Apprentissage de données structurées pour l'interprétation d'images

Structured Data Learning for Image Interpretation

Emanuel Aldea

Man's mind and spirit grow with the space in which they are allowed to operate.

(Krafft Ehricke, rocket engineer and space visionary)

Contents

1 Introduction **27**

2 Statistical learning theory and structured data **33**
2.1 Introduction . . . 33
2.2 Support Vector Machines . . . 34
2.2.1 Empirical Risk Minimization and Support Vector Machines . . . 34
2.2.2 A Geometrical Interpretation . . . 36
2.2.3 Solving the SVM . . . 38
2.3 Kernels and Reproducing Kernel Hilbert Spaces . . . 39
2.4 Characteristics of good kernels . . . 42
2.5 Alternatives to SVMs in machine learning . . . 42
2.6 Structured Data . . . 43
2.7 Convolution and Marginalized Kernels . . . 45
2.8 A Direct Application to Graph Learning . . . 46
2.8.1 Graphical Representations . . . 46
2.8.2 A Marginalized Kernel Based On Random Walks . . . 47

3 Towards More Expressiveness in Graph Labeling **53**
3.1 Kernel improvement on limited label set graphs . . . 53
3.2 Continuous Labels and More Complex Kernels for Vertex and Edge Features 55
3.3 Benefits and alternative solutions in graph kernel theory . . . 57
3.4 Graph matching vs. graph kernels using random walks . . . 57
3.4.1 Discrete kernels and discrete values . . . 61
3.4.2 Continuous kernels and discrete values . . . 62
3.4.3 Continuous kernels and continuous values . . . 63
3.5 Conclusions . . . 64

4 Graphical Models In Image Representation **67**
4.1 Introduction . . . 67
4.2 High-level semantic image retrieval . . . 70
4.2.1 Dealing with the semantic gap . . . 70
4.2.2 The task we address, and our approach . . . 70
4.3 The image model . . . 72

4.3.1 Image segmentation . . . 72
4.3.2 Alternative region detectors . . . 74
4.4 The region model . . . 74
4.4.1 Related work . . . 75
4.4.2 Our position within the structured representation framework . . . 76
4.5 Description of keypoint model constituents . . . 77
4.5.1 Basic descriptors . . . 78
4.5.2 Color . . . 78
4.5.3 Texture . . . 78
4.5.4 Shape . . . 80
4.5.5 Hybrid approaches . . . 81
4.6 Applying the marginalized graph kernel to image representations . . . 82
4.6.1 Unsupervised segmentation and graph extraction . . . 82
4.6.2 Label kernels . . . 85
4.6.3 Fast computation of the graph kernel . . . 86
4.6.4 Categorization performance . . . 87
4.6.5 Categorization with more than one feature . . . 88
4.7 Conclusions . . . 89

5 Fuzzy Spatial Relations and Graphical Models 91
5.1 Motivation for the use of fuzzy spatial relations . . . 91
5.1.1 Fuzzy information as an extension of binary relations . . . 91
5.1.2 Ambivalence of spatial relations . . . 92
5.2 Representation of objects by spatial fuzzy sets . . . 93
5.3 Image representations and fuzzy spatial relations . . . 94
5.3.1 Distance between regions . . . 94
5.3.2 Fuzzy adjacency . . . 95
5.3.3 Directional relative position. . . . 95
5.4 Fuzzy feature measures . . . 96
5.4.1 Adjacency measure based on fuzzy satisfiability . . . 97
5.4.2 Adjacency measure based on fuzzy resemblance . . . 98
5.5 Fuzzy spatial information and discriminative models . . . 98
5.5.1 The information gain . . . 99
5.5.2 Graph pruning . . . 99
5.6 Experiments and results . . . 101
5.6.1 Data set . . . 101
5.6.2 Experimental setup . . . 102
5.6.3 Categorization with strict adjacency structures . . . 103
5.6.4 Categorization with fuzzy adjacency structures . . . 107
5.6.5 Fuzzy measures of adjacency and frequent low values . . . 107
5.7 Discussion . . . 112
5.8 Conclusion . . . 112

6 Automated Visual Feature Selection Using Multiple Kernel Learning 117
6.1 Motivation for multiple kernel learning 117
6.2 Our approach using the multiple kernel learning framework 118
6.3 The multiple kernel learning formalism 118
6.4 MKL, structured data and relations 121
6.5 Experiments and results 122
6.6 Conclusion 126

7 Conclusion and perspectives 127

Abstract

Image interpretation methods use primarily the visual features of low-level or high-level interest elements. However, spatial information concerning the relative positioning of these elements is equally beneficial, as it has been shown previously in segmentation and structure recognition. Therefore, an interest for the integration of spatial information in the learning framework has emerged recently. The fact that spatial information is often perceived and expressed in a manner which is close to natural language, along with the fact that the absence of a spatial interaction is also relevant, hint at the usefulness of fuzzy spatial information for image representation. Fuzzy representations actually permit to assess at the same time the imprecision degree of a relation (e.g., "close to" or "to the left of") and the gradual transition between the satisfiability and the non-satisfiability of a relation. Among the solutions used to adapt image data to algorithm inputs, we adopt a representation structure which encodes explicitly image parts and spatial interactions in a graphical model.

The objective of this work is to explore techniques of spatial information representation and their integration in the learning process, within the context of image classifiers that make use of graph kernels. We motivate our choice of labeled graphs for representing images, in the context of learning with SVM classifiers. Graph kernels have been studied intensively in computational chemistry and the study of biologic systems, and an adaptation for image related graphs is necessary, since image structures and properties of the information encoded in the labeling are fundamentally different. We illustrate the integration of spatial information within the image graphical model by considering fuzzy adjacency measures between interest elements (regions), and we define a family of graph representations determined by different thresholds applied to these spatial measures. Finally, we employ multiple kernel learning methods in order to build up classifiers that can take into account different graphical representations of the same image at once. The results show that spatial information complements the visual features of distinctive elements in images and that adapting the discriminative kernel functions for the fuzzy spatial representations is beneficial in terms of performance.

Acknowledgments

My first and warmest acknowledgment goes to my advisor Isabelle Bloch, who constantly helped in ensuring my academic and moral well being since the very beginning of our collaboration. In every sense, none of this work would have been possible without her support and inspirational example, both as remarkable scientist and kind mentor.

I would like to thank Marinette Revenu and Alain Rakotomamonjy for having accepted to review this thesis and giving me valuable comments. Thank you also to Jean-Michel Jolion and Matthieu Cord for having made me the honour to take part in the jury.

I am extremely grateful to Jamal Atif who co-supervised my Master internship before leaving for Guyana where I also spent a month during my PhD, following his invitation. I take this opportunity to thank all the people from the Institute for Reasearch and Developement of Cayenne for their feedback, consideration and hospitality.

Far too many people to mention individually have assisted and surrounded me in so many ways during my work. They all have my sincere gratitude. In particular, I would like to thank Geoffroy Fouquier, Olivier Nempont, Jérémie Anquez, David Lesage, Camille Lemen, Nicolas Widynski, Carolina Vanegas, Julien Wojak, Pierre Blanchart, Darwin Martinez, all currently, or previously, of Télécom-ParisTech. I would also like to thank Henri Maître, Hichem Sahbi, Jérôme Darbon, Marine Campedel, Saïd Ladjal and several anonymous reviewers for their help and comments that improved various papers.

I also owe a huge debt of gratitude to many professors who nurtured and polished my interest for science, especially Jean-Éric Pin, Gilles Dowek and Philippe Baptiste. I am indebted as well to all the amazing students I encountered during my TA at Université Paris 6 and who waited diligently for the class to get a bit more interesting each day :-)

A final and wholehearted thank-you goes to Voichita, my parents, my brother and my old friends from Paris for always being there when I needed them most, for accepting me as I am, and for their support and encouragement throughout my journey.

Résumé long en français

Introduction

Le sujet principal de cette thèse est l'apprentissage de données structurées pour l'interprétation d'images, avec la prise en compte des interactions spatiales qui ont lieu entre les éléments détectés dans les images.

Contexte d'application

L'intérêt pour ce sujet dans le domaine du traitement d'images porte sur le fait que les méthodes de représentation de données et d'apprentissage classiques ne prennent pas naturellement en compte les images. La plupart des méthodes de classification s'appuient en premier lieu sur les attributs des objets d'intérêt dans les images. Cependant, les informations spatiales liées aux relations entre ces objets sont également utiles, comme cela a été montré en segmentation et reconnaissance de structures dans les images, et leur intégration dans des méthodes d'apprentissage et de classification commence à apparaître et évoluer. Le fait que ces informations soient souvent exprimées sous forme linguistique, et que l'absence d'une relation puisse constituer également une information utile pour la classification, suggère l'utilité des relations spatiales floues pour la modélisation d'images. En effet les modélisations floues permettent de représenter à la fois l'imprécision de la relation (par exemple *proche* ou *à gauche*) et le passage graduel de la satisfaction à la non satisfaction de cette relation.

Un deuxième point d'intérêt concernant l'intégration du raisonnement spatial est le fait que l'information spatiale implique la manipulation de concepts qui dépassent un niveau visuel rudimentaire. Les méthodes utilisant l'information spatiale doivent ainsi faire face à la divergence entre la perception humaine et la vision artificielle de bas niveau, un phénomène généralement appelé le fossé sémantique.

Finalement, les approches structurées en interprétation d'images imposent l'utilisation de données complexes sous la forme d'arbres ou de graphes, et le problème du compromis entre la richesse de la représentation et les performances en termes de complexité computationnelle est toujours d'actualité.

Travaux et contributions

La thèse explore des méthodes proposées en apprentissage supervisé pour apprendre à partir de données structurées, en particulier des graphes d'attributs flous relationnels. Les nœuds représentent par exemple des objets que l'on cherche à segmenter dans des images ou à classifier, et les arcs modélisent une ou plusieurs relations spatiales (adjacence, distance, direction relative, topologie, ...). Dans ce contexte, et en particulier pour les applications en interprétation d'images et en raisonnement spatial, de nombreuses sources d'imprécision doivent être prises en compte : sur les objets, à cause de leurs limites imprécises et des éventuelles difficultés à les segmenter, sur les relations (typiquement la relation "proche de" est intrinsèquement vague et dépend du contexte et de l'échelle des objets), et sur les connaissances manipulées, souvent exprimées sous forme linguistique. Ces structures modélisent par exemple les informations issues de l'analyse d'une image médicale, comme la nature des composantes (structures anatomiques, pathologies, etc.), les propriétés de ces composantes, les relations spatiales, les degrés de confiance en toutes ces informations.

Si de nombreux travaux ont porté sur la modélisation des connaissances floues et sur les structures de graphes, il n'existe pas de formalisme établi pour l'apprentissage de données à la fois structurées et floues. C'est là que se situe la contribution essentielle de la thèse.

Les machines à vecteurs support (SVM) sont des méthodes récentes d'apprentissage supervisé qui sont pertinentes pour cette problématique. Le travail de recherche porte sur l'adaptation de cette méthode performante de classification à la structure des entrées, à l'aide des mesures de similarité entre les graphes, appelées des fonctions noyau, qui garantissent, par leurs propriétés spécifiques, la convergence et la tractabilité de l'algorithme de classification.

Le schéma général de la méthode proposée est illustré sur la figure 6.1. Une étape de pré-traitement est nécessaire, afin d'identifier des zones d'intérêt dans l'image et de calculer des attributs numériques associés aux caractéristiques visuelles locales, telles que la couleur ou la texture. Ce pré-traitement peut être réalisé :

- de façon complètement automatique par un algorithme de segmentation (hiérarchique),
- de façon supervisée par un utilisateur qui facilite le choix de paramètres d'un algorithme de segmentation pour un certain type d'images,
- par une segmentation manuelle.

La structure du graphe de modélisation est enrichie et des attributs structurels représentant des relations spatiales telles que l'adjacence sont ajoutés. Ces relations sont modélisés sous forme floue. Si de tels modèles ont été assez largement utilisés pour la reconnaissance de structures dans les images, ils ne l'ont pas encore été en apprentissage, ce qui constitue donc une des originalités de notre approche. Des fonctions noyau

bien adaptées sont définies pour chaque type d'attribut, et ensuite des algorithmes connus en littérature sous le nom de *Multiple Kernel Learning(MKL)* sont employés pour que la machine SVM utilisée fasse un usage simultané optimal de toutes les informations visuelles et spatiales encodées dans la structure du graphe. Dans les paragraphes suivants, j'essaierai de décrire et motiver brièvement les choix essentiels qui ont été faits pour la réalisation de ce schéma d'interprétation d'images.

En ce qui concerne les attributs visuels extraits, on prend en compte des caractéristiques de couleur, forme et texture qui sont utilisées dans la plupart des méthodes d'interprétation d'images et qui sont très souvent aussi déterminantes pour l'étape de pré-traitement, et de détection des zones d'intérêt. Ensuite, l'ajout d'arcs comportant plus qu'une information implicite d'adjacence entre les zones d'intérêt nous permet donc non seulement de mieux décrire les informations structurelles, mais aussi d'améliorer la robustesse de la représentation. Le graphe généré ne sera donc plus un graphe d'adjacence classique.

Dans les travaux effectués, nous proposons donc une autre caractéristique, topologique, qui estime un degré d'adjacence entre deux régions. Parmi les méthodes possibles, nous estimons le degré d'adjacence entre deux régions en mesurant la corrélation entre la portion de l'espace "proche" de la première région dite de référence et la deuxième. Cette mesure est maximale lorsque la région de référence est imbriquée dans la région cible et elle est nulle si les deux régions sont trop éloignées. Une valeur moyenne implique que deux régions se trouvent à proximité au moins dans une sous-partie de leur voisinage immédiat dans l'image. Les représentations floues sont particulièrement appropriées pour modéliser l'imprécision de plusieurs relations spatiales telles que "proche". De cette manière, toute estimation floue d'une relation spatiale peut être intégrée facilement au modèle graphique.

Une difficulté à surmonter dans le traitement d'un modèle graphique avec un étiquetage complexe vient du fait qu'à la base, la fonction noyau pour les graphes prend en compte une paire *(attribut de région; relation spatiale)* à la fois, alors qu'une bonne discrimination est nécessairement obtenue à partir de l'ensemble des informations encodées dans le graphe. L'intérêt de l'appel aux méthodes d'apprentissage multiple est d'obtenir une seule fonction noyau et donc un seul classifieur à partir de caractéristiques hétérogènes. Nous avons intégré plusieurs méthodes récentes d'optimisation convexe pour l'apprentissage multiple, et nous avons entamé un travail pour adapter mieux l'algorithme d'apprentissage multiple au problème particulier de l'apprentissage à partir des images.

Apprentissage statistique et données structurées

Dans le cadre de l'apprentissage artificiel, les méthodes statistiques supervisées ont jusqu'à présent donné lieu à des algorithmes permettant de traiter des données d'entrée sous forme de vecteurs ou de séquences et de prédire des sorties sous la forme de valeurs réelles ou de catégories. Aujourd'hui, de nombreuses données ont une représentation

complexe sous la forme d'arbres, de graphes ou de données relationnelles, particulièrement dans les sciences du vivant mais aussi dans le traitement des images, les modèles de raisonnement, etc. De même, on voit apparaître de nouvelles tâches d'apprentissage qui nécessitent d'extraire, à partir de données, des relations ou des dépendances structurelles. Ces besoins permettent d'envisager l'apparition de méthodes d'apprentissage pour des tâches plus complexes et souvent liées à l'analyse de données de grand volume, issues de l'informatique médicale, de la biologie, du langage naturel, etc.

Dans le cadre de ce travail, notre attention porte particulièrement sur les algorithmes capables de gérer l'apprentissage sur ces modèles qui ont été créés et optimisés dans le contexte des applications bioinformatiques. Une adaptation pour l'apprentissage des graphes issus des images est nécessaire, car les propriétés des informations codées dans la structure graphique représentant une image sont fondamentalement différentes de celles utilisées en bioinformatique.

En effet, il y a une incompatibilité intrinsèque entre l'entrée des méthodes d'apprentissage classiques et les données structurées issues des images. Plutôt que d'essayer de modifier les algorithmes classiques, il est possible de chercher dans la direction opposée et d'adapter les données afin de modifier la complexité structurelle tout en préservant les caractéristiques qui permettent d'affecter les données à différentes classes. C'est le type d'approches développées en bio-informatique pour la classification des molécules par exemple.

Dans une représentation des données complexes par des graphes, les nœuds représentent par exemple des atomes (Tsuda et al, 2002; Kashima et al, 2003) ou des structures atomiques simples (Mahé et al, 2006), des protéines (Borgwardt and Kriegel, 2007), des objets que l'on cherche à segmenter dans des images ou à classifier (Aldea et al, 2007; Harchaoui and Bach, 2007), et les arêtes modélisent des interactions spécifiques, des relations d'ordonnancement et de dépendence, des relations spatiales (adjacence, distance, localisation relative, topologie, ...). Dans ce contexte, et en particulier pour les applications en interprétation d'images et en raisonnement spatial, de nombreuses sources d'imprécision doivent être prises en compte : sur les objets, à cause de leurs limites imprécises et des éventuelles difficultés à les segmenter, sur les relations et sur les connaissances manipulées, souvent exprimées sous forme linguistique. L'étiquetage des graphes nous permet de synthétiser les connaissances sur la nature et les propriétés des composantes du graphe, mais aussi le degré de confiance en toutes ces informations.

Le deuxième chapitre du manuscrit représente une courte partie introductive au concept fondamental de l'apprentissage statistique (2.1), et plus précisément à la théorie des séparateurs à vaste marge (2.2.3). Une partie essentielle du travail effectué se situe autour des fonctions noyau (2.21, 2.3) utilisées dans la théorie des SVM pour projeter les donneés à classifier dans des espaces de Hilbert $\mathbb{H}$, habituellement de grandes dimensions, où une classification linéaire est possible:

$$\min_{w\in\mathbb{H},b\in\mathbb{R},\xi\in\mathbb{R}^n} \langle w, w\rangle + C\sum_{i=1}^{n} \xi_i \tag{1}$$

$$y_i\left(\langle w, \Phi(x_i)\rangle + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \qquad i = 1, \ldots, n.$$

Les noyaux de graphes Dans le cadre le plus général, les données d'entrée sont représentées par des graphes orientés, dont la structure interne est définie par :

- un ensemble de nœuds étiquetés $\mathcal{V} = \{x_1, \ldots, x_{|\mathcal{V}|}\}$
- un ensemble d'arêtes étiquetées $\mathcal{E}$
- une fonction d'étiquetage de nœuds $v : \mathcal{V} \to \mathcal{S}_v$
- une fonction d'étiquetage d'arêtes $e : \mathcal{E} \to \mathcal{S}_e$

Le choix des fonctions d'étiquetage est étroitement lié aux propriétés des entités qui sont modélisées par les graphes. Dans une première étape, on peut considérer que l'ensemble d'étiquetage de nœuds est fini, et qu'il n'y a pas d'étiquetage d'arêtes.

La similarité entre graphes est définie par une méthode à noyaux et permet de construire un classifieur dans un espace de caractérisiques $\mathbb{H}$. Cette similarité entre deux graphes G et G' est évaluée par une fonction noyau de graphes de type noyau marginalisé (2.7) qui s'appuie sur les occurrences des sous-structures identiques des graphes. Les sous-structures considérées sont des marches aléatoires $h = (x_1, \ldots, x_n)$ de longueur $|h| = n$, où $(x_i, x_{i+1}) \in \mathcal{E}$. L'idée de l'approche (Kashima et al, 2003) est de définir une distribution de probabilité sur l'ensemble de marches aléatoires possibles (2.36) et d'estimer la similarité de graphes par la somme des similarités de tous les marches aléatoires possibles dans les deux graphes, pondérée par la probabilité d'apparition de ces marches :

$$K(G, G') = \sum_{h} \sum_{h', h'=h} p(h|G)p(h'|G') \tag{2}$$

Par rapport à d'autres contextes dans lesquels ce type de méthode a été utilisé (Mahé et al, 2004), en interprétation d'images l'espace d'étiquetage devient continu et multi-dimensionnel, et constitue une part prédominante de l'information. Par conséquent, la fonction de similarité pour des attributs à valeurs continues doit être moins discriminante que le noyau de Dirac classiquement utilisé en bio-informatique. En conséquence nous utilisons des noyaux gaussiens de variance σ^2 pour évaluer la similarité $k_a^{rbf}(a_1, a_2)$ entre deux valeurs a_1 et a_2 d'un attribut numérique a :

$$k_a^{rbf}(a_1, a_2) = \exp\left(-\frac{\|a_1 - a_2\|^2}{2\sigma^2}\right) \tag{3}$$

ou bien d'autres fonctions noyau adaptées aux différents attributs, mais dont l'estimation numérique est continue.

Afin de pouvoir intégrer facilement des fonctions noyau d'attributs plus complexes, l'amélioration majeure que l'on propose par rapport au modèle qui prend en compte

la probabilité jointe de h et h' seulement si $h = h'$ est d'intégrer les fonctions noyau d'attributs visuels et relationnels dans un noyau de convolution (2.4), et d'intégrer l'estimation du noyau de convolution au modèle probabiliste sous-jacent aux structures des graphes.

Etant donnés deux graphes G et G' à comparer, l'équation 2.39 permet d'évaluer la similarité $k_{v,e}(h, h')$ entre deux marches aléatoires $h = \{x_1, \ldots, x_n\}$ dans G et $h' = \{x'_1, \ldots, x'_n\}$ dans G', en combinant les similarités k_v et k_e des valeurs d'un attribut de nœuds v et respectivement les valeurs d'un attribut d'arêtes e le long de h et h':

$$k_{v,e}(h, h') = k_v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} k_e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) k_v(v_{x_i}, v_{x'_i}) \tag{4}$$

Enfin, le noyau entre G et G' est calculé en sommant les similarités de tous les chemins aléatoires h et h' possibles, pondérées par leurs probabilités d'apparition :

$$K_{v,e}(G, G') = \sum_{h} \sum_{h'} k_{v,e}(h, h') p(h|G) p(h'|G') \tag{5}$$

Du point de vue numérique, la solution que l'on propose afin d'éviter une augmentation en complexité due à la prise en compte de la fonction $k_{v,e}(h, h')$ est de pénaliser les probabilités jointes d'apparition des marches $h'h'$ par l'estimation de similarité de marches :

$$\begin{cases} \pi_s(x, x') = p_s^G(x) p_s^{G'}(x') \times K_v(v_x, v_{x'_i}) \\ \\ \pi_t((x_1, x'_1)|(x_2, x'_2)) = p_t^G(x_2|x_1) p_t^{G'}(x'_2|x'_1) \times K_e(e_{x_1x_2}, e_{x'_1x'_2}) \times K_v(v_{x_2}, v_{x'_2}) \end{cases} \tag{6}$$

où Π_s, Π_t décrivent une structure auxiliaire, appelée graphe produit (2.35), qui permet de calculer $K_{v,e}(G, G')$ de manière analytique. Cette fonction est ensuite utilisée dans un séparateur à vaste marge (SVM) pour construire un classifieur d'images. $K_{v,e}$ définit une matrice de similarité entre les paires de graphes à comparer.

Du point de vue algorithmique, cette approche pour l'intégration d'un étiquetage complexe dans le calcul de noyaux de graphes présente l'avantage majeur de ne pas augmenter l'ordre de complexité de la méthode de base (Tsuda et al, 2002) car la prise en compte de l'étiquetage se fait en amont de l'estimation analytique du noyau de graphes(2.46). En conséquence, la complexité de la méthode reste en $\mathcal{O}((|\mathcal{V}||\mathcal{V}'|)^3)$. Cette complexité est liée à l'inversion de la matrice Π_t qui décrit les probabilités de transition simultanées pour deux paires de nœuds adjacents dans les graphes G, G'. Les propriétés structurelles des modèles graphiques d'images sont différentes de celles rencontrées en bioinformatique (surtout en ce qui concerne le degré de connexité). Cette différence et le fait d'utiliser un étiquetage continu et des fonctions noyau d'attributs dont le support n'est pas compact font que la matrice Π_t présente moins de valeurs nulles. Néanmoins, nous avons montré (4.6.3) qu'on peut utiliser des schémas d'approximation numérique pour estimer le noyau de graphe avec une précision donnée, avec un temps de calcul de quelques ordres de grandeur inférieur aux temps demandés par l'approche analytique.

Modèles structurés et représentation d'images La suite de notre travail porte sur le rapport entre les modèles de représentation d'images, le fossé sémantique et les méthodes possibles pour relier les concepts aux descripteurs visuels.

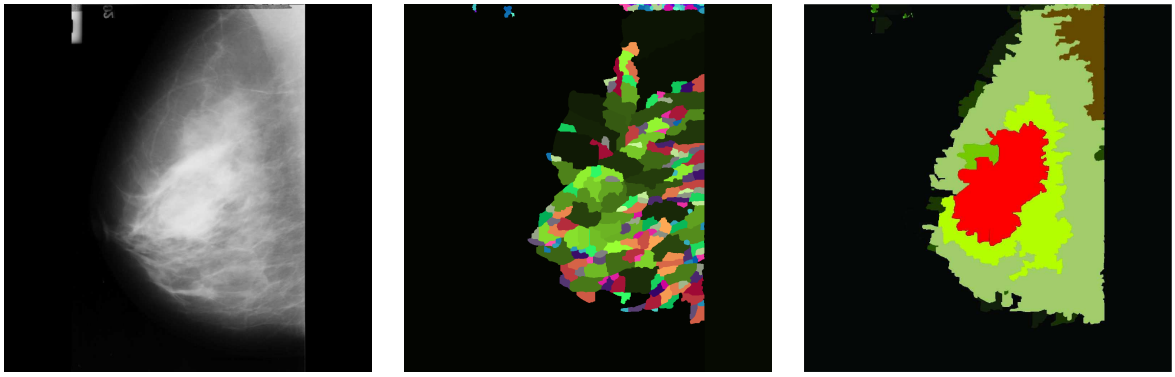
Dans le cas particulier des images, différentes approches ont été proposées pour la représentation. Historiquement, la première approche traite les images comme des objets indivisibles et exploite des caractéristiques globales telles que l'histogramme des couleurs (Chapelle et al, 1999). Aujourd'hui, les représentations globales sont remplacées par des ensembles d'attributs et descripteurs bien localisés, tels que les points de saillance, les caractéristiques associées aux régions, les caractéristiques spatiales, les descripteurs locaux de forme, etc. Cela est une conséquence directe du fait que les caractéristiques globales ne sont pas suffisamment bien adaptées pour dépasser le fossé sémantique, alors que les caractéristiques locales permettent plus facilement d'associer les concepts sémantiques aux parties des images.

En conséquence, une deuxième stratégie considère les images comme des "sacs" d'objets indépendants et s'appuie sur une vectorisation du contenu de l'image (Sivic and Zisserman, 2003; Dance et al, 2004; Sivic et al, 2005). Enfin, une dernière catégorie de méthodes s'appuient sur une représentation de l'image comme un ensemble structuré d'objets (Fei-Fei et al, 2003; Carbonetto et al, 2004; Bar Hillel et al, 2005; Carbonetto et al, 2008; Bar Hillel and Weinshall, 2008), faisant apparaître explicitement les constituants de l'image et leurs relations.

Ce type de représentation fondée sur des sous-parties est caractérisé par un compromis entre l'exploitation de caractéristiques visuelles très localisées et abondantes dans un "sac de mots visuels" sans tenir compte des interactions spatiales, et le modèle de constellation (Weber et al, 2000; Fergus et al, 2003; Lazebnik et al, 2004). Les modèles fondés exclusivement sur les sacs de mots doivent traiter le problème d'inclusion pendant la partie d'apprentissage de caractéristiques qui n'appartiennent pas aux objets d'intérêt, en l'absence d'informations spatiales. Les modèles de constellations, qui sont plus proches de la facon humaine de construire des représentations avec des contraintes géométriques mutuelles entre les parties des objets, sont en général complexes du point de vue computationnel. L'approche proposée dans le cadre de notre travail entre dans cette catégorie qui unit les représentations granulaires, sémantiquement pauvres, et les représentations de haut niveau du point de vue sémantique.

L'extraction du graphe Notre méthode d'interprétation d'images s'appuie sur des représentations sous la forme de modèles graphiques. Nous avons choisi d'utiliser comme sous-parties de l'image à analyser des régions issues d'une segmentation manuelle ou bien automatique, ce qui fait qu'on obtient dans une première étape des graphes d'adjacence de régions, en considérant comme relation spatiale l'adjacence stricte entre régions.

Les graphes d'adjacence de régions sont utilisés depuis longtemps en traitement d'images et surtout depuis le développement des algorithmes de segmentation, mais principalement pour la reconnaissance de formes par mise en correspondance. Les difficultés majeures liées à l'utilisation de graphes d'adjacence de régions pour l'apprentissage



(a) Image initiale (b) Segmentation directe (c) Régions fusionnées

Figure 1: Segmentation d'une mammographie

automatique sont dues principalement à la complexité computationnelle et à la variabilité structurelle des graphes en fonction du pré-traitement. La solution proposée pour la première source de difficulté est l'utilisation de la fonction noyau de graphes présentée antérieurement. Dans la section 4.5 on présente plus en détail les choix possibles pour ce qui peut constituer une région dans l'approche par régions et les descripteurs visuels qui sont couramment utilisés pour extraire l'information associée à une région. En 4.6.1 on présente une solution simple pour l'extraction automatique de régions de type fusion hiérarchique qui a néanmoins l'avantage de ne pas ajouter une complexité substantielle par rapport à l'étape de sur-segmentation par ligne de partage des eaux, et qui s'adapte automatiquement à la dynamique de l'image (figure 1).

Apprentissage sur les graphes d'adjacence de régions Une première validation pour cette approche catégorisation par apprentissage discriminatif sur les graphes d'adjacence de régions a été effectuée sur une base d'images constituée de mammographies (vue sagittale) et de radiographies de la tête, dont le protocole d'acquisition intra-classe n'est pas identique. La méthode est évidemment applicable aux images quelconques, avec la mention que la difficulté variable de l'étape de pré-traitement par segmentation et donc d'extraction de graphes peut imposer l'adaptation de l'algorithme de segmentation non supervisée présenté auparavant. Dans le graphe, les régions correspondent donc aux nœuds, et les arêtes encodent la relation d'adjacence stricte. En plus de l'information spatiale encodée par la structure du graphe, nous intégrons dans l'étiquetage des nœuds l'information intrinsèque correspondant aux propriétés visuelles des régions.

La similarité entre les graphes est estimée donc à l'aide d'une fonction noyau de graphes employée dans un classifieur SVM. Dans la section 3.2, nous avons montré comment le calcul de similarité de graphes se réduit à la définition de fonctions noyau pour les étiquettes qui annotent les nœuds et éventuellement les arêtes. Pour des valeurs scalaires, les noyaux gaussiens représentent un choix naturel parmi d'autres fonctions noyau (2.3,4.6.2), en général stationnaires. En ce qui concerne les histogrammes, des dis-

tances adaptées existent dans la littérature (Levina and Bickel, 2001; Arivazhagan et al, 2006) et la "kernelisation" (4.12) nous permet d'obtenir des fonctions noyau véritables à partir de ces distances.

Pour l'instant, les arêtes ne sont pas prises en compte au-delà de leur importance structurelle implicite ; on considère donc que l'étiquetage spatial est uniforme et on exploite surtout l'étiquetage visuel des nœuds. Pour cette première application, nous avons particulièrement analysé deux descripteurs qui s'adaptent bien à la dynamique de l'image: la surface relative et le niveau de gris relatif. Dans une première étape, nous avons comparé les performances des noyaux gaussiens et triangulaires. Les résultats (table 4.1) valident notre approche et montrent que le noyau gaussien reste plus robuste que le noyau triangulaire, malgré un coût computationnel légèrement plus élevé.

Catégorisation et plusieurs descripteurs La formulation typique du noyau marginalisé (section 3.2) impose l'utilisation d'une paire (K_v, K_e) dans la construction de la fonction. Le résultat immédiat est qu'on s'appuie sur un descripteur de nœuds et un descripteur spatial, et la similarité de graphes repose en conséquence sur ces deux attributs. Dans le cas général d'un ensemble de descripteurs $F = \{f_1, \ldots, f_n\}$ associé à un constituant du graphe, la fonction noyau peut être modifiée afin de prendre en compte plusieurs ou tous les éléments de F. La façon la plus directe d'accomplir cette tâche est de construire une combinaison linéaire de noyaux de base:

$$K_F = \sum_{i=1}^{|F|} \lambda_i K_{f_i} \tag{7}$$

où les facteurs s $\lambda_i \geq 0$ ont la propriété $\sum_{i=1}^{n} \lambda_i = 1$. Dans ce cas, la somme pondérée de fonctions définies positives garde cette propriété essentielle pour l'optimisation réalisée par l'algorithme SVM.

Dans une deuxième étape, nous avons construit une combinaison linéaire fondée sur les deux attributs s_{rel} et $gray_{rel}$, afin d'estimer la performance de catégorisation par rapport aux facteurs des deux fonctions noyau de base. Dans les expériences, l'importance d'un des attributs est augmentée progressivement de 0 à 1, au détriment de l'autre. Le résultat illustré en figure 4.10 montre que les performances sont améliorées pour la fonction noyau principale, même avec seulement deux attributs visuels et un étiquetage spatial minimal, assuré par l'information d'adjacence stricte.

Conclusions Dans cette partie du travail, on introduit les principaux problèmes rencontrés en interprétation d'images à partir du contenu, et les approches fondamentales utilisées afin de dépasser ces difficultés. Nous avons motivé le choix de l'utilisation d'un modèle graphique d'adjacence de régions comme un bon point de départ pour le compromis entre les modèles "sac-de-mots" et les modèles de constellations. Nous avons présenté une version du noyau marginalisé de graphes qui est une extension naturelle du celui utilisé en bioinformatique et qui permet l'interprétation de graphes comportant

un étiquetage plus riche. Nous avons employé cette fonction noyau pour une tâche d'apprentissage d'images, dont le pré-traitement a été effectué par un algorithme de segmentation et fusion hiérarchique générique et non supervisé. Les résultats obtenus valident cette approche d'apprentissage et soulignent également l'intérêt de la sélection et combinaison automatique d'attributs pertinents, ainsi que l'intérêt pour l'intégration d'une information spatiale plus riche.

Relations spatiales dans le contexte d'apprentissage de graphes

Le contexte spatial a été intégré avec succès en bioinformatique, pour la représentation par graphes de données complexes. Avec peu d'exceptions (Mahé et al, 2006), les relations spatiales utilisées sont de type binaire et modélisent de façon très exacte la présence ou l'absence d'une interaction entre les constituants de la structure globale. Dans ce cadre particulier, la construction du graphe complémentaire $\tilde{G}$ de la représentation (Borgwardt and Kriegel, 2007), qui code les absences d'interactions, et l'utilisation de cette information supplémentaire pour améliorer la fonction noyau du classifieur sont non-ambiguës et facilement réalisables (section 5.1.2):

$$K^*(G, G') = K(G, G') + K(\tilde{G}, \tilde{G}') \tag{8}$$

En ce qui concerne l'extraction de l'information spatiale à partir des images, la situation devient plus complexe. Les interactions spatiales ont une variabilité liée à l'interprétation sémantique humaine qui se situe bien au-delà du cas binaire mentionné ci-dessus. De plus, l'intégration de l'information spatiale floue et des attributs de régions s'avère aussi plus complexe que dans les situations traitées par l'équation (8). Toutefois, ces relations spatiales apportent une grande richesse dans la description des images. Cela a été largement montré dans des problèmes de segmentation et reconnaissance de structures dans les images. Ici nous montrons comment utiliser ces informations en apprentissage et classification.

Tout d'abord, les interactions spatiales présentent une variabilité sémantique qui va au-delà du modèle binaire utilisé auparavant. Cette situation est vraie pour les interactions spatiales complexes comme "entouré par", "le long de" etc., de même que pour les relations spatiales les plus simples comme "à côté de" ou "proche de". En plus, une représentation binaire n'est pas adaptée pour le type d'entrée typique fourni par les étapes de pré-traitement comme la segmentation en régions. En effet, lorsque l'adjacence ne dépend que de quelques pixels, le graphe obtenu pourra être différent en fonction de la méthode de segmentation utilisée. L'ajout d'arêtes comportant plus qu'une information implicite d'adjacence nous permet donc non seulement de mieux décrire les informations structurelles, mais aussi d'améliorer la robustesse de la représentation.

En ce qui concerne la représentation d'images et les relations spatiales floues (sections 5.2, 5.3), nous avons choisi de prendre en compte les relations topologiques entre deux objets, qui sont fondées sur les notions d'intersection, intérieur, extérieur (Dubois and Prade, 1980; Bloch et al, 1997; Hudelot et al, 2008).

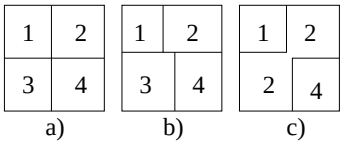


Figure 2: Défaut de robustesse de l'adjacence : différents résultats de segmentation probables pour quatre régions. (a) Toutes les régions sont adjacentes deux à deux. (b) Les régions 1 et 4 ne sont plus adjacentes. (c) Les régions 2 et 3 ont fusionné, les régions 1 et 4 ne sont plus adjacentes.

Dans la suite de notre travail, on propose l'utilisation d'autres relations spatiales donnant une information structurelle. Comme application, on prendra en compte la notion de distance, ainsi que celle de position directionnelle relative. Plus précisément, pour notre application, nous utiliserons une relation spatiale topologique représentée par un degré d'adjacence étendu. Notons que le cadre méthodologique proposé permet l'intégration d'autres relations spatiales également.

Distance entre régions. La distance entre deux régions R_1 et R_2 est calculée comme le minimum des distances euclidiennes entre deux points $p_i \in R_1$ et $q_j \in R_2$:

$$d(R_1, R_2) = \min_{p_i \in R_1, q_j \in R_2} (d_{Euclidean}(p_i, q_j)) \tag{9}$$

La distance, ainsi que l'orientation ne donnent pas toujours une information pertinente et robuste; par exemple, la distance entre deux régions est identique pour une adjacence stricte d'un seul pixel et pour une région qui est complètement entourée par la deuxième. En conséquence, nous proposons la prise en compte d'une caractéristique topologique qui estime la longueur de l'adjacence entre deux régions.

Mesures d'adjacence fondées sur une comparaison floue. Nous estimons le degré d'adjacence entre deux régions en mesurant la corrélation entre la portion de l'espace "proche" (Bloch, 2005) de la première région dite de référence et la deuxième. Cette mesure est maximale lorsque la région de référence est imbriquée dans la région cible. Elle est nulle si les deux régions sont trop éloignées. Une valeur moyenne implique que deux régions sont adjacentes pour au moins la moitié du contour de la région de référence.

Les représentations floues sont appropriées pour modéliser l'imprécision intrinsèque de plusieurs relations telles que "proche". Ici, nous considérons la formulation proposée dans Bloch (2005) : étant donné un objet de référence, la portion de l'espace dans laquelle la relation à cet objet est satisfaite est définie comme un ensemble flou spatial, dont la fonction d'appartenance en chaque point donne le degré de satisfaction de la relation en ce point. Cette représentation dans l'espace de l'image est propice à sa combinaison avec d'autres informations de type image.

La relation spatiale “proche” peut être déduite d’un intervalle flou de forme trapézoïdale f sur $\mathbb{R}^+$. Un sous-ensemble flou $\mu_{proche(R_1)}$ de l’espace de l’image $\mathcal{S}$ est alors construit en combinant f avec une carte de distance d_{R_1} à l’objet de référence R_1 : $\forall x \in \mathcal{S},\ \mu_{proche(R_1)}(x) = f(d_{R_1}(x))$, où $d_{R_1}(x) = \inf_{y \in R_1} d(x, y)$. Dans la figure 3 on présente un ensemble flou (a), une image d’entrée possible (b) et une segmentation (c) et la carte de distance pour une region (d).

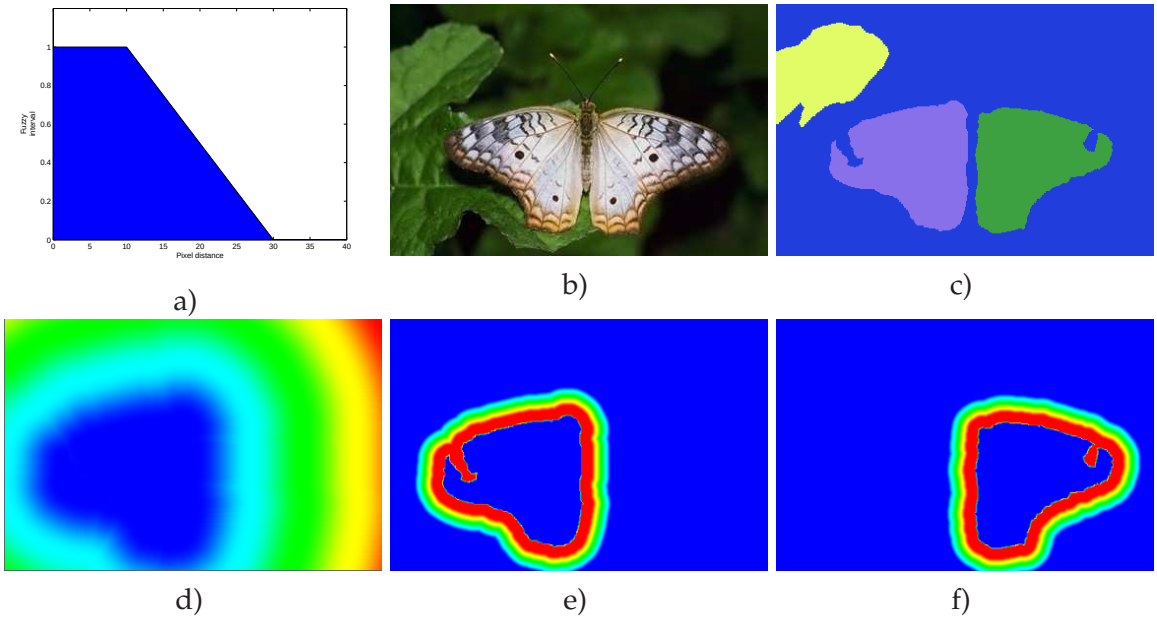


Figure 3: (a) Un sous-ensemble flou pour la relation de distance. (b) Image d’entrée. (c) Un résultat de segmentation, avec quatre régions distinctes (d) Carte de distance par rapport à la région correspondant à l’aile gauche. (e) Estimation par un sous-ensemble flou de la relation “proche de l’aile gauche" (la couleur rouge correspond aux valeurs élevées). (f) Estimation par un sous-ensemble flou de la relation “proche de l’aile droite".

