for example, rare object instances) or extrinsic (for example, biased acquisition due to human factor [Torralba 2011]). From machine learning point of view, data imbal-

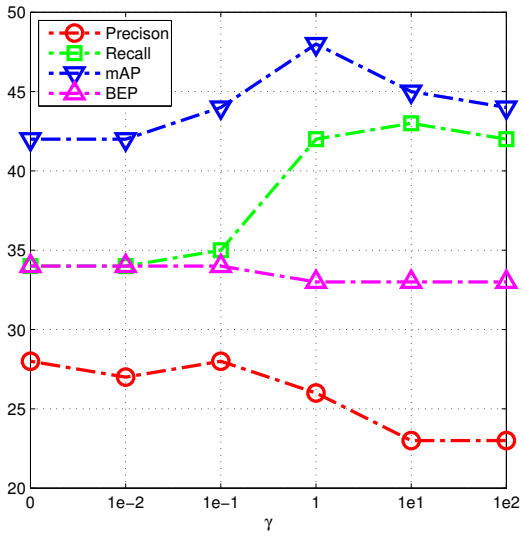


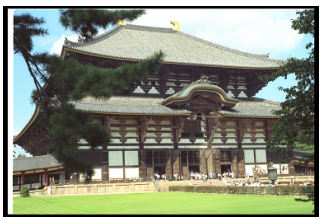
FIGURE 4.3 – The evolution of the evaluation measures with respect to the label-dependency parameter γ. As γ is increased from zero to one, precision tends to decrease slightly while recall increases from 35 % at $\gamma = 10^{-1}$ to 42 % at $\gamma = 1$. The label-dependency term allows our algorithm to retrieve more keywords (i.e the $N+$ measure in Table 4.3) which are the main factors that increase the average recall.

ance needs to be avoided, especially for nonparametric methods (such as k-Nearest Neighbors based methods [Makadia 2008] and SVM [Cortes 1995]) where a part of input data are kept for future prediction. For our particular case, the smoothness term in the TKL formula is affected by imbalanced training data. According to a survey [He 2009], popular rebalancing techniques include data re-sampling methods and cost-sensitive learning algorithms (for example SVM with imbalance class weights [Chang 2011]). We argue that such techniques are systematic since they neglect underlying causes of imbalance and do not consider relationships between data which may help. As shown above, the label dependency modeling can remedy such imbalance cases.

4.4.3.2 Comparison with Related works

Inductive methods. We consider three state-of-the-art methods : (i) standard SVM classifier [Vapnik 1998a] with 4 kernel choices (linear, RBF, χ^2, Histogram Intersection) ; (ii) Multiple Kernel Learning SVM (MKL) implemented by the SMO algorithm [Vishwanathan 2010a] ; (iii) SVM for multi-class classification implemented by M3L [Hariharan 2010]. Parameters of each method are optimally tuned using k-fold validation on the training data. For MKL, we use SMO solver with L_2 regularization. Note that MKL is extensively trained using 36 Gram matrices taken from the combination between the 6 kernels (linear, χ^2, Histogram Intersection, and RBF with 3 scales values) and the 6 visual descriptors listed in Section 4.4.1. We

(a) **Before** : ocean, coral, reefs, people, water ; **After** : coral, ocean, reefs, *fish*, fan.



(b) **Before** : tree, water, buildings, sky, mountain ; **After** : tree, roofs, water, arch, *temple*.

(c) **Before** : tree, sky, sunset, kauai, train ; **After** : kauai, locomotive, train, railroad, sunset.

FIGURE 4.4 – Examples of the Corel5K dataset that demonstrate how label-dependency could be used to improve the annotation task. Every example shows the annotation results before and after adding label-dependency ; underlined keywords are the correct labels while italic keywords are the correct ones discovered due to the label-dependency model. In Fig. 4.4(a), the occurrence of *coral*, *ocean*, and *reefs* imply a high probability that *fish* is also present in the scene. In Fig. 4.4(b), the co-occurrence of labels such as *roofs* and *arch* leads to the presence of *temple*. In Fig. 4.4(c), the occurrence of the two false labels *kauai* and *train* promotes the presence of other false labels *locomotive*, *train*, and *railroad* too.

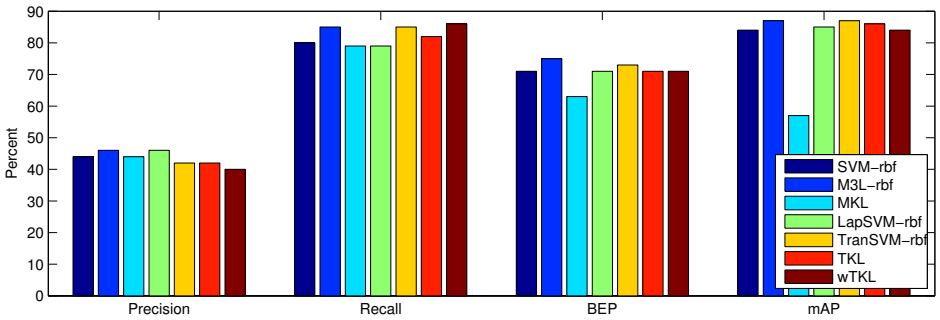


FIGURE 4.5 – The comparative results of the MSRC dataset between our methods (TKL and wTKL) and the best configurations of the related works. See Table 4.1 for numerical values.

use libSVM [1] as the standard implementation for SVM while the implementations of M3L and MKL are taken from the websites of their original authors.

Transductive Methods. LapSVM [Melacci 2011] and TranSVM [Joachims 1999] are taken into account for comparison. LapSVM is more related to our method since both include the smoothness regularization term. The implementation of LapSVM is taken from [Melacci 2011] and that of TranSVM from SVM^{light} [2].

1. http ://www.csie.ntu.edu.tw/ cjlin/libsvm/
2. http ://svmlight.joachims.org/

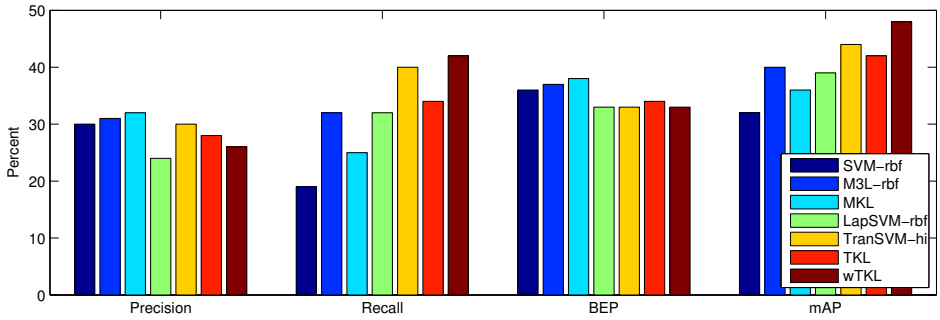


FIGURE 4.6 – The comparative results of the Corel5K dataset between our methods (TKL and wTKL) and the best configurations of the related works. See Table 4.2 for numerical values

Table 4.1 and plots in Fig. 4.5 show results and comparison on the MSRC dataset. A first conclusion indicates that methods relying on both labeled and unlabeled data provide better performance. Our method and other transductive methods LapSVM and TranSVM perform slightly better than inductive methods such as SVM and MKL ; however, the inductive M3L method is the most performant one. It turns out that the MSRC dataset is not challenging enough in order to see a clear advantage of transductive over inductive methods.

Differences between the two approaches become clearer on the Corel5K database (see Table 4.2 and Fig. 4.6). As expected the inductive methods perform worse than the transductive ones. In particular, SVM is the worst (with low recall and the number $N+$ of keywords whose recalls are positive is low too). MKL is better than SVMs ; however, it is not better than M3L and transductive methods including ours. This is easy to understand because M3L has the training loss specifically designed for multi-class problems and the prior constraint on label correlation as well. Among transductive methods, LapSVM performs worst while TranSVM slightly outperforms the standard version of our method. Nevertheless, our method with label-dependency is comparable with the best configuration of TranSVM, the one that uses histogram intersection kernel.

Finally, we compare annotation performance of our method against evaluations reported in some related works (see Table 4.3). In general our method performs better than recent works such as JEC [Makadia 2008] and GS (group sparse coding) [Zhang 2010] and the graph-based method MSC [Wang 2009, Liu 2010]. For the state-of-the-art TagProp [Guillaumin 2009], our method is competitive : it has similar recall, better at $N+$ and mAP, but slightly worse precision and BEP.

4.5 Summary

In this chapter we introduced the multi-class extension of our transductive kernel learning algorithm and demonstrated its use with the image annotation application. At the end of this chapter we accomplished two objectives. First, we proposed a

	SVM				M3L				MKL
	lin	**rbf**	**hi**	χ^2	**lin**	**rbf**	**hi**	χ^2	
P	43	44	44	44	46	46	46	46	44
R	78	80	81	80	85	85	83	85	79
N+	22	22	22	22	22	22	22	22	22
mAP	54	84	56	84	74	87	85	85	57
BEP	60	71	64	69	87	75	72	73	63

	LapSVM				TranSVM				TKL	
	linear	**rbf**	**hi**	χ^2	**linear**	**rbf**	**hi**	χ^2	std	w
P	41	46	43	41	42	42	40	40	42	40
R	77	79	78	77	85	85	83	85	82	86
N+	22	22	22	22	22	22	22	22	22	22
mAP	88	85	86	87	73	87	87	87	86	84
BEP	71	71	74	70	72	73	74	73	71	71

TABLE 4.1 – Comparative results of the MRSC dataset between our method and the related works : SVM [Vapnik 1998a], M3L [Hariharan 2010], MKL [Vishwanathan 2010a], Laplacian SVM [Melacci 2011], and Transductive SVM [Joachims 1999]. Four kernel functions (linear, Gaussian RBF, Histogram Intersection, χ^2) are tested for the methods SVM, M3L, LapSVM and TranSVM. The MKL-SVM is trained based on 36 combinations of six kernels (the four kernels mentioned above in which the RBF kernel has respectively 3 different bandwidths) and six visual descriptors.

more efficient way in applying transductive kernel learning algorithm into multi-class problems. Second, we proposed the label-dependency model which exploits prior information of the image database in order to improve the recall. Evaluations on the MSRC and Corel5k datasets show that our method is competitive with related works which are specifically designed for the image annotation task. The multi-class transductive kernel learning is also extended for image interpretation problem in the subsequent chapter. For the image annotation problem itself, our plan is to extend to images on social networks and consider auxiliary information of images such as tags, user profiles, and geographical locations in order to improve annotation quality.

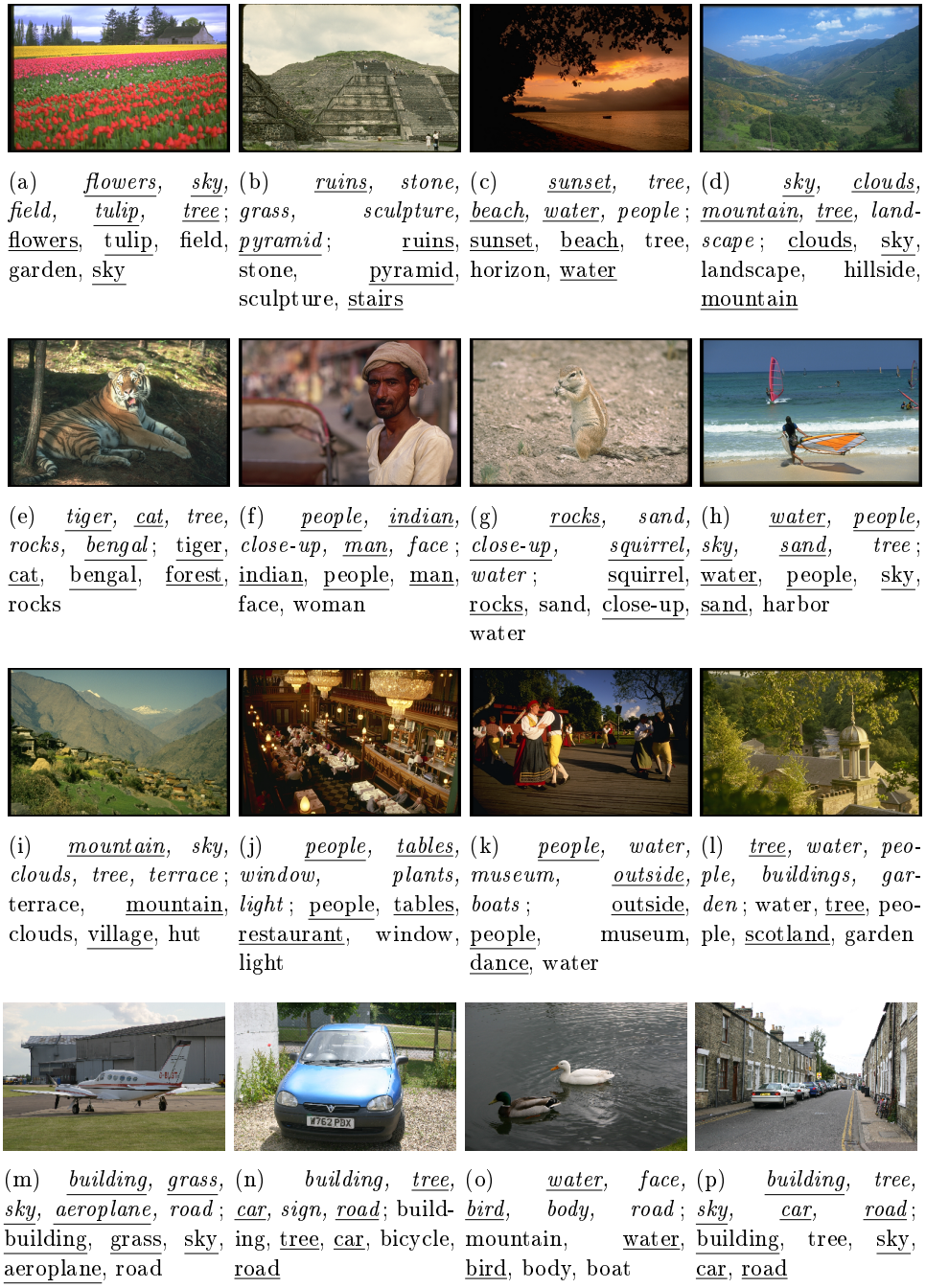
	SVM				M3L				MKL
	lin	**rbf**	**hi**	χ^2	**lin**	**rbf**	**hi**	χ^2	
P	24	30	28	26	28	31	31	32	32
R	15	19	18	17	32	32	30	31	25
N+	80	97	91	85	133	122	119	124	113
mAP	24	32	30	26	34	40	38	39	36
BEP	35	36	36	38	40	37	37	39	38

	LapSVM				TranSVM				TKL	
	linear	**rbf**	**hi**	χ^2	**linear**	**rbf**	**hi**	χ^2	std	w
P	25	24	24	24	26	28	30	30	28	26
R	32	32	30	32	35	37	40	39	34	42
N+	126	129	112	129	155	150	157	155	140	165
mAP	40	39	35	39	41	41	44	42	42	48
BEP	25	33	38	31	31	32	33	32	34	33

TABLE 4.2 – Comparative results of the Corel5K dataset between our method and the related works : SVM [Vapnik 1998a], M3L [Hariharan 2010], MKL [Vishwanathan 2010a], Laplacian SVM [Melacci 2011], and Transductive SVM [Joachims 1999]. Four kernel functions (linear, Gaussian RBF, Histogram Intersection, χ^2) are tested for the methods SVM, M3L, LapSVM and TranSVM. The MKL-SVM is trained based on 36 combinations of six kernels (the four kernels mentioned above in which the RBF kernel has respectively 3 different bandwidths) and six visual descriptors.

	Precision	Recall	N	mAP	BEP
CRM [Lavrenko 2003]	16	19	107	-	-
InfNet [Metzler 2004]	17	24	112	-	-
NPDE [Jeon 2003]	18	21	114	-	-
MBRM [Feng 2004]	24	25	122	-	-
SML [Carneiro 2007]	23	29	137	-	-
TGLM [Liu 2009a]	25	29	131	-	-
MEG [Liu 2010]	25	31	-	-	-
MSC [Wang 2009]	25	32	136	42	-
JEC [Makadia 2008]	27	32	139	-	-
GS [Zhang 2010]	30	33	146	-	-
TagProp [Guillaumin 2009]	33	42	160	42	36
PAMIR [Grangier 2008]	-	-	-	26	17
TKL	28	34	140	42	34
wTKL	26	42	165	48	33

TABLE 4.3 – Overview of performance of our proposed method and some previous works in annotating 500 test images of Corel5K dataset.



(a) *flowers*, *sky*, *field*, *tulip*, *tree*; flowers, tulip, field, garden, sky

(b) *ruins*, *stone*, *grass*, *sculpture*, *pyramid*; ruins, stone, pyramid, sculpture, stairs

(c) *sunset*, *tree*, *beach*, *water*, *people*; sunset, beach, tree, horizon, water

(d) *sky*, *clouds*, *mountain*, *tree*, *landscape*; clouds, sky, landscape, hillside, mountain

(e) *tiger*, *cat*, *tree*, *rocks*, *bengal*; tiger, cat, bengal, forest, rocks

(f) *people*, *indian*, *close-up*, *man*, *face*; indian, people, man, face, woman

(g) *rocks*, *sand*, *close-up*, *squirrel*, *water*; squirrel, rocks, sand, close-up, water

(h) *water*, *people*, *sky*, *sand*, *tree*; water, people, sky, sand, harbor

(i) *mountain*, *sky*, *clouds*, *tree*, *terrace*; terrace, mountain, clouds, village, hut

(j) *people*, *tables*, *window*, *plants*, *light*; people, tables, restaurant, window, light

(k) *people*, *water*, *museum*, *outside*, *boats*; outside, people, museum, dance, water

(l) *tree*, *water*, *people*, *buildings*, *garden*; water, tree, people, scotland, garden

(m) *building*, *grass*, *sky*, *aeroplane*, *road*; building, grass, sky, aeroplane, road

(n) *building*, *tree*, *car*, *sign*, *road*; building, tree, car, bicycle, road

(o) *water*, *face*, *bird*, *body*, *road*; mountain, water, bird, body, boat

(p) *building*, *tree*, *sky*, *car*, *road*; building, tree, sky, car, road

FIGURE 4.7 – Some annotation examples of our method with the Corel5K dataset (the three first rows) and the MSRC dataset (the bottom row). Underlined keywords are true positive labels while non-underlined keywords are false alarms. Shown in italic texts are predicted labels of TKL while the upright texts are the labels of wTKL (with label dependency). The label dependency model exhibits its ability to recall labels in examples (b,e,i,j,k,l), which are either related to visual content of test images or due to prior relationships between labels. If not predicting the true labels, wTKL also provides labels closely related to semantic content of images, for example see (a,c,d,h). There is virtually no improvement made by wTKL on the MSRC dataset.

Chapitre 5

Contextual Kernel Learning for Scene Interpretation

In this chapter we develop our algorithm based on transductive kernel learning in order to solve the problem of scene interpretation. This problem amounts to automatically segment objects from a scene and then name them with labels. This is the general case of the interactive object segmentation problem introduced in Chapter 3. However, different from that task where human intervention is required in order to give hints for the classifier, the proposed method in this chapter makes classification automatic ; a database of fully labeled images is given as a source providing necessary training examples (a.k.a hints) for every test case. The novel contributions of this chapter are regularization designs that make the multi-label transductive kernel learning introduced in Chapter 4 adapted to the scene interpretation problem. Our contextual model incorporates prior knowledge of label co-occurrence statistics computed from training data into the kernel learning process. Empirical results with the SiftFlow dataset show that the proposed model improves labeling coherency and obtain competitive performance with the state of the art.

Parts of this work were mentioned in the following submission :

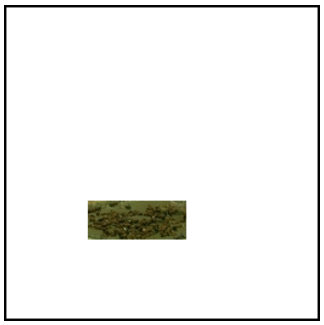
1. Phong Vo, Hichem Sahbi, *Contextual Kernel Map Learning for Scene Transduction*, ECCV, Switzerland, 2014.

5.1 Contextual Relationships in Scene Interpretation

Scene interpretation is the task of detecting, localizing and classifying visual objects in a given scene. If pixels of an input image are considered as data points, scene interpretation assigns labels to those pixels such that the resulting labeling forms a meaningful scene. In figuring out which pixels belong to which labels, visual descriptors are extracted and used as input features for a classification algorithm. These descriptors can be low-level (color histogram, textons, shape descriptors, Histogram of Gradient – HOG) or mid-level (Bag-Of-Word histograms of SIFT, texton, HOG). The discrimination power of those features allows the classifier to capture the variability of visual objects up to a certain degree but may fail in cases they are not discriminative enough.

Contextual features may help to disambiguate such complex cases ; those features are not derived from the appearance of objects itself but from their relations. For instance, Fig. 5.1 illustrates the usefulness of contextual information in order to infer the meaning of a scene. This example suggests us that scene comprehension is not as simple as recognizing a list of objects. Therefore, contextual relations are necessary for machine learning methods to exploit and improve the quality of scene interpretation.

In fact the use of contextual features in computer vision is related to the visual reasoning of an adult where contextual rules of the physical world have been learned during his infant period. According to findings in visual cognition and psychology [Biederman 1972, Bar 2004, Bar 2005, Cox 2004, Biederman 1982b], it is sufficient to induce scene coherency based on the following five rules : *support*, *interposition*, *probability*, *position*, and *familiar size*.



(a) An unidentified object.

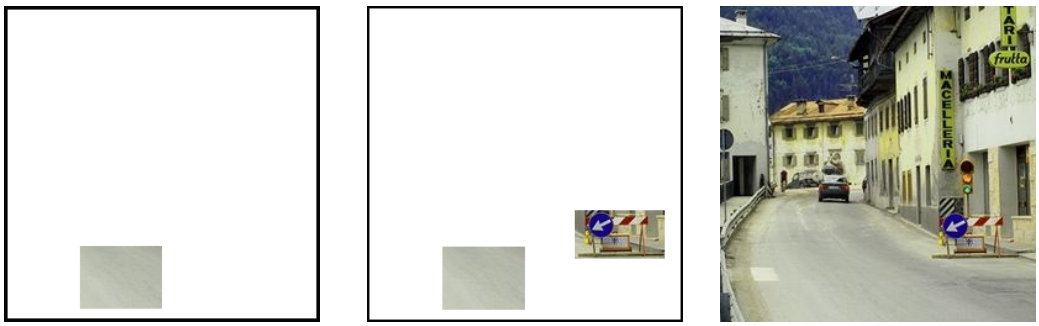
(b) The object is disambiguated with the help of its context.

FIGURE 5.1 – The left shows an image region of the right scene. By solely looking at this part we hardly recognize what are the tiny dark blobs. However, supplemental context provided by the right picture will figure out those blobs are cattle (cows, buffaloes, or something like that) because they are enclosed by a hedge-like object and there is also a farm house nearby.

Based on the definitions of these rules, we can further divide them into two groups [Biederman 1982b] : *physics* rules (*support* and *interposition*) and *semantic* rules (*probability*, *position*, and *size*). The first group – also called *syntactic* group – consists of physics rules of image formation ; the second group judges whether a scene is coherent based on the referential meaning of the objects. According to [Biederman 1982b], syntactic relations are more complex but less informative than semantic ones. In the following we discuss these semantic rules.

- **Probability.** It is the likelihood of spotting an object with respect to some scenes or some objects but not others [Wolf 2006]. In other word, it is the probability for an object to co-occur with other object or scene such as the co-occurrence between the cows and the field in Fig. 5.1 or between the street and the traffic sign in Fig. 5.2. The probability rules help disambiguating objects whose appearance not informative enough.
- **Position.** It is the probability for an object to be found at its familiar locations conditioned by the familiar locations of other objects. It is not difficult to find such examples : sky is always on the top of sea or sidewalks are on the side of streets. It is easy to realize that positional co-occurrence is conditioned by the semantics of the objects. In other word, the rule is not only associated to the spatial information but also the semantics of the objects. Thus the position rule is a probability rule with additional spatial constraints.
- **Scale.** This rule states that objects should admit a certain variation of size with respect to those of other objects in the scene. This rule exists as the relative sizes between objects in a scene. For instance, objects at close distances seem to be larger than the ones faraway. In order to use this rule, we must know a priori object identities as well as their extent in the image. In other word, the applicability of the scale rule is conditioned by the availability of the probability and the position rules. It also means that the scale rule is more difficult to use.

Discussions. In practice, the probability rule brings out the most valuable information for disambiguation. The reason is that scale and position relations may drastically vary from image to image while probability relations are quite invariant across scene images of a same category. The probability rule is also easy to compute ; one of the simplest realization of the probability rule is to compute the co-occurrence matrix between objects or regions. As a results, many of current scene interpretation algorithms, as we revise in the subsequent section, restrain themselves to the rules of probability and sometimes position rules are also considered. The scale rule can also be incorporated, but as we will see, it is rather encoded as feature descriptors than relations.



(a) What is this patch?

(b) The traffic sign suggests the identity of the texture-flatted patch.

(c) The original scene.

FIGURE 5.2 – Fig. 5.2 demonstrates another difficult case when the object appearance is homogeneous. The masked region in Fig. 5.2(a) does not give us any semantic meaning. However, with the co-occurrence of a sign near a barrier in Fig. 5.2(b), one can infer the gray patch as a part of the street. In this case, context helps improve the object discrimination.

5.2 Related Works

In this section we discuss major approaches that integrate semantic rules into machine learning models. Top-down approaches require an overall understanding of the whole scene. However, studies of top-down approaches just stop at designing visual descriptors for scene categorization [Oliva 2006]. For bottom-up approaches, image pixels can be aggregated in order to form more meaningful objects and parts ; the aggregation may continue in higher levels until a coherent scene is formed. A study in visual attention [Rutishauser 2004] has demonstrated that the bottom-up processing significantly improves object recognition in complex scenes. In this section, we revise bottom-up variants with different starting points of the “bottom" pixels, image regions, or objects.

5.2.1 Pixel-wise Interaction

This is the most basic contextual interaction in which neighboring pixels tend to get similar labels except at discontinuities [Shotton 2009, Wolf 2006, Liu 2011a]. Pixel-wise interaction does not need any preprocessing such as object segmentation or detection because the pixels themselves aggregate in order to form complete objects [Carbonetto 2004, Shotton 2009]. The limitation of this approach is that such interactions are short-range, thus they could not perform well on complex scenes where objects are highly cluttered. A wider interaction is necessary in order to cope with such complex cases.

5.2.2 Object Interaction

Given a list of object candidates detected from a scene, object interaction approaches relate those candidates by semantic rules in order to improve the contextual agreement between them, for instance the boosted classifiers [Fink 2003, Wolf 2006] and the network of logistic regression classifiers [Murphy 2003]. On the plus point, this approach is appropriate for scenes with rigid objects such as man-made objects of indoor scenes. However, this approach has problems with scalability. An object detector needs to be trained whenever a new object is considered. Moreover, object detection is time-consuming due to the exhaustive search of the detectors over the whole image at different scales.

5.2.3 Region-wise Interaction

This approach requires an input image to be over-segmented into regions (also called superpixels) such that the visual appearance within every region is quite homogeneous and bounded by discontinuities of object boundaries or the boundaries between two different texture and/or color regions. Region-based representation provides more meaningful information because more visual discrimination can be achieved from those regions ; moreover, the shapes of regions reflect a certain geometrical configurations of their belonging objects. For example, the scale of the unknown objects contained in the input image can be inferred based on the size of the region(s).

Because of such advantages, region-based interactions have been widely applied. For instance, [Galleguillos 2008, Rabinovich 2007b, Tighe 2010, Chen 2011a, Vieux 2011] model the interactions between over-segmented regions in order to merge them into complete objects with appropriate labels. Especially, [Galleguillos 2008, Chen 2011a] further consider relative location (below, above, around, inside) and label co-occurrence in order to exploit prior knowledge of the spatial arrangement of objects in scenes. A more sophisticated relative location prior is explored in [Gould 2008] while [Parikh 2008] learn the co-occurrence statistics conditioned on both position and scale.

5.2.4 Hierarchical Interaction

When region-wise interaction is not able to to model long-range dependencies, the range of interaction can be expanded via associating regions as a hierarchical tree in which its leaf nodes are image regions while inner nodes correspond to the union of several adjacent regions ; finally the root node covers the whole image. By relating adjacent regions using hierarchical models, contextual dependencies between remote regions can be taken into account. For instance, the two-layer hierarchical framework of [Kumar 2005] uses CRF model in order to interpret images at different levels of segmentation ; similarly, the hierarchical random field in [Ladicky 2009] allows the integration of the features computed at different quantization levels i.e.,

FIGURE 5.3 – The region-wise interaction approach requires an over-segmentation step in which the input image in partitioned into many regions (a.k.a superpixels). The figures above are the segmentation results of the graph-based algorithm [Felzenszwalb 2004] with different degrees of fraction. With over-segmentation (left), objects are likely to be divided into several regions ; with under-segmentation (right), several objects are merged with each other. Since the segmentation algorithm is unsupervised, choosing a good segmentation is very difficult. Nevertheless, this region-based approach reveals some information of the objects in the image : (i) some regions retain boundary information of the original objects, (ii) the size and the span of every region reflect the scale and the shape of the object(s) covered by that region.

pixel-, region- and inter-region levels. In contrast, [Munoz 2010], parses images into a hierarchy of regions and then solves every recognition subproblem for every region.

5.2.5 Source of Contextual Information

Contextual statistics are often computed from a database of labeled images, for instance [Rabinovich 2007b, Tighe 2010, Parikh 2008]. In other cases, they are computed from external sources such as Google Search's results and WordNet [Rabinovich 2007a]. Although retrieval results obtained by Google may be contaminated by noise, it is useful if the image database is poorly labeled or highly imbalanced. Recent works [Jain 2010, Malisiewicz 2009, Myeong 2012, Eigen 2012a, Tighe 2010] computed contextual statistics dynamically. For example, [Eigen 2012a, Tighe 2010, Myeong 2012] use the test image as a query in retrieving a subset of similar images from the labeled database. This subset is used as a source of training data and contextual statistics.

5.2.6 Machine Learning Techniques

Due to complex interactions between entities (pixels, objects, regions), graphical models [Bishop 2006] are used as a flexible machine learning framework. Graphical models provide convenient ways to mimic belief reasoning in human using probabilistic models. Specifically, they allow the factorization of a joint probability of random variables into a product of probabilities which are easier to model. Analogously, if

we define adjacent pixels or regions as random variables, then contextual interaction will be defined as the joint probability of those subset of variables. Graphical models are generally classified into *undirected* and *directed* models ; the former concern symmetric contextual relations between random variables and the latter emulates causal relationships between random variables [Torralba 2003, Singhal 2003, Li 2009]. Particular popular cases of undirected graphical models include Markov random field (MRF) [Carbonetto 2004, Tighe 2013, Myeong 2012, Eigen 2012a] and conditional random fields (CRF) [Kumar 2006, Kumar 2005, Rabinovich 2007b, Shotton 2009]. The major advantage of graphical models is their flexibility in modeling various types of contextual relations such as short-range (pixel-pixel or object-object), long-range (object-object or object-region), and hierarchical relations.

5.3 Contextual Kernel Learning

Different from the machine learning approaches mentioned above, we propose a novel solution for scene interpretation based on our transductive inference algorithm (introduced in Chapter 4). We adopt the data-driven approach [Liu 2011a, Russell 2007a, Tighe 2013, Eigen 2012a] in order to prepare the training data : the test image is used to retrieve from the labeled database a small set of training images which are similar to the test one. As depicted in Fig. 1.4, both the test and training images are over-segmented into superpixels ; they are the input for our algorithm based on multi-class transductive kernel learning. Beside transferring label information from the labeled superpixels to the unlabeled ones, the test image must be labeled with respect to contextual constraints. It means that prior knowledge about semantic relationships between object classes – the one that is extracted from the training data – are used to disambiguate predictions of superpixels whose appearances are not discriminative enough. We test with two regularization methods and found that such regularizers are useful for scene interpretation.

Compared with popular methods used for scene interpretation, our method is among the few algorithms which use transductive inference for scene interpretation. For instance, [Myeong 2012] uses a graph-based label propagation algorithm, which is a transductive method ; however, it is just a preprocessing step in order to learn contextual features fed to a later MRF-based inference stage. On the contrary, our scene interpretation framework is entirely based on multi-class transductive kernel learning, which jointly learns a shared low-dimensional kernel map and max-margin classifiers. While the multi-class formula is reused from Chapter 4, the contribution of this chapter is the introduction of new context regularizers. These regularizers implement two semantic rules of probability and position. In order to avoid confusion with mathematical notations of probability and statistics, “probability rules” is also referred to as “semantic context.”

Following the problem statement below, Sections 5.3.2 and 5.3.3 describe the construction of the semantic and position regularizers respectively. A nice property is that these two regularizers admit the same mathematical form so that the opti-

mization algorithm presented in Section 5.4 is identical for both of them. The end of this section is a short introduction of the spatial smoothing term and its use in smoothing recognition results.

5.3.1 Problem Statement

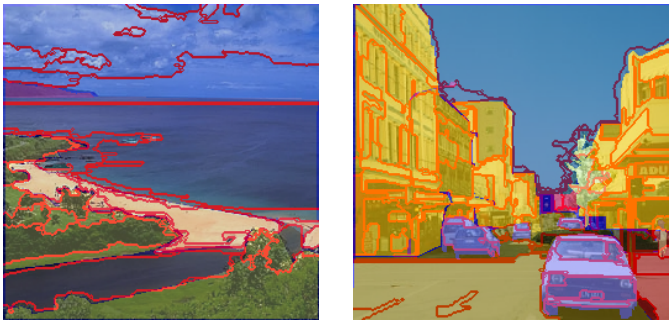
Given that the input data is the set $\mathcal{S} = \bigcup_{t=1}^{T+1}\{(\mathbf{x}_1^t, y_1^t), \ldots, (\mathbf{x}_{m_t}^t, y_{m_t}^t)\}$ which is the union of superpixels extracted from $(T+1)$ images in which the first T images are labeled and the $(T+1)^{\text{th}}$ one is unlabeled. In order to cancel image index t, we re-index the input data as follows. The first ℓ superpixels, which are extracted from the T training images, are labeled, i.e., $\mathbf{y}_i \neq 0,\ \forall i = 1, \ldots, \ell$; the next $(m-\ell)$ superpixels, which are extracted from the test image, are unlabeled; notice that $m = \sum_{t=1}^{T+1} m_t$. Our objective is to assign to each of the unlabeled superpixels $\{\mathbf{x}_i\}_{i=\ell+1}^{m}$ one of the K labels – these labels must appear at least once in the labeled images. Although a superpixel may contain more than one object part, for simplicity we assume that every superpixel is labeled by one class; in cases a superpixel contains several labels, then the most dominant one is chosen as the groundtruth. As a result, entries of label vector $\mathbf{y}_i \in \mathbb{R}^K$ $(i \leq \ell)$ are zeros except the c^{th} entry where c is the index of the semantic presented in superpixel i.

Given a finite training sample $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, a function of the form $\mathbf{y} = \mathbf{W}'\phi(\mathbf{x})$ is learned in order to relate an input descriptor $\mathbf{x}$ with the corresponding output label $\mathbf{y}$; here $\phi(\mathbf{x})$ is a kernel map and $\mathbf{W} \in \mathbb{R}^{p\times K}$ is the weight matrix where p is the dimensionality of the target feature space defined by ϕ. In the previous chapter we introduced the multi-class transductive kernel learning whose objective is to learn the kernel map $\mathbf{\Phi} = \phi(\mathbf{X})$ and the classifier $\mathbf{W}$ from the input data $\mathbf{X}$ whose label matrix $\mathbf{Y}$ is partially labeled, i.e., only the first ℓ columns are known. Using this formulation as a basis for the scene interpretation problem, we design additional regularizers in order to tailor our basic formulation to the specificities of the problem. The proposed regularizers exploit the semantic and position relationships between object labels in scene images such that the labeled result is coherent with respect to the contextual rules.

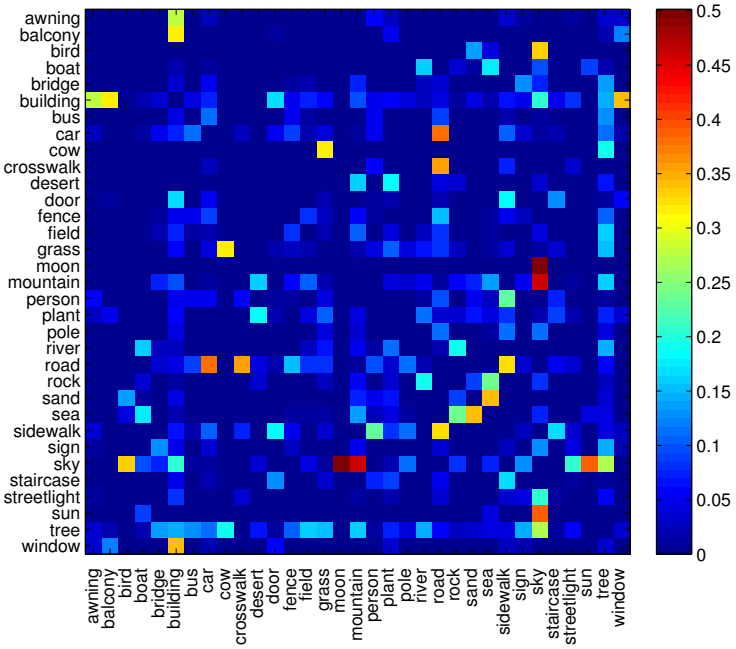
5.3.2 Regularization on the Semantic Context

If it is difficult to recognize an object by its appearance, then its surrounding objects (a.k.a its context) may reveal valuable information about the object of interest (see again the example in Fig. 5.2). In practice, this sort of contextual support is frequently used by human : we look for cars on roads, boats on waters, books on tables, flowers in gardens, etc. Objects in these examples not only co-occur together but also are close to each other. In other word, these examples are the instances of the probability rule mentioned in Section 5.1.

Let us denote $p(c|c')$ the conditional probability for an object with label c to be adjacent with another object with label c'; if the probability $p(c|c')$ is high, then it is frequent – in the image database – that objects of label c are adjacent



(a) Some examples of the data used to compute the co-occurrence statistics between object labels.



(b) The likelihood table which shows conditional probabilities of a label to appear close to another label. A strong probability is displayed with a hotter color, i.e., the probability values of $p(\text{sky}|\text{mountain})$, $p(\text{car}|\text{road})$, $p(\text{window}|\text{building})$.

FIGURE 5.4 – The computation of the co-occurrence statistics of the semantic context.

with objects of label c'. Furthermore, if $p(c|c') > p(c|c'')$, then objects with label c are more likely to co-occur with objects of label c' than those of label c''. If the conditional probability $p(c|c')$ is high enough, then it is more likely that $c \leftarrow \text{argmax}_b \mathbf{W}_b \mathbf{\Phi}_i$ and $c' \leftarrow \text{argmax}_b \mathbf{W}_b \mathbf{\Phi}_j$ given that superpixels i and j are adjacent. Without loss of generality, let us assume that the response $\mathbf{W}_c \mathbf{\Phi}_i$ is already high such that the superpixel i gets label c, then the following energy term is minimized

for the superpixel j to get label c' :

$$\frac{1}{4}\sum_{i=\ell+1}^{m}\sum_{j=\ell+1}^{m}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left(\mathbf{W}'_c\mathbf{\Phi}_i-\mathbf{W}'_{c'}\mathbf{\Phi}_j\right)^2 p(c|c')\delta(i\sim j). \tag{5.1}$$

In the above equation, the delta Dirac function $\delta(i\sim j)=1$ if only if the superpixel i is adjacent to the superpixel j. Our problem is how to compute $p(c|c')$ given a labeled database. Let us denote the counting matrix $\mathbf{N}\in\mathbb{R}^{K\times K}$ whose entry at row c and column c' equals

$$\mathbf{N}_{cc'}=\sum_{t=1}^{T}\sum_{i=1}^{m_t}\sum_{j=1}^{m_t}\delta(i\sim j)\times\left(\frac{[y_i^t]_c}{q_c^t}\times\frac{[y_j^t]_{c'}}{q_{c'}^t}\right), \tag{5.2}$$

in which $[y_i^t]_c$ and $[y_j^t]_{c'}$ are the values at c^{th} and c'^{th} entries of label vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ given that both i and j are two superpixels of the t^{th} training image. The normalization factors q_c^t and $q_{c'}^t$ are computed as

$$q_c^t=\sum_{i=1}^{m_t}[y_i^t]_c \quad \text{and} \quad q_{c'}^t=\sum_{i=1}^{m_t}[y_i^t]_{c'}. \tag{5.3}$$

The formula (5.2) means that the entry $\mathbf{N}_{cc'}$ is increased by an amount of $1/(q_c^t q_{c'}^t)$ whenever a co-occurrence between the labels c and c' is detected given that $\delta(i\sim j)=1$. The factors q_c^t and $q_{c'}^t$, which are the number of superpixels labeled by c in t^{th} image, are used to reduce the effect of uneven object size. For instance, images in Fig. 5.4(a) are over-segmented into dozens of superpixels ; a building is decomposed into many more superpixels than those of a car due to the relatively bigger size and more sophisticated details of a building than a car. If we increased by 1 the occurrence of a pair of adjacent superpixels labeled as building and car respectively, then the accumulating matrix $\mathbf{N}$ becomes superfluous because there may exist several superpixels of the building which are adjacent to one superpixel of the car. In other word, the entry $\mathbf{N}_{\text{building|car}}$ would be counted several times more than $\mathbf{N}_{\text{car|building}}$. Besides, our implication in using context is to support more for rare and small objects because more frequent or bigger objects can be identified easier : a larger area contains more visual details which lead to better discrimination (compare small versus large superpixels in Fig. 5.4(a)). Following (5.2), an object with a big size results into a large number of superpixels decomposed by that object and this leads to a larger value of the factor q_c^t. Eventually (5.2) counts more for co-occurrences of two small objects, less for those of a big and a small objects, and least for those of two big objects. Notice that object size is not determined in (5.2) but the preprocessing step of over-segmentation (see Fig. 5.4(a)). Consequently, (5.2) counts label co-occurrences and remedies imbalanced data at the same time.

In order to convert the counting values into probability, every entry $\mathbf{N}_{cc'}$ is divided by their sum of row and column respectively, i.e.,

$$p(c'|c)=\frac{\mathbf{N}_{cc'}}{\sum_{c''}\mathbf{N}_{cc''}} \quad \text{and} \quad p(c|c')=\frac{\mathbf{N}_{cc'}}{\sum_{c''}\mathbf{N}_{c''c'}}. \tag{5.4}$$

5.3.3 Regularization on the Position Context

The next contextual relationship that we consider is the position co-occurrence between labels. This relationship is the implementation of the position rule, which states that the co-occurrence between labels is likely to appear at some specific positions and less likely to appear at other positions. There are many examples supporting this rule, for instance the sky is above and the sea is below (and see more examples in Fig. 5.5). Given two superpixels i and j positioned at the areas R_i and R_j respectively in the test image, $p(c|c';R_i,R_j)$ is the probability for the superpixel i located at R_i to get label c conditioned on the superpixel j located at R_j gets label c'. In this notation, c and c' are random variables while R_i and R_j are not ; instead R_i and R_j are the fixed regions of the superpixels i and j. Similarly to the previous context regularizer, the position context is introduced into our learning framework by minimizing the following energy term

$$\frac{1}{4}\sum_{i=\ell+1}^{m}\sum_{j=\ell+1}^{m}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left(\mathbf{W}'_c\mathbf{\Phi}_i-\mathbf{W}'_{c'}\mathbf{\Phi}_j\right)^2 p(c|c';R_i,R_j). \tag{5.5}$$

The rest problem is how to evaluate the probability $p(c|c';R_i,R_j)$ given a database with labeled superpixels and their positions. Due to irregular shapes of superpixels, it is easier for us to estimate the co-occurrence frequency by partitioning an image using a rectangular grid ; it is a 10×10 grid in our case. Then we compute the joint probability $p(z,z',c,c')$ measuring the co-occurrence of labels c and c' at their respective positions z and z'. Here z and z' get one of $10\times 10=100$ different positions in the image. By counting the number of images whose positions at z and z' are labeled as c and c', we obtain the joint probability $p(c,c',z,z')$, i.e.,

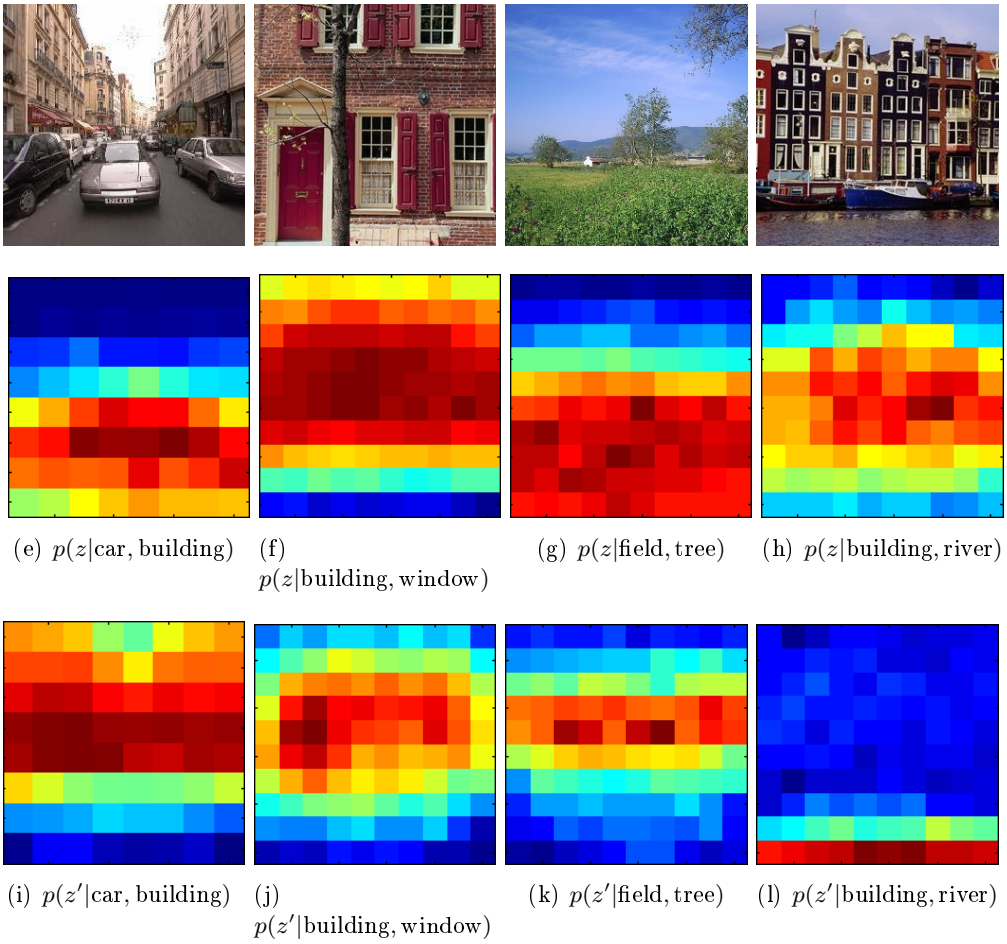
$$p(c,c',z,z')=\frac{\sum_{t=1}^{T}\delta\left(c=\mathcal{I}_t[z]\right)\delta\left(c'=\mathcal{I}_t[z']\right)}{\sum_{\bar{c}}\sum_{\bar{c}'}\sum_{\bar{z}}\sum_{\bar{z}'}\sum_t\delta\left(\bar{c}=\mathcal{I}_t[\bar{z}]\right)\delta\left(\bar{c}'=\mathcal{I}_t[\bar{z}']\right)}, \tag{5.6}$$

in which the Dirac delta function $\delta(c=\mathcal{I}_t[z])$ returns 1 only if the label at position z of image $\mathcal{I}_t$ is c, otherwise it returns zero. Examples in Fig. 5.5 illustrate the likelihood maps $p(z|c,c')$ between couples of labels, which are computed by marginalizing over the position variable z' : $p(z|c,c')=\sum_{z'}p(z,z'|c,c')=\sum_{z'}p(z,z',c,c')/p(c,c')$. Once joint probabilities are computed, the conditional probability $p(c|c';R_i,R_j)$ is derived as follows

$$p(c|c';R_i,R_j)=\frac{p(c,c';R_i,R_j)}{p(c';R_i,R_j)}=\frac{\sum_{z\in R_i}\sum_{z'\in R_j}p(c,c',z,z')}{\sum_{c''}\sum_{z\in R_i}\sum_{z'\in R_j}p(c'',c',z,z')}. \tag{5.7}$$

5.3.4 Spatial Smoothing

Objects are segmented into lots of superpixels and some of them may not be discriminative enough. When the contextual regularizers cannot help recovering the true labels of such superpixels, a simple operator such as spatial smoothing may



(e) $p(z|\text{car}, \text{building})$ (f) $p(z|\text{building}, \text{window})$ (g) $p(z|\text{field}, \text{tree})$ (h) $p(z|\text{building}, \text{river})$

(i) $p(z'|\text{car}, \text{building})$ (j) $p(z'|\text{building}, \text{window})$ (k) $p(z'|\text{field}, \text{tree})$ (l) $p(z'|\text{building}, \text{river})$

FIGURE 5.5 – By observing image examples in the top row, we would like to find the spatial correlation between labels such as *car* and *building*, *window* and *building*, *tree* and *field*, *building* and *river*. Shown in the middle and the bottom rows are the likelihood $p(z|c, c')$ of some label pairs $\{c, c'\}$ whose example are illustrated in the top row. These conditional probabilities are computed by marginalizing one out of the two position variables z and z', i.e., $p(z|c, c') = \sum_{z'} p(z, z'|c, c')$. The visualizations of the maps reflect our belief about the spatial relationship between those label pairs. For instance, Fig. 5.5(e) shows that *car* usually appears in the lower part of the scene ; Fig. 5.5(i) shows that *building* is likely to appear higher and above *car*.

be useful. The idea of this operator is borrowed from the pixel-based approach for scene interpretation (for instance [Tighe 2013]) where adjacent pixels are likely to get similar labels. Given a test image with unlabeled superpixels, we construct an adjacency matrix $\mathbf{S} \in \mathbb{R}^{(m-\ell)\times(m-\ell)}$ between superpixels, i.e., $\mathbf{S}_{ij} = 1$ if the underlying superpixels are adjacent and $\mathbf{S}_{ij} = 0$ otherwise. Similarly to the smoothness term (4.3), the spatial smoothing term admits the convex quadratic form

$$\frac{1}{4}\sum_{i=\ell+1}^{m}\sum_{j=\ell+1}^{m}\left\|\mathbf{W}'\mathbf{\Phi}_i - \mathbf{W}'\mathbf{\Phi}_j\right\|^2 \mathbf{S}_{ij}. \tag{5.8}$$

In our implementation, we normalize the matrix $\mathbf{S} \leftarrow (\mathbf{D}^s)^{-1/2}\mathbf{S}(\mathbf{D}^s)^{-1/2}$ where $\mathbf{D}^s = \mathrm{diag}(\mathbf{S}\mathbf{1}_{(m-\ell)})$, thus the corresponding graph Laplacian is $\mathbf{L}^s = \mathbf{I} - \mathbf{S}$.

5.4 Optimization

Since both (5.1) and (5.5) admit the same mathematical formula, they share the same objective function and optimization algorithm. Let us denote $\{\mathbf{P}^{cc'}\}$ the conditional probability matrices ; the entry $[\mathbf{P}^{cc'}]_{ij}$ of every square matrix $\mathbf{P}^{cc'} \in \mathbb{R}^{(m-\ell)\times(m-\ell)}$ is assigned : i) the conditional probability $p(c|c')\delta(i \sim j)$ if semantic context is used and ii) $p(c|c'; R_i, R_j)$ if position context is used. In order guarantee that $\mathbf{P}^{cc'}$ symmetric (which provides better labeling results), we assign $\mathbf{P}^{cc'} \leftarrow \frac{1}{2}\left(\mathbf{P}^{cc'} + \left(\mathbf{P}^{cc'}\right)'\right)$. We also denote $\{\mathbf{M}^{cc'}\}$ the diagonal matrices in which $\mathbf{M}^{cc'} = \mathrm{diag}(\mathbf{P}^{cc'}\mathbf{1})$. By padding zeros into matrices $\mathbf{M}$ and $\mathbf{P}$ such that

$$\tilde{\mathbf{M}}^{cc'} = \begin{pmatrix} \mathbf{0}_{\ell\times\ell} & \mathbf{0}_{\ell\times(m-\ell)} \\ \mathbf{0}_{(m-\ell)\times\ell} & \mathbf{M}^{cc'} \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{P}}^{cc'} = \begin{pmatrix} \mathbf{0}_{\ell\times\ell} & \mathbf{0}_{\ell\times(m-\ell)} \\ \mathbf{0}_{(m-\ell)\times\ell} & \mathbf{P}^{cc'} \end{pmatrix}, \tag{5.9}$$

the block matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{P}}$ have the same size $m \times m$. The matrix-based formula of (5.1) and (5.5) admit the same mathematical expression

$$\frac{1}{4}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left(\mathbf{W}'_c\mathbf{\Phi} - \mathbf{W}'_{c'}\mathbf{\Phi}\right)\tilde{\mathbf{P}}^{cc'}\left(\mathbf{W}'_c\mathbf{\Phi} - \mathbf{W}'_{c'}\mathbf{\Phi}\right)', \tag{5.10}$$

or equivalently

$$\frac{1}{4}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left(\mathbf{W}'_c\mathbf{\Phi}\tilde{\mathbf{M}}^{cc'}\mathbf{\Phi}'\mathbf{W}_c + \mathbf{W}'_{c'}\mathbf{\Phi}\tilde{\mathbf{M}}^{cc'}\mathbf{\Phi}'\mathbf{W}_{c'} - 2\mathbf{W}'_c\mathbf{\Phi}\tilde{\mathbf{P}}^{cc'}\mathbf{\Phi}'\mathbf{W}'_{c'}\right). \tag{5.11}$$

Due to the interchangeability between c and c', (5.11) is equivalent to

$$\frac{1}{2}\sum_{c=1}^{K}\sum_{c'=1}^{K}\left(\mathbf{W}'_c\mathbf{\Phi}\tilde{\mathbf{M}}^{cc'}\mathbf{\Phi}'\mathbf{W}_c - \mathbf{W}'_c\mathbf{\Phi}\tilde{\mathbf{P}}^{cc'}\mathbf{\Phi}'\mathbf{W}_{c'}\right). \tag{5.12}$$

Combining altogether equations (4.2), (4.3), (5.8) and (5.12) we obtain the optimization problem for scene interpretation

$$\begin{aligned}\min_{\mathbf{B},\mathbf{\Phi},\mathbf{W}} \quad & \tfrac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \tfrac{\gamma}{2}\sum_{c,c'}\left(\mathbf{W}_c'\mathbf{\Phi}\tilde{\mathbf{M}}^{cc'}\mathbf{\Phi}'\mathbf{W}_c - \mathbf{W}_c'\mathbf{\Phi}\tilde{\mathbf{P}}^{cc'}\mathbf{\Phi}'\mathbf{W}_{c'}\right) + \\ & + \tfrac{1}{2}\mathrm{tr}\Big(\mathbf{W}'\left(\mathbf{I} + \mathbf{\Phi}\left(\beta\mathbf{L} + \theta\mathbf{L}^s\right)\mathbf{\Phi}'\right)\mathbf{W}\Big) + \tfrac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}\\ \mathbf{0} & \mathbf{W}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2, \\ \text{s.t} \quad & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p\end{aligned} \tag{5.13}$$

in which γ is the context coefficient and θ is the spatial smoothing coefficient. Whenever necessary, we further specify γ_s and γ_p as the coefficients of the semantic and position contexts respectively.

5.4.1 Updating Classifier and Basis

There is a little change in the update rule for classifier compared with (3.6). Since classifiers $\mathbf{W}_c$'s depend on each other, an iterative optimization procedure is required for $\mathbf{W}$ to converges to a stationary solution. Assuming fixed $\mathbf{\Phi}^{(t)}$ (denoted simply as $\mathbf{\Phi}$) and enforcing the gradient of (5.13) to vanish (with respect to $\mathbf{W}$) leads to $\mathbf{W}^{(t)} = \tilde{\mathbf{V}}$ with $\tilde{\mathbf{V}} = \lim_{\varsigma\to\varsigma_{\max}} \mathbf{V}^{(\varsigma)}$ and

$$\mathbf{V}_c^{(\varsigma)} = \left(\mathbf{I} + \mathbf{\Phi}\left(\alpha\mathbf{C} + \beta\mathbf{L} + \gamma\sum_{c'}\tilde{\mathbf{M}}^{cc'} + \theta\mathbf{L}^s\right)\mathbf{\Phi}'\right)^{-1} \cdot \left[\alpha\mathbf{\Phi}\mathbf{C}\mathbf{Y}_c' + \gamma\sum_{c'}\mathbf{\Phi}\tilde{\mathbf{P}}^{cc'}\mathbf{\Phi}'\mathbf{V}_{c'}^{(\varsigma-1)}\right], \tag{5.14}$$

where $\mathbf{V}^{(0)} = \mathbf{W}^{(t-1)}$. In order to find $\mathbf{B}$, let us assume that $\mathbf{\Phi}^{(t)}$ and $\mathbf{W}^{(t)}$ fixed, then the following optimization problem

$$\min_{\mathbf{B}} \quad \tfrac{1}{2}\|\mathbf{X} - \mathbf{B}\mathbf{\Phi}\|_F^2 \quad \text{s.t} \quad \|\mathbf{B}_i\|_2^2 = 1, i = 1,\ldots,p. \tag{5.15}$$

is solved similarly as presented in Section 3.3.1.

5.4.2 Updating Kernel Map

Considering fixed $\mathbf{B}^{(t+1)}$ and $\mathbf{W}^{(t+1)}$ (denoted simply as $\mathbf{B}$, $\mathbf{W}$ in the remainder of this section), and the previous kernel map solution $\mathbf{\Phi}^{(t)}$, our goal is to find $\mathbf{\Phi}^{(t+1)}$ by solving (5.13). The optimization problem (5.13) admits a unique solution $\mathbf{\Phi}^{(t+1)} = \tilde{\mathbf{\Psi}}$ where $\tilde{\mathbf{\Psi}} = \lim_{\tau\to\tau_{\max}} \mathbf{\Psi}^{(\tau)}$ and

$$\mathbf{\Psi}_i^{(\tau)} = \left(\mu\mathbf{I} + \alpha\mathbf{B}'\mathbf{B} + \left(\alpha\mathbf{C}_{ii} + \beta\mathbf{D}_{ii} + \theta\mathbf{D}_{ii}^s\right)\mathbf{W}\mathbf{W}' + \gamma\sum_{cc'}\left[\tilde{\mathbf{M}}^{cc'}\right]_{ii}\mathbf{W}_c\mathbf{W}_{c'}'\right)^{-1} \cdot \left[\alpha\left(\mathbf{B}'\mathbf{X} + \mathbf{W}\mathbf{Y}\mathbf{C}\right) + \mathbf{W}\mathbf{W}'\mathbf{\Psi}^{(\tau-1)}(\beta\mathbf{A} + \theta\mathbf{S}) + \gamma\sum_{c,c'}\mathbf{W}_c\mathbf{W}_{c'}\mathbf{\Psi}^{(\tau-1)}\tilde{\mathbf{P}}^{cc'}\right]_i \tag{5.16}$$

where $\mathbf{\Psi}^{(0)} = \mathbf{\Phi}^{(t-1)}$. The process described in (5.16) allows us to recursively diffuse the kernel maps from the labeled to the unlabeled data, through the neighborhood system defined in the graph $\{\mathcal{V}, \mathcal{E}\}$. The algorithm terminates when either $\|\mathbf{\Psi}^{(\tau)} - \mathbf{\Psi}^{(\tau-1)}\| \leq \varepsilon$ or the iterative optimization algorithm reaches $t_{\max}$ iterations.

5.5 Experimental Setup

5.5.1 Dataset

In order to validate the proposed method, we use a standard subset of the LabelMe database. This subset, known as the SiftFlow dataset, is commonly used since [Russell 2007a] in order to benchmark scene interpretation methods. The subset contains 2688 images sampled from 8 scene categories : sea, forest, highway, city street, building, mountain, countryside, and skyscraper. Images within each category are diversified in locations, atmospheres, illuminations, and viewpoints. Following previous works, we split the subset into 2488 images for training and the rest 200 images for testing. In our experiments, images are over-segmented into superpixels using the graph-based segmentation algorithm [Felzenszwalb 2004]. Every superpixel of the training images is assigned a unique label based on ground-truth data. Since a superpixel may partly cover several objects, then its unique label is the most dominant one.

Although the number of images is approximately equal between scene categories, the dataset is still imbalanced with respect to the number of object instances ; this is due to the uneven occurrence frequency of the object classes. For instance, buildings appear more often than trees in urban scenes. This imbalance is necessary because popular objects such as *building* often have more appearance variances than less popular objects such as *tree* or *sidewalk*. However, the side effect of data imbalance in our problem is that the degree of data imbalance is exacerbated due to the region-based (a.k.a superpixel) representation which is introduced shortly. For labels with big size objects such as *building*, *sky* or *sea*, the image over-segmentation step creates several times larger the number of superpixels than the number of object instances ; the quantities of data with these labels are increased proportionally to their number of superpixels. The number of superpixels generated from small size objects are not so large as those of the big size objects. The data imbalance, as a result, becomes much worse. However, as discussed in Sections 5.3.2 and 5.3.3 and the empirical results going to be shown, context regularizers can alleviate this imbalance.

5.5.2 Subset Retrieval

Given a test image, we need to retrieve an image subset from the labeled database such that this subset contains images which are similar to the test one. This subset is used as a training set for our algorithm. This intermediate step is necessary because the database may be too large to be fitted into the memory ; in addition, irrelevant images may be a source of noise. By restricting the size of the training

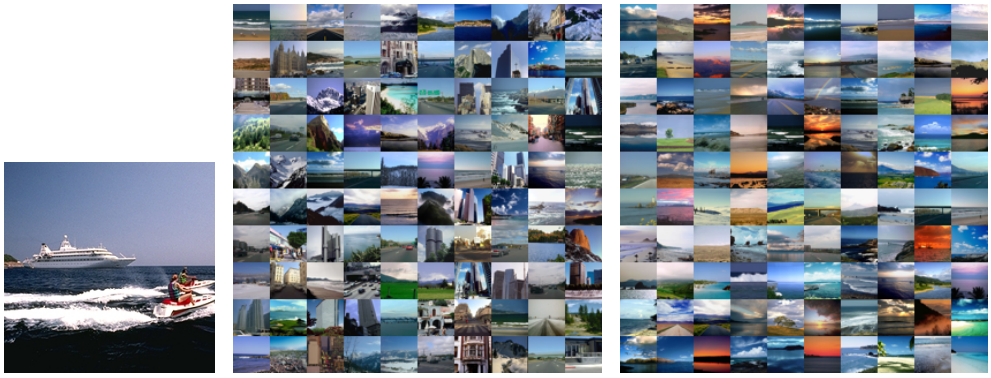
set, we obtain several advantages. First, more similar images provide a good source of labeled superpixels whose semantic content may be more relevant to the test one. Second, reducing data redundancy clearly saves computational cost.

The subset retrieval step is not new and has been used in [Liu 2011a, Russell 2007a, Tighe 2013, Eigen 2012a]. We follow the setup of [Tighe 2013] in which three holistic features – a pyramid of denseSIFT histogram [Lazebnik 2006b], GIST descriptor [Oliva 2006], and color histogram – are used to retrieve the T most similar labeled images. For every feature, the labeled images in the database are sorted in ascending order of the Euclidean distances between them and the test image. The three sorted lists are mixed into a final list by choosing from the lists the minimum rank of every image. The top T ranked images are chosen as the images in the retrieved subset. In general, there always exists in the retrieved result some images whose semantic content is not relevant to the test one. These images may be considered as a source of noise.

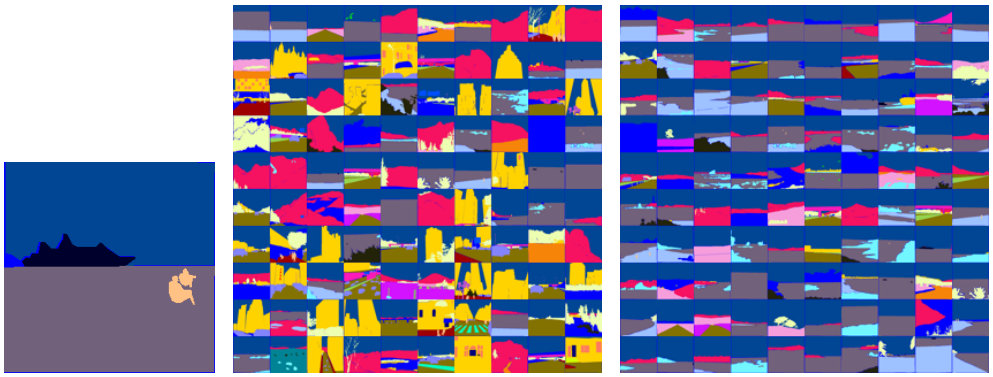
Another heuristic proposed by [Singh 2013] is to use label-based features in order to retrieve training images. This heuristic consists of two steps. In the first step, every training image in the database is labeled by our method in which the training data is retrieved based on visual features mentioned above. Once all the training images are labeled, we use this labeling information as the features for the second retrieving step. By applying a pyramidal grid consisting of three layers $1 \times 1, 2 \times 2, 3 \times 3$ into the labeling results of every image in the dataset, a histogram of label frequency for every cell of the grid is computed ; concatenating histograms of all the cells give us a label-localized histogram used in the fine labeling step. In this second step, image similarity is computed based on such label histograms. Although the labeling in the first step may not be correct with respect to the semantic content of images, some background objects such as *sky*, *building* and *road* can also be partly recognized. And this is an important cue for the second step to retrieve more similar images. An example comparing the two techniques is illustrated in Fig. 5.6.

5.5.3 Features Extraction and Graph Construction

After retrieving the subset of T similar images, we segment the training and test images into superpixels using the fast graph-based segmentation algorithm [Felzenszwalb 2004]. Superpixels are represented by a set of 26 visual descriptors [Malisiewicz 2009, Tighe 2010] of 5 feature groups : shape, location, texture & SIFT, color, and appearance. In particular, the shape group contains information about the bounding box of the superpixel, relative size and area of the superpixel with respect to the image. The location group consists of a downsampled image of the superpixel-masked image and the relative height of the superpixel with respect to the image. The texture/SIFT group consists of texton and SIFT histograms of the superpixels as well as its left, right, top, and bottom extents. The color group consists of color histogram, the means and standard deviations of each color channel. The appearance group consists of the thumbnail of the image of the superpixel, the thumbnail of the image masked by the superpixel, and gist descriptor of the



(a) Test image (b) The subset retrieved using visual features. (c) The subset retrieved using pyramidal features of labels.



(d) Ground-truth (e) Ground-truth of the visual features-based subset. (f) Ground-truth of the label features-based subset.

FIGURE 5.6 – The results of retrieving the subset of training data using (b,e) visual-based and (c,f) label-based features. Given a test image (a), we use it as the query image in order to retrieve from the image database a subset of images which are similar to the query one. The quality of the retrieved subset is an important factor for our algorithm to successfully interpret semantic content of the test image. There are two ways for the retrieval : (i) based on visual features, (ii) based on label features of a preliminary labeling step. By comparing the retrieved results between (e) and (f), it is clear that the second method is more effective in retrieving similar images.

superpixel.

Given m superpixels extracted from $(T+1)$ images, we obtained 26 sets of feature vectors $\{\{\mathbf{x}_i^j\}_{i=1}^m\}_{j=1}^{26}$. These features are used to construct an adjacency graph whose vertices represent superpixels and edges represent the visual similarity between superpixels. The similarity $\mathbf{A}_{ab}$ between two superpixels a and b is computed based on (4.10). The resulting similarity computation is a square matrix $\mathbf{A}$ of size m. For every row $\mathbf{A}_a$. of matrix $\mathbf{A}$, just k largest values are kept (except diagonal entry $\mathbf{A}_{aa}$) and the rest is set to zero. In other word, every superpixel is connected to its k

most similar superpixels, which are extracted from either one of training images or the test one. Since $\mathbf{A}$ may not be symmetric, we fix this by using $\mathbf{A} \leftarrow (\mathbf{A} + \mathbf{A}')/2$ instead of $\mathbf{A}$.

5.5.4 Evaluation

Similarly to related works, we evaluate the accuracy based on two criteria *per-pixel* classification rate and *per-class* classification rate. The first measurement computes the average of how many percent of pixels per image are accurately labeled. This is the most intuitive measurement because it is closely related to our feeling in judging how good an image is interpreted. However, the per-pixel rate favors big and popular objects such as *building* and *sky* and ignores errors in rare labels such as *window* and *sign*. The other measurement can overcome such situations ; per-class rate is not affected by the object's size because we compute – for every image – the average of the percentage of the pixels labeled correctly with respect to every label. This rate is then averaged over the number of test images. We prefer scene interpretation systems that obtain high values for both rates.

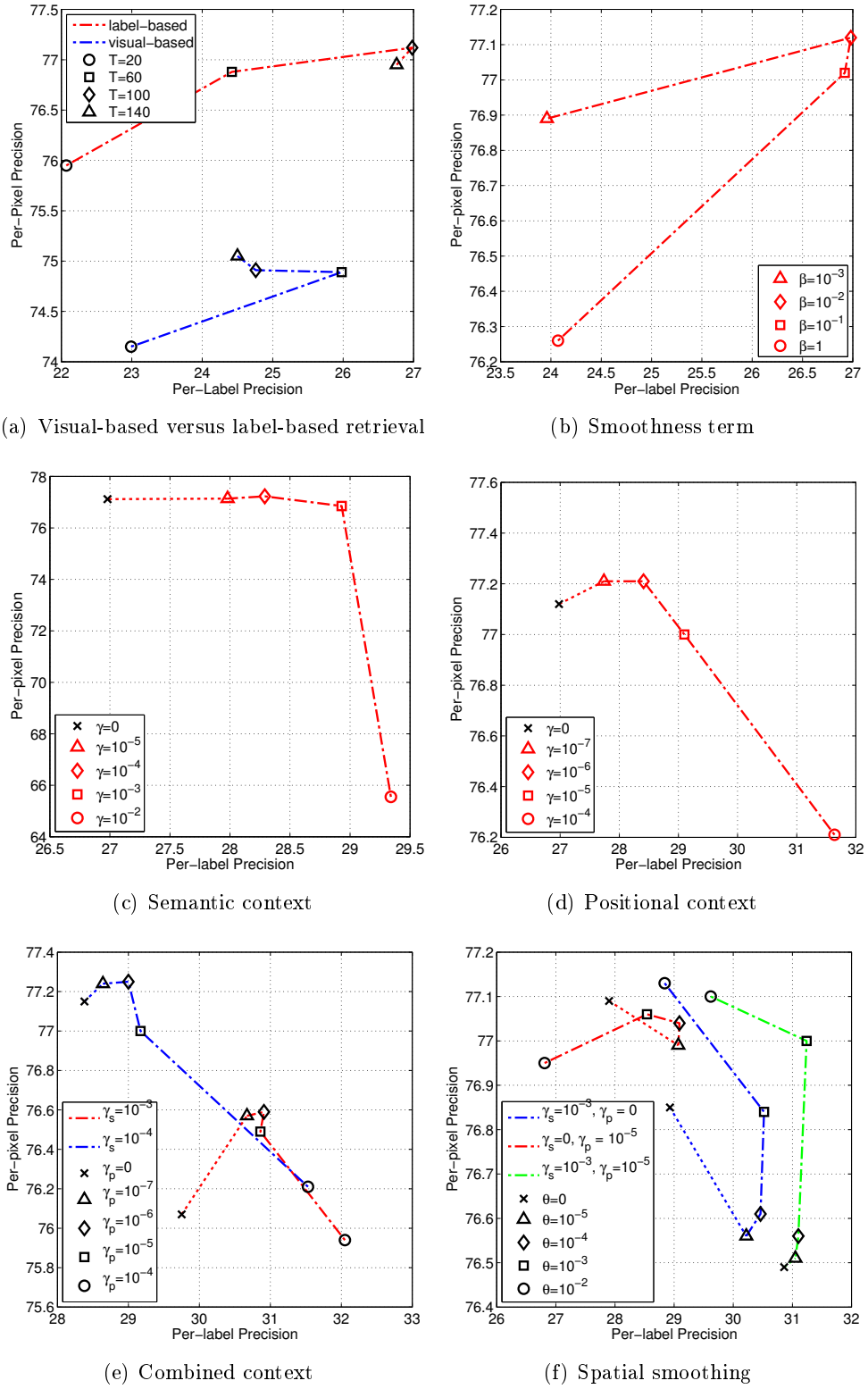
5.6 Results and Discussions

There are four factors that affect the result of our algorithm : the features used in the subset retrieval step, the size T of the subset, the role of the smoothness regularizer, and the role of the contextual regularizers. Each of these factors will be investigated in this section. The rest parameters are fixed as follows : the fidelity coefficient $\alpha = 1$, the neighborhood size $k = 20$, the max numbers of iterations $\varsigma_{\max} = 5$, $\tau_{\max} = 20$, $t_{\max} = 5$, and convergence criterion $\varepsilon = 10^{-2}$.

5.6.1 Analysis of Subset Retrieval

As mentioned earlier in this chapter, there are two methods in retrieving similar training images from the dataset. The first one is to use a fusion of pyramid Bag-of-Words SIFT histogram, color histogram, and GIST descriptor, in order to estimate the visual similarity between images. The second one is to use label-based features of the training images, which are preliminarily labeled by a baseline method. According to Fig. 5.7(a), using the label-based retrieval method may improve the interpretation results. However, the disadvantage of this method is that labeling errors made by the preliminary labeling step are kept in the main labeling step. Such erroneous label features are unrecoverable. Illustrated in Fig. 5.8 are some examples where the label-based method leads to a better labeling in the three first images and the visual-based method does it better in the next three images. A combination of the two feature types – label-based and visual-based – may resolve these counter examples.

The quality of the training data depends on not only its similarity to the test one but also its abundance. More training images means more superpixels ; thus our algorithm has a richer source of labeled data for the unlabeled superpixels in order to



(a) Visual-based versus label-based retrieval

(b) Smoothness term

(c) Semantic context

(d) Positional context

(e) Combined context

(f) Spatial smoothing

FIGURE 5.7 – Analyses on : (a) the effectiveness of the visual-based and the label-based retrieval heuristic, (b) the behavior of the smoothness term that diffuses the labels from the labeled to the unlabeled data, (c-d) how individuals and (e) combination of contextual regularizers help improve the per-label rate, and (f) how the spatial smoothing can correct labeling defects.