Parmi les mesures d’estimation possibles, on considère comme critère une *M-mesure de satisfiabilité* (Bouchon-Meunier et al, 1996) définie comme:

$$Sat^k(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)} \tag{10}$$

où $\mathcal{S}$ représente le domaine spatial.

Cette mesure floue estime la précision de la position de l’objet dans la région où la relation est satisfaite. La mesure est maximale si le deuxième objet couvre entièrement le noyau de $\mu_{near(R_1)}$. On rappelle que la taille de l’ensemble flou initial est adaptative et peut être modifiée par rapport aux caractéristiques de l’image. Si l’objet R_2 est binaire, cette mesure se réduit à $\frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)}$, et donc représente la partie de $\mu_{near(R_1)}$ qui est couverte par l’objet R_2.

Il y a des cas où une estimation légèrement différente peut également convenir. Si l'objet R_2 est relativement petit, la mesure floue est faible même si l'objet R_2 est entièrement compris dans le noyau de $\mu_{near(R_1)}$. En conséquence, on propose de modifier l'estimation de la manière suivante:

$$Sat^o(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \mu_{R_2}(x)} \tag{11}$$

Toujours si l'objet R_2 est binaire, la mesure se réduit à $\frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in \mathcal{S}} \mu_{R_2}(x)}$, qui représente cette fois la partie de l'objet R_2 contenue dans $\mu_{near(R_1)}$.

Mesure d'adjacence fondée sur la ressemblance floue En plus de la mesure de satisfiabilité, on choisit de prendre en compte également une mesure symétrique, *M-mesure de ressemblance* (Bouchon-Meunier et al, 1996) définie comme :

$$Res(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \max(\mu_{near(R_1)}(x), \mu_{R_2}(x))}$$

Cette mesure représente le rapport entre la cardinalité de l'intersection de $\mu_{near(R_1)}$ et μ_{R_2}, et la cardinalité de leur union. Elle est maximale si l'objet et la relation sont identiques: elle considère le positionnement de l'objet ainsi que la précision de l'ensemble flou.

Dans la figure 3(e) et la figure 3(f), nous avons illustré les sous-ensembles flous correspondant aux deux ailes. Avec la première mesure de satisfiabilité présentée, on obtient une estimation de 0,1 pour "aile droite proche de l'aile gauche" et de 0,109 pour "aile gauche proche de l'aile droite". L'exemple montre aussi l'intérêt de ces estimations d'adjacence floue pour les régions qui sont déconnectées par rapport à la relation d'adjacence stricte.

Position relative directionnelle. Nous avons choisi ici de représenter cette information en utilisant des histogrammes d'angles (Miyajima and Ralescu, 1994). Cela nous permet de représenter toutes les directions possibles entre deux régions. Si R_1 et R_2 sont deux ensembles de points $R_1 = \{p_1, ..., p_n\}$ et $R_2 = \{q_1, ..., q_n\}$, la position relative directionnelle entre les régions R_1 et R_2 est estimée à partir de la position de chaque point q_j de R_2 par rapport à chaque point p_i de R_1. L'histogramme d'angles $H_{R_1R_2}$ est défini en 2D par :

$$H_{R_!R_2}(\theta) = \left| \{(p_i, q_j) \in R_1 \times R_2 / \angle(\overrightarrow{i}, \overrightarrow{p_iq_j}) = \theta\} \right|$$

où $\angle(\overrightarrow{i}, \overrightarrow{p_iq_j})$ représente l'angle entre un vecteur de référence $\overrightarrow{i}$ et le vecteur $\overrightarrow{p_iq_j}$. L'extension à trois dimensions est une fonction de deux angles. Un exemple est illustré figure 4.

Afin d'obtenir une valeur réelle, nous calculons le centre de gravité de l'histogramme. Le calcul de l'histogramme d'angles étant coûteux en termes de temps de calculs, nous

a) Région 1 b) Région 2 c) Histogramme normalisé

Figure 4: Histogramme d'angles entre les régions 1 et 2. Le centre de gravité est représenté par une ligne verticale.

pouvons être amenés à travailler avec des images réduites, ce qui affecte peu le calcul du centre de gravité.

Dans le reste du travail, on considère que $\mathcal{R}$ représente une de ces mesures d'adjacence floues, mais on souligne que $\mathcal{R}$ peut être substituée par toute autre estimation de l'interaction spatiale entre les sous-parties d'une image, comme par exemple "parallèle à" ou "le long de" (Vanegas et al, 2009; Takemura et al, 2005), si ces relations sont bien adaptées pour le contenu de l'image concernée.

Information spatiale floue et modèles discriminatifs

L'apprentissage discriminatif, et également celui concernant les graphes, s'appuie fortement sur les estimations de similarité entre les objets d'entrée, de telle manière que la similarité entre deux graphes augmente si les graphes présentent de nombreuses structures similaires. Dans l'étiquetage spatial flou, la similarité fréquente entre des étiquettes qui correspondent à des estimations faibles masque la contribution des estimations fortes qui encodent l'information spatiale essentielle. Comme résultat, la fonction noyau de graphe sur-estime la mesure de similarité globale et diminue sa capacité de discrimination (section 5.5).

Une solution à ce problème est d'enlever des arêtes en seuillant les valeurs des relations spatiales, pour qu'une arête apparaisse seulement si l'estimation de la relation est supérieure à un certain seuil θ. Cette opération fait apparaître une structure graphique qui dépend de θ et qu'on appelle par la suite θ-graphe:

Définition (θ-graphe)**.** *Considérons une partition en régions d'une image I et une relation spatiale $\mathcal{R}$. Par $\mathcal{V}(I)$ on note l'ensemble des nœuds correspondant aux régions de I. Un θ-graphe G_θ est défini par:*

$$G_\theta = \{\mathcal{V}(I); (v_1, v_2) \in \mathcal{V}^2(I) | \mathcal{R}(v_1, v_2) \geq \theta\} \tag{12}$$

où $\mathcal{R}(v_1, v_2)$ est la relation spatiale générique entre les régions (nœuds) v_1 et v_2.

De cette manière, on peut associer à chaque paire $(I, \mathcal{R})$ un ensemble infini de représentations graphiques $\mathcal{G} = \{G_\theta\}_{\theta \geq 0}$. En réalité, la taille de cet ensemble est finie et bornée par $\frac{\|\mathcal{V}(I)\| \cdot (\|\mathcal{V}(I)\| - 1)}{2}$.

Projection de G_a dans $\mathcal{G}$ Puisque G_a n'appartient pas nécessairement à l'ensemble $\mathcal{G}$ de θ-graphes, et étant donné l'importance perceptuelle du graphe de segmentation, une question pertinente serait : "Quelle est la projection la plus pertinente et robuste $\hat{G}$ du graphe de segmentation G_a dans la famille $\mathcal{G}$?". Une distance structurelle qui est adaptée pour estimer la disparité entre G_a et les graphes de $\mathcal{G}$ est la distance d'édition (Riesen et al, 2007) :

Définition (Distance d'édition de graphes)**.** *Considérons $G = (\mathcal{V}, \mathcal{E})$ la source et $G' = (\mathcal{V}', \mathcal{E}')$ le graphe destination. La distance d'édition entre G et G' est définie par:*

$$\delta(G, G') = \min_{(v_1, \ldots, v_k) \in \Gamma(G, G')} \sum_{i=1}^{k} \psi(v_i)$$

où l'ensemble de chemins d'édition $\Gamma(G, G')$ contient toutes les séquences finies possibles d'opérations élémentaires qui transforment G en G'.

Dans la suite de notre travail, nous avons proposé un ensemble d'opérations élémentaires et de coûts associés qui nous permettent de calculer avec un effort computationnel réduit (section 5.5.2) la projection de G_a dans $\mathcal{G}$ pour une image et une relation spatiale données:

$$\begin{aligned} \delta(G_a, G_\theta) &= \text{card}\{(v_1; v_2) \in \mathcal{V}^2(G_a) | (v_1, v_2) \in \mathcal{E}(G_a) \wedge \mathcal{R}(v_1, v_2) < \theta\} \\ &+ \text{card}\{(v_1; v_2) \in \mathcal{V}^2(G_a) | (v_1, v_2) \notin \mathcal{E}(G_a) \wedge \mathcal{R}(v_1, v_2) \geq \theta\} \end{aligned} \tag{13}$$

Expériences

La base d'images IBSR[1] contient des données cliniques réelles et est couramment utilisée en tant que base d'images par résonance magnétique (IRM) de cerveaux sains en trois dimensions. Elle contient les segmentations de 18 cerveaux, réalisées manuellement par un expert, et chacune est disponible selon trois plans de coupe (ou vues) différents : axial, coronal et sagittal. Chaque élément de la base de données est un ensemble de coupes qui couvre le cerveau dans son ensemble.

Cette base de données est principalement utilisée comme un moyen d'évaluer les performances d'algorithmes de segmentation, mais la qualité des segmentations proposées et le fait qu'elle soit disponible gratuitement la rend utile pour nos expériences. La plupart des attributs proposés réussissent à classifier les images de deux vues différentes avec 100% de réussite. En conséquence, nous devons construire une base plus

[1]Internet Brain Segmentation Repository, disponible sur http://www.cma.mgh.harvard.edu/ibsr/

difficile à classifier, et nous avons donc tenté de classifier des images provenant de la même vue. Chaque élément de la base d'une vue va fournir trois coupes successives pour la première classe, puis trois autres coupes successives pour la seconde. Le tableau 1 recense les indices des coupes utilisées dans chacune des trois vues pour définir les classes. Concernant la construction et l'étiquetage du graphe, les nœuds représentent les régions segmentées manuellement et les arêtes modélisent les relations spatiales entre les régions. Pour l'étiquetage des nœuds, nous utilisons des attributs visuels normalisés (niveau de gris, surface, compacité de la région),alors que les relations spatiales fondées sur les mesures d'adjacence floues introduites auparavant fournissent l'étiquetage des arêtes, respectivement. Nous effectuons des validations croisées sur les données ainsi construites($n = 10$), et chaque expérience d'apprentissage est répétée m fois ($m = 10$); les taux de performance présentés représentent chaque fois la moyenne de ces m exécutions.

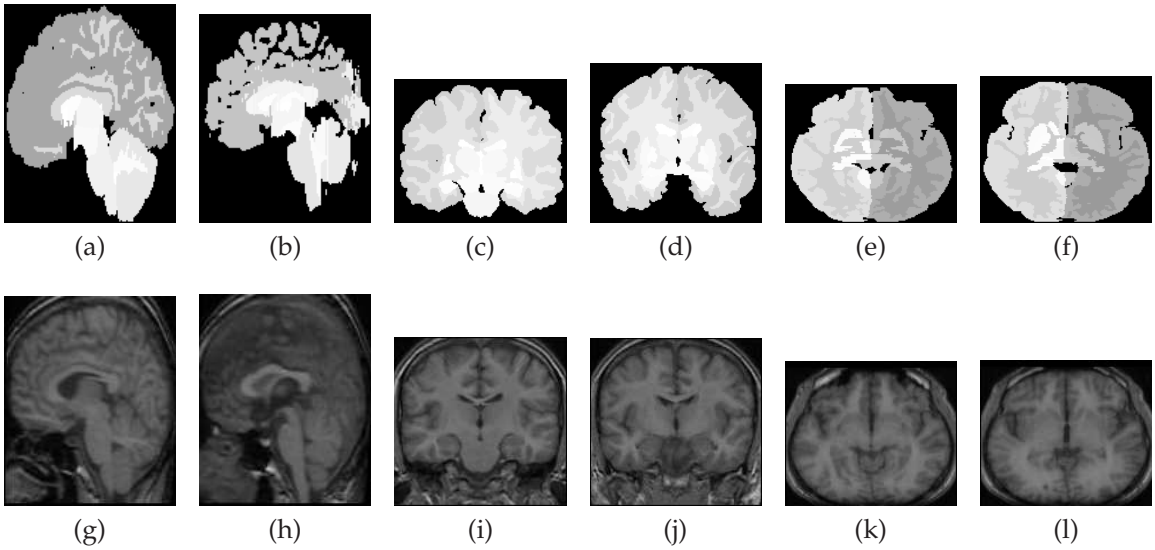


Figure 5: Exemples d'images provenant de la base IBSR. Les niveaux de gris représentent des étiquettes. À gauche, en haut et en bas : deux coupes provenant d'une vue sagittale du même volume IRM 3D et appartenant à chacune des classes. Au centre : vue coronale. à droite : vue axiale.

Vue	Nb coupes	Classe 1	Classe 2
A	255	121, 122, 123	126, 127, 128
C	128	58, 59, 60	64, 65, 66
S	255	121, 122, 123	126, 127, 128

Table 1: Recensement des coupes composant les bases de données des volumes 3D pour chaque vue : axiale (A), sagittale (S) et coronale (C).

Catégorisation et structures d'adjacence stricte G_a Pour un attribut de régions et une relation spatiale donnés, on construit le graphe d'adjacence stricte G_a avec l'inclusion de l'information spatiale numérique dans l'étiquetage d'arêtes, et nous utilisons des noyaux gaussiens pour les attributs dans le cadre d'une approche de type grid-search (figures 5.6, 5.7 et 5.8).

La nouveauté de cette approche est que le graphe d'adjacence stricte inclut dans son étiquetage les estimations floues de l'adjacence de régions. Toutefois, certaines régions qui sont adjacentes strictement présentent une valeur faible de l'adjacence floue, alors que des régions déconnectées peuvent très bien présenter une valeur élevée de $\mathcal{R}$.

Par la suite, nous analysons l'impact de l'ajout d'information structurelle qui n'est pas liée à l'adjacence stricte de régions. La difficulté majeure de cette opération est liée au choix de l'élément θ qui nous permet de sélectionner à partir de l'ensemble $\mathcal{G}$ une représentation pour une image. Notre proposition est d'utiliser la distribution des estimations observées sur la base d'apprentissage (table 5.2) et d'estimer pour l'ensemble de la base le seuil qui minimise la distance structurelle d'édition entre les graphes d'adjacence stricte et leurs θ-graphes correspondants.

Pour la base utilisée, le seuil optimal de la mesure Sat^o est $\hat{\theta} = 0,934$. Cette valeur élevée montre que dans la plupart des cas les régions strictement adjacentes apportent des valeurs qui dépassent ce seuil, ce qu'on peut également percevoir à partir de la proportion élevée de valeurs maximales (figure 5.12. Les deux autres mesures ont un profil différent, en pénalisant très rapidement l'absence d'une forte adjacence (figure 5.13 and 5.14). Pour ces deux cas, les estimations de R associées à l'adjacence stricte sont dispersées dans un intervalle de valeurs plus faibles, et donc les seuils optimaux sont translatés également vers l'intervalle de valeurs faibles: $\hat{\theta} = 0,101$ pour Sat^k, et $\hat{\theta} = 0,091$ pour Res.

L'ensemble des résultats des expériences effectuées pour les trois mesures d'adjacence floue, et pour les différents modèles considérés (graphes d'adjacence stricte, graphes d'adjacence floue) se retrouve dans les tableaux 5.3, 5.4 et 5.5. Tout d'abord, l'intérêt d'ajouter un étiquetage flou même sur la structure du graphe d'adjacence stricte est clairement montré dans tous les situations par le taux de catégorisation de la quatrième colonne par rapport à celui de la troisième. En suite, un résultat qui ressort de la comparaisons entre les trois mesures d'adjacence est que les deux dernières sont plus adaptées pour un apprentissage discriminatif, puisque les estimations élevées sont bien étalées, et la discrimination entre ces valeurs intéressantes peut se faire plus finement. Finalement, cette dernière partie des résultats montre que l'on peut dépasser la représentation structurelle fournie par le graphe d'adjacence stricte et passer aux représentations structurelles floues, tout en gardant un taux de classification élevé. En pratique, l'avantage majeur de cette dernière approche où on passe à une structure graphique fondée uniquement sur les mesures d'adjacence floue est qu'on peut utiliser un espace de recherche des différents paramètres des fonctions noyau plus réduit (un grid-search plus rapide) pour atteindre le même ordre de performance.

Conclusion Dans cette partie de notre travail nous avons estimé les avantages fournis par les représentations d'images par graphes, ainsi qu'en prenant en compte des descripteurs flous pour l'information spatiale. Les modèles graphiques permettent une intégration souple des descripteurs visuels associés aux différentes parties des images et des interactions entre ces parties. Nous avons montré que l'information spatiale est pertinente pour l'apprentissage et que l'intégration d'informations spatiales pose le problème de masquer l'information spatiale pertinente. Une solution qu'on propose afin de filtrer l'information non essentielle est d'enlever une partie des arêtes en s'appuyant sur des seuils adaptés aux propriétés des relations considérées. Les résultats montrent que l'information spatiale floue est bénéfique pour l'apprentissage d'images dans un contexte SVM et que la stratégie de seuillage proposée peut filtrer l'information spatiale non discriminative.

Fusion d'attributs

Nous présentons maintenant comment différents attributs sont combinés dans le noyau de graphes. La méthode de fusion décrite est générale, et sera appliquée dans la suite aux trois descripteurs de structure introduits dans la partie précédente ainsi qu'à des attributs de régions.

L'intérêt de la fusion est de fournir un seul noyau pour la classification à partir de caractéristiques hétérogènes, telles que les attributs considérés. À partir d'un ensemble d'apprentissage et des graphes associés, la première étape est de construire les matrices des noyaux de base $\{K_{a_1}, \ldots, K_{a_n}\}$ correspondant à chaque attribut a_i. Chacune d'entre elles, calculée selon l'équation 5, ne représente qu'une vue partielle des données. La classification d'images fondée sur une seule vue n'est en général pas performante car une bonne discrimination ne peut pas être obtenue à partir d'un seul attribut. La fusion des informations apportées par chaque noyau est alors nécessaire. La manière la plus simple de combiner les noyaux est d'en faire une combinaison linéaire (équation 7) qui permet une compensation entre différents points de vue sur les données, et améliore ainsi la souplesse de la classification.

Une méthode d'obtention du vecteur de poids optimal λ a été proposée dans Lanckriet et al (2004a). Elle considère le cône convexe P des matrices symétriques et définies positives:

$$P = \{X \in \mathbb{R}^{p \times p} \mid X = X^T, X \succeq 0\}.$$

Sur ce cône, le problème dual du SVM est optimisé globalement :

$$\min_{\lambda \in \mathbb{R}^n, K \in P} \max_{\alpha \in \mathbb{R}^m} 2\alpha^T e - \alpha^T D(y)\, K\, D(y)\alpha \tag{14}$$

sous la contrainte :

$$C \geq \alpha \geq 0,\ trace(K) = c,\ K = \sum_{i=1}^{n} \lambda_i K_{a_i},\ \alpha^T y = 0,$$

où m désigne la taille de la base d'apprentissage, $e \in \mathbb{R}^m$ le vecteur dont les éléments sont tous égaux à 1, et $D(y) \in \mathbb{R}^m \times \mathbb{R}^m$ la matrice dont les éléments sont nuls sauf sur la diagonale où ils prennent les étiquettes (+1 ou -1) des exemples d'apprentissage ($D(y)_{ii} = y_i$).

Dans ce problème d'optimisation, C représente le paramètre définissant la marge du SVM et $c \geq 0$ définit la trace de la matrice résultante. L'intérêt de cette méthode est qu'elle minimise la fonction de coût du classifieur en termes à la fois de frontières entre classes et des paramètres λ_i. Le résultat est un ensemble de poids et une fonction discriminante séparant les classes, combinant l'information de plusieurs noyaux.

L'inconvénient de cette formulation est que le passage a l'échelle est difficile à cause du fait que dans le système primal correspondant la régularisation est faite pour une norme mixte (l_2-l_1), à la place de la norme classique l_2. Plus récemment, une méthode qui resout le problème d'apprentissage multiple dans un système d'optimisation faisant intervenir une régularisation par norme l_2 a été proposée (Rakotomamonjy et al, 2007):

$$\min_{\lambda_k, f_k, \xi_i, b} \sum_k \frac{1}{\lambda_k} \|f_k\|^2 + C \sum_i \xi_i, \tag{15}$$

$$y_i \left(\sum_k f_k(x_i) + b \right) \geq 1 - \xi_i, \sum_k \lambda_k = 1, \xi_i \geq 0, \lambda_k \geq 0$$

où $f_k \in \mathcal{H}_k$ est le RKHS associé à la fonction K_k.

Pour chaque noyau d'attribut visuel, nous construisons un noyau de graphe qui permet de mesurer la similarité entre graphes à partir de ce seul attribut. Certaines caractéristiques sont plus discriminantes que d'autres pour certaines données. La fusion permet alors de construire une fonction de décision exploitant l'hétérogénéité des attributs et qui pondère l'importance de chacun en fonction de sa pertinence pour la classification, via les paramètres estimés λ_i. Cette fusion permet d'améliorer les performances de la classification dans le cas où aucun attribut ne permet seul une bonne classification (ce qui est généralement le cas).

Expériences Les avantages de cette dernière étape d'apprentissage sont illustrés expérimentalement dans les tableaux 6.1, 6.2 and 6.3 qui présentent les performances pour l'algorithme d'apprentissage multiple pour les trois vues de la base de données.

Dans la plupart des cas, les résultats montrent l'amélioration de la performance par rapport aux taux de catégorisation initiaux, et donc l'intérêt d'utiliser cette méthode non supervisée de sélection d'attributs pour l'apprentissage appliqué aux images. Dans la projection sagittale, les fonctions noyau sélectionnées par l'apprentissage multiple ont de bonnes performances, et leur combinaison améliore encore ces performances. Un comportement fréquent est la sélection de fonctions noyau différentes qui s'appuient sur le même attribut visuel, ce qui représente une amélioration fondamentale par rapport aux approches grid-search classiques, où un seul choix des paramètres des fonctions noyau nous est imposé. Le noyau multiple ne peut pas garantir dans tous les cas une

minimisation stricte de l'erreur par rapport aux composantes (par exemple tableau 6.1) mais son intérêt vient du fait qu'en pratique on peut sélectionner un sous-ensemble de descripteurs pertinents et qu'on arrive à obtenir une bonne performance collective si les informations qu'ils apportent sont pertinentes et complémentaires.

Conclusion et perspectives

Le but du travail présenté dans ce document est de proposer un cadre général et modulaire pour l'interprétation d'images fondé sur des représentations structurées et l'intégration d'informations spatiales floues. Même si nous avons choisi d'illustrer notre méthode en utilisant des applications médicales, la méthode n'est pas fondamentalement liée à un type spécifique d'images d'entrée. L'aspect modulaire de notre méthode réside dans la polyvalence du modèle graphique sous-jacent , qui assure l'indépendance entre les points d'intérêt que nous avons choisis de représenter en tant que sommets, et les relations spatiales que nous choisissons pour les associer aux arêtes.

Dans le cadre de cette solution pour la représentation d'images, nous avons commencé notre travail en présentant un noyau graphique marginalisé qui étend celui utilisé en biologie computationnelle, et qui permet l'interprétation des modèles graphiques extraits à partir d'images. Nous avons appliqué cette approche pour une tâche simple de catégorisation, dont la partie d'extraction est fondée sur une méthode de segmentation générique. Les résultats obtenus valident notre approche et soulignent l'intérêt de prendre en compte des information spatiales plus riches que l'adjacence stricte, ainsi que de construire un classifieur hétérogène à partir de caractéristiques extraites.

Ensuite, nous avons proposé l'utilisation d'un ensemble de descripteurs spatiaux flous qui sont liés aux relations topologiques entre les régions et nous avons étudié les avantages offerts par les représentations qui utilisent ce type d'information supplémentaire. Nous avons montré que la compréhension de la répartition de l'information de ces descripteurs est importante pour une meilleure intégration avec le reste du modèle et que l'ajout d'information spatiale, qui implique des changements dans la structure des représentations, peut influencer dans une large mesure la performance de l'algorithme d'apprentissage. Nous avons également présenté une approche qui peut être utilisée pour ces type de modèles graphiques, afin de sélectionner automatiquement et de fusionner, parmi un ensemble de caractéristiques visuelles et spatiales, l'information qui donne la meilleure performance discriminative dans un contexte spécifique.

La direction adoptée par ces travaux est justifiée par l'écart existant entre l'importance de l'information spatiale dans l'interprétation d'images, et sa faible présence dans les modèles proposés dans la littérature. La méthode que nous proposons est modulaire et adaptable, et même si les traitements sont moins rapides que dans les méthodes typiques "sacs de mots", le coût que l'on dépense dans l'augmentation de la complexité structurelle n'est pas prohibitif. En outre, un avantage potentiel de cette méthode est que les calculs critiques de similarité demandés par le classificateur peuvent être exécutés en parallèle, ce qui la rend adaptée à des environnements de calcul distribué.

Dans le contexte de l'apprentissage structurel pour l'interprétation des images, une lacune majeure des machines à vecteurs de support, liée à la compréhension du modèle discriminatif appris, est en partie atténuée par l'apprentissage multiple de noyaux différents. L'utilisation de la sortie MKL nous permet de mieux comprendre les fonctions qui sont efficaces et les corrélations entre les différentes vues des données qui sont discriminatoires.

La difficulté principale que nous avons clairement identifiée pour notre méthode est l'obligation de fournir une entrée visuelle structurée qui soit appropriée pour une interprétation de haut niveau. Le défaut de le faire a deux conséquences négatives. Tout d'abord, en présence d'erreurs de pré-traitement, les structures graphiques d'entrée sont généralement plus grandes que nécessaire (ex. sursegmentation), avec une conséquence négative sur les temps de calcul. Deuxièmement, si les nœuds des graphe ne sont pas véritablement associés aux points clés des images, notre méthode a une utilité limitée.

Perspectives De nombreuses perspectives sont mises en évidence par notre travail, puisque chaque partie de notre cadre d'apprentissage peut être améliorée et étendue de façon indépendante. Il existe cependant quelques directions majeures qui sont sans doute très intéressantes à explorer.

Une perspective prometteuse est liée aux caractéristiques des points d'intérêt qui sont associés aux nœuds. La nature des points d'intérêt est théoriquement contrainte par les limites de la méthode de calcul (une complexité polynomiale relativement élevée par rapport au nombre de nœuds). Cependant, la complexité sémantique des points d'intérêt détermine directement le niveau sémantique de la méthode d'apprentissage. Dans le cas des régions, le niveau sémantique est plus élevé que dans le cas des approches "sacs de mots", mais le principal inconvénient est que la qualité de la partie de pré-traitement (dans ce cas, la segmentation d'image) doit être élevée aussi. D'autre descripteurs fondés sur les régions, notamment ceux liés aux mécanismes de pré-attention, sont moins complexes sémantiquement et robustes par rapport au pré-traitement. Ces descripteurs ont déjà été employés pour la modélisation, mais le principal défi serait probablement de déterminer ceux qui sont les mieux adaptés pour l'apprentissage discriminatif, tout en gardant la taille du modèle graphique assez petite pour que les temps de calcul soient raisonnables. Dans ce cas, une perspective supplémentaire est représentée par des relations spatiales floues adaptées à d'autres types de points d'intérêt.

Une autre perspective intéressante est liée à la formalisation du gain d'information pour les relations spatiales floues. Pour les mesures d'adjacence floue, nous avons montré que l'information d'adjacence peut prendre des valeurs plus ou moins informatives, et la distribution de ces valeurs utiles est corrélée à la signification de chaque relation spatiale. La prise en compte de la distribution pour les modèles graphiques qui représentent des données spécifiques permet à l'algorithme d'apprentissage de s'appuyer plus sur le concept sémantique qui est derrière la relation spatiale.

Certes, un cadre de raisonnement en utilisant des points clés fondés sur les régions qui sont liés par des relations spatiales complexes présente un grand potentiel d'inférence

avec des concepts sémantiques, mais il est naturellement sensible à la qualité structurelle des données d'entrée. Une perspective prometteuse est d'associer cette méthode avec un traitement de bas niveau dans le but de "fermer la boucle" et de permettre aux deux parties d'effectuer itérativement les tâches de traitement et d'interprétation de façon mutuellement bénéfique.

Chapter 1

Introduction

In machine learning, supervised statistical methods have been related to the development and the progress of reliable algorithms that cope very well with numerical inputs such as N-dimensional arrays, and perform various predictive tasks, such as classification or regression.

Improved usability, technological breakthroughs and other various factors have metamorphosed computers and digital devices, and reinforced their presence in our everyday activities and in most, if not all, of the scientific domains. Often, we become aware that our data processing algorithms are overwhelmed and do not cope with new types of information, issued from these activities or domains. Complex inputs that we can not convert into the usual array representations emerge from natural sciences (biology, chemistry), social sciences (linguistics, psychology, sociology), health sciences, engineering, etc., in the form of trees, graphs or hypergraphs. The fundamental property that distinguishes these representations is the presence of *structure*, in the form of parts and relations between/among parts.

The objective of this work is to extend novel methods that have been recently proposed in supervised learning for structured data represented as graphs, and to take into account fuzzy relations. Typically, graph vertices represent basic entities which present specific features, and graph edges encode relations between these basic entities. These relations might point to a physical interaction in a chemical representation, or to a dependency in a compilation diagram. Reasoning in a fuzzy framework means that we can associate a degree of confidence to a specific relation or feature. This formalism is especially beneficial for representations related to domains where uncertainty or vagueness has to be taken into account: signal processing, natural language processing, robotics, etc. Even though a lot of research has been devoted to finding appropriate representations of knowledge under uncertainty assumptions, there is no defined formalism for learning structured data representations which incorporate fuzzy information. The main contribution of this work is to propose an approach for including fuzzy relations into structured data learning.

Figure 1.1: A pair of images that we use in order to illustrate some fundamental issues in image processing.

A motivating example One of the domains which has benefited most from the fuzzy formalism has been image processing. In graphical models, vertices represent visual entities and edges encode spatial relations between entities such as adjacency, distance, relative positioning, etc. Spatial imprecision between entities is generated by the linguistic vagueness of the spatial concepts, by the morphological variability of the underlying structures being represented, and also by the shortcomings imposed by the preprocessing algorithms that extract the visual entities from the image. The fuzzy formalism is adapted for addressing these imprecisions and is known to improve the robustness of visual representations.

We mention some open problems in image processing using as illustration the pair of images in Figure 1.1:

1. *Where are the objects located in these images?*

2. *Does each image represent a single object, an arbitrary scene devoid of any meaning, or a semantic concept?*

3. *If appropriate, do the images represent the same class of objects/the same semantic concept?*

4. *If appropriate, do the images represent the same object (class instance)?*

5. *If appropriate, what is the most accurate class model that may be built using these two objects?*

All these problems are more or less related, and any learning algorithm attempting to approach them would greatly benefit from an adapted representation of spatial information depicting the image content. Even for special types of visual data, such as medical or satellite images, that may benefit easier from prior knowledge in the form of atlases, reference models or ontologies for example, interpretation algorithms must cope with imprecision and uncertainty. Structure, as well as part features, represents a valuable source of information that may be employed with very effective results.

Our interest focuses on discriminative learning for *image categorization*. This particular task consists in labeling an input image into one of some predefined categories, and has proved to be a difficult task in computer vision. It addresses directly the last three problems depicted above. Image categorization can also help at solving the first two problems, and at the same time benefit from effective solutions that address the first two problems.

Related work Existing approaches in image interpretation are very heterogeneous, a fact which is related to the difficulty of the underlying tasks and to the diversity of image types.

Probabilistic frameworks are well adapted to the variability of the visual features of interest points in the image. In the absence of spatial reasoning, these frameworks endow learning algorithms with the ability to cope with uncertainty at low level or high level, to some extent. The difference between various methods is defined, among other things, by the approach taken in order to take into account spatial information. In some approaches (which are nowadays rarely used due to their obvious limitations), spatial information may be completely discarded, the image being effectively vectorized. In other approaches, local low-level features are clustered, thus adding to the resulting vectorial image representation an implicit structural information. Finally, some approaches use explicit structural representations such as graphs, trees, ontologies, that are either inferred from examples or given by prior knowledge, but models are more or less rigid and since they do not tolerate well variability, they are usually appropriate only for specific types of images.

Our objectives In a first step, our objective is to adapt discriminative machine learning methods that are used for learning structured data represented as graphs in other domains , and to estimate the applicability of these methods on graphical models of images where we take into account simple topological spatial information such as strict pixel adjacency of image parts. We are interested in creating labeled graphical models that are structurally similar to the underlying images, and that embed relevant image part features into the corresponding graph vertices. Under this approach, difficulties arise from the fundamental differences in graph structural properties and vertex labeling between image related graphs and simpler graphical models from domains such as computational biology or chemistry.

A subsequent step is to determine adapted fuzzy representations of topological spa-

tial relations between image parts, in the context of discriminative supervised learning. The major difficulty of this part of our work consists in efficiently adding this new type of information to a graphical representation, since by adding fuzzy edges we are constantly changing the representation. Therefore, this task may be rephrased as finding the appropriate graphical model of the image that is based on fuzzy spatial relations and that carries the optimal amount of useful information for the image interpretation algorithm.

Finally, a challenge that must also be addressed is that different spatial relations create different representations of the same image, and we should be able to take into account these representations simultaneously in order to benefit from different views of the data. In reality, the regular machine learning algorithms use different features and relations sequentially, and afterwards we must use appropriate learning methods in order to enhance the method performance by combining all the basic learners optimally.

The structure of the document

Chapter 2: this chapter introduces the central concept of statistical learning, support vector machines, the first examples of structured data, and an overview of different kernel functions for structured data, and particularly for labeled graphs.

Chapter 3: we present some challenges that must be addressed when specific kernel functions are extended to graphical models with more complex labeling, and we also discuss the advantages of methods using graph kernels, as compared to methods using subgraph matching.

Chapter 4: this part reviews the main types of image features and representations that are currently used in image processing, along with the most popular approaches to image interpretation. We propose a first methodological approach to structured image interpretation based in an original way on strict adjacency graphs and marginalized graph kernels.

Chapter 5: we introduce the fuzzy formalism, and as an original feature of the method, adapted measures of fuzzy adjacency between image regions. We investigate optimal graphical representations that take into account fuzzy spatial relations and their impact on the performance of discriminative learning algorithms, and we propose a method for filtering out detrimental spatial information. This is another contribution of this thesis.

Chapter 6: finally, as a last contribution, we introduce the multiple kernel learning framework and we explore its applicability for the specific case of classifiers related to different fuzzy spatial representations of input images.

Chapter 7: we present the conclusion of our work, and different perspectives.

Chapter 2

Statistical learning theory and structured data

2.1 Introduction

Statistical learning is an important topic in machine learning, and a specific formulation of the more general concept of learning (Simon, 1984):

> *"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time"*

Let us start by defining the formal representation of data that statistical learning handles. We assume that the input for a learning problem consists of a set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ that are used to learn a function $f : \mathcal{X} \to \mathcal{Y}$. In a statistical learning framework, we assume that the pairs in $\mathcal{D}$, along with the future pairs (x, y), are generated by the same, unknown probability P on $\mathcal{X} \times \mathcal{Y}$.

In order to estimate the quality of our learner, one needs to take into account a *loss function* $L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$, $L(x, y, f(x))$ being smaller when the estimated response $f(x)$ is better. Since it is unfeasible to estimate the fitness of an estimation $f(x)$ for a specific pair (x, y), one considers the *expected* loss of f over the entire input variables space. This is called the *risk* of f:

$$\mathcal{R}_{L,P}(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(x, y, f(x))dP(x, y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(x, y, f(x))dP(y|x)dP_{\mathcal{X}} \tag{2.1}$$

We assume that a function f is better the smaller its risk $\mathcal{R}_{L,P}(f)$ is. We are therefore interested in the quantity:

$$\mathcal{R}^*_{L,P} = \inf_{f:\mathcal{X}\to\mathbb{R}} \mathcal{R}_{L,P}(f) \tag{2.2}$$

which is the smallest possible risk over all possible functions. The distribution P is unknown and therefore it is impossible to find the minimizer without additional information. In statistical learning, the set $\mathcal{D}$ carries this additional information, thus

allowing us to build a function $f_{\mathcal{D}} : \mathcal{X} \to \mathbb{R}$ whose risk $\mathcal{R}_{L,P}(f_{\mathcal{D}})$ is close to the optimal risk $\mathcal{R}^*_{L,P}$.

We can formalize the learning property of our system according to the definition above in the following way, known as the *universal consistency* principle. This principle requires that for all possible distributions P on $\mathcal{X} \times \mathcal{Y}$, the functions $f_{\mathcal{D}}$ computed by the learner satisfy

$$\mathcal{R}_{L,P}(f_{\mathcal{D}}) \to \mathcal{R}^*_{L,P},\ n \to \infty \tag{2.3}$$

Although the requirement that Equation 2.3 holds for all P is quite strong, for certain loss functions it has been shown that one can build such learning methods (Stone, 1977).

The universal consistency assumption does not imply anything about the speed of convergence, however. A formalization of this issue states that for sufficiently ominous distributions P, the average risk of any classification method may tend *arbitrarily* to zero (Devroye, 1982). Nevertheless, it is possible under certain additional mild assumptions on P to derive uniform convergence rates (Lehmann and Casella, 2003; Rieder, 1994; Györfi et al, 2002).

2.2 Support Vector Machines

2.2.1 Empirical Risk Minimization and Support Vector Machines

We will now continue by introducing support vector machines (SVMs) and we will focus on a particular type of learning problem, namely binary classification. The purpose of binary classification being the prediction of one of two states, we can redefine the output space by $\mathcal{Y} = \{-1, 1\}$. Consequently, the generic classification loss function L_c is defined as:

$$L_c = \begin{cases} 0\ ,\ if\ f(x) = y \\ 1\ ,\ otherwise \end{cases} \tag{2.4}$$

The purpose of the learning part is to find a function f^* that minimizes the estimation of the risk (2.2), where $L = L_c$. Since the distribution P is unknown, we use the set $\mathcal{D}$ by approximating the risk function by its empirical counterpart:

$$\mathcal{R}_{L,\mathcal{D}} = \frac{1}{n} \sum_{i=1}^{n} L\left(x_i, y_i, f(x_i)\right) \tag{2.5}$$

Although for a *specific* f, $\mathcal{R}_{L,\mathcal{D}}(f)$ is asymptotically converging to $\mathcal{R}_{L,P}(f)$, the optimal learner for $\mathcal{R}_{L,\mathcal{D}}(f)$ is not an approximation of the minimizer for $\mathcal{R}_{L,P}(\cdot)$. This phenomenon, called *overfitting*, outputs a learner that models too closely the examples in $\mathcal{D}$ and fails to perform well on future data.

Overfitting may be avoided by restricting the set $\mathcal{F}$ of functions f that is supposed to hold a reasonably good learner. Under this new assumption, known as *empirical risk minimization*, the risk function to minimize becomes:

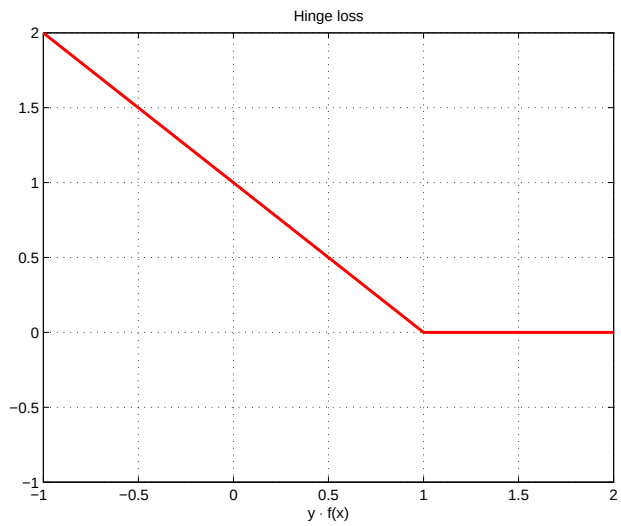


Figure 2.1: The hinge loss plot. If $yf(x) > 1$, the point is correctly classified and the penalty is null. Otherwise, the loss is positive and increases linearly.

$$\mathcal{R}^*_{L,P,\mathcal{F}} = \inf_{f \in \mathcal{F}} \mathcal{R}_{L,P}(f) \tag{2.6}$$

Although extremely useful, this restriction of the class of functions f to the set $\mathcal{F}$ raises two major issues. First of all, the prior information concerning the unknown distribution P is too scarce to allow for a good identification of the set $\mathcal{F}$. Therefore, we cannot bound the error

$$\mathcal{R}^*_{L,P,\mathcal{F}} - \mathcal{R}^*_{L,P} \tag{2.7}$$

of this restricted model with respect to the general one. This is usually avoided by increasing the size of the set $\mathcal{F}$ with n. The second issue is that often solving Equation 2.6 is computationally infeasible.