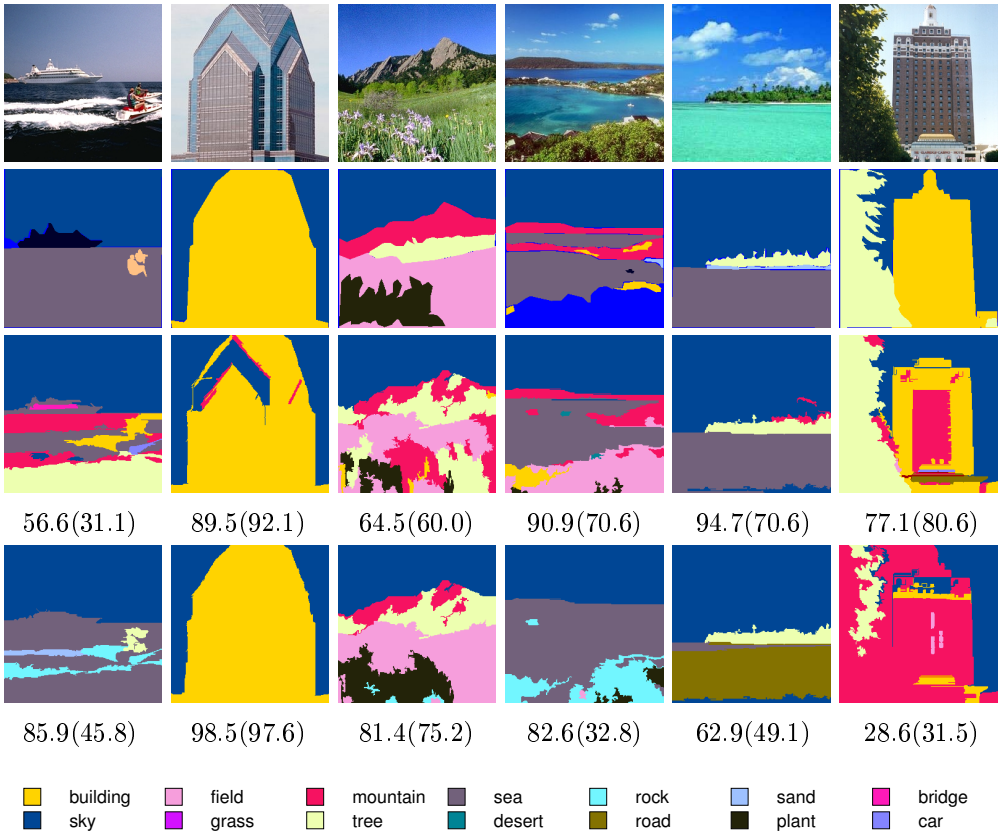


FIGURE 5.8 – Comparison of the methodologies used in the subset retrieval step. **1**st row : test images ; **2**nd row : ground-truth ; **3**rd row : results with visual-based subset retrieval ; **4**th row : results with label-based subset retrieval. The three first columns shows examples in which the label-based method provides better results ; the rest columns are the examples where the visual-based method is better. The per-pixel rate is shown under every result while the per-label rate is shown in bracket.

find their good matches. It is particularly true for rare labels. As indicated in Fig. 5.8, adding more training images helps our algorithm to fix labeling errors caused by the lack of labeled data. The plot in Fig. 5.7(a) also indicates that by increasing the subset size T from 20 to 100 the per-pixel rate increases from 76.0% to 77.1% and the per-label rate increases from 22.1% to 27.0%. It seems that rare labels benefit most from the abundance of training images. Related examples in Fig. 5.8 with the plot in Fig. 5.7(a) allow us to confirm this fact. However, performance slightly drops as T increases up to 140. This may be due to the fact that images ranked from 100^{th} to 140^{th} may be pretty irrelevant to the test one ; as a consequence, superpixels of those images are rather noisy than informative. The optimal choice of T may not be fixed but changes from one database to another.

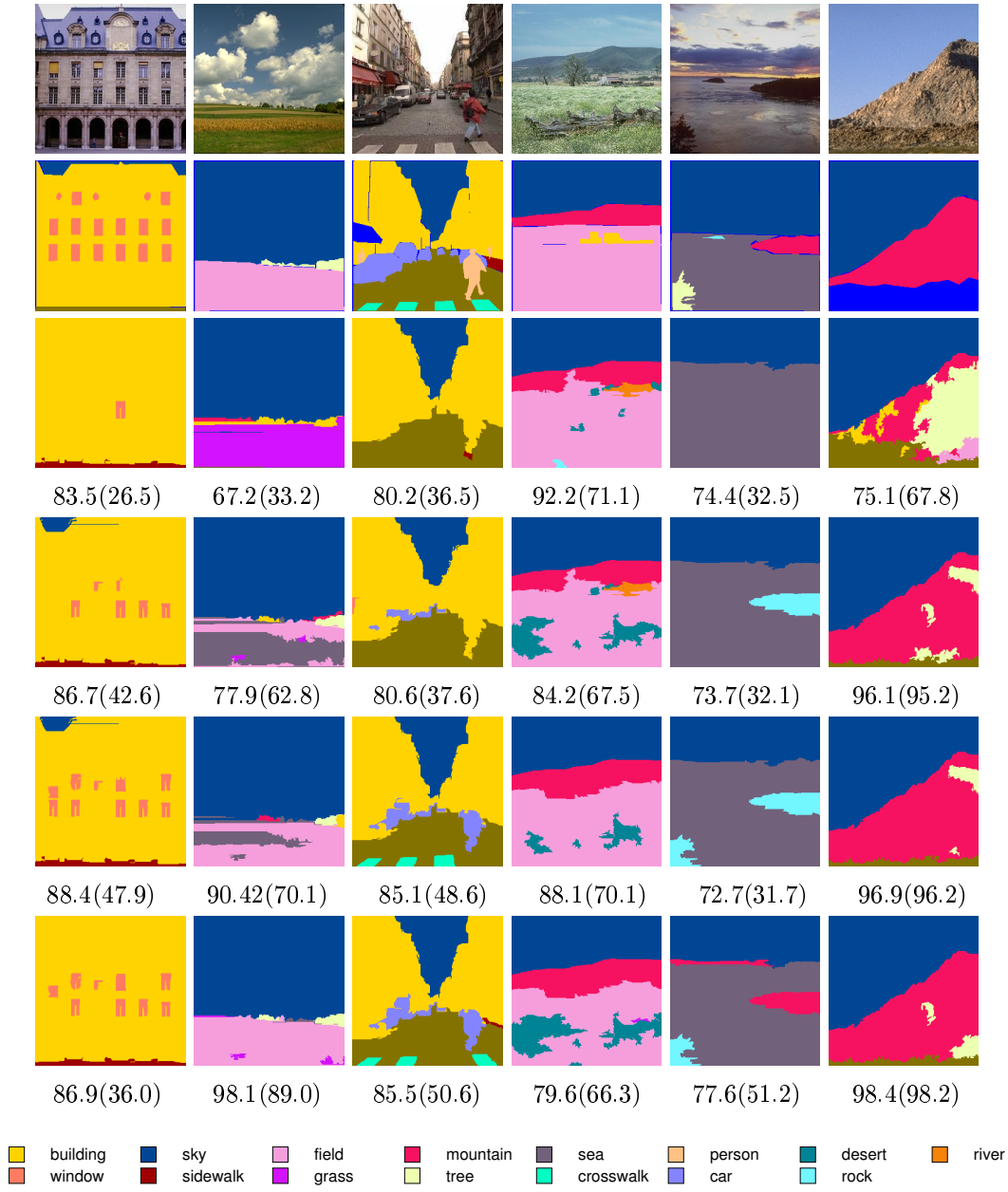


FIGURE 5.9 – Comparison of the results produced by our method with different values of subset size T. $\mathbf{1}^{\text{st}}$ row : test images ; $\mathbf{2}^{\text{nd}}$ row : ground-truth ; $\mathbf{3}^{\text{rd}}$ row : results with $T = 20$; $\mathbf{4}^{\text{th}}$ row : results with $T = 60$; $\mathbf{5}^{\text{th}}$ row : results with $T = 100$; $\mathbf{6}^{\text{th}}$ row : results with $T = 140$. In almost of the cases, adding more training images leads to improved labellings. The only exception is at the example in the 4^{th} column. The labeling results are improved in half of the examples (columns $2^{\text{nd}}, 5^{\text{th}}, 6^{\text{th}}$) when increasing T from 100 to 140.

5.6.2 Analysis of the Smoothness Regularizer

As shown in Chapter 3, the smoothness regularizer is responsible for label diffusion. The goodness of the labeling also depends on this regularizer, which in turn depends on the graph Laplacian $\mathbf{L}$ and the smoothness coefficient β. Given $\mathbf{L}$ fixed, an appropriate value of β, i.e., $\beta = 10^{-2}$, can boost classification performance ; however, larger values of β, i.e., $\beta = 1$, degrades the performance because the smoothness regularizer will favor more frequent labels than rare ones. As shown in Fig. 5.7(b) and Table. 5.1 the optimal value of β is fixed at $\beta = 10^{-2}$ whose per-pixel and the per-label rates are 77.1% and 27.0% respectively. This optimal value of β is kept for subsequent experiments.

5.6.3 Analysis of the Semantic Context

Unlabeled superpixels extracted from the test image must have certain contextual relationships. Semantic context aims to exploit prior information of such relationships from the database and apply it into the test image. In particular, our semantic regularizer focuses on invoking co-occurrences between adjacent objects. Based on the likelihood table computed in Section 5.3.2, we investigate how the overall performance is improved with the presence of semantic regularization. The semantic context with $\gamma = 10^{-4}$ increases the per-label rate from 27.0% to 28.3%. A larger value of γ, i.e., $\gamma = 10^{-3}$, increases the per-label rate to 28.9% while decreases the per-pixel rate from 77.2% to 76.9%. This rate drastically drops as γ increases further (see Fig. 5.7(c)). Some examples shown in Fig 5.10 demonstrate how the presence of the semantic context helps correcting false predictions (the 1^{st}, 2^{nd}, 3^{rd}, and 6^{th} columns of the 5^{th} row). In order to understand the working mechanism of the semantic regularizer, it is useful to observe in Fig. 5.11(a) the changes of classification responses as the semantic context is involved more into the interpretation process ; by increasing the coefficient γ from 10^{-5} to 10^{-4} and 10^{-3}, the labeling result is improved (see how the mislabeling of *tree*, *sky*, and *pole* is corrected in Fig. 5.11(a)).

5.6.4 Analysis of the Position Context

Having the same goal as of the semantic regularizer but exploiting contextual information in a different way, the position regularizer handles label co-occurrences in terms of their absolute positions in the test image. After estimating the co-occurrence statistics between every possible label pair and storing them in the likelihood matrices $\{\mathbf{P}^{cc'}\}$, we test the effect of the position context with several values of the coefficient γ. From Fig. 5.7(d) we observe that the per-label and per-pixel rates are improved up to 29.1% and 77% when $\gamma = 10^{-5}$. Compared with the semantic regularizer, the position regularizer is slightly better at the per-label rate (29.1% versus 28.9%) but slightly worse at the per-pixel rate (77% versus 77.2%). A noticeable weakness of the position context is the inflexible grid-based model. For example, the horizon line may be positioned from very high (in some images) to very low

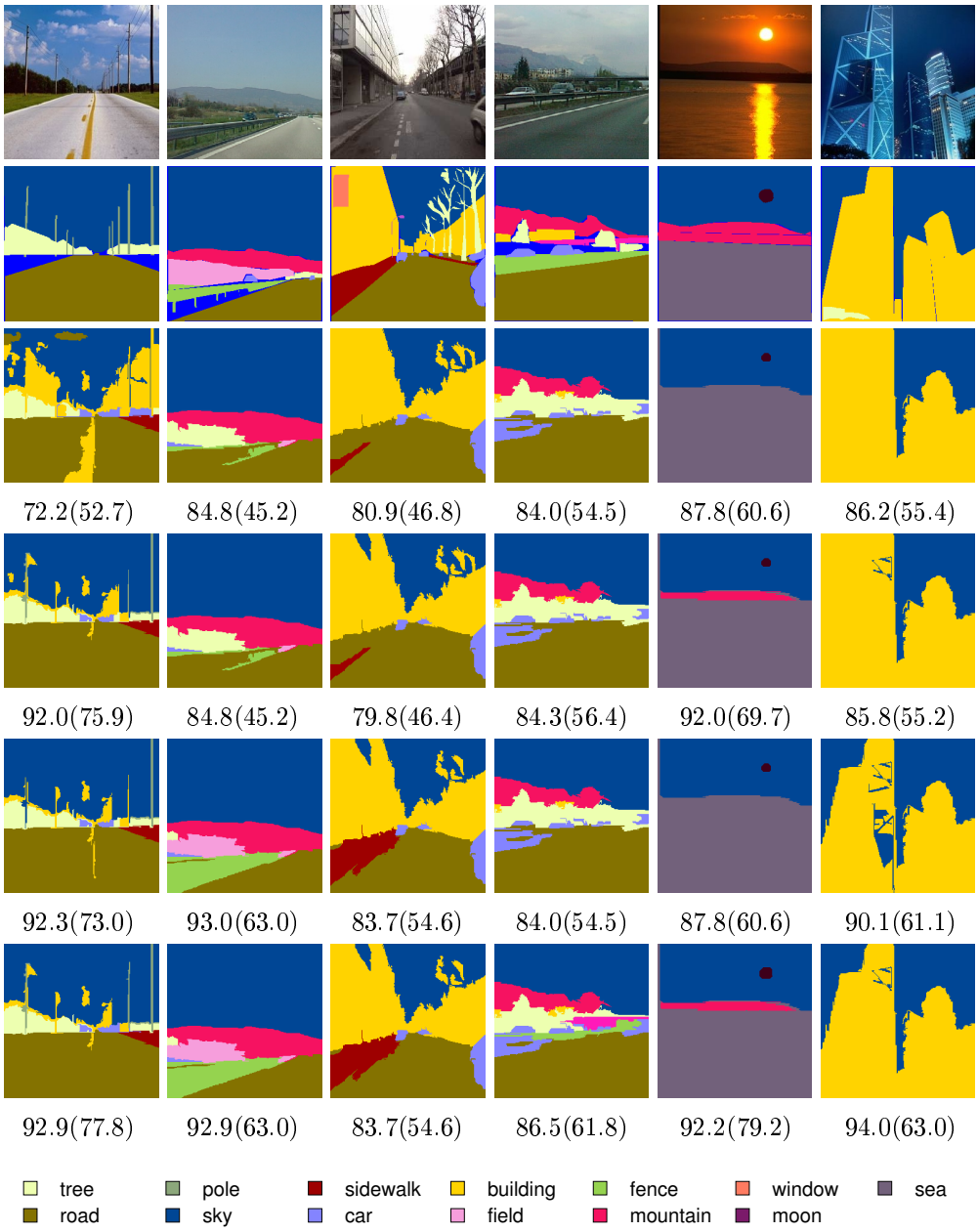


FIGURE 5.10 – Examples comparing the effects of context regularization. **1**st row : Test images ; **2**nd row : Ground-truths ; **3**rd row : the basic results ; **4**th row : with position context ; **5**th row : with semantic context ; **6**th row : with both position and semantic contexts and spatial smoothing. The two context regularizers help each other : some false predictions are fixed by either one of the two regularizers. When combined together they provide better interpretation results compared to individual contexts.

(in other images). In normal cases where the horizon line is located at the middle of the image, the position context is useful (see the 1^{st} and 5^{th} columns of the 4^{th} row in Fig. 5.10). This may explain why the position context is slightly worse than the semantic one in terms of the per-pixel rate. For the per-label rate, the position context is better than the semantic one, especially for the object labels whose positions are often fixed, for example *sun*, *moon*, *pole*, and *sidewalk* (compare labeling results between Fig. 5.11(a) and Fig. 5.11(b)). Differences between the two contexts are further depicted in Fig. 5.12.

5.6.5 Analysis of the Combination of Contexts

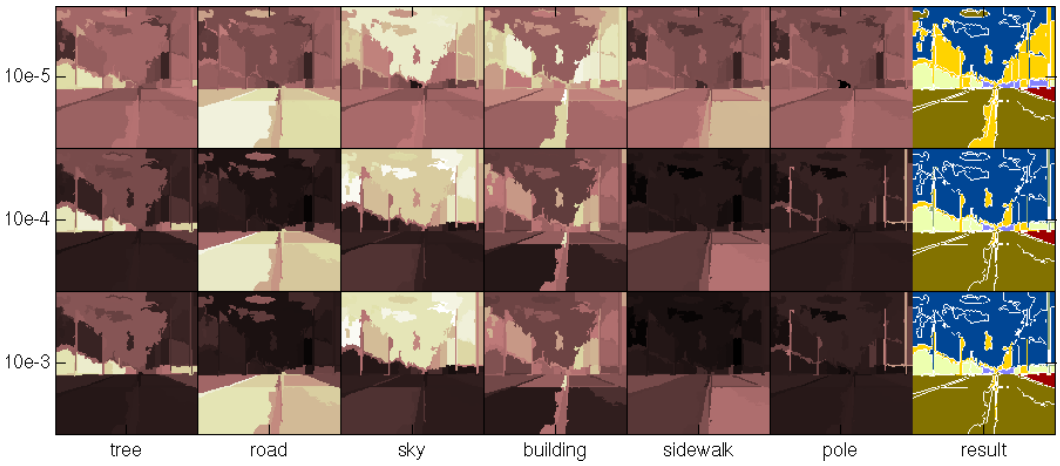
We wonder whether a combination of semantic and position contexts will improve further the performance. In order to test this, the context regularizer (5.13) is duplicated into two versions whose coefficients γ_s and γ_p are used to emphasize the importance of the semantic and position regularizers respectively. Update rules for the optimization algorithm are modified in the same way. While the semantic regularizer is good at capturing local relationships, the position regularizer is likely to capture semantic layout at the image level; we expect that the two factors are complementary. Fig. 5.7(e) shows empirical evidences that support our expectation. We test with two settings of γ_s (10^{-3} and 10^{-4} correspond to red and blue dashed lines in Fig. refchap4 :fig :analyzecombine) and for each of the value of γ_s we steadily increase the value of γ_p from 10^{-7} to 10^{-4}. We observe that both of the settings obtain their peaks with $\gamma_p = 10^{-6}$. That value of γ_p makes the position regularizer obtaining its best performance. Compared with the improvements made by the individual context regularizers (the **x** dots in Figs. 5.7(d) and 5.7(c)), the combined context benefits both per-pixel and per-class rates.

5.6.6 Analysis of the Spatial Smoothing

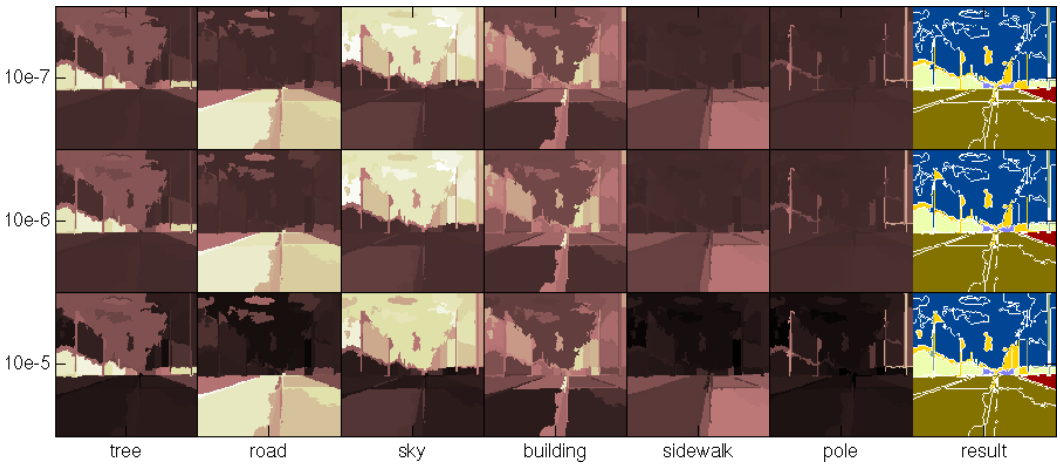
As mentioned in Section 5.3.4, spatial smoothing addresses the fact that adjacent superpixels of an image are likely to have similar labels. The smoothness term (4.3) associates superpixels by edges if they are visually similar; however, it may not guarantee that adjacent superpixels of the same object in the test image are connected. That is why a spatial smoothing term is necessary. We test this term with three settings : semantic context, position context, and the combined context.

For the position context with $\gamma_p = 10^{-5}$ (γ_s is set to zero), the spatial smoothing coefficient helps increasing slightly per-label rate but with a small degradation of per-pixel rate (see the red dashed line in Fig. 5.10).

Similarly for the semantic context ($\gamma_s = 10^{-3}, \gamma_p = 0$), the per-label rate is improved from 28.9% to 30.5% as we introduce the spatial smoothing term at $\theta = 10^{-3}$; however, its per-pixel rate is slightly degraded from 76.9% to 76.6% (the blue dashed line in Fig. 5.7(f)). Fortunately, the per-pixel and per-label rates recover to 76.8% and 30.5% respectively if we continue to increase θ up to $\theta = 10^{-3}$. Increasing more on the spatial smoothing term, for example $\theta = 10^{-2}$, does not gain but reduces

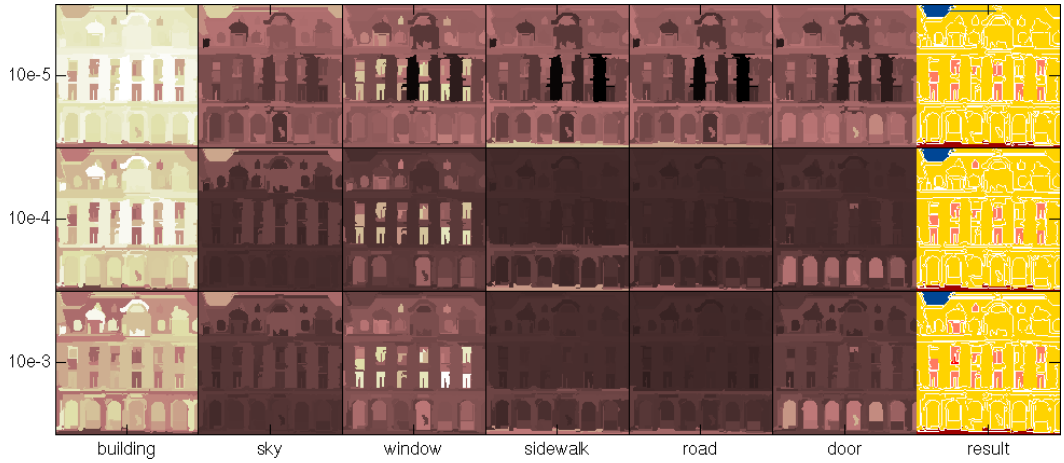


(a) The prediction maps conveying how semantic context correct false predictions. By increasing the coefficient γ to an appropriate value, i.e., $\gamma = 10^{-4}$, the labeling responses are improved (see the contrast improvements in some prediction maps when increasing γ from 10^{-5} to 10^{-3}). For example, *tree* objects become more visible with $\gamma = 10^{-4}$ than with $\gamma = 10^{-5}$; clouds in the *sky*, which is misled with *road* when $\gamma = 10^{-5}$, are now labeled correctly with $\gamma = 10^{-4}$ or $\gamma = 10^{-3}$; similarly, the lane marker at the center of the *road*, which are misled with *building*, are partly corrected with the semantic context; interestingly, *pole* objects emerge clearly from the background when increasing γ.

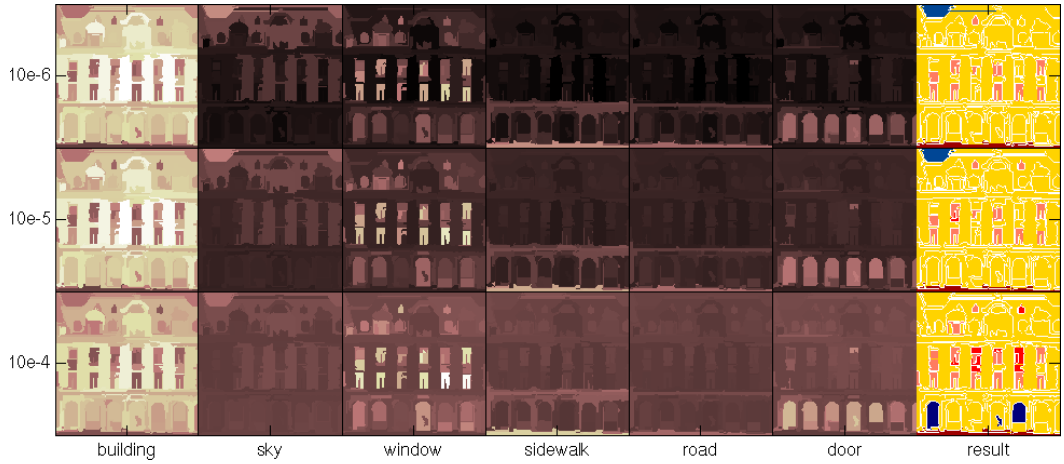


(b) The effect of the position context is generally different from the semantic context. Because of the grid-based co-occurrence model, they are visible in the maps of *road* and *sky* horizon lines dividing the image into two halves; above the line is the *sky* and below the line is the *road*. In fact the position context is useful in this case; and in general, it is useful for landscape scenes.Notice how the rare label *pole* is prioritized as γ is increased.

FIGURE 5.11 – How the use of contextual regularizers affects final predictions. The prediction maps of six labels (*tree*, *road*, *sky*, *building*, *sidewalk*, and *pole*) are shown for every value of the contextual coefficient γ (vertical axes). Prediction maps in (a) are the results of the semantic context while the maps in (b) are the results of the position context.



(a) By increasing γ, more *window* labels appear in the final prediction. Notice that there are black holes (which are strong responses of the label *building*) in the prediction maps of *window*, *sidewalk*, *road*, and *door*. However, just the *window* superpixels, which are adjacent to those black holes, are emphasized. The support from *building* for *window* is evenly clear as γ is increased.



(b) With $\gamma = 10^{-6}$, the response map of *sky* is stronger at the top (lighter color), while that of *window* is stronger at the image center and those of *sidewalk*, *road*, and *door* maps are at the bottom. As we increase the value of γ, some *door* labels appear in the final prediction. This does not happen with the semantic context (see figure (a)).

FIGURE 5.12 – Compare the difference between semantic and position regularizations.

the per-label rate.

With the combined context configuration, spatial smoothness coefficient $\theta = 10^{-3}$ leads to the improvement of both per-pixel and per-label rates from 76.5% to 77% and from 30.9% to 31.2% respectively. This is also the highest performance obtained by our method with the SiftFlow dataset (see Table 5.1).

		Per-pixel	Per-label
Related works	[Russell 2007a]	76.7	-
	[Tighe 2010]	76.9	29.4
	[Tighe 2013]	77.0	30.1
	[Farabet 2012]	78.5	29.6
	[Eigen 2012b]	77.1	32.5
	[Singh 2013]	79.2	33.8
	[Myeong 2012]	77.1	32.3
Our method	no context	77.1	27.0
	semantic	77.2	28.3
	position	77.0	29.1
	smoothed semantic	77.1	28.8
	smoothed position	77.0	29.1
	semantic + position	77.3	29.0
	smoothed semantic + position	77.0	31.2

TABLE 5.1 – Comparison between our method (best configurations) and related works.

5.6.7 Comparison with Related Works

Except the scene alignment work of [Russell 2007a], the other methods in the Table 5.1 adopt a similar framework of scene interpretation. A test image is used to query a small set of labeled images ; this set is the training data for a machine learning algorithm such as MRF or CRF. Label information is propagated – via the inference process of the learning algorithm – from the training data to the test ones, which are pixels or superpixels of the test image. Although our method has a different methodology perspective, the performance of our method on the SiftFlow dataset is effective and is among the state of the art. In particular, our method is better compared to scene alignment [Russell 2007a] and SuperParsing [Tighe 2010, Tighe 2013]. The success of our method is due to our context regularizers and the label-based subset retrieval heuristic which is adopted from [Singh 2013]. The former factor helps improving the per-label rate from 27% to 31.2% ; the latter factor boosts the per-pixel rate from 75% to 77.1% and the per-label rate from 26% to 27%. Compared to [Farabet 2012] where they use a convolutional network for feature learning, our method is better in terms of the per-label rate but worse in terms of the per-pixel rate. Compared to [Eigen 2012b] and [Myeong 2012], our method is slightly inferior in terms of the per-label rate. According to [Eigen 2012b], they adopt distance learning in order to reweight the importance of visual descriptors with respect to every object label ; visual descriptors are not reweighted in our method. According to [Myeong 2012], they use a link prediction method in data mining in order to infer contextual links between superpixels in the test image ; although their method turns out to be more effective than ours, it is computationally expensive.

5.7 Summary

In this chapter we introduced extensions of our transductive kernel learning method for scene interpretation. Based on the multi-class formulation developed in Chapter 4, we incrementally design regularizers that account for contextual relationships between objects in scene. Two regularizers are introduced which exploit the co-occurrence between object labels in order to obtain a more consistent labeling of test images. Based on the subset retrieval step introduced by [Tighe 2010] and improved in [Singh 2013], our efficient optimization algorithm achieves competitive results with the SiftFlow dataset. Good empirical results demonstrate that our method effectively tackles the scene interpretation problem.

Chapitre 6

Transductive Subspace Learning for Image Search

In this chapter we introduce a novel transductive approach in order to learn semantic low-dimensional representations for images. Our goal is to use this representation for query design, interactive image search, semantic-driven image representation, and ranking. This chapter is our third main contribution of the thesis.

Recall that with transductive kernel map learning in Chapter 4, we have applied it to image annotation, which is an important building block for keyword-based image search. So in this new chapter, the image search problem is revisited again but solved using a different method. The idea of our method is to learn a low-dimensional embedding from an input space to a well-defined semantic space. Our method is supported by the fact that an interactive and semantic-based querying tool is more effective and efficient for users to express their mental queries. Furthermore, our method is scalable and useful in situations where annotated data is scarce. Extensive empirical results with satellite images and generic scene images demonstrate competitive performance of our method with respect to state-of-the-art.

Parts of this chapter were mentioned in the followings papers :

1. Phong Vo, Hichem Sahbi, *Semantic Subspace Learning for Mental Search in Satellite Image*, IGARSS, Australia, 2013.
2. Phong Vo, Hichem Sahbi, *Spacious : An Interactive Mental Search Interface*, ACM SIGIR, Ireland, 2013.

6.1 Introduction

Social networks and Web 2.0 have opened tremendous opportunities as well as challenges for multimedia creation and sharing. When more and more people use Internet, information search is very demanding. An early form of information search is text search. After a time, it has been commercialized as search engines such as Yahoo and Alta Vista. Google and Bing are nowadays masters in text search technology. The success story of text search, however, has not yet been fully transposed to multimedia data. This is a very challenging problem for machine learning and computer vision methods because it is still difficult to build intelligent machines in order to comprehend visual data.

In seeking for intelligent image search algorithms, transferring techniques of text search into visual data is a mainstream. The principle is straightforward : images must be annotated by text, that is to associate content of an image to corresponding keywords. Once annotated, images are ready to be retrieved as if they were text. The key advantage of this approach lies in the conciseness and diversity of natural languages. Its disadvantage, however, is threefold. First, a user may find it difficult to use text in order to express his mental picture. Some concepts are described better with visual content than by text. Second, representing retrieved results is not always intuitive. In particular, search engines such as Google and Bing simply list retrieved images in flat web pages ; this representation is inappropriate to multi-dimensional visual data. Third, multimedia data nowadays are too massive to be annotated manually and current automatic image annotation approaches are not sufficiently mature. Hence, we are searching for alternatives in this work.

Query-by-example [Heesch 2008, Rubner 2001, Kovashka 2012, Ferecatu 2007] is a pioneer approach in using visual example for image retrieval. Its idea is to replace keyword-based queries with some images that are similar to the mental picture of the user. With this query approach, annotation is not necessary ; the system uses a proxy image as a visual pattern to search for other images in the database ; the most similar ones are the retrieved results (see Fig. 6.1). Image matching is done by comparing images using low-level descriptors such as color, texture, and shape. However, query-by-example lacks of high-level semantics. Since current vision algorithms are not be able to generalize low-level features to abstract concepts, low-level features alone cannot disambiguate images including visual objects with strong variability. Furthermore, query images may not be easy to find. Thus choosing an appropriate example is not a trivial task. In other cases, the mental picture is not well described by those examples.

Relevance feedback [Zhou 2003c, Ferecatu 2007, Gosselin 2008] tries to solve this problem using knowledge obtained from the user. It is an iterative querying process in which examples similar to the user's mental target are chosen at every iteration. In order to give feedbacks, the user must select a few representative images, which are examples used for the next iteration ; these representatives must contain both positive and negative samples. Positive samples mean that the user wants to see more images like these in the next round ; negative samples mean that he does not

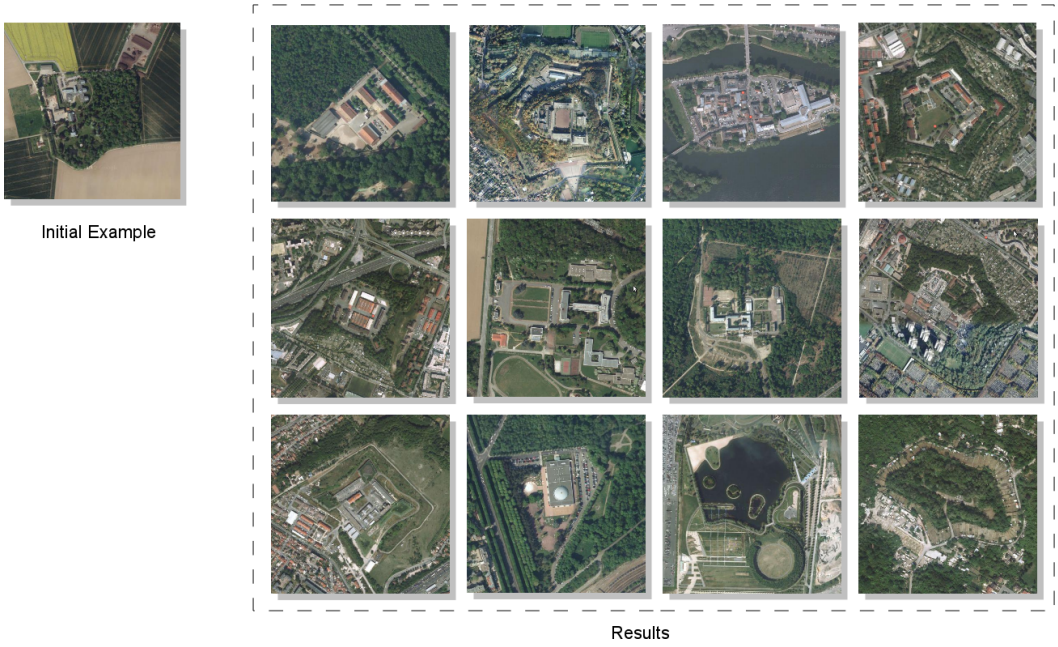


FIGURE 6.1 – Query-by-example requires from the user an image example which is similar to his mental target. This example is then used to match against images in the database ; the most similar ones are used as the retrieved result. In the example above, the mental target is to find "Vauban fortress." Since the user does not have any fortress image, he uses another one which is marginally similar to his target. The retrieved result consists of both true positives and false alarms.

want to see such images anymore. These images correspond to user's feedback (see Fig. 6.2). The feedback process may not be stopped until the user finds his image of interest, otherwise the user cancels and restarts the search process.

The advantage of relevance feedback is due to not only the use of many examples but also the selection of those examples. Since the user has to judge the relevance of those examples before choosing them as representatives, semantics is implicitly induced into the search process. This depends on the goodness of image matching techniques and also examples of earlier feedbacks. However, the process may spend many iterations before reaching the target.

An alternative, known as query-by-semantics, maps images from their input space to a semantic space. We are interested in a family of machine learning methods, for example [Siddiquie 2011, Russakovsky 2010b, Kumar 2009], that uses ranking approaches in order to define such a mapping. These learning to rank methods can be classified into two approaches :

– The first one [Joachims 2002b, Siddiquie 2011] applies one-vs-rest classifiers in order to learn ranking functions. For every semantic, a classifier such as SVM-rank [Joachims 2002b] is learned from labeled data of image pairs. The learning process takes into account relative comparison of semantic abundance

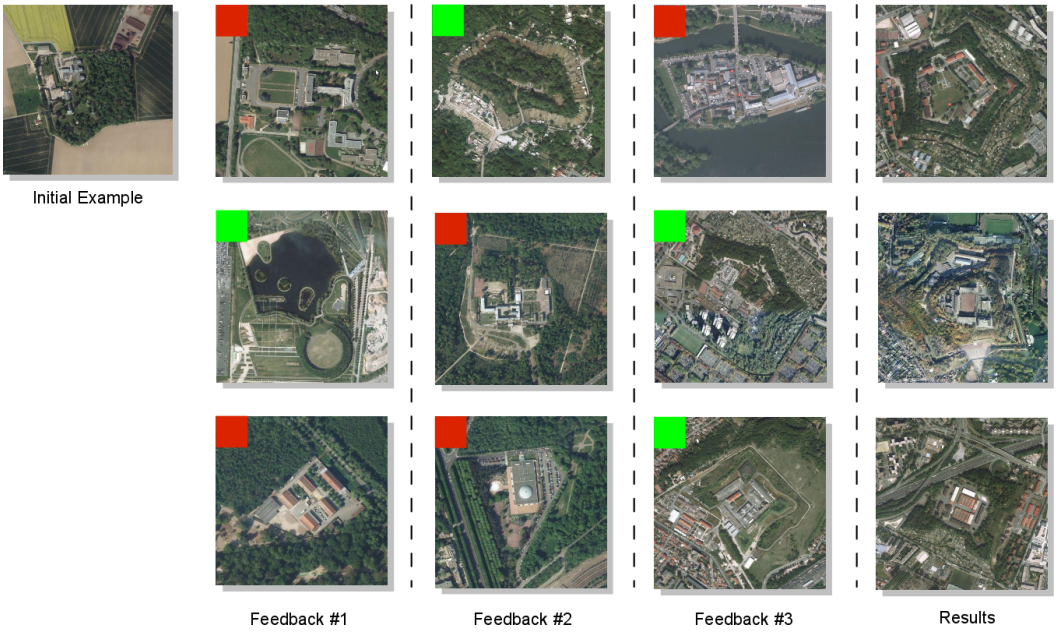
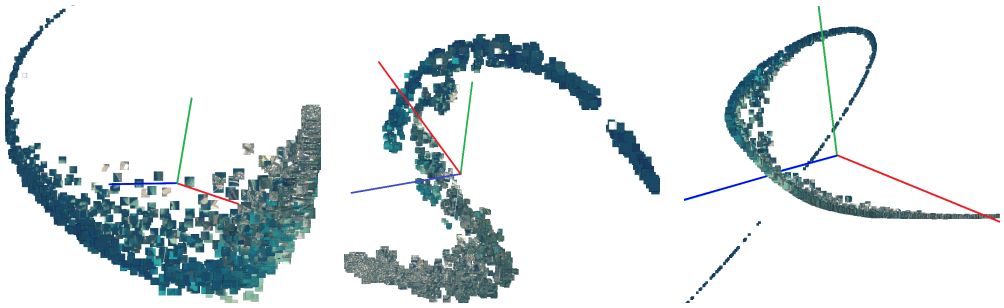


FIGURE 6.2 – Since a single example may not be enough to characterize the mental picture of the user, the solution of relevance feedback is to let the user refine his mental query based on interactive feedback mechanism. From the retrieved result at every iteration, the user gives extra image examples which are annotated as relevant or irrelevant to the target image. In the example above, three examples are used to give feedbacks at every iteration. The ones similar to the mental target are marked as green, otherwise marked as red. At the fourth iteration, more correct images appear.

between two images in a pair. Learned ranking functions associate images to relevance scores such that the scores between two images of a training pair must respect their relative semantic abundances. If every ranking function is considered as a basis, a full span of the basis defines a semantic space and this space is interpretable to the user.