We continue by explaining how support vector machines are fundamentally designed to cope with these two challenges. First of all, the classification loss L_c (Equation 2.4) is replaced by a convex approximation, namely the hinge loss (Figure 2.1):

$$L_{hinge}(y, f(x)) = \max\{0, 1 - yf(x)\}, \quad y \in \{-1, 1\}, \quad f(x) \in \mathbb{R} \tag{2.8}$$

Since $\mathcal{R}_{L_{hinge},\mathcal{D}}(f)$ is convex in f and under the assumption that $\mathcal{F}$ is convex, the problem $\inf_{f \in \mathcal{F}} \mathcal{R}_{L_{hinge},\mathcal{D}}(f)$ is also convex, thus computationally feasible. The second important property of support vector machines addresses at the same time the computational feasibility of the learner and the variation of $\mathcal{F}$. SVMs consider specific sets of functions, denoted as *reproducing kernel Hilbert spaces* H, which will be introduced more thoroughly in Section 2.3.

2.2.2 A Geometrical Interpretation

The maximal margin (maximal distance between the border separating the two classes and the elements of $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$) idea is relatively old and was employed in the *generalized portrait algorithm* (Vapnik and Lerner, 1963). We assume that the input space $\mathcal{X}$ is a subspace of $\mathbb{R}^d$, and that we have an element $w \in \mathbb{R}^d$ with $\|w\|_2 = 1$, and a real number b such that:

$$\begin{cases} \langle w, x_i \rangle + b > 0, \ , \ \forall i \ with \ y_i = +1 \\ \langle w, x_i \rangle + b < 0 \ \ , \ \forall i \ with \ y_i = -1. \end{cases} \tag{2.9}$$

The intuition behind these constraints is simply that elements with the same sign will be placed on the same side of the separation hyperplane $\langle w, x \rangle + b = 0$. The solution of the algorithm is the pair $(w_{\mathcal{D}}, b_{\mathcal{D}})$ with $\|w_{\mathcal{D}}\|_2 = 1$ that satisfies 2.9 and has maximal margin. This pair generates the following classifier function:

$$f_{\mathcal{D}}(x) = \operatorname{sgn}\left(\langle w_{\mathcal{D}}, x \rangle + b_{\mathcal{D}}\right) \tag{2.10}$$

The maximization problem along with the constraints 2.9 can be reorganized into the following optimization problem:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \langle w, w \rangle \ \text{ subject to} \tag{2.11}$$

$$y_i \left(\langle w, x_i \rangle + b\right) \geq 1 \qquad i = 1, \ldots, n.$$

This simple and effective algorithm (Figure 2.2a) has two major drawbacks:

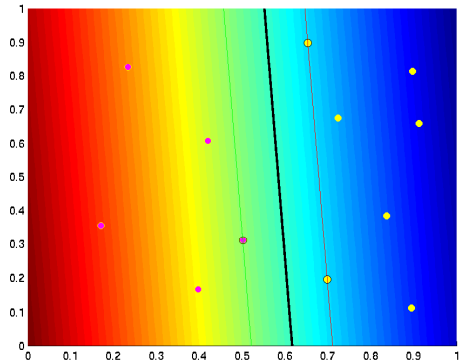
- In most real classification problems, linear classifiers are not appropriate (Figure 2.2c) and/or a linear solution does not exist (Figure 2.2b).
- In order to avoid overfitting, some training points (possible outliers) must be intentionally misclassified.

The intuition behind the adjustment that overcomes the first drawback is the following: if some points are not linearly separable in a given input space, they may become so if we project them into another space, usually a higher dimensional one. In Boser et al (1992), input data points are mapped into a Hilbert space $\mathbb{H}$ known as the *feature space* by a typically non-linear *feature map* $\Phi : \mathcal{X} \to \mathbb{H}$. Then, the portrait algorithm is applied to the mapped data set $((\Phi(x_1), y_1), \ldots, (\Phi(x_n), y_n))$, the linear separation thus taking place in $\mathbb{H}$:

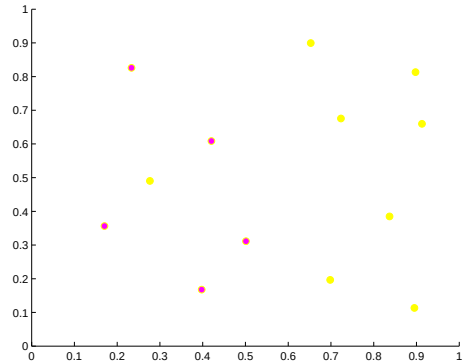
$$\min_{w \in \mathbb{H}, b \in \mathbb{R}} \langle w, w \rangle \ \text{ subject to} \tag{2.12}$$

$$y_i \left(\langle w, \Phi(x_i) \rangle + b\right) \geq 1 \qquad i = 1, \ldots, n.$$

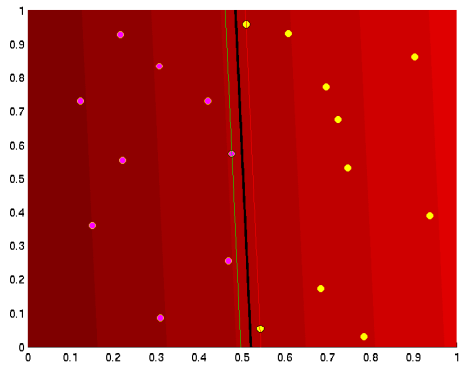
For certain mappings Φ, the first drawback is successfully addressed (Figure 2.2d); however, plunging the input data in a higher dimensional feature space aggravates the second issue, caused by overfitting (Figure 2.2e).



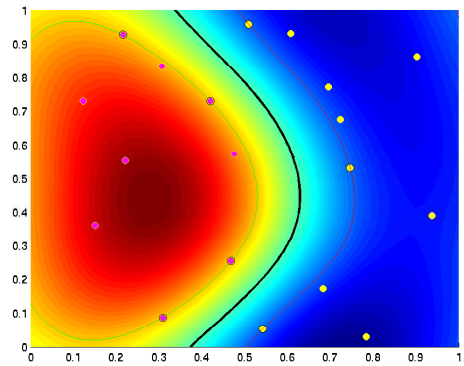
(a) In this first example, the portrait algorithm performs well.



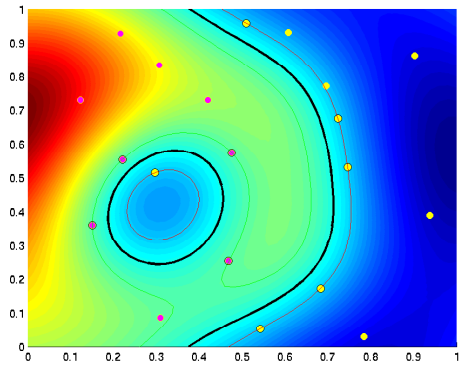
(b) With an extra point as outlier, the problem can not be solved by the portrait algorithm.



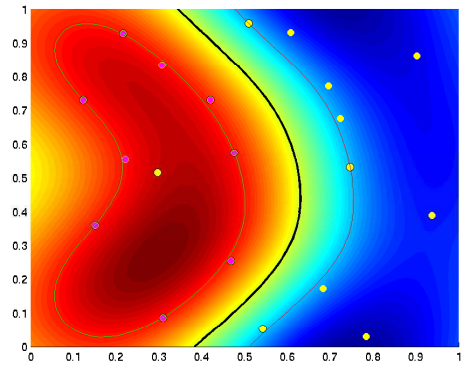
(c) Even if a linear classifier exists for this second toy problem, it is quite poor.



(d) A better, non linear classifier for the same problem (RBF, $\sigma = 0.4$).



(e) For this last toy problem, a large $C = 10^5$ leads to overfitting (RBF, $\sigma = 0.4$).



(f) With $C = 5$, the classifier ignores the outlier and generalizes better (RBF, $\sigma = 0.4$).

Figure 2.2: Several binary classification examples on toy datasets and within different contexts.

More recent work (Cortes and Vapnik, 1995) addressed this latter issue, by introducing *soft margin support vector machines*. In the initial approach, the constraints $y_i(\langle w, x_i\rangle + b) \geq 1$ imposed that each element in $\mathcal{D}$ would be correctly classified by $f_{\mathcal{D}}(x)$. The novelty in soft margin SVMs is the introduction of *slack variables* ξ_i that relax the initial constraints in the following way: $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$. Adding the slack variables to the objective function keeps their values relatively low, so that the initial constraints do not become trivial. The resulting optimization problem is:

$$\min_{w\in\mathbb{H}, b\in\mathbb{R}, \xi\in\mathbb{R}^n} \langle w, w\rangle + C\sum_{i=1}^{n} \xi_i \text{ subject to} \tag{2.13}$$

$$y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \qquad i = 1, \ldots, n.$$

where $C > 0$ is a fixed parameter that is used to weight the second term of objective function against the first. The lower C is, less will the classifier penalize outliers that it will choose to ignore in order to optimize the margin (Figure 2.2f).

2.2.3 Solving the SVM

The optimization problem 2.13 takes place in the feature space $\mathbb{H}$ which is often high-dimensional, or infinite-dimensional. Therefore, we can process it by using *Lagrange multipliers* $\alpha = (\alpha_1, \ldots, \alpha_n) \geq 0$ for the constraints $y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i$ and $\beta = (\beta_1, \ldots, \beta_n) \geq 0$ for the constraints $\xi_i \geq 0$:

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left[\xi_i - 1 + y_i(w^T\Phi(x_i) + b)\right] - \sum_{i=1}^{n} \beta_i \xi_i \tag{2.14}$$

In order to find the optimum of 2.13, we need to minimize the Lagrangian with respect to (w, b, ξ):

$$\frac{\partial \mathcal{L}}{\partial w}(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{n} y_i \alpha_i \Phi(x_i) = 0 \tag{2.15}$$

$$\frac{\partial \mathcal{L}}{\partial b}(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{2.16}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i}(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0, \qquad i = 1, \ldots, n. \tag{2.17}$$

Using Equations 2.15 and 2.16 into 2.14, we obtain the *dual problem*:

$$\max_{\alpha\in[0,C]^n, \sum_{i=1}^{n} y_i\alpha_i = 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \Phi(x_i)^T \Phi(x_j), \tag{2.18}$$

where the boxing constraint $\alpha \in [0, C]^n$ is a direct implication of 2.17.

Once α is found using one of the many methods that were proposed in the litterature (Keerthi et al, 2001; Fan et al, 2005; Chen et al, 2006), we explicit β using $\beta_i = C - \alpha_i, \forall i = 1, \ldots, n$. In order to compute b, we use the KarushÐKuhnÐTucker conditions for the non-zero Lagrange multipliers. For a certain i with $0 < \alpha_i < C$ the constraints $\xi_i = 0$ and $\xi_i - 1 + y_i(w^T\Phi(x_i) + b) = 0$ hold, therefore;

$$b = y_i - w^T\Phi(x_i) = y_i - \sum_{j=1}^{n} y_j\alpha_j\Phi(x_j)^T\Phi(x_i). \tag{2.19}$$

When predicting the class of a new object $x \in \mathcal{X}$, we estimate the classifier response as:

$$f_{\mathcal{D}}(x) = w^T\Phi(x) + b = \sum_{j=1}^{n} y_j\alpha_j\Phi(x_i)^T\Phi(x) + b, \tag{2.20}$$

and predict the class of x depending on the sign of this function.

2.3 Kernels and Reproducing Kernel Hilbert Spaces

In the previous section we mentioned that feature maps and feature spaces are useful tools by which the initial linear SVM approach produces non-linear decision functions.

A crucial observation is that SVMs require computing inner products $\langle\Phi(x), \Phi(x')\rangle_{\mathcal{H}}$ in order to estimate $f_{\mathcal{D}}(x)$ for a new element $x \in \mathcal{X}$ that we want to classify. By $k(x, x')$ we denote the function $\langle\Phi(x), \Phi(x')\rangle_{\mathcal{H}}$. Instead of defining Φ then computing the necessary inner products, we can use support vector machines with a restricted family of functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that express at the same time inner products in possibly unknown feature spaces. The purpose of this section is to briefly describe the constraints on these functions, known as *kernel functions*, that ensure good classifier performance and efficient computation.

Definition 2.1. *Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$ is called a* kernel *on $\mathcal{X}$ if there exists a $\mathbb{K}$-Hilbert space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$ we have:*

$$k(x, x') = \langle\Phi(x), \Phi(x')\rangle_{\mathcal{H}} \tag{2.21}$$

We call Φ a feature map *and $\mathcal{H}$ a* feature space *of k.*

Definition 2.2. *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called* positive-definite *if, for all $n \in N, \alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and all $x_1, \ldots, x_n \in \mathcal{X}$, we have:*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j k(x_i, x_j) \geq 0. \tag{2.22}$$

k is said to be strictly positive definite *if for mutually distinct $x_1, \ldots, x_n \in \mathcal{X}$ equality only holds for $\alpha_i = 0, i = 1, \ldots, n$. Finally, k is called* symmetric *if $k(x, x') = k(x', x)$.*

If k is an $\mathbb{R}$-valued kernel with feature map $\Phi : \mathcal{X} \to \mathcal{H}$ then k is obviously symmetric. It is also positive definite since for $n \in N, \alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and all $x_1, \ldots, x_n \in \mathcal{X}$, we have:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \left\langle \sum_{i=1}^{n} \alpha_i \Phi(x_i), \alpha_j \Phi(x_j) \right\rangle_{\mathcal{H}} \geq 0. \tag{2.23}$$

The following theorem states states that symmetry and positive definiteness are not only neecessary for k to be a kernel, but also sufficient.

Theorem 2.1 (Symmetric, positive definite functions are kernels)**.** *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if it is symmetric and positive definite.*

Theorem 2.1 is very useful for checking whether a function k that we consider with respect to a given input space $\mathcal{X}$ is a kernel or not. Some of the most commonly used kernel functions are:

- linear kernel, $k(x, x') = \langle x, x' \rangle$,
- polynomial kernel, $k(x, x') = (\langle x, x' \rangle + l)^p$,
- sigmoid kernel, $k(x, x') = \tanh(\gamma \langle x, x' \rangle)$,
- Gaussian kernel, $k(x, x') = \exp\left[- \|x - x'\|^2 / 2\sigma^2 \right]$.

The family of kernel functions exhibits a number of closure properties (Genton, 2002; Taylor and Cristianini, 2004), amongst which the simplest are the linear combination, the multiplication, the exponentiation etc. These properties are useful for the design of more complex kernels that might be more adapted to particular input data.

In the following part of this section, we will try to explicit the way a kernel k determines implicitly the function space $\mathcal{F}$ that we use to restrict our search for an optimal learner f.

Definition 2.3. *Let $\mathcal{X}$ be a non-empty set and $\mathcal{H}$ be a $\mathbb{K}$-Hilbert space over $\mathcal{X}$ that consists of functions mapping from $\mathcal{X}$ into $\mathbb{K}$.*

- *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$ is called a* reproducing kernel *of $\mathcal{H}$ if we have $k(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and the reproducing property:*

$$f(x) = \langle f, k(\cdot, x) \rangle$$

- *The space $\mathcal{H}$ is called a* reproducing kernel Hilbert space *(RKHS) over $\mathcal{X}$ if for all $x \in \mathcal{X}$ the Dirac functional $\delta_x : \mathcal{H} \to \mathbb{K}$ defined by :*

$$\delta_x(f) = f(x), \qquad f \in \mathcal{H}$$

  is continuous.

Theorem 2.2. *Every kernel k has a unique RKHS (for which k is a reproducing kernel).*

Rather than simply minimizing $\mathcal{R}_{L,\mathcal{D}}(f)$ (Equation 2.5), support vector machines take into account a supplementary regularization term that is tied to the complexity of the function f, using a fixed $\lambda > 0$:

$$\inf_{f \in \mathcal{H}} \lambda \left\|f\right\|_{\mathcal{H}}^2 + \mathcal{R}_{L,\mathcal{D}}(f), \tag{2.24}$$

in order to reduce the danger of overfitting. The following theorem is extremely useful for solving this optimization problem.

Theorem 2.3 (Representer theorem (Kimeldorf and Wahba, 1971))**.** *Let $L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ be a convex loss and $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$ the input for the learning process. Let $\mathcal{H}$ be a RKHS over $\mathcal{X}$. Then, for all $\lambda > 0$ there is a unique $f_{\mathcal{D},\lambda} \in \mathcal{H}$ satisfying :*

$$\lambda \left\|f_{\mathcal{D},\lambda}\right\|_{\mathcal{H}}^2 + \mathcal{R}_{L,\mathcal{D}}(f_{\mathcal{D},\lambda}) = \inf_{f \in \mathcal{H}} \lambda \left\|f\right\|_{\mathcal{H}}^2 + \mathcal{R}_{L,\mathcal{D}}(f) \tag{2.25}$$

Moreover, there exists $\alpha \in \mathbb{R}^n$ such that

$$f_{\mathcal{D},\lambda}(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i), \qquad x \in \mathcal{X} \tag{2.26}$$

The main implication of this result is that the minimizer $f_{\mathcal{D},\lambda}$ is a weighted average of at most n functions $k(x_i, \cdot)$. A remarkable direct consequence of the representer theorem is the fact that $f_{\mathcal{D},\lambda}$ is contained in a known finite dimensional space, even if the space $\mathcal{H}$ itself is substantially larger (even infinite dimensional).

If we consider a hinge loss L (Equation 2.8), we can rewrite the optimization associated to our geometrical approach of SVMs (Equation 2.13) in the following way:

$$\min_{(w,b) \in \mathcal{H}_0 \times \mathbb{R}} \lambda \langle w, w \rangle + \frac{1}{n} \sum_{i=1}^{n} L\left(x_i, y_i, f_{(w,b)}(x_i)\right), \tag{2.27}$$

with $\lambda = 1/(2nC)$. There is a major difference between Equation 2.27 and the RKHS formulation:

$$\inf_{f \in \mathcal{H}} \lambda \left\|f\right\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} L\left(x_i, y_i, f(x_i)\right). \tag{2.28}$$

In the first case, we consider a general Hilbert space $\mathcal{H}_0$ and we search for a hyperplane specified by (w, b), while in the second case we start with a RKHS $\mathcal{H}$. However, modulo the offset term b, the two approaches are similar on account of the following theoretical result:

$$\left\|f\right\|_{\mathcal{H}} = \inf\left\{\|w\|_{\mathcal{H}_0} : w \in \mathcal{H}_0 \;\; with \;\; f = \langle w, \Phi(\cdot) \rangle\right\} \tag{2.29}$$

where Φ is the feature map associated to $\mathcal{H}_0$.

2.4 Characteristics of good kernels

We have explained so far how kernel functions implicitly create a data mapping into a feature space (this fact being summerized by Equation 2.21). The choice of a kernel function is therefore of outmost importance for the learning process. However, beside positive definiteness, kernels should present some additional properties that qualify them as good for a specific input space $\mathcal{X}$.

A very important property, usually denoted as *completeness*, is related to Φ being an injection on its domain $\mathcal{X}$. If $\Phi(x) = \Phi(x')$ for $x \neq x'$, this implies that we are not able to distinguish between x and x', which may cause difficulties. In terms of the kernel function, completeness is present if $k(x, \cdot) = k(x', \cdot)$ implies that $x = x'$.

Another property, known as *appropriateness* of the kernel, requires that elements that are close (or similar, in the absence of a metric) in the input space $\mathcal{X}$ be close in the feature space, too. Since the distance in the feature space is given by:

$$d(x, x')^2_{\mathcal{H}} = \|\Phi(x) - \Phi(x')\|^2_{\mathcal{H}} = k(x, x) + k(x', x') - 2k(x, x'), \tag{2.30}$$

this can help us decide if a kernel is appropriate or not. For a large class of kernels, which are called stationary and which depend only on the difference between input elements: $k(x, x') = k_S(x - x')$, the distance is decreasing with respect to $k(x, x')$:

$$d(x, x')^2_{\mathcal{H}} = 2k_S(0_{\mathcal{X}}) - 2k(x, x') = 2k_S(0_{\mathcal{X}}) - 2k_S(x - x'). \tag{2.31}$$

Hence, it makes sense to interpret the kernel as a similarity estimation function for the pair (x, x'): if the similarity increases, the distance between elements decreases.

2.5 Alternatives to SVMs in machine learning

There exists a broad effort in the scientific community directed towards developing learning algorithms for pattern recognition and data mining. The books by Duda et al (2001) or Bishop (2007) provide a valuable insight on alternatives to support vector machines, along with their advantages and weak points: k-nearest-neighbor methods (Fix and Hodges, 1951; Stone, 1977), logistic regression (Cox and Snell, 1989), neural networks (Bishop, 1996; Anthony and Bartlett, 1999; Vidyasagar, 2002) or boosting (Schapire, 1990; Freund and Schapire, 1997). Somewhat along support vector machines, various kernel based methods have been proposed and investigated too (Cristianini and Shawe-T 2000; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004). Comparisons between all these methods is beyond the scope of our document, but some of the advantages of SVMs that should be considered when choosing the right approach for a problem are the predictive power, the ability to detect simple and complex dependencies, and the robustness with respect to outliers (Steinwart and Christmann, 2009).

2.6 Structured Data

Support vector machines can be effectively applied to all types of data that can be embedded in a Euclidean space, typically $\mathbb{R}^n$. However, for many machine learning applications, input data cannot be naturally depicted using numerical array representations. This is typically the case of *structured data*, which will be described in this section.

Very often, information extraction on a real world entity will output sub-entities, relations and labels. Typical examples of real world entities include gene sequences, lexical units, social networks, drug interactions etc. Integrating all this information into a computer comprehensible data structure is a non-trivial task, and the choice of a particular data structure depends on the properties of the initial entity. Strings (Lodhi et al, 2002; Leslie et al, 2002; Vert, 2002a; Leslie and Kuang, 2003; Leslie et al, 2004), trees (Collins and Duffy, 2001; Vert, 2002b; Vishwanathan and Smola, 2002) and graphs (Holm and Sander, 1993; Manning and Schütze, 1999; Tsuda et al, 2002; Kashima et al, 2003; Gaertner et al, 2003) are usually used for this purpose. In the following, we will assume that $\mathcal{X}$, the input space for the learning task, is the data structure space, and not the definition space of the real world entity that the data structure represents.

The major difficulty when learning with structured data is that input spaces $\mathcal{X}$ are not endowed with a universally accepted metric. The advantage of using support vector machines, and kernel methods in general, is that by further mapping elements in $\mathcal{X}$ into a feature space using a kernel function alleviates the need of a metric in $\mathcal{X}$.

Kernels for data structures are based more or less on the same idea. The structure is decomposed into substructures (substrings, subtrees, walks etc.) that form a feature vector, which usually has a high dimensionality. Kernels employ these feature vectors afterwards, and ensure tractability by using efficient programming techniques. Under these circumstances, we now can see that embedding real world entities into a Hilbertian space comes at the cost of a gradual simplification and inherent loss of information (Fig. 2.3):

- during *information extraction*, we assume that relevant properties of entities are observed and enclosed conveniently in the corresponding data structure. In practice, this process may be flawed because we do not always discern the link between features and properties, therefore we may miss important information (for example, certain medical properties of a substance may depend of a chemical feature that is being ignored by a simple chemical model of that substance).

- when we compute a *simplified representation* of a data structure, we are usually not able to find computationally useful injections. For example, if we are approximating a graph by its set of walks (paths of adjacent vertices) of length n, different graphs (corresponding to entities with different properties) may generate the same set of walks. This implies that simplified representations may discard useful information when they are not adapted to the learning problem.

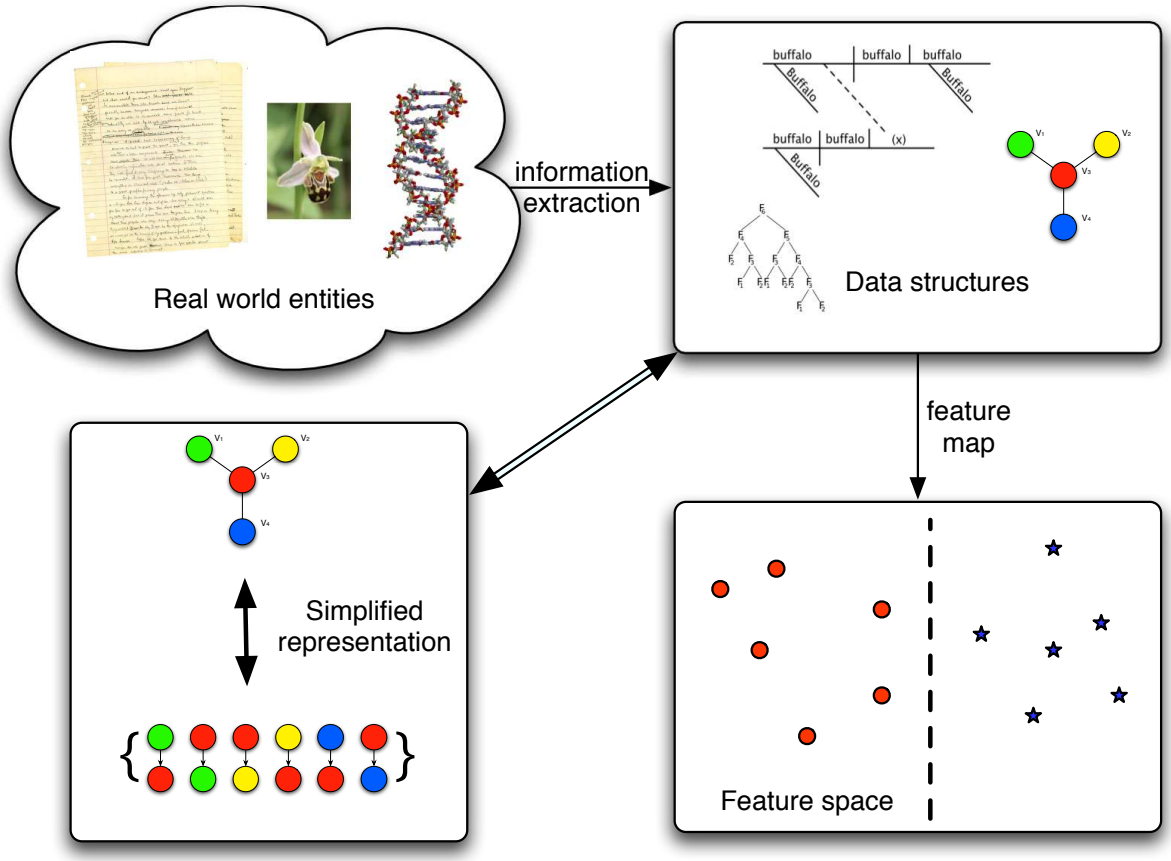


Figure 2.3: A general approach in structural classification. First, information extraction is performed on various entities like genes, protein networks, images, lexical units, molecules etc., and an appropriate data structure is used for each type of input. Then, the data structure is split into simpler parts (a graph is represented by the set of pairs of neighboring vertices). Finally, an appropriate positive definite function is used to project these sets into a feature space where linear classification is performed. Performance is influenced by the quality of information extraction, by the choice of substructures and by the appropriateness of the kernel function with respect to the substructures and to the relations among them.

- *kernel properties* (see Section 2.4) are also important when creating a feature space where distance is linked to perceivable differences between input entities.

In order to build a good classifier, we must ensure that the information bottlenecks mentioned above do not filter out data features that are essential for differentiation. In this chapter, we will take a closer look at the simplified representation and the kernel functions for graphical models.

2.7 Convolution and Marginalized Kernels

Relations in objects Let us assume that $x \in \mathcal{X}$ is a structured object, with its components $p = \{x_1, \ldots, x_D\} \in \mathcal{X}_1 \times \ldots \mathcal{X}_D$. While the components p are still substructures, they are distinctive in the sense that they form together the initial object x. This feature may be formalized using a *relation* R on $\mathcal{X}_1 \times \ldots \times \mathcal{X}_D \times \mathcal{X}$, where $R(x_1, \ldots, x_D, x)$ is true iff $x_1, \ldots, x_D$ are the components of x. We define $R^{-1}(x)$ as $\{x_1, \ldots, x_D : R(x_1, \ldots, x_D, x)\}$. We say that R is finite if $R^{-1}(x)$ is finite for all $x \in \mathcal{X}$.

For example, in the space $\mathcal{S}$ of finite strings over a finite alphabet, we may define $R(x_1, x_2, x)$ iff $x_1 \circ x_2 = x$, where $\circ$ stands for string concatenation. In this example, $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{S}$, and the decomposition $p = \{x_1, x_2\}$ is not unique: $|R^{-1}(x)| = 1 + |x|_{\mathcal{S}}$, where $|\cdot|$ and $|\cdot|_{\mathcal{S}}$ represent set size and string length, respectively.

Typically, designing a kernel for objects x endowed with a relation R resorts to using kernels $K^{(1)}, \ldots, K^{(D)}$ for the elements of $R^{-1}(x)$. While various options are available, convolution kernels have been successfully proposed (Haussler, 1999) to address the problem of assembling part kernels in the following way:

$$K_R(x, x') = \sum_{\substack{\{x_1,\ldots,x_D\} \in R^{-1}(x) \\ \{x'_1,\ldots,x'_D\} \in R^{-1}(x')}} \prod_{d=1}^{D} K^{(d)}(x_d, x'_d) \tag{2.32}$$

When considering a relation R, we can abbreviate this computation by $K_R(x, x') = K^{(1)} \star \ldots \star K^{(D)}$. The following theorem specifies under what circumstances $K_R(x, x')$ is a kernel:

Theorem 2.4 (Convolution kernel (Haussler, 1999))**.** *If $K^{(1)}, \ldots, K^{(D)}$ are kernels on $\mathcal{X}_1 \times \mathcal{X}_1, \ldots, \mathcal{X}_D \times \mathcal{X}_D$, respectively, and R is a finite relation on $\mathcal{X}_1 \times \ldots \times \mathcal{X}_D \times \mathcal{X}$, then $K^{(1)} \star \ldots \star K^{(D)}$ is a kernel on $\mathcal{X} \times \mathcal{X}$.*

Although convolution kernels are general and intuitive, choosing R for a specific application represents the main challenge of this approach. Pioneering examples of such relations may be found in works related to language processing (Collins and Duffy, 2001; Zelenko et al, 2003) and text classification (Lodhi et al, 2002) where strings and trees have been frequently used as data representations.

For graph representations however, it is much more difficult to find a finite relation R which retains enough information from the initial graph. The ideal kernel, related to the feature space generated by subgraphs, has been proven to be intractable (Gaertner et al, 2003), therefore we must restrict this feature space and use simpler representations, based on walks or graph subtrees (Taylor and Cristianini, 2004). Under this type of approach, we must consider the fact that the simplified graph representation may consist in an infinite set of substructures, thus requiring an adapted kernel, that we present below.

The Marginalized Kernel Given a generic class of objects $\mathcal{X}$, we assume that the constituents $x \in \mathcal{X}$ are generated according to a latent variable model which consists

of the visible variable x and of a hidden variable θ, being considered jointly in a pair $z = [\theta, x]$. As we need a kernel $K(x, x')$ for the visible variables, we define first a *joint kernel* $K_z(z, z')$ for the mixed pair, which is used in a *marginalized kernel* (Tsuda et al, 2002) defined as the expectation of the joint kernel over all the values of the hidden variable:

$$K(x, x') = \int_{\theta,\theta' \in \Theta} p(\theta|x)p(\theta'|x')K_z(z, z')d\theta d\theta' \tag{2.33}$$

where Θ refers to the domain of the hidden variable. In a discrete setting, the value of the marginalized kernel is estimated by:

$$K(x, x') = \sum_{\theta,\theta' \in \Theta} p(\theta|x)p(\theta'|x')K_z(z, z') \tag{2.34}$$

The difficulties to be considered when estimating marginalized kernels are the computational burden which is related to the dimension of Θ, and the estimation from the data of the probabilistic model $p(\theta|x)$. Therefore, the choice of the model $p(\theta|x)$ should maximize the relevance for the specific data $x \in \mathcal{X}$ under the tractability constraint of $K(x, x')$. With respect to the properties of the function $K(x, x')$, as long as the joint kernel $K_z(z, z')$ is positive semidefinite, the kernel $K(x, x')$ is also positive semidefinite, since the class of positive semidefinite kernels is closed under addition and multiplication (Genton, 2002). The kernel may also be interpreted as the inner product of the two vectors $p(\theta|x)$ and $p(\theta'|x')$.

2.8 A Direct Application to Graph Learning

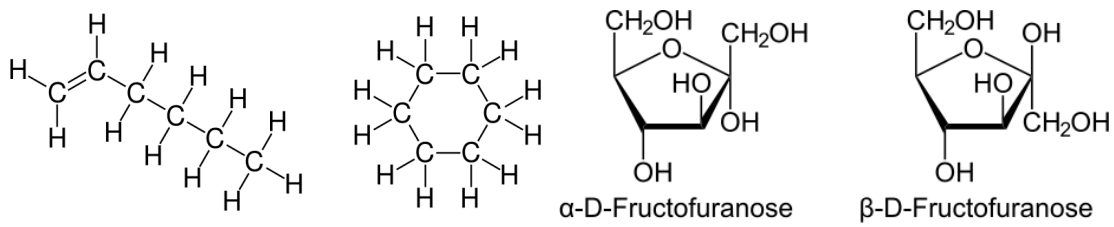
2.8.1 Graphical Representations

We start this section by presenting the formal graphical representation used in statistical learning, with one notable exception that will be important in the next chapters: we consider directed graphs, and cope with undirected ones as particular cases (one indirected edge being considered as two directed edges). In the most general setting, the input data structure is a directed graph G, consisting of:

- a set of labeled vertices $\mathcal{V} = \{x_1, \ldots, x_{|\mathcal{V}|}\}$
- a set of labeled edges $\mathcal{E}$
- a vertex label function $v : \mathcal{V} \to \mathcal{S}_v$
- an edge label function $e : \mathcal{E} \to \mathcal{S}_e$

For the sake of brevity, we write $v(x)$ by v_x and $e(x_1, x_2)$ by $e_{x_1 x_2}$. The sets $\mathcal{S}_v$ and $\mathcal{S}_e$ are two label sets. Their choice depends of the entities that are being represented by the graphical model. For example, in the case of chemical compounds, $\mathcal{S}_v$ is formed of

atom labels, while for $\mathcal{S}_e$ different solutions are available. In a simple setting, we may use a unique label for all edges (which is equivalent to using unlabeled edges), but this may cause issues with certain isomers (Fig. 2.4). Therefore, the choice of these label sets is closely related to the properties of input entities. In a first step, we assume however that label sets are finite.



(a) Structural isomers. (b) Stereoisomerism.

Figure 2.4: Unlabeled edges are adequate for the purpose of modeling structural isomers. For stereoisomers, chemical bond types and spatial information have to be taken into account in edge labels.

A *walk* in a directed graph $G = (\mathcal{V}, \mathcal{E})$ consists in a sequence of graph vertices $h = (x_1, \ldots, x_n)$ of length $|h| = n$, such that $(x_i, x_{i+1}) \in \mathcal{E}$. Graph walks are the substructures that will be used in order to obtain a simplified representation of graphs.

Another useful tool that we employ is the product graph (Imrich and Klavzar, 2000) of two input graphs.

Definition 2.4 (Product graph)**.** *The product graph $G^\times$ of the two input graphs G and G' is determined by the following vertex and edge sets:*

$$\begin{cases} \mathcal{V}^\times = \{(x_1, x_2) \in \mathcal{V} \times \mathcal{V}' \mid v_{x_1} = v_{x_2}\} \\ \mathcal{E}^\times = \{\big((x_1, x_1'), (x_2, x_2')\big) \in \mathcal{V}^{\times 2} \mid \\ \qquad (x_1, x_2) \in \mathcal{E} \wedge (x_1', x_2') \in \mathcal{E}' \wedge e_{x_1 x_2} = e_{x_1' x_2'} \end{cases} \tag{2.35}$$

The fundamental property of the product graph is that an edge in $G^\times$ corresponds to two identically labeled source vertices, edges and destination vertices in G and G', respectively.

2.8.2 A Marginalized Kernel Based On Random Walks

We present now in brief a probabilistic construction (Kashima et al, 2003) that allows the use of marginalized kernels on labeled graphs, when finite label sets $\mathcal{S}_v$ and $\mathcal{S}_e$ have relatively small sizes. The main concept used here is the random walk, i.e. a finite length walk which represents the realisation of a random variable in the space of possible finite walks in a graph. If we refer to the formal definition of a marginalized kernel (Section 2.7), the input graph G represents the visible variable, and the random

walk h represents the hidden variable. In order to estimate a graph kernel, we need to define a probability distribution $p(h|G)$for the hidden variable, along with the joint kernel K_z.

Let us define the probability distribution that generates the hidden variable h. The first element of the walk is a vertex x_1 given by a certain probability distribution p_s over $\mathcal{V}$. This step being completed, we pass into a transition phase. At each subsequent moment, the walk may end at the current vertex x_i with a fixed probability $p_q(x_i)$ or it may continue by advancing to one of the neighboring vertices x_{i+i} with the transition probability $p_t(x_{i+1}|x_i)$.

Each walk $h = (x_1, \ldots, x_n)$ of length $|h| = n$, labeled as $l_h = (v_{x_1}, e_{x_1 x_2}, v_{x_2}, \ldots, v_{x_n})$ has therefore an associated probability that may be expressed as:

$$p(h|G) = p_s(x_1) \prod_{i=2}^{n} p_t(x_i|x_{i-1}) \tag{2.36}$$

in which p_s and p_t have to be chosen in order to build $p(h|G)$ as a probability distribution in the random walk space $\mathcal{V}^* = \cup_{i=1}^{\infty} \mathcal{V}^i$ (the union of all random walk spaces of a certain finite length i). Referring to Section 2.7, we define the kernel for the joint pair $z = [h, G]$ as:

$$K_z(z, z') = \begin{cases} 0 & , \; if \; |h| \neq |h'| \\ K_h(h, h') & , \; if \; |h| = |h'| \end{cases} \tag{2.37}$$

In the equation above, $K_h(h, h')$ represents the kernel function between two hidden variables h and h' which are independently generated in the two graphs to be compared, G and G'. The constraint that $|h| = |h'|$ is necessary for the tractability of the problem.

Accordingly, the kernel (2.34) between two graphs G and G' measures the similarity of all the possible random walks, weighted by their probabilities of apparition. As for the kernel $K_h(h, h')$, we can employ a convolution kernel (Equation 2.32) by considering the simple following relation between a walk h and its constituent parts:

$$R^{-1}(h) = \{x_1, (x_1, x_2), x_2, \ldots, (x_{n-1}, x_n), x_n\}, \quad R^{-1}(h) \in \mathcal{V} \times (\mathcal{E} \times \mathcal{V})^{n-1} \tag{2.38}$$

Using two kernel functions $K_v : \mathcal{S}_v \times \mathcal{S}_v \to \mathbb{R}$ and $K_e : \mathcal{S}_e \times \mathcal{S}_e \to \mathbb{R}$ for vertex and edge labels, respectively, we can explicit the expression of the convolution kernel for random walks:

$$K_h(h, h') = K_v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} K_e(e_{x_{i-1} x_i}, e_{x'_{i-1} x'_i}) K_v(v_{x_i}, v_{x'_i}). \tag{2.39}$$

In order to have an efficient computation for this type of graph kernel, we follow a general family of probability functions $p(h|G)$ that enable us to express $K(G, G')$ as a matrix power series (Gaertner et al, 2003). We fix the following stochastic constraints for $p(h|G)$ (Equation 2.36):

$$\begin{cases} p_s(v) = p_0(v)p_q(v) \\ \\ p_t(u|v) = \frac{1-p_q(v)}{p_q(v)} p_a(u|v)p_q(u) \end{cases} \tag{2.40}$$

with $0 < p_q(x) < 1$, $\sum_{x \in \mathcal{V}} p_0(x) = 1$ and an advancing probability $\sum_{x' \in \mathcal{V}} p_a(x'|x) = 1$ distributed on the outgoing edges of a vertex x: $p_a(x'|x) > 0 \Leftrightarrow (x, x') \in \mathcal{E}$.

For most graphical models, a uniform start probability as well as a uniform advancing probabilities for each vertex are appropriate, unless specific prior knowledge can enhance the fitness of these distributions.Usually, there is no reason to prefer a certain vertex as start point, or to prefer certain transitions to the detriment of others, therefore we use the following:

$$p_0(x) = \frac{1}{|\mathcal{V}|}, \quad p_a(\cdot|x) = \frac{1}{|\{x'|(x, x') \in \mathcal{E}\}|} \tag{2.41}$$

In terms of p_0, p_a, p_q, the expression of $p(h|G)$ (Equation 2.36) can be rewritten as:

$$\begin{cases} p(h|G) = p_0(x_1) \left[\prod_{i=2}^{n} \left(1 - p_q(x_{i-1})\right) p_a(x_i|x_{i-1}) \right] p_q(x_n), \quad n \geq 2 \\ \\ p(h|G) = p_0(x_1)p_q(x_1), \quad n = 1 \end{cases} \tag{2.42}$$

Although this form is more explicit, the form in Equation 2.36 is preferred for computational reasons. A direct consequence of this choice for p_0, p_a, p_q is that $p(h|G)$ will define a valid probability distribution on $\mathcal{V}^*$: $\sum_{h \in \mathcal{V}^*} p(h|G) = 1$.

As mentioned above, we assume that the label sets $\mathcal{S}_v, \mathcal{S}_e$ have a small number of elements. This situation arises for example in biological sequence (Tsuda et al, 2002) or chemical compound classification problems (Mahé et al, 2004), where this kernel model originated. Therefore, an appropriate kernel for assessing vertex or edge label similarity is the Kronecker delta function:

$$K_\delta(z, t) = \begin{cases} 1 \ , \ if \ z = t \\ \\ 0 \ , \ if \ z \neq t \end{cases} \tag{2.43}$$

Based on the constraints 2.40, computing the marginalized kernel $K(G, G')$ is tractable because its value can be rewritten as an infinite matrix power series. In order to explicit $K(G, G')$, we have to consider the product graph $G^\times$ of the two input graphs G and G'. We rely upon two variables, the first one $\Pi_s = ((\pi_s(x, x'))_{(x,x') \in \mathcal{V} \times \mathcal{V}'}$ is a $|\mathcal{V}||\mathcal{V}'|$ vector containing the joint start probabilities of two vertices $x \in \mathcal{V}$ and $x' \in \mathcal{V}'$ if they have the same label, and 0 otherwise. The second variable needed for the kernel computation $\Pi_t = ((\pi_t((x_1, x_1')|(x_2, x_2'))))_{(x_1,x_1'),(x_2,x_2') \in \mathcal{V} \times \mathcal{V}'}$ is a $(|\mathcal{V}||\mathcal{V}'|)^2$ square matrix whose elements assess the joint transition probability between two pairs of vertices belonging to the first and to the second graph, if and only if these vertex pairs and the corresponding edge

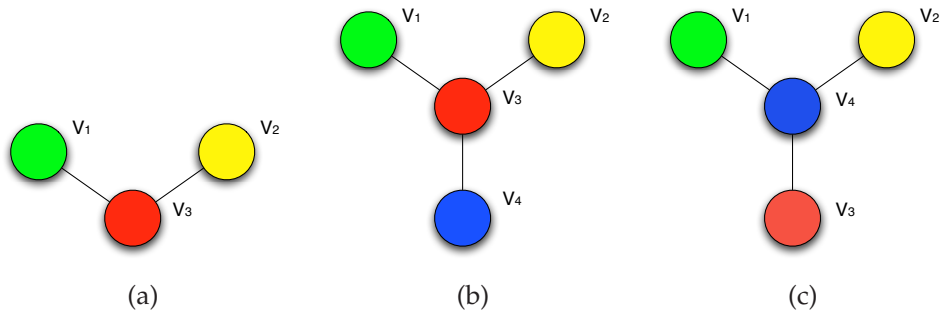


Figure 2.5: A basic example of labeled graph inference: each vertex has an associated feature label. In this example, the first two graphs are intuitively more alike, but the third one is closer to the second than to the first one. However, with a different relation being considered between vertices (which would imply a different localization of the edges), the second graph could become more similar to the third one.

pair are identically labeled (otherwise the probability is null):

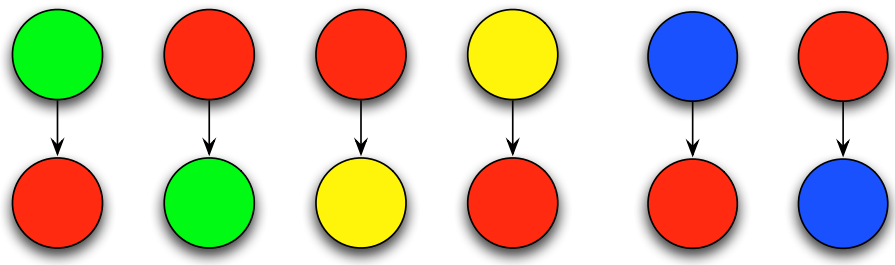
$$\begin{cases} \pi_s(x, x') = p_s^G(x) p_s^{G'}(x') \\ \pi_t((x_1, x_1')|(x_2, x_2')) = p_t^G(x_2|x_1) p_t^{G'}(x_2'|x_1') \end{cases} \tag{2.44}$$

In $G^\times$, Π_s and Π_t represent exactly the array of vertex start probabilities and the full matrix of transition probabilities. The building strategy of $G^\times$ implies there is a bijection between *identically labeled* walks in the initial graphs G and G', and *possible* walks in the product graph $G^\times$. Furthermore, the probability distributions in Equation 2.40 correspond in $G^\times$ exactly to the functions presented in Equation 2.44. In this way, multiplying probabilities associated only to identical walks in G and G' in order to estimate Equation 2.34 boils down to a much simpler task, that of summing all the probabilities associated to walks in $G^\times$.

The stochastic process of random walk generation in $G^\times$ which is controlled by Π_s and Π_t, is a Markov chain, in agreement with the independence between past transitions that end in a vertex x and the outcome of future transitions starting from x. The probability to get a random walk ($x_1 \xrightarrow{n} x_2$) of length (measured in number of edges) n starting in vertex $x_1 \in G^\times$ and ending in vertex $x_2 \in G^\times$ is:

$$p(x_1 \xrightarrow{n} x_2|G^\times) = \pi_s(x_1) \times \Pi_t^n(x_1, x_2) \tag{2.45}$$

where $\Pi_t^n(x_1, x_2)$ represents the value in the matrix Π_t^n which is found at the intersection between the row corresponding to the source x_1, and the column corresponding to the destination x_2.



(a) The first and the second graph have four common random walks. There are no common random walks of length two (or more) between the first graph and the third one.

(b) The second and the third graphs have only two common random walks.

Figure 2.6: Common random walks of length two between pairs of graphs depicted in Figure 4.5. Similarity between graphs is computed by observing similar random walks; more common random walks account for greater similarity.

The graph kernel $K(G, G')$ can be rewritten in terms of this quantity:

$$\begin{aligned} K(G, G') &= \sum_{n=0}^{\infty} \sum_{x_1, x_2 \in G^{\times}} p(x_1 \xrightarrow{n} x_2 | G^{\times}) \\ &= \sum_{n=0}^{\infty} \Pi_s^T \Pi_t^n \mathbb{1} = \Pi_s^T (I + \Pi_t + \Pi_t^2 + \ldots) \mathbb{1} \\ &= \Pi_s^T (I - \Pi_t)^{-1} \mathbb{1} \end{aligned} \tag{2.46}$$

where $\mathbb{1}$ represents the vector with all its elements equal to 1.

Table 2.1: Marginalized kernel values between the graphs depicted in Figure 4.5 ($p_q(v) = 0.4$). A higher value accounts for a higher graph similarity.

	Graph a)	Graph b)	Graph c)
Graph 4.5a	0.0741374	0.0535117	0.0400000
Graph 4.5b	0.0535117	0.0543478	0.0427273
Graph 4.5c	0.0400000	0.0427273	0.0543478

Due to the inversion of $(I - \Pi_t)$ which dominates the computation cost, the problem has an order of complexity of $O((|\mathcal{V}||\mathcal{V}'|)^3)$. There are numerical approximation methods (Barrett et al, 1994; Borgwardt et al, 2005) that are able to improve the computational speed of the algorithm, but many of these improvements are conditioned by a low cardinality of $\mathcal{S}_v$and $\mathcal{S}_e$, and by a low degree of vertex connectivity. We will discuss these methods in a subsequent chapter.

As an illustration of this approach, we present in Figure 2.6 the possible random walks, of length two only, between the three example graphs depicted in Figure 4.5. Equation (2.34) is equivalent to detecting similar random walks, for all lengths, and adding up the product of their associated probabilities. The actual marginalized kernel values between these graphs are presented in Table 2.1. Values are in agreement with the perception that the first two graphs are quite similar, and that the third one is closer to the second than to the first one.

Chapter 3

Towards More Expressiveness in Graph Labeling

3.1 Kernel improvement on limited label set graphs

The need for a labeled graph kernel arose initially from performing specific prediction tasks in domains like biology and chemistry. The objects to be analyzed, be it amino acid sequences or chemical compounds, are endowed with a well defined structure and, most importantly, their label sets are rather reduced in size. Therefore, the choice of a strongly discriminative kernel like the one depicted in Section 2.43 and the simple use of graph adjacency for modeling spatial adjacency were entirely justified and resulted in state of the art performances (Tsuda et al, 2002; Kashima et al, 2003). Due to the kernel simplicity, both conceptually and in terms of computational challenges, further work has been done in order to make the marginalized graph kernel suitable to more complex classification tasks.

This extension work revolved around finding efficient methods to incorporate richer descriptive labeling of the constituent parts that form the object to classify, along with including contextual information in this labeling.

One strategy has been to label similar components differently, based on the local structure of the graph. This translates into incorporating relational information in vertex labeling, at the cost of increasing the size of the label set $\mathcal{S}_v$. As paths have been replaced with paths labeled with their environment, this has increased the relevance of the features; the potential downfall of this approach is that the kernel might discriminate too strongly based on the environment and, as a result, the similarity rates might drop too drastically for similar contexts. The reliability of the kernel has been successfully proved (Mahé et al, 2004) in a setting where the variability of the environment is intrinsically limited. An incremental approach has been used to study the impact on the kernel relevance of the size of the local environment that has been taken into account.

The same study has explored another strategy to adapt the kernel, that of denying the random walks the possibility to return, i.e. imposing $x_i \neq x_{i-2}$ for some i. This

behavior, denoted tottering, does not bring new structural information concerning the object to classify. The only walks being considered by this version of the algorithm have the following property:

$$h = (x_1, \ldots, x_n), \quad x_i \neq x_{i+2}, \forall i = 1, \ldots, n-2 \tag{3.1}$$

This approach belongs to a wider class that alters the probability laws in Equation 2.37 according to a particular type of object or classification task. The price to pay in this case for improving structural relevance is the creation of a larger model, if we want to preserve the Markov property of our walk generation process. More precisely, the transition probabilities in Equation 2.40 change to:

$$p_t(u|v,w) = \frac{1 - p_q(v)}{p_q(v)} p_a(u|v,w) p_q(u) \tag{3.2}$$

for cases when w, the vertex anterior to v, exists. This more complex model allows for an explicit definition of $p_a(u|v,w) = 0$ if $u = w$, which makes returning on an edge impossible. The drawback of taking into account the two previous states is obviously an increase in computational complexity. We can bring this problem down to the regular kernel computation by transforming a graph G into a form G' in the following way:

- each vertex v or edge e in G becomes a vertex in G'; the label of e in G' becomes the label of its destination in G:
$$\mathcal{V}' = \mathcal{V} \cup \mathcal{E}$$
- an edge is placed in G' between a vertex v in G and an edge e in G originating in v,
- an edge is placed in G' between two edges in G that do not create a loop:
$$\mathcal{E}' = \{(v,(v,t)) \,|v \in \mathcal{V}, (v,t) \in \mathcal{E}\} \cup \{((u,v),(v,t)) \,|(u,v),(v,t) \in \mathcal{E}, u \neq t\}$$

Given the way G' is generated from G, there is a bijection between non-tottering random walks in G and all possible random walks in G'. This implies that computing the necessary second order random walk kernels on the initial graphs G is equivalent to computing regular random walk kernels on their transformed graphs G', which accounts for an increase in complexity from $O((|\mathcal{V}_1||\mathcal{V}_2|)^3)$ to:

$$O((|\mathcal{V}'_1||\mathcal{V}'_2|)^3) = O((|\mathcal{V}_1 + \mathcal{E}_1||\mathcal{V}_2 + \mathcal{E}_2|)^3) \tag{3.3}$$

These improvements, however, are successful in settings that still allow the sets $\mathcal{S}_v$ and $\mathcal{S}_e$ to have a small size (or to have a small subset of frequently used labels) and, at the same time, to be able to grasp, using this limited labeling, the most relevant features of a complex object that is being modeled by the graph.

When it comes to graph representations of images, the high variability of region features requires a rich set $\mathcal{S}_v$ of descriptors of their geometric and photometric properties.

Furthermore, we can equally expect the set $\mathcal{S}_e$ to cope with complex relations or constraints between regions, which also increases connectivity within the graph. A higher connectivity decreases the benefits of preventing graph tottering and does not justify the computational burden, since the phenomenon is much less frequent. A similar situation arises in chemical compound property prediction tasks, when the distance between two adjacent structures may influence the overall behavior of the molecule; therefore, the adjacency property must no longer be a binary function, but one that gives an actual physical distance (Mahé et al, 2006).

3.2 Continuous Labels and More Complex Kernels for Vertex and Edge Features

The original graph kernel $K(G, G')$ defined in Equation (2.46) is estimated by summing the similarities of all pairs of random walks of equal length. For a certain pair of such random walks (h, h'), let us suppose that we get simultaneously to a pair of corresponding vertices (x_i, x'_i). At this point we analyze the next transition in each walk. If the labels of the next two edges $(e_{x_i x_{i+1}}, e_{x'_i x'_{i+1}})$ and the labels of the next two vertices $(v_{x_{i+1}}, v_{x'_{i+1}})$ are not identical, the similarity brought by these walks will be null and we start analyzing another pair of walks. Otherwise we multiply the current similarity of the walks by the probabilities for the two transitions occurring in each walk. This leaves us with a probability of getting these random walks from start to end of:

$$p(h, h') = \left(p_s^G(v_1) \prod_{i=2}^{n} p_t^G(v_i | v_{i-1}) \right) \cdot \left(p_s^{G'}(v'_1) \prod_{i=2}^{n} p_t^{G'}(v'_i | v'_{i-1}) \right) \tag{3.4}$$

in which we suppose implicitely that for the walks (h, h'), the labels of all the corresponding constituent parts are identical. Using a Kronecker delta function for vertex and edge similarity, it is obvious that random walk kernels in Equation 2.39 will be also delta functions, so the graph kernel in Equation 2.34 is reduced to the direct sum of all the probabilities $p(h, h')$ above (Equation 3.4) computed for identically labeled random walks.

This strategy works for discrete ranged kernel functions, but in the case of region features like gray level or surface, we need a less discriminative kernel. There are many possible solutions to this problem, depending of the type of numerical representation of the numerical features.

The way similarity is computed between two random walks in Equation 2.39 using a highly discriminative kernel (Equation 2.43) is not adapted to the case when the ranges $\mathcal{S}_v$, $\mathcal{S}_e$ are subspaces of a continuous space like $\mathbb{R}^k, k \geq 1$. A sequence of similar (but not necessarily identical) vertex and edge labels to be compared should get a similarity value $K_h(h, h')$ that is neither 0 nor 1, but some value in between. Moreover, a variation in any of the factors $K_v(v_{x_i}, v_{x'_i})$ or $K_e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i})$ that form $K_h(h, h')$ (Equation 2.39) should entail a matching (in terms of sign) variation in $K_h(h, h')$. This is a consequence of the

importance each pair of constituents $(v_{x_i}, v_{x'_i})$ or $(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i})$ has in the overall similarity of the two paths h, h'.

Concerning positive definiteness, let us recall why it is possible to use any positive definite kernel in the place of the delta function. Functions K_v and K_e integrate into the walk kernel K_h, which is a convolution kernel, as mentioned in Section 2.8.2. Therefore, the closure property of convolution kernels (Theorem 2.4) guarantees the positive definiteness of K_h and consequently of the graph kernel, if K_v and K_e are also positive definite.

The next step is to integrate these values to the graph kernel computation. If we employ in the graph kernel definition the joint probability from Equation 3.4 and we replace the delta function $K(h, h')$ with Equation 2.39, we obtain:

$$K(G, G') = \sum_h \sum_{h'} \Bigg[K_v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} \Big(K_e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) \cdot K_v(v_{x_i}, v_{x'_i}) \Big) \\ \times p_s^G(x_1) \cdot p_s^{G'}(x'_1) \cdot \prod_{i=2}^{n} \Big(p_t^G(x_i|x_{i-1}) p_t^{G'}(x'_i|x'_{i-1}) \Big) \Bigg] \tag{3.5}$$

By comparing this revised form with the previous kernel expression that was determined in Equation 2.46 using the properties of the product graph , we can determine what has to be done in order to take into account vertex and edge kernels. More precisely, when we label the product graph, we have to take into account not only the probability distribution, which is related to the graph structure, but also the initial graph labels and include them *explicitly* into the new labels. Incorporating these additional values to the model requires an adjustment of the expressions of Π_s and Π_t in Equation 2.44: in order to build the product graph, we will not select only the vertices having the same label, because this would not be helpful under the continuity assumption. Rather than doing this, we will penalize each vertex-vertex combination by a factor representing their similarity function. The same strategy applies to the edges of the graph product:

$$\begin{cases} \pi_s(x, x') = p_s^G(x) p_s^{G'}(x') \times K_v(v_x, v_{x'_i}) \\ \pi_t((x_1, x'_1)|(x_2, x'_2)) = p_t^G(x_2|x_1) p_t^{G'}(x'_2|x'_1) \times K_e(e_{x_1x_2}, e_{x'_1x'_2}) \times K_v(v_{x_2}, v_{x'_2}) \end{cases} \tag{3.6}$$

The vertex and edge kernel functions appear in this model as probability multipliers along transitions, which penalize paths with respect to their constituent dissimilarities. Using the revised variables Π_s and Π_t from Equation 3.6, we can now employ the formula 2.46 to compute the revised graph kernel from Equation 3.5.

The initial case depicted in Equation 2.44 appears now clearly as a particular case of Equation 3.6 when using a highly discriminative kernel function (Equation 2.43) for $K_v(v_{x_i}, v_{x'_i})$ and $K_e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i})$.

3.3 Benefits and alternative solutions in graph kernel theory

In this general case of directed labeled graphs, the method presented above has some clear advantages. Equation 3.6 makes it extremely clear to understand how structural and label-related similarities are connected. Even though different graph kernels exist and are especially useful for structural similarity, the random walk kernel is preferred for labeled graphs and this method does not add to the initial complexity factor of the random walk kernel for discrete labels.

The additional computational burden comes from estimating more complex vertex and edge kernel functions, and from the fact that the frequency of null similarity values between features decreases, which in turn rules out the benefits of sparsity-related numerical improvements.

In Kashima et al (2003), an approximation of the graph kernel value is computed, by iterating on a formula which converges to the exact value $K(G, G')$. Using dynamic programming, the contribution of random walks of increasing lengths is being considered at each step. The authors need in practice approximately 20-30 iterations on Mutag (Srinivasan et al, 1996) and PTC (Helma et al, 2001) pharmaceutical compound datasets in order to attain convergence. This is related however to the sparsity of graph adjacency matrices, which is uncommon when other types of data, as images, are being represented.

3.4 Graph matching vs. graph kernels using random walks

Random walks and, more recently, quantum walks have already been successfully used in pattern recognition for graph matching (Gori et al, 2005; Neuhaus and Bunke, 2005; Emms et al, 2007). Even though graph matching presents a wide range of applicability in image interpretation, we will try to underline the potential that a random walk graph kernel has in certain cases.

Designing algorithms for exact and approximate graph matching is still a challenge, as this is closely related to graph isomorphism. Random walk based strategies perform very well on structural graphs (for instance, the result of Delaunay triangulation of Harris corners in an image) either because they succeed in extracting a specific graph signature, or because they succeed in efficiently pruning the search space of isomorphic graphs. Label information is discarded, or dealt with in a particular adapted extension, as its presence complicates the matching task.

In the case of graph kernels, label information is crucial for estimating graph similarity. In Fig. 3.1 we present a simple case of two graphs with a different structure; if the labeling is absent along the vertices and edges, the graph kernel cannot discriminate using only the graph structure alone. The proof of this is presented below.

Lemma 3.1. *In the absence of vertex and edge labels, given a random walk kernel K and for any*

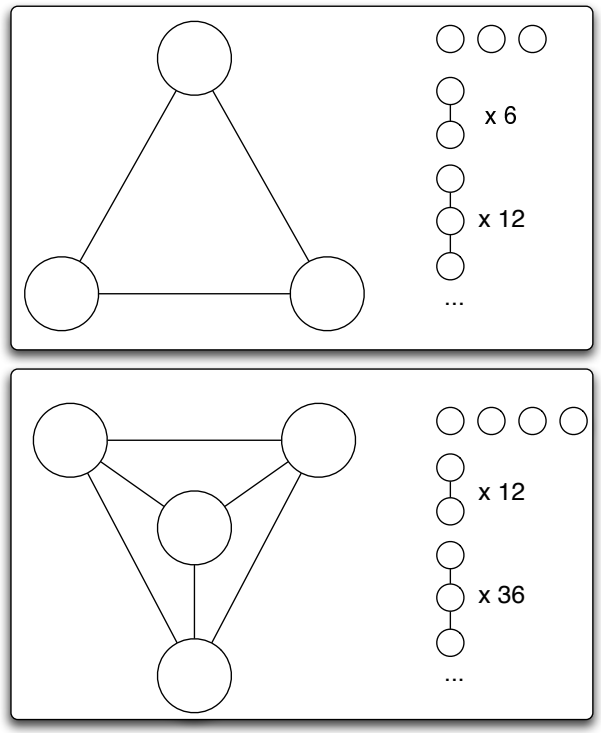


Figure 3.1: Two structurally different, unlabeled graphs, along with the first elements of their infinite random walk sets. The random walk kernel cannot differentiate them: even if the random walk sets generated by two unlabeled graphs are different, the similar walks weighted by their probabilities will add up to the same amount.

undirected graphs G and G', we have $K(G,G) = K(G,G')$.

Proof. Let $G = (\mathcal{V}, \mathcal{E})$ and $G' = (\mathcal{V}', \mathcal{E}')$ be two undirected unlabeled graphs. In this case, the random walk kernel for these graphs is:

$$K(G,G') = \sum_{n=1}^{\infty} \sum_{\substack{h \in \mathcal{V}^* \\ |h|=n}} \sum_{\substack{h' \in \mathcal{V}'^* \\ |h'|=n}} p(h)p(h'), \tag{3.7}$$

since $K(h,h') = 1$ for any h, h' with $|h| = |h'|$. As we mentioned in Section 2.8.2, a distribution $p(h)$ is built in the common case on an uniform vertex quit probability $0 < \mathrm{p_q} < 1$, a start probability p_s which is evenly distributed to vertices $p_s(v) = p_q/|\mathcal{V}|$ and a transition probability p_t for the outgoing edges :

$$\begin{aligned} p(h|G) &= p_s(v_1) \prod_{i=2}^{n} p_t(v_i|v_{i-1}) \\ &= \frac{p_q(v_1)}{|\mathcal{V}|} \prod_{i=2}^{n} \frac{1 - p_q(v_{i-1})}{p_q(v_{i-1})} p_a(v_i|v_{i-1}) p_q(v_i) \end{aligned} \tag{3.8}$$

As the vertex quit probabilities $p_q^G(v_i), p_q^{G'}(v_i^{'})$ are uniform and equal to the constant $\mathrm{p_q}$, the expression of $p(h)$ simplifies to:

$$p(h) = \frac{\mathrm{p_q}(1-\mathrm{p_q})^{n-1}}{|\mathcal{V}|} \prod_{i=2}^{n} p_a(v_i|v_{i-1}) \tag{3.9}$$

The random walk kernel can be written as:

$$\begin{aligned} K(G,G') &= \sum_{n=1}^{\infty} \sum_{\substack{h\in\mathcal{V}^* \\ |h|=n}} \sum_{\substack{h'\in\mathcal{V}'^* \\ |h'|=n}} \frac{\mathrm{p_q}^2(1-\mathrm{p_q})^{2n-2}}{|\mathcal{V}||\mathcal{V}'|} \times \prod_{i=2}^{n} p_a^G(v_i|v_{i-1}) p_a^{G'}(v_i^{'}|v_{i-1}^{'}) \\ &= \sum_{n=1}^{\infty} \frac{\mathrm{p_q}^2(1-\mathrm{p_q})^{2n-2}}{|\mathcal{V}||\mathcal{V}'|} \times \sum_{\substack{h\in\mathcal{V}^* \\ |h|=n}} \sum_{\substack{h'\in\mathcal{V}'^* \\ |h'|=n}} \prod_{i=2}^{n} p_a^G(v_i|v_{i-1}) p_a^{G'}(v_i^{'}|v_{i-1}^{'}) \end{aligned} \tag{3.10}$$

The last sum of products of this equation can be computed:

$$\begin{aligned} &\sum_{\substack{h\in\mathcal{V}^* \\ |h|=n}} \sum_{\substack{h'\in\mathcal{V}'^* \\ |h'|=n}} \prod_{i=2}^{n} p_a^G(v_i|v_{i-1}) p_a^{G'}(v_i^{'}|(v_{i-1}^{'}) = \\ &= \left[\sum_{\substack{h\in\mathcal{V}^* \\ |h|=n}} \prod_{i=2}^{n} p_a^G(v_i|v_{i-1}) \right] \left[\sum_{\substack{h'\in\mathcal{V}'^* \\ |h'|=n}} \prod_{i=2}^{n} p_a^{G'}(v_i^{'}|(v_{i-1}^{'}) \right] \\ &= |\mathcal{V}||\mathcal{V}'| \end{aligned}$$

as a direct consequence of the property $\sum_{u\in\mathcal{V}} p_a(u|v) = 1$, for any $v \in \mathcal{V}$. The expression of $K(G,G')$ becomes then:

$$\begin{aligned} K(G,G') &= \sum_{n=1}^{\infty} \mathrm{p_q}^2(1-\mathrm{p_q})^{2n-2} = \frac{\mathrm{p_q}^2}{(1-\mathrm{p_q})^2} \sum_{n=1}^{\infty} \left[(1-\mathrm{p_q})^2\right]^n \\ &= \frac{\mathrm{p_q}}{2-\mathrm{p_q}} \end{aligned} \tag{3.11}$$

This result is independent of the degree, connectivity patterns or any other structural features of graphs G and G'. □

Beyond the proof, the result is quite intuitive if we consider the reason of using random walks in order to describe graphical structures. If we start generating a random walk, at each step we have a stop probability, and a global probability to pass to an unlabeled neighbor, which in this case is structure invariant since we do not discriminate among neighbors.

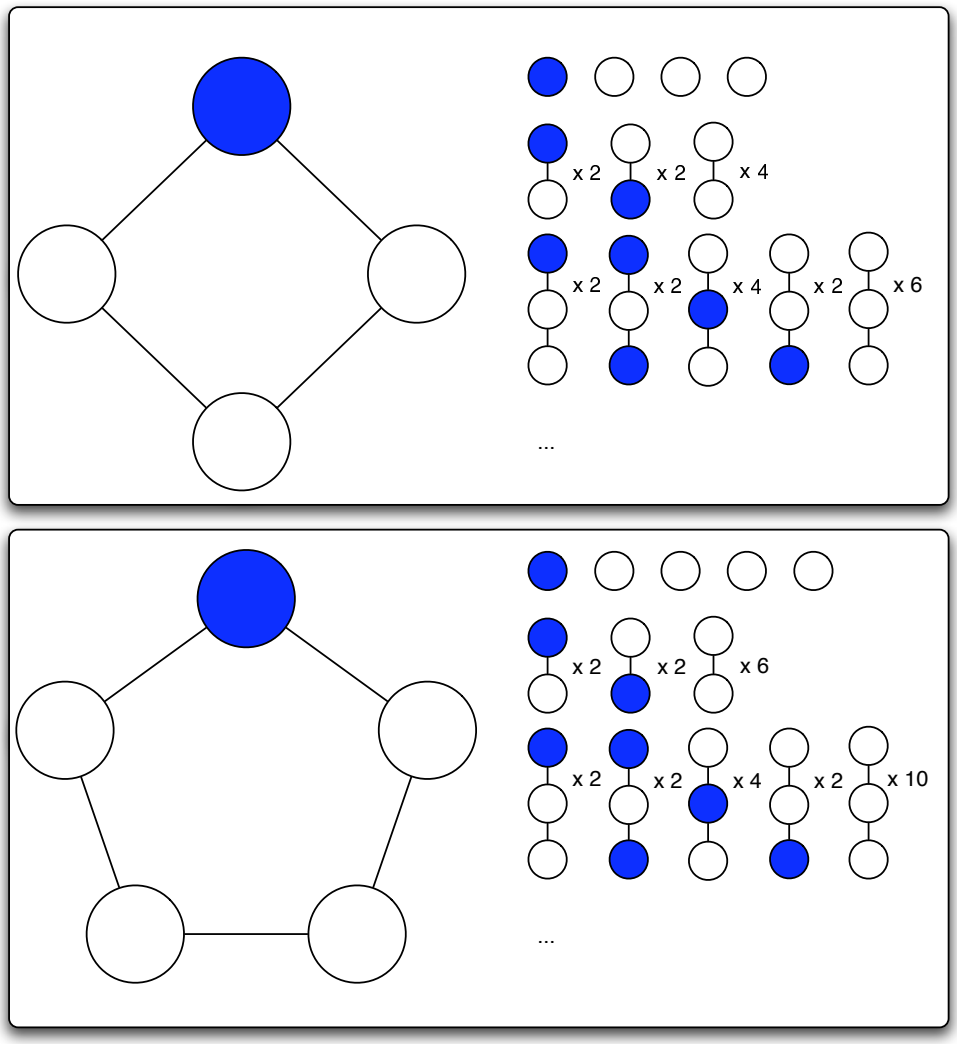


Figure 3.2: The elements C_4 and C_5 in the class C of closed chain graphs, along with the first elements of their infinite random walk sets. Their vertices are uniformly labeled, except for one, which has the same distinct value across all the family graphs. For higher degrees, perceiving the difference between elements using random walks becomes more difficult.

An obvious consequence of this result is that marginalized graph kernels are unsuited when coping with unlabeled graphs. However, if labels are present, we witness a considerable improvement in performance, even when discriminating within a graph class with very fine differences among its elements. The reason is that if a vertex is different with respect to its neighborhood, it can be used as a landmark in order to describe this neighborhood.

3.4.1 Discrete kernels and discrete values

Let us consider the class C of closed chain graphs, with all vertices being identically labeled except one. By $C_n \in C$ we denote the closed chain with exactly n vertices (Fig. 3.2). In the feature space generated by the graph kernel function, we compute the distance between C_{25} and each of the chain graphs $C_3 - C_{40}$ (the value of the distance in the feature space has been explicited in Equation 2.30). The results are presented in Fig. 3.3 for three different values of the random walk parameter p_q. They show that the kernel exploits the slight differences between the random walks on the closed chains, and succeeds in creating a metric (Aronszajn, 1950) which is in agreement with our intuition of an order in C.

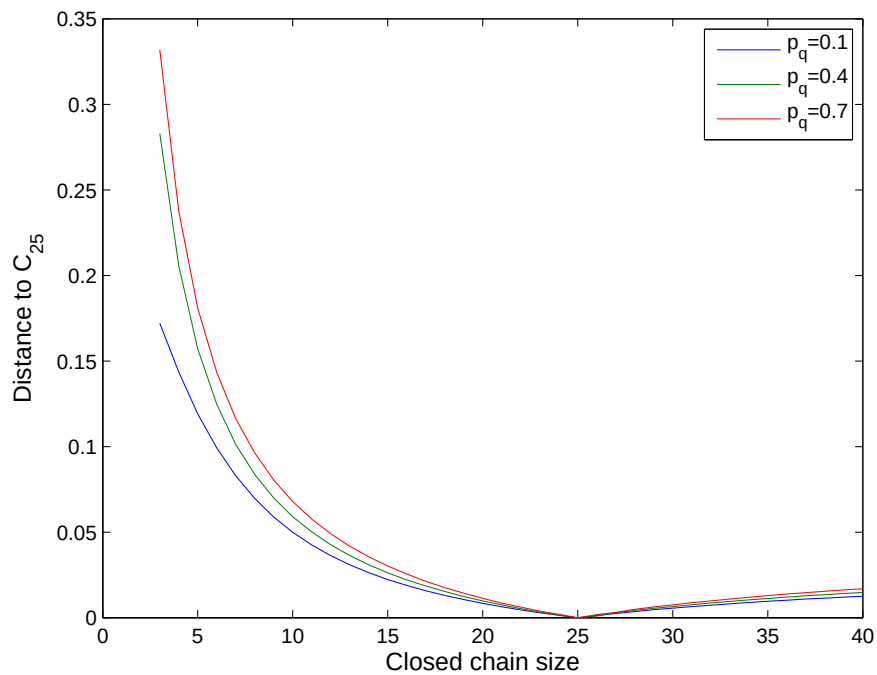


Figure 3.3: Distances in the graph feature space between C_{25} and other closed chains. The distances are computed using a Kronecker delta kernel for vertex labels, for three different values of the random walk parameter p_q.

In real-life scenarios, we expect that labels have numerical values which are distributed depending on the model that we study and on various observation errors. We analyze the progressive transition between discrete and continuous labels in the next section.

3.4.2 Continuous kernels and discrete values

Let us consider that all vertices are labeled using a discrete value μ_1, except the single vertex which is labeled using μ_2. This time we compute the same distances with respect to C_{25} using a Gaussian vertex kernel $K_v(x, y) = \exp\{-(x - y)^2/2\sigma^2\}$.

In the case of discrete labels, the use of a continuous kernel does not fundamentally affect the graph metric for the following reasons. $K_v(x, y) = 1$ for $x = y$, which is similar to the previous case, when we used the Kronecker delta function. However, if the kernel parameter σ is relatively high with respect to the distance $|\mu_1 - \mu_2|$, the value of the vertex kernel between different labels increases from 0 to $\exp\{-(\mu_1 - \mu_2)^2/2\sigma^2\}$. As a result, the values of the graph kernels increase, and a result of this increase in similarity is that distances between feature points decrease. In Figure 3.4 we present the distances to C_{25} for $\mu_1 = 0, \mu_2 = 1$ with the initial discrete kernel, and with two Gaussian kernels. For $\sigma = 0.1$, the distances are identical to the ones computed using the discrete kernel.For a larger $\sigma = 0.5$, distances decrease while preserving the ordering. Obviously, using a continuous kernel for discrete labels is unnecessary in most cases, but we can conclude that for a reasonable choice of kernel parameters, impact on the metric in the feature space is limited.

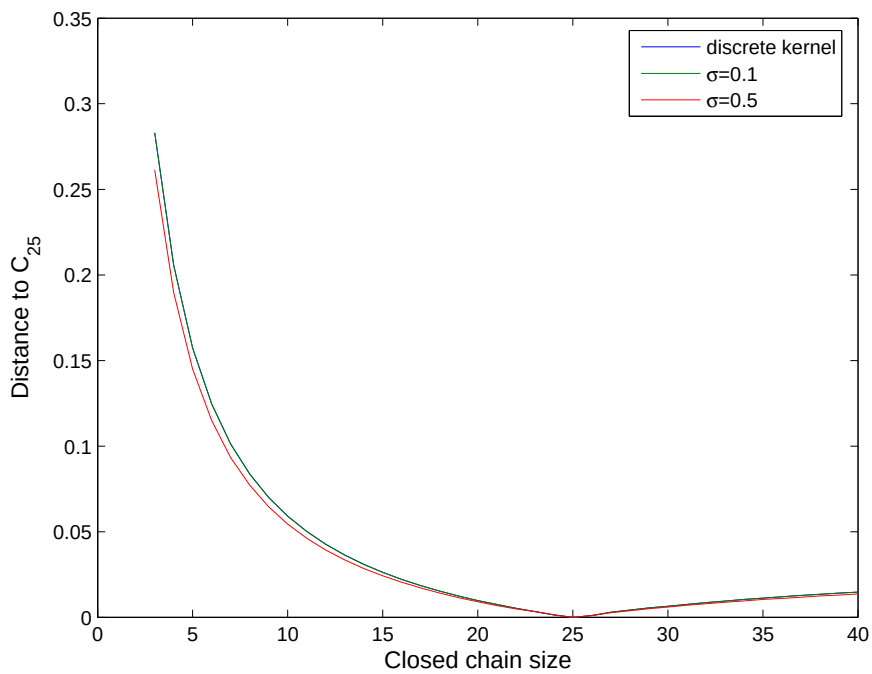


Figure 3.4: Distances in the graph feature space between C_{25} and other closed chains with $\mu_1 = 0, \mu_2 = 1, p_q = 0.4$ and Gaussian kernels for vertices. For $\sigma = 0.1$, the distances are identical to the ones on the reference line, and for a larger $\sigma = 0.5$, values decrease while preserving the distance ordering between different graphs and C_{25}.

3.4.3 Continuous kernels and continuous values

In a following step, we consider that the elements in C are uniformly labeled with a value generated by a normal distribution $\mathcal{N}_1 = \mathcal{N}(\mu_1, \sigma_1)$, except for the one different label, which is generated according to $\mathcal{N}_2 = \mathcal{N}(\mu_2, \sigma_2)$. Then, we use a Gaussian vertex kernel characterized by its parameter σ.

Even for this artificial example, the choice of σ is not trivial. In kernel design, there is no absolute rule for choosing the right kernel, but kernels should capture a meaningful similarity measure. For our particular distributions, the Gaussian kernel $K_v(x, y)$ should have the highest possible value if the input pair (x, y) is generated according to $(\mathcal{N}_1, \mathcal{N}_1)$ or $(\mathcal{N}_2, \mathcal{N}_2)$, and the lowest possible value if (x, y) is generated according to $(\mathcal{N}_1, \mathcal{N}_2)$ or $(\mathcal{N}_2, \mathcal{N}_1)$. In order to express this mathematically we can estimate the kernel values for the random variables used as labels via their expected values:

$$
\begin{aligned}
\mathbb{E}_1 &= \mathbb{E}\left(K_v(x, y)\right) \bigwedge \{P(x) = \mathcal{N}_1(x), P(y) = \mathcal{N}_2(y)\} = \frac{\sigma}{\sqrt{\sigma^2 + \sigma_1^2 + \sigma_2^2}} \exp\left(\frac{-(\mu_1 - \mu_2)^2}{\sigma^2 + \sigma_1^2 + \sigma_2^2}\right) \\
\mathbb{E}_2 &= \mathbb{E}\left(K_v(x, y)\right) \bigwedge \{P(x) = \mathcal{N}_2(x), P(y) = \mathcal{N}_1(y)\} = \mathbb{E}_1 \\
\mathbb{E}_3 &= \mathbb{E}\left(K_v(x, y)\right) \bigwedge \{P(x) = \mathcal{N}_1(x), P(y) = \mathcal{N}_1(y)\} = \frac{\sigma}{\sqrt{\sigma^2 + 2\sigma_1^2}} \\
\mathbb{E}_4 &= \mathbb{E}\left(K_v(x, y)\right) \bigwedge \{P(x) = \mathcal{N}_2(x), P(y) = \mathcal{N}_2(y)\} = \frac{\sigma}{\sqrt{\sigma^2 + 2\sigma_2^2}}
\end{aligned}
\tag{3.12}
$$

The ideal parameter σ minimizes $\mathbb{E}_1, \mathbb{E}_2$ while maximizing $\mathbb{E}_3$ and $\mathbb{E}_4$. This means that σ should be large enough for a high estimation between labels within the same distribution, and small enough for a low estimation between labels of different distributions. A straightforward solution is to maximize the following expression:

$$
\max_{\sigma} \mathbb{E}_3 + \mathbb{E}_4 - \mathbb{E}_1 - \mathbb{E}_2 \tag{3.13}
$$

On the same example of graph family, let us choose $\mu_1 = 0, \mu_2 = 1$ and $\sigma_1 = 0.05, \sigma_2 = 0.1$. The numerical solution of the maximization problem above indicates an optimal value for the kernel parameter as $\sigma^* = 0.43$. In Figure 3.5, we present the metrics determined by different values of the kernel parameter σ. Low values of σ amplify the noise within the labeling because random variables generated by the same distribution may appear to have a low similarity. High values of σ determine a metric that is too uniform, and values around σ^* are a good compromise.

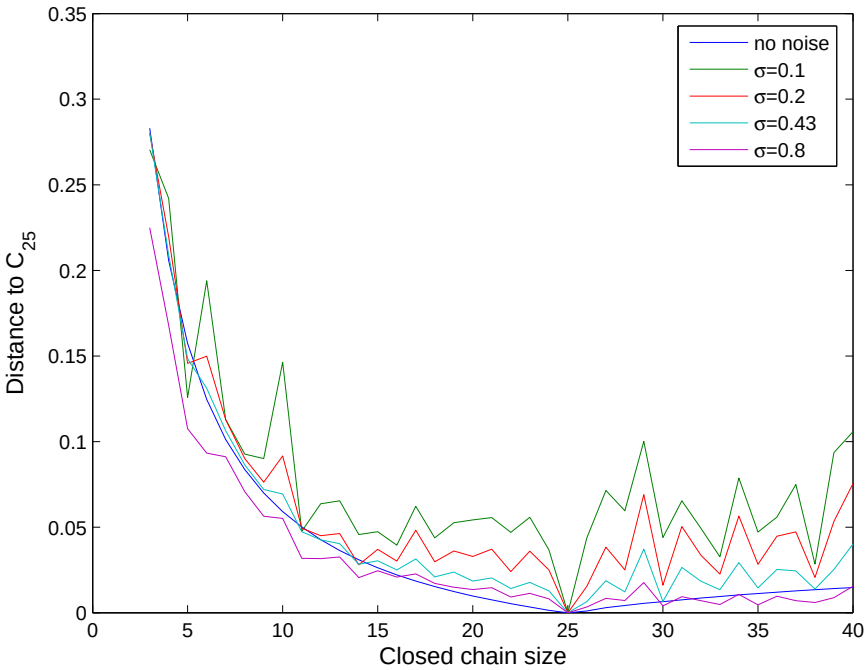


Figure 3.5: Distances in the graph feature space between C_{25} and other closed chains. Vertex labels are generated according to $\mathcal{N}(\mu_1 = 0, \sigma_1 = 0.05)$ except one vertex label per graph which is generated according to $\mathcal{N}(\mu_2 = 1, \sigma_2 = 0.1)$. $p_q = 0.4$ We use Gaussian kernels for vertices with different σ parameters. The reference line represents the distance values in the absence of Gaussian noise on the labels. The other functions show the effect that different values of the RBF parameter σ have on the distances. Note that small values of σ amplify the label noise, and large values determine too much uniformity.