– The second approach is based on manifold assumption, which was introduced in Chapter 2. Its idea is to use smoothness assumption in order to enforce ranking functions to preserve the orders of the training data [Liu 2011b, Cai 2007, Lin 2005, Zhou 2003b, He 2004].

In general, ranking methods somewhat relate to automatic image annotation due to their similar goals in assigning high-level features to images. However, automatic image annotation aims to tag as many as possible visual objects present into an images ; image ranking focuses on ranking images based on amounts of semantic abundances. Once an image is ranked, we can measure how much or less a semantic appears in that image ; furthermore it can be compared to another ranked image. This is beneficial to the image search problem because we can use simple comparison



(a) KernelPCA (b) t-SNE (c) Laplacian Eigenmaps

FIGURE 6.3 – The visualizations of satellite image patches using nonlinear dimensionality reduction techniques. Despite of being able to discover the dynamic of data (gradual changes of visual appearances from urban to sea), those techniques fail to interpret the semantic of dynamic with respect to subspace coordinates. In other word, the dimensions of the subspace are not necessarily associated with any semantics ; equivalently, coordinate values of a point in the target subspace cannot tell us anything about which semantics are more likely to appear at that position.

operators for querying.

The comparative advantage of query-by-semantics with respect to query-by-example is that the semantic-based representation is more interpretable. The rest problem is how to design an intuitive querying interface. Some works [Kumar 2009, Siddiquie 2011] propose to use keywords (query-by-text) in order to search for semantic content. We do not adopt this approach because text-based methods are restrictive. For instance, it is difficult when using keywords to name abstract visual concepts or express the amount of semantic abundance of targeted images. Our goal therefore is to find an alternative which helps expressing semantics into queries.

Inspired by the problem of selecting initial examples for a query, we wonder how to quickly spot and pick them from an image database. This is novel because current methods just focus on extracting representatives of database [Fauqueur 2006, Ferecatu 2007]. For databases whose sizes range from small to moderate, such methods are sufficient, i.e., using sampling techniques to obtain representatives. At larger scales, however, sampling techniques are not effective. We think that this issue can be solved using data visualization techniques [Rubner 2001, Heesch 2008, Schaefer 2010] whose core parts are based on nonlinear dimensionality reduction algorithms (ISOMAP [Tenenbaum 2000], Locally Linear Embedding [Roweis 2000], Laplacian Eigenmaps [Belkin 2001], etc.). Via transforming the data from a high-dimensional space to a low-dimensional space, data visualization techniques can help the user to perceive an overview of the database. Thus the user can build his cognitive map of the dynamic of the data which serves as a mean for his mental picture to be expressed [Spence 1999]. Based on such a cognitive map, the user interacts with a visualization system and seeks for his mental target without requiring any query example. The main challenge of dimensionality reduction methods, however, is the

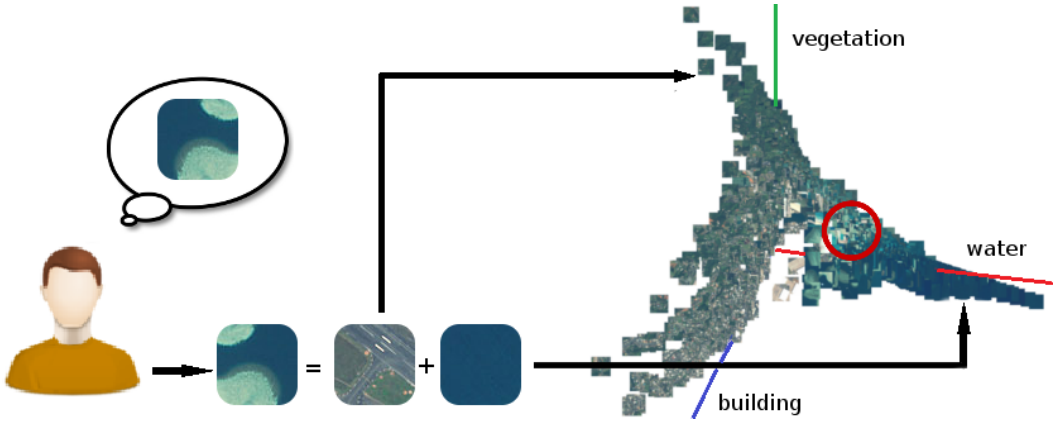


FIGURE 6.4 – The processing pipeline of mental querying : (i) the user has a mental target, says a coastal image ; (ii) after observing the visualization of the database, the user knows how to interpret his mental target as a combination of the semantics available in the database ; (iii) based on the user's judgement on the amount of semantic abundances, the user will examine the corresponding location in the semantic space. Repeating the process several times, and combining with systematic navigation, the user is expected to find his target.

lack of semantic as these methods are unsupervised. As a result, the low-dimensional representation is not associated to any semantic, which makes the search process difficult to follow (see Fig. 6.3).

As the third contribution, we introduce a novel algorithm for semantic subspace learning, which learns a semantic representation of the data. The algorithm is designed using a novel principle, that unmixes semantics from images and maps them from an initial ambient space (related to low level visual features including texture, color and shape) to an output space spanned by a well defined semantic basis. An arbitrary image is given a unique position in the new representation ; the coordinates of this position will mention the amount of semantic abundances contained in the image. As will be shown, we cast this problem as convex quadratic programming (QP) optimization, constrained in a simplex spanned by few pure semantic endmembers. The advantages of the proposed approach are threefold ; firstly, it significantly reduces the dimensionality of the input space (which is difficult to explore/visualize) ; secondly, it learns features which are semantically interpretable, i.e., their values are highly correlated with the defined semantics ; thirdly, it provides global access to the data in contrast to relevance feedback where user judgment is limited to the retrieved results. Thereby, searching for a mental target, with our model, simply reduces to scanning and targeting data according to their coordinates in the learned semantic subspace.

The rest of this chapter is organized as follows. Section 6.2 presents semantic subspace learning. The next Section 6.3 explains how to optimize the proposed criterion

for small and large-scale databases. Next in Section 6.4 are our experiments of data visualization and search for satellite and generic images. A software implementing our idea is presented in Section 6.5. Quantitative evaluations of image ranking and retrieval with relevance feedback are investigated in Section 6.6. We conclude the chapter in Section 6.7.

6.2 Method

6.2.1 Mathematical notation

We consider the following notation : $\mathbf{A}_k$ (resp. $\mathbf{A}_{k\cdot}$) denotes the k^{th} column (resp. row) of a given matrix $\mathbf{A}$ while $\mathbf{A}_{ki}$ denotes the k^{th} entry of its i^{th} column. In particular, $\mathbf{I}_n$ denotes an identity matrix of size $n \times n$; $\mathbf{1}_{n\times n}$ and $\mathbf{0}_{n\times m}$ denotes all one's and all zero's matrices respectively. We also denote $\mathbf{A}'$ as transpose of $\mathbf{A}$ and $\mathbf{A} \succeq 0$ if $\mathbf{A}_{ij} \geq 0, \forall i, j$. Consider $a \subseteq \{1, \ldots, m\}$, $b \subseteq \{1, \ldots, n\}$, the sub-matrix of $\mathbf{A} \in \mathbb{R}^{n\times m}$ containing entries $\mathbf{A}_{ij}$ in which $i \in a$ and $j \in b$ is denoted as $[\mathbf{A}]_{ab}$. Two matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{p\times q}$ are vertically stacked as a block matrix $[\mathbf{A}; \mathbf{B}] \in \mathbb{R}^{(n+p)\times m}$ iff $m = q$ and horizontally concatenated as a block matrix $[\mathbf{A}\ \ \mathbf{B}] \in \mathbb{R}^{n\times(m+q)}$ iff $n = p$. The trace $\text{tr}(\mathbf{A})$ of the square matrix $\mathbf{A}$ equals to the sum of diagonal elements of $\mathbf{A}$. For vector notations, we denote $\text{diag}(\mathbf{a}) = \mathbf{A}$ as a diagonal matrix of size $n \times n$ in which its main diagonal entries equal to those of the vector $\mathbf{a} \in \mathbb{R}^n$; $\mathbf{1}_n$ and $\mathbf{0}_n$ denotes all one's and all zero's vectors respectively. Other notations if necessary will be defined later.

6.2.2 Semantic Subspace Learning

We are interested in learning a particular low dimensional subspace. The underlying assumption is : the probability distribution generating the input data admits a density with respect to the canonical measure on a sub-manifold of the Euclidean input space. The goal of our method is to define a mapping that preserves the local distances while capturing a global topology. The latter is defined by the dynamic of variation of data (through different intrinsic dimensions) which should be consistent with the semantics assigned to these dimensions.

Let $\mathbf{X} \in \mathbb{R}^{D\times m}$ be a matrix of m input data points with feature vectors $\mathbf{X}_i$'s and $\mathbf{Y} \in [0, 1]^{K\times m}$ the underlying membership matrix of m data points whose ℓ first elements are labeled ; here K corresponds to the number of semantics. A given entry $\mathbf{Y}_{ki}$ of a membership vector $\mathbf{Y}_i$ in which $i \leq \ell$ is strictly positive iff the k^{th} semantic is present into $\mathbf{X}_i$; since the rest $(m - \ell)$ data points are unlabeled, their membership vectors $\mathbf{y}_i$'s equal zero. When only one entry of $\mathbf{Y}_i$ is positive and also equal to 1 and the rest equals zero, then $\mathbf{Y}_i$ is referred to as *endmember*. Notice that, per construction, the endmembers of different semantics as well as the underlying semantics should be mutually uncorrelated. Our algorithm is proposed based on the *endmember condition* which states that the training data (first ℓ data points) must be endmembers. Later in this chapter we will show that the satisfaction of this

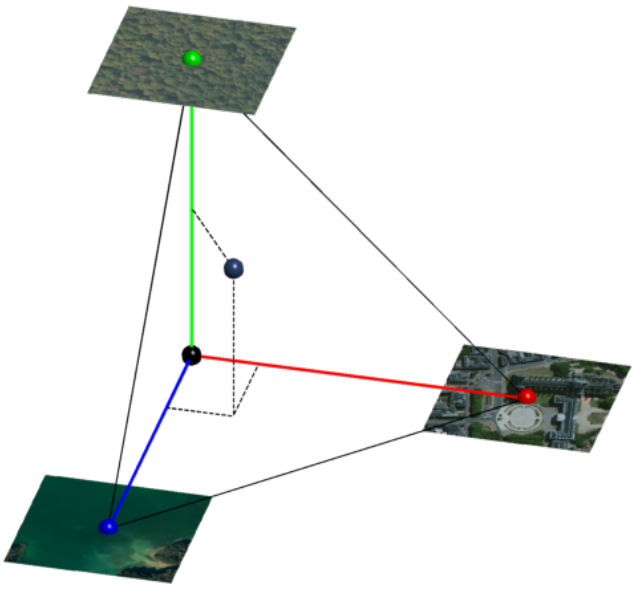


FIGURE 6.5 – Our proposed method learns an embedding of the data which resides in a subspace whose coordinates are associated with a set of predefined semantics. In the figure above, our subspace is in $\mathbb{R}^3$ whose axes are vegetation, building, water. Illustrated at the corresponding axes are the training examples which we call *endmembers* in our context.

condition depends on databases and that the condition is not required to be strictly satisfied.

Our goal is to learn an embedding $\mathbf{\Phi} \in [0,1]^{K \times m}$ which is in fact the membership matrix of both ℓ labeled and $(m-\ell)$ unlabeled data. Each entry $\mathbf{\Phi}_{ki}$ corresponds to a mapping of $\mathbf{X}_i$ into the k^{th} semantic dimension ; a high value of $\mathbf{\Phi}_{ki}$ indicates that the k^{th} semantic is present into $\mathbf{X}_i$ with a high probability and vice-versa. In order to find $\mathbf{\Phi}$, we introduce the following optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{\Phi} \in \mathbb{S}} & \frac{1}{2}\text{tr}\left(\mathbf{\Phi L \Phi'}\right) \\ \text{s.t} & \mathbf{\Phi C} = \mathbf{Y} \end{array} \tag{6.1}$$

In the above objective function, elements in $\mathbf{\Phi}$ are taken from a unit $(K-1)$-simplex, that belongs to the positive orthant, and spanned by the canonical basis of $\mathbb{R}^K$, i.e., $\mathbb{S} = \{\mathbf{\Phi} \in \mathbb{R}^{K \times m},\ \mathbf{\Phi} \succeq 0,\ \mathbf{\Phi'} \mathbf{1}_K = \mathbf{1}_m\}$. This simplex condition $\mathbf{\Phi} \in \mathbb{S}$ is essential ; given an image, this model measures the abundance of each semantic into that image so these abundances should be positive and their sum equal to 1. For instance, if a photo contains more green space (field, grass, tree), then the image area for other semantics such as building, street, sky, must be reduced proportionally.

The only term in (6.1) is a regularizer that ensures similar embedding for neigh-

boring data in $\mathbf{X}$

$$\begin{aligned} \tfrac{1}{2}\mathrm{tr}\left(\mathbf{\Phi L \Phi}'\right) &\triangleq \tfrac{1}{2}\mathrm{tr}\left(\mathbf{\Phi D \Phi}'\right) - \tfrac{1}{2}\mathrm{tr}\left(\mathbf{\Phi W \Phi}'\right) \\ &= \sum_{i=1}^{m} \left\|\mathbf{\Phi}_i\right\|^2 \mathbf{D}_{ii} - \sum_{i,j=1}^{m} \mathbf{\Phi}_i' \mathbf{\Phi}_j \mathbf{W}_{ij} \\ &= \sum_{i,j=1}^{m} \left\|\mathbf{\Phi}_i - \mathbf{\Phi}_j\right\|^2 \mathbf{W}_{ij} \end{aligned} \tag{6.2}$$

where $\mathbf{W} \in \mathbb{R}^{m \times m}$ is a symmetric similarity matrix and $\mathbf{L}$ the normalized graph Laplacian [Chung 1997]. The equivalence between $\mathbf{L}$ and $\mathbf{W}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D} = \mathrm{diag}(\mathbf{W}\mathbf{1}_{\mathrm{m}})$. In our work, we use the normalized version of graph Laplacian, in which the affinity matrix $\mathbf{W}$ is normalized such that the sum of edge weights of a vertex is equal to 1, i.e., $\mathbf{L} \leftarrow \mathbf{I}_m - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. The similarity $\mathbf{W}_{ij}$ between two feature vectors $\mathbf{X}_i$ and $\mathbf{X}_j$ is computed based on the Gaussian RBF function considering the k-nearest neighbors set $\mathcal{N}_i$ of $\mathbf{X}_i$,

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|_2^2}{\sigma^2}\right) & \text{iff } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

in which $\sigma = \sum_{ij} \|\mathbf{X}_i - \mathbf{X}_j\|_2^2 / (km)$ is the average distance between nodes of the graph. The optimization will stress more on the smoothness between similar patches (significant weight $\mathbf{W}_{ij}$'s), while has no effect with unconnected patches. If two points are close to each other in the embedded space, they must be nearby to each other in the embedding space. In other word, if two patches are visually similar to each other, i.e., the Euclidean distance between two feature vectors is small, then they should be placed closely in the semantic subspace (see Fig. 6.6). We expect to maintain a smooth geometrical structure of the data in the semantic space.

The constraint $\mathbf{\Phi C} = \mathbf{Y}$, where $\mathbf{C} = [\mathbf{I}_\ell; \mathbf{0}_{(m-\ell)\times\ell}]$, guarantees that the embedding $\{\mathbf{\Phi}\}_i$ of endmembers coincides with the vertices of the simplex.

6.3 Optimization

Our formulation in (6.1) is a constrained quadratic programming problem. We convert it to the canonical form by vectorizing $\mathbf{\Phi}$ as follows

$$\begin{aligned} \min_{\boldsymbol{\alpha} \succeq 0} \quad & \tfrac{1}{2}\boldsymbol{\alpha}' \mathbf{H} \boldsymbol{\alpha} \\ \text{s.t} \quad & \mathbf{A}\boldsymbol{\alpha} = \mathbf{1}_{(m-\ell)} \\ & \mathbf{B}\boldsymbol{\alpha} = \boldsymbol{\xi} \end{aligned} \tag{6.4}$$

in which $\boldsymbol{\alpha} = \mathrm{vec}\left(\mathbf{\Phi}\right)$, $\boldsymbol{\xi} = \mathrm{vec}\left(\mathbf{Y}\right)$, and $\mathrm{vec}(\cdot)$ denotes an operator that vectorizes a given matrix by concatenating its columns. Through the use of the Kronecker tensor product $\otimes$, the matrix $\mathbf{H} = \mathbf{L} \otimes \mathbf{I}_K$ is still positive (semi-)definite. Simplex constraints and membership constraints in (6.4) are transformed to $\mathbf{A}\boldsymbol{\alpha} = \mathbf{1}_{(m-\ell)}$ and $\mathbf{B}\boldsymbol{\alpha} = \boldsymbol{\xi}$ in which $\mathbf{A} = \left[\mathbf{0}_{(m-\ell)\times\ell}\ \mathbf{I}_{(m-\ell)}\right] \otimes \mathbf{1}_K'$ and $\mathbf{B} = \mathbf{C}' \otimes \mathbf{I}_K$. The optimization problem (6.4) is convex and admits a global solution $\boldsymbol{\alpha}^*$. Any generic QP

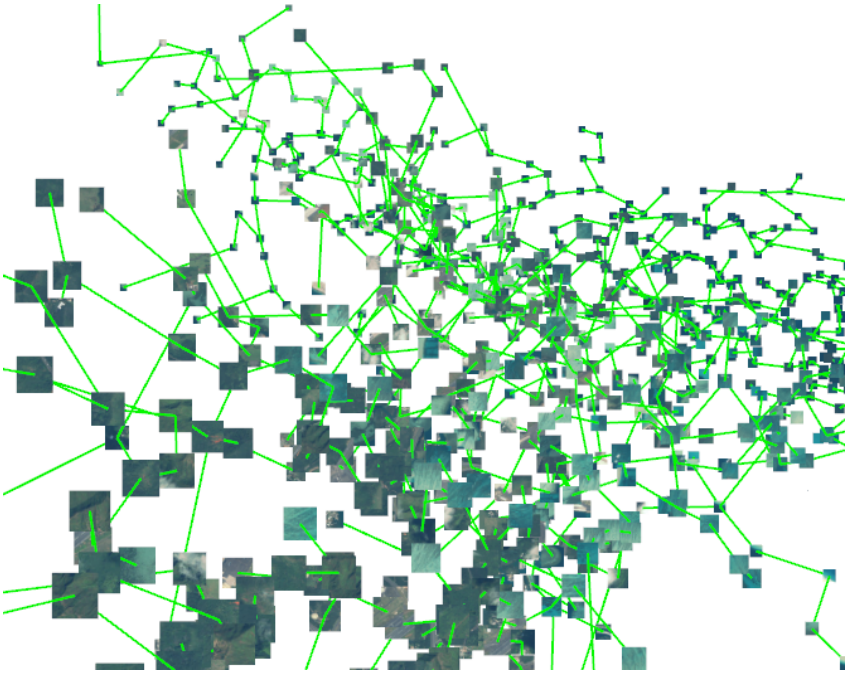


FIGURE 6.6 – Based on the k-nearest neighbors graph constructed from the data, the topology of the data is preserved by maintaining neighboring system from the embedded to the embedding space. The figure above illustrates the graph of satellite image patches.

solver is enough to solve this problem in small scale and an example with toy data is shown in Fig. 6.7.

6.3.1 Large-Scale Optimization

Optimizing (6.4) is prohibitively expensive when the data scales up and the optimization problem includes a mix of equality and inequality constraints. In order to develop an efficient and effective large-scale optimization algorithm, the original problem must be divided into subproblems which are easy to solve. We present in the following two such algorithms.

6.3.1.1 Chunking

This algorithm treats the embedding $\mathbf{\Phi} \in \mathbb{R}^{K \times m}$ as $K \times m$ scalar variables and stochastic optimization is used to approximate the solution. The pseudo code of the algorithm is listed in Algorithm 3. At every iteration, a subset of variables are chunked and solved by (6.5), which is the chunking version of (6.4). These subproblems are also convex because the square submatrices $\tilde{\mathbf{H}}$ are positive semidefinite. As a result, the optimization technique used to solve (6.1) can be applied to solve (6.5).

The chunking algorithm is also an instance of alternating minimization. The chunked variables are "active" variables while the rest are "inactive" ones. Via opti-

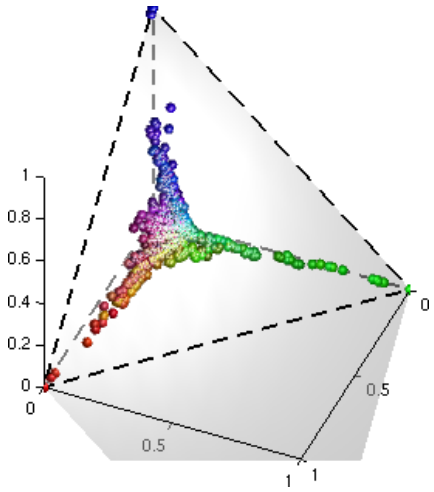


FIGURE 6.7 – A toy dataset consists of 1,000 points which are randomly sampled from a 3D RGB cube. The semantic space is defined by three semantics whose endmember examples are red $[1\ 0\ 0]'$, green $[0\ 1\ 0]'$ and blue $[0\ 0\ 1]'$. The embedding result reveals the dynamic of the data (progressive changes of colors) with respect to the three axes.

mizing (6.5) with respect to the active ones while keeping the inactive ones fixed, a smooth representation is achieved step by step. The alternating minimization asymptotically converges to the global solution as if using exact optimization algorithm (6.1). In practice, the optimization terminates when there is no significant change detected in $\mathbf{\Phi}$. Notice that active variables $\tilde{\boldsymbol{\alpha}} = \text{vec}\,(\mathbf{\Phi}_{ba})$ must be jointly optimized in order to guarantee the simplex condition in (6.1). If one updates $\mathbf{\Phi}_{ua}$ and $\mathbf{\Phi}_{va}$ separately, then this condition cannot be satisfied. A counter example with toy data in Fig. 6.9 shows the geometrical explanation for this condition of joint optimization.

6.3.1.2 Simplex Projection

An alternative solution is to uncouple the two constraints in (6.1) into two subproblems. The first one minimizes the smoothness term with respect to the equality constraints of the labeled data ; the second one projects the optimum of the first subproblem onto the unit simplex, which belongs the positive orthant and the L_1 ball. In this way, the first subproblem can be solved in large scale using exact optimization algorithms ; the second subproblem is a low-complexity projection of a variable into a convex set. Mathematical details of the solution is in the following.

Let us introduce an auxiliary variable $\mathbf{\Psi}$ and an approximation term between $\mathbf{\Psi}$ and $\mathbf{\Phi}$, (6.4) becomes

$$\begin{aligned} \min_{\mathbf{\Phi}\in\mathbb{R}^{K\times m};\mathbf{\Psi}\in\mathbb{S}} \quad & \tfrac{1}{2}\text{tr}\,(\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}') + \tfrac{\gamma}{2}\left\|\mathbf{\Psi}-\mathbf{\Phi}\right\|_F^2 \\ \text{s.t} \quad & \mathbf{\Phi}\mathbf{C} = \mathbf{Y} \end{aligned} \tag{6.7}$$

Algorithm 3 Chunking optimization algorithm

Input graph Laplacian $\mathbf{L}$ and label matrix $\mathbf{Y}$ whose ℓ first columns are endmembers.
Output Learned semantic maps $\mathbf{\Phi}^*$.
Set $\mathbf{\Phi}^{(0)} = \frac{1}{K}\,\mathbf{1}_K \times \mathbf{1}'_m$ and $t \leftarrow 0$
Repeat

1. Pick randomly two indices $b = \{u, v\} \in \{1, \ldots, K\}$ such that $u \neq v$.

2. Pick randomly indices $a \subseteq \{1, \ldots, m\}$ s.t $|a| \ll |\bar{a}|$ in which $\bar{a} = \{1, \ldots, m\} \backslash a$; for convenience let us denote $\breve{a} = a \backslash \{1, \ldots, \ell\}$ as the set containing non-endmember indices in a, and $\hat{a} = a \cap \{1, \ldots, \ell\}$ as the set containing endmember indices in a.

3. Let $m = |a|$, $m' = |\breve{a}|$ and $\tilde{\boldsymbol{\alpha}} = \mathrm{vec}\left(\mathbf{\Phi}^{(t)}_{ba}\right)$.

4. Solve the following QP

$$\begin{array}{ll} \underset{\tilde{\boldsymbol{\alpha}} \succeq 0}{\operatorname{argmin}} & \frac{1}{2}\tilde{\boldsymbol{\alpha}}'\tilde{\mathbf{H}}\tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\beta}}'\tilde{\boldsymbol{\alpha}} \\ \text{s.t} & \tilde{\mathbf{A}}\tilde{\boldsymbol{\alpha}} = \mathbf{1}_{m'} - \sum_{i \notin b} \mathbf{\Phi}^{(t)}_{i\breve{a}} \\ & \tilde{\mathbf{B}}\tilde{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\xi}} \end{array}, \tag{6.5}$$

   $\tilde{\mathbf{H}} = \mathbf{L}_{aa} \otimes \mathbf{I}_2$, $\tilde{\boldsymbol{\xi}} = \mathrm{vec}(\mathbf{Y}_{b\hat{a}})$, $\tilde{\boldsymbol{\beta}} = \mathrm{vec}\left(\mathbf{\Phi}^{(t)}_{b\bar{a}}\mathbf{L}_{\bar{a}a}\right)$, $\tilde{\mathbf{A}} = [\mathbf{I}_{m'}\ \mathbf{0}_{m' \times (m-m')}] \otimes \mathbf{1}'_2$, $\tilde{\mathbf{B}} = (\mathbf{C}_{a\hat{a}})' \otimes \mathbf{I}_2$.

5. Update

$$\begin{array}{lll} \mathrm{vec}\left(\mathbf{\Phi}^{(t+1)}_{ba}\right) & \leftarrow & \tilde{\boldsymbol{\alpha}}^* \\ \mathbf{\Phi}^{(t+1)}_{b\bar{a}} & \leftarrow & \mathbf{\Phi}^{(t)}_{b\bar{a}} \end{array} \tag{6.6}$$

6. $\mathbf{\Phi}^* \leftarrow \mathbf{\Phi}^{(t+1)}$

Until $\left\|\mathbf{\Phi}^{(t+1)} - \mathbf{\Phi}^{(t)}\right\|_F^2 \leq \varepsilon$

in which γ controls the amount of difference between $\mathbf{\Phi}$ and $\mathbf{\Psi}$. Again, the above formulation is nonconvex with respect to both $\mathbf{\Phi}$ and $\mathbf{\Psi}$ but convex with respect to one out of the two given that the other is fixed. If alternating minimization is applied to (6.7), we obtain two subproblems in which (6.8), as a function of $\mathbf{\Phi}$, seeks for a smooth embedding and (6.9), as a function of $\mathbf{\Psi}$, enforces simplex constraint. The complete algorithm is listed in Algorithm 4 where details of every step is presented below.

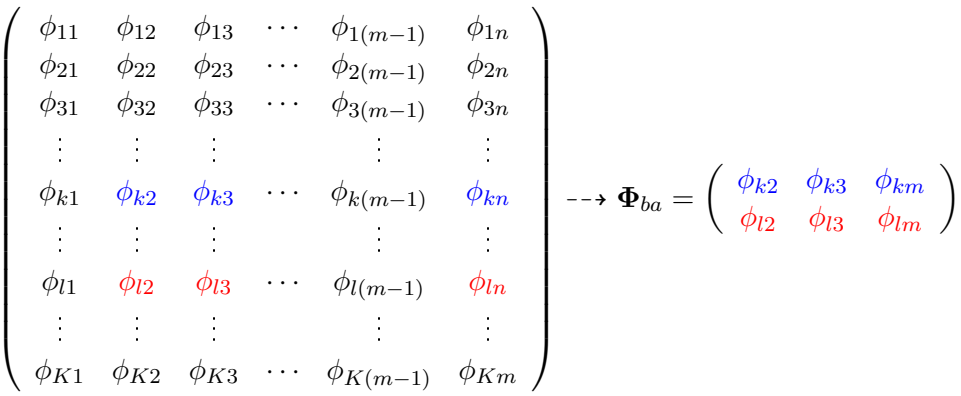


FIGURE 6.8 – According to the chunking approach, at every iteration, our algorithm randomly pick two row indices $b = \{k, l\}$ and few column indices $a \subset \{1, \ldots, n\}$; according to the alternating minimization, the submatrix $\mathbf{\Phi}_{ba}$ as active variables while the rest is fixed.

Updating $\mathbf{\Phi}$ Given $\mathbf{\Psi}$ fixed, minimizing (6.7) with respect to $\mathbf{\Phi}$ so that we get :

$$\begin{array}{ll} \min\limits_{\mathbf{\Phi}} & \frac{1}{2}\mathrm{tr}\left(\mathbf{\Phi}\left(\mathbf{L}+\gamma\mathbf{I}\right)\mathbf{\Phi}'\right) - \gamma\mathrm{tr}\left(\mathbf{\Psi}'\mathbf{\Phi}\right) \\ \text{s.t} & \mathbf{\Phi C} = \mathbf{Y} \end{array} . \tag{6.8}$$

The new subproblem (6.8) no longer includes simultaneously inequality and equality constraints and this reduces the complexity especially for large-scale problems. In our experiment, we solve (6.8) using Matlab Optimization toolbox [1].

Updating $\mathbf{\Psi}$ Given $\mathbf{\Phi}$, minimizing (6.7) with respect to $\mathbf{\Psi}$ so that we get

$$\min_{\mathbf{\Psi}\in\mathbb{S}} \quad \frac{1}{2}\left\|\mathbf{\Psi}-\mathbf{\Phi}\right\|_F^2 . \tag{6.9}$$

The above equation corresponds to m Euclidean projections of vector $\mathbf{\Phi}_i$'s onto the convex L_1 ball in $\mathbb{R}^K$ in which every projection corresponds to

$$\min_{\mathbf{v}\in\Delta} \quad \frac{1}{2}\left\|\mathbf{v}-\mathbf{u}\right\|_2^2 , \tag{6.10}$$

where $\mathbf{v}$ must be in the convex set $\Delta = \{\mathbf{v} \in \mathbb{R}^K | v_i \geq 0, \sum_i v_i = 1\}$ while $\mathbf{u}$ is in $\mathbb{R}^K$. Here $\mathbf{u}$ and $\mathbf{v}$ substitute for columns of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ respectively. Based on the simplex projection algorithms proposed in [Kyrillidis 2012, Duchi 2008], we apply them in order to solve (6.10). This algorithm calculates the optimal projection in $\min\left(\mathcal{O}(\rho K), \mathcal{O}(K\log(K))\right)$-time using the greedy selection and soft thresholding, i.e., by picking the ρ-largest entries of $\mathbf{\Phi}_i$. Mentioned in Definition 6.1 is an algorithm

1. A subspace trust-region method based on the interior-reflective Newton method [Coleman 1996] is applied in which each iteration involves the approximate solution of a large linear system using the method of preconditioned conjugate gradients (PCG).

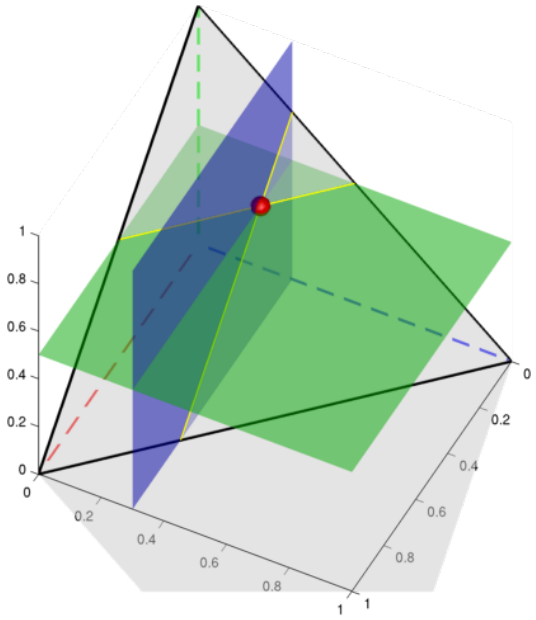


FIGURE 6.9 – In Algorithm 3, the necessary condition for (6.5) to work is that the row indices b must satisfy $|b| \geq 2$. Assume that $|b|$=1, then the number of simplex constraints of (6.5) is equal to $n - m$; since $|b| = 1$, the system of simplex constraints has a unique solution, which makes (6.5) cannot be optimized. The above figure illustrates a simple case where $K = 3$ so that the size of the subproblem must satisfies the condition $2 \leq |b| \leq K$. One can obtain a subproblem by fixing one out of three unknowns and optimizing with respect to the rest. In this example, the green coordinate z (the green plane) is fixed at 0.5 so that $x+y = 1-z = 0.5$, which means that the values of x and y can be changed with respect to to the constraint $x + y = 0.5$. The point (x, y, z) can be anywhere along the line segment which is the intersection between the plane $z = 0.5$ and $x + y + z = 1$ (the gray plane). If one fixes another coordinate, for example $y = 0.3$ (the blue plane), then the point (x, y, z) is permanently stuck at (0.5, 0.3, 0.2) (the red point).

that allows determining adaptively the threshold ρ. See [Kyrillidis 2012, Duchi 2008] for more details.

Definition 6.1 *Let assume that vector* $\mathbf{u} \in \mathbb{R}^K$ *is sorted in descending order so that* $u_1 \geq u_2 \geq \ldots \geq u_K$. *The image of* $\mathbf{u}$ *via the projection* $\mathcal{P}_\Delta$ *that maps* $\mathbf{u}$ *onto* Δ *is* $\mathbf{v}$ *whose coordinates are defined as*

$$v_i = \left(\mathcal{P}_\Delta\left(\mathbf{u}\right)\right)_i = \left(u_i - \tau\right)_+ , \tag{6.11}$$

where $\tau = \tau_\rho$, $\tau_j = \frac{1}{j}\left(\sum_{i=1}^{j} u_i - 1\right)$ *and* $\rho := max\{j : u_j > \tau_j\}$.

Algorithm 4 Simplex projection algorithm

Input Laplacian matrix $\mathbf{L}$, and label matrix $\mathbf{Y}$ whose ℓ first columns are endmembers.

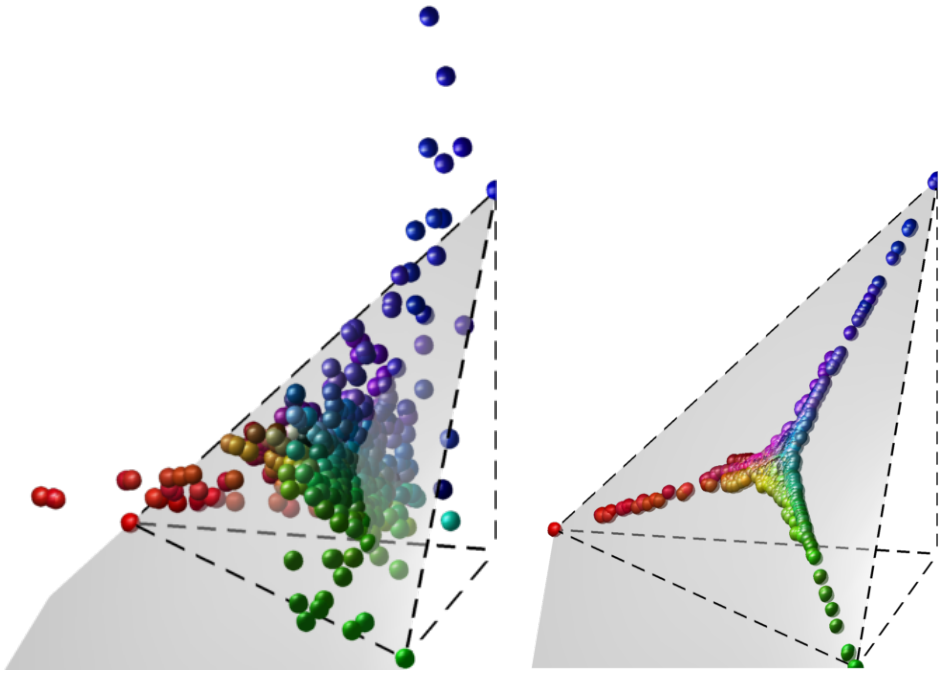
Output Learned semantic maps $\mathbf{\Psi}^*$.

Initialize $\mathbf{\Psi}^{(0)}$ with random values

Repeat

1. Solve the quadratic problem (6.8) and set $\mathbf{\Phi}^{(t)} \leftarrow \mathbf{\Phi}^*$
2. For $k : 1 \longrightarrow m$
   (a) Sort $\mathbf{\Phi}_k^{(t)}$ into $\mathbf{u} : u_1 \geq u_2 \geq \ldots \geq u_K$
   (b) Find index ρ such that $\rho := \max \left\{ j : u_j \geq \frac{1}{j} \left(\sum_{i=1}^{j} u_i - 1 \right) \right\}$
   (c) Compute $\tau = \frac{1}{\rho} \left(\sum_{i=1}^{\rho} u_i - 1 \right)$
   (d) $\mathbf{\Psi}_{ik}^{(t)} = (u_i - \tau)_+, \quad i = 1, \ldots, m.$

Until $\left\| \mathbf{\Psi}^{(t)} - \mathbf{\Psi}^{(t-1)} \right\| \leq \varepsilon$



(a) The visualization of $\mathbf{\Phi}$ (b) The visualization of $\mathbf{\Psi}$

FIGURE 6.10 – Illustrations of a single step of the simplex projection algorithm Alg. 4. On the left is the learned embedding $\mathbf{\Phi}$ via minimizing (6.8). We can see the smooth change of color in the along paths from one vertex to another. On the right is the image of $\mathbf{\Phi}$ via the projection in (6.9). Notice that data points are now lying on the simplex surface $x + y + z = 1$ as a result of simplex constraint.

(a) Eiffel tower (b) Vosge, Paris (c) Central park, NYC (d) Grand Palais, Paris

FIGURE 6.11 – Professional use of satellite images includes land management, structures surveillance, disaster prevention, agriculture (crop, forest) surveillance. For example, an architect may be interested in finding buildings with some specific structures while he does not know their names or location. Without tagging information, it is difficult to find such structures using systematic image browsing such as panning and zooming.

6.4 Data Visualization

As the first part of this section, we apply (6.1) to data visualization with huge satellite images and a generic image database. As the second part, Section 6.5 describes how to deploy visualization results for interactive image search.

6.4.1 Satellite Images

With the rapid growth of remote sensing technology and high performance computers as well as high-speed Internet, mapping services are nowadays emerging (Google Maps, GeoPortail, Bing, etc.). They are useful in terms of geographical navigation to professional tasks such as land management, city planning, agriculture, foresting, and large public purposes such as tourism and driving aid. Currently, locations in satellite images are searchable only if those maps are properly annotated with names about countries, regions, streets, landmarks, etc. (see Fig. 6.11). Navigating without these metadata turns out to be tedious. For instance, using conventional navigation tools in order to explore large satellite images – let us say with more than $10,000 \times 10,000$ pixels – turns out to be helpless. Indeed, navigation becomes systematic and often tedious as the user spends his effort in zooming (in/out) and shifting from one area to another with the only navigation criterion being "geographic proximity" ; this burden is further amplified when meta-data are scarce or unintuitive to the user.

In this experiment we investigate how our proposed method will improve the way that visual contents are searched in satellite images. An experiment is set up as follows. The input image is divided into patches using a rectangular grid with appropriate resolution (Fig. 6.12). The patch size should be adjusted according to the image resolution otherwise a patch contains too much or few visual object details. In our experiment, the original satellite image of dimension $6,876 \times 7,265$ pixels,

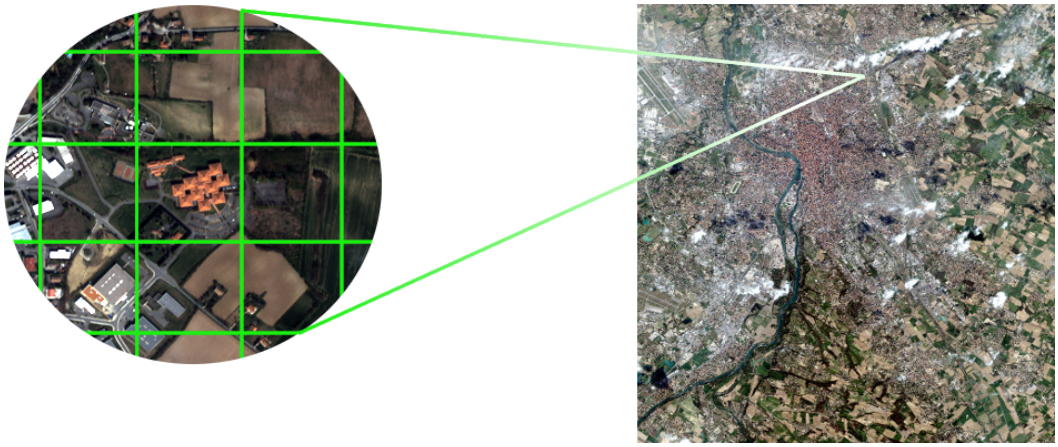


FIGURE 6.12 – A thumbnail view of the real size satellite image is shown in the right while a particular zoomed region is depicted on the left. As the image is equally divided into rectangular cells, the amount of semantics contained in every cell is kept fixed, i.e., if there are more buildings then there is less vegetation (grass, plantation).

is partitioned into approximately $12,000$ patches of 64×64 pixels. Based on these patches, a k-nearest neighbors graph is constructed with $k = 3$; weights associated with edges represent visual similarity between patches. This weight is computed using RBF function (4.10) that fuses visual similarity of color histogram and WLD features [Chen 2010] (combination of intensity and gradient).

We define $K = 4$ usual semantics (building, road, vegetation and water) and we select 15 patch examples per semantic. The underlying membership vectors in $\mathbf{Y}$ are set accordingly as shown in (6.12) ; again we assume that each selected example includes only one semantic so its corresponding membership vector is pure. Selected examples, even though few, are representative enough and cover at some extent the diversity of these four semantics.

$$\mathbf{Y} = \left(\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & & 0 & & 0 & \text{vegetation} \\ 0 & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & \text{building} \\ 0 & 0 & 1 & 0 & & 0 & & 0 & \text{water} \\ 0 & 0 & 0 & 1 & & 0 & & 0 & \text{road} \\ \end{array} \right. \quad \left. \begin{array}{ccccccc} \multicolumn{9}{c}{} \\ \multicolumn{9}{c}{} \\ \multicolumn{9}{c}{} \\ \multicolumn{9}{c}{} \\ \end{array} \right. \quad (6.12)$$

The learning result is shown in Fig. 6.13 in which just three out of the four semantics are displayed. Zoomed at the vertices of the simplex are the endmember patches of the three semantics : building, vegetation, and road. We can observe the decrease of green area, by tracking appearance changes of image patches, when shifting from semantics of vegetation and road to the semantic of building. For example, traversing in the visualization from vertex road to building, roads in image patches become smaller (we are leaving rural areas where there are highways), but more

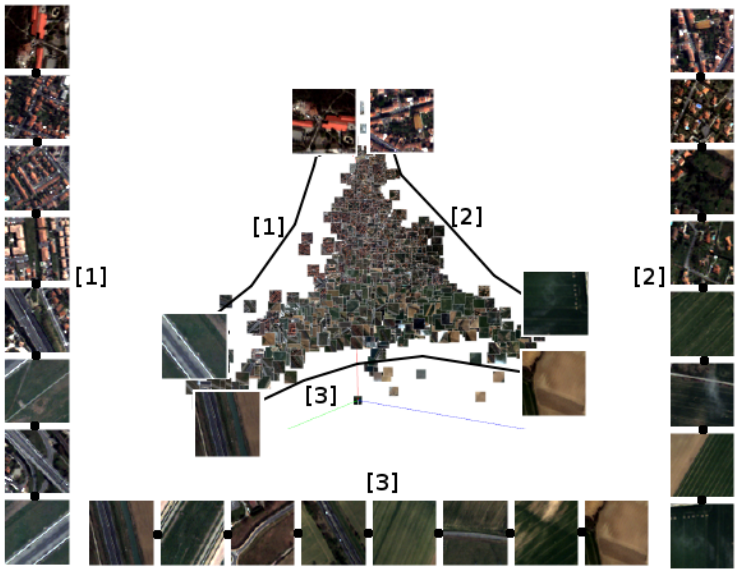


FIGURE 6.13 – At the center is the visualization result of the learned subspace. This images cloud has the shape of simplex whose extremities at three vertices are superimposed by pure membership examples (top : "building," right : "vegetation," left : "road"). Traversing along the paths of the simplex (for example paths [1], [2], [3]), one can observe semantic changes from an extremity to another one.

houses and buildings appear. Approaching near the vertex building, big buildings and houses dominate image patches. This opens a new way of semantic search in which the user just scans the visualization on the regions visually similar to his mental picture.

Satellite images have several special properties. First, image patches are visually similar to each other within local geographical locations. Thus smooth appearance changes lead to smooth semantic changes. Second, it is not difficult to obtain end-member examples because monotonic scenes are popular, i.e., urban areas with lots of houses or vast forests. Third, the simplex constraints (abundance of semantics must sum to one) are easy to satisfy because the sum of proportions of semantic objects in satellite images corresponds to the fixed size of image patches.

6.4.2 Scene Images

In this section we extend the application of our method to generic images. The SiftFlow dataset, which is a subset of the LabelMe database, is used in our experiment. The dataset contains 2688 scenes acquired from categories such as coast, mountain, forest, open country, street, inside city, tall buildings and highways. Those scenes cover different variations in illumination, perspective, distance, intra-variant of object appearance. In all of 2688 images, there are approximately 29,000 instances of 33 object classes. In our experiment, we consider object classes as semantics and since the number of instances per class is highly imbalanced, 12 most frequent classes

– compared based on their numbers of instances – are chosen for visualization.

In order to prepare endmember examples, a training set of 200 images is randomly selected from the dataset. Based on the given groundtruth of every training image, object segments are extracted from the training images (see Fig. 6.15 and 6.16). The set of object segments of all the training data is used as the source of endmembers. For every semantic (a.k.a object class), 30 segments are chosen randomly ; the number of endmember examples occupies about 1% of the number of objects in the dataset. The number of endmembers in total is $12 \times 30 = 360$ while the number of test points is $2688 - 200 = 2488$. An affinity graph of neighborhood size $k = 5$ is constructed from 2488 unlabeled images and 360 labeled endmembers. Similarity measure is computed using Gaussian RBF function (4.10) whose inputs are four visual descriptors texton histogram, dense-SIFT descriptor histogram, color histogram, and GIST descriptor.

Shown in Fig. 6.14 and 6.15 are the embedding learned from the SiftFlow dataset. Similarly to the experiment with satellite images, the dynamic of the data corresponds to the predefined semantics while image content smoothly changes when traversing from a vertex to another (see Fig. 6.17).

6.4.3 Summary

There are two key properties of the visualizations produced by our method. First, the low-dimensional representation provides an overview of the database, which serves as the a "page zero" for the user to start his query. This "page zero" is an overview of the database in which the user can quickly interpret where in the map his mental target could be found. Second, the new representation is well described by semantics which help the user to navigate easily. In the following section we will present an interactive search software Spacious, which is our effort in utilizing the visualization results for search applications.

6.5 Interactive Mental Search

This section introduces our software Spacious for interactive visualization. This software is our effort in experimenting visualization-based search as depicted by the diagram in Fig. 6.4. The objective of Spacious is to provide a dual view for image databases. As shown in Fig. 6.18, the *manifold view* shows the semantic visualization of a satellite image ; the *map view* displays images as they are. In the case of satellite images, the map view shows the whole satellite image, which is also the geographical map of the location captured in the image. About the technical details of Spacious, it is built on the top of Partiview[2], a simple but powerful graphics engine that uses OpenGL for 3D rendering. With this engine, we are able to visualize tens of thousands image patches in large scale databases.

2. http ://www.haydenplanetarium.org/universe/partiview

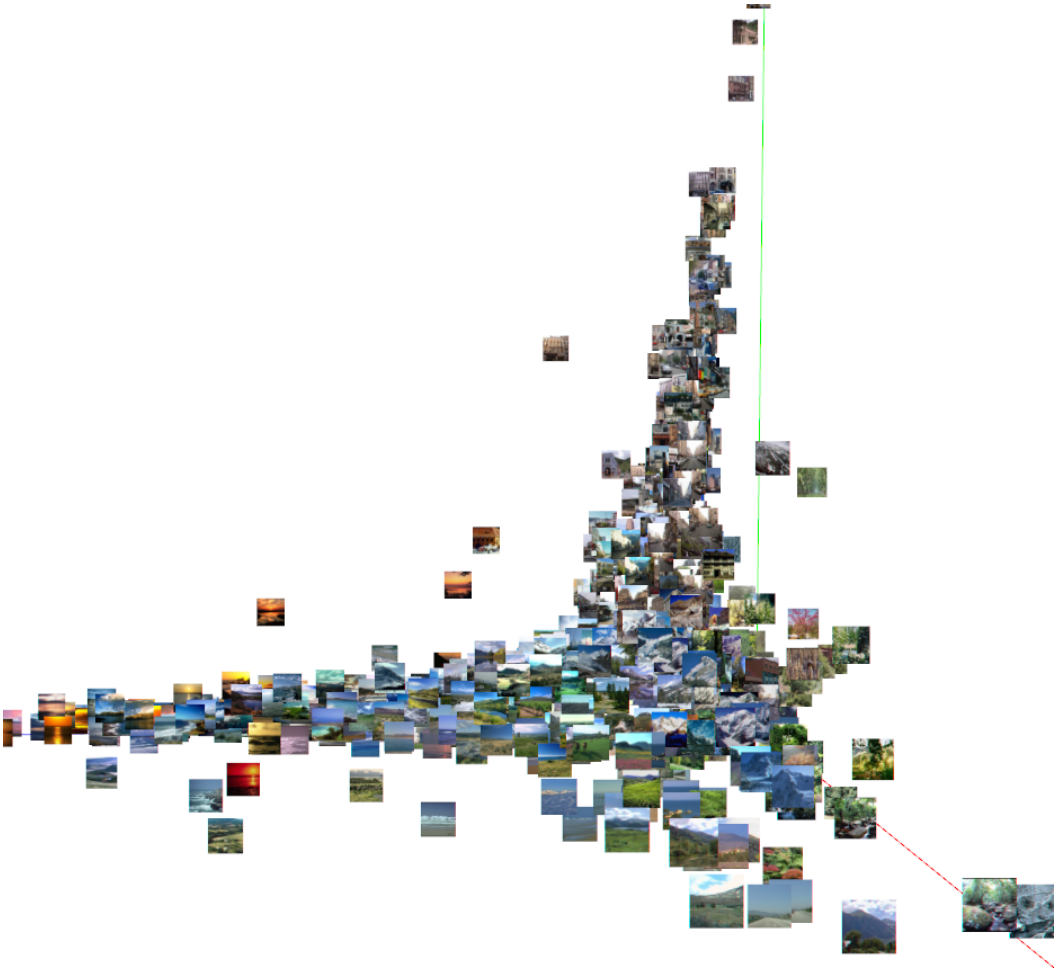


FIGURE 6.14 – The visualization of three out of twelve semantics of the embedding learned from scene images. This visualization shows that the learned embedding can reveal dynamics of the data. Endmember examples of three semantics : sky, building, and mountain are located at simplex vertices. At every semantic there is a concentration of images having similar semantic contents. For example, images containing sky and sea are located near the vertex sky. Moving farther from it, images with sky and sea are progressively replaced by coast, sand, rock and even less semantically relevant ones such as tree and mountain. At the barycenter of the visualization there are images mixing several semantics.

After the learned embedding is loaded into Spacious, the user can explore the visualization using various supporting tools such as different navigation modes (orbital, flight, rotate, translate), subset selection, subset filtering, dimension switching. Especially, dimension switching functionality is useful in case the embedding has more than three dimensions. It allows the user to select maximum three out of the dimensions of the embedding. Every time the user interacts with the data in one

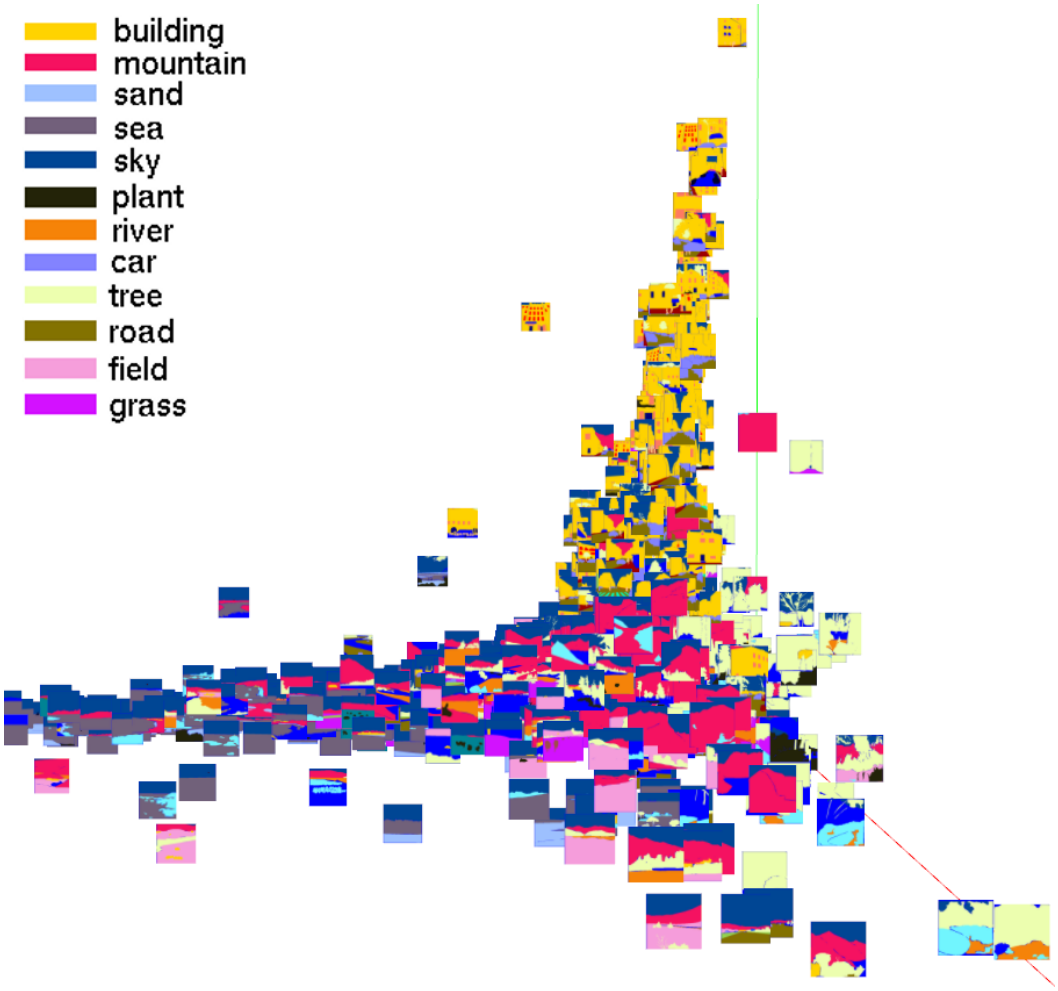


FIGURE 6.15 – The visualization of Fig. 6.14 with ground truth. Due to appearance ambiguities, there are some confusions between *tree* and *mountain* in the bottom right corner of the visualization.

of the two views, Spacious synchronizes the two views so that the user always gets two sources of information, which help him to search more efficiently.

Initially, Spacious is designed for satellite image navigation. Every data point in the visualization corresponds to a unique location in the satellite map, and vice-versa. If the mental query is "find a location with the same proportions of vegetation, road, and building," then the user firstly switches three dimensions to the semantics vegetation, road, and building. Secondly the user uses mouse or sliders in order to direct the selector (the yellow cube in Fig. 6.21(a)) to focus somewhere at the barycenter of the simplex (see Fig. 6.19). The locations of the selected patches are then promptly highlighted in the map view. Navigation tools such as panning, zooming and subset selection of the map view are available for the user to refine

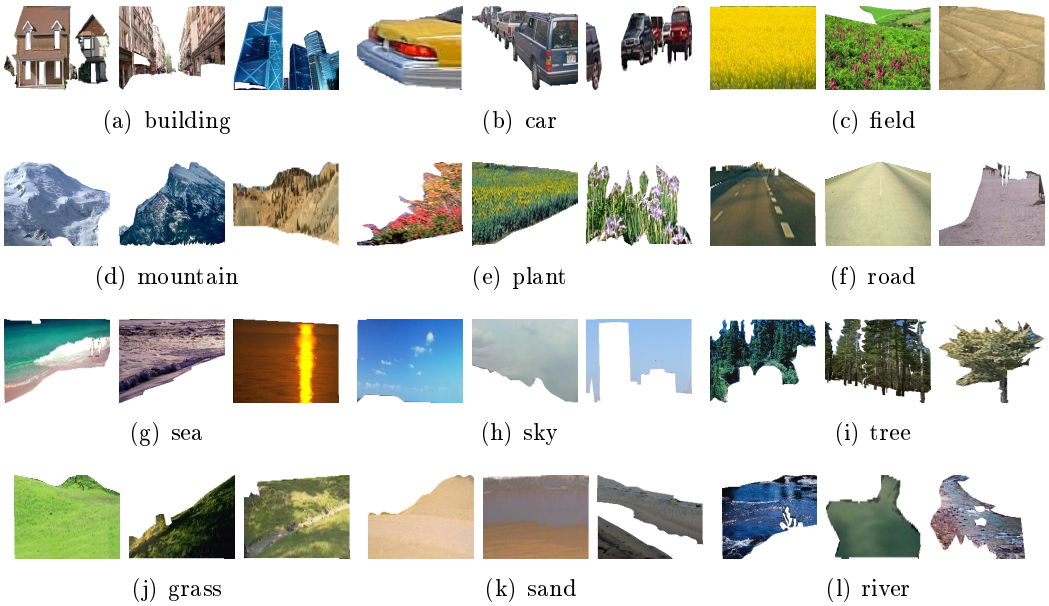


FIGURE 6.16 – Examples of endmembers taken from 12 object classes in the SiftFlow dataset. Since groundtruth (object boundaries) are available for every training image, we easily obtain endmember examples by masking out irrelevant objects in images.

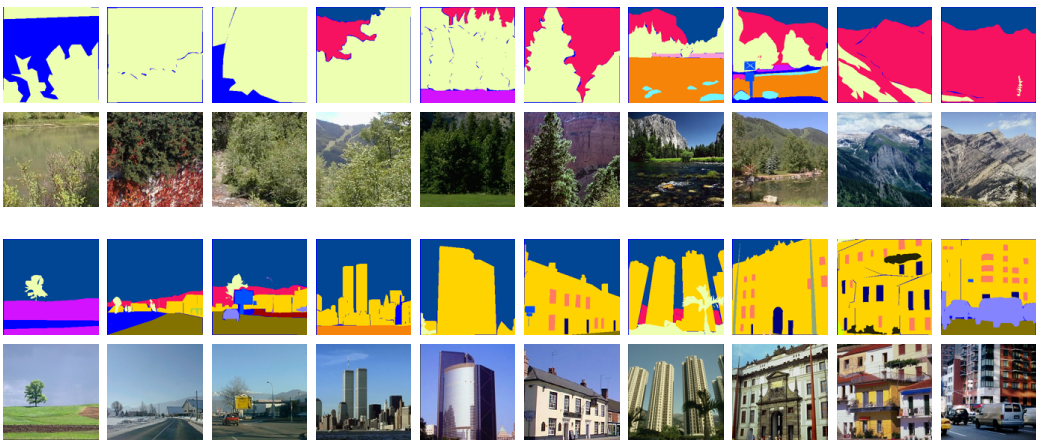


FIGURE 6.17 – From left to right are the images sampled from directed paths along the visualization of SiftFlow dataset in Fig. 6.14. The top row shows landscape changes of the path that originates from countryside to mountain. The sampled images show progressive changes from field scenes with bushes to forest scenes with trees and hills, and finally to highland. Similarly, the bottom row shows landscape changes as one traverses from highways to cities.

locations he is interested in. This querying process can be looped for several times before the user is satisfied with his findings.

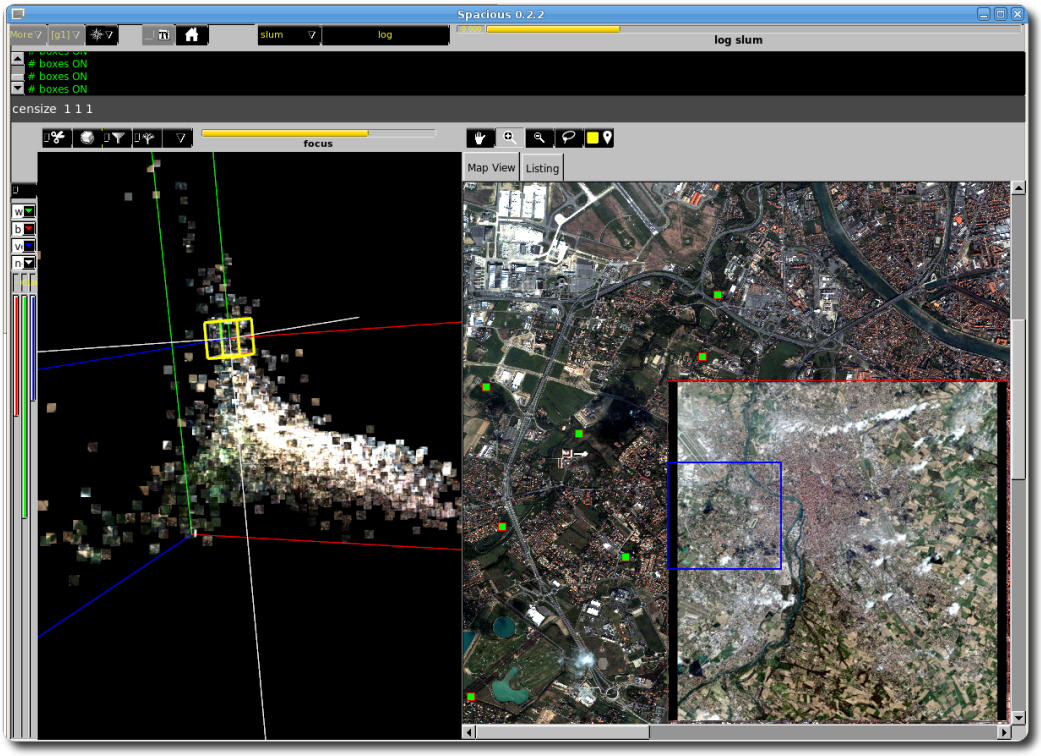


FIGURE 6.18 – Spacious is dual view. The *map view* on the right displays geographical map of the location captured by the satellite image. Systematic navigation tools such as panning and zooming are used to navigate the map. The *manifold view* on the left visualizes the learned low-dimensional embedding. Spatial navigation tools such as rotation, translation, scaling, and dimension switching are used to interact with the visualization system. Besides, selection tools are equipped in the both views. Their use is explained by the following example. Assume that the user has a mental target ; since he already grasped the visualization of the image database, he probably knows where in the embedding his mental target could be found. The cube selector (in yellow) is used to select that potential location. The cube size can be adjusted by the slider *focus*. The user can clip off the patches outside the cube so that he is not distracted. As soon as the selector is positioned, the locations whose associated patches are within the cube are highlighted as green dots in the map view. Thus the user knows which geographical locations are selected by the cube. Patch refinement can be accomplished in this right view by selecting a sub-group of highlighted points. This action is promptly updated in the left view. The user either finds his mental target or continues to refine the result or invokes a new query.

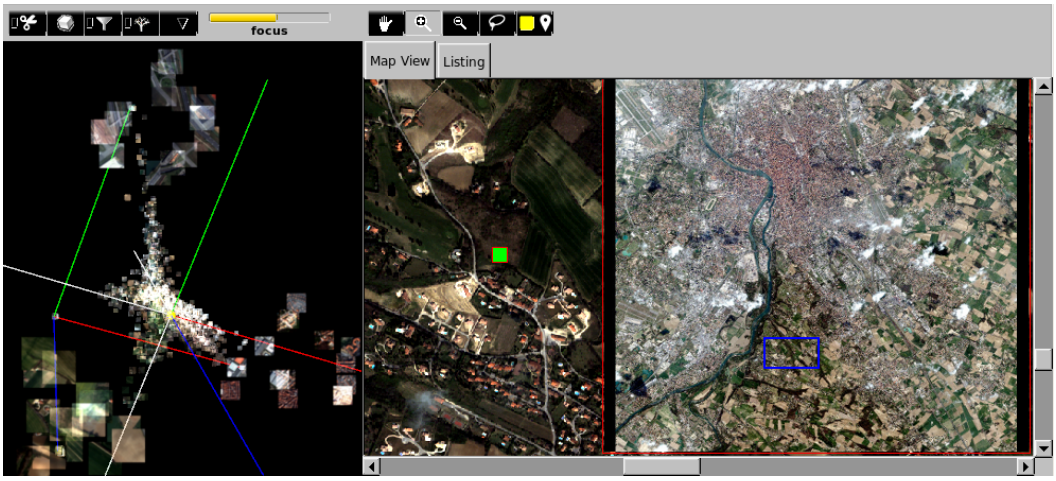


FIGURE 6.19 – A snapshot of Spacious captured at the moment of querying satellite image. Three out of four semantics are shown ; their endmember examples are zoomed and superimposed for clarity. A mental query "find a location with the same proportions of vegetation, road, and building" corresponds to positioning the cube selector to the barycenter of the visualization. On the right view there is a highlighted position whose surrounding landscape includes fields, roads, and houses. Thus this location satisfies conditions of the mental query. The map preview in the bottom right shows that the highlighted position is located in rural area, which explains why there are simultaneous appearance of vegetation, houses, and roads.

6.6 Semantics Ranking and Relevance Feedback

As shown in Section 6.4 and 6.5, our method successfully produces meaningful embeddings and demonstrates their use for interactive search. In this section, we conduct quantitative analyses in order to measure how good our method can performs compared to related works. In particular, two problems are investigated : semantic ranking and image retrieval with relevance feedback. Evaluating the efficiency of information search is not trivial because it relates to subjective judgments of human users. Nonetheless, there are tasks such as image ranking that can be evaluated independently from user interaction. However, for the second problem, it should be evaluated based on real querying. Due to limited time and resources, we conduct our experiments by simulated querying and feedback.

6.6.1 Semantic Ranking

The objective of semantic ranking is to learn a parametrization of a given image database such that those images are ranked with respect to their semantic abundances. From this point of view, our method can be considered as a ranking algorithm because it also learns a representation of the data subject to smooth changes of semantic content. Together with the smoothness regularizer, simplex

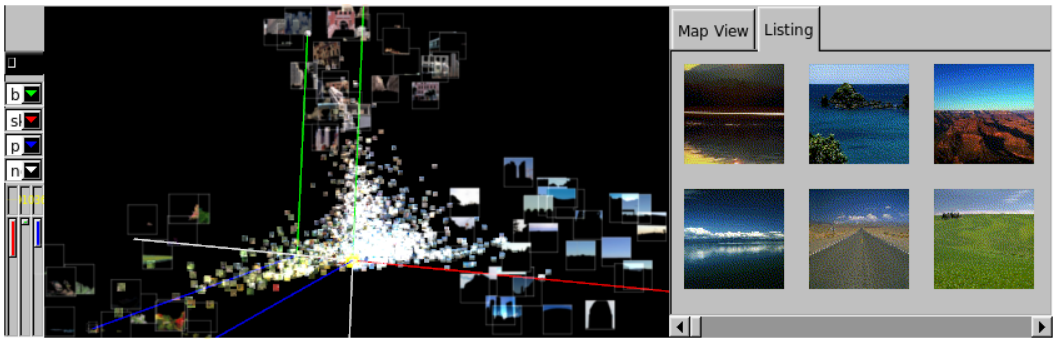
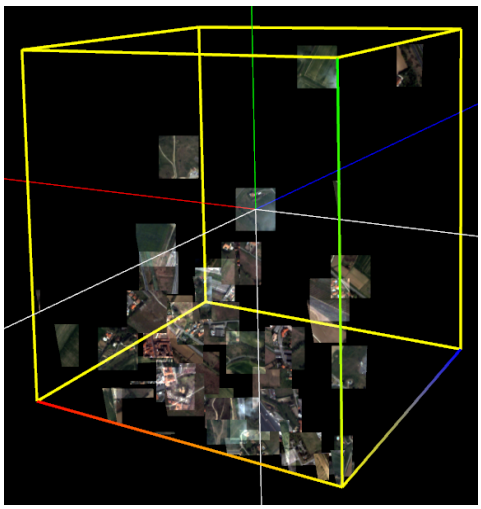
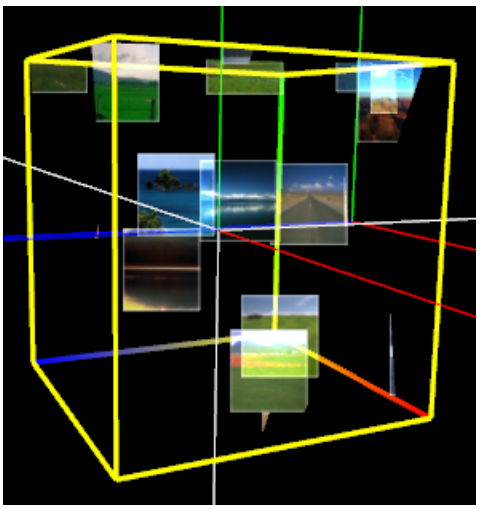


FIGURE 6.20 – A snapshot of Spacious captured at the moment of querying images of the SiftFlow dataset. The dimensions of the manifold view are switched to plant, sky, and mountain ; their endmember examples (image segments) are superimposed for clarity. A mental query "find photos with sky and plant" corresponds to positioning the cube selector somewhere at the middle of the simplex edge that connects vertices plant and sky. On the right view is the thumbnail view of the selected images. A small selection of the cube results in scenes that share a common configuration of open space with horizon.



(a) Shown in the cube are the image patches selected by the cube selector in Fig. 6.19. Notice that almost of them have all three semantic vegetation, building, and road.



(b) Shown in cube are the images selected by the cube selector in Fig. 6.20. About half of them are the scenes composed of sky and plant.

FIGURE 6.21 – The cube selectors of satellite images and SiftFlow dataset. Notice that image patches outside the cube are clipped for easy observation.

constraints produce embeddings which are well ranked with respect to multiple semantics. These constraints enforce the tradeoff between semantics of an image such that the total of semantic abundances sums to one. As a consequence, our subspace

learning method is different from other approaches [Joachims 2002b, Siddiquie 2011, Liu 2011b, Lin 2005, Zhou 2003b] in which they just learn a single semantic ranking function per training. In order to compare with them, we have to convert embedding coordinates of every data point to membership values which can be used as ranking scores. For example, in order to rank images of a database with respect to the i^{th} semantic, then the i^{th} row of $\mathbf{\Phi}$ is selected and sorted in descending order. It is trickier to assign ranking scores to multi-semantic queries. Let us denote $\mathcal{Q}$ as a set consisting of semantics mentioned in a mental query and $|\mathcal{Q}|$ denotes the number of semantics contained in $\mathcal{Q}$. Let us define a vector $\mathbf{q} \in \mathbb{R}^K$ whose entries q_k's equal $\frac{1}{|\mathcal{Q}|}$ only if all $|\mathcal{Q}|$ semantics contained in that query have similar proportions. Here $\mathbf{q}$ can be seen as the highest ranked point which contains exactly $|\mathcal{Q}|$ semantics with equal membership values. For example, a double-semantic query has $\mathbf{q} = [\frac{1}{2}\ \frac{1}{2}]'$ and a triple-semantic query has $\mathbf{q} = [\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]'$. The ranked result of that query is obtained by sorting in descending order the Euclidean distances $\|\mathbf{q} - \mathbf{\Phi}_i\|_2$ between $\mathbf{q}$ and the the embedding $\{\mathbf{\Phi}_i\}$ of the data. Notice that NDCG@k scores are compared between queries having the same number of semantics.

6.6.1.1 Evaluation Criterion

Normalized Discount Cumulative Gain (NDCG) is used to evaluate ranking performance. NDCG at rank k is defined as

$$\text{NDCG@}k = \frac{1}{Z} \sum_{j=1}^{k} \frac{2^{\text{rel}(j)-1}}{\log(1+j)} \tag{6.13}$$

where $\text{rel}(j)$ is the relevance of the j^{th} ranked image and Z is a normalization factor which guarantees that a perfect ranking result corresponds to NDCG value of 1. The NDCG formula favors relevant documents appearing higher in the list while concerns less the ones near the bottom of the list. A list that obtains higher NDCG is better ranked. For single-semantic queries, $\text{rel}(j)$ gets binary value ; for multi-semantic queries, $\text{rel}(j)$ gets integer values in range $[1, \ldots, |\mathcal{Q}|]$. The most relevant images that contain all the mentioned semantics in $\mathcal{Q}$ will have $\text{rel}(j) = |\mathcal{Q}|$; the images that contain $(|\mathcal{Q}| - 1)$ out of $\mathcal{Q}$ semantics will have $\text{rel}(j) = |\mathcal{Q}| - 1$; and so on and so forth.