3.5 Conclusions

Using the results from the experiments on the graph family, we can reach some meaningful conclusions concerning the applicability of labeled graph kernels. Unlike graph matching methods, graph kernels are appropriate for pattern analysis based on structural *and* labeling properties. Labels may be inferred from structure alone, but graph kernels are particularly useful when increasingly complex labels characterize the data.

The graphs that we used in order to exemplify the gradual extension to continuous labels are inherently simpler than real data. In reality, we do not have very precise information concerning the underlying label distributions. Moreover, the optimization that we used in Equation 3.13 is one possible choice among many measures which may indicate different optimum points (some relevant work may be found in fairness measures used for network resource allocation (Lan et al, 2009)). This ambivalence and

the specificity of different learning problems and datasets may suggest that extensive kernel parameter search methods like the grid search are recommended for fine-tuning the kernel functions.

More importantly, the major advantage of using kernels over graph matching is that the distance which endows the graph feature space allows us to employ a SVM classifier, rather than simply find the closest matches of a given input. More precisely, the prediction of the classifier is closely related to the distance between pairs of objects in the feature space, but the optimization of the decision function takes into account the entire training set. Therefore, the closest points to an object to classify do not necessarily indicate its class under this global optimization assumption.

Chapter 4

Graphical Models In Image Representation

If the doors of perception were cleansed, every thing would appear to man as it is, infinite.

(William Blake, *The Marriage of Heaven and Hell*)

4.1 Introduction

Image interpretation requires a general representation of images that have to be processed. The difficulty of this task is that a robust representation of a category builds on information that goes beyond primary visual features, such as color and shape. Intuitively, what makes humans so effective in image processing tasks is their ability to interpret these primary visual features and, furthermore, to identify useful prior knowledge and incorporate it into category representations. Prior knowledge is very often related to the presence of multiple objects in an image, to spatial information between these objects and helps the observer analyze the image against personal semantic concepts.

This difficulty relates specifically to the concept widely known as the *semantic gap*. This phenomenon became obvious once image processing went past low-level feature extraction, some of the most remarkable operations being:

- edge detection (Roberts, 1977; Prewitt and Mendelsohn, 1966; Sobel, 1970; Canny, 1986; Hildreth and Marr, 1979; Petrou and Kittler, 1991)
- curvature detection and corner extraction (Harris and Stephens, 1988; Mokhtarian and Suomela, 1998)
- patch analysis and saliency operators (Kadir and Brady, 2001; Lowe, 1999, 2004)

- motion detection by differencing (Barnard and Fischler, 1987; Horn and Schunck, 1981; Barron et al, 1994)

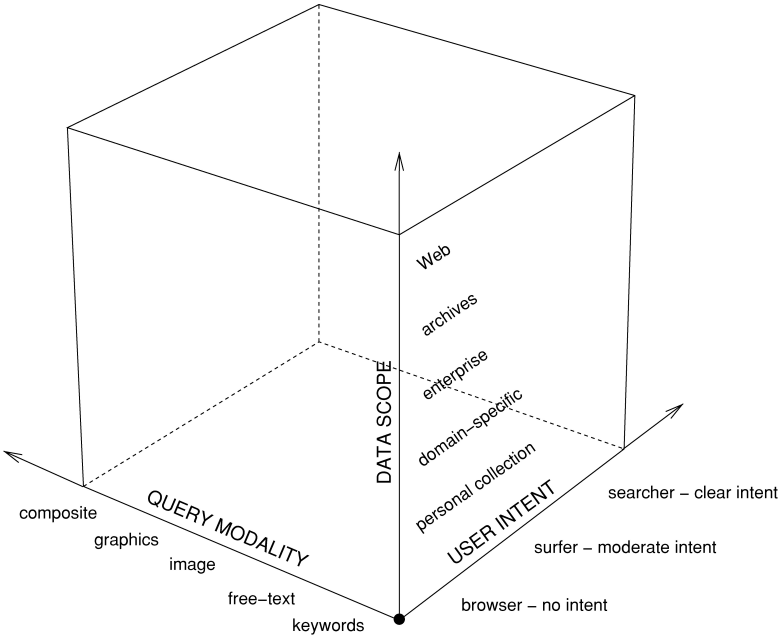
The interest towards more complex processing tasks increased as recreational and professional use of images gained large interest. For instance, in the context of Internet accessibility and wide availability of digital cameras, people may conveniently share personal images with others. By definition, low-level features disregard shape information and spatial relations, thus being unadapted to what people perceive in images. High-level feature extraction revolves around find shapes in digital images, during a complex process which is quite similar to the way in which we perceive the visual information around us. Typical tasks may consist in detecting particular combinations of features (e.g., red cars), a specific setting (e.g., a pile of books), a specific event (e.g., a penalty kick) or a subjective concept (e.g., a sad person or a nice house). The fact that some tasks seem to be more difficult than others was formalized by defining three levels of complexity (Eakins, 1993, 1996, 2002):

- Level 1 interpretation is based on primary features such as colour, texture, shape or the spatial location of image elements. At this level, the features being used are rather objective, their estimation is performed straightforwardly, and a knowledge base is not required. At this level, the applicability is usually limited to specific image categories and tasks.

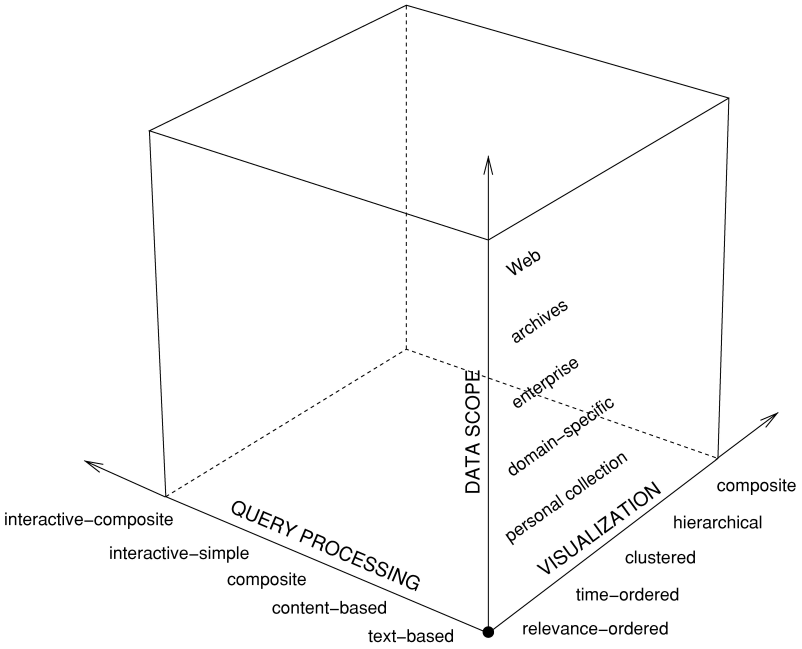
- Level 2 interpretation involves some degree of logical inference on derived features concerning the elements depicted in the image. At this level, queries are performed in order to retrieve objects of a given type (e.g. fighter planes) or in order to retrieve a specific object or person (e.g. Il Colosseo, or Sigmund Freud). The need for a knowledge base that goes beyond the scope of the interpreted image is obvious. Demand for level 2 interpretation is superior to that for level 1 interpretation, and a large amount of queries that people articulate while performing common tasks fall into this category.

- Level 3 interpretation makes use of abstract features and involves a significant amount of high-level reasoning about the meaning and purpose of the objects or scenes depicted. At the same time, complex reasoning and subjective judgement are also required in order to link image content to concepts.

Within this classification of query types, strengths and limitations of different techniques are easier to determine. The most significant gap lies between levels 1 and 2, and since levels 2 and 3 together are traditionally referred to as semantic image retrieval (Gudivada and Raghavan, 1995), we denote the gap between levels 1 and 2 as the *semantic gap*. In reality, query classification is even more complex, and can be more accurate if we take into account different user and system perspectives based on the data scope and the specificity of queries (Figure 4.1).



(a) User perspective on image retrieval



(b) System perspective on image retrieval

Figure 4.1: A comprehensive understanding of image retrieval shows the complexity of an ideal multilevel interaction between user and system (Datta et al, 2006)

4.2 High-level semantic image retrieval

4.2.1 Dealing with the semantic gap

The main effort in high-level semantic image retrieval is directed towards relating low-level features to semantics. The most effective approaches to date are:

- using object ontology in order to define high-level concepts,
- using machine learning methods in order to associate high-level concepts to low-level features,
- taking into account user feedback in a relevance loop in order to cope with subjective concepts,
- infering visual content based on textual information extracted from image context.

Most content based image retrieval systems exploit a combination of two or more of these approaches in order to perform high-level semantic image retrieval. Comprehensive surveys on mixed CBIR strategies for bridging the semantic gap between user and machine can be found in Smeulders et al (2000); Eakins (2002); Datta et al (2006); Liu et al (2007); Vasconcelos (2008). Although the trend is promising, the problem of designing systems that understand image content at semantic level remains open.

4.2.2 The task we address, and our approach

Our work falls in the second approach depicted above, the learning of concepts using machine learning methods, for the purpose of *image categorization*. This particular task consists in labeling an input image by one of some predefined categories, and has proved to be a difficult problem in computer vision. Some of the reasons that explain this factual outcome are structural variability which leads to different image representations of the same object or of object instances belonging to the same class, clutter and occlusion, and complex associations between an object defined as a semantic concept, a subjective representation of the concept and image context (Figure 4.2). Complementarily, the fact that the same image may have multiple subjective semantic interpretations also increases the difficulty of this problem.

Image categorization has often been employed as a preprocessing step for speeding-up more complex image retrieval tasks in large databases, or for performing automatic image annotation. Therefore, just as unsupervised clustering in the absence of label data, categorization is considered a fundamental early step in image retrieval. Image categorization is an interesting option if the image database is well specified, and labeled training samples are available. This is often the case for domain specific datasets, such as medical, remotely sensed of art-related images.

Categorization may be performed using either discriminative or generative modeling approaches. In discriminative modeling, classification boundaries or posterior

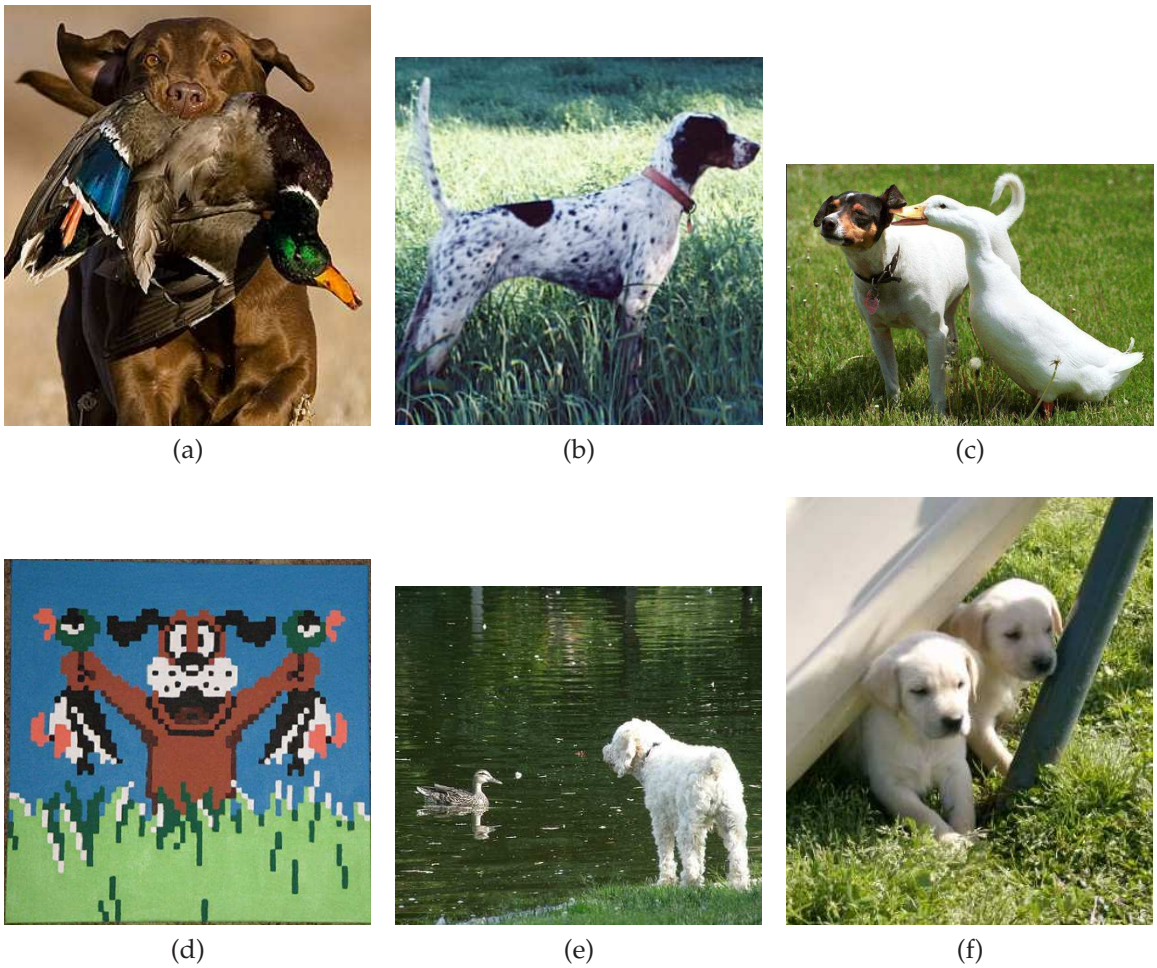


Figure 4.2: Find instances of the class “hunting dog": an example of how image categorization must cope with variability, conceptual perception of a semantic notion and contextual inference. 4.2a Hunting scene with occluding relevant elements. 4.2b Prior knowledge of hunting pose or breed is necessary. 4.2c The sole presence of one of more specific elements (in this case a goose) is neither necessary nor sufficient for taking a decision. 4.2d An outlier which could nonetheless be included in the “hunting dog" category. 4.2e A difficult scene the interpretation of which requires thorough attitude and context analysis. 4.2f A scene addressing personal interpretation of the term “hunting dog" and possibly prior breed knowledge.

probabilities of classes are estimated; in generative modeling, the density of data within each class is estimated and the Bayes formula is used to compute the posterior. Each approach is beneficial in a certain way: discriminative modeling directly optimizes class boundaries, whilst generative modeling incorporates prior knowledge more easily and is more practical if many classes are present.

In the following sections, we address the two fundamental issues in content based

image retrieval: how to formally describe an image, and how to estimate the similarity between image abstract representations. An interesting point worth mentioning, which is at the core of computer vision challenges, is that these two processes are entwined since the representation determines the similarity measure and its outcome, which in turn may help refine or rethink the representation.

4.3 The image model

If we refer to earlier work in image interpretation, a solid improvement in recent years has been made concerning the extraction of new features and the construction of image representations based on these features. This advancement has also fueled the growth of novel methods better suited for estimating representation similarity.

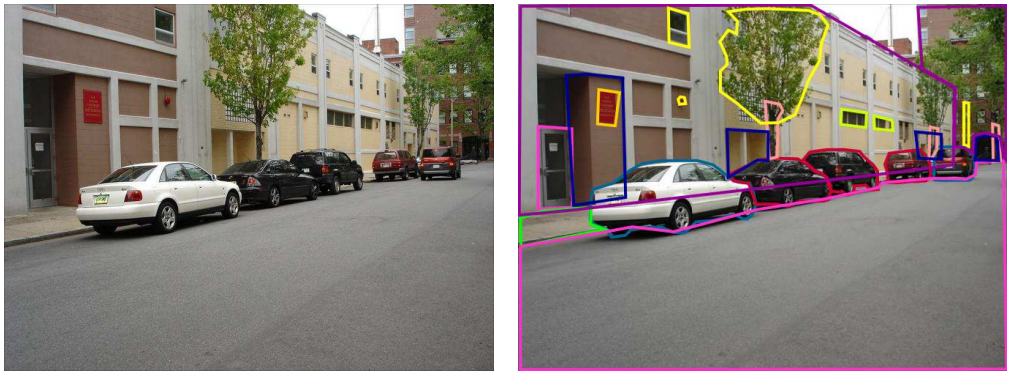
More specifically, we witness a major shift from global feature representations for images, such as color histograms and global shape descriptors, to sets of localized features and descriptors, such as salient points, region-based features, spatial model features, and local shape characterizations. This is a consequence of the comprehension that global features are insufficiently adapted to reduce the semantic gap, whilst local features often correspond with more meaningful image components and make association of semantics with image portions more straightforward. In the following, we focus on this approach for image representation.

4.3.1 Image segmentation

The first step in the buildup of a region-based image representation is the segmentation process. The importance of segmentation as a preprocessing step is critical, since the values of features that are extracted afterwards may be influenced drastically by segmentation precision. A review of current segmentation advancements is beyond the scope of our work, but since hih-quality segmentation remains hard to achieve, we mention here some current issues that affect segmentation based image processing.

Segmentation evaluation is one of these topics (Zhang, 1996; Roman Roldan et al, 2001; Paglieroni, 2004). It remains difficult to estimate whether an algorithm produces more accurate segmentations than another, because segmentation itself is usually "ill-posed" (what is the perfect segmentation of a complex scene?). Therefore, the most common method for evaluating the performance of a segmentation algorithm remains human evaluation, which is usually a tedious, subjective and non scalable process. Automated, unsupervised evaluation methods which should be employed in large scale region based image representation are even less reliable (Zhang et al, 2008).

In Figure 4.3, we present an image along with a set of labeled parts from the LabelMe project (Russell et al, 2008). The issue of evaluating a given segmentation is illustrated very well, since the regions highlighted by human subjects change with the observation scale, and there is no ideal trade-off between segmentation simplicity and representation accuracy.



(a) (b)

Figure 4.3: An image from the LabelMe project (Russell et al, 2008) and various labels suggested by human contributors, which exemplify the subjective, scale dependent aspect of image segmentation.

For a similar reason, in the case of iterative segmentation algorithms that use low-level information, the absence of semantic cues hinders the use of prior knowledge for evaluating the current state. Machine learning is needed in order to take into account presumptive features of regions, that would correspond to real-world objects. The drawback of this approach is that segmentation is no longer a preprocessing step, but becomes a part (and a possible bottleneck) of an image interpretation loop. In Figure 4.4 we present an input image from the Corel dataset along with two segmentations generated during an unsupervised automated hierarchical process (one of the most accurate intermediate results, and the segmentation situated at the end of the hierarchy). Even in situations where image scale is not an issue, natural scenes are reputedly difficult to segment, and choosing the most useful segmentation result is not trivial.

(a) (b) (c)

Figure 4.4: Unsupervised hierarchical segmentation (Meyer, 2001) performed on a Corel image. In the context of natural scenes, unsupervised segmentation performance is usually low, therefore identifying the best intermediate result is critical. This example also illustrates the need of prior information and semantic cues in order to improve the segmentation process.

Many segmentation algorithms that are widely used nowadays are based on previous

work that used watersheds (Beucher and Meyer, 1993), normalized cuts (Shi and Malik, 2000), Markov random fields and EM (Zhang et al, 2001), mean-shift techniques (Comaniciu and Meer, 2002), Markov chain Monte Carlo methods (Tu and Zhu, 2002) or Gaussian mixture models (Carson et al, 2002). On top of this existing framework, the innovative efforts are directed towards finding perceptual grouping principles based on statistical inference, prior knowledge and object models.

4.3.2 Alternative region detectors

The common notion of region can be extended to image keypoints that must show invariant and stable properties in order to be detected with high repeatability in images taken from multiple viewpoints. The obvious advantage is that identifying these keypoints allows for a compact but still pertinent image representation, just like in the case of common image regions. Given the interest for scale, rotation or occlusion invariant detectors, a large number of different local descriptors have been and are being proposed. Among the most popular detectors we mention the difference of Gaussian (DoG) (Lowe, 2004), the Hessian-affine detector (Mikolajczyk and Schmid, 2004), the maximally stable extremal region detector (MSER) (Matas et al, 2002), the intensity extrema-based region detector (IBR) (Tuytelaars and Gool, 2004) and the Fast-Hessian (Bay et al, 2006). Each specific detector is known to perform better under certain circumstances (Li and Allinson, 2008) and the output representation may be more or less suited for the machine learning algorithm that performs the interpretation task afterwards.

4.4 The region model

Our approach is to use a part-based model, which has been shown to cope well with inner category variability in specific (Mikolajczyk et al, 2004; Suard et al, 2005; Toews et al, 2006; Cour and Shi, 2007) or more general (Lowe, 2001; Fei-Fei et al, 2003; Bar Hillel et al, 2005) settings, in image categorization as well as in other image processing tasks. This type of model provides a realistic mechanism to combine local features into loose geometric assemblies and to make use of correlating positions of parts.

The part-based model is characterized by a tradeoff between exploiting abundant local visual features in a "bag of features" approach while disregarding geometric information, and extracting a constellation model. The first methods must usually cope with the problem of incorporating features that do not belong to interest objects into the object model as a result of no geometry information being used during training. The latter, which are closer to the human natural way of representing objects as parts with mutual geometric constraints, tend to be more complex and computationally intensive. The issues that our work addresses are represented by joining fundamentally different types of visual and spatial information into a flexible model and using them efficiently together. More precisely, the purpose of our work is the creation of a framework able to include image keypoints along with meaningful spatial information concerning them,

stored in a form which is adapted for visual spatial representations. By keypoints, we denote any entity contained in an image which is useful for its interpretation, and more specifically, categorization. A landmark could be as basic as the result of Harris corner detection, or as complex as a region highlighted by an image segmentation algorithm. There is no constraint whatsoever on the semantic complexity of this element; however, in order to be relevant, keypoints must exhibit more or less explicitly some visual invariance properties.

4.4.1 Related work

Different strategies in image categorization have been outlined in recent years. The initial approach of interpreting images as single indivisible objects and using global image features (Chapelle et al, 1999) evolved in order to cope with the major sources of difficulty mentioned before (Figure 4.2). A subsequent approach has been to treat images as *bags-of-features* (Sivic and Zisserman, 2003; Dance et al, 2004; Sivic et al, 2005), thus taking into account primarily the vectorization of the image content. Although this coarse representation, which disregards any spatial information between the detected features, drastically simplifies the input data, results have been surprisingly promising. In the ideal setting, the predefined visual dictionary contains words that are frequently discriminative when taken individually. Therefore, it is possible to perform image interpretation without the need to segment the image and infer object relative positioning.

An alternative approach has been the one of modeling objects as constellations of parts (Weber et al, 2000; Fergus et al, 2003; Lazebnik et al, 2004) into a probabilistic generative model. Its major advantage over the previous approach is that taking into consideration spatial information in the model improves its performance with respect to object shape and appearance variability. Although the bag-of-features approach implicitly preserves some spatial information, the constellation model allows for more complex inference, taking into account the appearance variability into an explicit spatial context. Thus, the constellation model offers a very intuitive and practical way of representing many classes of real-life objects. However, the high-dimensionality of the representation raises difficulties in using standard machine learning algorithms and increases computational complexity. At the same time, constellation models are ideally fit to represent rigid structured objects that are easily associated to a class model.

A lot of effort is currently directed towards designing image interpretation methods that benefit from advantages of both of these two fundamentally distinct approaches. Bag-of-features methods that take into account *local* subrepresentations (Lazebnik et al, 2006; Gosselin et al, 2007a,b) and flexible constellation models are bridging from both ends the same gap that exists between structureless, granular, semantically poor image representations and semantically meaningful, detail stripped models. A deep and thorough insight into the limits of bag-of-features models that takes into account this gradual transformation may be found in Marszałek (2008).

Under these circumstances, efforts have been made to steadily extend the bag-of-

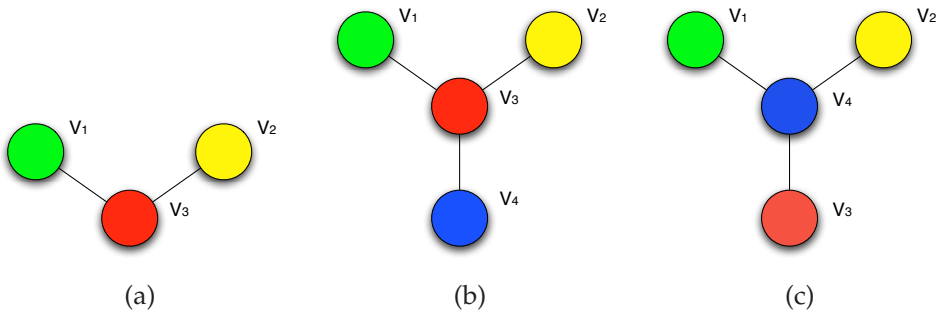


Figure 4.5: A basic example of labeled graph inference. Each vertex has an associated visual feature label and each edge encodes a spatial relation. In this example, the first two graphs are intuitively more alike, but the third one is closer to the second than to the first one. However, with a different spatial relation being considered between vertices (which would imply a different localization of the edges), the second graph could become more similar to the third one. In a real image representation, vertices and edges are associated to sets of labels, and inferring on labeled graph representations is challenging.

features model by adding more spatial information and feature correlation information into probabilistic discriminative methods. While this type of approach does not compute a category model, it should be able to rely on category prior knowledge and relations between parts, as generative models do explicitly. Previous attempts have used a mixture model using expectation maximization (Carbonetto et al, 2004) on local feature and spatial information, experiencing convergence issues. In Chen and Wang (2004), a diverse density function measuring a co-occurrence of similar regions from segmented images (which accounts for the spatial local information) has been used in a SVM hyper-space mapping, disregarding spatial information among regions whatsoever. A recent approach used Bayesian learning and conditional random fields for information propagation across image interest points (Carbonetto et al, 2008) with a very interesting result on selecting local features that best identify an object category. In Bar Hillel and Weinshall (2008), the authors are proposing a generative "star" graphical model (Fergus et al, 2005) with discriminative-like parameter optimization enhanced by boosting, which benefits from both approaches: generative modeling power and discriminative efficiency.

4.4.2 Our position within the structured representation framework

In our work, we aim to provide an alternative method for automatic feature selection in image categorization. We are motivated by the fact that there is still a gap between the performance of low-level feature based methods and that of semantically meaningful ones. We address the ambivalence between a complete lack of image semantics and good performance with a discriminative model that is able to passively infer concepts based

on various visual features along with spatial ones, such as “a dark object located close to a round one and far from other large objects". Image keypoints are associated as vertices into a probabilistic graphical model, and spatial relations are used for creating links between these keypoints. The advantage of this model is that it offers a very intuitive representation of image content. Moreover, by allowing the user to select the keypoint family and the spatial relations to be considered it can be more easily matched to a particular categorization task, thus tackling the “no free lunch" paradigm (Duda et al, 2000) that superior results on some problems must come at the cost of inferior results on other problems.

More specifically, we are annotating each vertex, representing a keypoint, with numerical visual descriptors. Each edge is annotated with descriptors of various spatial relations which are being considered and which are representative of the two nodes linked by the edge. While this type of structure has been used for model-based generative recognition tasks, its use in a discriminative machine learning framework is novel. The major challenge and the purpose of our proposed method is to be able to detect discriminative patterns into this labeled graphical structure during training. In Figure 4.5, we have depicted a simple example of a possible graph inference. In each of the three graphs, v_i represent evaluations of a certain descriptor for keypoints i while e_{ij} represents a spatial relation label associated to the edge (i, j), with $i, j \in \{1, 4\}$, if the edge (relation) exists. If we consider that, across the graphs, numerical values of the labels associated to the same indices are close, we want to be able to evaluate the similarity of these graphs and obtain some intuitive results, like *the first two graphs are alike with respect to the third one*, and *the third graph is more similar to the second one than to the first one*. Using this function in a discriminative classifier, we want to implicitly detect patterns that make the first two graphs alike and different from the third one.

Since vertex and edge labels are represented by sets of descriptors, another challenge arising in our approach is to make use of multiple descriptors synchronously for object category discrimination, as each descriptor should bring supplementary information concerning input data. This aspect, related to information fusion, is another original point of our approach.

4.5 Description of keypoint model constituents

We present now the major features that are associated in our model (and in part-based models in general) to segmented regions and, where applicable, to region based local descriptors. Bear in mind that an exhaustive review of region descriptors is beyond the scope of this section, but we encourage the reader to consult comprehensive work related to region and visual descriptors in general, such as Nixon and Aguado (2008).

4.5.1 Basic descriptors

One of the simplest possible scalar measures we can associate to a region is represented by its size or area. This value is defined as:

$$a_i = \sum_x \sum_y I_i(x, y) \tag{4.1}$$

where $I_i(x, y)$ is 1 if the pixel (x, y) belongs to region i, and 0 otherwise. This simple descriptor is invariant to rotation and it changes with scale. Another simple descriptor is represented by the region perimeter, which can be expressed as:

$$p_i = \sum_j \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2)} \tag{4.2}$$

where (x_j, y_j) represents the j-th pixel forming the border of the region i.

4.5.2 Color

Color is one of the most perceptually significant object properties that is involved in pre-attentive visual processing (Treisman, 1985). We rely heavily on color to identify and categorize objects, and color represents a fundamental variable in digital image processing and a plethora of related domains. This is reflected by the fact that early image interpretation systems were entirely based on color information (Swain and Ballard, 1991) and later proposals continued to rely heavily on this feature. However, color perception is a complex phenomenon, and its understanding is related, among other things, to the issue of how lightness and brightness are determined, and how color differences are perceived afterwards. Therefore, various color spaces have been introduced in order to offer as precise a numerical representation of color as possible. Ideally, a color descriptor should be illumination and pose invariant, and capable to take into account specific surface properties that depend on the composition of the object.

In a certain color system, the color is usually encoded by the mean value if the region displays chromatic uniformity, or by the histogram which keeps a count of pixels exhibiting color values in predefined intervals. Quantization of these color spaces may be an issue (Syeda Mahmood, 1997) and must be mapped according to perceptual realities.

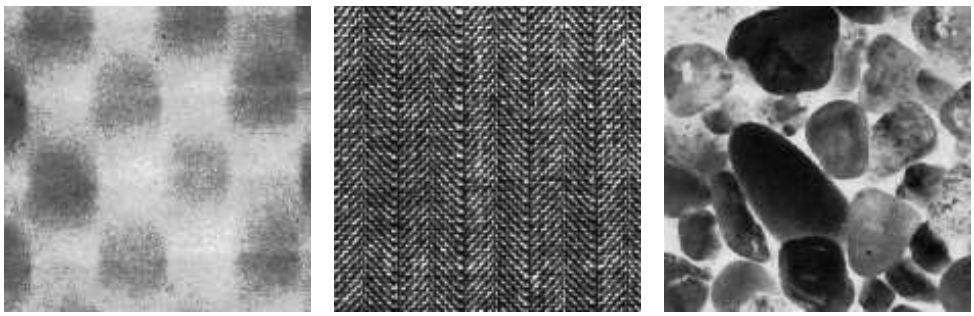
Obviously, these descriptors account for color occurrence exclusively, and not for localization, but the latter is present however, given the spatial information encoded implicitly in the determination of the concerned region.

4.5.3 Texture

There are multiple definitions and interpretations as to the nature of texture, many possible ways to represent and extract it, and a lot of research devoted to texture analysis.

A question such as "Is there any texture in this image?" is far from trivial and may be even considered ambiguous without further specifications (Karu et al, 1996). Therefore, uncommon definition alternatives may be useful, such as stating that texture is a quantity for which texture extraction algorithms provide meaningful results, or using an appropriate example database (Figure 4.6).

In general, an image region is textured if it displays repetitive patterns. Therefore, texture usually takes into account spatial variation and directionality in the area of interest, and may be very helpful in describing the region content.



(a) D8 (b) D16 (c) D23

Figure 4.6: Texture examples from Brodatz texture database (Brodatz, 1968)

Various fundamental approaches exist for texture representation and analysis; we can mention the use of multivariate Gaussian distributions (Weldon et al, 1996) , Fisher Linear Discriminant Analysis (Saito and Coifman, 1995), nearest neighbor classification (Strand and Taxt, 1994; Singh et al, 2001), learning vector quantization (Randen and Husoy, 1994), support vector machines (Kim et al, 2002), Gaussian Markov random fields (Cross and Jain, 1983), hidden Markov models (Povlow and Dunn, 1995) and hidden Markov trees (Choi and Baraniuk, 1999) etc.

For further references, we refer the reader to a review of early methods (Randen and Husoy, 1999) and towards a comprehensive more recent survey (Petrou, 2006).

We have experimented with fast Gabor filter approximations implemented using three-pole Infinite Impulse Response (IIR) filters (Bernardino and Santos Victor, 2006). A Gabor filter has a response in the space domain given by:

$$h(x, y, u, \theta) = \exp\left(-\frac{1}{2}\left[\frac{(x\cos\theta + y\sin\theta)^2}{\sigma_x^2} + \frac{(-x\sin\theta + y\cos\theta)^2}{\sigma_y^2}\right]\right)\cos\left(2\pi u(x\cos\theta + y\sin\theta)\right) \tag{4.3}$$

where u represents the modulating frequency along direction θ.

The output of the filtering process is an energy distribution $G_i(x, y)$ that corresponds to each filter h_i in the filter bank, and describes the response of the input image for different scales, orientations and wavelengths. Gabor features are found by calculating

statistics (mean and variance) corresponding to each image response:

$$\mu_i = \frac{\sum_x \sum_y |G_i(x, y)|}{|x| \cdot |y|} \tag{4.4}$$

$$\sigma_i = \sqrt{\frac{\sum_x \sum_y |G_i(x, y)| - \mu_i}{|x| \cdot |y|}} \tag{4.5}$$

The feature vector $F = [\mu, \sigma]$ for texture representation is created afterwards by concatenating the statistics above. In order to adapt to the specific structure of F, we use a distance metric defined by Arivazhagan et al (2006) which normalizes the representation with respect to the dominant orientation, which is equivalent to pre-aligning the textures before filtering.

4.5.4 Shape

Very often, color information is unreliable or absent, and shape can be used to describe images and locate objects within images. Shape is a fundamental property of an object, although less localized than color or texture. Defining a reliable shape descriptor is a difficult task, since various new issues such as occlusion, object flexibility, pose variations add to of the previous illumination change induced difficulties. Shape descriptors that are available satisfy some but not all of these requirements. Their complexity varies widely and, in general, sophisticated descriptors are more expressive but demand a very good quality of the input.

Consequently, and in contrast with the previous features, shape features require good segmentation to detect objects or regions. In the context of (unsupervised) image segmentation, the performance of the underlying task is so poor that the input for shape feature extraction is completely unfit and the overall results are inadequate. This difficulty is hard to overcome, for the fundamental reason that the segmentation process uses implicitly a window of information which might simply be smaller than the window of information needed for robust shape detection.

One of the simplest descriptors available is the compacity of a region, defined as the normalized ratio between its surface and its squared perimeter:

$$c = \frac{4\pi a}{p^2} \tag{4.6}$$

Statistical moments have been also widely used as shape descriptors for a long time (Hu, 1962; Prokop and Reeves, 1992). They have been proven to be quite robust to noise, which explains their popularity and success, but an important assumption concerning the input is that there is no occlusion.

The two-dimensional Cartesian moment of order p and q is defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{4.7}$$

This simple representation can be centralized and normalized in order to achieve translation and rotation invariance. Other invariant descriptors are Zernike and Legendre moments (Teague, 1980; Mukundan and Ramakrishnan, 1995), Chebyshev moments (Ping et al, 2002) etc., each family exhibiting specific advantages and disadvantages.

These sets of descriptors uniquely determine the original shape, in a process which is similar from this point of view to a Fourier transform. The fundamental difference however is that these moments are merely descriptors and cannot be used in practice to reconstruct the original input (Teague, 1980), unless they do not form an orthonormal family.

Other interesting edge-oriented approaches are based on embedding region edges in appropriate feature spaces that provide the model with invariant properties. This may be typically achieved by Kernel PCA (Romaniuk et al, 2004; Sahbi, 2007), but even though these models cope very well with morphologically reasonable variations, they are not very well adapted to occlusions and other issues commonly encountered in image processing.

4.5.5 Hybrid approaches

In more difficult settings, it may be beneficial to take into account the local correlations of texture, color and shape during feature extraction. More specifically, independent first-order statistics may not discriminate between complex, different inputs and their simplicity has the disadvantage of relating worse to human perception. The main motivation behind unified descriptors is the following: correlations are generated by the fundamental fact that intensity gradients exhibited by pixels are systematically related, and the concepts of color, texture and shape involve regularity at different scales.

This topic has a long history in computer vision, as well (Caelli and Reye, 1993) and many new descriptors have been progressively introduced in order to grasp higher-order correlations among color, texture and shape (Gagaudakis and Rosin, 2002, 2003; Prasad et al, 2004; Lin et al, 2009).

It is often believed that better image analysis can be achieved with more complex feature descriptors, but this turns out to be false. Not all features are helpful for image analysis in all situations, and unadapted descriptors may be harmful for the interpretation process and may actually decrease performance. It is thus important that descriptors are adapted to the complexity of the task and to the properties of the input.

4.6 Applying the marginalized graph kernel to image representations

The approach that we adopt consists in adapting a marginalized graph kernel function (Kashima et al, 2003) for image specific labeling, on image graphical representations. As any marginalized kernel, this kernel function between two image models is built upon the similarity of random paths that are extracted from the two graphs. The basic principle behind this method is that in similar graphs, one finds more frequently similar random paths (for the detailed description of the underlying algorithm we refer the reader to Section 2.2.1). Marginalized kernels have been used for labeled graph classification (Tsuda et al, 2002; Mahé et al, 2004) but in a discrete labeling setting. In image processing, region based image representations are used, but segmentation graphs are usually considered for specific image categories (satellite or medical images), in the context of subgraph matching. Learning using the whole segmentation graph raises the issues of graph distance and graph similarity, which are not trivial, given the labeling complexity and the inherent variability associated to segmentation.

4.6.1 Unsupervised segmentation and graph extraction

Our initial approach (Aldea et al, 2007) explores a methodological image categorization method using marginalized kernels for segmentation graphs.

The first step of our method consists in extracting and graph modeling image information. Within the segmentation graph, distinct regions correspond to vertices, while edges model the adjacency spatial relation between corresponding regions. Beside this spatial information brought by the structural expressivity of the graph, we integrate in the vertex labeling intrinsic information related to region visual properties. From this point of view, the interest of using a graphical model goes beyond the structural flexibility, and relates to the easiness to add and link visual features to specific components.

As input data, we settled for one class of head X-rays (coronal view) and a second class of mammographies (sagittal view); elements within the same class had not been acquired with the same medical systems following the same protocol. We mention that the method is not specifically tied to medical images and may accept any type of images as input. However, a higher difficulty to segment the input requires an adaptation of the unupervised algorithm that we present below, or its substitution with more sophisticated methods.

For the processing step, we adopted a generic hierarchical image segmentation paradigm (Beaulieu and Goldberg, 1989; Haris et al, 1998; Brun et al, 2005) that allowed us to perform unsupervised segmentation over all the input images. We suppose that each image is divided into components that may be further divided into subcomponents. This decomposition may be represented by a tree whose root node is the whole image and whose leaf nodes represent a partition of tiny regions built at the beginning of the processing step. This partition may be for example the set of pixels of the image. The

advantage of employing a hierarchical segmentation method is that changes are gradual, unlike for other methods where the variation of one parameter may induce a completely different segmentation map. This aspect is relevant because medical images which have been acquired using different protocols but show the same body parts are sensitive to segmentation methods that use absolute thresholds. As opposed to that, hierarchical segmentation gives emphasis to relative relationships between image subconstituents.

To generate the leaf node partition of the tree, instead of employing each pixel as a terminal node, we use a watershed over-segmentation that leaves us with a large number of small regions. However, in order to smoothen out very small variations in color and texture while preserving the overall boundary of objects, we apply anisotropic diffusion averaging before over-segmentation. Anisotropic diffusion has been shown to preserve important image structure while reducing unwanted noise (Perona and Malik, 1990; Black et al, 1998), and therefore it is useful as a pre-processing step before the actual image segmentation, since it helps to keep at low levels the apparition of small spurious regions.

At this point, we start "climbing" in the tree structure by merging adjacent regions that have the closest average gray levels:

$$dif_g(r_1, r_2) = \left| av_g(r_1) - av_g(r_2) \right| \tag{4.8}$$

where $av_g(r)$ denotes the average gray level in the region r.

Regarding the stopping condition for the region fusion process, we chose to set a dynamic threshold t_f. If the smallest gray level difference between two neighboring regions is higher than the threshold, we decide that the regions are not similar enough for the fusion to be performed and we stop. The threshold is dynamic because we compute it at each step as a (fixed) fraction f of the difference between the highest and the lowest region gray levels that exist in the image:

$$t_f = f \cdot (\max_{r \in I} av_g(r) - \min_{r \in I} av_g(r)) \tag{4.9}$$

The overall segmentation schema is presented below (Algorithm 1).

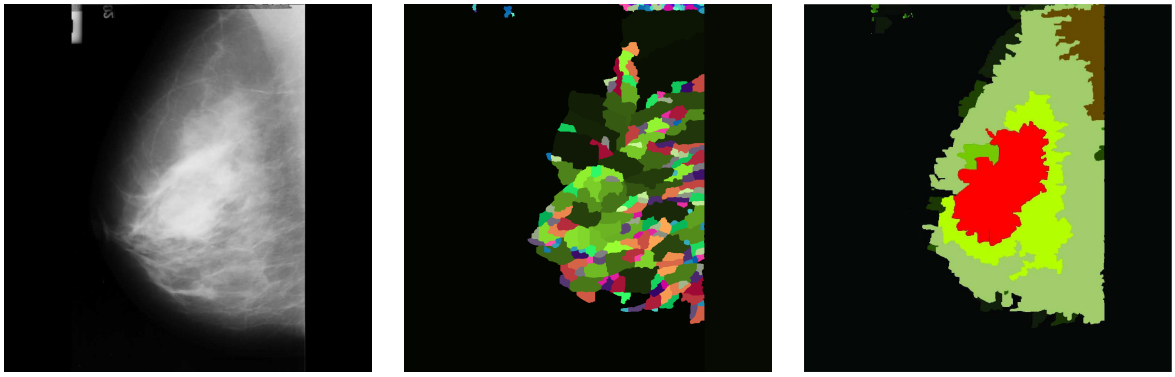
Once the fusion has ended, we compute the following features for the resulting regions, encoded as vertices:

- region area in pixels,
- relative area, a real value that represents the percent of the image covered by the specified region,
- average gray value of the region,
- relative average gray value, corresponding to an affine transform with respect to the highest and lowest average gray values in the image, g_{min} et g_{max}:

$$gray_{rel}(r) = \frac{g - g_{min}}{g_{max} - g_{min}}, \tag{4.10}$$

```
Data: Gray level image I, fusion threshold f
Result: Region segmentation {r_i} of image I
begin
    ApplyAnisotropicDifussion(I) ;
    OverSegmentation(I) ;
    foreach edge (r_1, r_2) do
        compute dif_g(r_1, r_2)
    end
    stop ⟵ false ;
    repeat
        compute t_f;
        compute (r*_1, r*_2) = arg min_{r_1,r_2} dif_g(r_1, r_2) ;
        if dif_g(r*_1, r*_2) ≤ t_f then
            r* ⟵ fusion(r*_1, r*_2);
            foreach r which borders r* do
                update dif_g(r, r*)
            end
        else
            stop ⟵ true ;
        end
    until stop;
end
```

Algorithm 1: Generic hierarchical segmentation algorithm



(a) Initial image (b) Direct segmentation (c) Region fusion

Figure 4.7: Mammography segmentation

- region perimeter,
- region compacity,

The only relationship encoded by the edges (implicitely) is the strict adjacency.

As an example, we present a typical mammography in Figure 4.7a, along with one of the best possible human-assisted watershed based segmentations (Figure 4.7b) and the result of the unsupervised method presented above (Figure 4.7c). The same situation is depicted for the other class in Figures 4.8a, 4.8b and 4.8c respectively.

4.6.2 Label kernels

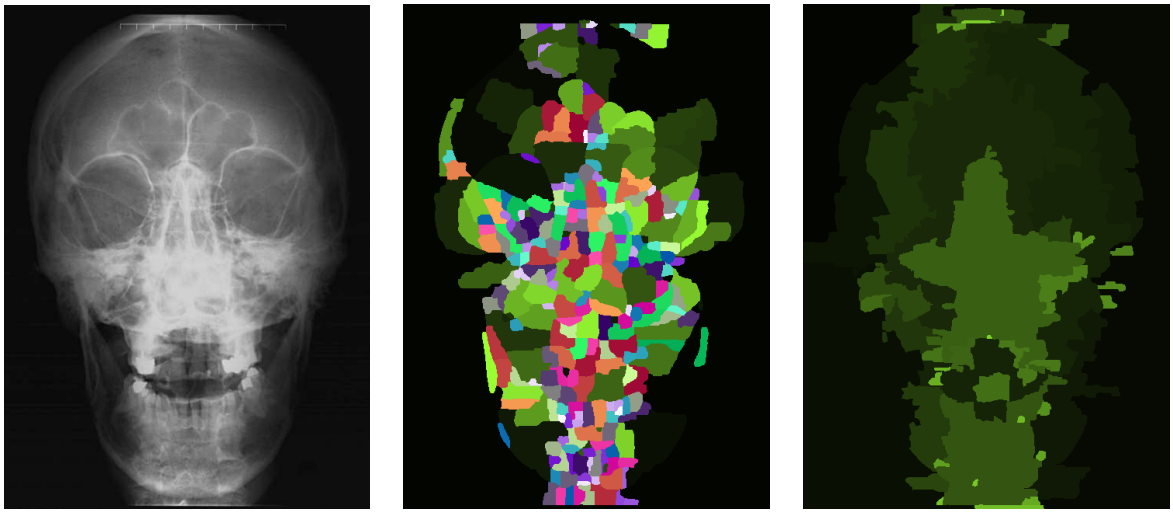
The graph similarity is assessed using a marginalized kernel function and is employed in a SVM classifier. We have explained in Section 3.2 how defining the graph similarity boils down to defining kernel functions for the labels that annotate the graph vertices and edges. We indicate now what choices are appropriate by default for different types of labels, unless specific numerical properties motivate the usage of uncommon kernels.

For scalar values, K^{RBF} Gaussian kernels (Section 2.3) are the most usual choice. Another possibility, which is as powerful as the Gaussian kernel in terms of separability and exhibits interesting scaling invariance properties, is the triangular kernel:

$$K_a^{\Delta}(f_1, f_2) = \max(1 - \|f_1 - f_2\|/C, 0) \tag{4.11}$$

This kernel is not positive definite, but belongs to the class of conditionally positive definite kernels (Berg, 1984), therefore we can still use it in the SVM algorithm. It is faster than the Gaussian kernels, and its compact support accounts for more sparsity in numerical computations.

For features represented as histograms or arrays, specific distances have been shown to perform very well. This is for example the case for Earth Mover Distance (EMD)



(a) Initial image (b) Direct segmentation (c) Region fusion

Figure 4.8: Head X-ray segmentation

(Levina and Bickel, 2001), χ^2, L^n or the Gabor feature distance presented in Section 4.5.3. The procedure employed in order to get a valid positive definite kernel based on such a distance d is known as *kernelization*:

$$k_d(h_1, h_2) = e^{-\alpha \cdot d(h_1, h_2)} \tag{4.12}$$

4.6.3 Fast computation of the graph kernel

As it has been highlighted in Section 2.8.2 , direct computation of the exact kernel value in Equation 2.46 is expensive since it involves the inversion of the $(I - \Pi_t)$ matrix, which scales as $O((|\mathcal{V}||\mathcal{V}'|)^3)$. There exist however alternative approaches that speed up the computation in practice up to a few orders of magnitude (Gardiner et al, 1992; Nocedal and Wright, 1999; Vishwanathan et al, 2006).

We have adopted an approximation method based on fixed point iterations. Starting from Equation 2.46, we denote by z the quantity:

$$z = (I - \Pi_t)^{-1} \mathbb{1} \tag{4.13}$$

We can further develop this expression as:

$$\begin{aligned} (I - \Pi_t) \cdot z &= \mathbb{1} \\ z - \Pi_t \cdot z &= \mathbb{1} \\ z &= \Pi_t \cdot z + \mathbb{1} \end{aligned} \tag{4.14}$$

and solving the approximation is equivalent to finding the fixed point of this iteration. By setting $z_0 = \mathbb{1}$, we can iterate by using:

$$z_t = \Pi_t \cdot z_{t-1} + \mathbb{1} \tag{4.15}$$

until a convergence condition $\|z_t - z_{t-1}\| < \epsilon$ is satisfied.

The approximation algorithm is presented below:

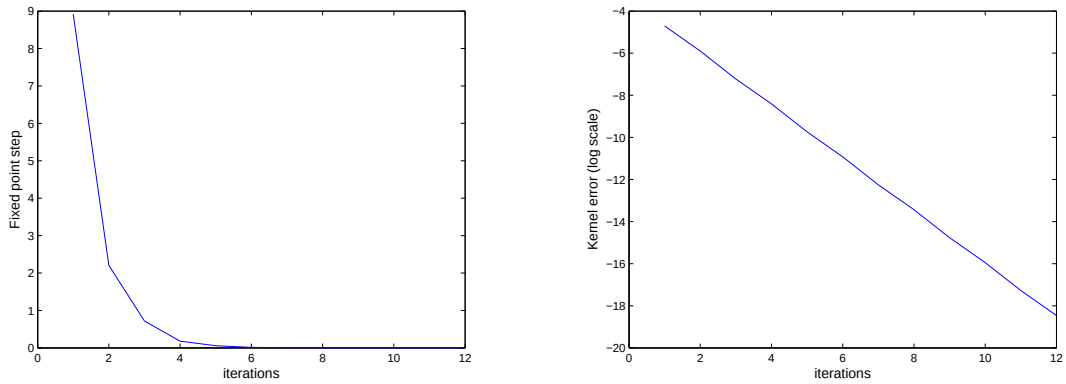
```
Data: Π_s, Π_t, ε
Result: Graph kernel K
begin
    z ⟵ 𝟙 ;
    z' ⟵ 𝟙 ;
    stop ⟵ false;
    repeat
        z ⟵ Π_t · z' + 𝟙 ;
        if ||z − z'|| > ε then
            z' ⟵ z ;
        else
            stop ⟵ true ;
        end
    until stop;
    K ⟵ Π_s^T · z ;
end
```

Algorithm 2: Fixed point graph kernel computation

We have verified this approach on a typical computation involving an image with $N = 39$ regions, generating a graph G with 39 vertices. The direct computation of $K(G, G)$ took $t_1 = 1.964$ s, and the approximative method ($\epsilon = 10^{-5}$) needed 12 iterations, performed in $t_2 = 0.076$ s. We present in Figure 4.9a the evolution of the stopping criterion quantity $\|z_t - z_{t-1}\|$, and in Figure 4.9b the iterative evolution of the distance between the approximation computed at iteration k and the exact value $K(G, G)$. This method performs significantly faster and is also convenient due to the fact that we are able to choose the desired precision.

4.6.4 Categorization performance

For the moment, edges are not taken into account, beside their implicit structural importance. Therefore, we consider them as having the same label and we focus on the richer vertex features. We have particularly analyzed two of them which are adjusting to global image content: the relative surface s_{rel} with respect to the image surface and the relative average gray value $gray_{rel}$ defined in Equation (4.10). They are less prone to perturbations, rescaling, contrast or brightness variations, etc.



(a) Evolution of $\|z_t - z_{t-1}\|$

(b) Distance to the exact value of $K(G, G)$ (log scale)

Figure 4.9: Fast computation of the graph kernel

In a first phase of our experiment, we have compared the performances of K^{RBF} and K^{Δ} kernels (Equation 4.12) for the s_{rel} feature and for different parameterizations of C and respectively σ, on a sample of 62 images, with 20 of them used for training. Given the fact that for two values (f_1, f_2) of a feature f, the similarity given by K^{Δ} is null if $\|f_1 - f_2\| \geq C$, we consider that we can compare K^{Δ} with a Gaussian kernel such that $C = 3\sigma$.

Categorization rate	$C = 0.05$ $\sigma = 0.017$	$C = 0.15$ $\sigma = 0.050$	$C = 0.2$ $\sigma = 0.067$	$C = 0.5$ $\sigma = 0.167$	$C = 0.6$ $\sigma = 0.200$	$C = 1.0$ $\sigma = 0.333$
K^{Δ}	0.81	0.74	0.83	0.83	0.83	0.86
K^{RBF}	0.93	0.95	0.93	0.86	0.86	0.86

Table 4.1: Categorization rates based on the relative surface feature s_{rel}

Results in Table 4.1 show that the Gaussian kernel performs well in the case of a strong discrimination (i.e., if region areas differ by more than one tenth of the image surface, the kernel returns a very small similarity value). While simplifying the discrimination function, the triangular kernel does not manage to discriminate as efficiently as the Gaussian kernel in the initial range of the surface feature.

4.6.5 Categorization with more than one feature

The typical formulation of the marginalized graph kernel (Section 3.2) prescribes the use of a kernel pair (K_v, K_e) in the buildup of a graph kernel. This means implicitly that we choose a vertex feature and an edge feature, and consequently the graph similarity estimation is based on these two values. However, in the general case of a feature set

$F = \{f_1, \ldots, f_n\}$ associated to a graph component, the kernel function may be extended in order to take into account several, if not all the elements of F. Kernel functions related to these various features allow us to treat them in a unified way, merging them in a global similarity estimate. As each kernel provides us with a partial description of data properties, we are intuitively interested in building a parameterized combination that employs each feature according to its relevance. The most straightforward and impartial method is to employ a linear combination of base kernels:

$$K_F = \sum_{i=1}^{|F|} \lambda_i K_{f_i} \tag{4.16}$$

where the multipliers $\lambda_i \geq 0$ satisfy $\sum_{i=1}^{n} \lambda_i = 1$. Here as well, the weighted sum of definite positive functions preserves the key property of definite positiviness of the result.

In a second step, we have built a linear kernel as in Equation (4.16) based on both s_{rel} and $gray_{rel}$, in order to analyze the classification performance as a function of the individual kernel multipliers. The tests are performed, as before, on the sample of 42 images. The discrimination thresholds are fixed at 0.2 and 0.5 for the surface and gray level features respectively. Gray level weight is gradually increased from 0 to 1 in the linear kernel combination.

The graph shown in Fig. 4.10 illustrates that performance is improved by combining multiple features in the global kernel function, even by using only two vertex features in the absence of edge labeling. Ideally, the weighted combination of kernels should be able to use information from multiple data sources by assessing the relative importance of each of them in an unsupervised manner. This issue is related to a very active topic in machine learning and will be developed in a following chapter.

4.7 Conclusions

In this chapter, we have presented the fundamental issues in content based image retrieval and various popular approaches that have been used in order to overcome these issues. We have also mentioned some major image representation solutions, along with their advantages that make them suitable for specific CBIR approaches, and their shortcomings. We have motivated our choice for a region based graphical model and we have reviewed the visual features that are the most adapted for this kind of representation. We have presented the use of a version of marginalized graph kernel which extends the one being used in computational chemistry and which allows the interpretation of image-based graphs. We have applied this approach for a simple medical image classification task, based on a generic segmentation method. The results validate this model and show that further work is necessary for investigating in an unsupervised manner which of the possible features are relevant for graph-based image representation and classification in a specific context. We are also interested in labeling

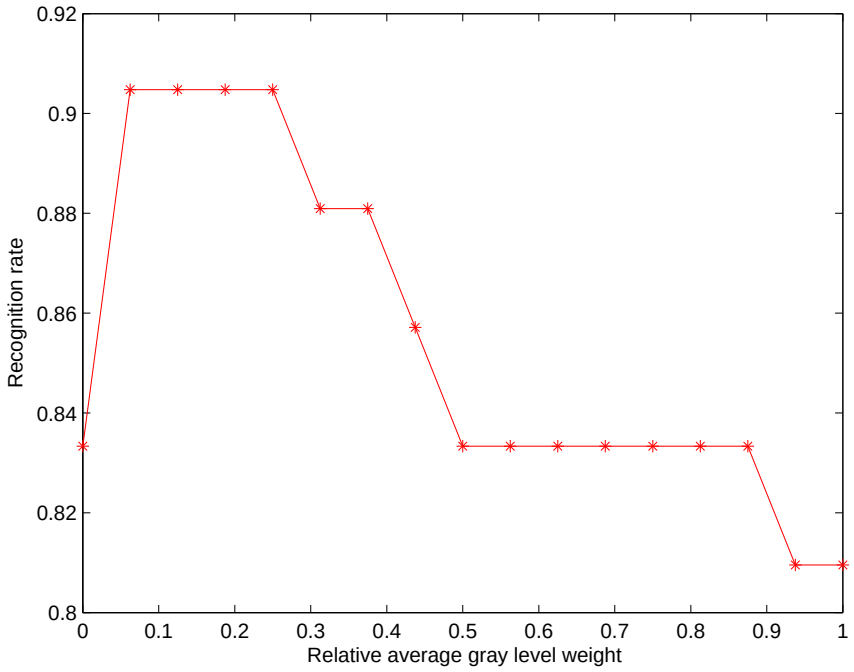


Figure 4.10: Performance of a linear combination between a triangular kernel for relative surface (C=0.2) and a triangular kernel for relative average gray level (C=0.5)

edges with spatial relation features that go beyond planar neighborhood and that are essential for expressing image content comprehensively.

Chapter 5

Fuzzy Spatial Relations and Graphical Models

5.1 Motivation for the use of fuzzy spatial relations

Graphs have been used as object models or in scene representation in order to benefit from their ability to encode relations between vertices. The most common visual graphical models are denoted as spatial graphs, since their vertices represent image keypoints, and edges represent spatial relations, such as the adjacency property that we reviewed in the previous chapter, but also more complicated relations related to distance, relative positioning, topology etc. Other graphs that encode more complex knowledge indirectly related to spatial information (such as semantic or conceptual graphs) are often structurally related to spatial graphs, too. However, we strictly focus here on graphical model based representations of input images, without adding prior structural or vertex related information.

Obviously, the use of graphical representations of entities is not restricted to the different fields of computer vision. Life sciences employ frequently spatial graphs for entity representation, but these structures model binary relations. These binary relations represent in fact the simplest possible relations between two entities, such as the presence or the absence of a chemical interaction, that may be further particularized using a set of features (e.g. type of chemical interaction). With few exceptions (Mahé et al, 2006), spatial relations being used follow this pattern, and simply encode the presence of an interaction between two elements of the whole structure.

5.1.1 Fuzzy information as an extension of binary relations

As to the extraction of spatial information for image representations, the situation is more complex. First of all, spatial interactions present an inherent semantic variability which goes well beyond the binary case mentioned above. This is accurate not only for complex spatial interactions such as "surrounded by", "along" etc., but even for

some of the simplest spatial relations such as the estimation of the distance between two objects. First of all, a binary crisp model is *unable* to encode and differentiate between subtle concepts such as "near", "close to" or "next to", and for a complex scene this fact is independent of the competence of a human subject who would be estimating the relation. Secondly, a binary model is often *unadapted* for the input that is provided in computer vision by preprocessing tasks, such as segmentation. In cases where the strict adjacency depends on a few pixels, the output graph may change severely as different segmentation approaches are employed. Figure 5.1 shows different possible segmentation results due to an ambiguous decision concerning the central area of the image.

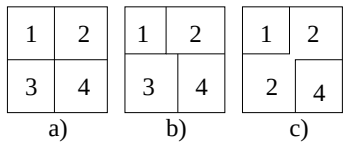


Figure 5.1: Strict adjacency does not exhibit robustness to preprocessing tasks such as segmentation. Here, we present different segmentation outputs that are possible due to an ambiguous decision concerning the central area of the image (a) All regions are adjacent to the others. (b) Regions 1 and 4 are not adjacent anymore. (c) Regions 2 and 3 merged , and regions 1 and 4 are not adjacent anymore.

Therefore, being able to encode more than an implicit strict adjacency is beneficial not only for a more precise description of the structural properties, but at the same time improves the overall representation robustness. The spatial graph is not a segmentation graph in the strict sense of the term, but an extended graph which may become even a full graph if needed.

5.1.2 Ambivalence of spatial relations

Even under the simplified binary relation model, it has been shown that information brought by the absence of interactions may increase performance, in relevant applications. For example, in protein-protein interaction (PPI) networks (Figure 5.2) the absence of protein interactions is also relevant for disease prediction. Therefore, an outcome prediction method presented by Borgwardt and Kriegel (2007) explicitly builds the complement graph $\tilde{G}$ of the initial interaction graph G, which encodes the absence of interactions: $\tilde{G} = \{\tilde{\mathcal{V}} = \mathcal{V}, \tilde{\mathcal{E}} = \mathcal{V} \times \mathcal{V} \backslash \mathcal{E}\}$. Then a composite graph kernel is built, using an initial simple kernel K:

$$K^*(G, G') = K(G, G') + K(\tilde{G}, \tilde{G}') \tag{5.1}$$

which leads to noteworthy improvements in classification accuracies on disease outcome prediction for cancer patients.

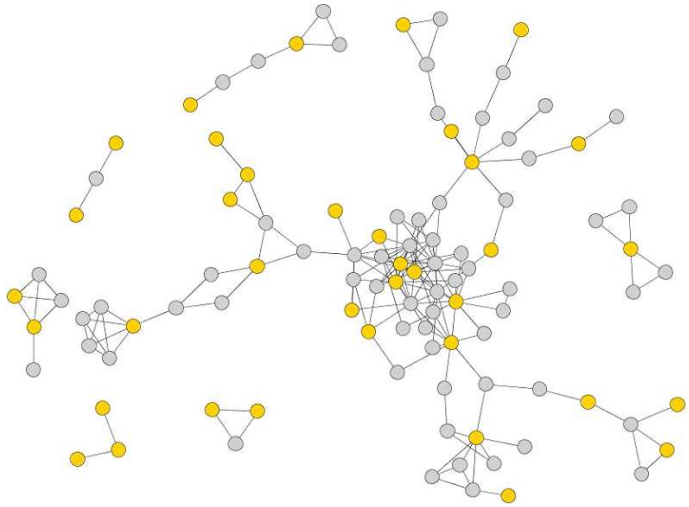


Figure 5.2: A typical PPI network example, in this case depicting proteins known to be involved in infertility (yellow nodes), and their interaction partners, depicted as gray nodes. (Center for Disease Systems Biology, Denmark)

This hints to the fact that, in computer vision, representing only positive estimations of a spatial relation may be detrimental, since the degree of absence of a spatial interaction may be in itself relevant to interpretation. We can think for example that "next to" and "far from" are not necessarily complementary, although they may be interpreted in a simplified manner as the two extrema of a unique specter. Therefore, the use of complement graphs is unadapted. Again, being able to express these nuances of spatial concepts in a compact form is beneficial.

5.2 Representation of objects by spatial fuzzy sets

Concrete domains are expressive means that enable to describe concrete properties of real world objects such as their size, their spatial extension or their color. A fuzzy extension can be obtained with *fuzzy sets* defined on these concrete domains, as degrees of acceptance of a property according to usual modeling methods in fuzzy set theory (Dubois and Prade, 1980), which may be represented by a trapezoidal membership function for instance. Consequently, axioms, rather than being satisfied (true) or unsatisfied (false) in an interpretation, become a degree of truth in [0, 1]. More details about the semantics of fuzzy description logics can be found in (Straccia, 2005).

In image interpretation, we can consider the image domain as a concrete domain. As a consequence, these fuzzy extensions defined over concrete domains can be a powerful means of reducing the semantic gap. More specifically, a spatial fuzzy set is a fuzzy set defined on the image space, denoted by $\mathcal{S}$, $\mathcal{S}$ being typically $\mathbb{Z}^2$ or $\mathbb{Z}^3$ for 2D or 3D images. Its membership function μ (defined from $\mathcal{S}$ into $[0, 1]$) represents the imprecision

on the spatial definition of the object (its position, size, shape, boundaries, etc.). For each point x of $\mathcal{S}$ (pixel or voxel in digital 2D or 3D images), $\mu(x)$ represents the degree to which x belongs to the fuzzy object. Objects defined as classical crisp sets are particular cases, for which μ takes only values 0 and 1. In the following, all definitions will include the crisp case as a particular case, so that the complete framework applies for both crisp and fuzzy objects and relations.

Different approaches can be chosen to model spatial relations by fuzzy sets and as a consequence, the fuzzy representations of spatial relations can be of various natures. The choice of the representation depends on the relation but also on the type of question raised and the type of reasoning one wants to perform. For our work, we will focus specifically on topological relations between two objects, which are based on notions of intersection, interior, exterior (for more specific information, Vieu (1997) performs a review of the qualitative formalisms). Fuzzy representations of topological relations allows us to cope with imprecision related to concepts such as inclusion, exclusion (Dubois and Prade, 1980), adjacency (Bloch et al, 1997) and other more complex concepts. An excellent overview of fuzzy spatial relations for image interpretation that introduces an ontology related to their semantics may be found in (Hudelot et al, 2008).

5.3 Image representations and fuzzy spatial relations

In addition to keypoint based features, we propose to improve the relevance of the graph structure by adding edges carrying supplemental structural information. Since spatial relations should be considered according to keypoint characteristics, we will illustrate our method by choosing segmentation regions as keypoints in the image and introducing some spatial relations that are particularly useful in this context.

Previous region based methods (Aldea et al, 2007; Harchaoui and Bach, 2007) use adjacency graphs: the edges indicate pairs of regions which are strictly adjacent at pixel level. The adjacency graph may be informationally poor since adjacency is a relation that is highly sensitive to the segmentation process and to the scene specificity, and whether it is satisfied or not may depend on one pixel only.

We propose the use of additional features based on structural information (Kuipers, 1978). As an application, we choose here to take into account two fundamental types of spatial information: distance and directional relative position. Note that other relations could be added as well, using the same framework.

5.3.1 Distance between regions

The distance between two regions R_1 and R_2 is computed as the minimal Euclidean distance between two points $p_i \in R_1$ and $q_j \in R_2$ (note that it is not a true distance):

$$d(R_1, R_2) = \min_{p_i \in R_1, q_j \in R_2} (d_{Euclidean}(p_i, q_j)) \tag{5.2}$$

Distance, as well as orientation, may not always be relevant, for instance the distance between two regions is the same if those two regions are adjacent by only one pixel, or if a region is surrounded by another region. Therefore we propose to consider a topological feature that measures the adjacency length between two regions.

5.3.2 Fuzzy adjacency

For our application, we consider a topological spatial relation represented by an extended degree of adjacency, described below.

The notion of adjacency is fundamental since it is related to a generic relationship between objects or regions in an image, and is therefore widely used as a feature in structural pattern recognition. We have seen in Sections 5.1.1 and 5.1.2 that a strict (crisp) definition of adjacency does not exhibit robustness in the presence of different sources of noise such as segmentation errors. One way to estimate fuzzy adjacency is to compute the matching between the area "near" a reference region and another region. This measure is maximal in the case where the reference region is embedded into the second, and is minimal if the two regions are far away from each other.

Fuzzy representations are entirely appropriate to model the intrinsic imprecision of several relations such as "near" (Bloch, 2005). We define the region of space in which a relation to a given object is satisfied as a fuzzy set. The membership degree of each point to this fuzzy set corresponds to the satisfiability degree of the relation at that point (Bloch, 2005). Note that this representation is in the image space and thus may be more easily merged with a representation of another region.

The spatial relation "near" is defined as a distance relation. A distance relation can be defined as a fuzzy interval f of trapezoidal shape on $\mathbb{R}^+$. A fuzzy subset μ_d of the image space $\mathcal{S}$ can then be derived by combining f with a distance map d_R to the reference object R:

$$\forall x \in \mathcal{S},\ \mu_d(x) = f(d_R(x)), \quad where \quad d_R(x) = \inf_{y \in R} d(x, y) \tag{5.3}$$

In our experiments, the fuzzy interval f is defined with the following fixed values: 0, 0, 10, 30 (Figure 5.3a). We exemplify using a butterfly image (Figure 5.3b) and the result of a segmentation exhibiting four distinct regions (Figure 5.3c). We illustrate the distance map to the region represented by the left wing (Figure 5.3d) and the fuzzy subset corresponding to the relation "near the left wing" (Figure 5.3e) which uses the distance map and the fuzzy interval defined above. Similarly, we compute fuzzy subsets for the right wing (Figure 5.3f) as well as for any other regions designated by the segmentation.

5.3.3 Directional relative position.

Several methods have been proposed to define the directional relative position between two objects, which is an intrinsically vague notion. In particularly, fuzzy methods are

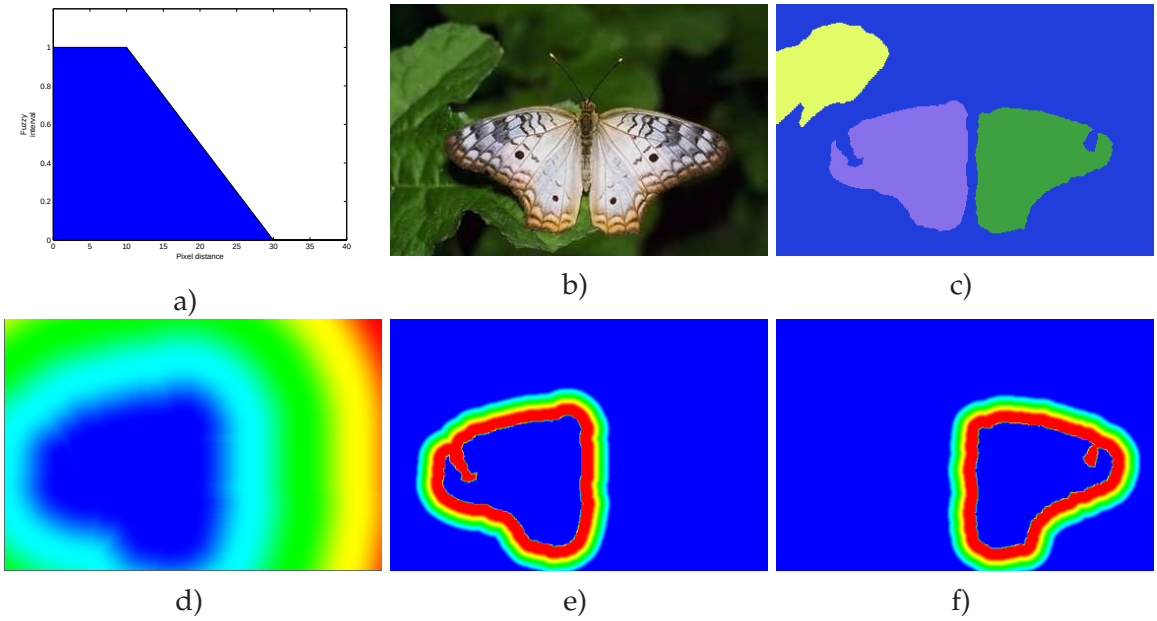


Figure 5.3: (a) Fuzzy interval for the distance relation. (b) Input image. (c) Segmentation result, four distinct regions. (d) Distance map to the region represented by the left butterfly wing. (e) Fuzzy subset corresponding to the relation "near the left wing" (red corresponds to highest values). (f) Fuzzy subset corresponding to the relation "near the right wing".

appropriate (Bloch and Ralescu, 2003), and we choose here to represent this information using histograms of angles (Miyajima and Ralescu, 1994).

This allows us to represent all possible directional relations between two regions. If R_1 and R_2 are two sets of points $R_1 = \{p_1, ..., p_n\}$ and $R_2 = \{q_1, ..., q_n\}$, the relative position between regions R_1 and R_2 is estimated from the relative position of each point q_j of R_2 with respect to each point p_i of R_1. The angle histogram $H_{R_1R_2}$ is defined as a function of the angle θ and $H_{R_1R_2}(\theta)$ is the frequency of the angle θ:

$$H_{R_1R_2}(\theta) = \left|\{(p_i, q_j) \in R_1 \times R_2 / \angle(\overrightarrow{i}, \overrightarrow{p_iq_j}) = \theta\}\right| \tag{5.4}$$

where $\angle(\overrightarrow{i}, \overrightarrow{p_iq_j})$ denotes the angle between a reference vector $\overrightarrow{i}$ and $\overrightarrow{p_iq_j}$. In order to derive a real value, we compute the center of gravity of the normalized histogram. A simple example is provided in Figure 5.4.

5.4 Fuzzy feature measures

A classification of comparison measures between fuzzy sets has been proposed in Bouchon-Meunier et al (1996). The advantage of this classification is that it takes into

a) Region 1 b) Region 2 c) Normalized histogram

Figure 5.4: Angular histogram between regions 1 and 2. The gravity center is represented by the vertical line.

account the semantic function of the underlying feature or relation, therefore allowing us to consider the most appropriate measure among similarity, dissimilarity, satisfiability, inclusion or resemblance.

5.4.1 Adjacency measure based on fuzzy satisfiability

So far we have defined the area of the image in which the relation "near to" a reference object is defined. The next step consists in estimating the matching between this fuzzy representation and a second region. Among the possible fuzzy measures, we consider as criterion a *M-measure of satisfiability* (Bouchon-Meunier et al, 1996) defined as:

$$Sat^k(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)} \tag{5.5}$$

where $\mathcal{S}$ denotes the spatial domain.

This fuzzy measure estimates the precision of the position of the object in the region where the relation is satisfied. It is maximal if the second object covers entirely the kernel of $\mu_{near(R_1)}$. Note that the size of the initial fuzzy set is not restricted and may be modified accordingly with respect to the image space. If the object R_2 is crisp, this measure reduces to $\frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)}$, thus representing the portion of $\mu_{near(R_1)}$ that is covered by the object R_2.

There are situations where a slightly different formulation is also convenient. If the object R_2 is relatively small, the fuzzy measure is low even if the object R_2 is entirely contained in the kernel of $\mu_{near(R_1)}$. Therefore, we can employ the following measure:

$$Sat^o(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \mu_{R_2}(x)} \tag{5.6}$$

Again, if the object R_2 is crisp, the measure reduces to $\frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in \mathcal{S}} \mu_{R_2}(x)}$, which represents this time the portion of the object R_2 which is contained in $\mu_{near(R_1)}$.

5.4.2 Adjacency measure based on fuzzy resemblance

Beside satisfiability, we also choose a symmetric measure, the *M-measure of resemblance* (Bouchon-Meunier et al, 1996) defined as :

$$Res(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \max(\mu_{near(R_1)}(x), \mu_{R_2}(x))}$$

This measure represents the ratio between the cardinality of the intersection of $\mu_{near(R_1)}$ and μ_{R_2}, and the cardinality of their union. It is maximal if the object and the relation are identical: this resemblance measure accounts for the positioning of the object and for the precision of the fuzzy set as well.

In Figure 5.3(e) and Figure 5.3(f) we have illustrated the fuzzy subsets corresponding to the two wings. With the fuzzy satisfiability measure defined above, we get a response of 0.100 for “right wing near the left wing" and 0.109 for “left wing near the right wing". It is equally worth noting that the two regions are disconnected with respect to crisp adjacency.

In the remaining sections, we will denote by a spatial relation $\mathcal{R}$ one of these measures of fuzzy adjacency, but we underline the fact that $\mathcal{R}$ could be substituted for other functions that estimate the interaction between elements of the image structure. The choice of the spatial relation of adjacency for our illustration is immediate because fuzzy adjacency information extends naturally one of the most simple and pertinent relations between image regions, the crisp adjacency. However, taking into account more complex spatial relations such as “parallel to” or “along”, along with their fuzzy measures of satisfiability (Vanegas et al, 2009; Takemura et al, 2005), is possible as long as these spatial relations are appropriate for the content of the input images.

5.5 Fuzzy spatial information and discriminative models

We can see that fuzzy spatial relations for images extend very conveniently the binary relations that are being used in other domains and achieve to merge at the same time information concerning the presence and the absence of an interaction. However, there are difficulties that arise when using this type of relations for discriminative learning.

In the context of image representation using spatial relations, interesting work has been done using binary relations (Deruyver et al, 2009) supported by a specific ontology, or by using a count vector (Lebrun et al, 2008) which estimates simple relative positioning. Typically, each independent spatial relation builds in itself a novel data representation, therefore additional work may be necessary in order to make use of different spatial features simultaneously and efficiently.

In this section, we focus on the fact that a single fuzzy spatial relation creates *by itself* an infinite set of different representations. Rather than using multiple spatial relations for learning, we try to underline the specific challenges that a discriminative learning algorithm has when using a family of representations generated by the same fuzzy spatial relation.

5.5.1 The information gain

Very often, there is a correlation between the value of a fuzzy spatial relation and the information gain: if the response is high, it means that the relation that the function has been designed for is much present. Consequently, low responses may be frequent (e.g. in the case of the "near" spatial relation) and may not bring the same amount of information.

Discriminative learning, and discriminative learning for labeled graphs in particular, makes intensive use of similarity assessments between input objects. The similarity score between two graphs increases if these graphs exhibit many similar substructures. From the learner point of view, similar but less meaningful labels are just as relevant as the labels that carry more structural information. Therefore we may experience a phenomenon where the repetitive similarity of less meaningful labels masks the similarity computed by the graph kernel using the occurrences of information rich labels. As a result, the graph kernel function will over-increase the graph similarity measure, thus diminishing its discriminative ability in the feature space. The cause of this phenomenon may also be highlighted within the following statement:

Proposition 5.5.1. *If regions A and B are connected in the graph G, and regions C and D are connected in the graph G', and if A is similar to C and if B is similar to D, and (A,B) and (C,D) are equally near, then the similarity between G and G' is increased.*

This result is a direct consequence of the behavior of stationary kernels, such as the RBF kernel; if features exhibit close values, the value of the kernel will be high and will not depend on the feature values. This behavior denotes that the discriminative model is polluted by the quantity of information that is not equally important for the task and that masks the relevant information. A reasoning model that would be more adapted for our situation, and that we need to adopt in order to be able to cope with noisy spatial information, is depicted by the following proposition:

Proposition 5.5.2. *If regions A and B are connected in the graph G, and regions C and D are connected in the graph G', and if A is similar to C and if B is similar to D, and (A,B) and (C,D) are equally near,* and A is near to B, and C is near to D, *then the similarity between G and G' is increased.*

5.5.2 Graph pruning

A straightforward solution to this situation is to prune the edges by thresholding the spatial relation values, so that edges will appear only when the fuzzy adjacency estimation between two vertices goes beyond a minimum value θ. We consider that, below a certain value, there is no relation whatsoever between two regions. This also implies that we consider that the absence of the spatial relation does not bring any help at all in the discriminative task. In the following, we denote by $\mathcal{V}(G)$ the vertex set of graph G, and by $\mathcal{E}(G)$ its edge set.

Definition 5.1 (θ-graph)**.** *Let us consider a region representation of an image I and a spatial relation $\mathcal{R}$. By $\mathcal{V}(I)$ we denote the set of vertices corresponding to the regions of I. A θ-graph G_θ is defined by:*

$$G_\theta = \{\mathcal{V}(I); (v_1, v_2) \in \mathcal{V}^2(I) | \mathcal{R}(v_1, v_2) \geq \theta\} \tag{5.7}$$

where $\mathcal{R}(v_1, v_2)$ is the generic spatial relation function between regions (vertices) v_1 and v_2.

In this way, we can associate to the pair $(I, \mathcal{R})$ an infinite set of graphical representations $\mathcal{G} = \{G_\theta\}_{\theta \geq 0}$. In reality, the size of this set is finite and is bounded by $\frac{\|\mathcal{V}(I)\| \cdot (\|\mathcal{V}(I)\| - 1)}{2}$. Two elements of interest in $\mathcal{G}$ are the full graph $G_f = G_\theta | \theta = 0$, and the fully disconnected graph $G_d = G_\theta | \theta > \max_{v_1, v_2 \in \mathcal{V}(I)} \mathcal{R}(v_1, v_2)$. Since G_f and G_d contains the maximum and the minimum possible number of edges respectively, we may state that from this structural point of view, the other elements of $\mathcal{G}$ are situated "between" these two particular members.

However, the structure of the strict adjacency graph enriched by the labeling determined by $\mathcal{R}$, that we further denote by G_a:

$$\begin{aligned} G_a = \{ & \mathcal{V} = \mathcal{V}(I); \\ & \mathcal{E} = \{(v_1, v_2) \in \mathcal{V}^2 | \, \text{border}(v_1, v_2) > 0; \\ & \mathcal{S}_v = \{\text{label}(v) | v \in \mathcal{V}\}; \\ & \mathcal{S}_e = \{\mathcal{R}(v_1, v_2) | (v_1, v_2) \in \mathcal{E}\}\} \end{aligned} \tag{5.8}$$

does not necessarily belong to this set $\mathcal{G}$ of θ-graphs. Given the perceptual importance of the segmentation graph, a valid question that is raised now is: "Which is the most consistent and reliable projection $\hat{G}$ of the segmentation graph G_a into the family $\mathcal{G}$?". The overall purpose of determining this projection is that if $\mathcal{R}$ is well adapted to preprocessing noise, then strict adjacency graphs modeling the same entity will have the tendency to project into more similar graphs $\hat{G}$, thus improving the representations robustness.

In order to project G_a into $\mathcal{G}$, we need an appropriate distance which we describe below.

Graph edit distance

One possible way to estimate a structural difference between two labeled graphs is to define a set Υ of allowed elementary operations υ, such as insertion, deletion or label modification of a vertex or an edge, and a set Ψ of costs ψ associated to the elements of Υ. Let us recall the general definition of the graph edit distance (Riesen et al, 2007):

Definition 5.2 (Graph edit distance)**.** *Let $G = (\mathcal{V}, \mathcal{E})$ be the source and $G' = (\mathcal{V}', \mathcal{E}')$ be the target graph. The graph edit distance between G and G′ is defined by:*

$$\delta(G, G') = \min_{(\upsilon_1, \ldots, \upsilon_k) \in \Gamma(G, G')} \sum_{i=1}^{k} \psi(\upsilon_i)$$

where the set of edit paths $\Gamma(G, G')$ contains all the possible finite sequences of elementary operations that transform G into G'.

Expliciting $\delta(G, G')$ for strict adjacency and θ-graphs

An important property that all the graphical models display here is that the vertex set is the same, as a unique representation of the regions of I: $\forall G, \mathcal{V}(G) = \mathcal{V}(I)$. Under these circumstances, the graph edit distance $\delta(G_a, G_\theta)$ between G_a and a $G_\theta \in \mathcal{G}$ depends strictly of the differences in their edge sets. In addition, if two edges are present in G_a and G_θ between the same vertices (v_1, v_2), they have the same label $\mathcal{R}(v_1, v_2)$. This means that we can restrict the set Υ to elementary operations of edge insertion and deletion only.