6.6.1.2 Experimental Setup

SiftFlow dataset. This is the dataset used in the previous sections for data visualization. We choose the 12 most popular object classes out of 33 of the dataset as the semantic set. We compare single-semantic ranking between our method and the method proposed in [Kovashka 2012], which is a modification of SVM-rank [Joachims 2002b]. In this experiment "SVM-rank" is used to denote their method. We use their implementation available online and run it on the SiftFlow dataset. For every semantic, an SVM rank function is learned from relevant images sampled from 200 training images. Among them, 200 random image pairs are chosen

as the training data for SVM-rank ; within each pair there is a relative comparison (less than, equal, more than) between the two images subject to the semantic. After filtering out any existing duplicate in the 200 pairs, these pairs are used to train the modified SVM-rank [Kovashka 2012]. We also evaluate multiple-semantic ranking in which there are 126 double-semantic queries and 1068 triple-semantic queries. Although multiple-semantic ranking has not been addressed in related works, they are good indications for us to understand difficulties of the ranking problem when more criteria are added.

For our part, we construct a similarity graph that connects visually similar image pairs based on the weight computed by the Gaussian RBF function in (4.10). Four visual descriptors are used to compute visual similarity : texton histogram, SIFT histogram, color histogram and GIST.

Outdoor Scene Recognition (OSR) dataset. This dataset reuses the SiftFlow data with another semantic set. This set consists of 6 scene attributes natural, open, perspective, size-large, diagonal-plane, and depth-close (see [Oliva 2001a] for more details). Database groundtruth with respect to this semantic set is provided as predicate matrices in Fig. 6.22. We do not run SVM-rank code with this dataset but re-use their published result [Kovashka 2012]. For our part, 200 endmembers are randomly chosen from 2888 images of the dataset ; membership values of these endmember examples are computed based on the predicate matrices and GIST [Oliva 2001a] descriptor is used to build a similarity graph. It is important to notice that the endmember condition is not satisfied in the case of OSR dataset. This is evident as we look at the predicate tables, i.e., there is no exclusive occurrence of a semantic in any of the images in the database. Thus it is interesting to see how good our method performs in the absence of this condition, which often occurs in real world databases.

Shoes dataset. This is a subset of the Attribute Discovery Dataset [Berg 2010] with minor modification made by [Kovashka 2012]. The dataset contains 10 classes and 14,658 shoes images. Every image can have from at least 1 to maximum 10 attributes. Attributes of the shoes displayed in the image will be the semantics of that image. For example, this shoes is open, sporty, and feminine ; that shoes is bright and shiny. The groundtruth of Shoes dataset is predicate rules shown in Fig. 6.22. Again we reuse the published results in [Kovashka 2012] for Shoes dataset. For our part, 200 endmembers are randomly chosen from 2888 images ; their membership values are inferred from the predicate tables in Fig. 6.22. In order to construct a similarity graph, color histogram and GIST descriptor are fed to (4.10) in order to compute edge weights between images. Similarly to the case of OSR dataset, Shoes datasets does not satisfy the endmember condition. However, and as shown in the following results, the absence of this condition does not prevent our algorithm from being applied to real world problems effectively.

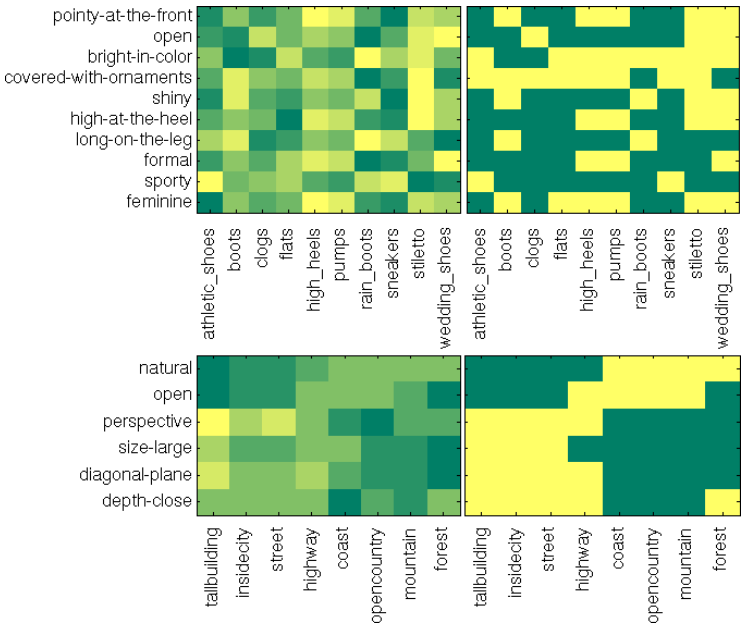
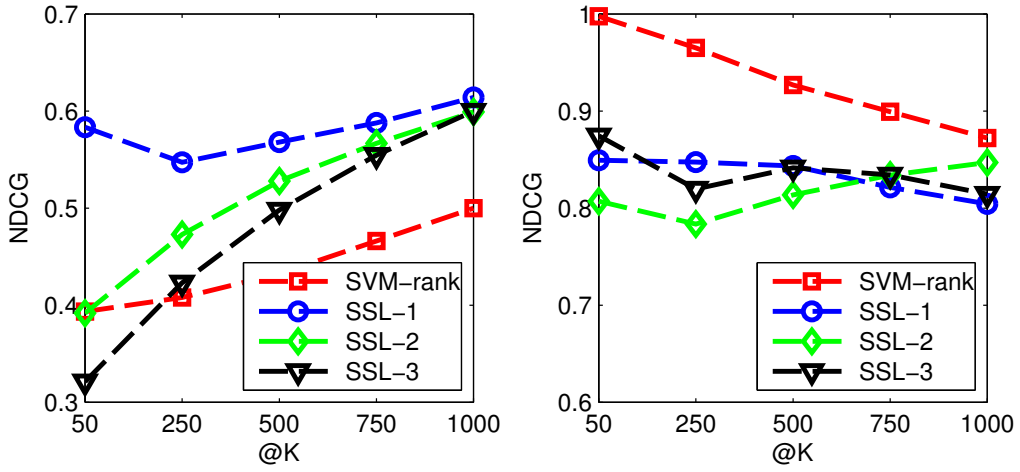


FIGURE 6.22 – The predicate matrices of Shoes [Berg 2010, Kovashka 2012] (top row) and OSR [Oliva 2001b] (bottom row) datasets. Every object class (labeled in columns) owns a tuple of attributes (labeled in rows). We use those attributes as semantic concepts for our problem. Every entry of a row represents the likelihood $P(\text{object}|\text{attribute})$; these likelihoods are normalized with respect to every row. On the right are the binary membership tables, which are the thresholded versions of the left ones. While training data preparation uses predicate rules from the left matrices, the right ones are used in ranking evaluation (binary membership values indicate whether an object or image contains that object is relevant to an attribute).

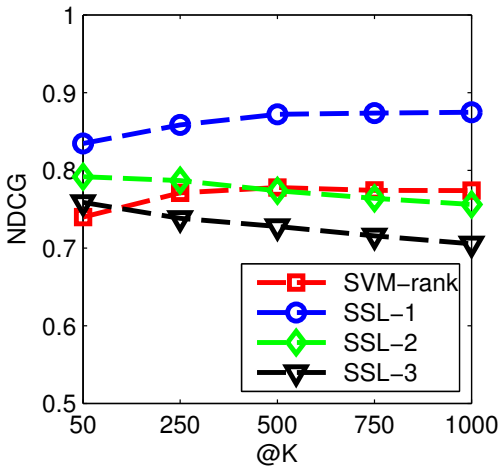
6.6.1.3 Discussion

Comparative results on the three datasets are shown in Fig. 6.23 ; our method performs better than SVM-rank with LabelMe and Shoes datasets and worse with OSR dataset. This may be due to high correlation between semantics in OSR dataset (see the predicate rules of OSR dataset in Fig. 6.22). For example, semantics within groups perspective, size-large, diagonal-plane, depth-close and natural, open are mutually correlated. Due to this property, the endmember condition mentioned in (6.1) cannot be satisfied. In other word, labeled examples of OSR dataset are not pure endmembers, i.e., labeled images contain more than one semantic. Equivalently, those examples are not positioned at the vertices of the simplex so the learned representation tends to collapse forming a straight line. This behavior is attenuated with Shoes dataset despite the use of labeled data which are not necessarily pure endmembers. Indeed, we observe that these data are not very correlated with each other (see again Fig. 6.22) and hence not grouped into clusters but well scattered in the simplex. We also observe from experiments that ranking, with multiple seman-



(a) LabelMe dataset

(b) OSR dataset



(c) Shoes dataset

FIGURE 6.23 – Evaluation of image ranking problem between semantic subspace learning (SSL) and SVM-rank [Kovashka 2012]. The vertical axis NDCG is the Normalized Discount Cumulative Gain that favors relevant documents ranked near the top of the list ; the horizontal axis is the top K documents of the ranked list. SSL-1, -2, -3 denote single-, double-, and triple-semantic ranking results. Note that the SVM-rank curve is single semantic.

tics, is more difficult than ranking with a single semantic. Indeed, by comparing the curves SSL-1, SSL-2, and SSL-3, it is clear that more semantics make the ranking task more difficult.

6.6.2 Images Search with Relative Feedback

A search result with high NDCG is just a basic indicator of a good image search system. Thus one needs a closer look on how querying happen in order to exactly evaluate the efficiency of search systems. Spending less time for querying and retrieving is desirable. In this section, we measure the efficiency of an image search system which is defined as the required time for a querying process to obtain desired results. Such a measure is difficult to be computed because evaluating a query is mostly done implicitly and subjectively by the user. In order to measure it, relevance feedback is introduced into the querying process. By counting the amount of feedbacks for a query process to retrieve the initial mental target, the efficiency can be estimated. A system that uses less feedbacks is more preferred. As shown at the end of this section, our method achieves better evaluation result in all the three datasets SiftFlow, OSR, and Shoes.

6.6.2.1 Relative Feedback

This idea is proposed by [Kovashka 2012] and its goal is to change the way that feedback is given. Recall that conventional relevance feedback mechanisms [Zhou 2003c, Ferecatu 2007] require the user to judge on the relevance between his target and few images sampled from the retrieved result of the previous iteration ; these sampled images are used to guide the next iteration, i.e., images similar to these exemplars should appear more. With relative feedback, relevance decisions of the user are not binary but with comparison in terms of semantics. For instance, "there are less buildings in these images than my target." For such a relative feedback, we understand that in the next iteration the system should find images with more buildings. When combined with ranking algorithms, relative feedback becomes more efficient. In particular, if an image is ranked at k^{th} with respect to the semantic i, then it certainly contains more amount of the semantic i than images at $(k+1)^{\text{th}}$, $(k+2)^{\text{th}}$, etc., and less amount of semantic i than images at $(k-1)^{\text{th}}$, $(k-2)^{\text{th}}$, ..., 2^{nd}, 1^{st}. If a relative feedback states that k^{th} image has more semantic i than the target, images ranked below k^{th} are excluded from subsequent search iterations. In other word, the search space is narrowed down quickly.

6.6.2.2 Feedback Simulation

Due to limited resources [3], our experiments use randomization and ground truth in order to simulate human behaviors in querying and giving feedback. A query is

3. Experiments with user feedback are ideal if human users can anticipate the test and give "real" queries as well as feedbacks; they are more subjective, hence, reflect how mental querying happens in practice.

FIGURE 6.24 – This is an example of mental querying using relative feedback. On the leftmost is the simulated target that the user wants to find. Next, every column corresponds to two reference images randomly chosen from retrieved results of the previous iteration. For every reference image, one feedback is created based on a randomly chosen semantic. For example, the simulated target is "less sporty" and "more pointy-at-the-front" compared with the black shoe at the top row and the white shoe at the bottom row. Until the 5^{th} iteration, we can see the resemblance between the retrieved results (shoes at the rightmost column) and the simulated target.

created by randomly picking one image from the database ; this image is considered as the mental target. Its identity is of course kept secret during the querying process. However, its semantic properties are accessible for relative feedback. In order to start querying, initial examples are necessary. For this problem, 16 images – chosen randomly from the given database – define a "page zero." k images ($k = 2, 4, 8$) from page zero are randomly chosen as references ; for every reference, a semantic is randomly chosen in order to compare, in terms of membership value, with that of the corresponding semantic of the mental target. Each comparison is a relative feedback. If k references are used, then there are k feedbacks. In subsequent iterations, references will not be the 16 initial images but selected from retrieved result of the previous iteration. The search will not stop before 10 iterations unless the target is found.

As mentioned above, search efficiency is measured in terms of the average number of feedbacks required in order to find the mental target. However, this criterion does not tell us how much time is spent by the user within each iteration to spot images. It turns out that the mental target is easier to be found if it is ranked – with respect to semantic criteria accumulated from feedbacks – near to the top of the list. This is logical because the rank of the target should be raised to the top as more feedbacks are provided. As a result, search efficiency is evaluated based on the target rank versus the number of feedbacks. In order to rank retrieved results, a relevance score is maintained for every image ; this score is increased by one if the image associated with it must satisfy all the relative semantic comparisons mentioned in k feedbacks

at t^{th} iteration. The rank of the simulated mental target is reported after every feedback ; higher ranks mean the retrieval algorithm is more efficient.

6.6.2.3 Discussion

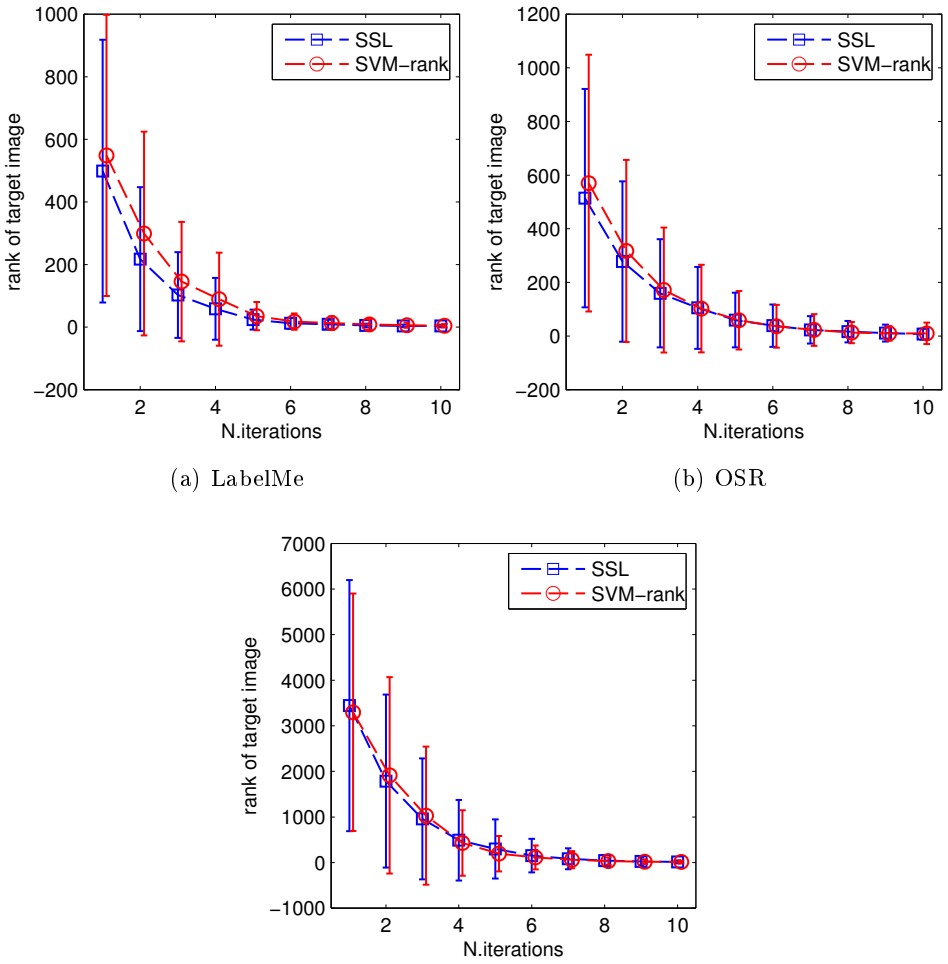
In this experiment we set the number of feedbacks per iteration $k = 2$. We randomize 100 queries and report the average ranks of the simulated targets at every iteration and show it in Fig. 6.25 ; an example is illustrated in Fig. 6.24. Our method outperforms SVM-rank on all the three datasets. This is clearly seen with LabelMe than OSR or Shoes datasets. Again, it is due to the lack of pure membership condition in both OSR and Shoes datasets. For the case of OSR dataset, our method slightly outperforms SVM-rank with OSR dataset while our ranking evaluation is worse than SVM-rank. It seems that feedback quality, not ranking quality, is the key factor of search efficiency. Due to the lack of human factor in this experiment, the visualization aspect of our method was not taken into account. The presence of visualization part may improve further search efficiency.

6.7 Summary

In this chapter we introduced a novel semantic subspace learning method ; a low-dimensional embedding is learned based on a transductive approach. The proposed formula is convex and can be solved using any generic QP solver. Two optimization algorithms are derived in order to solve this formula for large-scale problems. Based on our method, we propose an interactive search scenario in which the user seeks for his mental target by navigating in a semantic visualization of the input image database. A software with an interactive interface is also developed. Besides, our method is also useful for conventional problems such as ranking and image retrieval. Extensive empirical results on three datasets have shown that our method obtains competitive performance.

A major drawback of our method is the dense distribution of the data at the simplex barycenter. For large-scale data, this makes the visualization less clear. In order to resolve this problem, our plan is to induce sparsity into the embedding. If the data admit a sparse representation, every data point has just few memberships so that the data will no longer concentrate around the simplex barycenter.

In the long term, we can deploy the proposed method by building a complete search application. Experiments with human users are also necessary to evaluate the efficiency of this application.



(a) LabelMe

(b) OSR

(c) Shoes

FIGURE 6.25 – Evaluation of search efficiency that uses relative feedback with semantic subspace learning (SSL) and SVM-rank [Kovashka 2012] on three datasets. The vertical axis is the relevance score of the mental target ; this score is recorded at every feedback iteration as shown in the horizontal axis. As shown in these figures, the rank of the mental target is improved when more feedbacks are available, i.e., a smaller rank means that the mental target moves near to the top of the ranked list. It is easier for the user to discover his target if it is around the top.

Chapitre 7

Conclusions and Perspectives

7.1 Summary

Our research problem is to design new transductive algorithms for image interpretation and search. It is motivated by the suitability of transductive inference to computer vision problems which emerge along with the rapid growth of multimedia. Given a classification problem with training and test data drawn from an unknown distribution P, transductive classifiers transfer categorical information from the training to the provided test data but not to other unseen data. This is in opposite to inductive classifiers which learn decision rules which generalize well to arbitrary test point (also drawn from P). Our motivation in adopting transductive learning is twofold. First, training data is expensive or even scarce to obtain. Second, using both training and test data for learning may provide better results. Based on transuctive setting, we proposed algorithms for two fundamental computer vision problems : i) a novel transductive learning algorithm is proposed in order to solve binary classification problems, which are frequently used in object detection, recognition, and automatic annotation ; ii) a novel transductive subspace learning algorithm is proposed as an effective solution for semantic-based image search and ranking. A common characteristic of these problems is that each of these algorithms learns a low-dimensional representation from data. However, the former learns a linearly separable representation whose dimensionality is finite and bounded. The latter learns a representation which is semantically interpretable to human users ; in other word, the learning outcome is a manifold embedded in a finite-dimensional semantic subspace.

The thesis consists of four main chapters : Chapter 3, 4, 5 are dedicated to transductive kernel learning which are applicable to object segmentation, scene interpretation, and image annotation ; Chapter 6 is dedicated to the semantic subspace learning whose applications include data visualization, image ranking, image retrieval with relevance feedback.

7.1.1 Transductive Kernel Learning

Initially in Chapter 3, interactive object segmentation is used to validate the basic formula of transductive kernel learning. Given a test image where objects in that image are marked by the user, our algorithm finds a complete labeling, i.e., segmenting the whole shape of those marked objects and identifying them. Interpretation results of our method are compared and found to be better than related inductive

and transductive methods. We derive more sophisticated variants based on the basic formula and presented them in Chapter 5 ; performance evaluations of these variants are conducted on the scene interpretation problem – the non-interactive version of the object segmentation problem presented in Chapter 3. With this non-interactive version, training data is automatically retrieved based on a given test image so that user intervention is not required. We tested our methods with the standard scene dataset SiftFlow [Russell 2007b] ; comparison results show that our method is competitive with state of the arts.

Besides scene interpretation, the proposed transductive kernel learning is also applied to image annotation and this is presented in Chapter 4. Notice that image annotation is just one step before image search : every test image is assigned few labels reflecting semantic contents brought by that image ; searching for an image(s) having some semantic properties is equivalent to searching for an annotated image(s) with corresponding semantic labels.

There already exist many transductive methods but almost all of them follow the implicit feature mapping approach (see again Section 2.3). The transductive kernel learning is novel in the sense that it learns an explicit mapping and this mapping is linearly separable according to max-margin fashion. The advantage of explicit mapping is twofold. First, our method is model-free because we do not have to choose predefined kernels and tune their parameters. Second, it is theoretically guaranteed that a finite dimensional mapping generalizes better to data. Our idea is to factorize the input data into a basis and a kernel map with respect to the low-rank condition of the map ; furthermore, the transduction setting enforces a smoothly varying kernel map in order to diffuse labels from training to test data. The optimization procedure adopts alternating minimization ; at every iteration the kernel map to be learned is optimized conditioned on its linear discrimination with respect to a jointly learned max-margin classifier. According to the spirit of transduction adopted in our formulation, the learned kernel map and the classifier use both information provided in training and test data in order to predict more precise ; however the inference result is not available to unseen data.

The transductive kernel map method is then extended to multi-class problems with data dependency. In particular, we derive the multi-class kernel map learning in which multiple classifiers share one kernel map. Due to this, we can introduce domain knowledge into our formula so that it can model relationships between data points. Such relationships can be statistical correlations between classes. Those are similarities in languages, taxonomies, and origins ; for instance, there exist correlations between *reef* and *coral*, *Scotland* and *whiskey*, *ruin* and *Pharaoh*. Those are co-occurrences in daily life such as *window* always occurs with *building* and *sky* is above *mountain*. When incorporated into the basic formula of transductive kernel learning as convex regularizers, these modelings improve labeling results.

7.1.2 Semantic Subspace Learning

Chapter 6 is a fresh view of image search. Different from the annotation-based search methods introduced Chapter 4, we proposed a novel mental search model in which a visualization of the image database is the interaction mean between the user and the database ; a representation is learned such that dynamics of the data are well aligned with respect to predefined semantics. Keywords are no longer used to annotate semantic contents ; instead the coordinates of every image in the representation correspond to the membership values of the semantics presented in that image. Finding a mental target corresponds to navigate inside the visualization according to the the semantic memberships of the target.

In particular, our new transductive dimensionality reduction technique learns a low-dimensional representation given a predefined semantic subspace. The novelty lies in the transductive setting and constraints of the semantic subspace ; they bring interpretation – which is absent in unsupervised dimensionality reduction techniques – into the learned representation. About the former condition, it enforces a smooth variation of membership values from endmembers to other endmembers. About the latter condition, our algorithm requires a small amount of labeled data and we call them endmembers because they must contain exclusively one and only one semantic. Furthermore, if an image (labeled or unlabeled) contains a mix of semantics, or equivalently speaking it has memberships of several semantics, then the total membership values must equals one. It turns out that the coordinates of every image in the learned embedding are also the membership values and the data is supposed to lie on the surface of the unit simplex. As the optimization converges, the image database is mapped into points in the defined semantic coordinate system ; any mental target can be found into this space as a point whose coordinates reflect the amounts of semantics present in that target.

7.2 Discussions and Perspectives

7.2.1 Representation Learning

Targeted to either classification or visualization, our algorithms aim to learn suitable representations. This depends a lot on extracting and selecting features. The reality however is that we still know very little about the working mechanism of human visual cortex (parts of the brain served for visual perception) so that we do not know how to build descriptive and discriminative features. Currently, mostly machine learning algorithms use hand-crafted features and off-the-shelf similarity functions so that mistakes caused by visual variability (intra-class variation, object camouflage, uneven illumination, etc.,) are unavoidable. One of promising trends of machine learning is how to design flexible representation learning techniques that depend less on domain-specific knowledge engineering. In order to achieve this flexibility, learning methods are often based on generic priors. These priors, as analyzed in [Bengio 2012] and Chapter 2, include smoothness, natural clustering, and mani-

fold.

Smoothness Prior

The smoothness prior is so essential that it is present in almost learning algorithms, especially non-parametric ones, i.e., SVM [Cortes 1995], MRF [Ephraim 1989]. In the case of SVMs, prediction value $\hat{y}$ of a data point $\mathbf{x}$ is obtained based on the interpolation $\hat{y} = \sum_i \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$ where $\{\mathbf{x}_i\}$ are a subset of training samples (also called support vectors). If the kernel $\kappa(\cdot, \cdot)$ is the Gaussian RBF (its graph is a bell-shaped one) then the similarity between $\mathbf{x}$ and $\mathbf{x}_i$ exponentially decreases as the distance between $\mathbf{x}$ and $\mathbf{x}_i$ increases. If $\mathbf{x}$ is near to $\mathbf{x}_i$, then the prediction should smoothly vary from y_i to $\hat{y}$, i.e., $y_i \approx \hat{y}$.

Kernel machines use the smoothness prior by default and this is also true for other non-parametric methods. When the amount of training data is insufficient, semi-supervised learning such as Laplacian SVM [Belkin 2006] uses unlabeled data [1] as a bridge between training to test data for the smoothness prior to be held. In these cases, the smoothness prior guarantees smooth predictions across the data (see Section 2.4.2).

Non-parametric methods however suffer from the curse of dimensionality [Scholkopf 2001b, Lee 2007]. Addressed in [Chapelle 2006b] and more recent in [Bengio 2012], the smoothness prior just well behaves in low-dimensional spaces because a high-dimensional volume is too spacious for the data to be considered as "smooth." That is why we stated earlier in Chapter 3 that a finite-dimensional representation may provide better generalization. Besides restricting dimensionality of the learned representation so that (local) smoothness still holds, adopting *non-local learning* [Bengio 2005] is another way to defeat the curse of dimensionality. Deep learning [2] realizes this idea by learning features from data using deep neural networks.

Althought not mentioned in [Bengio 2012] as a solution to circumvent the curse of dimensionality, however, the diffusion process [Lafon 2004] is also a way to incorporate non-local information into learning. By diffusing label information from training to test data, prediction of a test point is computed based on not only close-by training data but also remote ones. As shown in optimization algorithms, our method also includes this prior. This may explain why our method is more performant than LapSVM in all the experiments of Chapter 3 and 4.

Cluster Prior

Previously stated in Chapter 2, the cluster prior is about natural partition of data into groups. The notion of group is defined as an area with relative higher density of data distribution than the surrounding. The cluster prior moves beyond the smoothness prior because it considers not only locality but also natural formations

1. In the case of transduction, unlabeled data is also the test data.
2. http://deeplearning.net/

of data (as known in Chapter 2, points belong the the same cluster are likely getting the same labels). Based on this prior, Transductive SVM [Vapnik 1977] obtains better results than Laplacian SVM [Belkin 2006] in the automatic image annotation task (see Chapter 4) ; however, Transductive SVM is not better than Laplacian SVM in the interactive object segmentation task (see Chapter 3) because the amount of data in every segmentation problem is just few hundreds so that it is insufficient in order to form clusters.

Manifold Prior

The prior assumes that there exists an underlying low-dimensional structure that generates the data (see Section 2.5.3). Due to properties of manifolds, which are the low dimensionality and the smoothness, manifold learning techniques find their use mainly in data visualization, for example Isomap [Tenenbaum 2000], LLE [Roweis 2000], LE [Belkin 2001], SNE [Hinton 2002] and our work in Chapter 6. In the study of [Weinberger 2006], they pointed out that manifold learning does not tend to benefit classification. Coordinate-free is another disadvantage of manifold learning because there is no explicit way to map new data into learned manifolds. While Nystrom extension (see Section 2.3.2) can provide such a mapping, it is data-dependent thus unstable.

Deep Learning

These approaches [Bengio 2009, Bengio 2012] learn semantic representations of images from raw data based on semantic hierarchy of images. In other word, this prior is the abstraction of semantic concepts across multiple levels : low-level features such as pixel intensity, color, textures at the lowest level are aggregated into more meaningful visual parts and objects at higher levels. Based on this prior, deep learning methods are able to learn representations which are at least as good as hand-crafted ones. Currently deep models have achieved state of the art in many problems including object recognition [Krizhevsky 2012]. However, deep learning has a burden of hyperparameters to be tuned and determining network architectures is like an art.

7.2.2 Learning by Transduction : Revisited

As introduced in Chapter 2 and 3, learning by transduction concerns predicting values of an implicit decision function at particular test points ; induction does more than that : it learns that decision function so that the prediction for a test point is available everywhere. Semi-supervised learning, in the other hand, uses unlabeled data in order to improve the generalization of the decision function to be learned. There are technical differences between (inductive) semi-supervised and (inductive) supervised methods but induction and transduction are two different philosophical point of views.

As stated in Chapter 3, given sufficient amount of test data, then transduction is supposed to exploit knowledge about probability distribution of data. The necessary condition of transduction is the presence of more than one test point. More test data provide more information about their distribution. This is beneficial especially when labeled data is scarce. However, this is not the only situation where transduction is better than induction. The availability of test data is a good opportunity to exploit its special structure, for instance the dependencies between data points. Recall that data dependency is often domain-specific and not considered in conventional statistical learning methods (see Section 2.1) ; nevertheless, we demonstrated in Chapter 4 and 5 how transduction exploits data dependency from test data.

7.2.3 Semantic Endmembership

A problem of semantic subspace learning algorithm is the endmember condition. Just for a reminder, this condition states that labeled examples must contain only one semantic and semantics should be uncorrelated one from another. It turns out from our experiments that this condition guarantees a good visualization ; however it is not easy to meet in practice except of some specific data domains such as satellite images. For generic images, endmember examples are rare because they often contain several semantics and entangled in complex ways. Thus the lack of endmember examples is a major obstacle for our method to be applied widely in different domains. A solution for this problem may be like the ability to factorize semantic endmember features from training data and to use them instead of endmember examples.

7.2.4 Data Imbalance and Regularizations

Data imbalance frequently occurs in vision databases and it is not only due to extrinsic reasons in acquisition and sampling but also being an intrinsic property. For instance, *road* always appear in highway scenes but that is not true for *car* ; thus the occurrence frequency of *road* is higher than that of *car*. The impact of imbalanced data to non-parametric models – whose predictions are computed based on (a part of) training data – is even worse. This explains why our method suffers from the same problem [3].

We have tried various ways in order to reduce the effect of data imbalance. A straightforward solution is to rebalance the data, which means to modify training data (re-sampling, duplicating, pruning) such that the proportions of training data per class become equal. An alternative is to reweight training losses in optimization equations (positive and negative classes in the case of binary classification). Since the imbalance is an intrinsic property of data, those rebalancing methods are superficial. Our solution is to exploit categorical dependency from data. We focus on strong

3. The diffusion process of the optimization algorithm favors popular classes so that rare object classes are hardly detected (Chapter 4) or popular classes tend to have higher recall rates compared to less popular ones (Chapter 5).

correlations between classes, especially between a more- and less-frequent classes. For instance, correlations are modeled as co-occurrence probabilities conditioned on the spatial adjacency between two objects in the same image. A high prediction result of more frequent classes will support less-frequent ones if they are highly correlated. This modeling is applicable for any problem whose data includes dependencies.

7.3 Future Works

There are still rooms for improvement in our thesis. For the transductive kernel learning algorithm, similarity connections between data points of adjacency graph can be made more precise by using learned distance functions subject to visual classes ; it means that label information of every class is diffused on its own graph whose edge connections are customized based on the most discriminative features of that class. For the semantic subspace learning algorithm, the learned embedding may admit sparse solution with respect to the number of semantics contained in every image. The sparsity is not only the nature of image semantic but also a way to reduce the data density concentrated at the barycenter of the simplex.

For long-term goals, we believe that learning by transduction is promising in the era of big data. Instead of using inductive techniques in order to learn generalized but inflexible decision rules, we can select a subset of labeled data which are relevant to test data and invoke transductive settings. An effective transductive learning in the future may apply transduction approach and several generic priors in order to learn more powerful kernels.

Annexe A

Transductive Kernel Learning for Imbalanced Data

In Chapter 3 we introduced the transductive kernel map method and how that suffers from imbalanced data. This phenomena is also observed in multi-class problems. In Chapter 4 the label-dependency model alleviates the effect of data imbalance by introducing label correlation into the formula (4.6). In this appendix we introduce the variant of our method for binary classification problems with imbalanced data. Let us start by reminding the objective function of transductive kernel map learning

$$\begin{aligned} \min_{\mathbf{B},\mathbf{\Phi},\mathbf{w}} \quad & \underbrace{\frac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \frac{1}{2}\mathbf{w}'\left(\mathbf{I}_p + \beta\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\right)\mathbf{w}}_{\text{regularization}} + \underbrace{\frac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{1\times p} & \mathbf{w}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2}_{\text{data term}}, \\ \text{s.t} \quad & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \end{aligned} \tag{A.1}$$

where the factorization $\mathbf{X} \approx \mathbf{B\Phi}$ is due to learn an overcomplete basis $\mathbf{B} \in \mathbb{R}^{n\times p}$ (i.e., $p > n$) and a new kernel map $\mathbf{\Phi} \in \mathbb{R}^{p\times m}$. Subsequent terms in (A.1) respectively enforce the kernel map to be low-dimensional, linear separability, and smoothness. Inequality constraints in (A.1) mean that labeled data points $\mathbf{\Phi}_i$'s must be correctly classified with respect to their labels y_i's.

Let us assume that the classification problem (A.1) is imbalanced, in the following we introduce the transductive kernel learning with extra regularizers used to control the effect of imbalanced data

$$\begin{aligned} \min_{\mathbf{B},\mathbf{\Phi},\mathbf{w}} \quad & \tfrac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \tfrac{1}{2}\mathbf{w}'\left(\mathbf{I}_p + \beta\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\right)\mathbf{w} + \tfrac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{1\times p} & \mathbf{w}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2 + \\ & + \textstyle\sum_{i=\ell+1}^{m}\left(C^+\left(\mathbf{w}'\mathbf{\Phi}_i\right)_+ + C^-\left(-\mathbf{w}'\mathbf{\Phi}_i\right)_+\right) \\ \text{s.t} \quad & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \end{aligned} \tag{A.2}$$

where the hinge loss is defined as $(\cdot)_+ = \max(0,\cdot)$. Positive cost coefficients C^+ and C^- penalize a test point to be classified respectively as positive and negative class. Without loss of generality, let us assume that there are much more negative than positive training data, then in order to rebalance the labeling result of test data, C^- must be set larger than C^+. In the opposite case where negative data is much less than positive one, we set C^+ to be smaller than C^-. Return to our case, test points will move toward the positive half-space because C^+ is smaller than C^-; once they left the negative half-space and goes into the other half-space, they will be penalized

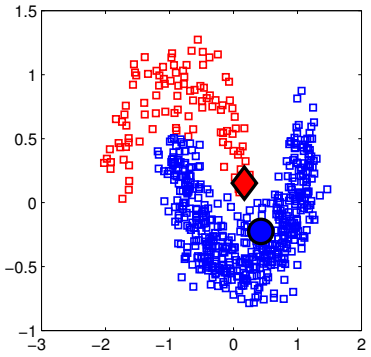


FIGURE A.1 – A toy data example of imbalanced data ; enlarged markers denote labeled data while smaller square dots denote unlabeled ones. The labeling result (shown in colors) demonstrates that (A.4) can balance test data.

by a cost C^+ in order to prevent them moving "too far." The larger C^- than C^+ is, the more test points are labeled as positive (see Fig. A.2).

In order to make (A.2) easier to solve, hinge losses are replaced by two-halve parabols of the form $y = \varphi(x)\, x^2$ (see Fig. A.1) in which the coefficient $\varphi(x)$ is defined as

$$\varphi(x) = \begin{cases} \frac{C^+}{2} & \text{iff } x > 0 \\ \frac{C^-}{2} & \text{iff } x < 0 \\ 0 & \text{iff } x = 0 \end{cases}. \tag{A.3}$$

The new formula is rewritten as follows :

$$\begin{array}{ll} \underset{\mathbf{B},w,\Phi}{\operatorname{argmin}} & \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{\mu}{2}\|\mathbf{\Phi}\|_F^2 + \frac{1}{2}\mathbf{w}'\mathbf{\Phi}\Big(\beta\mathbf{L} + \mathbf{Q}\Big)\mathbf{\Phi}'\mathbf{w} + \frac{\alpha}{2}\left\|\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix} - \begin{pmatrix}\mathbf{B} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{1\times p} & \mathbf{w}'\end{pmatrix}\begin{pmatrix}\mathbf{\Phi}\\ \mathbf{\Phi C}\end{pmatrix}\right\|_F^2 . \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ \forall i = 1,\ldots,p \end{array} \tag{A.4}$$

which is very similar to the original problem (3.5) except that (A.4) has the matrix $\mathbf{Q} = \operatorname{diag}\left(\underbrace{0,\ \ldots,\ 0}_{\ell}, \varphi(\mathbf{w}'\mathbf{\Phi}_{\ell+1}), \ldots, \varphi(\mathbf{w}'\mathbf{\Phi}_m)\right)$. The optimization algorithm of (A.4) is listed in Algorithm 5 and an example with toy data is shown in Fig. A.1.