The edge set differences are generated by strictly adjacent regions that are not close enough in terms of θ-fuzzy adjacency, and non strictly adjacent regions that are close in terms of θ-fuzzy adjacency. A very simple definition of costs for the elementary operations is that we only apply a unit cost for adding or deleting an edge between pairs of regions. The graph edit distance may be expressed as:

$$\begin{aligned}\delta(G_a, G_\theta) = &\operatorname{card}\{(v_1; v_2) \in \mathcal{V}^2(G_a) | (v_1, v_2) \in \mathcal{E}(G_a) \wedge \mathcal{R}(v_1, v_2) < \theta\} \\ &+ \operatorname{card}\{(v_1; v_2) \in \mathcal{V}^2(G_a) | (v_1, v_2) \notin \mathcal{E}(G_a) \wedge \mathcal{R}(v_1, v_2) \geq \theta\}\end{aligned} \tag{5.9}$$

The graph or graphs $\hat{G} \in G$ that minimize δ are the closest (structurally) to the adjacency graph G_a. These are the projections of G_a in the set $\mathcal{G}$, and are ideally robust generalizations of G_a with respect to the spatial relation R.

The element that bridges the informational gap between G_a and the set $\mathcal{G}$ is the complete graph G_f, which includes (structurally) any element $G_\theta \in \mathcal{G}$, as well as the strict adjacency graph G_a. Ideally, the learning algorithm should exhibit the best performance with G_f, but then it should be able to cope well with the issue generated by a lot of similar low-information edge labels.

5.6 Experiments and results

5.6.1 Data set

The Internet Brain Segmentation Repository (IBSR) data set[1] contains real clinical data and is a widely used 3D healthy brain magnetic resonance image (MRI) database. It provides eighteen manually-guided expert brain segmentations, each of them being available for three different views, along reference planes: axial, sagittal and coronal. Each element of IBSR is a set of slices that cover the whole brain.

[1]The MR brain data sets and their manual segmentations were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at http://www.cma.mgh.harvard.edu/ibsr/

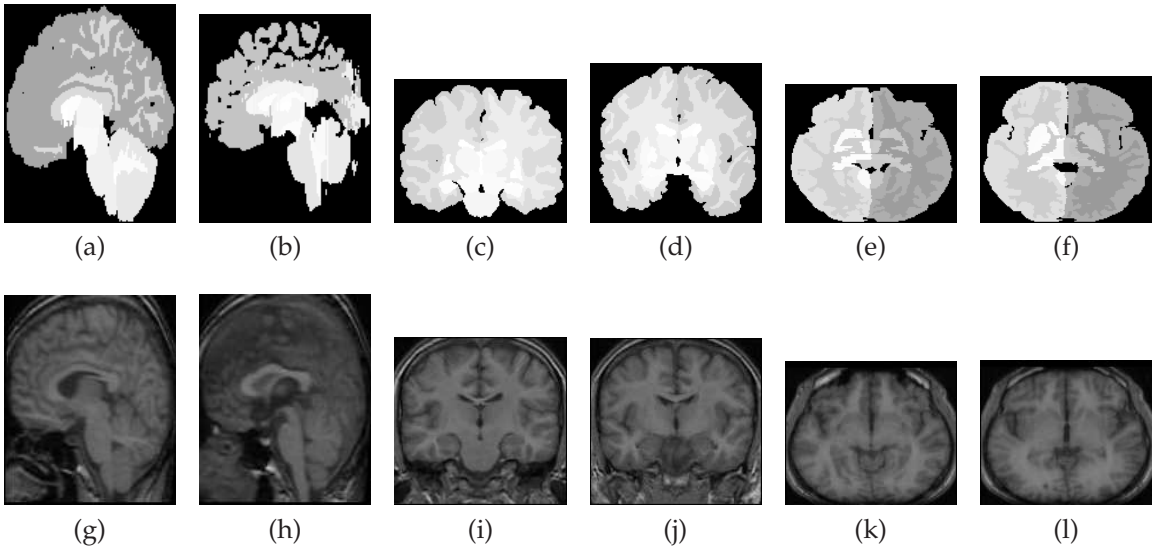


Figure 5.5: Samples from IBSR data set. 5.5a, 5.5b Two slices of the sagittal view of the same 3D MRI volume representing the two categories. 5.5c, 5.5d Coronal view. 5.5e, 5.5f Axial view. The original images are presented below their corresponding manual segmentations.

The main purpose of the data set is to provide a tool for evaluating the performance of segmentation algorithms. However, the fact that it is freely available and that it offers high quality segmentations as input data makes it also useful for our experiments, in the context of unsupervised image segmentation methods and their unsupervised evaluations still being far from perfect (Section 4.3).

5.6.2 Experimental setup

Image categorization between images belonging to different views in the data set (sagittal, coronal, axial) is performed with a 100% success rate for many of the features that we take into account. Therefore, we built a more challenging categorization problem by considering images belonging to the same view; a secondary benefit of this approach is that by choosing certain slices we can control the difficulty of the task.

Since the brain representation is made up of consecutive slices in any of the three views and the brain structure varies progressively, we created one of the classes by selected three consecutive slices at the same localization over all the eighteen 3D brain segmentations. A second class was built using three consecutive slices which are positioned at a certain distance from the first block; as the distance between the two blocks of slices decreases, the difficulty of the categorization task increases. We found out that choosing a distance of only two or three slices between the training blocks, along with category intra-variability, would account for a difficult categorization task. Table 5.1 references the total number of slices in each 3D brain view and the indices of slices being

Table 5.1: Identification of the slices composing the database in each view of the 3D volume, for the three possible views: sagittal (S), coronal (C) and axial (A)

View	Slices	Slices cat. 1	Slices cat. 2
S	256	121, 122, 123	126, 127, 128
C	128	58, 59, 60	64, 65, 66
A	256	121, 122, 123	126, 127, 128

used for defining each category; Figure 5.5 presents typical class elements for all views. Each brain view will provide three images for each class, thus creating a class definition of 54 images.

Concerning the graph construction and labeling, nodes are represented by manually segmented regions while edges account for spatial relations between regions. For vertex labeling, we use normalized region visual features: the mean gray level (which is normalized according to the lightest and darkest regions in the image), the relative region area (normalized according to the total image area) and the normalized region compacity, defined as the normalized ratio between its surface and its squared perimeter. In the following part of this section, we will experiment with the coronal view and with the mean gray level as region feature. Spatial relations based on adjacency measures being considered between image regions build up the edge labeling, respectively.

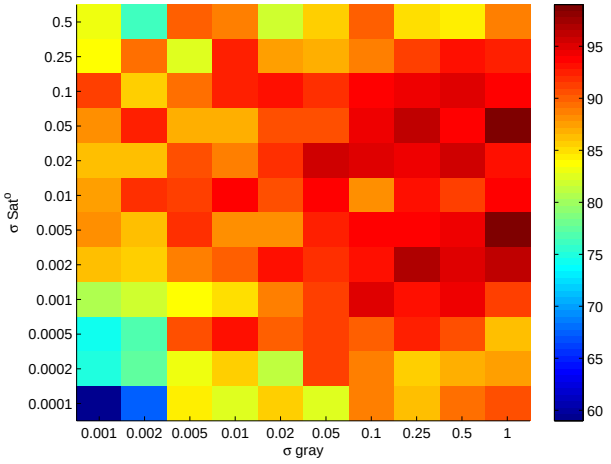
We perform n-fold cross validation on the data set ($n = 10$), and we repeat the classification task m times ($m = 10$); the performance given below is the mean value of these m executions.

5.6.3 Categorization with strict adjacency structures

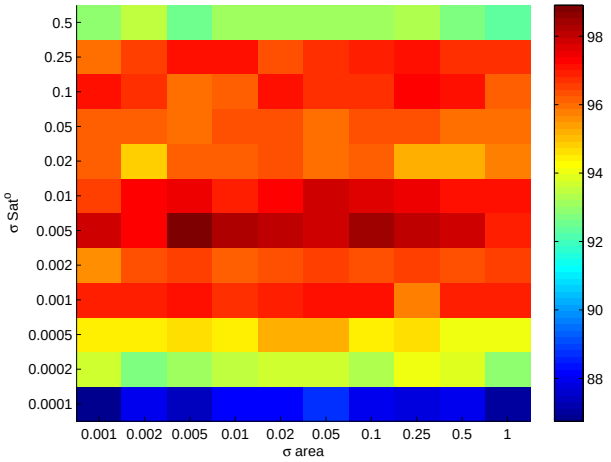
Given a region feature and a spatial relation, we build the *labeled* strict adjacency graphs G_a and we use RBF kernels with parameters that are adapted to the range $[0, 1]$ of these normalized features (see Figures 5.6, 5.7 and 5.8). Afterwards, we set up a grid search in the σ parameter space for each of the two kernel functions.

For each element of the grid, we try multiple values for the regularization parameter C of the SVM, $C \in \{10^{-2}, 10^{-1}, \ldots, 10^{6}\}$. Figures 5.6, 5.7 and 5.8 present the best classification performance for each pair $(\sigma_{vertex}, \sigma_{edge})$, for the values and features specified on the axis.

The novelty of this approach is that the adjacency graphs include as edge labels fuzzy estimations of the region adjacency. However, some regions which are strictly adjacent might have a small estimation of the fuzzy adjacency, while regions which are disconnected may present a higher value of $\mathcal{R}$. Next, we analyze the impact of adding structural information which is not necessarily tied to the strict adjacency between image regions.

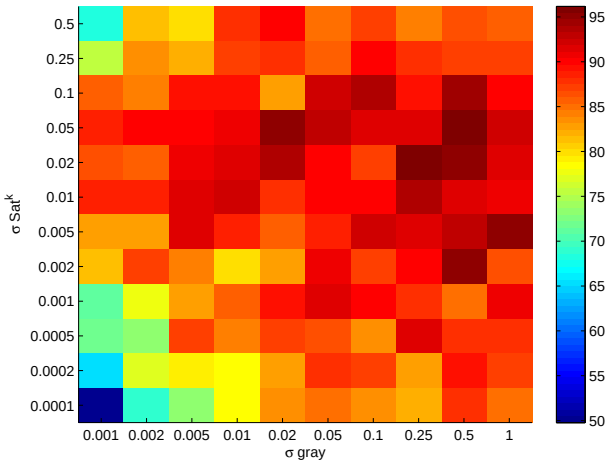


(a) Using the mean gray level region feature and the Sat^o measure, the best performance, 99.00%, is attained for $\sigma_{vertex} = 1$ and $\sigma_{edge} = 0.05$.

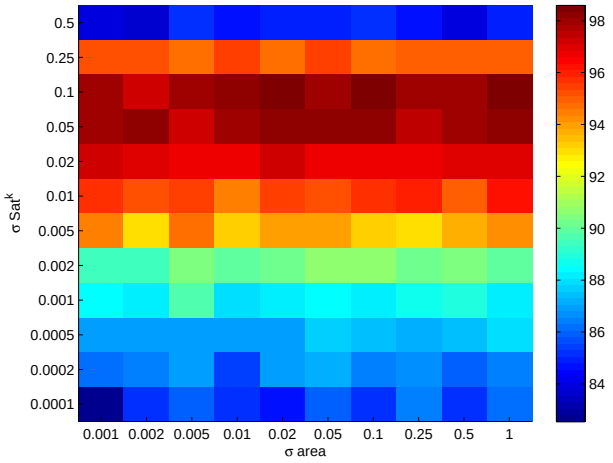


(b) Using the relative area region feature and the Sat^o measure, the best performance, 98.91%, is attained for $\sigma_{vertex} = 0.005$ and $\sigma_{edge} = 0.005$.

Figure 5.6: Categorization performance for the Sat^o measure and two different region features, using grid search in the space of kernel parameters. At this point, we model input data using strict adjacency graphs.

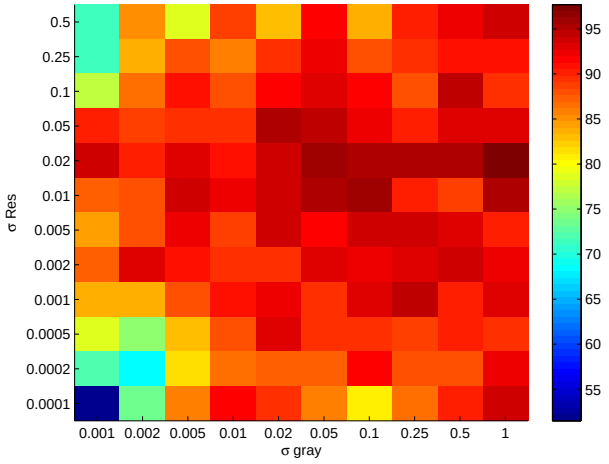


(a) Using the mean gray level region feature and the Sat^k measure, the best performance, 96.16%, is attained for $\sigma_{vertex} = 0.5$ and $\sigma_{edge} = 0.05$.

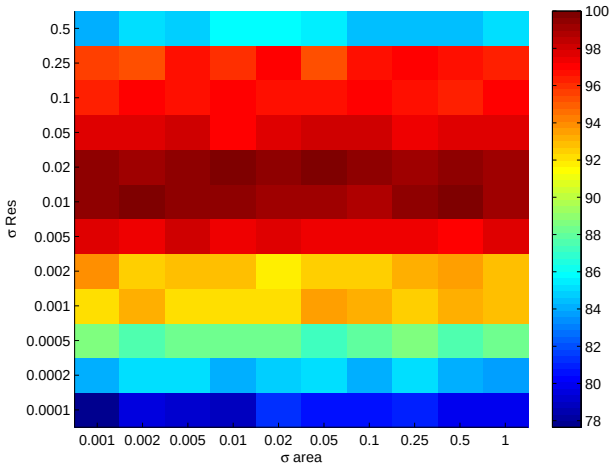


(b) Using the relative area region feature and the Sat^k measure, the best performance, 98.59%, is attained for $\sigma_{vertex} = 0.02$ and $\sigma_{edge} = 0.1$.

Figure 5.7: Categorization performance for the Sat^k measure and two different region features, using grid search in the space of kernel parameters. At this point, we model input data using strict adjacency graphs.



(a) Using the mean gray level region feature and the *Res* measure, the best performance, 97.68%, is attained for $\sigma_{vertex} = 1$ and $\sigma_{edge} = 0.02$.



(b) Using the relative area region feature and the *Res* measure, the best performance, 100.00%, is attained for $\sigma_{vertex} = 0.05$ and $\sigma_{edge} = 0.02$.

Figure 5.8: Categorization performance for the *Res* measure and two different region features, using grid search in the space of kernel parameters. At this point, we model input data using strict adjacency graphs.

5.6.4 Categorization with fuzzy adjacency structures

In order to understand better the nature of these measures, we have indexed all the region pair interactions in all the images in the dataset. Table 5.2 shows some interesting observations related to the properties of these estimations. If we consider the numerical distribution, we notice that Sat^o attains the maximal value, whilst the other two estimations are harder to satisfy. This results from the fact that small objects may be found entirely in the kernel of a reference object, but never include the whole kernel of the reference object.

Another observation that motivates the analysis of θ-graphs is that strict adjacency and a high value of the estimation of fuzzy adjacency do not imply each other (line two and three of Table 5.2). The obvious implication is that it is sometimes advisable to place edges in graph representations even in the absence of strict adjacency. The second implication is that it is equally advisable to avoid placing edges in some cases where strict adjacency is present.

Section 5.5.2 presented a method that assists decision making in such circumstances. In order to estimate the impact of different estimation thresholds θ on the structure of θ-graphs, we compute the sum of of graph edit distances between the set of strict adjacency graphs and θ-thresholded graphs with respect to the relation R and using Equation 5.9, for all the image representations in the database. The evolutions of the distances are presented in Figure 5.9, Figure 5.10 and Figure 5.11, for Sat^o,Sat^k and Res estimations respectively . The threshold $\hat{\theta}$ that would minimize globally the structural difference between G_a and $G_\theta \in G$ is given by the value corresponding to the minimum in the distance sum profiles (Table 5.2).

For our dataset, the optimal threshold for the Sat^o satisfiability measure is $\hat{\theta} = 0.934$. This high value means that most of the times, strictly adjacent regions account for satisfiability values that go beyond the threshold, as it is perceivable from the high proportion of maximum values. The next measures exhibit a different behavior; they penalize very fast the absence of a strong adjacency. In this cases, the R values associated to the strict adjacency relation are more scattered in an interval of low values, thus the optimal thresholds are situated further from the maximum values: .$\hat{\theta} = 0.101$ for Sat^k, and $\hat{\theta} = 0.091$ for Res.

5.6.5 Fuzzy measures of adjacency and frequent low values

Let us analyze next the impact of adding structural information which is not necessarily tied to the strict adjacency between image regions. We have already highlighted the fact that for two of the three estimations, high values are absent. The complementary aspect which is interesting from the statistical learning point of view (Section 5.5) concerns the frequency of null estimations, which carry relatively low information.

In order to visualize the repartition between important adjacency information and less important adjacency interactions, we present the histograms in Figures 5.12,5.13 and 5.14 that count the values of the estimations computed as in Section 5.4, between

Table 5.2: Observations related to the properties of the three measures of fuzzy adjacency on the set of all the region pairs in all the images in the dataset

Observation	Satisfiability Sat^o	Satisfiability Sat^k	Resemblance Res
$\max_{v1,v2} \mathcal{R}(v_1, v_2)$	1.000	0.629	0.386
$\max_{\text{border}(v1,v2)=0} \mathcal{R}(v_1, v_2)$	1.000	0.481	0.287
$\min_{\text{border}(v1,v2)>0} \mathcal{R}(v_1, v_2)$	6.25e-4	1.07e-4	8.60e-5
$\hat{\theta}$	0.934	0.101	0.091
$\mathcal{R}(v_1, v_2) < 0.01$	28.47 %	52.66%	55.63 %

all the region pairs in all the dataset images. These figures depict very well the gradual transition between crisp adjacency (a histogram with all the values in 0 and 1, and nothing between) and the three estimations of fuzzy adjacency.

From these figures, we notice that the first type of measure, Sat^k, takes the maximum value more often, while the frequency of low values is significant. For the second measure, Sat^o, maximum values are quite low even for adjacent regions (the maximum value in all the dataset being 0.629), and low values are even more frequent. Finally, the third type of measure, Res, exhibits values which are even more concentrated in the lower part of the support, the maximum value being 0.386. The exact frequency of very low values ($\mathcal{R}(v_1, v_2) < 0.01$) is also presented in Table 5.2, for each of the three measures. In each case a high proportion of low estimations of fuzzy adjacency is concentrated in this part of the distribution.

In Tables 5.3, 5.4 and 5.5 we compare the categorization performances for different settings involving spatial relations. As a reference, we use the best classifier detected for a certain region feature-spatial relation pair, using grid search. This classifier relies on the strict adjacency graph extracted from the image, but the edges are labeled using the spatial relation value between the corresponding vertices. The performance of this classifier is recalled in the fourth column of the graph. The interest of incorporating spatial relation information to the edge labeling is proven in the third column by the weak performance of the classifier on the adjacency graph which uses only the region feature information (the edge kernel k_e being fixed set as $k_e = 1$).

Next, we pass to the threshold graphs G_θ in the set G. In our setting, the spatial relations $\mathcal{R}$ are represented using values in $[0, 1]$, therefore the threshold θ is also a number in $[0, 1]$. We estimate the categorization performance along the set G in specific points adapted to the value distribution of $\mathcal{R}$, such as the complete graph $G_f|\theta = 0$, $G_{\hat{\theta}}$ and values in between. We repeat the same experience for some suboptimal kernels for the image features, in order to observe the variation with θ in this kind of points too.

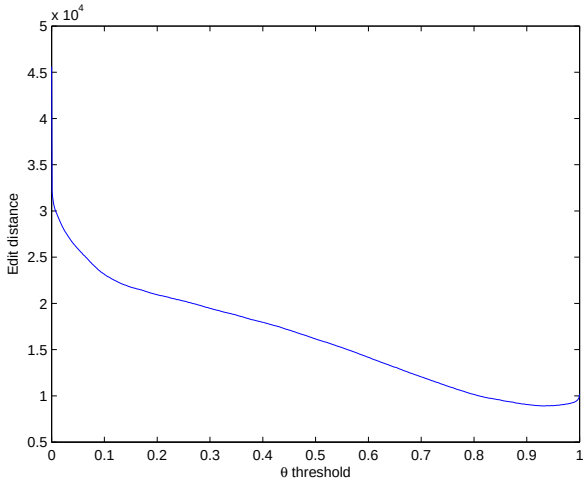


Figure 5.9: For a given Sat^o threshold θ, we show the number of different edges between the set of strict adjacency graphs of the dataset and the set of θ-threshold graphs associated to the same images. The minimal value accounts for the highest overall structural similarity between the strict adjacency graphs and the θ-threshold graphs.

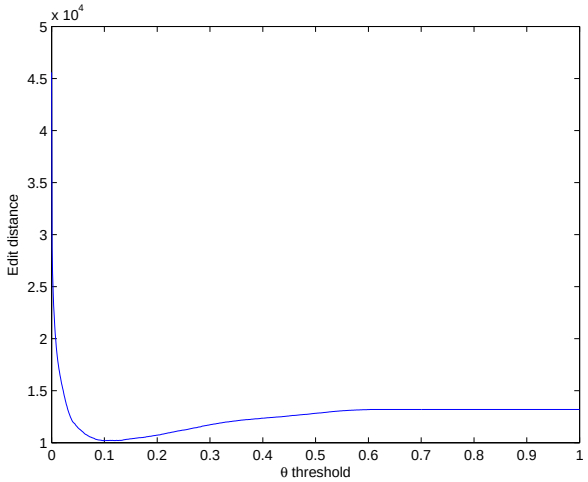


Figure 5.10: For a given Sat^k threshold θ, we show the number of different edges between the set of strict adjacency graphs of the dataset and the set of θ-threshold graphs associated to the same images. The minimal value accounts for the highest overall structural similarity between the strict adjacency graphs and the θ-threshold graphs.

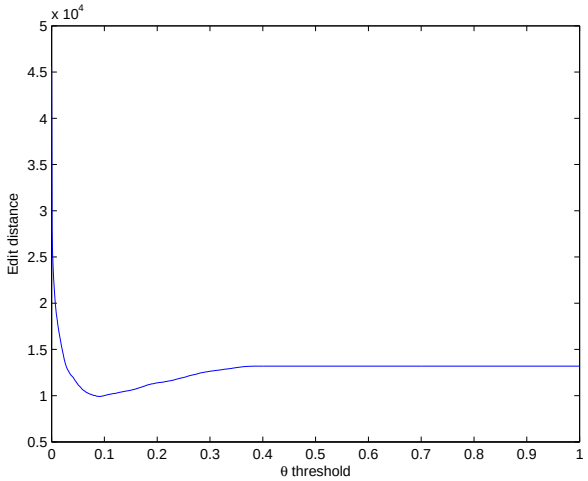


Figure 5.11: For a given *Res* threshold θ, we show the number of different edges between the set of strict adjacency graphs of the dataset and the set of θ-threshold graphs associated to the same images. The minimal value accounts for the highest overall structural similarity between the strict adjacency graphs and the θ-threshold graphs.

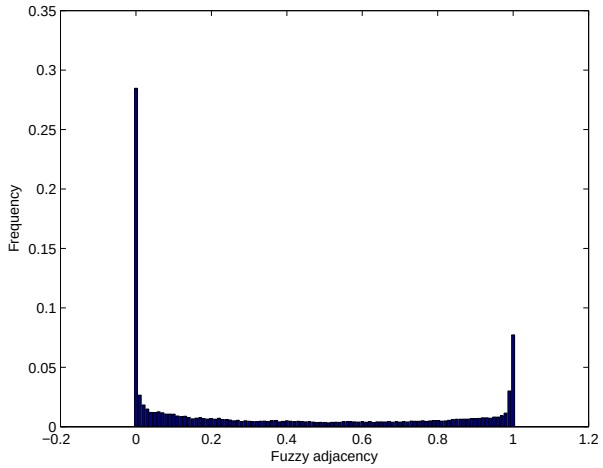


Figure 5.12: Histogram of the satisfiability Sat^o measure of fuzzy adjacency. Null values are the most frequent ones, but maximal values are frequent too.

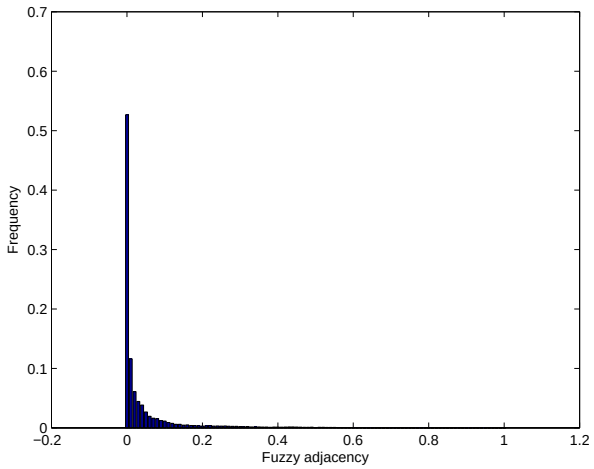


Figure 5.13: Histogram of the satisfiability Sat^k measure. Much lower values are observed, as this measure estimates the presence of the second object over a circular neighborhood of the reference region.The maximum value in all the dataset is 0.629.

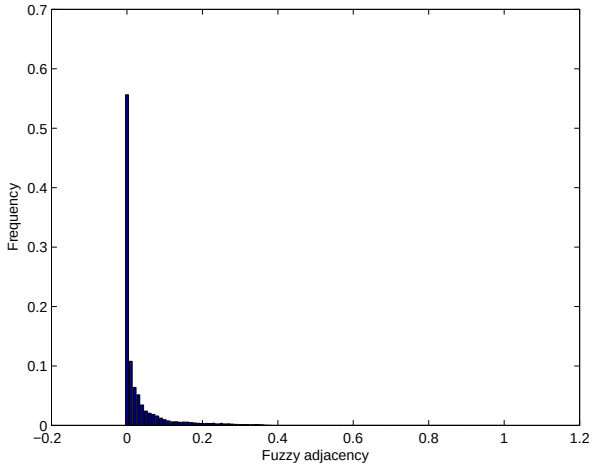


Figure 5.14: Histogram of the resemblance *Res* measure. Null values are equally the most frequent ones, but this measure penalizes very fast the absence of a strong adjacency and the maximum value in all the dataset is 0.386.

5.7 Discussion

In the first place, the benefits of enriching the strict adjacent graph structure with fuzzy spatial labels are clearly visible from the discrepancy between the third and the fourth column in all the final tables. Performing a grid search in the feature kernel space for a region feature and a spatial relation is a tedious process but the resulting parameters (σ_v^*, σ_e^*) guarantee a high performance.

From this point on, we may notice that the spatial relation Sat^o does not cope well with the transition to thresholded labels. For multiple values of θ that are chosen as to take out null estimation edges, or to select only high estimation edges, the performances of the classifiers are not comparable with the previous case, which was based on the enriched adjacency graph.

For the other two estimations, overall results show that once we leave G_a and pass to structures G_θ which are based entirely on thresholded fuzzy spatial relations, classification results are comparable with the best performance witnessed on the strict adjacency graph structure with the optimal kernels provided by (σ_v^*, σ_e^*). However, we do improve systematically the classification results using G_θ if the parameters (σ_v, σ_e) are suboptimal. This means that by using an adapted threshold θ we can decrease the grid search size and still obtain good results while experiencing a noticeable speed gain.

In order to find the adapted threshold θ which is specific to each $\mathcal{R}$, we notice that if $\theta > \hat{\theta}$, performance is systematically reduced; this means that pruning edges beyond this point has a clear harmful result on the image representation. Adding edges to $G_{\hat{\theta}}$ however gradually increases the performance, up to a point when G_θ is very close to becoming G_f, when performance falls drastically. In our results, this is visible for the threshold $\theta = 0.01$, which rejects a high proportion of low information edges that would otherwise flood the graph classifier.

5.8 Conclusion

In this part of our work, we studied the benefits offered by image representations using labeled graphical models, as well as by employing fuzzy descriptors for spatial information. Graphical models allow for a flexible integration between intrinsic visual features of image parts and the spatial interactions taking place between image parts. We showed that fuzzy information is highly beneficial for the learning process when we use it to enrich the labeling of strict adjacency graphical structures, but that if we extend the edge set beyond strict adjacency, loose spatial interactions may screen more relevant spatial information and that generic kernel functions are not well adapted to take into account the entirety of spatial relations within images. A solution that we propose in order to filter low information in edge labeling is to prune the edge set using thresholds that are adapted to the nature of each spatial relation estimation. Results show that integrating fuzzy spatial information is beneficial for image interpretation, and that thresholding the spatial values correctly is an efficient method for filtering out

less important spatial information.

Table 5.3: Categorization performance for (region feature - Sat^o) image information. The parameters for the kernel functions are the optimal values found using the grid search, and a pair of suboptimal values. In the third column, the strict adjacency graph is used, but no spatial relation labels are added to the graph. In the fourth column, we use fuzzy adjacency labeling on the strict adjacency graph. Afterwards, we use different θ-threshold fuzzy adjacency graphs.

Region feature	Spatial relation	Strict adj. No relation	Strict adj. Fuzzy labeling	Fuzzy θ-graphs
Mean gray (σ=1.0)	Sat^o Satisfiability (σ=0.05)	53.76%	99.00%	θ=0.00 (75.91%) θ=0.05 (78.72%) θ=0.20 (73.22%) $\hat{\theta}$=0.93 (73.17%) θ=0.99 (75.89%)
Mean gray (σ=0.05)	Sat^o Satisfiability (σ=0.05)	51.93%	90.45%	θ=0.00 (57.94%) θ=0.05 (68.85%) θ=0.20 (69.67%) $\hat{\theta}$=0.93 (63.38%) θ=0.99 (64.23%)
Rel. area (σ=0.005)	Sat^o Satisfiability (σ=0.005)	51.07%	98.91%	θ=0.00 (78.47%) θ=0.05 (82.21%) θ=0.20 (72.12%) $\hat{\theta}$=0.93 (77.38%) θ=0.99 (81.64%)
Rel. area (σ=0.005)	Sat^o Satisfiability (σ=0.5)	51.07%	92.54%	θ=0.00 (82.36%) θ=0.05 (85.66%) θ=0.20 (84.94%) $\hat{\theta}$=0.93 (74.45%) θ=0.99 (79.71%)

Table 5.4: Categorization performance for (region feature - Sat^k) image information. The parameters for the kernel functions are the optimal values found using the grid search, and a pair of suboptimal values. In the third column, the strict adjacency graph is used, but no spatial relation labels are added to the graph. In the fourth column, we use fuzzy adjacency labeling on the strict adjacency graph. Afterwards, we use different θ-threshold fuzzy adjacency graphs.

Region feature	Spatial relation	Strict adj. No relation	Strict adj. Fuzzy labeling	Fuzzy θ-graphs
Mean gray (σ=0.5)	Sat^k Satisfiability (σ=0.05)	52.18%	96.16%	θ=0.00 (95.01%) θ=0.01 (93.81%) θ=0.02 (92.90%) $\hat{\theta}$=0.10 (88.73%) θ=0.50 (85.63%)
Mean gray (σ=0.01)	Sat^k Satisfiability (σ=0.05)	51.93%	90.49%	θ=0.00 (73.51%) θ=0.01 (90.49%) θ=0.02 (87.04%) $\hat{\theta}$=0.10 (85.74%) θ=0.50 (80.18%)
Rel. area (σ=0.02)	Sat^k Satisfiability (σ=0.1)	50.18%	98.59%	θ=0.00 (95.86%) θ=0.01 (98.14%) θ=0.02 (97.98%) $\hat{\theta}$=0.10 (97.74%) θ=0.50 (73.87%)
Rel. area (σ=0.02)	Sat^k Satisfiability (σ=0.001)	50.18%	88.11%	θ=0.00 (93.45%) θ=0.01 (97.15%) θ=0.02 (93.84%) $\hat{\theta}$=0.10 (83.66%) θ=0.50 (71.81%)

Table 5.5: Categorization performance for (region feature - *Res*) image information. The parameters for the kernel functions are the optimal values found using the grid search, and a pair of suboptimal values. In the third column, the strict adjacency graph is used, but no spatial relation labels are added to the graph. In the fourth column, we use fuzzy adjacency labeling on the strict adjacency graph. Afterwards, we use different θ-threshold fuzzy adjacency graphs.

Region feature	Spatial relation	Strict adj. No relation	Strict adj. Fuzzy labeling	Fuzzy θ-graphs
Mean gray (σ=1.00)	Resemblance (σ=0.02)	53.76%	97.68%	θ=0.00 (78.64%) θ=0.01 (94.49%) θ=0.02 (92.00%) $\hat{\theta}$=0.09 (74.43%) θ=0.30 (65.56%)
Mean gray (σ=0.25)	Resemblance (σ=0.01)	53.32%	90.06%	θ=0.00 (57.03%) θ=0.01 (95.49%) θ=0.02 (90.35%) $\hat{\theta}$=0.09 (88.55%) θ=0.30 (58.33%)
Rel. area (σ=0.05)	Resemblance (σ=0.02)	51.94%	100.00%	θ=0.00 (96.57.%) θ=0.01 (98.63%) θ=0.02 (99.81%) $\hat{\theta}$=0.09 (99.05%) θ=0.30 (73.07%)
Rel. area (σ=0.05)	Resemblance (σ=0.002)	51.94%	92.37%	θ=0.00 (95.18%) θ=0.01 (99.09%) θ=0.02 (95.45%) $\hat{\theta}$=0.09 (85.27%) θ=0.30 (58.33%)

Chapter 6

Automated Visual Feature Selection Using Multiple Kernel Learning

6.1 Motivation for multiple kernel learning

When one chooses one of the many solutions for image representation that we have mentioned in Chapter 4, it is not always clear whether a given set of features is appropriate for that particular application. This problem gets more complicated if different features generate individual representations which define independent views of the same input data. Determining the optimal way of making use simultaneously of different views of the data, and filtering out features that are useless in a specific context, is not a trivial task.

The interest for integrating heterogeneous data using multiple kernels originally arose in computational biology, as a large number of fitting kernels exist for entity representations. A general framework for integrating and drawing inferences from a collection of kernel observations has been proposed and successfully applied for membrane protein prediction (Lanckriet et al, 2003) and protein function prediction (Lanckriet et al, 2004b). Since then, the multiple kernel learning algorithms evolved and their applicability extended greatly.

If we relate specifically to our method and to the preprocessing step, images are segmented and a strict adjacency graph is built upon the resulting adjacent regions. Intrinsic region features like color or shape are computed. The only structural information retrieved from the image is the adjacency relation between regions that is implicitly stored in the graph structure by the presence of an edge between two vertices. Once the graph is built, a marginalized kernel extension relying on the features mentioned above is used to assess the similarity between two graphs and to build a classifier. However, the method is limited in the sense that only one feature or one pair (visual feature-spatial relation) at a time is used to build a classifier. This kind of situation where multiple classifiers use limited knowledge on the input data is not very rare; we can mention, among other situations, the example of hyperspectral image classification where individual

bands are used by separate learners (Tuia et al, 2009). Closer to region based methods, in Harchaoui and Bach (2007), the authors consider finite random walks on segmentation graphs and use multiple kernel learning for different random walk lengths. The visual information being considered is region color and the only spatial relation is also the adjacency between segmented regions.

6.2 Our approach using the multiple kernel learning framework

In this paper, we propose to automatically create a kernel based on multiple visual features and spatial relations. The presence of multiple features emphasizes the importance of a generic, reliable method that combines data sources in building the discriminant function of the classifier (Lanckriet et al, 2004a). As we enrich the graph by adding complex structural information retrieved from the image, such as topological relations or metric spatial relations, this raises specific methodological problems, that are addressed further, in particular by using different kernels for each type of relation and combining them under a global optimization constraint.

The framework is open to the introduction of any other feature that describes image regions or relations between them. However, we stress the importance of selecting relevant features and of finding positive definite kernels that give an intuitive similarity measure between them. The general outline of the proposed method is illustrated in Figure 6.1.

6.3 The multiple kernel learning formalism

The purpose of multiple kernel learning (MKL) is to provide a single kernel function for heterogeneous data, in this case exemplified by different types of features annotating the graph representation of an image.

For a given categorization task using a training set $\mathcal{X}$ with m elements, image information is processed in the form of a symmetric and positive definite matrix $K \in \mathbb{R}^{m \times m}$, commonly designated as the kernel matrix:

$$\mathrm{K}_{i,j} = K(x_i, x_j) \tag{6.1}$$

with $x_i, x_j \in \mathcal{X}$. The first step of the method is to build these basis kernel matrices $\{\mathrm{K}_{f_1}, \ldots, \mathrm{K}_{f_n}\}$ corresponding to kernel functions $\{K_{f_1}, \ldots, K_{f_n}\}$ of different features $\{f_1, \ldots, f_n\}$ taken into account. These matrices are basic in the way that each of them represents a narrow view of the data. For a challenging image set, categorization in such basic feature spaces might not be efficient, because a reliable discrimination is not performed using only one feature. In these cases, fusion of the information brought by

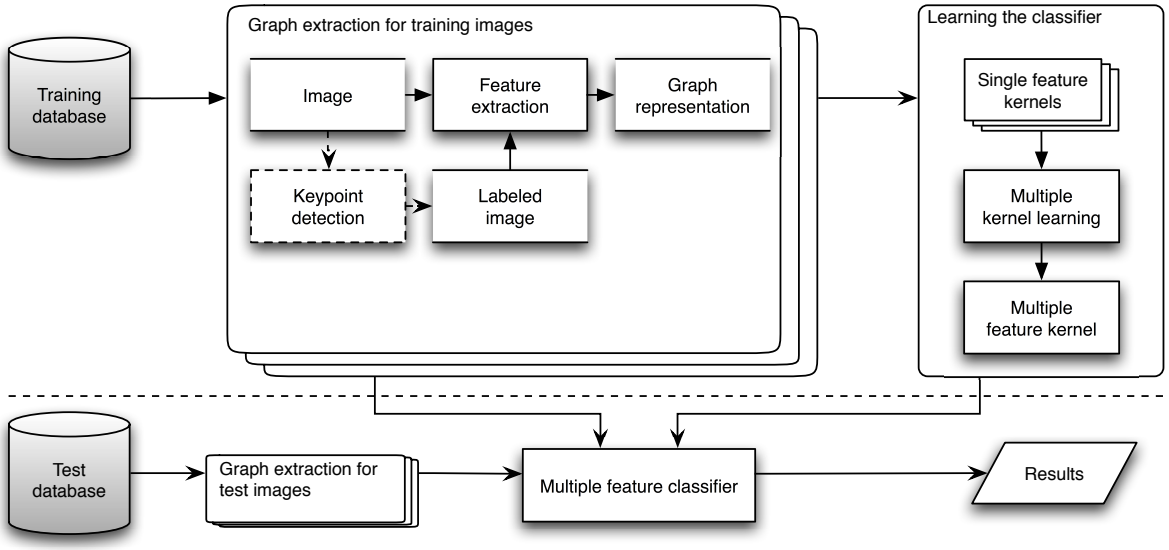


Figure 6.1: The diagram of our method. In the training part, for each image, keypoints (possibly highlighted in a detection block) along with spatial information are used to create a graph representation. Different kernel functions for individual features that label the graph representation are used to build base SVM classifiers. Finally, multiple kernel learning is used on these functions, and an optimal multiple feature kernel is determined. In the testing part, a graph is extracted from each image of the test database, and is classified by a support vector machine using the multiple feature kernel and the graph representations of training images.

each kernel is necessary. The most straightforward solution to this problem is to build a linear combination of the basis kernels:

$$\widetilde{K} = \sum_{i=1}^{n} \lambda_i K_{f_i} \tag{6.2}$$

This type of linear combination represents a compromise that allows mutual compensation among different views of the data, thus ameliorating the categorization flexibility. The methodological problem of optimally retrieving the weight vector λ has been addressed in Lanckriet et al (2004a) and consists in globally optimizing over the convex cone P of symmetric, positive definite matrices:

$$\mathrm{P} = \{X \in \mathbb{R}^{m \times m} \mid X = X^T, X \geq 0\} \tag{6.3}$$

the following SVM-like dual problem:

$$\min_{\substack{\lambda \in \mathbb{R}^n \\ K \in \mathrm{P}}} \max_{\alpha \in \mathbb{R}^m} 2\alpha^T e - \alpha^T D(y)\, \mathrm{K}\, D(y)\alpha, \text{ subject to} \tag{6.4}$$

$$C \geq \alpha \geq 0,\ \text{trace}(\mathrm{K}) = c,\ \mathrm{K} = \sum_{i=1}^{n} \lambda_i \mathrm{K}_{f_i},\ \alpha^T y = 0$$

where m is the size of the training database, $e \in \mathbb{R}^m$ is the vector the elements of which are equal to 1 and $D(y) \in \mathbb{R}^m \times \mathbb{R}^m$ is the matrix of null elements except the diagonal ones which are the labels (+1 or -1) of the training examples, $D(y)_{ii} = y_i$.

In the problem specified above, C represents the soft margin parameter, while $c \geq 0$ fixes the trace of the resulting matrix. The interest of this program is that it minimizes the cost function of the classifier with respect to both the discriminant boundary and the parameters λ_i. The output is a set of weights and a discriminant function that combines information from multiple kernel spaces.

The problem can be transposed into the following quadratically constrained quadratic program (Lanckriet et al, 2004a), the primal-dual solution of which indicates the optimal weights λ_i:

$$\min_{\alpha,t} 2\alpha^T e - ct,\ \text{subject to} \tag{6.5}$$

$$t \geq \frac{1}{\text{trace}\,(\mathrm{K}_{f_i})} \alpha^T\ D(y)\ \mathrm{K}_{f_i}\ D(y)\ \alpha, \quad i = 1, \ldots, n$$

$$C \geq \alpha \geq 0, \alpha^T y = 0$$

The drawback of this approach is that solving the problem becomes unfeasible as the number n of available kernels increases, mainly because in the SVM primal of 6.4 the regularization of the weight vector a mixed (l_2-l_1)-norm is used, instead of the classical l_2-norm. Recently, a more efficient multiple learning algorithm that solves the MKL problem through a primal formulation involving a weighted l_2-norm regularization has been proposed (Rakotomamonjy et al, 2007):

$$\min_{\lambda_k, f_k, \xi_i, b} \sum_k \frac{1}{\lambda_k} \|f_k\|^2 + C \sum_i \xi_i,\ \text{subject to} \tag{6.6}$$

$$y_i \left(\sum_k f_k(x_i) + b \right) \geq 1 - \xi_i, \sum_k \lambda_k = 1, \xi_i \geq 0, \lambda_k \geq 0$$

with $f_k \in \mathcal{H}_k$, the reproducing kernel Hilbert space (RKHS) associated with the kernel K_k. The advantage of this new formulation over the initial MKL problem introduced by Equation 6.5 is that even though the two systems are convex, only the latter is smooth due to the weighted l_2-norm regularization.

Using MKL in this form for our specific problem boils down to defining basis kernel functions for each feature, with parameters being selected according to their feature variability in the data. More precisely, the threshold for the discrimination function should roughly indicate the smallest distance between two feature values that would trigger a 0-similarity decision for an observer. This threshold is closely correlated to the feature type and equally to the data being analyzed.

For each of the basis kernels above, we build a graph kernel that will provide us with a graph similarity estimate based on a single feature of the data. Some features are more discriminative than others for a specific data set and therefore generate a better classifier. The multiple kernel learning framework presented above allows us to build a heterogenous decision function that weighs each feature based on its relative relevance in the feature set through its weight λ_i, thus providing optimal performance with the given feature kernels as inputs (Fig. 6.2).

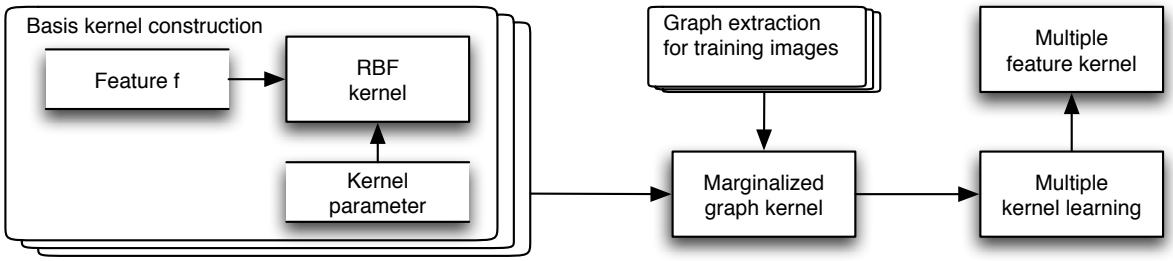


Figure 6.2: A detail of the multiple kernel learning step of our method. For each feature, RBF kernels built using different parameters are computed and used in the general marginalized graph kernel in order to get a representation of training set through the selected feature. Finally, the MKL algorithm is used and the selected basis kernels are used together in a multiple feature kernel.

6.4 MKL, structured data and relations

The standard MKL framework copes very well with the common situation, when different types of kernels belonging to a set $\mathcal{K} = \{K_1, \ldots, K_n\}$ are computed independently on a training set $\mathcal{X}$, and then the multiple kernel is computed using Equation 6.6. An implicit assumption is that the elements of the set $\mathcal{K}$ are computationally independent (each element taken individually is sufficient to get a classifier) and we can train one classifier per kernel $K_i \in \mathcal{K}$.

Our setting is slightly different and more complex; more precisely, the graph representations in $\mathcal{X}$ have independent parts: the vertex labels and the edge labels. Therefore, we have two distinct families of kernel candidates, which are combined within the graph marginalized kernel computation in Equation 3.5, simply denoted here by the function f: $K_G = f(K_v, K_e)$. We call K_G a *composite* kernel because it depends on more than one kernel function at a time. In fact, K_G presents this property because it resorts to a convolution kernel for computing path similarities in graphs, and convolution kernels are composite:

$$K(x, x') = \sum_{\substack{\{x_1,\ldots,x_D\}\in R^{-1}(x)\\ \{x'_1,\ldots,x'_D\}\in R^{-1}(x')}} \prod_{d=1}^{D} K^{(d)}(x_d, x'_d) \tag{6.7}$$

In practice, we have a large family of n hypotheses $\mathcal{K}_v$ for vertex kernels and another one $\mathcal{K}_e$ of m edge kernels. Vertices or edges may be multi-labeled and therefore variability in vertex kernels may arise as a consequence of label variability or because of parameter variability for a certain label. For example, vertex labels may encode texture, color, shape information for regions or more general visual interest areas in an image, and edge labels account for spatial information, such as distance, orientation, symmetry etc. Assuming that we consider exhaustively all the possible $n \cdot m$ pairs of kernels $(K_v^\alpha; K_e^\beta) \in (\mathcal{K}_v; \mathcal{K}_e)$ that correspond to vertex and edge features $(\alpha; \beta)$, we will get a graph kernel $K^{\alpha,\beta}$ which will be useful in detecting patterns depending only of α and β. In other words, we would assume that there are no higher order correlations between features, which is not entirely accurate.

The general MKL formulation imposes that we provide a selection family for K_G and thus, MKL is performed at graph kernel level, and not at the vertex or edge level. The drawback of this constraint is that, starting from an individual vertex or edge kernel, we are supposed to provide a graph kernel and thus to consider an implicit value for the missing factor (K_e if we analyze a vertex kernel and K_v if we analyze an edge kernel). In both cases we set the missing value to 1, being committed to discriminating with a basis kernel using one feature only. However, we are convinced that a MKL formulation that would optimize the terms K_v and K_e directly inside the function f (instead of optimizing the result of f) would perform better. To our knowledge, this problem of optimizing directly the terms of such a composite kernel has not yet been addressed, most probably because it is quite unusual that an object $x \in \mathcal{X}$ needs more than one kernel function at the same time for its substructures. Addressing this problem is an interesting perspective of future work.

6.5 Experiments and results

In order to perform the necessary tests for this chapter, we will slightly modify the setting in order to adapt to the new algorithm flow, while at the same time maintaining a reasonable experimental size. More specifically, for each of the three categorization sets defined in the previous chapter, we are going to select randomly a smaller train set of 30 images that we are going to keep unchanged in the following part of the experiment. Concerning the graph construction and labeling, we use three normalized visual features as region descriptors: the mean gray level (which is normalized according to the lightest and darkest regions in the image), the relative region area (normalized according to the total image area) and the normalized region compacity, defined as the normalized ratio between its surface and its squared perimeter. Spatial relations

being considered between image regions are the fuzzy adjacency using resemblance or satisfiability estimations.

For estimating feature similarity, we use RBF kernels with thresholds that are again adapted to the range $[0, 1]$ of these normalized features: $\sigma \in \{0.05, 0.1, 0.2, 0.4, 0.8\}$. Likewise, RBF kernels are used for estimating spatial relation similarity between pairs of regions, but since their numerical values tend to be smaller, the kernel parameter will be selected from the following set: $\sigma \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$. As mentioned above, for each classification task we use $N = 30$ training graphs and the rest as$T = 78$ test graphs, evenly divided for the two categories.

Each feature or relation kernel is integrated in a graph marginalized kernel that we use to get a matrix (Equation 6.1) which corresponds to a certain view of our training data set; this results in a set of 25 training matrices. At this point, we use SimpleMKL[1] which solves Equation 6.6 in order to find an optimal combination among the kernels that had been computed. For the regularization parameter C of the SVM that controls the trade-off between maximizing the margin and minimizing the l_1-norm of the slack vector ξ, we perform a grid-search with uniform resolution in $\log_{10}$ space: $\log_{10} C \in \{-3, \ldots, 6\}$. Our results correspond to observations reported by Rakotomamonjy et al (2007), that as the value of the regularization parameter C increases, the number of kernels taken into consideration in Eq. 6.2 increases too, and stabilizes at a certain point.

Table 6.1: Categorization error in sagittal projection for the kernels selected by MKL and for the multiple kernel, with $C = 10^3$ and $C = 10^5$.

Feature kernel (σ value)	$C = 10^3$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.10	0.31	0.21	0.38
gray level (0.20)	0.13	0.21	0.17	0.06
compacity (0.05)	0.03	0.10	0.06	0.24
area (0.05)	0.00	0.18	0.09	0.32
MK	0.03	0.03	0.03	-
Feature kernel (σ value)	$C = 10^3$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.08	0.28	0.18	0.38
gray level (0.20)	0.03	0.21	0.12	0.06
compacity (0.05)	0.00	0.10	0.05	0.24
area (0.05)	0.00	0.21	0.10	0.32
MK	0.03	0.03	0.03	-

[1]SimpleMKL, a Matlab multiple kernel learning toolbox, available at http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html

Table 6.2: Categorization error in coronal projection for the kernels selected by MKL and for the multiple kernel, with $C = 10^3$ and $C = 10^5$.

Feature kernel (σ value)	$C = 10^3$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.38	0.21	0.29	0.28
compacity (0.05)	0.23	0.18	0.21	0.48
area (0.05)	0.26	0.03	0.14	0.24
MK	0.15	0.10	0.13	-
Feature kernel (σ value)	$C = 10^5$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.33	0.28	0.31	0.36
compacity (0.05)	0.13	0.21	0.17	0.40
area (0.05)	0.33	0.03	0.18	0.24
MK	0.18	0.03	0.10	-

Finally, for a given parameter C, the effectiveness of individual feature or relation based kernels and of the multiple kernel is tested in their corresponding SVM classifiers; higher values of C correspond to a higher performance, therefore we present below results for a saturated value $C = 10^5$ along with those for C corresponding to the transition range of the kernel weight vector λ, $C = 10^3$. For the sake of brevity, among the estimations for the 25 kernels being considered each time, we will present those for the individual kernels that have been selected by the MKL algorithm.

For each of the three different projections that correspond to three different categorization tasks and for each of the feature kernels selected by MKL, we present the global categorization error ,the individual error for each of the two classes, and the feature kernel weight in the multiple kernel. Finally the performance of the multiple kernel is presented.

Discussion In most cases, results show an amelioration of the performance compared to the initial categorization rates, thus proving the interest of the multiple kernel learning approach for image kernels. Tables 6.1, 6.2 and 6.3 present the performance evaluation for the sagittal, coronal and axial projections respectively.

In sagittal projection, individual kernels that have been selected by the MKL algorithm have an overall good performance, and the multiple kernel succeeds in obtaining a clear amelioration over its constituents. A noticeable behavior, which is also common to the other two categorization tasks, is that kernels built upon small, discriminative σ values perform better and are also selected in the multiple kernel equation. However, in the case of the sagittal projection categorization, two different kernels are selected for the same visual feature; this is an improvement over the typical parameter grid search,

Table 6.3: Categorization error in axial projection for the kernels selected by MKL and for the multiple kernel, with $C = 10^3$ and $C = 10^5$.

Feature kernel (σ value)	$C = 10^3$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.41	0.26	0.33	0.49
compacity (0.05)	0.51	0.18	0.35	0.38
area (0.05)	0.28	0.26	0.27	-
resemblance (0.01)	0.05	0.03	0.04	0.12
MK	0.31	0.05	0.18	-
Feature kernel (σ value)	$C = 10^3$			
	Cat.1 error	Cat.2 error	Global error	λ
gray level (0.05)	0.49	0.28	0.38	0.35
compacity (0.05)	0.38	0.13	0.26	0.23
area (0.05)	0.08	0.08	0.08	0.24
resemblance (0.01)	0.05	0.00	0.03	0.18
MK	0.31	0.00	0.15	-

where different parameter values use better some category elements at the disadvantage of others and where we have to make a choice taking into account the global performance over the training set.

The multiple kernel does not guarantee the minimum of the test error over individual kernels (for example, in Table 6.1, Cat. 1 error for the area kernel is inferior to the MK error), but its success is represented by the potential amelioration of the test error over all the individual kernels, as a result of simultaneously using different types of visual features (Cat. 2 error in the same table).

For the coronal projection, basis kernels are overall less successful due to more subtle differences between the two categories, but still the multiple kernel uses the complementary views on data successfully and ameliorates the categorization performance. The axial projection problem is the most difficult and visual features alone discriminate weakly between the two categories; however, in this case a very good performance is exhibited by the spatial relation of resemblance. The multiple kernel performance is noticeably improved by the spatial information, especially in the detection of the second category, but overall it does not get close to the results acquired using the resemblance relation. We assume that in these particular situations, a MKL approach that would take into account the structure of a composite kernel as the graph marginalized kernel would use more efficiently spatial relation and visual feature information altogether.

6.6 Conclusion

We have described a graph probabilistic model for image representation and categorization, which provides a flexible method to encode image keypoints and their visual properties along with spatial information concerning keypoint relative localization. This model does not impose a certain semantic complexity of keypoints nor a particular choice of spatial relations, thus generalizing strict structured representations and those disregarding spatial information completely. A method for image categorization based on marginalized kernels on these graph models has been proposed. The key step in our approach is to simultaneously use visual features and adapted spatial information as to maximize the discriminative capacity of our categorization algorithm. In particular, we show that a graph representation of the image, enriched with numerical attributes characterizing both the image keypoints and the spatial relations between them, associated with a multiple kernel learning of these features and relations, leads to improved performance. Kernels are created for each feature and relation, and image categorization is performed using multiple weighted views of the data, in which view weights are automatically estimated so as to give more importance to the most relevant features and relations. We have shown that this method is successful and that learning from complementary representations of data can ameliorate the categorization performance.

Chapter 7

Conclusion and perspectives

The work that has been performed is aimed to propose a general and modular framework of image interpretation based on structured image representations and fuzzy spatial information. Even though we have chosen to illustrate our method using medical applications, the method is not fundamentally tied to a specific type of input images. The modular aspect of our method resides in the versatility of the underlying graphical model, which ensures the independence between the interest points that we choose to represent as vertices, and the spatial relations that we choose to encode as edges.

In the context of this solution for image representation, we have started our work by presenting a marginalized graph kernel which extends the one being used in computational biology, and which allows the interpretation of image-based graphs. We have applied this approach for a simple medical image classification task, based on a generic segmentation method. The results validated our approach and laid the foundations for taking into account more complex spatial information, in the form of estimations of fuzzy adjacency between regions that go beyond planar adjacency and that are essential for expressing image content comprehensively. At the same time, we have shown that taking into account different region features within the same image classifier is beneficial for performance.

Afterwards, we proposed the use of a series of fuzzy spatial descriptors that are related to topological relations between image regions and we studied the benefits offered by image representations using this additional type of information. We showed that understanding the distribution of the information gain of these descriptors is important for a better integration with the rest of the image model, and that adding spatial information, which implies changing the graphical structures, may influence to a considerable degree the performance of the learning algorithm.

We also presented a method that may be used for these types of graphical models in order to automatically select and to merge, among a set of visual features and spatial relations, the information which gives the best discriminative performance in a specific context. This is also a major advantage that accounts for the adaptability of our framework for different types of interpretation tasks.

Main contributions The main contributions related to our work might be summarized along the following directions:

- the proposition of fuzzy graphical models as a compromise between the bag-of-feature image representations and the constellation model based representations used in image interpretation,
- the use of marginalized graph kernels for image based graphical models,
- the inclusion of fuzzy spatial information into the region based graphical model, and the study of its impact on discriminative learning,
- optimal automatic selection of multiple features for the marginalized graph kernel.

The central direction of research has been motivated by the current gap between the importance of spatial information in image interpretation, and its scarce presence in the models proposed in the litterature. The method that we propose is modular and adaptable, and even if it lacks the speed of typical bag-of-features methods, the cost that one pays for the increase in structural complexity is not prohibitive. Moreover, an advantage of our kernel based method is that the critical computations required by the classifier may still be executed at the same time on different devices, making it adapted for distributed computing environments.

In the context of structure learning for image interpretation, a major shortcoming of support vector machines, related to the interpretability of the learned model, is partly alleviated by the multiple kernel learning framework. Using MKL output, one may understand better which features are effective and which correlations between the different views of the data are discriminative. Our method makes use of this advantage of MKL framework over plain support vector machines.

The main difficulty that we have clearly identified with our method is the requirement for a visual structured input which is appropriate for high level interpretation. Failure to do so has two negative impacts. Firstly, in the presence of preprocessing errors, input graphs are usually much larger than needed (e.g. meaningless oversegmentation), thus failing to remain within acceptable dimensions, from a computational point of view. Secondly, in the absence of graph vertices being genuinely associated to image keypoints, our method has a limited usability.

Perspectives There are many different perspectives that are highlighted by our work, since each part of our learning framework may be improved and extended independently. There are however four special directions that are probably very interesting to explore further.

One promising perspective is related to the interest points that are associated to vertices. The nature of the interest points is theoretically restricted only by the computational limitations of our method (relatively high polynomial complexity in the number of vertices). However, the semantic complexity of interest points determines directly the

semantic level that the learning framework addresses. In the case of regions, the semantic level is higher than that of bag-of-features approaches, but the main disadvantage is that the quality of the required preprocessing task (in this case, image segmentation) must be elevated too. Other region based descriptors, especially the ones related to pre-attentive mechanisms, are less complex semantically and more robust to rough preprocessing. If these descriptors were to be employed instead of segmented regions, the main challenge would be probably to detect the most meaningful descriptor set for a given image, while keeping the set size low enough for reasonable computation time. In this case, a secondary perspective of further research is represented by fuzzy spatial relations that are better adapted for other types of interest points.

Another interesting perspective is related to the formalization of the information gain in fuzzy spatial relations. Using fuzzy adjacency measures, we have proven that adjacency information may take more or less informative values. The distribution of the most useful values is specific to the meaning of each spatial relation. Knowledge regarding this distribution within structured entities allows for a more expressive spatial labeling of entity graphical models, which in discriminative learning is directly related to absence of uniformity, and therefore discriminative fitness.

Certainly, a reasoning framework using region based keypoints that are linked by complex spatial relations presents a greater potential for infering with semantic concepts, but it is inherently sensitive to the structural quality of the input (e.g. manual segmentation). One promising perspective is to associate this method with low-level oriented CBIR concepts, in order to "close the loop", or to perform iteratively low level and high level image processing tasks that are mutually beneficial.

Finally, we highlighted the fact that learning with heterogenuos data of structured visual representation might benefit from specific adaptations of multiple kernel learning algorithms.

Bibliography

Aldea E, Atif J, Bloch I (2007) Image Classification using Marginalized Kernels for Graphs. In: Graph based Representations in Pattern Recognition, Alicante, Spain, pp 103–113

Anthony M, Bartlett P (1999) Neural network learning : theoretical foundations. Cambridge University Press

Arivazhagan S, Ganesan L, Priyal SP (2006) Texture classification using gabor wavelets based rotation invariant features. Pattern Recognition Letters 27(16):1976–1982

Aronszajn N (1950) Theory of reproducing kernels. American Mathematical Sociey 68(3):337–404

Bar Hillel A, Weinshall D (2008) Efficient learning of relational object class models. International Journal of Computer Vision 77(1-3):175–198

Bar Hillel A, Hertz T, Weinshall D (2005) Object class recognition by boosting a part-based model. In: Computer Vision and Pattern Recognition, pp I: 702–709

Barnard S, Fischler M (1987) Computational and biological models of stereo vision. Image Understanding Workshop 90:439–448

Barrett R, Berry M, Chan TF, Demmel J, Donato JM, Dongarra J, Eijkhout V, Pozo R, Romine C, der Vorst HV (1994) Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. Philadalphia: Society for Industrial and Applied Mathematics.

Barron JL, Fleet DJ, Beauchemin SS (1994) Performance of optical flow techniques. International Journal of Computer Vision 12(1):43–77

Bay H, Tuytelaars T, Gool LJV (2006) Surf: Speeded up robust features. In: European Conference on Computer Vision, pp 404–417

Beaulieu JM, Goldberg M (1989) Hierarchy in picture segmentation: A stepwise optimization approach. Pattern Analysis and Machine Intelligence 11(2):150–163

Berg C (1984) Harmonic Analysis on Semigroups (Theory of Positive Definite and Related Functions). Springer

Bernardino A, Santos Victor J (2006) Fast iir isotropic 2-d complex gabor filters with boundary initialization. Image Processing 15(11):3338–3348

Beucher S, Meyer F (1993) The morphological approach to segmentation: The watershed transformation. Mathematical Morphology in Image Processing pp 433–482

Bishop CM (1996) Neural Networks for Pattern Recognition, 1st edn. Oxford University Press, USA

Bishop CM (2007) Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. Springer

Black M, Sapiro G, Marimont D, Heeger D (1998) Robust anisotropic diffusion. Image Processing 7(3):421–432

Bloch I (2005) Fuzzy Spatial Relationships for Image Processing and Interpretation: A Review. Image and Vision Computing 23(2):89–110

Bloch I, Ralescu A (2003) Directional Relative Position between Objects in Image Processing: A Comparison between Fuzzy Approaches. Pattern Recognition 36:1563–1582

Bloch I, Maître H, Anvar M (1997) Fuzzy adjacency between image objects. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 5(6):615–654

Borgwardt K, Vishwanathan S, Schraudolph N, Kriegel HP (2005) Protein function prediction via faster graph kernels. In: NIPS Bioinformatics Workshop

Borgwardt KM, Kriegel HP (2007) Graph kernels for disease outcome prediction from protein-protein interaction networks. In: Pacific Symposium on Biocomputing, pp 4–15

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Computational Learning Theory, ACM Press, pp 144–152

Bouchon-Meunier B, Rifqi M, Bothorel S (1996) Towards general measures of comparison of objects. Fuzzy Sets and Systems 84(2):143–153

Brodatz P (1968) Textures: A Photographic Album for Artists and Designers. Reinhold, New York

Brun L, Mokhtari M, Meyer F (2005) Hierarchical watersheds within the combinatorial pyramid framework. In: Discrete Geometry for Computer Imagery, Springer, pp 34–44

Caelli T, Reye D (1993) On the classification of image regions by colour, texture, and shape. Pattern Recognition 26(4):461–470

Canny J (1986) A computational approach to edge detection. Pattern Analysis and Machine Intelligence 8:679–698

Carbonetto P, de Freitas N, Barnard K (2004) A statistical model for general contextual object recognition

Carbonetto P, Dorko G, Schmid C, Kuck H, de Freitas N (2008) Learning to recognize objects with little supervision. International Journal of Computer Vision 77(1-3):219–237

Carson C, Belongie S, Greenspan H, Malik J (2002) Blobworld: Image segmentation using expectation-maximization and its application to image querying. Pattern Analysis and Machine Intelligence 24(8):1026–1038

Chapelle O, Haffner P, Vapnik V (1999) Svms for histogram-based image classification. In: IEEE Transactions on Neural Networks, special issue on Support Vectors

Chen PH, Fan RE, Lin CJ (2006) A study on smo-type decomposition methods for support vector machines. IEEE Transactions on Neural Networks 17(4):893–908

Chen Y, Wang JZ (2004) Image categorization by learning and reasoning with regions. Journal of Machine Learning Research 5:913–939

Choi H, Baraniuk R (1999) Image segmentation using wavelet-domain classification. In: SPIE Conference on Mathematical Modeling, Bayesian Estimation, and Inverse Problem

Collins M, Duffy N (2001) Convolution kernels for natural language. In: Advances in Neural Information Processing Systems 14, MIT Press, pp 625–632

Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence 24(5):603–619

Cortes C, Vapnik V (1995) Support-vector networks. In: Machine Learning, pp 273–297

Cour T, Shi J (2007) Recognizing objects by piecing together the segmentation puzzle. In: Computer Vision and Pattern Recognition, pp 1–8

Cox DR, Snell EJ (1989) Analysis of Binary Data. Chapman and Hall, London, 2nd edition

Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press

Cross G, Jain A (1983) Markov random field texture models. Pattern Analysis and Machine Intelligence 5(1):25–39

Dance C, Willamowski J, Fan L, Bray C, Csurka G (2004) Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision

Datta R, Joshi D, Li J, James, Wang Z (2006) Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 39:2007

Deruyver A, Hodé Y, Brun L (2009) Image interpretation with a conceptual graph: Labeling over-segmented images and detection of unexpected objects. Artificial Intelligence 173(14):1245–1265

Devroye L (1982) Any discrimination rule can have an arbitrarily bad probability of error for finite sample size. Pattern Analysis and Machine Intelligence 4(4):154–157

Dubois D, Prade H (1980) Fuzzy sets and systems - Theory and applications. Academic press, New York

Duda R, Hart P, Stork D (2001) Pattern Classification. Wiley

Duda RO, Hart PE, Stork DG (2000) Pattern Classification. Wiley-Interscience Publication

Eakins J (2002) Towards intelligent image retrieval. Pattern Recognition 35(1):3–14

Eakins JP (1993) Design criteria for a shape retrieval system. Computers in Industry 21(2):167–184

Eakins JP (1996) Automatic image content retrieval - are we getting anywhere? In: De Montfort University, Milton Keynes, pp 123–135

Emms D, Hancock ER, Wilson RC (2007) Graph similarity using interfering quantum walks. In: Computer Analysis of Images and Patterns, pp 823–831

Fan RE, Chen PH, Lin CJ (2005) Working set selection using second order information for training support vector machines. Journal of Machine Learning Research 6:1889–1918

Fei-Fei L, Fergus R, Perona P (2003) A bayesian approach to unsupervised one-shot learning of object categories. In: International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA

Fergus R, Perona P, Zisserman A (2003) Object class recognition by unsupervised scale-invariant learning

Fergus R, Perona P, Zisserman A, Fergus R, Perona P, Zisserman A (2005) A sparse object category model for efficient learning and exhaustive recognition. In: Computer Vision and Pattern Recognition, pp I: 380–387

Fix E, Hodges J (1951) Discriminatory analysis, nonparametric classifications. Tech. rep., USAF School Aviation Medical Report

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1):119–139

Gaertner T, Flach P, Wrobel S (2003) On graph kernels: Hardness results and efficient alternatives. In: Computational Learning Theory, pp 129–143

Gagaudakis G, Rosin P (2002) Incorporating shape into histograms for cbir. Pattern Recognition 35(1):81–91

Gagaudakis G, Rosin P (2003) Shape measures for image retrieval. Pattern RecognitionL 24(15):2711–2721

Gardiner JD, Laub AJ, Amato JJ, Moler CB (1992) Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. Transactions on Mathematical Software 18(2):223–231

Genton MG (2002) Classes of kernels for machine learning: a statistics perspective. Journal of Machine Learning Research 2:299–312

Gori M, Maggini M, Sarti L (2005) Exact and approximate graph matching using random walks. Pattern Analysis and Machine Intelligence 27(7):1100–1111

Gosselin PH, Cord M, Philipp-Foliguet S (2007a) Kernels on bags for multi-object database retrieval. In: CIVR, pp 226–231

Gosselin PH, Cord M, Philipp-Foliguet S (2007b) Kernels on bags of fuzzy regions for fast object retrieval. In: ICIP, pp 177–180

Gudivada VN, Raghavan VV (1995) Content-based image retrieval systems - guest editors' introduction. Computer 28(9):18–22

Györfi L, Kohler M, Krzyzak A, Walk H (2002) A Distribution-Free Theory of Nonparametric Regression. Springer

Harchaoui Z, Bach F (2007) Image classification with segmentation graph kernels. In: Computer Vision and Pattern Recognition, pp 1–8

Haris K, Estradiadis SN, Maglaveras N, Katsaggelos AK (1998) Hybrid image segmentation using watersheds and fast region merging. Image Processing 7(12):1684–1699

Harris C, Stephens M (1988) A combined corner and edge detector. In: Alvey Vision Conference, pp 147–152

Haussler D (1999) Convolution kernels on discrete structures. Tech. rep., UCSC

Helma C, King RD, Kramer S, Srinivasan A (2001) The predictive toxicology challenge 2000-2001. Bioinformatics 17(1):107–108

Hildreth E, Marr D (1979) Theory of edge detection. In: MIT AI Memo

Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. Journal of Molecular Biology 233(1):123–138

Horn B, Schunck B (1981) Determining optical flow. Artificial Intelligence 17(1-3):185–203

Hu MK (1962) Visual pattern recognition by moment invariants. IRE Transactions on Information Theory IT-8:179–187

Hudelot C, Atif J, Bloch I (2008) Fuzzy spatial relation ontology for image interpretation. Fuzzy Sets and Systems 159(15):1929–1951

Imrich W, Klavzar S (2000) Product Graphs: Structure and Recognition. Wiley

Kadir T, Brady M (2001) Saliency, scale and image description. International Journal of Computer Vision 45(2):83–105

Karu K, Jain A, Bolle R (1996) Is there any texture in the image? Pattern Recognition 29(9):1437–1446

Kashima H, Tsuda K, Inokuchi A (2003) Marginalized kernels between labeled graphs. In: International Conference on Machine Learning, pp 321–328

Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2001) Improvements to platt's smo algorithm for svm classifier design. Neural Computing 13(3):637–649

Kim K, Jung K, Park S, Kim H (2002) Support vector machines for texture classification. Pattern Analysis and Machine Intelligence 24(11):1542–1550

Kimeldorf G, Wahba G (1971) Some results on tchebycheffian spline function. Journal of Mathematical Analysis and Applications 33(1):82–95

Kuipers B (1978) Modeling spatial knowledge. Cognitive Science 2:129–153

Lan T, Kao D, Chiang M, Sabharwal A (2009) An axiomatic theory of fairness. Computing Research Repository abs/0906.0557

Lanckriet G, Cristianini N, Bartlett P, El Ghaoui L, Jordan M (2004a) Learning the Kernel Matrix with Semidefinite Programming. Journal of Machine Learning Research 5:27–72

Lanckriet GRG, De Bie T, Cristianini N, Jordan MI, Stafford Noble W (2003) A framework for genomic data fusion and its application to membrane protein prediction. Tech. Rep. UCB/CSD-03-1273, EECS Department, University of California, Berkeley

Lanckriet GRG, Deng M, Cristianini N, Jordan MI, Noble WS (2004b) Kernel-based data fusion and its application to protein function prediction in yeast. In: Pacific Symposium on Biocomputing, pp 300–311

Lazebnik S, Schmid C, Ponce J (2004) Semi-local affine parts for object recognition. In: British Machine Vision Conference, pp 959–968

Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, pp II: 2169–2178

Lebrun J, Philipp-Foliguet S, Gosselin PH (2008) Image retrieval with graph kernel on regions. In: International Conference on Pattern Recognition, pp 1–4

Lehmann EL, Casella G (2003) Theory of Point Estimation (Springer Texts in Statistics). Springer

Leslie CS, Kuang R (2003) Fast kernels for inexact string matching. In: Computational Learning Theory, pp 114–128

Leslie CS, Eskin E, Noble WS (2002) The spectrum kernel: A string kernel for svm protein classification. In: Pacific Symposium on Biocomputing, pp 566–575

Leslie CS, Eskin E, Cohen A, Weston J, Noble WS (2004) Mismatch string kernels for discriminative protein classification. Bioinformatics 20(4)

Levina E, Bickel P (2001) The earth mover's distance is the mallows distance: Some insights from statistics. In: International Conference on Computer Vision, pp II: 251–256

Li J, Allinson NM (2008) A comprehensive review of current local features for computer vision. Neurocomputing 71(10-12):1771–1787

Lin C, Chen R, Chan Y (2009) A smart content-based image retrieval system based on color and texture feature. Image and Vision Computing 27(6):658–665

Liu Y, Zhang D, Lu G, Ma WY (2007) A survey of content-based image retrieval with high-level semantics. Pattern Recognition 40(1):262–282

Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins CJCH (2002) Text classification using string kernels. Journal of Machine Learning Research 2:419–444

Lowe D (1999) Object recognition from local scale-invariant features. In: International Conference on Computer Vision, pp 1150–1157

Lowe D (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110

Lowe DG (2001) Local feature view clustering for 3d object recognition. Computer Vision and Pattern Recognition I:682–688

Mahé P, Ueda N, Akutsu T, Perret JL, Vert JP (2004) Extensions of marginalized graph kernels. In: Greiner R, Schuurmans D (eds) International Conference on Machine Learning), ACM Press, pp 552–559

Mahé P, Ralaivola L, Stoven V, Vert JP (2006) The pharmacophore kernel for virtual screening with support vector machines. Journal of Chemical Information and Modeling 46(5):2003–2014

Manning CD, Schütze H (1999) Foundations of Statistical Natural Language Processing. The MIT Press

Marszałek M (2008) Past the limits of bag-of-features. PhD thesis, Institut Polytechnique de Grenoble

Matas J, Chum O, Urban M, Pajdla T (2002) Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference

Meyer F (2001) Hierarchies of partitions and morphological segmentation. In: Scale-Space, pp 161–182

Mikolajczyk K, Schmid C (2004) Scale and affine invariant interest point detectors. International Journal of Computer Vision 60(1):63–86

Mikolajczyk K, Schmid C, Zisserman A (2004) Human detection based on a probabilistic assembly of robust part detectors. In: European Conference on Computer Vision, pp Vol I: 69–82

Miyajima K, Ralescu A (1994) Spatial organization in 2D segmented images: representation and recognition of primitive spatial relations. Fuzzy Sets and Systems 65(2-3):225–236

Mokhtarian F, Suomela R (1998) Robust image corner detection through curvature scale space. Pattern Analysis and Machine Intelligence 20(12):1376–1381

Mukundan R, Ramakrishnan K (1995) Computation of legendre and zernike moments. Pattern Recognition 28(9):1433–1442

Neuhaus M, Bunke H (2005) Edit distance based kernel functions for attributed graph matching. In: Graph based Representations in Pattern Recognition, pp 352–361

Nixon M, Aguado AS (2008) Feature Extraction and Image Processing, Second Edition. Academic Press

Nocedal J, Wright S (1999) Numerical Optimization. Springer series in operations research, NY

Paglieroni D (2004) Design considerations for image segmentation quality assessment measures. Pattern Recognition 37(8):1607–1617

Perona P, Malik J (1990) Scale space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence 12(7):629–639

Petrou M (2006) Image Processing: Dealing with Texture. Wiley

Petrou M, Kittler J (1991) Optimal edge detectors for ramp edges. Pattern Analysis and Machine Intelligence 13(5):483–491

Ping Z, Wu R, Sheng Y (2002) Image description with chebyshev-fourier moments. Journal of the Optical Society of America 19(9):1748–1754

Povlow B, Dunn S (1995) Texture classification using noncausal hidden markov-models. Pattern Analysis and Machine Intelligence 17(10):1010–1014

Prasad B, Biswas K, Gupta S (2004) Region-based image retrieval using integrated color, shape, and location index. Computer Vision and Image Understanding 94(1-3):193–233

Prewitt J, Mendelsohn M (1966) The analysis of cell images. Annals of the New York Academy of Sciences 128:1035–1053

Prokop R, Reeves A (1992) A survey of moment-based techniques for unoccluded object representation and recognition. Graphical Models and Image Processing 54(5):438–460

Rakotomamonjy A, Bach F, Canu S, Grandvalet Y (2007) More efficiency in multiple kernel learning. In: International Conference on Machine Learning, pp 775–782

Randen T, Husoy J (1994) Multichannel filtering for image texture segmentation. Optical Engineering 33(8):2617–2625

Randen T, Husoy J (1999) Filtering for texture classification: A comparative study. Pattern Analysis and Machine Intelligence 21(4):291–310

Rieder H (1994) Robust Asymptotic Statistics. Springer

Riesen K, Neuhaus M, Bunke H (2007) Bipartite graph matching for computing the edit distance of graphs. In: Graph based Representations in Pattern Recognition, pp 1–12

Roberts L (1977) Machine perception of 3-d solids. In: Computer Methods in Images Analysis, pp 285–323

Roman Roldan R, Gomez Lopera J, Atae Allah C, Martinez Aroza J, Luque Escamilla P (2001) A measure of quality for evaluating methods of segmentation and edge detection. Pattern Recognition 34(5):969–980

Romaniuk B, Desvignes M, Revenu M, Deshayes M (2004) Shape variability and spatial relationships modeling in statistical pattern recognition. Pattern RecognitionL 25(2):239–247

Russell B, Torralba A, Murphy K, Freeman W (2008) Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision 77(1-3):157–173

Sahbi H (2007) Kernel pca for similarity invariant shape recognition. Neurocomputing 70(16-18):3034–3045

Saito N, Coifman R (1995) Local discriminant bases and their applications. Journal of Mathematical Imaging and Vision 5(4):337–358

Schapire RE (1990) The strength of weak learnability. Machine Learning 5:197–227

Schölkopf B, Smola AJ (2002) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). The MIT Press

Shawe-Taylor J, Cristianini N (2004) Kernel Methods for Pattern Analysis. Cambridge University Press

Shi J, Malik J (2000) Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence 22(8):888–905

Simon HA (1984) Why should machines learn? In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine Learning: An Artificial Intelligence Approach, Springer, Berlin, Heidelberg, pp 25–37

Singh S, Haddon J, Markou M (2001) Nearest-neighbour classifiers in natural scene analysis. Pattern Recognition 34(8):1601–1612

Sivic J, Zisserman A (2003) Video google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA

Sivic J, Russell BC, Efros AA, Zisserman A, Freeman WT (2005) Discovering object categories in image collections. In: International Conference on Computer Vision

Smeulders A, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence 22(12):1349–1380

Sobel I (1970) Camera models and machine perception. In: Ph.D.

Srinivasan A, Muggleton S, Sternberg MJE, King RD (1996) Theories for mutagenicity: A study in first-order and feature-based induction. Artificial Intelligence 85(1-2):277–299

Steinwart I, Christmann A (2009) Support Vector Machines. Springer

Stone C (1977) Consistent nonparametric regression. Annals of Statistics 5(4):595–645

Straccia U (2005) Description logics with fuzzy concrete domains. In: Uncertainty in Artificial Intelligence, pp 559–567

Strand J, Taxt T (1994) Local frequency features for texture classification. Pattern Recognition 27(10):1397–1406

Suard F, Guigue V, Rakotomamonjy A, Benshrair A (2005) Pedestrian detection using stereo-vision and graph kernels. Intelligent Vehicles Symposium, 2005 Proceedings IEEE pp 267–272

Swain M, Ballard D (1991) Color indexing. International Journal of Computer Vision 7(1):11–32

Syeda Mahmood T (1997) Data and model-driven selection using color regions. International Journal of Computer Vision 21(1-2):9–36

Takemura CM, Cesar RM, Bloch I (2005) Fuzzy modeling and evaluation of the spatial relation "along". In: Iberoamerican Congress on Pattern Recognition, pp 837–848

Taylor JS, Cristianini N (2004) Kernel Methods for Pattern Analysis. Cambridge University Press

Teague M (1980) Image analysis via the general theory of moments. Journal of the Optical Society of America 70(8):920–930

Toews M, Collins DL, Arbel T (2006) A statistical parts-based appearance model of inter-subject variability. In: Medical Image Computing and Computer Assisted Intervention, pp 232–240

Treisman A (1985) Preattentive processing in vision. Computer Vision, Graphics, and Image Processing 31(2):156–177

Tsuda K, Kin T, Asai K (2002) Marginalized kernels for biological sequences. Bioinformatics 18(Suppl 1):268–275

Tu Z, Zhu S (2002) Image segmentation by data-driven markov chain monte carlo. Pattern Analysis and Machine Intelligence 24(5):657–673

Tuia D, Matasci G, Camps G, Kanevski M (2009) Learning the relevant image features with multiple kernels. In: International Geoscience and Remote Sensing Symposium

Tuytelaars T, Gool LJV (2004) Matching widely separated views based on affine invariant regions. International Journal of Computer Vision 59(1):61–85

Vanegas C, Bloch I, Maître H, Inglada J (2009) Approximate Parallelism Between Fuzzy Objects: Some Definitions. In: International Workshop on Fuzzy Logic and Applications, Palermo, Italy, vol LNAI 5571, pp 12–19

Vapnik V, Lerner A (1963) Pattern recognition using generalized portrait method. Automation and Remote Control 24:774–780

Vasconcelos N (2008) From pixels to semantic spaces: Advances in content-based image retrieval. Computer 40(7):20–26

Vert JP (2002a) Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. In: Pacific Symposium on Biocomputing, pp 649–660

Vert JP (2002b) A tree kernel to analyse phylogenetic profiles. In: Intelligent Systems for Molecular Biology, pp 276–284

Vidyasagar M (2002) A Theory of Learning and Generalization: With Applications to Neural Networks and Control Systems. Springer-Verlag New York, Inc.

Vieu L (1997) Spatial Representation and Reasoning in Artificial Intelligence. In: Stock O (ed) Spatial and Temporal Reasoning, Kluwer, pp 3–41

Vishwanathan SVN, Smola AJ (2002) Fast kernels for string and tree matching. In: Neural Information Processing Systems, pp 569–576

Vishwanathan SVN, Borgwardt KM, Schraudolph NN (2006) Fast computation of graph kernels. In: Neural Information Processing Systems, pp 1449–1456

Weber M, Welling M, Perona P (2000) Unsupervised learning of models for recognition. In: European Conference on Computer Vision, pp 18–32

Weldon T, Higgins W, Dunn D (1996) Efficient gabor filter design for texture segmentation. Pattern Recognition 29(12):2005–2015

Zelenko D, Aone C, Richardella A, K J, Hofmann T, Poggio T, Shawe-Taylor J (2003) Kernel methods for relation extraction. Journal of Machine Learning Research 3:2003

Zhang H, Fritts J, Goldman S (2008) Image segmentation evaluation: A survey of unsupervised methods. Computer Vision and Image Understanding 110(2):260–280

Zhang Y (1996) A survey on evaluation methods for image segmentation. Pattern Recognition 29(8):1335–1346

Zhang Y, Brady J, Smith S (2001) Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. Medical Imaging 20(1):45–57