Algorithm 5 Transductive Kernel Map Learning with Balancing Constraint

Input : labeled $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{m}$, graph Laplacian $\mathbf{L}$
Initialization : compute normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{A}$ with normalized affinity matrix $\mathbf{A}$ and $\mathbf{D} = \mathbf{I}$; $t \leftarrow 0$
Repeat

1. Compute $\mathbf{Q} = \operatorname{diag}\left(\underbrace{0, \ldots, 0}_{\ell}, \varphi\left(\mathbf{w}'\mathbf{\Phi}_{\ell+1}\right), \ldots, \varphi\left(\mathbf{w}'\mathbf{\Phi}_{m}\right)\right)$
2. Update classifier

$$\mathbf{w}^{(t+1)} = \alpha\left(\mathbf{I}_p + \mathbf{\Phi}\left(\alpha\mathbf{C} + \beta\mathbf{L} + \mathbf{Q}\right)\mathbf{\Phi}'\right)^{-1}\mathbf{\Phi}\mathbf{C}\mathbf{Y}$$

3. Update basis

$$\operatorname*{argmax}_{\lambda}\left[\operatorname*{argmin}_{\mathbf{B}}\left(\frac{1}{2}\left\|\mathbf{X} - \mathbf{B}\mathbf{\Phi}\right\|_F^2 + \sum_{i=1}^{p}\lambda_i\left(\|\mathbf{B}_i\|_2^2 - 1\right)\right)\right],$$

$$\Rightarrow \mathbf{B}^{(t+1)} = \mathbf{X}\mathbf{\Phi}'\left(\mathbf{\Phi}\mathbf{\Phi}' + \operatorname{diag}(\boldsymbol{\lambda}^*)\right)^{-1}$$

4. Update kernel map $\mathbf{\Phi}^{(t+1)} = \tilde{\mathbf{\Psi}}$ where $\tilde{\mathbf{\Psi}} = \lim_{\tau \to \tau_{\max}} \mathbf{\Psi}^{(\tau)}$ and

$$\mathbf{\Psi}_i^{(\tau)} = \left(\mu\mathbf{I} + \alpha\mathbf{B}'\mathbf{B} + (\alpha\mathbf{C}_{ii} + \beta\mathbf{D}_{ii} + \mathbf{Q}_{ii})\mathbf{w}\mathbf{w}'\right)^{-1}\left[\alpha(\mathbf{B}'\mathbf{X} + \mathbf{w}\mathbf{Y}\mathbf{C}) + \beta\mathbf{w}\mathbf{w}'\mathbf{\Psi}^{(\tau-1)}\mathbf{A}\right]_i$$

Until $\left\|\mathbf{\Phi}^{(t+1)} - \mathbf{\Phi}^{(t)}\right\| \leq \varepsilon$
Output : kernel maps $\{\mathbf{\Phi}_i\}_{i=\ell+1}^{m}$ and labels $\{y_i\}_{i=\ell+1}^{m}$ where $y_i = \left(\mathbf{w}^{(t+1)}\right)'\mathbf{\Phi}^{(t+1)}$

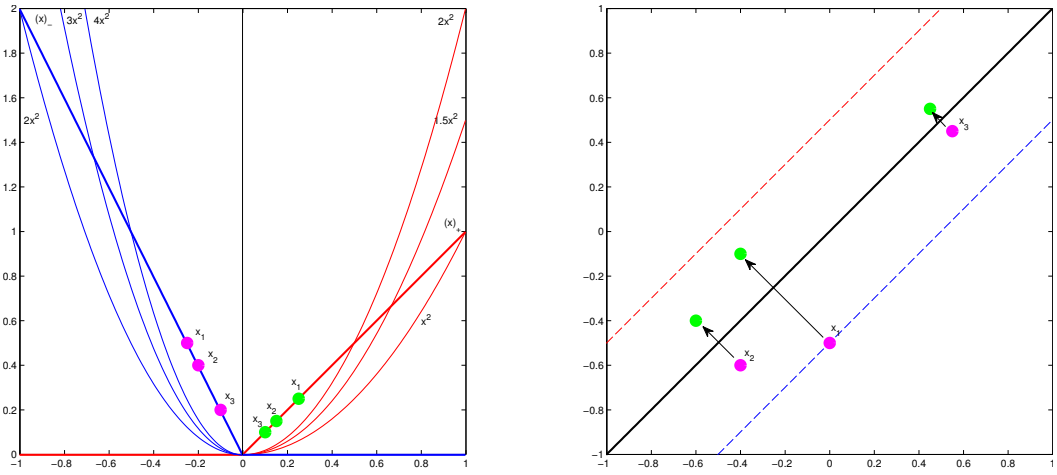


FIGURE A.2 – **Left** : the graphs of the negative (blue) and positive (red) data-rebalancing losses ; straight lines are hinge losses whose proxies are two-halves parabolas. **Right** : given three test points (colored as pink) are lying on the negative half-space ; since $C^- > C^+$, then these points tend to move toward the positive half-space after few iterations and their new positions are marked as green points.

Annexe B

Transductive Kernel Learning with Nuclear Norm

In order to induce the low-rank property into $\mathbf{\Phi}$, the nuclear norm is used :

$$\begin{array}{ll} \min\limits_{\mathbf{\Phi},\mathbf{W},\mathbf{B}} & f(\mathbf{B},\mathbf{W},\mathbf{\Phi}) + \|\mathbf{\Phi}\|_* \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ i = 1,\ldots,m \end{array} \tag{B.1}$$

in which

$$f(\mathbf{B},\mathbf{W},\mathbf{\Phi}) = \frac{\alpha}{2}\left(\|\mathbf{X} - \mathbf{B}\Phi\|_F^2 + \left\|\mathbf{Y} - \mathbf{W}'\mathbf{\Phi}\mathbf{C}\right\|_F^2\right) + \beta\mathrm{tr}(W'\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}'\mathbf{W}) + \frac{1}{2}\|\mathbf{W}\|_F^2 . \tag{B.2}$$

The above constrained optimization problem can be solved using Augmented Lagrangian Multiplier [**?**] and Singular Value Thresholding [Cai 2010] by introducing the auxiliary variable $\mathbf{J}$ and the multiplier matrix Z :

$$\begin{array}{ll} \min\limits_{\mathbf{\Phi},\mathbf{J},\mathbf{W},\mathbf{B},Z} & f(\mathbf{B},\mathbf{W},\mathbf{\Phi}) + \|\mathbf{J}\|_* + \mathrm{tr}\left(Z'(\mathbf{\Phi} - \mathbf{J})\right) + \frac{\mu}{2}\|\mathbf{\Phi} - \mathbf{J}\|_F^2 \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ i = 1,\ldots,m \end{array} \tag{B.3}$$

Alternating minimization can be used to optimize for every variable while fixing the rest. Since $\mathbf{J}$ decouples $\mathbf{\Phi}$ from the nuclear norm, the low-rank approximation problem can be easily solved without relating to the complex term $f(\mathbf{B},\mathbf{W},\mathbf{\Phi})$. The optimization algorithm is derived in the following.

Algorithm 6 j

Input : labeled $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ and unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{m}$, graph Laplacian $\mathbf{L}$
Initialization : compute normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{A}$ with normalized affinity matrix $\mathbf{A}$ and $\mathbf{D} = \mathbf{I}$; $t \leftarrow 0$
Repeat

1. Update $\mathbf{W}^{(t+1)}$:

$$\mathbf{W}^* \leftarrow \alpha \left(\mathbf{I} + \mathbf{\Phi}(\alpha\mathbf{C} + \beta\mathbf{L})\mathbf{\Phi}'\right)^{-1} (\mathbf{\Phi}\mathbf{C}\mathbf{Y}) \tag{B.4}$$

2. Update $\mathbf{J}^{(t+1)}$:

$$\mathbf{J}^* \leftarrow \operatorname{argmin} \frac{1}{\mu}\|\mathbf{J}\|_* + \frac{1}{2}\|\mathbf{J} - (\mathbf{\Phi} + Z/\mu)\|_F^2 \tag{B.5}$$

3. Update kernel map $\mathbf{\Phi}^{(t+1)} = \tilde{\mathbf{\Psi}}$ where $\tilde{\mathbf{\Psi}} = \lim_{\tau \to \tau_{\max}} \mathbf{\Psi}^{(\tau)}$, $\mathbf{\Psi}^{(0)} = \mathbf{\Phi}^{(t)}$ and

$$\begin{aligned} \mathbf{\Psi}_i^{(\tau)} \leftarrow & \left(\mu\mathbf{I} + \mathbf{B}'\mathbf{B} + [\alpha\mathbf{C} + \beta\mathbf{L}]_{ii}\mathbf{W}\mathbf{W}'\right)^{-1} \\ & \left[\alpha(\mathbf{B}'\mathbf{X} + \mathbf{W}\mathbf{Y}\mathbf{C}) + \beta\mathbf{W}\mathbf{W}'\mathbf{\Psi}^{(\tau-1)}\mathbf{A} + (\mu\mathbf{J} - Z)\right]_i \end{aligned} \tag{B.6}$$

4. check the convergence conditions

$$\|\mathbf{\Phi} - \mathbf{J}\|_\infty < \varepsilon \tag{B.7}$$

5. update the multipliers

$$Z = Z + \mu(\mathbf{\Phi} - \mathbf{J}) \tag{B.8}$$

6. update the parameters $\mu := \mu\rho$ where ρ is step size.
7. update basis $\mathbf{B}$ by

$$\begin{array}{ll} \underset{\mathbf{B}}{\operatorname{argmin}} & \frac{1}{2}\|\mathbf{X} - \mathbf{B}\mathbf{\Phi}\|_F^2 \\ \text{s.t} & \|\mathbf{B}_i\|_2^2 = 1,\ i = 1, \ldots, m \end{array} \tag{B.9}$$

Until $\left\|\mathbf{\Phi}^{(t+1)} - \mathbf{\Phi}^{(t)}\right\| \leq \varepsilon$
Output : kernel maps $\{\mathbf{\Phi}_i\}_{i=\ell+1}^{m}$ and labels $\{y_i\}_{i=\ell+1}^{m}$ where $y_i = \left(\mathbf{w}^{(t+1)}\right)' \mathbf{\Phi}^{(t+1)}$

Annexe C

More on the Convergence of Kernel Map Optimization

Proposition C.0.1 *Let hyperplane* $\mathbf{w}$ *classifies training data* $\{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ *at a margin* ρ*, then the upper bound for* β *to satisfy the Proposition 1 is*

$$\beta < \frac{8\rho^2}{m} \tag{C.1}$$

where m *is the total number of training and test data.*

Proof. As mentioned in Proposition 1, the iterated function $\psi(\cdot)$ convergences if $\beta < \|\mathbf{w}\mathbf{w}'\|_1^{-1}\|\mathbf{A}\|_1^{-1}$. We first relate $\|\mathbf{w}\mathbf{w}'\|_1$ to the margin concept. According to the definition of margin in max-margin classification methods such as SVM, ρ is proportional to the inverse of $\|\mathbf{w}\|_2$ (see (D.7) in Appendix D), i.e. $\|\mathbf{w}\|_2 = \frac{1}{2\rho}$. The following inequality always holds

$$\|\mathbf{w}\mathbf{w}'\|_1 = \sum_{i,j=1}^{p} \|w_i w_j\|_1 > \sum_{i=1}^{p} w_i^2 = \|\mathbf{w}\|_2^2 = \frac{1}{4\rho^2}. \tag{C.2}$$

Given that the affinity matrix $\mathbf{A}$ is normalized so that the corresponding degree matrix $\mathbf{D}$ is an identity matrix, then

$$\|\mathbf{A}\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{A}_{ij} = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{\mathcal{N}_k(i)} \mathbf{A}_{ij} = \frac{1}{2}\sum_{i=1}^{m} \mathbf{D}_{ii} = \frac{m}{2}. \tag{C.3}$$

Combining (C.2) and (C.3) into (3.9), we obtain the bound

$$\beta < \frac{1}{\|\mathbf{w}\|_1} \cdot \frac{1}{\|\mathbf{A}\|_1} < 4\rho^2 \cdot \frac{2}{m} = \frac{8\rho^2}{m} \tag{C.4}$$

□

Even though the bound (C.1) derived in Proposition C.0.1 is looser than what mentioned in Proposition 3.3.1, nevertheless it provides insightful interpretations about relationships between the smoothness, the max-margin classification, and data quantity.

First, (C.1) states that the smoothness is proportional to the margin width; if the margin is large, then the smoothness can be enforced more into the kernel learning process in order to speed up the convergence speed; if the margin is small,

the smoothness should be controlled in order to keep the monotonical decrease of $\psi(\cdot)$, which guarantees the convergence of the kernel map (L-Lipschitz continuity in which $L < 1$). This could be comprehensibly understood by observing the evolution of kernel maps in the toy example Fig. 3.2 while adjusting β.

Second, (C.1) reveals interesting facts about the behavior of kernel map learning when introducing more data. If m is large, then β should be small ; this makes sense because more data mean more connections between data points which boost the label diffusion process ; therefore β should be smaller in order to guarantee the kernel learning convergences with L-Lipschitz continuity in which $L < 1$.

Notice that the margin ρ is not a fixed quantity in our formulation because it can be controlled via adjusting the value of the coefficient α. For instance, if α is small while minimizing $\mathbf{w}$ and keeping $\mathbf{B}_i$'s at unit magnitudes, kernel maps at training points are more likely to move further from the hyperplane (due to the minimization of the squared loss on the training data), which widens the margin ρ. In brief, parameter tuning for α and β can be done systematically via the smoothness bound (C.1).

Annexe D

Support Vector Machine

Primal SVM

Given a set of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ drawn from an unknown distribution P in the joint domain $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} = \{-1, +1\}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ associated in which $\mathcal{H}$ is some high-dimensional space, our goal is to learn a linear classifier $f(\phi(\mathbf{x})) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$ that separates negative (-1) points from positive $(+1)$ points while generalizing well to unseen data (also drawn from P). The classifier is characterized by the normal vector $\mathbf{w}$ and the intercept b. Thus our objective corresponds to searching for a tuple $(\mathbf{w}, b)$ satisfying the following inequality constraint

$$y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 0, \; i = 1, \ldots, n. \tag{D.1}$$

Shown in Fig. D.1(a) are different solutions of the separating hyperplane f satisfying (D.1). Since there is more than one solution, our question is "whether there is exist a solution that optimally generalizes to unseen data." Support Vectors Machine (SVM) [Vapnik 1998b] is such a solution. The idea of SVM is to find a hyperplane that maximally separates the positive from the negative data ; the maximal separation is expressed by the margin concept, denoted as ρ (see Fig. D.1). Based on the margin concept, the inequality constraints (D.1) are rewritten as follows

$$y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq \rho, \quad \forall i = 1, \ldots, n. \tag{D.2}$$

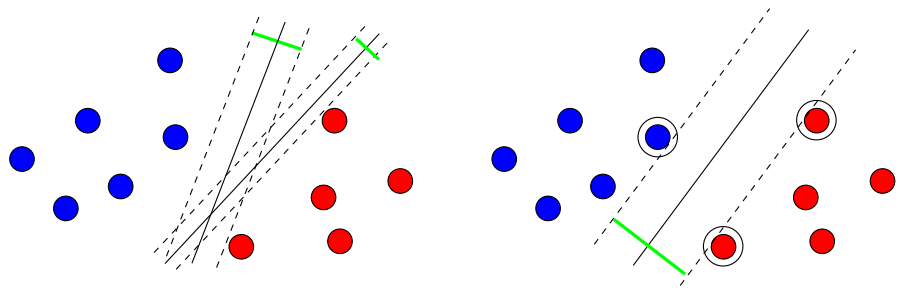
By increasing ρ, the uncertainty of predictions is decreased. Since the value of ρ cannot be larger than half of the geometric distance between the two convex hulls of positive and negative samples, then the margin ρ is guaranteed to be bounded (given $\mathbf{w}$ and b are not set to infinite). SVM aims to find the optimal minimizers $\mathbf{w}$ and b, given that ϕ is fixed, such that ρ is maximized ; its objective is equivalent to the following optimization problem

$$(\mathbf{w}^*, b^*) \leftarrow \arg \max_{\mathbf{w}, b, \rho \geq 0} \quad \rho \quad \text{s.t.} \quad y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq \rho, \; i = 1, \ldots, n, \; . \tag{D.3}$$

Since ρ can be made arbitrarily large by increasing $\mathbf{w}$ to infinity, we prevent this by normalizing $\mathbf{w}$ to have an unit magnitude :

$$\max_{\mathbf{w}, b, \rho \geq 0} \quad \rho^2 \quad \text{s.t.} \quad y_i \left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \phi(\mathbf{x}_i) \right\rangle + b \right) \geq \rho, \; i = 1, \ldots, n, \; . \tag{D.4}$$

Notice in (D.4) that the energy term ρ is replaced by ρ^2 without changing the original objective function. Since $\mathbf{w}$ and b are not constrained in their magnitudes,



(a) Multiple solutions for linear separation. (b) Maximal margin separation.

FIGURE D.1 – The classification problem considered by linear SVM. Blue dots are negative examples ; red dots are positive examples ; encircled dots are support vectors ; the solid lines are the decision boundary ; the dashed lines are margins and the green lines denote margin widths.

we can rescale them with an amount of $\frac{1}{\rho}$ in order to eliminate the margin variable ρ,

$$\max_{\mathbf{w},b,\rho\geq 0} \quad \rho^2 \quad \text{s.t.} \quad y_i \left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|\rho}, \phi(\mathbf{x}_i) \right\rangle + \frac{b}{\rho} \right) \geq 1,\ i = 1, \ldots, n, \tag{D.5}$$

or equivalently with the substitutions $\tilde{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|\rho}$ and $\tilde{b} = \frac{b}{\rho}$,

$$\min_{\tilde{\mathbf{w}},\tilde{b}} \quad \|\tilde{\mathbf{w}}\|^2 \quad \text{s.t.} \quad y_i \left(\langle \tilde{\mathbf{w}}, \phi(\mathbf{x}_i) \rangle + \tilde{b} \right) \geq 1,\ i = 1, \ldots, n. \tag{D.6}$$

The equivalence between (D.6) and (D.5) is based on the following fact

$$\|\tilde{\mathbf{w}}\| = \left\| \frac{\mathbf{w}}{\|\mathbf{w}\|\,\rho} \right\| = \frac{1}{|\rho|} \cdot \left\| \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\| = \frac{1}{\rho}. \tag{D.7}$$

Consequently, (D.3) is equivalent to

$$(\mathbf{w}^*, b^*) \leftarrow \arg \min_{\mathbf{w},b} \quad \tfrac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 1,\ i = 1, \ldots, n. \tag{D.8}$$

Dual SVM

In order to solve (D.1), one uses Lagrange multiplier method in order to transform it into the dual form which is easier to solve ; the dual form allows us to apply kernel trick (see Appendix E) in order to apply SVM to nonlinear data. The dual form of (D.1) is

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^{n} \left(\alpha_i - \alpha_i y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \right) \tag{D.9}$$

in which α_i's are non-negative Lagrange multipliers and the number of multipliers equals to that of inequality constraints. Since L is convex with respect to $\mathbf{w}$ and b and the derivatives $\partial_{\alpha_i} L = 0$, (D.8) is equivalent to the following maximization problem conditioned on the vanishing points of the derivatives $\partial_{\mathbf{w}} L$ and $\partial_b L$:

$$\max_{\alpha_i} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle \quad \text{s.t.} \quad \alpha_i \geq 0, \forall i \ , \tag{D.10}$$

where

$$\partial_{\mathbf{w}} L = \mathbf{w} - \sum_i \alpha_i y_i \phi(\mathbf{x}_i) = 0 \Longleftrightarrow \mathbf{w} = \sum_i \alpha_i y_i \phi(\mathbf{x}_i), \tag{D.11}$$

and

$$\partial_b L = \sum_i \alpha_i y_i = 0. \tag{D.12}$$

The meaning of (D.11) is that $\mathbf{w}^*$ is the weighted combination of the *support vectors* $\phi(\mathbf{x}_i)$'s whose α_i's are non-zero.

From Section 2.3 and Appendix E, we know that kernel methods allows us to replace the dot-product $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ in (D.10) by the kernel value $\kappa(\mathbf{x}_i, \mathbf{x}_j)$. As a result,(D.10) can be rewritten as

$$\max_{\alpha_i} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad \text{s.t.} \quad \alpha_i \geq 0, \, i = 1, \ldots, n. \tag{D.13}$$

Similarly, the decision function of SVM can also be written using the kernel notion

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \right\rangle + b = \sum_{i=1}^{n} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b. \tag{D.14}$$

This function exhibits an important property of SVM that it is a non-parametric method in which a subset of training data is selected based on two conditions max-margin and data separation. This subset is kept for future prediction of unseen data. The decision is computed based on local similarities imposed by kernel $\kappa(\cdot, \cdot)$ between a test point $\mathbf{x}$ and support points $\mathbf{x}_i$'s of the selected subset.

Annexe E

Kernel Trick

As known in previous sections, the kernel trick allows non-parametric methods such as SVM to be applied to nonlinear data with an inexpensive cost. In this appendix we give a brief overview on some theoretical aspects of kernel trick. More explanation can be found in [Scholkopf 2001b].

Definition E.1 ***(Gram Matrix)*** *Given a function $\kappa : \mathcal{X}\times\mathcal{K} \to \mathbb{R}$ and a finite sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$, the square matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ whose entries $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is called the Gram matrix (of kernel matrix) with respect to $\{\mathbf{x}_i\}$'s.*

Definition E.2 ***(Positive Semi-Definite Matrix)*** *A matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ satisfying $\mathbf{x}'\mathbf{K}\mathbf{x} \geq 0,\ \forall \mathbf{x} \in \mathbb{R}^n$ is called positive (semi-)definite.*

Definition E.3 ***(Kernel)*** *A function κ on $\mathcal{X} \times \mathcal{X}$ which for all $n \in \mathbb{N}$ and all $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$ gives rise to a positive (semi-)definite Gram matrix is called a (positive semi-definite) kernel.*

In the following we define a map ϕ from $\mathcal{X}$ into the space $\mathcal{H}$ of functions mapping $\mathcal{X}$ into $\mathbb{R}$ as

$$\begin{array}{rl} \phi : & \mathcal{X} \to \mathcal{H} \\ & \mathbf{x} \mapsto \kappa(\cdot, \mathbf{x}) \end{array}, \tag{E.1}$$

then we will show that i) $\mathcal{H}$ is equipped with a dot-product and ii) any positive definite kernel $\kappa(\cdot,\cdot)$ can be thought of as a dot-product in another space : $\langle\phi(\mathbf{x}), \phi(\mathbf{z})\rangle = \kappa(\mathbf{x}, \mathbf{z})$.

Firstly, let us assume that $\mathcal{H}$ as a vector space closed with addition and multiplication

$$\mathcal{H} = \left\{ \sum_{i=1}^{n} \alpha_i \kappa(\cdot, \mathbf{x}_i) : \alpha_i \in \mathbb{R}, \mathbf{x}_i \in \mathcal{X}, n \in \mathbb{N} \right\}, \tag{E.2}$$

given $f \in \mathcal{H}$ and $g \in \mathcal{H}$, we define

$$\langle f, g\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j\kappa(\mathbf{x}_i, \mathbf{x}_j), \tag{E.3}$$

in which

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i \kappa(\cdot, \mathbf{x}_i) \tag{E.4}$$

and

$$g(\cdot) = \sum_{j=1}^{m} \beta_j \kappa(\cdot, \mathbf{z}_j). \tag{E.5}$$

We will prove that $\langle \cdot, \cdot \rangle$ is a dot-product of $\mathcal{H}$. From (E.3) and the fact that $\kappa(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{z}, \mathbf{x})$, then the dot-product $\langle \cdot, \cdot \rangle$ is symmetric. Furthermore, $\langle \cdot, \cdot \rangle$ is bilinear and positive definite because

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{z}_j) = \sum_{i=1}^{n} \alpha_i g(\mathbf{x}_i) = \sum_{j=1}^{m} \beta_j f(\mathbf{z}_j), \tag{E.6}$$

and

$$\langle f, f \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{E.7}$$

Therefore feature space $\mathcal{H}$ is associated with a dot-product $\langle \cdot, \cdot \rangle$. □

Based on (E.5), we obtain $g(\cdot) = \kappa(\cdot, \mathbf{z})$ as a result of substituting $m = 1$ and $\beta_1 = 1$; applying this result into (E.6) we obtain the *reproducing property*, i.e.,

$$\langle f, \kappa(\cdot, \mathbf{z}) \rangle = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{z}) = f(\mathbf{z}). \tag{E.8}$$

If we set $f = \kappa(\cdot, \mathbf{x})$ (assign $n = 1$ and α_1=1 in (E.4)) and $g = \kappa(\cdot, \mathbf{z})$, then (E.8) gives us the desired property :

$$\langle f, g \rangle = \langle \kappa(\cdot, \mathbf{x}), \kappa(\cdot, \mathbf{z}) \rangle \overset{\text{(C.1)}}{=} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \overset{\text{(C.3)}}{=} \kappa(\mathbf{x}, \mathbf{z}), \tag{E.9}$$

which means any positive (semi-)definite kernel can be thought of as a dot-product in another space. □

Bibliographie

[Aflalo 2011] Jonathan Aflalo, Aharon Ben-Tal, Chiranjib Bhattacharyya, Jagarlapudi Saketha Nath and Sankaran Raman. *Variable Sparsity Kernel Learning*. J. Mach. Learn. Res., vol. 12, pages 565–592, February 2011. (Cited on page 40.)

[Asa 2008] Asa, Ong Cheng Soon, Sonnenburg Sören, Schölkopf Bernhard and Rätsch Gunnar Ben-Hur. *Support Vector Machines and Kernels for Computational Biology*. PLoS Comput Biol, vol. 4, no. 10, 10 2008. (Cited on pages 8 and 40.)

[Bach 2004] Francis R. Bach, Gert R. G. Lanckriet and Michael I. Jordan. *Multiple kernel learning, conic duality, and the SMO algorithm*. In ICML, 2004. (Cited on pages 8, 27, 40 and 63.)

[Bach 2008a] Francis Bach. *Exploring Large Feature Spaces with Hierarchical Multiple Kernel Learning*. In NIPS, pages 105–112, 2008. (Cited on page 27.)

[Bach 2008b] Francis R. Bach. *Consistency of the Group Lasso and Multiple Kernel Learning*. Journal of Machine Learning Research, vol. 9, pages 1179–1225, 2008. (Cited on pages 8, 27, 40 and 63.)

[Bakir 2007] Gükhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar and S. V. N. Vishwanathan. Predicting structured data (neural information processing). The MIT Press, 2007. (Cited on pages 6 and 21.)

[Bar 2004] M. Bar. *Visual objects in context*. Nature Reviews Neuroscience, vol. 5, no. 8, pages 617–629, 2004. (Cited on page 80.)

[Bar 2005] Moshe Bar, Elissa Aminoff, Jasmine Boshyan, Mark Fenske, Nurit Gronauo and Karim Kassam. *The contribution of context to visual object recognition*. Journal of Vision, vol. 5, no. 8, page 88, 2005. (Cited on page 80.)

[Bar 2010] Leah Bar and Guillermo Sapiro. *Hierarchical dictionary learning for invariant classification*. In ICASSP, pages 3578–3581, 2010. (Cited on page 34.)

[Belkin 2001] M. Belkin and P. Niyogi. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*. In NIPS, pages 585–591, 2001. (Cited on pages 11, 37, 111 and 145.)

[Belkin 2004] M. Belkin, I. Matveeva and P. Niyogi. *Regularization and Semi-supervised Learning on Large Graphs*. In John Shawe-Taylor and Yoram Singer, editeurs, Learning Theory, volume 3120 of *Lecture Notes in Computer Science*, pages 624–638. Springer Berlin / Heidelberg, 2004. (Cited on page 31.)

[Belkin 2006] M. Belkin, P. Niyogi and V. Sindhwani. *Manifold Regularization : A Geometric Framework for Learning from Labeled and Unlabeled Examples*.

J. Mach. Learn. Res., vol. 7, pages 2399–2434, December 2006. (Cited on pages 21, 30, 31, 40, 42, 45, 56, 62, 63, 144 and 145.)

[Bengio 2005] Yoshua Bengio, Hugo Larochelle and Pascal Vincent. *Non-Local Manifold Parzen Windows.* In NIPS, 2005. (Cited on page 144.)

[Bengio 2009] Yoshua Bengio. *Learning Deep Architectures for AI.* Foundations and Trends in Machine Learning, vol. 2, no. 1, pages 1–127, 2009. (Cited on pages 6 and 145.)

[Bengio 2012] Yoshua Bengio, Aaron Courville and Pascal Vincent. *Representation Learning : A Review and New Perspectives.* Rapport technique, Department of computer science and operations research, U. Montreal, 2012. (Cited on pages 143, 144 and 145.)

[Berg 2010] Tamara L. Berg, Alexander C. Berg and Jonathan Shih. *Automatic Attribute Discovery and Characterization from Noisy Web Data.* In ECCV (1), pages 663–676, 2010. (Cited on pages 133 and 134.)

[Bertsekas 1999] D.P. Bertsekas. Nonlinear programming. Athena Scientific, 1999. (Cited on page 24.)

[Biederman 1972] Irving Biederman. *Perceiving Real-World Scenes.* Science, vol. 177, pages 77–80, 1972. (Cited on page 80.)

[Biederman 1982a] Irving Biederman. *Scene perception : detecting and judging objects undergoing relational violations.* Cognitive Psychology, vol. 14, pages 143–177, 1982. (Cited on page 2.)

[Biederman 1982b] Irving Biederman, Robert J. Mezzanotte and Jan C. Rabinowitz. *Scene perception : Detecting and judging objects undergoing relational violations.* Cognitive Psychology, vol. 14, no. 2, pages 143–177, April 1982. (Cited on pages 80 and 81.)

[Bishop 2006] Chirstopher M. Bishop. Pattern recognition and machine learning. Springer, 2006. (Cited on pages 16, 21, 22 and 84.)

[Blei 2003] David M. Blei, Andrew Y. Ng and Michael I. Jordan. *Latent Dirichlet Allocation.* Journal of Machine Learning Research, vol. 3, pages 993–1022, 2003. (Cited on page 62.)

[Boden 2006] Margeret A. Boden. Mind as machine : A history of cognitive science. Oxford University Press, Oxford, England, 2006. (Cited on page 2.)

[Cai 2007] Deng Cai, Xiaofei He and Jiawei Han. *Spectral regression : a unified subspace learning framework for content-based image retrieval.* In ACM Multimedia, pages 403–412, 2007. (Cited on page 110.)

[Cai 2010] Jian-Feng Cai, Emmanuel J.Candes and Zuouwei Shen. *A SINGULAR VALUE THRESHOLDING ALGORITHM FOR MATRIX COMPLETION.* SIAM Journal on Optimization, vol. 20, no. 4, pages 1–26, 2010. (Cited on page 153.)

[Carbonetto 2004] Peter Carbonetto, Nando de Freitas and Kobus Barnard. *A Statistical Model for General Contextual Object Recognition.* In ECCV (1), pages 350–362, 2004. (Cited on pages 82 and 85.)

[Carneiro 2007] Gustavo Carneiro, Antoni B Chan, Pedro J Moreno and Nuno Vasconcelos. *Supervised learning of semantic classes for image annotation and retrieval.* IEEE transactions on pattern analysis and machine intelligence, vol. 29, no. 3, pages 394–410, March 2007. (Cited on pages 62 and 76.)

[Chang 2011] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM : A library for support vector machines.* ACM Transactions on Intelligent Systems and Technology, vol. 2, pages 27 :1–27 :27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. (Cited on pages 26 and 72.)

[Chapelle 2002] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet and Sayan Mukherjee. *Choosing Multiple Parameters for Support Vector Machines.* Machine Learning, vol. 46, no. 1-3, pages 131–159, 2002. (Cited on pages 27 and 40.)

[Chapelle 2005] Olivier Chapelle and Alexander Zien. *Semi-Supervised Classification by Low Density Separation.* In AI STATS, 2005. (Cited on page 40.)

[Chapelle 2006a] O. Chapelle, B. Schölkopf and A. Zien, editeurs. Semi-supervised learning. MIT Press, Cambridge, MA, 2006. (Cited on pages 40, 62 and 63.)

[Chapelle 2006b] Olivier Chapelle, Bernard Scholkopf and Alexander Zien. Semi-supervised learning. The MIT Press, 2006. (Cited on pages 29 and 144.)

[Chapelle 2007] Olivier Chapelle. *Training a Support Vector Machine in the Primal.* Neural Computation, vol. 19, no. 5, pages 1155–1178, 2007. (Cited on page 24.)

[Chatpatanasiri 2010] Ratthachat Chatpatanasiri, Teesid Korsrilabutr, Pasakorn Tangchanachaianan and Boonserm Kijsirikul. *A new kernelization framework for Mahalanobis distance learning algorithms.* Neurocomputing, vol. 73, no. 10-12, pages 1570–1579, June 2010. (Cited on pages 40 and 63.)

[Chen 2010] J. Chen. *WLD : A Robust Local Image Descriptor.* PAMI, vol. 32, no. 9, 2010. (Cited on page 123.)

[Chen 2011a] Xi Chen, Arpit Jain, Abhinav Gupta and Larry S. Davis. *Piecing together the segmentation jigsaw using context.* In CVPR, pages 2001–2008, 2011. (Cited on pages 6 and 83.)

[Chen 2011b] Xiangyu Chen, Xiaotong Yuan, Shuicheng Yan, Jinhui Tang, Yong Rui and Tat-Seng Chua. *Towards multi-semantic image annotation with graph regularized exclusive group lasso.* Proceedings of the 19th ACM international conference on Multimedia - MM '11, page 263, 2011. (Cited on page 63.)

[Cherkassky 2002] Vladimir Cherkassky and Yunqian Ma. *Selection of Meta-parameters for Support Vector Regression.* In José R. Dorronsoro, editeur, ICANN, volume 2415 of *Lecture Notes in Computer Science*, pages 687–693. Springer, 2002. (Cited on page 23.)

[Chung 1997] F. R. K. Chung. Spectral graph theory. American Mathematical Society, 1997. (Cited on pages 31 and 115.)

[Coleman 1996] Thomas F. Coleman and Yuying Li. *A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on Some of the Variables.* SIAM J. on Optimization, vol. 6, no. 4, pages 1040–1058, April 1996. (Cited on page 119.)

[Cortes 1995] Corinna Cortes and Vladimir Vapnik. *Support-vector networks.* Machine Learning, vol. 20, no. 3, pages 273–297, 1995. (Cited on pages 6, 18, 21, 22, 23, 56, 62, 63, 64, 72 and 144.)

[Cortes 2009a] Corinna Cortes, Mehryar Mohri and Afshin Rostamizadeh. *L2 Regularization for Learning Kernels.* In UAI, pages 109–116, 2009. (Cited on page 40.)

[Cortes 2009b] Corinna Cortes, Mehryar Mohri and Afshin Rostamizadeh. *Learning Non-Linear Combinations of Kernels.* In NIPS, pages 396–404, 2009. (Cited on page 27.)

[Cortes 2009c] Corinna Cortes, Mehryar Mohri and Afshin Rostamizadeh. *Learning Non-Linear Combinations of Kernels.* In NIPS, pages 396–404, 2009. (Cited on page 40.)

[Cox 2004] David Cox, Ethan Meyers and Pawan Sinha. *Contextually Evoked Object-Specific Responses in Human Visual Cortex.* Science, vol. 304, no. 5667, pages 115–117, 2004. (Cited on page 80.)

[Cristianini 2001] Nello Cristianini, John Shawe-Taylor, André Elisseeff and Jaz S. Kandola. *On Kernel-Target Alignment.* In NIPS, pages 367–373, 2001. (Cited on pages 40 and 41.)

[Cristianini 2010] Nello Cristianini and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, 2010. (Cited on page 26.)

[Deng 2010] Jia Deng, Alexander C Berg, Kai Li and Li Fei-fei. *What Does Classifying More Than 10 , 000 Image Categories Tell Us ?* In ECCV, pages 71–84, 2010. (Cited on page 62.)

[Deng 2011] Jia Deng and Alexander C Berg. *Hierarchical Semantic Indexing for Large Scale Image Retrieval.* In CVPR, 2011. (Cited on page 62.)

[Draghici 1997] Sorin Draghici. *A neural network based artificial vision system for licence plate recognition.* International Journal of Neural Systems, vol. 8, no. 01, pages 113–126, 1997. (Cited on page 6.)

[Duchenne 2008] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce and F. Segonne. *Segmentation by transduction.* In IEEE Conference on Computer Vision and Pattern Recognition, 2008. (Cited on page 8.)

[Duchi 2008] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer and Tushar Chandra. *Efficient projections onto the l_1-ball for learning in high dimensions.* In ICML, pages 272–279, 2008. (Cited on pages 119 and 120.)

[Duygulu 2002] Pinar Duygulu, Kobus Barnard, J.F.G de Freitas and David A Forsyth. *Object Recognition as Machine Translation : Learning a Lexicon for a Fixed Image Vocabulary.* ECCV, 2002. (Cited on page 62.)

[Eigen 2012a] D. Eigen and R. Fergus. *Nonparametric image parsing using adaptive neighbor sets.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2799–2806, june 2012. (Cited on pages 84, 85 and 94.)

[Eigen 2012b] David Eigen and Rob Fergus. *Nonparametric image parsing using adaptive neighbor sets.* In CVPR, pages 2799–2806, 2012. (Cited on pages 6 and 105.)

[Ephraim 1989] Yariv Ephraim, Amir Dembo and Lawrence R. Rabiner. *A minimum discrimination information approach for hidden Markov modeling.* IEEE Transactions on Information Theory, vol. 35, no. 5, pages 1001–1013, 1989. (Cited on pages 62 and 144.)

[Everingham 2010] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The Pascal Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, June 2010. (Cited on page 9.)

[Fan 2004] Jianping Fan, Yuli Gao and Hangzai Luo. *Multi-level annotation of natural scenes using dominant image components and semantic concepts.* ACM, 2004. (Cited on page 62.)

[Farabet 2012] Clément Farabet, Camille Couprie, Laurent Najman and Yann LeCun. *Scene parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers.* In ICML. icml.cc / Omnipress, 2012. (Cited on page 105.)

[Farhadi 2009] Ali Farhadi, Ian Endres, Derek Hoiem and David Forsyth. *Describing Objects by their Attributes.* In CVPR, 2009. (Cited on page 62.)

[Fauqueur 2006] Julien Fauqueur and Nozha Boujemaa. *Mental image search by boolean composition of region categories.* Multimedia Tools Appl., vol. 31, pages 95–117, 2006. (Cited on page 111.)

[Felzenszwalb 2004] Pedro F Felzenszwalb and Daniel P Huttenlocher. *Efficient Graph-Based Image Segmentation.* Int. J. Comput. Vision, vol. 59, no. 2, pages 167–181, 2004. (Cited on pages 84, 93 and 94.)

[Feng 2004] Shaolei Feng, Raghavan Manmatha and Victor Lavrenko. *Multiple Bernoulli Relevance Models for Image and Video Annotation.* In CVPR (2), pages 1002–1009, 2004. (Cited on pages 62 and 76.)

[Feng 2013] Zheyun Feng, Rong Jin and Anil Jain. *Large-scale Image Annotation by Efficient and Robust Kernel Metric Learning.* In ICCV, 2013. (Cited on page 63.)

[Ferecatu 2007] Marin Ferecatu and Donald Geman. *Interactive Search for Image Categories by Mental Matching.* In ICCV, pages 1–8, 2007. (Cited on pages 108, 111 and 136.)

[Fergus 2009] Rob Fergus, New York and Antonio Torralba. *Semi-supervised Learning in Gigantic Image Collections.* In NIPS, pages 1–9, 2009. (Cited on page 63.)

[Fink 2003] Michael Fink and Pietro Perona. *Mutual Boosting for Contextual Inference.* In NIPS. MIT Press, 2003. (Cited on page 83.)

[Fletcher 1987] R. Fletcher. Practical methods of optimization ; (2nd ed.). Wiley-Interscience, New York, NY, USA, 1987. (Cited on page 24.)

[Galleguillos 2008] C. Galleguillos, A. Rabinovich and S. Belongie. *Object categorization using co-occurrence, location and appearance.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, 2008. (Cited on page 83.)

[Gao 2010] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia and Peilin Zhao. *Local features are not lonely - Laplacian sparse coding for image classification.* In CVPR, pages 3555–3561, 2010. (Cited on page 34.)

[Gehler 2008] Peter Gehler and Sebastian Nowozin. *Infinite kernel learning.* 2008. (Cited on page 27.)

[Golub 1996] Gene H. Golub and Charles F. Van Loan. Matrix computations. The Johns Hopkins University Press, 3rd édition, 1996. (Cited on page 44.)

[Gosselin 2008] Philippe Henri Gosselin and Matthieu Cord. *Active Learning Methods for Interactive Image Retrieval.* IEEE Transactions on Image Processing, vol. 17, no. 7, pages 1200–1211, 2008. (Cited on page 108.)

[Gould 2008] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan and Daphne Koller. *Multi-Class Segmentation with Relative Location Prior.* Int. J. Comput. Vision, vol. 80, no. 3, pages 300–316, December 2008. (Cited on page 83.)

[Grandvalet 2002] Yves Grandvalet and Stéphane Canu. *Adaptive Scaling for Feature Selection in SVMs.* In NIPS, pages 553–560, 2002. (Cited on page 40.)

[Grangier 2008] David Grangier and Samy Bengio. *A Discriminative Kernel-Based Approach to Rank Images from Text Queries.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 8, pages 1371–1384, 2008. (Cited on pages 63, 70 and 76.)

[Guillaumin 2009] Matthieu Guillaumin, Thomas Mensink, Jakob J. Verbeek and Cordelia Schmid. *TagProp : Discriminative metric learning in nearest neighbor models for image auto-annotation.* In ICCV, pages 309–316, 2009. (Cited on pages 63, 74 and 76.)

[Hariharan 2010] Bharath Hariharan, Lihi Zelnik-Manor, S. V. N. Vishwanathan and Manik Varma. *Large Scale Max-Margin Multi-Label Classification with Priors.* In ICML, pages 423–430, 2010. (Cited on pages 72, 75 and 76.)

[He 2004] Xiaofei He. *Incremental semi-supervised subspace learning for image retrieval.* In ACM Multimedia, pages 2–8, 2004. (Cited on page 110.)

[He 2009] Haibo He and Edwardo A. Garcia. *Learning from Imbalanced Data.* IEEE Trans. on Knowl. and Data Eng., vol. 21, no. 9, pages 1263–1284, September 2009. (Cited on pages 64, 71 and 72.)

[Heesch 2008] Daniel Heesch. *A survey of browsing models for content based image retrieval.* Multimedia Tools Appl., vol. 40, no. 2, pages 261–284, 2008. (Cited on pages 11, 108 and 111.)

[Hertz 1991] John Hertz, Richard G. Palmer and Anders S. Krogh. Introduction to the theory of neural computation. Perseus Publishing, 1st édition, 1991. (Cited on page 22.)

[Hinton 2002] Geoffrey E. Hinton and Sam T. Roweis. *Stochastic Neighbor Embedding*. In NIPS, pages 833–840, 2002. (Cited on page 145.)

[Hofmann 1999] Thomas Hofmann. *Probabilistic Latent Semantic Analysis*. In UAI, pages 289–296, 1999. (Cited on page 62.)

[Horn 1990a] Roger A. Horn and Charles R. Johnson. Matrix analysis. Cambridge University Press, 1990. (Cited on pages 33 and 37.)

[Horn 1990b] Roger A. Horn and Charles R. Johnson. Matrix analysis, chapter 5. Cambridge University Press, 1990. (Cited on page 44.)

[Hubel 1988] David H. Hubel. Eye, brain, and vision. W H Freeman & Co, NewYork, 1988. (Cited on page 2.)

[Jain 2009] Prateek Jain, Brian Kulis, Jason V Davis and Inderjit S Dhillon. *Metric and Kernel Learning using a Linear Transformation*. October, 2009. (Cited on page 40.)

[Jain 2010] Arpit Jain, Abhinav Gupta and Larry S. Davis. *Learning what and how of contextual models for scene labeling*. In Proceedings of the 11th European conference on Computer vision : Part IV, ECCV'10, pages 199–212, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on page 84.)

[Jeon 2003] Jiwoon Jeon, Victor Lavrenko and R. Manmatha. *Automatic image annotation and retrieval using cross-media relevance models*. In SIGIR, pages 119–126, 2003. (Cited on page 76.)

[Joachims 1999] T. Joachims. *Transductive Inference for Text Classification using Support Vector Machines*. In ICML, pages 200–209, 1999. (Cited on pages 8, 30, 40, 56, 63, 73, 75 and 76.)

[Joachims 2002a] T. Joachims. Learning to classify text using support vector machines – methods, theory, and algorithms. Kluwer/Springer, 2002. (Cited on pages 8, 40 and 63.)

[Joachims 2002b] Thorsten Joachims. *Optimizing search engines using clickthrough data*. In KDD, pages 133–142, 2002. (Cited on pages 109 and 132.)

[Joachims 2003] Thorsten Joachims. *Transductive Learning via Spectral Graph Partitioning*. In In ICML, pages 290–297, 2003. (Cited on page 8.)

[Jolliffe 1986] I. T. Jolliffe. Principal component analysis. Springer, New York, 1986. (Cited on page 32.)

[Kindermann 1980] Ross Kindermann, James Laurie Snell*et al.* Markov random fields and their applications, volume 1. American Mathematical Society Providence, RI, 1980. (Cited on page 6.)

[Kovashka 2012] Adriana Kovashka, Devi Parikh and Kristen Grauman. *WhittleSearch : Image search with relative attribute feedback*. In CVPR, pages 2973–2980, 2012. (Cited on pages 108, 132, 133, 134, 135, 136 and 139.)

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks.* In NIPS, pages 1106–1114, 2012. (Cited on pages 6 and 145.)

[Kulis 2010] Brian Kulis and U C Berkeley Eecs. *Inductive Regularized Learning of Kernel Functions.* In NIPS, numéro x, pages 1–9, 2010. (Cited on pages 40 and 63.)

[Kumar 2005] Sanjiv Kumar and Martial Hebert. *A Hierarchical Field Framework for Unified Context-Based Classification.* In ICCV, pages 1284–1291, 2005. (Cited on pages 83 and 85.)

[Kumar 2006] Sanjiv Kumar and Martial Hebert. *Discriminative Random Fields.* International Journal of Computer Vision, vol. 68, no. 2, pages 179–201, 2006. (Cited on page 85.)

[Kumar 2009] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur and Shree K. Nayar. *Attribute and simile classifiers for face verification.* In ICCV, pages 365–372, 2009. (Cited on pages 109 and 111.)

[Kyrillidis 2012] Anastasios T. Kyrillidis, Stephen Becker and Volkan Cevher. *Sparse projections onto the simplex.* CoRR, vol. abs/1206.1529, 2012. (Cited on pages 119 and 120.)

[Ladicky 2009] Lubor Ladicky, Christopher Russell, Pushmeet Kohli and Philip H. S. Torr. *Associative hierarchical CRFs for object class image segmentation.* In ICCV, pages 739–746, 2009. (Cited on page 83.)

[Lafon 2004] Stephane Lafon. *Diffusion Maps and Geometric Harmonics.* PhD thesis, Yale University, 2004. (Cited on pages 31 and 144.)

[Lanckriet 2004a] Gert R G Lanckriet, Peter Bartlett and Michael I Jordan. *Learning the Kernel Matrix with Semidefinite Programming.* Journal of Machine Learning Research, vol. 5, pages 27–72, 2004. (Cited on pages 41 and 63.)

[Lanckriet 2004b] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui and Michael I. Jordan. *Learning the Kernel Matrix with Semidefinite Programming.* Journal of Machine Learning Research, vol. 5, pages 27–72, 2004. (Cited on page 40.)

[Lavrenko 2003] Victor Lavrenko, R. Manmatha and Jiwoon Jeon. *A Model for Learning the Semantics of Pictures.* In NIPS, 2003. (Cited on pages 62 and 76.)

[Lawrence 1997] Steve Lawrence, C Lee Giles, Ah Chung Tsoi and Andrew D Back. *Face recognition : A convolutional neural-network approach.* Neural Networks, IEEE Transactions on, vol. 8, no. 1, pages 98–113, 1997. (Cited on page 6.)

[Lazebnik 2006a] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories.* In CVPR (2), pages 2169–2178, 2006. (Cited on page 27.)

[Lazebnik 2006b] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories.* In CVPR (2), pages 2169–2178. IEEE Computer Society, 2006. (Cited on page 94.)

[Lee 2006] Honglak Lee, Alexis Battle, Rajat Raina and Andrew Y. Ng. *Efficient sparse coding algorithms.* In NIPS, pages 801–808, 2006. (Cited on page 47.)

[Lee 2007] John A. Lee and Michel Verleysen. Nonlinear dimensionality reduction. Springer, 2007. (Cited on pages 30, 34 and 144.)

[Li 2009] L-J. Li, R. Socher and L. Fei-Fei. *Towards Total Scene Understanding :Classification, Annotation and Segmentation in an Automatic Framework.* In Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 2009. (Cited on page 85.)

[Li 2010] Zechao Li, Jing Liu, Xiaobin Zhu, Tinglin Liu and Hanqing Lu. *Image Annotation Using Multi-Correlation Probabilistic Matrix Factorization.* In ACM Multimedia, pages 10–13, 2010. (Cited on page 62.)

[Lin 2005] Yen-Yu Lin, Tyng-Luh Liu and Hwann-Tzong Chen. *Semantic manifold learning for image retrieval.* In ACM Multimedia, pages 249–258, 2005. (Cited on pages 110 and 132.)

[Liu 2009a] Jing Liu, Mingjing Li, Qingshan Liu, Hanqing Lu and Songde Ma. *Image annotation via graph learning.* Pattern Recognition, vol. 42, no. 2, pages 218–228, 2009. (Cited on page 76.)

[Liu 2009b] Wei Liu and Shih-Fu Chang. *Robust multi-class transductive learning with graphs.* In CVPR, pages 381–388. IEEE, 2009. (Cited on page 8.)

[Liu 2010] Dong Liu, Shuicheng Yan, Yong Rui and Hong-Jiang Zhang. *Unified Tag Analysis With Multi-Edge Graph.* In ACM Multimedia, pages 25–34, 2010. (Cited on pages 68, 74 and 76.)

[Liu 2011a] Ce Liu, Jenny Yuen and Antonio Torralba. *Nonparametric Scene Parsing via Label Transfer.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pages 2368–2382, 2011. (Cited on pages 82, 85 and 94.)

[Liu 2011b] Yang Liu, Yan Liu, Sheng hua Zhong and Keith C. C. Chan. *Semi-supervised manifold ordinal regression for image ranking.* In ACM Multimedia, pages 1393–1396, 2011. (Cited on pages 110 and 132.)

[Livingstone 2008] Margaret S. Livingstone. Vision and Art : The Biology of Seeing. Abrams, 2008. (Cited on page 2.)

[Ma 2011] Zhigang Ma and Jasper Uijlings. *Exploiting the Entire Feature Space with Sparsity for Automatic Image Annotation.* In ACM Multimedia, pages 283–292, 2011. (Cited on page 63.)

[Mairal 2008] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro and Andrew Zisserman. *Supervised Dictionary Learning.* In NIPS, pages 1033–1040, 2008. (Cited on page 34.)

[Mairal 2009] Julien Mairal, Francis Bach, Jean Ponce and Guillermo Sapiro. *Online dictionary learning for sparse coding.* In Andrea Pohoreckyj Danyluk, Léon Bottou and Michael L. Littman, editeurs, ICML, volume 382 of *ACM International Conference Proceeding Series*, page 87. ACM, 2009. (Cited on page 47.)

[Maji 2008] S. Maji, A-C. Berg and J. Malik. *Classification using intersection kernel support vector machines is efficient.* In CVPR, 2008. (Cited on pages 8, 40 and 56.)

[Maji 2013] Subhransu Maji, Alexander C. Berg and Jitendra Malik. *Efficient Classification for Additive Kernel SVMs.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pages 66–77, 2013. (Cited on pages 27 and 28.)

[Makadia 2008] Ameesh Makadia, Vladimir Pavlovic and Sanjiv Kumar. *A New Baseline for Image Annotation.* In ECCV (3), pages 316–329, 2008. (Cited on pages 63, 72, 74 and 76.)

[Malisiewicz 2008] Tomasz Malisiewicz and Alexei A. Efros. *Recognition by Association via Learning Per-exemplar Distances.* In CVPR, June 2008. (Cited on page 52.)

[Malisiewicz 2009] Tomasz Malisiewicz and Alexei A. Efros. *Beyond Categories : The Visual Memex Model for Reasoning About Object Relationships.* In NIPS, pages 1222–1230, 2009. (Cited on pages 84 and 94.)

[Marr 1982] David Marr. Vision : A computational investigation into the human representation and processing of visual information. Henry Holt and Co., Inc., New York, NY, USA, 1982. (Cited on page 2.)

[Marvin 1969] Minsky Marvin and Seymour Papert. Perceptrons. Oxford, 1969. (Cited on page 22.)

[Mazumder 2009] Rahul Mazumder, Jerome Friedman and Trevor Hastie. *SparseNet : Coordinate Descent with Non-Convex Penalties*, 2009. (Cited on page 47.)

[McCulloch 1943] WarrenS. McCulloch and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115–133, 1943. (Cited on page 6.)

[Melacci 2011] Stefano Melacci and Mikhail Belkin. *Laplacian Support Vector Machines Trained in the Primal.* J. Mach. Learn. Res., pages 1149–1184, July 2011. (Cited on pages 63, 73, 75 and 76.)

[Mensink 2011] Thomas Mensink, Jakob J. Verbeek and Gabriela Csurka. *Learning structured prediction models for interactive image labeling.* In CVPR, 2011. (Cited on page 62.)

[Metzger 2006] Wolfgang Metzger. Laws of seeing. The MIT Press, September 2006. (Cited on page 2.)

[Metzler 2004] Donald Metzler and R. Manmatha. *An Inference Network Approach to Image Retrieval.* In CIVR, pages 42–50, 2004. (Cited on page 76.)

[Mundy 2006] Joseph L. Mundy. *Object Recognition in the Geometric Era : A Retrospective.* In Toward Category-Level Object Recognition, pages 3–28, 2006. (Cited on page 4.)

[Munoz 2010] Daniel Munoz, J. Andrew Bagnell and Martial Hebert. *Stacked Hierarchical Labeling.* In ECCV (6), pages 57–70, 2010. (Cited on page 84.)

[Murphy 2003] K. Murphy, A. Torralba and W.T. Freeman. *Using the forest to see the trees : a graphical model relating features, objects and scenes.* Advances in Neural Information Processing Systems, vol. 16, 2003. (Cited on page 83.)

[Murphy 2004] Gregory Murphy. The big book of concepts. The MIT Press, January 2004. (Cited on page 2.)

[Myeong 2012] Heesoo Myeong, Ju Yong Chang and Kyoung Mu Lee. *Learning object relationships via graph-based context model.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2727–2734, june 2012. (Cited on pages 84, 85 and 105.)

[Narayanan 2006] Hariharan Narayanan, Mikhail Belkin and Partha Niyogi. *On the Relation Between Low Density Separation, Spectral Clustering and Graph Cuts.* In NIPS, pages 1025–1032, 2006. (Cited on page 40.)

[Oliva 2001a] Aude Oliva and Antonio Torralba. *Modeling the Shape of the Scene : A Holistic Representation of the Spatial Envelope.* International Journal of Computer Vision, vol. 42, no. 3, pages 145–175, 2001. (Cited on page 133.)

[Oliva 2001b] Aude Oliva and Antonio Torralba. *Modeling the Shape of the Scene : A Holistic Representation of the Spatial Envelope.* International Journal of Computer Vision, vol. 42, no. 3, pages 145–175, 2001. (Cited on page 134.)

[Oliva 2006] A. Oliva and A. Torralba. *Building the Gist of a Scene : The Role of Global Image Features in Recognition.* Visual Perception, Progress in Brain Research, vol. 155, 2006. (Cited on pages 82 and 94.)

[Olshausen 1997] B. A. Olshausen and D. J. Field. *Sparse coding with an overcomplete basis set : a strategy employed by V1 ?* Vision Res, vol. 37, pages 3311–25, 1997. (Cited on pages 21 and 33.)

[Ong 2005] Cheng Soon Ong, Alexander J. Smola and Robert C. Williamson. *Learning the Kernel with Hyperkernels.* Journal of Machine Learning Research, vol. 6, pages 1043–1071, 2005. (Cited on page 40.)

[Parikh 2008] Devi Parikh, C. Lawrence Zitnick and Tsuhan Chen. *From appearance to context-based recognition : Dense labeling in small images.* In CVPR, 2008. (Cited on pages 83 and 84.)

[Picard 2010] David Picard, Nicolas Thome and Matthieu Cord. *An efficient system for combining complementary kernels in complex visual categorization tasks.* In ICIP, pages 3877–3880. IEEE, 2010. (Cited on page 27.)

[Rabinovich 2007a] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie. *Objects in Context.* In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8, 2007. (Cited on page 84.)

[Rabinovich 2007b] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora and Serge Belongie. *Objects in Context.* In ICCV, pages 1–8, 2007. (Cited on pages 83, 84 and 85.)

[Rakotomamonjy 2008] Alain Rakotomamonjy and Francis R Bach. *SimpleMKL.* Journal of Machine Learning Research, pages 1–34, 2008. (Cited on pages 8, 27, 40, 56, 57 and 63.)

[Rakotomamonjy 2011] Alain Rakotomamonjy, Rémi Flamary, Gilles Gasso and Stéphane Canu. *ell$_p$-ell$_q$ Penalty for Sparse Linear and Sparse Multiple Kernel Multitask Learning.* IEEE Transactions on Neural Networks, vol. 22, no. 8, pages 1307–1320, 2011. (Cited on page 40.)

[Ramírez 2010] Ignacio Ramírez, Pablo Sprechmann and Guillermo Sapiro. *Classification and clustering via dictionary learning with structured incoherence and shared features.* In CVPR, pages 3501–3508, 2010. (Cited on page 34.)

[Riesenhuber 1999] Maximilian Riesenhuber and Tomaso Poggio. *Hierarchical models of object recognition in cortex.* Nature Neuroscience, 1999. (Cited on page 6.)

[Roberts 1963] Lawrence G. Roberts. Machine perception of three-dimensional solids. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963. (Cited on page 4.)

[Rosenblatt 1958] Frank Rosenblatt. *The perceptron : A probabilistic model for information storage and organization in the brain.* Psychological Review, vol. 65, 1958. (Cited on pages 21 and 22.)

[Roweis 2000] S-T. Roweis and L-K. Saul. *Nonlinear Dimensionality Reduction by Locally Linear Embedding.* Science, vol. 290, pages 2323–2326, 2000. (Cited on pages 11, 36, 111 and 145.)

[Rowley 1996] Henry A. Rowley, Shumeet Baluja and Takeo Kanade. *Neural Network-Based Face Detection.* In CVPR, pages 203–208. IEEE Computer Society, 1996. (Cited on page 6.)

[Rubner 2001] Y. Rubner and C. Tomasi. Perceptual Metrics for Image Database Navigation. Springer, 2001. (Cited on pages 11, 13, 108 and 111.)

[Russakovsky 2010a] Olga Russakovsky and Li Fei-fei. *Attribute learning in large-scale datasets.* In ECCV, 2010. (Cited on page 62.)

[Russakovsky 2010b] Olga Russakovsky and Fei-Fei Li. *Attribute Learning in Large-Scale Datasets.* In ECCV Workshops, pages 1–14, 2010. (Cited on page 109.)

[Russell 2007a] Bryan C. Russell, Antonio Torralba, Ce Liu, Robert Fergus and William T. Freeman. *Object Recognition by Scene Alignment.* In John C. Platt, Daphne Koller, Yoram Singer and Sam T. Roweis, editeurs, NIPS. Curran Associates, Inc., 2007. (Cited on pages 85, 93, 94 and 105.)

[Russell 2007b] Bryan C. Russell, Antonio Torralba, Ce Liu, Robert Fergus and William T. Freeman. *Object Recognition by Scene Alignment.* In NIPS, 2007. (Cited on page 142.)

[Rutishauser 2004] U. Rutishauser, D. Walther, C. Koch and P. Perona. *Is Bottom-Up Attention Useful for Object Recognition.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 11–37, 2004. (Cited on page 82.)

[Schaefer 2010] G. Schaefer. *A next generation browsing environment for large image repositories.* Multimedia Tools and Applications, vol. 47, pages 105–120, 2010. (Cited on pages 11 and 111.)

[Schölkopf 2001a] B. Schölkopf and AJ. Smola. Learning with kernels : Support vector machines, regularization, optimization, and beyond. The MIT Press, December 2001. (Cited on pages 40 and 42.)

[Scholkopf 2001b] Bernard Scholkopf and Alexander J. Smola. Learning with kernels. The MIT Press, 2001. (Cited on pages 6, 8, 21, 24, 25, 26, 65, 144 and 161.)

[Shawe-Taylor 2004] John Shawe-Taylor and Nello Cristianini. Kernel methods for pattern analysis. Cambridge University Press, New York, NY, USA, 2004. (Cited on pages 6, 25, 26, 33 and 65.)

[Shotton 2009] Jamie Shotton, John M. Winn, Carsten Rother and Antonio Criminisi. *TextonBoost for Image Understanding : Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context.* International Journal of Computer Vision, vol. 81, no. 1, pages 2–23, 2009. (Cited on pages 82 and 85.)

[Siddiquie 2011] Behjat Siddiquie, Rogério Schmidt Feris and Larry S. Davis. *Image ranking and retrieval based on multi-attribute queries.* In CVPR, pages 801–808, 2011. (Cited on pages 109, 111 and 132.)

[Singh 2013] Gautam Singh and Jana Kosecka. *Nonparametric Scene Parsing with Adaptive Feature Relevance and Semantic Context.* In CVPR, pages 3151–3157. IEEE, 2013. (Cited on pages 94, 105 and 106.)

[Singhal 2003] Amit Singhal, Jiebo Luo and Weiyu Zhu. *Probabilistic Spatial Context Models for Scene Content Understanding.* In CVPR (1), pages 235–241. IEEE Computer Society, 2003. (Cited on page 85.)

[Sonnenburg 2006] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer and Bernhard Schölkopf. *Large Scale Multiple Kernel Learning.* Journal of Machine Learning Research, vol. 7, pages 1531–1565, 2006. (Cited on pages 8, 40 and 63.)

[Spence 1999] Robert Spence. *A framework for navigation.* Int. J. Hum.-Comput. Stud., vol. 51, pages 919–945, November 1999. (Cited on page 111.)

[Subrahmanya 2010] N. Subrahmanya and Y.C. Shin. *Sparse Multiple Kernel Learning for Signal Processing Applications.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 5, pages 788–798, 2010. (Cited on pages 21 and 27.)

[Sutton 2012] Charles A. Sutton and Andrew McCallum. *An Introduction to Conditional Random Fields*. Foundations and Trends in Machine Learning, vol. 4, no. 4, pages 267–373, 2012. (Cited on page 6.)

[Tenenbaum 2000] J-B. Tenenbaum, V. de Silva and J-C. Langford. *A Global Geometric Framework for Nonlinear Dimensionality Reduction*. Science, vol. 290, no. 5500, pages 2319–2323, 2000. (Cited on pages 11, 35, 111 and 145.)

[Tighe 2010] J. Tighe and S. Lazebnik. *SuperParsing : Scalable Nonparametric Image Parsing with Superpixels*. In ECCV (5), pages 352–365, 2010. (Cited on pages 51, 52, 83, 84, 94, 105 and 106.)

[Tighe 2013] Joseph Tighe and Svetlana Lazebnik. *Superparsing - Scalable Nonparametric Image Parsing with Superpixels*. International Journal of Computer Vision, vol. 101, no. 2, pages 329–349, 2013. (Cited on pages 6, 85, 91, 94 and 105.)

[Torralba 2003] Antonio Torralba. *Contextual Priming for Object Detection*. International Journal of Computer Vision, vol. 53, no. 2, pages 169–191, 2003. (Cited on page 85.)

[Torralba 2011] Antonio Torralba and Alexei A. Efros. *Unbiased look at dataset bias*. In CVPR, pages 1521–1528. IEEE, 2011. (Cited on page 71.)

[Tsai 2011] David Tsai, Yushi Jing, Yi Liu, Henry A. Rowley, Sergey Ioffe and James M. Rehg. *Large-scale image annotation using visual synset*. In ICCV, 2011. (Cited on pages 62 and 64.)

[Ulges 2011] Adrian Ulges, Marcel Worring and Thomas M. Breuel. *Learning Visual Contexts for Image Annotation From Flickr Groups*. IEEE Transactions on Multimedia, vol. 13, no. 2, 2011. (Cited on pages 62 and 64.)

[van de Sande 2010] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek. *Evaluating Color Descriptors for Object and Scene Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pages 1582–1596, 2010. (Cited on page 68.)

[Vapnik 1977] V. Vapnik and A. Sterin. *On structural risk minimization or overall risk in a problem of pattern recognition*. Automation and Remote Control, vol. 10, no. 3, pages 1495–1503, 1977. (Cited on pages 29, 30, 40, 43, 62 and 145.)

[Vapnik 1998a] V. Vapnik. Statistical learning theory. Wiley, New York, 1998. (Cited on pages 40, 41, 42, 43, 56, 72, 75 and 76.)

[Vapnik 1998b] Vladimir Vapnik. Statistical learning theory. Wiley, 1998. (Cited on pages 17, 18, 20, 65 and 157.)

[Varma 2009] Manik Varma and Bodla Rakesh Babu. *More generality in efficient multiple kernel learning*. In ICML, page 134, 2009. (Cited on pages 40 and 63.)

[Vedaldi 2012] A. Vedaldi and A. Zisserman. *Efficient Additive Kernels via Explicit Feature Maps*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, no. 3, pages 480–492, 2012. (Cited on page 28.)

[Vieux 2011] Remi Vieux, Jenny Benois-Pineau, Jean-Philippe Domenger and Achille Braquelaire. *Segmentation-based multi-class semantic object detection.* Multimedia Tools and Applications, pages 1 – 22, March 2011. 22. (Cited on page 83.)

[Vishwanathan 2010a] Bharath Hariharan S V N Vishwanathan. *Efficient Max-Margin Multi-Label Classification with Applications to Zero-Shot Learning.* Rapport technique, Microsoft Research Technical Report MSR-TR-2010-141, 2010. (Cited on pages 72, 75 and 76.)

[Vishwanathan 2010b] S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt and M. Varma. *Multiple Kernel Learning and the SMO Algorithm.* In NIPS, 2010. (Cited on page 63.)

[von Luxburg 2008] Ulrike von Luxburg and Bernhard Schoelkopf. *Statistical Learning Theory : Models, Concepts, and Results.* Rapport technique, Max Planck Institute for Biological Cybernetics, 2008. (Cited on pages 16, 17 and 18.)

[Wallach 2004] Hanna M Wallach. *Conditional random fields : An introduction.* Technical Reports (CIS), page 22, 2004. (Cited on page 6.)

[Wang 2009] Changhu Wang, Shuicheng Yan, Lei Zhang and Hong jiang Zhang. *Multi-label sparse coding for automatic image annotation.* 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1643–1650, June 2009. (Cited on pages 63, 74 and 76.)

[Weinberger 2004] Kilian Q Weinberger and Lawrence K Saul. *Learning a kernel matrix for nonlinear dimensionality reduction.* Machine Learning, no. July, 2004. (Cited on page 41.)

[Weinberger 2006] Kilian Q. Weinberger and Lawrence K. Saul. *An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding.* In AAAI, pages 1683–1686, 2006. (Cited on page 145.)

[Williams 2000] Christopher K. I. Williams and Matthias Seeger. *The Effect of the Input Density Distribution on Kernel-based Classifiers.* In ICML, pages 1159–1166, 2000. (Cited on page 27.)

[Wolf 2006] Lior Wolf and Stanley M. Bileschi. *A Critical View of Context.* International Journal of Computer Vision, vol. 69, no. 2, pages 251–261, 2006. (Cited on pages 81, 82 and 83.)

[Wu 2006] Mingrui Wu, Bernhard Schölkopf and Gokhan Bakir. *A Direct Method for Building Sparse Kernel Learning Algorithms.* Journal of Machine Learning Research, vol. 7, pages 603–624, 2006. (Cited on pages 8, 40 and 63.)

[Xue 2011] Xiangyang Xue, Wei Zhang 0016, Jie Zhang, Bin Wu, Jianping Fan and Yao Lu. *Correlative multi-label multi-instance image annotation.* In ICCV, 2011. (Cited on page 62.)

[Yuan 2011] Ying Yuan, Fei Wu, Yueting Zhuang and Jian Shao. *Image Annotation by Composite Kernel Learning with Group Structure.* In ACM Multimedia, pages 1497–1500, 2011. (Cited on page 63.)

[Zhang 2010] Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li and Dimitris N. Metaxas. *Automatic image annotation using group sparsity.* In CVPR, pages 3312–3319, 2010. (Cited on pages 74 and 76.)

[Zheng 2011] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu and Deng Cai. *Graph Regularized Sparse Coding for Image Representation.* IEEE Transactions on Image Processing, vol. 20, no. 5, pages 1327–1336, 2011. (Cited on page 34.)

[Zhou 2003a] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston and Bernhard Schölkopf. *Learning with Local and Global Consistency.* In NIPS, 2003. (Cited on page 40.)

[Zhou 2003b] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet and Bernhard Schölkopf. *Ranking on Data Manifolds.* In NIPS, 2003. (Cited on pages 110 and 132.)

[Zhou 2003c] Xiang Sean Zhou and Thomas S. Huang. *Relevance feedback in image retrieval : A comprehensive review.* Multimedia Syst., vol. 8, no. 6, pages 536–544, 2003. (Cited on pages 108 and 136.)

Transductive Inference for Image Interpretation and Search

Abstract :

Inference transductive pour l'interpretation et la recherche d'images

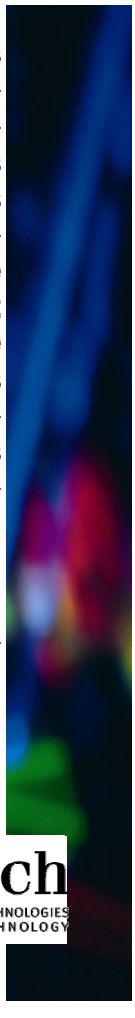
Dinh-Phong VO

RESUME : Dans cette thèse, on s'intéresse à l'apprentissage automatique pour traiter deux problèmes fondamentaux en vision par ordinateurs. Le premier concerne l'interprétation d'images qui consiste à classer des images ou des objets en catégories. Les techniques classiques sont généralement inductives et exigent des données d'apprentissage étiquetées afin d'apprendre explicitement des classifieurs. Dans certaines applications, les données d'apprentissage étiquetées sont rares ce qui affecte les capacités de généralisation des classifieurs sous-jacents. Dans cette thèse on s'intéresse à l'apprentissage transductif qui vise à estimer la réponse d'un classifieur implicite sur un ensemble fini incluant à la fois les données d'apprentissage et de test.

On présente d'abord un nouveau cadre d'apprentissage transductif des noyaux pour l'interprétation des images. Cette méthode, contrairement aux noyaux classiques, apprend une projection explicite des noyaux, en exploitant la topologie des données d'apprentissage et de test. Le problème d'optimisation sous-jacent vise à minimiser une énergie mélangeant i) un terme de reconstruction, qui décompose une matrice des données en un produit impliquant un dictionnaire et une nouvelle représentation liée au noyau appris, ii) un terme d'attache aux données qui assure la consistance des étiquettes inférées par rapport à celles des données d'apprentissage et iii) un terme de régularisation qui garantit des étiquettes similaires pour des données semblables. La représentation du noyau et le critère de décision obtenus garantissent la séparabilité linéaire des données et de bonnes performances de généralisation. En partant de cette formulation, on propose une extension qui permet d'exploiter les dépendances contextuelles et les liens sémantiques entre les catégories d'images afin d'améliorer encore plus les performances de notre méthode d'annotation et d'interprétation des images. Cette extension a été motivée par des expériences en psychologie, qui montrent que les informations contextuelles sont essentielles et permettent de faciliter la reconnaissance d'objets chez les humains.

Le deuxième problème abordé dans la thèse concerne la recherche mentale dans les bases d'images. Au départ, on rappelle les limites des paradigmes de recherche classiques (basés sur les mots clés, exemples visuels et requêtes par croquis) dans l'interprétation des requêtes mentales des utilisateurs ; notamment lorsque les cibles mentales des utilisateurs sont difficiles à exprimer avec des mots clés ou lorsque les exemples des requêtes ne sont pas disponibles. La solution alternative proposée construit une représentation qui préserve la topologie globale des données en les projetant dans un espace Euclidien exprimé à travers une base sémantique. L'avantage de la méthode est double ; d'une part elle permet de réduire significativement la dimension des données, et d'autre part, la méthode permet de définir une nouvelle représentation des données qui est plus facile à exploiter par l'utilisateur afin de retrouver sa cible. Ainsi, retrouver une cible mentale revient simplement à scanner et pointer les données selon leurs coordonnées dans l'espace sémantique appris. Les expériences effectuées en visualisation, ordonnancement et recherche d'images avec contrôle de pertinence, sur des bases génériques, montrent que l'approche proposée est effective.

MOTS-CLEFS : cartes du noyaux, l'apprentissage transductive, classification, réduction de la dimensionnalité, visualisation



TELECOM ParisTech

EDITE

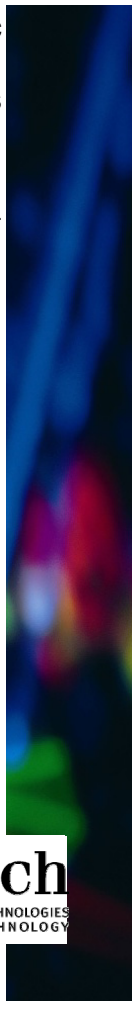
ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

ABSTRACT : In this thesis, we use machine learning in order to tackle two fundamental problems of computer vision. The first one is image interpretation which consists in classifying images and objects into categories. Conventional inductive learning models require some training data from which classifiers are learned. If training data is scarce, classifiers hardly generalize well to test data. We are interested in transductive learning - the approach that aims to estimate the response of an implicit classifier at particular test points using both training and test data.

We first introduce a new transductive kernel learning framework for image interpretation. Our method, in contrast to many usual kernels, learns an explicit kernel map based on topological structure of both training and test data. The underlying optimization problem minimizes an energy function mixing i) a reconstruction term that decomposes a matrix of input data as a product of a learned dictionary and a kernel map ii) a fidelity term that ensures consistent label predictions with respect to those provided by training data and iii) a smoothness term which guarantees similar labels for neighboring data. The resulting decision criterion and the new kernel map guarantee the linear separability of training data and good generalization performance. Based on this formulation, we also study how to harness contextual dependencies between categories into images and how to use their semantic relationships during inference in order to further improve image annotation and scene understanding performances. This extension was motivated by experiments in psychology, which have shown that contextual information includes important cues for human vision in order to recognize objects effortlessly.

The second fundamental problem is mental search ; we address the limitation of current multimedia search paradigms (based on keywords, image examples, and sketches) in interpreting mental targets of users, especially if those targets are difficult to express verbally or visual examples are not ready to hand. We introduce a novel alternative solution which builds a mapping that preserves the global topology of the input data while associating them into an Euclidean subspace spanned by well defined semantics. The advantage of the method is twofold. On the one hand, it significantly reduces the dimensionality of the data ; on the other hand, it defines a new data representation which is more friendly and easy to use. Thereby, searching for a mental target simply reduces to scanning and targeting data according to their coordinates in the learned semantic subspace. Quantitative evaluations in data visualization, image ranking and retrieval with relevance feedback, using generic image databases, show that the proposed method is effective.

KEY-WORDS : kernel maps, transductive learning, classification, dimensionality reduction, visualization



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY