Sparsity regularization and graph-based representation in medical imaging

Aikaterini Gkirtzou

Sparsity regularization and graph-based representation in medical imaging

by

Katerina Gkirtzou

A thesis submitted in partial fulfillment for the degree of Doctor of Ecole Centrale Paris

in the

Specialty of Applied Mathematics

defended on December 17, 2013

Committee

Chairman :	Andreas Bartels	-	University of Tübingen
Reviewers :	Rafeef Abugharbieh	-	University of British Columbia
	Georg Langs	-	Medical University of Vienna
Advisors :	Nikos Paragios	-	École Centrale Paris
	Matthew Blaschko	-	École Centrale Paris
Examiners :	Guillaume Bassez	-	Université Paris-Est
	Gaël Varoquaux	-	INRIA Saclay

Dedicated to Panagiotis Koutsourakis

Acknowledgements

It is a true pleasure for me to thank all the people who made this thesis possible.

First of all, I would like to thank the director of the lab, Nikos Paragios, for giving me the chance to accomplish my PhD thesis at Ecole Centrale Paris and for providing me with the opportunity to meet and collaborate with incredible scientists. This long journey was an incredible experience I will never forget.

I owe my honest and deepest gratitude and respect to my supervisor Matthew Blaschko, both as a professor and as a person. I would like to thank him for all our passionate discussions and brainstorming sessions concerning my thesis and machine learning in general and for all the interesting diverse subjects as well, for all the time he dedicated to explain the tiniest detail to me, for all the meticulous effort to improve every single paper and presentation. Moreover, I feel gratitude for his strong support, his continuous encouragement whenever difficulties arose and for his invaluable guidance and inspiration during the time it took to accomplish this thesis. It was a real privilege and a great pleasure to work with him.

Next, I am grateful to Rafeef Abugharbieh and Georg Langs for their fruitful reviews on my manuscript. They made it a point of honor to comment thoroughly on my work, while being generous with enthusiastic remarks. I also address many thanks to Andreas Bartels, Guillaume Bassez and Gaël Varoquaux for participating in my thesis committee. It was a privilege to have such well-known researchers evaluating my work.

During my thesis, I had the opportunity to collaborate closely with great scientists. I would like to thank our collaborators from the United States for providing me with the cocaine addiction dataset and for their valuable expertise: Jean Honorio, Dimitris Samaras and Rita Goldstein. A special thanks to Jean Honorio for his insightful comments and for the fruitful discussions we had. I would also like to thank Deux Jean-François, Bassez Guillaume, Sotiras Aristeidis, Rahmouni Alain and Varacca Thibault for the muscle dataset, and Andreas Bartels for providing the free viewing dataset.

I would also like to thank all the people I met at Ecole Centrale Paris who made my staying there a wonderful and cheerful experience. First of all, I would like to thank Xiang Bo, my office roomy, for her support and her help, especially for helping find a very cozy place to live in during the last months of my Phd. Then I would like to thank all the people who welcomed me to the MAS-Vision lab, when I arrived in Paris: Aris Sotiras, Olivier Teboul, Loic Simon, Chaohui Wang, Ahmed Besbes, Salma Essafi, Radhouene Neji, Regis Behmo, Fabrice Michel, Nicolas Honnorat and Pierre-Yves Baudin. Also I would like to thank all the people from the CVC lab, who made the lab during the last period of the thesis and the most difficult one a very pleasant place to work in: Evgenios Kornaropoulos, Enzo Ferrante, Punnet Kumar, Siddartha Chandra, Stefan Kinauer, Sarah Parisot, Vivian Fecamp, Stavros Alchatzidis, Haithem Boussaid, Stavros Tsogkas and Gao Lufang. Lastly, I would like to thank all the administration personnel, especially the lab secretaries: Sylvie Dervin – the secretary of MAS-Vision who helped me a lot with the bureaucratic procedure when I entered the Ecole Centrale – as well as Carine Morotti-Delorme and Natalia Leclercq – our CVC lab secretaries who were “life savers” for the lab and whose help for the paperwork of the defense was extremely valuable.

Outside the professional sphere, I would like to thank all my whole family and especially my mother, my father and my brother, who always showed me love and unconditional support to chase my dreams and who I sometimes take for grated and forget to show them how precious they are to me.

There are also some friends that I consider as family and that I would also like to acknowledge. They all supported and helped me, sometimes without even knowing it. My dear friend Sofia Maschalidi (Sofouko) who stood by me like a rock, always bringing me love, help and support. She will always have my ever lasting love and gratitude. Also I enjoyed the support of many friends who were in Paris: Eirini Papagiakoumou, Dimitris Papadopoulos, Anna Athanasopoulou, Manos Saridakis and Anthi Koskina and their little girls (Eliza, Peggi and Nicole), Dimitra Konsta, Rania Panousi, Lida Kotronaki, Alexandros Giannopoulos, Anestis Charadontis, Yves Piriou and Aline Barazer. It is very important to have close friends to have fun with and enjoy a nice drink or a stroll in Paris.

I also had the support from friends who although they were far away, they were actually really close to me: my ex-roomy Elias Grinias, my old girlfriends Angeliki Siafaka, Eftichia Ioannidou and Ioanna Manoutsoglou, my “pararell” friends : Manos Avgeridis (Manolito) who offered me shelter when I was writing my thesis, Eva Paradissi and Katerina Stavroula with whom I had great summer vacations when I needed them most, Marina Kontara (la Belge), Giorgos Antonopoulos (Jorgito), Petros Stergiou, Tasos

Petropoulos, Nikos Manikas, Kostis Anagnostakis, Michalis Vourekas, Dimitra (Mitsi), Giannis Chatzidimitrakis and Mary Sitmaki for their long discussions and trolling sessions that kept me calm and happy. All of them are like brothers and sisters and they have my ever lasting gratitude.

Last but definitely not least, my thoughts are with Panagiotis Koutsourakis, a beloved and dear friend to whom this thesis is dedicated as if it weren't for him, this thesis would never have come true.

Abstract

Medical images have been widely used in modern medicine to depict the anatomy or function for both clinical purposes and for studying normal anatomy. Analyzing medical images efficiently and with high accuracy is a crucial step. The high-dimensionality and the non-linear nature of medical imaging data makes their analysis a difficult and challenging problem. In this thesis, we address the medical image analysis from the viewpoint of statistical learning theory and we concentrate especially on the use of regularization methods and graph representation and comparison.

First, we approach the problem of graph representation and comparison for analyzing medical images. Graphs are a commonly used technique to represent data with inherited structure. Exploiting these data, requires the ability to efficiently compare and represent graphs. Unfortunately, standard solutions to these problems are either NP-hard, hard to parametrize and adapt to the problem at hand or not expressive enough. Graph kernels, which have been introduced in the machine learning community the last decade, are a promising solution to the aforementioned problems.

Despite the significant progress in the design and improvement of graph kernels in the past few years, existing graph kernels focus on either unlabeled or discretely labeled graphs, while efficient and expressive representation and comparison of graphs with complex labels, such as real numbers and high-dimensional vectors, remains an open research problem. We introduce a novel method, the *pyramid quantized Weisfeiler-Lehman graph representation* to tackle the graph comparison and representation problem for continuous vector labeled graphs. Our algorithm considers statistics of subtree patterns based on the Weisfeiler-Lehman algorithm and uses a pyramid quantization strategy to determine a logarithmic number of discrete labellings. As a result, we approximate a graph representation with continuous or vector valued labels as a sequence of graphs discrete labels with increasing granularity. We evaluate our proposed algorithm on two different tasks with real datasets, on a fMRI analysis task and on the generic problem of 3D shape classification.

Second, we examine different regularization methods for analyzing medical images, and more specifically MRI data. Regularization methods are a powerful tool for improving the predicted performance and avoid overfitting by introducing additional information to an ill-posed problem, such as the analysis of medical images. Towards this direction, we introduce a novel regularization method, the *k-support regularized Support Vector Machine*. This algorithm extends the ℓ_1 regularized SVM to a mixed norm of both ℓ_1 and ℓ_2 norms. This enables the use of a correlated sparsity regularization with the power of the SVM framework. We evaluate our novel algorithm in a neuromuscular disease classification task using MRI-based markers. We furthermore explore the importance of diffusion tensor imaging for the discrimination between neuromuscular conditions.

Overall, as graphs are fundamental mathematical objects and regularization methods are widely used to control ill-pose problems, both the *pyramid quantized Weisfeiler-Lehman graph representation* and the *k-support regularized SVM* are potentially applicable to a wide range of applications domains in computer vision, analysis of medical images and data mining.

Keywords: Weisfeiler-Lehman algorithm, graph kernels, regularization, k-support norm, MRI, DTI, 3D shape classification

Résumé

Les images médicales ont largement utilisées en médicine moderne afin de représenter l'anatomie ou les fonctions, à la fois dans un objectif cliniques ou d'étude de l'anatomie normale. L'analyse efficace et précise d' images médicales est une étape critique. La dimensionnalité élévée et le caractère non-linéaire des données d'imagerie médicale rendent leur analyse difficile. Dans cette thèse, nous nous intéressons à l'analyse d'images médicales du point de vue de la théorie statistique de l'appretissage et nous concentrons spécialement sur l'utilisation de méthodes de régularisation et de la représentation et comparaison des graphes.

Tout d'abord, nous nous intéressons un problème de représentation et comparaison des graphes pour l'analyse des images médicales et de façon plus générale. Les graphes sont une technique largement utilisée pour la représentation des données ayant une structure héritée. L'exploitation des ces données nécessite la capacité de comparer et représenter efficacement des graphes. Malheureusement, les solutions usuelles à ces problèmes sont soit NP-complets, difficiles à paramétrer et à adapter au problème donnée, soit insuffisamment expressives. Les noyaux sur graphes, introduits à la communauté de l'apprentissage statistique au cours de la dernière décennie, offrent une solution promettante aux problèmes mentionnés ci-dessus.

Malgré le progrès significatif dans le domaine de la conception et amélioration des noyaux sur graphes au cours des dernières années, les noyaux sur graphes existants se concentrent à des graphes non-labellisés ou labellisés de façon dicrète, tandis que la représentation et comparaison efficaces et expressives de graphes avec des labels complexe, comme des nombres réels ou des vecteurs à grande dimension, demeure une problème de recherch ouvert. Nous introduisons une nouvelle méthode, l'algorithme de Weisfeiler-Lehman pyramidal et quantifié (pyramid quantized Weisfeiler-Lehman algorithm), afin d'aborder le problème de la représentation et comparaison des graphes labellisés pqr des vecteurs continus. Notre algorithme considère les statistiques de motifs sous arbre, basé sur l'algorithme Weisfeiler-Lehman ; il utilise une stratégie de quantification pyramidale pour déterminer un nombre logarithmique de labels discrets. Par conséquent, nous approximons une représentation de graphe avec des labels continus ou vecteur, comme une séquence de graphes avec des labels discrèts de plus en plus granulaires. Nous évaluons notre algorithme proposé sur deux tâches différentes et des bases des données réelles : un tâché d'une analyse IRMf et une tâche de problème générique de la classification de formes en trois dimensions.

Ensuite, nous examinons différentes méthodes de régularisation pour analyser les images médicales, et plus spécifiquement des données d'IRM. Les méthodes de régularisation sont un outil puissant pour l'amélioration de la performance prédite et pour éviter le sur-apprentissage via l'introduction d'informations additionelles à un problème mal-posé tel que l'analyse d'images médicales. Dans cette direction, nous introduisons une nouvelle méthode de régularisation, la k-support regularized Support Vector Machine (les machines à vecteurs de support régularisées k-support). Cet algorithme étend la SVM régularisée ℓ_1 à une norme mixte de toutes les deux normes ℓ_1 et ℓ_2. Ceci permet l'utilisation d'une régularisation parcimonieuse corrélée à la puissance des SVM. Nous évaluons notre original algorithme sur une tâche de classification de maladies neuromusculaires, en utilisant des marqueurs à base de IRM. Par la suite, nous explorons l'importance de l'imagerie du tenseur de diffusion pour la discrimination entre les conditions neuromusculaires.

Globalement, les graphes étqnt des objets mathématiques fondamentaux et les méthodes de régularisation étant largement utilisées pour contrôler des problèmes mal-posés, l' algorithme de Weisfeiler-Lehman pyramidal et quantifié (pyramid quantized Weisfeiler-Lehman algorithm) et la SVM régularisées k-support (k-support regularized SVM), pourraient bien être appliqués sur un grand éventail d'applications dans les domaines de vision artificielle, l'analyse d'images médicales et l'exploration de données.

Mots-clefs : algorithme Weisfeiler-Lehman, noyaux de graphes, régularisation, norme k-support, IRM, IDT, classification de la forme en trois dimensions.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Medical imaging consists of a number of different techniques that create images of the human body showing anatomy or function and are used for clinical purposes, such as diagnosing or monitoring the progression of a disease, or for studying normal anatomy and physiology. Over the years, different modalities of medical imaging have been developed, including (*a*) x-ray based methods - such as conventional x-rays, computed tomography and mammography - (*b*) magnetic resonance imaging (MRI) - such as T1-weighted images and diffusion tensor imaging - (*c*) molecular imaging - such as positron emission tomography (PET) - and (*d*) ultrasound, each with their own advantages and disadvantages. They are widely used in daily clinical routines, due to the fact that are generally non-invasive, relatively fast, allowing to image the human body and providing relevant anatomical or function information to the doctor, while minimizing the patient's discomfort.

The medical imaging field continuously improves as technology evolves providing more accurate and rich information. In order to extract the relevant information and provide it to the physician, analyzing medical images is an essential step in modern medicine. The high dimensionality and the non-linearity of the data makes medical image analysis a difficult and challenging problem. In this thesis, we approach medical image analysis from the perspective of statistical learning theory [Vapnik, 1995, Hastie et al., 2009] and more specifically we focus on the use of graph representation and different regularization methods.

Graphs are a general, powerful, flexible and natural way to mathematically represent complex data with integrated structure. A graph consists of a set of nodes – which represents the objects of interest – and a set of edges – which represents the relations between them [Diestel, 2010, Gibbons, 1985]. For example, a molecule can be represented as graph by taking the atoms as nodes, while when a pair of atoms is connected with a bond this relationship can be represented with an edge. Extra information, such as the type of the chemical bond in the previous example, can be incorporated into the graph as labels in the edges. Applications involving graph representation are numerous and they occur in a number of different fields. We list below examples from a number of representative fields.

Computer vision and biomedical imaging Graphs have been widely used in computer vision problems over the past decades. There are often used for representing images – either as grid of pixels or as a graph of adjacency regions or segmented parts – and for solving problems, such as segmentation [Greig et al., 1989] or finding correspondences between two images [Torresani et al., 2008]. Graphs are also used in biomedical image analysis to represent and model organs, such as the brain [Ng et al., 2012a, Rao et al., 2010], which can potentially be used in diagnosis or studying the human body.

Bioinformatics Advances in technology in the last 15 years allow the generation of vast amount of genome sequences and gene expression levels, as well as the detection of biomolecular interactions. These various data produce various types of graph representations, such as protein-protein interactions [Canutescu et al., 2003], metabolic pathways [Wagner and Fell, 2001], transcriptional pathways and evolutionary relationships [Goldstein, 1979]. A number of interesting questions raise from the analysis of these graphs such as which genes regulate others, how the phenotype is influenced, whether we can we predict the interaction between a pair of proteins based on their structure, *etc.* These graph representations can contain complex labels and incomplete information making their analysis a challenging task.

Social networks The wide spread use of internet in more and more domains with more and more people, the augmentation of email exchange, the expand of new means of communications such as blogs, social networks or instant messages, create a vast amount of data that can be represented as graphs. The analysis of these networks is both of scientific and commercial interest. On the one hand, psychologists want to study the complex social dynamics among humans and biologists want to explore the social rules in a group of animals. On the other hand, industries want to analyze these networks for marketing purposes. Detecting influential individuals in a group of people is relevant for marketing, as companies could then focus their advertising efforts on these individuals, which can influence the behavior of the whole group.

Chemoinformatics Chemistry is another domain where graph representation and graph comparison is applied [Hapke, 2005]. Finding chemical compounds with a specific property is a common problem in chemistry and pharmacology. A common assumption is that molecules with similar structure share also similar functional properties and as chemical molecules have been widely represented as graph – where atoms represent vertices and bonds represent the edges – so being able to compare graphs and find similarity among them is a crucial problem in chemoinformatics.

On the other hand, regularization, in the fields of machine learning and statistics, refers to the process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting [Hastie et al., 2009]. This information usually has the form of a penalty on the complexity of the learned model or restrictions for smoothness. Regularizers have been extensively used in various problems, among them the reconstruction of PET images [Kaufman and Neumaier, 1996], image segmentation [Woolrich et al., 2005] and classification or regression problems.

In this thesis, we focus on the following tasks: (a) the analysis of fMRI data through the representation as labeled graphs and (b) the analysis of MRI data using regularization methods. Towards this direction, we also introduce two novel learning algorithms, the *pyramid quantized Weisfeiler-Lehman graph representation* and the *k-support regularized SVM.*

1.2 Statistical learning

Given a set of n paired observations $\{(x_i, y_i)\}_{1 \leq i \leq n} \in \mathbb{R}^d \times \mathbb{R}$ that is assumed to be independent and identically drawn from the joint distribution $p_{\mathbf{XY}}$, the goal of statistical learning is to learn a function $f(x) \in \mathcal{F}$ for predicting the output y given the input x. This *prediction function* $f(x)$ is built via the evaluation of a *loss function* $\mathcal{L}(f(x), y)$ that penalizes errors in prediction. This leads us to a criterion for choosing f, the *risk or generalization error* of the prediction function which is defined as:

$$\mathcal{R}(f) = \int \mathcal{L}(f(x), y) dp_{\mathbf{XY}}(x, y), \tag{1.1}$$

where $\mathcal{L}$ is a loss function and $p_{\mathbf{XY}}$ is the joint distribution, covering the probability of a label and an input being uncover together. Ideally, we would like to learn a prediction function f that will minimize the risk. However, since the joint probability $p_{\mathbf{XY}}$ is unknown, the risk is also unknown. Nonetheless we can approximate the risk $\mathcal{R}(f)$ and empirically calculate it through the given set of paired observation $\{(x_i, y_i)\}_{1 \leq i \leq n} \in \mathbb{R}^d \times \mathbb{R}$, that is called the *training set.* Assuming that the training set is sampled independently and from the same joint distribution $p_{\mathbf{XY}}$ (*i.e.* the i.i.d assumption holds) then the risk can be approximated as follows:

$$\mathcal{R}(f) \approx \hat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(x_i), y_i) \tag{1.2}$$

Equation 1.2 is called the *empirical risk* and as $n \to \infty$, the empirical risk will approach the true risk, $\hat{\mathcal{R}}(f) \to \mathcal{R}(f)$ and we have statistical consistency for an estimator that returns $\arg\min_{f\in\mathcal{F}} \hat{\mathcal{R}}(f)$.

In real datasets we always have finite sample sizes, so choosing a prediction function $f \in \mathcal{F}$ that minimizes the empirical risk (see Equation 1.2), often leads to *overfitting*. This means that the empirical risk $\hat{\mathcal{R}}$ is much lower that the real risk $\mathcal{R}$. One way to avoid overfitting on the training dataset and being able to generalize well on new data is by adding a regularization term $\lambda\Omega(f)$, where $f \in \mathcal{F}$ is the prediction function we would like to learn, λ is a scalar parameter that controls the degree of regularization and $\Omega : \mathcal{F} \to \mathbb{R}$ is a scalar valued function that penalizes the "complexity" of the prediction function. A number of regularizers have been proposed in the literature, among them the Tikhonov regularization [Tikhonov, 1943], the LASSO [Tibshirani, 1996] and Elastic Net [Zou and Hastie, 2005]. By adding the regularization term to the empirical risk from Equation 1.2 the problem is formulated as follows:

$$\arg\min_{f\in\mathcal{F}} \; \lambda\Omega(f) + \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(x_i), y_i) \tag{1.3}$$

For the problem defined in Equation 1.3 in this thesis we examine novel combinations of regularizers and prediction functions for the analysis of medical images. We justify those choices both theoretically and empirically in the following chapters.

1.3 Thesis outline

The thesis is organized as follows: we first concentrate on the graph comparison problem and on our proposed method that efficiently compares graphs with continuous or vector labels, the *pyramid quantized Weisfeiler-Lehman algorithm*. To the best of our knowledge, this is the first method for efficiently comparing graphs with continuous vector labels. In Chapter 2 we start by introducing basic notations and background from graph theory (see Section 2.1) and we then explore the graph comparison problem in detail (see Section 2.2 and Section 2.3). In Chapter 3 we introduce our novel algorithm, the *pyramid quantized Weisfeiler-Lehman algorithm*. Our algorithm is based on the Weisfeiler-Lehman test of isomorphism, described in Section 3.1, which was recently employed as a graph kernel to compare graphs with discrete labels (see Section 3.2). In order to make use of the efficiency of the Weisfeler-Lehman algorithm and apply it to continuous or vector labeled graphs, in Section 3.3 we present a pyramid quantization strategy and transform the graph representation with continuous or vector valued labels as a sequence of graphs with increasingly granular discrete labels. Finally, we explore different tactics for combining information from the various pyramid quantization levels in Section 3.4.

In Chapter 4 we evaluate our proposed algorithm on two different tasks. The first one, described in Section 4.1, is from the area of fMRI analysis and its objective is to discriminate between cocaine abusers and healthy control subjects. The second one, described in Section 4.2, is from the area of 3D shape classification. In this task, we use two datasets with 3D meshes, one that comes from the medical area, whose objective is to discriminate between healthy and patient subjects that suffer from a neuromuscular dystrophy, while in the second dataset we tackle a multiclass problem of generic object classification.

Apart from the graph comparison problem, in Chapter 5 we explore different regularization methods for analyzing medical images, and more specifically MRI data. In Section 5.1 we present the regularizers under investigation and we also introduce our novel learning algorithm the *k-support regularized SVM* (see Section 5.1.3.2). In our first experiment in Section 5.2, we investigate the use of regularization methods, the well-known LASSO and Elastic Net and the newly introduced k-support norm with squared loss, in the analysis of fMRI images. In the following Section 5.3, we evaluate our newly introduced learning algorithm, the *k-support regularized SVM* in the discriminative task of neuromuscular disease classification using features extracted from T1-weighted, T2-weighted and diffusion tensor imaging. Although DTI imaging is widely used in neuroimaging studies, it has been recently introduced in the clinical analysis of the calf muscle and we also investigate its significance for neuromuscular classification. Finally, in Chapter 6, we conclude the thesis by summarizing the contributions (Section 6.1) and by offering some future perspectives (Section 6.2).

1.4 Published work appearing in this thesis

This thesis contains material, modified or in extended form, from several published articles. We list these publications below, indicating the corresponding sections of this manuscript.

- Katerina Gkirtzou, Jean Honorio, Dimitris Samaras, Rita Goldstein, and Matthew B. Blaschko. MRI Analysis with Sparse Weisfeiler-Lehman Graph Statistics. In *4th*

International Workhop on Machine Learning in Medical Imaging, Nagoya, Japan, September 2013a
Parts from Section 3.3.1 and 3.4.2 from Chapter 3 and Section 4.1 is based on the work presented in this paper.

- Katerina Gkirtzou, Jean Honorio, Dimitris Samaras, Rita Goldstein, and Matthew B. Blaschko. FMRI analysis of cocaine addiction using k-support sparsity. In *International Symposium on Biomedical Imaging*, San Francisco, USA, January 2013b
Section 5.2 of Chapter 5 is based on the work presented in this paper.

- Katerina Gkirtzou, Jean-François Deux, Guillaume Bassez, Aristeidis Sotiras, Alain Rahmouni, Thibault Varacca, Nikos Paragios, and Matthew B. Blaschko. Sparse classification with MRI based markers for neuromuscular disease categorization. In *4th International Workhop on Machine Learning in Medical Imaging*, Nagoya, Japan, September 2013c
Section 5.1.3.2 and Section 5.3 of Chapter 5 is based on the work presented in this paper.

Certain parts of this thesis are based on unpublished work done in collaboration with other researchers. “Data-guided binning” from Section 3.3.1, Section 3.3.2, Section 3.3.3 and Section 3.4.1 are based on unpublished research with Matthew B. Blaschko. Material in Section 4.2 is based on unpublished research with Nikos Paragios and Matthew B. Blaschko.

Chapter 2

Related work on graph comparison

Graphs are commonly used to represent objects and the relationships among them in a general, powerful and flexible way. Graphs consist of a set of nodes, which typically represents the objects of interest, and a set of edges, which expresses the relationships among the objects. Graph representations are widely employed in a number of areas, such as bioinfomatics, social network analysis, *etc.* (for more details see Section 1.1).

In this chapter we explore the graph comparison problem in detail. We first introduce key concepts and notation from graph theory in Section 2.1 and we then review the most related work on the graph comparison problem, which can be classified in the following categories, (a) graph comparison methods (Section 2.2) and (b) graph kernel methods (Section 2.3).

2.1 Graph theory basics and notation

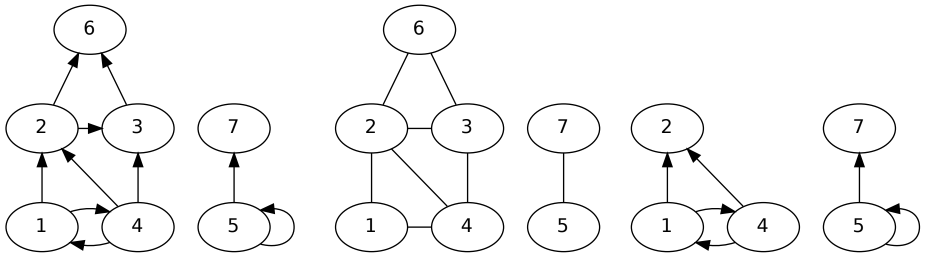
In order to understand the importance of graphs and especially the problem of graph comparison, we will need some basic background of graph theory. In this section we will define the terminology and the basic notation that will be used for the rest of the thesis. Most of the graph-theoretic terminology follows the monograph of Diestel [Diestel, 2010] or the monograph of Gibbons [Gibbons, 1985].

2.1.1 Directed, undirected and labeled graphs and subgraphs

Definition 2.1 (Graph). A *graph* G is a pair of sets (V, E), where V is the *vertex set* and its elements are called *vertices* (also known as nodes or points) and $E \subseteq V \times V$ is the *edge set* which represents a binary relation on V and its elements are called *edges* (also known as arcs or lines).

The *order or size* of a graph G is defined as the number of vertices $|V|$. Graphs are *finite, infinite, countable* and so on according to their order.

Definition 2.2 (Directed and Undirected graph). A graph $G = (V, E)$ is called *directed* when the edge set E consists of *ordered pairs* of vertices, that is $(u, v) \in E$ is considered to be directed from u to v and $u, v \in V$. When the edge set E contains *unordered pairs* , that is $(u, v) \in E$ and $(v, u) \in E$ are considered to be the same edge $\forall u, v \in V$, it is called *undirected*.



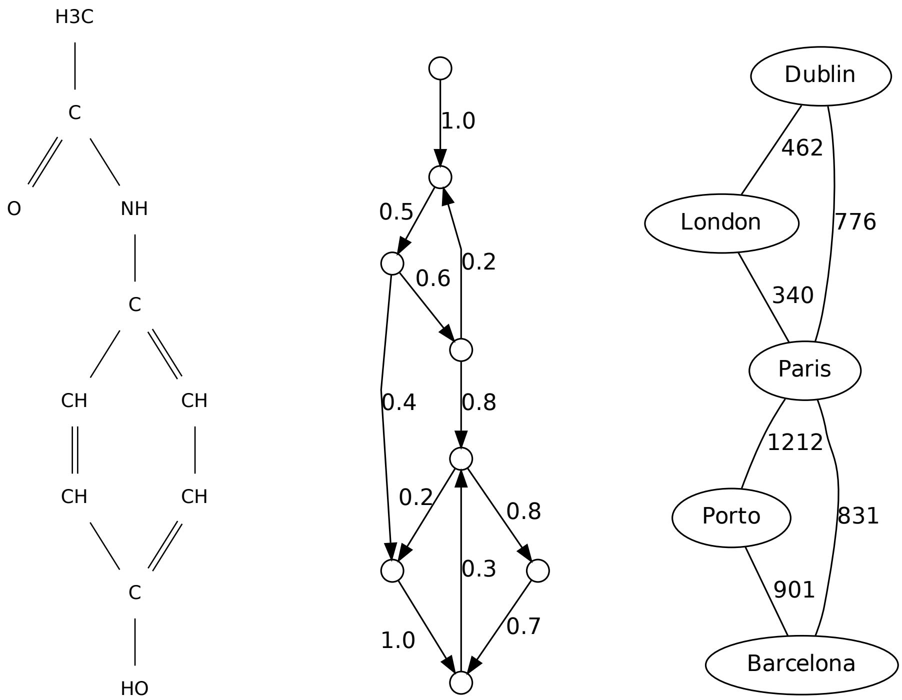
(a) Directed Graph (b) Undirected Graph (c) Induced Subgraph

FIGURE 2.1: Figure 2.1(a) shows an example of a directed graph with $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{(1, 2), (1, 4), (2, 3), (2, 6), (3, 6), (4, 1), (4, 2), (4, 3), (5, 5), (5, 7)\}$. Figure 2.1(b) shows an undirected graph similar to the direct graph of Figure 2.1(a). Figure 2.1(c) shows the induced subgraph of the graph from Figure 2.1(a) when $V' = \{1, 2, 4, 5, 7\}$.

Figure 2.1(a) and Figure 2.1(b) show examples of directed and undirected graphs respectively. The vertices are represented with circles, while edges are represented as arrows for the directed graph and as lines for the undirected one. When a graph does not contain multiple edges between the same pair of nodes (and of the same direction in a directed graph) as well as *self-loops* – edges from a vertex to itself – then it is called *simple graph*. In this thesis, we mainly focus on simple graphs.

A graph can have labels on its nodes and/or on its edges and in this case the graph is called *labeled*.

Definition 2.3 (Labeled Graph). A *labeled* graph is defined as a triplet $G = (V, E, \mathcal{L})$, where V is the vertex set, E is the edge set and $\mathcal{L} : X \to \Sigma$ is a function assigning a label from an alphabet Σ to each element of the set X, which can be either V, E or $V \cup E$ depending on whether only nodes, only edges or both are labeled.



(a) Node-labeled Graph

(b) Edge-labeled Graph

(c) Node and edge labeled Graph

Figure 2.2: Figure 2.2(a) shows a molecule of acetaminophen as an example of node-labeled graph, Figure 2.2(b) shows a flow network as example of edge-labeled graph, while Figure 2.2(c) shows an abstract map, where nodes represent different cities, edges represent connections between cities and the distance between them is used as edge label, as an example of graph with labels in both nodes and edges.

A graph with labels on its nodes is called *node-labeled*, a graph with labels on its edges is called *edge-labeled*. The most common cases of labeled graphs are the *weighted graphs* where each edge is associated with a continuous value, also known as *weight*. Examples of node-labeled, edge-labeled and both node and edge labeled graphs can be seen in Figure 2.2.

Definition 2.4 (Induced subgraph). A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$ and is denoted as $G' \subseteq G$. Given a set $V' \subseteq V$ the subgraph of G that is *induced* by V' is the graph $G' = (V', E')$ where $E' = \{(u, v) \in E : u, v \in V'\}$.

An example of an *induced subgraph* can be seen in Figure 2.1(c)

2.1.2 Neighborhood in graphs

Given an edge $e = (u, v) \in E$, we say that e is *incident with* u or v when u or v is an end-point of the edge e and nodes u, v are said to be *adjacent* or neighbors. Similarly, when two edges $e_i \neq e_j$ share a common node then they are also adjacent.

Definition 2.5 (Neighborhood and Degree of node). The *neighborhood* of a vertex v in a graph G, denoted as $N(v)$, is the induced subgraph of G consisting of all vertices adjacent to v and all edges connecting two such vertices. The *degree* of a vertex u, denoted as $d(u)$, is the number of edges incident with u.

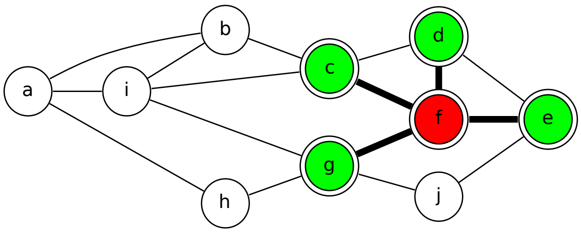


FIGURE 2.3: The neighborhood of node f (in red color) is the set of nodes $\{c, d, e, g\}$ depicted with green color. The degree of this node f is four.

An example of the neighborhood of a node and its degree can be seen in Figure 2.3. The neighborhood information of a graph is commonly represented as an adjacency matrix, which is defined as follows:

Definition 2.6 (Adjacency matrix). The *adjacency matrix* $A = (A_{ij})_{n \times n}$ of a graph $G = (V, E)$ is defined as :

$$A_{ij} = \begin{cases} 1 & \text{if } (u_i, u_j) \in E, \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

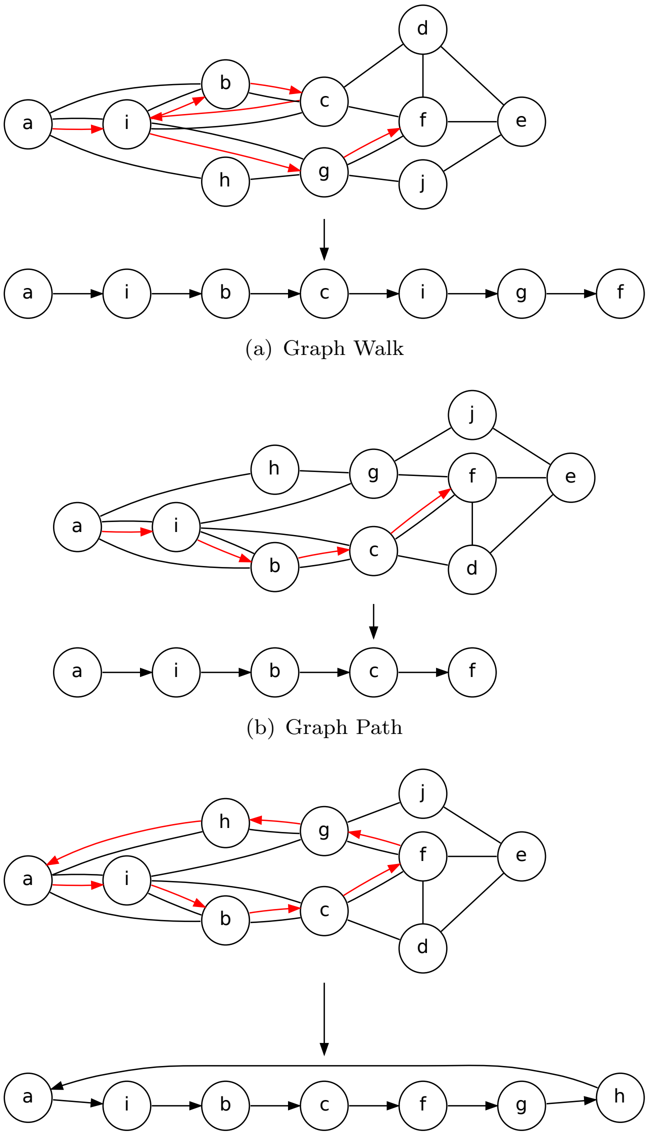
where u_i and u_j are nodes from G.

2.1.3 Walks, paths, cycles, trees, subtrees and subtree patterns

Definition 2.7 (Walk, path and cycle). A *walk* w of length l in a graph $G = (V, E)$ is a sequence of nodes and adjacent edges $(v_1, e_1, v_2, e_2, \ldots, e_{l-1}, v_l)$ such that $e_i = (v_i, v_{i+1})$ $\forall i$ $1 \leq i \leq (l-1)$. A *path* is a walk that contains only distinct nodes, while a *cycle* is a closed walk, where $v_1 = v_l$.

Sometimes in the literature the walk is also called a path, in that case the path is then called a simple path. Illustrations of a walk, a path and a cycle on a graph can be found in Figure 2.4.

Definition 2.8 (Connected and disconnected graph). A graph $G = (V, E)$ is said to be *connected* if for every pair of distinct vertices $u, v \in V$ there is a path joining them. A graph that is not connected is referred to as *disconnected*. The *distance* between two vertices $u, v \in V$ in graph $G = (V, E)$ is length of the shortest path from u to v in G, or ∞ if such path does not exist.



(a) Graph Walk

(b) Graph Path

(c) Graph Cycle

Figure 2.4: On the undirected graphs G shown above we represent with red arrows examples of a walk from node a to node f in Figure 2.4(a), a path from node a to node f in Figure 2.4(b) and a cycle from node a in Figure 2.4(c).

Definition 2.9 (Forest, tree and subtree). A graph G when it has no cycles is called *acyclic or forest*. A connected forest is called *tree*. A *rooted tree* is a *tree* with a specified *root* vertex v_0. A *subtree* is a subgraph of a graph that contains no cycles. When it has also a designated root node is called *rooted subtree*.

The *height* of a rooted tree or subtree is the maximum distance between the designated root vertex and any other node in the tree or subtree respectively. Similarly to the way of the notion of walk is extending the notion of path by allowing nodes [Bach, 2008] extended the notion of subtrees to *subtree patterns*, also known as *tree-walks*, which can have nodes that are equal.

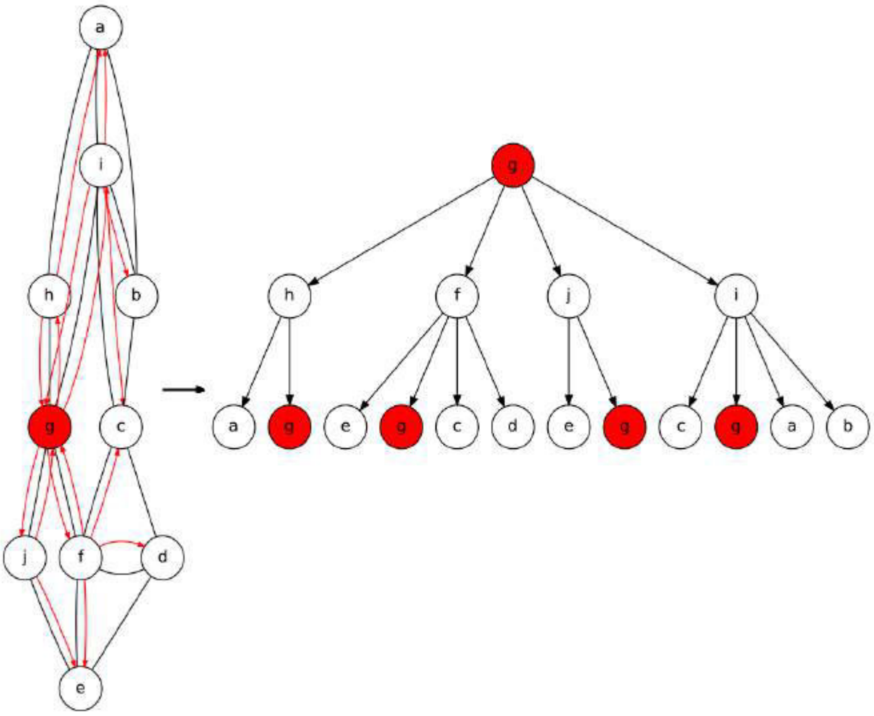


Figure 2.5: On the right is the initial graph and on the left a subtree pattern of height 2 rooted at vertex g, depicted with red color. It should be noted that the repetitions of vertices on the subtree pattern allows the pattern to be cycle-free.

Definition 2.10 (Subtree patterns). *Subtree patterns* are labeled trees extracted from a labeled graph G for a given depth h and a given vertex v. The vertices in the subtree pattern are labeled in accordance of the labels of the initial vertices in G, so the labels of neighbors in the subtree pattern are also neighbors in the G. The subtree pattern is represented by a tree structure T over the vertex set $\{1, \ldots, |T|\}$ and a sequence of consistent but possibly non distinct labels $\mathcal{L} \in V^{|T|}$.

An example of subtree pattern is shown in Figure 2.5.

2.1.4 Graph and subgraph Isomorphism

The first step towards graph comparison is the ability to check whether two graphs are identical or not. The problem of deciding that is called *graph isomorphism* and is defined as follows:

Definition 2.11 (Graph Isomorphism). Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. We call G and G' *isomorphic* and write $G \simeq G'$, if there exists a bijection $\phi : V \to V'$ with $(u, v) \in E \Leftrightarrow (\phi(u), \phi(v)) \in E'$ $\forall u, v \in V$.

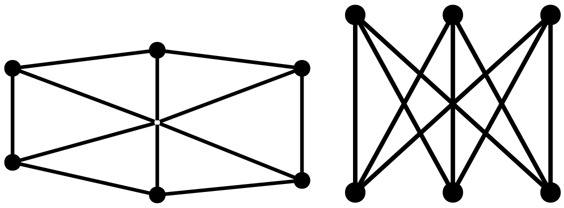


Figure 2.6: Two isomorphic representations of the $K_{3,3}$ graph.

Such a map ϕ is called *isomorphism* and if $G = G'$ it is called *automorphism*. Figure 2.6 shows two graphs which are isomorphic, each being a representation of the well known $K_{3,3}$ graph.[1] Apart from deciding whether two graphs are identical or not, one could determine whether the graph G contains a subgraph that is isomorphic to G', a problem know as *subgraph isomorphism*.

Definition 2.12 (**Subgraph Isomorphism**). Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. We call G' *subgraph isomorphic* to G, if there is a subgraph $G_0 = (V_0, E_0) \subseteq G : V_0 \subseteq V, E_0 = E \cap (V_0 \times V_0)$ such that exists a bijection $\phi : V_0 \to V'$ with $(u, v) \in E_0 \Leftrightarrow (\phi(u), \phi(v)) \in E' \; \forall u, v \in V_0$.

2.2 Graph comparison methods

As graphs are rich representations of data with inherited structure, they are consequently a promising tool in many domains, as we have already seen in Section 1.1. A common and challenging problem when dealing with graphs is to be able to compare them and provide a similarity measurement, a problem well-known as *graph comparison*.

Definition 2.13 (**Graph Comparison**). Given a set $\mathcal{G}$ of graphs, the problem of graph comparison is defined as a function

$$k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$$

such that $k(G, G')$ for $G, G' \in \mathcal{G}$ quantifies the similarity of G and G'.
In this section we review classical approaches to this problem.

2.2.1 Isomorphism-based methods

A first approach towards this problem is to quantify whether two graphs are identical, *i.e.* isomorphic (see Definition 2.11). This produce a binary similarity measure, which

[1] The $K_{i,j}$ is a special family of graphs, known as bipartite graphs, that it is possible to partition the vertices of the graph G into two subsets V_1 and V_2 such that every edge of G connects a vertex in V_1 to a vertex in V_2 and i, j is the degree of every vertex in the V_1 and V_2 respectively.

equals to 1 when the two graphs are isomorphic, otherwise equals to 0. Despite the fact that this idea of graph isomorphic is intuitive, no efficient algorithm is known for it. The graph isomorphism problem is known to be within NP, but neither a proof of NP-completeness nor a polynomial time algorithm are known [Garey and Johnson, 1979, Chapter 7]. Other similarity measures are based on concepts related to isomorphism, such as subgraph isomorphism or the largest common subgraph. Subgraph isomorphism (see Definition 2.12) is analogous to graph isomorphism but it could be used also when two graphs have different sizes. Unlike, the graph isomorphism problem, the subgraph isomorphism problem has been proven to be NP-complete [Garey and Johnson, 1979, Section 3.2.1]. A similarity measure can also be defined based on the size of the largest common subgraph in two graphs. Unfortunately, also the problem of finding the maximum common subgraph is known to be NP-hard [Garey and Johnson, 1979, Section 3.3].

2.2.2 Graph edit distances

Apart from being computationally expensive, similarity measures based on concepts of isomorphism have another disadvantage. They are too restrictive in the sense that the graphs have to be exactly identical or they should share large identical subgraphs, in order to be consider similar. This is an important problem when we produce graphs from noisy data. More flexible similarity measures that have been proposed in the literature as part of the inexact graph matching problem are similarity measures based on graph edit distances (GED) [Gao et al., 2010]. These GED algorithms are based on the concept of transformation a graph to another one by a finite sequence of graph edit operations, such as node addition or deletion, edge addition or deletion and node or edge relabeling. These operations can have different costs and the similarity measure is defined by the least-cost edit operation sequence. Figure 2.7 illustrates an example of graph edit distance methodology for a pair of node labeled graphs. Unfortunately, finding the optimal cost for a particular application is a hard problem, since it requires solving NP-complete problems as intermediate steps.

2.3 Graph kernel methods

As we have seen in Section 2.2 the first approaches proposed to solve the graph comparison problem suffer from intractable computational time, since in the worst case they require exponential runtime, or are hard to parametrize. Another family of approaches, that have been introduced the past few years, come from the statistical learning perspective. This family of approaches, known as *Graph Kernels*, tackles both the problem

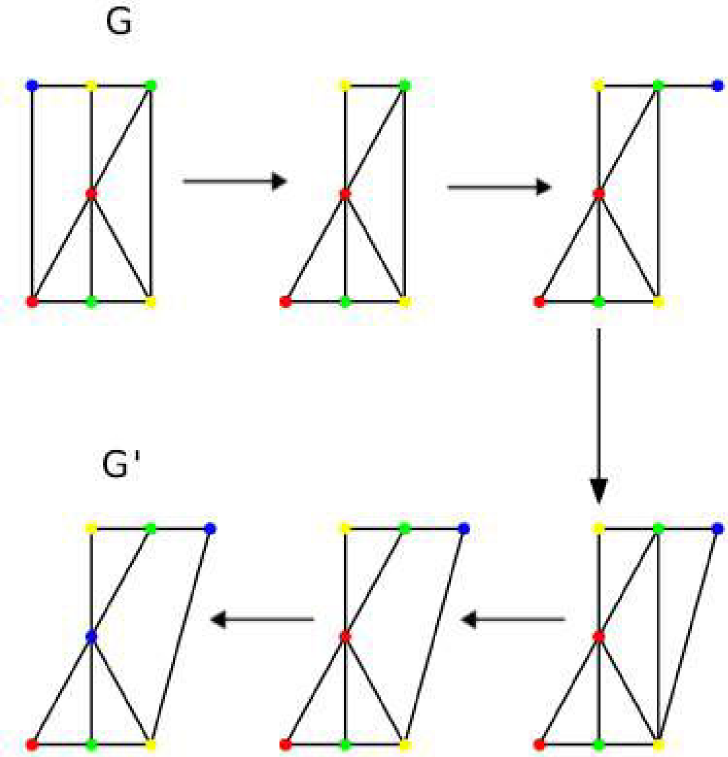


FIGURE 2.7: Example of the graph edit distances methods. Given two labeled graphs $G, G^{'}$ a sequence of graph edit operations are applied to transform G to $G^{'}$. More specifically, the first operation is node deletion with its adjacency edges, the second operation is node addition, then edge addition, followed by an edge deletion and a node relabeling.

of graph representation and graph comparison through the exploitation of the graph topology by decomposing the graph into substructures and aggregating statistics over these substructures. This strategy considers a measure of similarity between the graphs as a form of inner product.

Graph kernels are instances of the family of the R-convolution kernels by [Haussler, 1999]. R-convolution kernels are a generic way of constructing kernels on a set whose elements are discrete structures such as strings, trees and graphs. The idea is based of decomposing the complex object into discrete structures and compare the respective objects by comparing all pairs of decompositions. Every new decomposition in the graphs would yield a new graph kernel. A first approach would be to decompose the graphs into all possible subgraphs. However, calculating all subgraphs is at least as hard

Table 2.1: An overview of the graph kernel methods. From left to right we show the type of subgraphs used, the algorithm, its complexity when that is known and whether the kernel works on unlabeled, discretely, continuous or vector labeled graphs. Note that n is the number of graphs under comparison, v is the maximal number of nodes, e is the maximal number of edges, h is the height of subtree patterns, d is the maximum degree and k is the size of graphlets.

	Algorithm	**Complexity**	**Unlabeled**	**Discrete**	**Continuous**	**Vector**
Walks	[Gärtner et al., 2003]	$\mathcal{O}(n^2v^6)$	✓	✓	✓	✓
	[Mahé et al., 2004]		✓	✓		
	[Vishwanathan et al., 2010]	$\mathcal{O}(n^2v^3)$	✓	✓	✓	✓
Paths	[Borgwardt and Kriegel, 2005]	$\mathcal{O}(n^2v^4)$	✓	✓	✓	✓
	[Ralaivola et al., 2005]		✓	✓		
Graph lets	[Horváth et al., 2004]		✓	✓		
	[Shervashidze et al., 2009]	$\mathcal{O}(vd^{k-1})$	✓			
	[Costa and De Grave, 2010]		✓	✓		
Subtree Pat- terns	[Ramon and Gaertner, 2003]	$\mathcal{O}(n^2v^2h4^d)$	✓	✓		
	[Bach, 2008]		✓	✓		
	[Mahé and Vert, 2009]		✓	✓		
	[Shervashidze et al., 2011]	$\mathcal{O}(nhe + n^2hv)$	✓	✓		

as deciding whether two graphs are isomorphic [Gärtner et al., 2003]. As a result, it is necessary to limit the decomposition of the graphs only into specific types of subgraphs that are computable in polynomial time [Vishwanathan et al., 2010, Shervashidze et al., 2011].

There are three main categories of graph kernels, graph kernels based on (a) walks [Gärtner et al., 2003] and paths [Borgwardt and Kriegel, 2005], (b) small size subgraphs [Shervashidze et al., 2009] and (c) subtree patterns [Shervashidze et al., 2011]. Table 2.1 shows a summary of the state of the art of graph kernels grouped per category, while we review extensively each of these categories in the following Sections 2.3.1, 2.3.2 and 2.3.3.

2.3.1 Graph kernels based on walks and paths

Graph kernels based on walks and paths (see Definition 2.7) count the number of matching pairs of walks and paths in two graphs respectively. Different proposed kernels use different methods to compute similarities between walks and paths. For example, Gärtner *et al.* propose a random walk kernel that counts the number of nodes in the walk which have the same label, which requires $O(v^6)$ computational time for a pair of graphs [Gärtner et al., 2003], where v is the number of vertices in the graphs. Vishwanathan in [Vishwanathan et al., 2010] reduced the runtime complexity of the random walk kernel for a pair of graphs to $O(v^3)$ by restating the problem in terms of Kronecker products. Although, this is an important gain in efficiency, allowing to compute kernel on random walks faster by an order of magnitude, the complexity $O(v^3)$ is still too high for many applications. Apart from the computation time, kernels based on random walks have to deal with two extra problems, the *tottering problem* and the *halting problem*.

The tottering problem [Mahé et al., 2004] raises from the fact that walks allow the repetitions of nodes and edges, which means that the same node or edge can contribute repeatedly in the similarity measure. The same can be said for shared cycles or paths as well. Therefore, the similarity score between two graphs can drastically increase, although they two graphs do not share many structural elements. The halting problem [Borgwardt, 2007] also arises from the fact that walks allow the repetitions of nodes. As the repetition of nodes is allowed, the number of walks within a graph is infinitely large. In order to halt the problem, a decaying factor is commonly used to downweight longer walks. The effect of this decaying factor is that longer walks are completely neglected compared to shorter walks.

The marginalized graph kernel [Mahé et al., 2004] proposed two extensions on the random walk kernels to overcome both the tottering problem and reduce their computation time. They modify the label of each vertex with the use of the Morgan index [Morgan, 1965], which is defined as

Definition 2.14 (Morgan Index). Given a graph $G = (V, E)$, the Morgan index of order k for node $v \in V$ is defined as

$$M_k(G, v) = \begin{cases} 1 & \text{if } k = 0 \\ \sum_{v' \in N(v)} M_{k-1}(v') & \text{otherwise.} \end{cases} \tag{2.2}$$

Note that the Morgan index of order k for a node v is the number of walks of length k starting at v in that graph G. So incorporating the Morgan index into the label of each vertex, it increases the specificity of labels by adding information with the number of walks staring at that vertex. In addition, they proposed a modification to prevent the walk from coming back to a vertex that was just visited. Another approach based on dynamic programming to speed up the computations of the random walk kernel was proposed in [Harchaoui and Bach, 2007], at the cost of considering only walks of fixed size.

[Borgwardt and Kriegel, 2005] proposed a graph kernel that compares the length of shortest path between pairs of nodes with matching source and sink labels in two graphs, which requires $O(v^4)$ complexity time. Ralaivola in [Ralaivola et al., 2005] proposed a specialized graph kernel for chemoinformatics. Their approach is based on molecular fingerprinting techniques and counts labeled paths of length p that can be retrieved by depth-first search from each vertex. This can be an efficient approach for graphs with an average node degree of 2 or 3.

2.3.2 Graph kernels on small size subgraphs

A number of graph kernels have been proposed in the literature that are based on limited size subgraph structures called *graphlets*. A naive computation of all graphlets of a graph, without considering labels, requires $O(v^k)$ computations, where v is the number of nodes and k is the size of subgraphs, usually $k \in \{3, 4, 5\}$. Since enumerating all graphlets is prohibitively expensive, even for small k values, [Shervashidze et al., 2009] showed that sampling a fixed number of graphlets suffices to bound the deviation of the empirical estimates of the graphlet distribution from the true distribution and for graphs of degree bounded by d, the exact number of all graphlets of size k can be determined in time $O(vd^{k-1})$. Another kernel is the cyclic pattern kernels [Horváth et al., 2004], which counts pairs of matching cyclic and tree patterns in two graphs. In the general case, the cyclic pattern kernel is NP-hard, but in specific cases the kernel can be computed efficiently. Finally [Costa and De Grave, 2010] proposed the neighborhood subgraph pairwise distance kernel, which decomposes a graph into all pairs of neighborhood subgraphs of small radius r at increasing distances d.

2.3.3 Graph kernels on subtree patterns

In 2003, Ramon and Gärtner were the first to introduce a graph kernel based on subtree patterns [Ramon and Gaertner, 2003]. The Ramon-Gärtner subtree kernel with subtree height h compares all pairs of nodes from two labeled graphs by iteratively comparing their neighborhoods. Although the subtree kernel is more expressive than kernels based on walks, unfortunately it is computationally expensive. For a set of n graphs it requires $O(n^2v^2h4^d)$, where v is the number of nodes, h is the height of subtree patterns considered and d is the maximum node degree in the graph set. Both subtree kernels by Mahé and Vert [Mahé and Vert, 2009] and Bach [Bach, 2008] refine the Ramon-Gärtner subtree kernel for application in chemoinformatics and hand-written digit recognition respectively. In [Mahé and Vert, 2009], Mahé and Vert proposed a new kernel with a parameter to control the complexity of the subtrees used as features to represent the graphs. This parameter allows to smoothly combine graph kernels based on walks and kernels based on subtrees. In [Bach, 2008], Bach proposed a graph kernel that considers α-ary subtrees with most α children per node. Unfortunately, the complexity of both kernels are still exponential in the smoothing and a parameter respectively, and both kernels are feasible on small size graphs only. Recently a new kernel with subtree patterns was introduced by Shervashidze [Shervashidze et al., 2011]. The Weisfeiler-Lehman subtree kernel uses the Weisfeiler-Lehman test of isomorphism [Weisfeiler and Lehman, 1968] to efficiently compute subtree patterns up to height h for discretely labeled graphs.

For n pairs of discretely labeled graphs, the Weisfeiler-Lehman subtree kernel requires $O(nhe + n^2hv)$, where e is the maximal number of edges, v is the maximal number of vertices and considers subtree patterns up to height h.

Chapter 3

The pyramid quantized Weisfeiler-Lehman graph representation

Chapter 2 presented the problem of graph comparison in detail. The majority of the methods focus on either unlabeled or discretely labeled graphs, while an efficient and expressive representation and comparison of graphs with complex labels, such as real numbers and high-dimensional vectors, remains an open research problem.

In this chapter we introduce a novel method, *the pyramid quantized Weisfeiler-Lehman graph representation* that compares labeled graphs with complex labels. Our method makes use of the efficiency of the Weisfeiler-Lehman kernel [Shervashidze et al., 2011] for discrete labels by considering a pyramid quantization strategy that approximates the continuous or vector labeled graphs with a sequence of discretely labeled graphs.

Firstly, we introduce the Weisfeiler-Lehman test of isomorphism, and then we explore how key concepts of the test is used in the framework of the Weisfeiler-Lehman kernel for comparing discrete labeled graphs in Section 3.1 and in Section 3.2, respectively. Then in Section 3.3 we present our pyramid quantization scheme and we conclude by exploring different strategies for combing the pyramid levels in Section 3.4.

3.1 The Weisfeiler-Lehman test of isomorphism

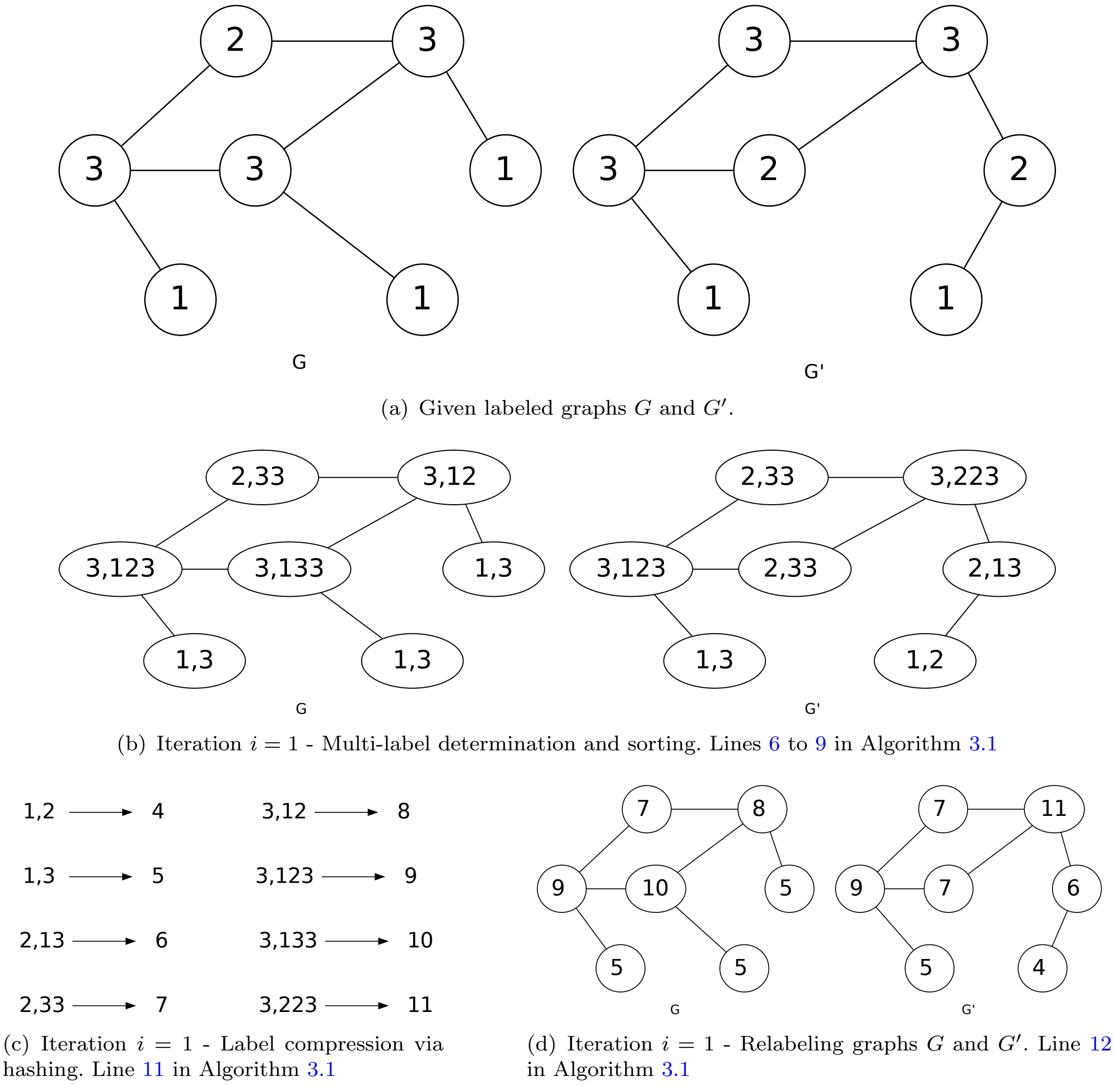
Our proposed algorithm uses the statistics introduced by the Weisfeiler-Lehman kernel, which exploits the key concepts from the Weisfeiler-Lehman test of isomorphism [Weisfeiler and Lehman, 1968] and more specifically its one dimensional variant. Given

Algorithm 3.1 The one dimensional Weisfeiler-Lehman test of graph isomorphism

```
Require: Two graphs, $G = (V, E, \mathcal{L})$, $G' = (V', E', \mathcal{L}')$, with discrete labelings $\mathcal{L} : V \mapsto \Sigma$ and $\mathcal{L}' : V' \mapsto \Sigma$ over vertices, where $\Sigma$ is a vertex label set and the maximum number of iterations $h$.
$i \leftarrow 0$
repeat
  if $i = 0$ then
    {________ Multiset-label initialization ________}
    $M_i(v) := l_0(v) = \mathcal{L}(v)$.
  else if $i \geq 1$ then
    {________ Multiset-label determination ________}
    Assign a multiset-label $M_i(v)$ to each node $v$ in $G$ and $G'$ which consists of the multiset $\{l_{i-1}(u)|u \in \mathcal{N}(v)\}$, where $\mathcal{N}(v)$ denotes the neighbor set of $v$.
    {________Sorting each multiset ________}
    Sort the elements in $M_i(v)$ in ascending order.
    Concatenate the elements in $M_i(v)$ into a string $s_i(v)$.
    Add $l_{i-1}(v)$ as a prefix to $s_i(v)$.
    {________ Sorting the set of multisets ________}
    Sort all of the strings $s_i(v)$ for all $v$ from $G$ and $G'$ in ascending order.
    {________ Label compression via hashing ________}
    Map each string $s_i(v)$ to a new compressed label using a function $f : \Sigma^* \mapsto \Sigma$ such that $f(s_i(v)) = f(s_i(w)) \iff s_i(v) = s_i(w)$.
    {________ Relabeling ________}
    Set $l_i(v) := f(s_i(v))$ for all nodes in $G$ and $G'$.
  end if
  $i \leftarrow i + 1$
until $\{l_i(v)|v \in V\} \neq \{l_i(v')|v' \in V'\}$ or $i > h$
```

two graphs G and G' the Weisfeiler-Lehman test of isomorphism determines whether they are isomorphic or not. Algorithm 3.1 provides pseudocode for the one dimensional Weisfeiler-Lehman test of graph isomorphism.

The key idea of the Weisfeiler-Lehman test of graph isomorphism is the construction of augmented node labels from all the neighbor nodes and the compression into new short labels. This process is repeated until either the label sets of the two graphs under comparison differ or the maximum number of iterations has been reached. If the two label sets differ, then the two graphs are non-isomorphic, while if the maximum number of iterations is reached and the two label sets don't differ, then the test was not able to determine that they are not isomorphic. [Cai et al., 1989] provide examples of graphs that cannot be distinguished by this algorithm or its higher-dimensional variants. Figure 3.1 shows all steps of the Weisfeiler-Lehman test of graph isomorphism for iteration $i = 1$ given the two labeled graphs G and G' shown in Figure 3.1(a). Note that in this figure the nodes in the two graphs have been initially labeled by their corresponding degree of node $d(v)$ (see Definition 2.5). Moreover the two graphs G and G' would directly be identified as non-isomorphic by the Weisfeiler-Lehman test, as their label sets differ from the beginning.



(a) Given labeled graphs G and G'.

(b) Iteration $i = 1$ - Multi-label determination and sorting. Lines 6 to 9 in Algorithm 3.1

(c) Iteration $i = 1$ - Label compression via hashing. Line 11 in Algorithm 3.1

(d) Iteration $i = 1$ - Relabeling graphs G and G'. Line 12 in Algorithm 3.1

FIGURE 3.1: An illustration of the computation of the Weisfeiler-Lehman test of graph isomorphism for the iteration $i = 1$. Note that the label of each node v is its degree of node $d(v)$.

A straightforward definition and implementation of the mapping function $f : \Sigma^* \to \Sigma$, in order to be an injective function, is to sort all the strings $s_i(v)$ $\forall v \in \{V \cup V'\}$ and to keep a counter variable that records the number of unique strings that f has already compressed. So f assigns the current value of the aforementioned counter to a string when the string has already been compressed, while when a new string occurs, we increment the counter by one and assign the new value as a compressed label. The sorting of the elements with each multiset $M_i(v)$ (Line 7 in Algorithm 3.1) allows us to define the same string $s_i(v)$ for two nodes that share the same connectivity and label pattern independent of the order of accessing their respective neighbor nodes in the construction of the multiset, while the sorting of the strings $s_i(v)$ (Line 10 in Algorithm 3.1) guarantees

Algorithm 3.2 Sorting each multiset at iteration i

```
for all graphs $G = (V, E)$ do
  for all nodes $u \in V$ do
    for all nodes $v \in N(u)$ do
      append the pair $(G, u)$ to bucket $l_{i-1}(v)$. [1]
    end for
    $s_i(u) \leftarrow l_{i-1}(u)$
  end for
end for
for $k = 1 \rightarrow |\Sigma^*|$ do
  for all $(G, u)$ in bucket $k$ do
    append $k$ to $s_i(u)$ in $G$
  end for
end for
```

that all identical strings will be mapped to the same compressed labels, as they occur in blocks. This implementation requires that the alphabet Σ has to be sufficiently large in order for f to be injective. For two graphs $G = (V, E)$ and $G' = (V', E')$ of order $|V| = |V'| = v$, $|\Sigma| = 2v$ is sufficient. Of course any other injective mapping could be used and will give an equivalent result. Finally, it should be noted that in Algorithm 3.1 the same node labeling function $l_0, \ldots, l_h$ has been used form both G and G' in order not to overlap the notation. The same simplified notation will be used through the chapter assuming without loss of generality that the domain of these functions $l_0, \ldots, l_h$ is the set of all nodes in the input graphs. In the case of Algorithm 3.1 the domain is defined in $V \cup V'$

Complexity The efficiency of the compression of the labels via the hashing scheme described above depends on the complexity of the sorting method. Given the fact that the labels of the graphs are discrete and their cardinality is upper-bounded by $|\Sigma| = 2v$, the counting sort algorithm is appropriate for sorting the multisets. The counting sort algorithm has a complexity of $O(n + k)$, where n is the number of elements to be sorted and k the number of buckets. In our case the number of elements to be sorted for the multisets $M_i(v)$ (Line 7 in Algorithm 3.1) is in the worst case linear to the maximal number of edges e and if we select a $k = O(e)$, we end up with a complexity of $O(e)$. Analytical pseudocode for the counting algorithm is provided in Algorithm 3.2. Sorting the resulting strings $s_i(v)$ (Line 10 in Algorithm 3.1) is also of time complexity $O(e)$ via the radix sort algorithm. As the Weisfeiler-Lehman test runs in h iterations the total runtime is $O(he)$.

[1] Note that in the pair (G, u) by G we declare the identifier of the graph G in the graph dataset.

Link with subtree patterns There is a strong link between the compressed labels and subtree patterns (see Definition 2.10). More specifically a compressed label $l_i(v)$ corresponds to a subtree pattern rooted at node v of height i. For example, in Figure 3.1 if a node has a new compressed label, 9, this means that there is a subtree pattern of height 1 rooted at this node, where this root node has label 3 and its neighbor nodes have labels 1, 2 and 3 respectively.

3.2 The linear Weisfeiler-Lehman subtree kernel

As mentioned above in Section 3.1 the Weisfeiler-Lehman test of graph isomorphism is iterative. For each iteration i, we obtain new compressed labels $l_i(v)$ for all nodes v as we have seen in Line 12 in Algorithm 3.1. We emphasize that the labeling is concordant between the graphs under comparison, that means if and only if the nodes in G and G' have identical string $s_i(v)$, they will get identical new labels $l_i(v)$. These compressed labels, *i.e.* the subtree patterns, have been recently employed in a kernel for graph comparison, the Weisfeiler-Lehman subtree kernel [Shervashidze et al., 2011, Definition 4] which is defined as follows:

Definition 3.1 (The linear Weisfeiler-Lehman subtree kernel). Let G and G' be graphs. Define $\Sigma_i \subseteq \Sigma$ as the set of symbols that occur as node labels at least once in G or G' at the end of the i-th iteration of the Weisfeiler-Lehman algorithm. Let Σ_0 be the set of original node labels of G and G'. Assume all Σ_i are pairwise disjoint. Without loss of generality, assume that every $\Sigma_i = \{\sigma_{i1}, \ldots, \sigma_{i|\Sigma_i|}\}$ is ordered. Define a map $\eta_i : \{G, G'\} \times \Sigma_i \to \mathbb{N}$ such that $\eta_i(G, \sigma_{ij})$ is the number of occurrences of the letter σ_{ij} in the graph G.

The linear Weisfeiler-Lehman subtree kernel on two graphs G and G' with h iterations is defined as

$$k^{(h)}_{l-WLsubtree}(G, G') = \langle \phi_{(h)}(G), \phi_{(h)}(G') \rangle \tag{3.1}$$

where

$$\phi_{(h)}(G) = \left(\eta_0(G, \sigma_{01}), \ldots, \eta_0(G, \sigma_{0|\Sigma_0|}), \ldots, \eta_h(G, \sigma_{h1}), \ldots, \eta_h(G, \sigma_{h|\Sigma_h|})\right) \tag{3.2}$$

and

$$\phi_{(h)}(G') = \left(\eta_0(G', \sigma_{01}), \ldots, \eta_0(G', \sigma_{0|\Sigma_0|}), \ldots, \eta_h(G', \sigma_{h1}), \ldots, \eta_h(G', \sigma_{h|\Sigma_h|})\right) \tag{3.3}$$

TABLE 3.1: An example of the linear Weifeiler-Lehman subtree kernel between the two graphs shown in Figure 3.1(a) for subtree patterns up to depth $h = 1$. The first row shows the labels encountered up to depth $h = 1$, which contains the original node labels Σ_0 as well as the compressed node labels Σ_1 for the iteration $i = 1$. The second and the third row contains the histogram over the labels for graph G and G' respectively, while the fourth row shows the final kernel value between the given graphs.

$$\begin{array}{rcl}
\text{Labels } \{\Sigma_0, \Sigma_1\} & = & \{\overbrace{1, \quad 2, \quad 3,}^{\text{Original node labels } \Sigma_0} \quad \overbrace{4, \quad 5, \quad 6, \quad 7, \quad 8, \quad 9, \quad 10, \quad 11}^{\text{Compressed node labels } \Sigma_1}\} \\
\phi_{(1)}(G) & = & (3, \quad 1, \quad 3, \quad 0, \quad 3, \quad 0, \quad 1, \quad 1, \quad 1, \quad 1, \quad 0) \\
\phi_{(1)}(G') & = & (2, \quad 2, \quad 3, \quad 1, \quad 1, \quad 1, \quad 2, \quad 0, \quad 1, \quad 0, \quad 1) \\
k^{(1)}_{l-WLsubtree}(G, G') & = & \left\langle \phi_{(h)}(G), \phi_{(h)}(G') \right\rangle = 23
\end{array}$$

Note that the features $\phi_{(h)}$ are histograms of the original and compressed labels, *i.e.* histograms of subtree patterns of depths $0, \ldots, h$. Table 3.1 shows an example of the linear Weisfeiler-Lehman subtree kernel $k^{(1)}_{l-WLsubtree}(G, G')$ and their respective features $\phi_{(1)}$ between the two graph G, G' shown in Figure 3.1(a) for subtree patterns up to depth $h = 1$.

Complexity A key advantage of these statistics $\phi_{(h)}(G)$ is that they are computable in linear time in the number of edges in the graphs and in the depth of the subtree patterns. More specifically, for n graphs and subtree patterns of depth up to h, the complexity of a joint computation of all statistics over all graphs is $\mathcal{O}(nhe + n^2hv)$, where e is the maximal number of edges, and v the maximal number of vertices [Shervashidze et al., 2011, Theorem 7]. This complexity can be achieved by processing all n graphs simultaneously, meaning conducting the steps of multiset label determination, sorting each multiset, label compression and relabeling of Algorithm 3.1 for all n graphs simultaneously. After we have calculate the $\phi(h)$ feature explicitly on each graph G, we can calculate the pairwise inner products efficiently. Moreover the use of an efficient hashing scheme allows the algorithm to enumerate relevant (non-zero) dimensions of an exponentially sized feature space efficiently. In this way, the matching can be done in constant time, and the hash needs only to store patterns present in the graph instances, thereby maintaining constant complexity across iterations. As before, in order for the mapping scheme f to be injective, a sufficiently large label set Σ is required. In the case of n graphs and h iterations, $|\Sigma| = nv(h + 1)$ suffices.

Memory requirements in practice The memory requirements of these statistics $\phi_{(h)}(G)$ in practice depends on the used learning setting. In the case of an inductive learning setting, where initially a kernel is computed over only the training set of graphs, in order to classify any test graph the explicit mapping scheme f occurred in the training set is required. That means that one needs to maintain record of all the mappings $l_i(v) = f(s_i(v))$ for each iteration i and for each distinct $s_i(v)$, ending with a $O(neh)$ memory in the worst case. In contrast, if a trasductive setting is used, where the test set is already known, the kernel matrix of the whole date set (both training and testing) can be computed without having to keep the mapping f, minimizing the memory requirements.

In addition to these computational benefits, linear Weisfeiler-Lehman graph kernels have been shown to perform comparably to or better than a number of more computationally complex kernels [Shervashidze et al., 2011]. Finally, we note that the linear Weisfeiler-Lehman algorithm at depth 0 computes exactly the bag of words representation commonly used in natural language processing [Harris, 1954, Ko, 2012] and computer vision [Qiu, Fei-fei, 2005].

3.3 The pyramid quantization strategy for continuous labels

The Weisfeiler-Lehman algorithm is efficient precisely because it makes use of a discrete labeling over nodes, which enables an efficient hashing scheme in order to scale linearly in the number of edges and in the height of subtree patterns. A problem occurs when extending this method to continuous vector labeled graphs. To overcome this, we propose a pyramid quantization strategy similar to the one used by [Grauman and Darrell, 2007a,b] to determine a logarithmic number of discrete labelings with increasing granularity for which we run the Weisfeiler-Lehman algorithm. In other words, we approximate a graph representation with continuous valued labels as a sequence of graphs with discrete labels of increasing granularity.

3.3.1 The pyramid quantization strategy

Given a vector labeled graph $G = (V, E, \mathcal{L})$, where $\mathcal{L} : V \to \mathbb{R}^d$ is the function assigning a d-dimensional vector label to each vertex, we want to derive a hierarchical decomposition of $\mathbb{R}^d$ as multi-resolution quantizations. The multi-resolution quantizations will then be used to determine the discrete labeling of increasing granularity. This can be expressed as a two step process, first we construct a quantization function $Q^{(l)} : \mathbb{R}^d \to \Sigma_0^{(l)}$ that will encode the $\mathbb{R}^d$ into a quantization of a given resolution $|\Sigma_0^{(l)}| = 2^l$. The quantization function $Q^{(l)}$ is repeated for $l \in \{0, \ldots, L\}$ to determine multi-resolutions of increasing granularity, where $L = \lceil \log_2 D \rceil$, $D \leq |V| = v$ is the number of unique values in the image[2] of the set V under $\mathcal{L}$. Note that the single quantization bin for $Q^{(0)}$ is big enough so that all data points from the image of the set V under $\mathcal{L}$ receive the same discrete label, while as quantization resolution moves from coarser to finer, we end up with $Q^{(L)}$ that contains quantization bins that are small enough so each unique data point from the image of the set V under $\mathcal{L}$ falls into its own quantization bin.

The second step is to compose the quantization function $Q^{(l)}$ with the labeling function $\mathcal{L}$, $\forall l \in \{0, \dots, L\}$, so we can approximate our initial vector labeled graph G as a sequence of graphs with discrete labels of increasing granularity:

$$G = (V, E, \mathcal{L}) \overset{Q^{(l)} \circ \mathcal{L}}{\approx} \left(G^{(0)}, \dots, G^{(L)} \right) = \left((V, E, \mathcal{L}^{(0)}), \dots, (V, E, \mathcal{L}^{(L)}) \right), \tag{3.4}$$

where $\mathcal{L}^{(l)} : V \to \Sigma_0^{(l)}$ is defined to be $Q^{(l)} \circ \mathcal{L}$, and $\Sigma_0^{(l)}$ is the discrete label alphabet for a given level l of quantization. Note that the topology of the graph does not change in the sequence of graphs, only the continuous labels are discretized. To achieve the discretization of the labels, we explore two different strategies for the quantization function: (a) a fixed binning scheme and (b) a data guided one in the following sections.

3.3.1.1 Fixed Binning

In the fixed binning scheme the quantization function recursively decomposed into quantization resolution, where in each quantization level the bins that partition the space are half the size in all d dimensions of the input space compared to the previous one. The number of bin of each $Q^{(l)})$ is given by $r^{(l)} = \left(\frac{D}{2^l \sqrt{d}}\right)^d$, where D is the number of unique values in the image of the vertex set V under label function $\mathcal{L}$. The fixed binning scheme can be performed efficiently in high dimensions using, e.g. k-d trees [Bentley, 1975]. The complexity of the resulting quantization is bounded by $\mathcal{O}(d \max(v, k) L)$, where d is the dimension of the input space, v is the number of vertices to be quantized, k is the maximum histogram index value in a single dimension and L is the number of pyramid levels [Grauman and Darrell, 2007a]. With constraining $k \leq v$ and ensuring that L is logarithmic in v, we end up with a simplified complexity of $\mathcal{O}(dv \log v)$.

3.3.1.2 Data guided binning

The idea of data guided binning is to derive a hierarchical but *data-dependent* decomposition of the feature space that will encode the multi-dimensional features as multi-resolution histograms with non-uniformly shaped bins. The first step in this scheme

[2] Note that when we are interested in the quantization of the $\mathbb{R}^d$ given a set of vector labeled graphs $\mathbf{G} = \{G_i = (V_i, E_i, \mathcal{L}_i)\}_{1 \leq i \leq n}$, where $\mathcal{L}_i : V_i \to \mathbb{R}^d$, then D is number of unique values in the union of the images of the label functions $\mathcal{L}_i$ $\forall i \in \{1, \dots, n\}$.

is to generate the structure of the data guided pyramid hierarchy that will define the bin placements. To achieve this we perform an agglomerative hierarchical clustering on image of the vertex set V under the labeling function $\mathcal{L}$ using the Ward's minimum variance method [Ward, 1963].[3] Ward's methods minimizes the total within-cluster sum of squares criterion which is defined as follow for two clusters i, j:

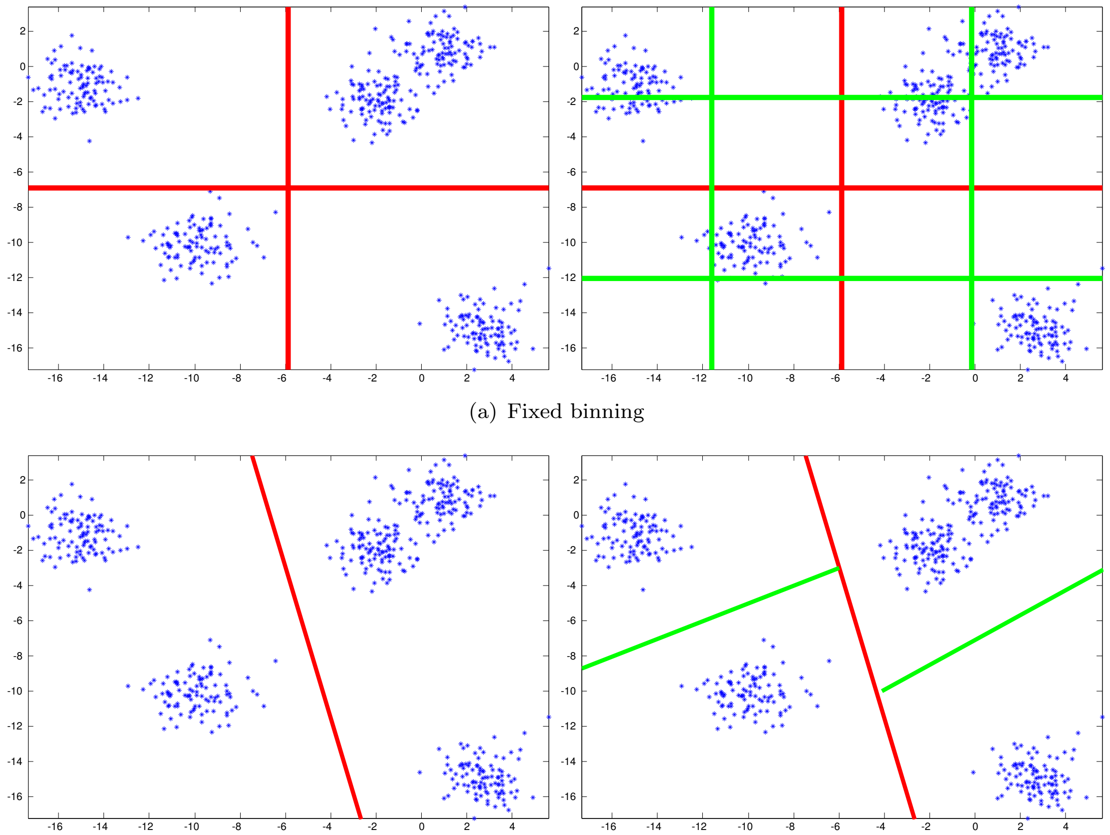
$$d(i,j) = \sqrt{\frac{2n_i n_j}{(n_i + n_j)}} \|\overline{x}_i - \overline{x}_j\|_2 \tag{3.5}$$

where n_i and n_j are the number of elements in clusters i and j, $\overline{x}_i$ and $\overline{x}_j$ are the centroids of clusters i and j and $\| \cdot \|_2$ denotes the Euclidean distance. At the initial step, all clusters are singletons (*i.e.* clusters containing a single unique point). Then we apply the algorithm recursively and at each step the pair of clusters with minimum between-cluster distance given from Equation 3.5 are merged.

Once the data guided bins have been constructed (*i.e.* the centroid of each cluster in all L levels has been determined), we can embed each d-dimensional data point $\mathbf{c}_i \in \mathbb{R}^d \; \forall i$ to the multi-resolution quantization bins of L levels, where in each level l there are 2^l bins. In order for a point $\mathbf{c}_i$ to be mapped at the correct bins at the quantization pyramid, we need to compare it to the two appropriate centroids at each l pyramid level using the Euclidean distance and pushed down the hierarchical tree along that branch that is rooted with the closest centroid at each level. At each comparison with the centroids, we also keep a record of a binary bin index p with its path along the quantization pyramid. As the quantization pyramid has L level, in total $2L$ distances must be computed between a point and the pyramid's bin centroids. Concerning the space requirements, this approach requires $O(2^L) = O(v)$ d-dimensional feature vectors to store for all the bin centroids of the hierarchical pyramid, since L is logarithmic in v, and $O(L) = O(\log_2 v)$ binary indexes for a set of v feature vectors. Note that the bin centroids are calculated only once using the Ward's method during the training. Finally, any other hierarchical clustering technique, could also be used without changing the main idea.

In Figure 3.2, we illustrate the difference between the two pyramid quantization strategies defined above for the same 2D space. In Figure 3.2(a) we see that the space is partitioned into uniform-shaped bins, while in Figure 3.2(b) the data themselves determine the partitioning, as a result the bins better decompose the space into clusters.

[3] In practice when the input space is very large, we can randomly select a subset of representative data to perform the agglomerative hierarchical clustering.



(a) Fixed binning

(b) Data guided binning

FIGURE 3.2: An illustration of the two different strategies of quantization of a complex space. Both Figures 3.2(a) and 3.2(b) depict the space partition boundaries for two resolution levels for the same 2D space. In both figures, the left plot contains the coarser resolution level, while the right plot contains the finer one. In Figure 3.2(a) the complex space is carved into uniformly-shaped partitions, while in Figure 3.2(b), the data guide the resulting partition of the complex space. As a result, the bins on the right are better positioned to decompose the space based on the data clusters.

3.3.2 The intersection Weisfeiler-Lehman subtree kernel

Independent of the binning strategy we follow, each graph with vector valued labels after the pyramid quantization step described in Section 3.3.1 is represented as a sequence of graphs with nested quantizations of increasing granularity of discrete labels as described in Equation 3.4. We run the Weifeiler-Lehman algorithm on each graph $G^{(l)}$ of the sequence in order to produce the features $\phi_{(h)}^{(l)}$ of subtree patterns up to a given height h.

Definition 3.2 (The intersection Weisfeiler-Lehman subtree kernel).
Let $G^{(l)} = (V, E, \mathcal{L}^{(l)})$ and $G'^{(l)} = (V', E', \mathcal{L}'^{(l)})$ be two graphs of the same quantization level l, where $\mathcal{L}^{(l)} : V \to \Sigma_0^{(l)}$ and $\mathcal{L}'^{(l)} : V' \to \Sigma_0^{(l)}$, of two vector labeled graphs $G = (V, E, \mathcal{L})$ and $G' = (V', E', \mathcal{L}')$, where $\mathcal{L} : V \to \mathbb{R}^d$ and $\mathcal{L}' : V' \to \mathbb{R}^d$. Define $\Sigma_i^{(l)} \subseteq \Sigma^{(l)}$ as the set of symbols that occur as node labels at least once in $G^{(l)}$ or $G'^{(l)}$

at the end of the i-th iteration of the Weisfeiler-Lehman algorithm. Let $\Sigma_0^{(l)}$ be the set of original node labels of $G^{(l)}$ and $G'^{(l)}$. Assume all $\Sigma_i^{(l)}$ are pairwise disjoint. Without loss of generality, assume that every $\Sigma_i^{(l)} = \{\sigma_{i1}^{(l)}, \ldots, \sigma_{i|\Sigma_i^{(l)}|}^{(l)}\}$ is ordered. Define a map $\eta_i : \{G^{(l)}, G'^{(l)}\} \times \Sigma_i^{(l)} \rightarrow \mathbb{N}$ such that $\eta_i(G^{(l)}, \sigma_{ij}^{(l)})$ is the number of occurrences of the letter $\sigma_{ij}^{(l)}$ in the graph $G^{(l)}$.

The intersection Weisfeiler-Lehman subtree kernel on two graphs G and G' with h iterations is defined as

$$k_{i-WLsubtree}^{(h)}(G^{(l)}, G'^{(l)}) = \mathcal{I}\left(\phi_{(h)}^{(l)}(G^{(l)}), \phi_{(h)}^{(l)}(G'^{(l)})\right) \tag{3.6}$$

where

$$\phi_{(h)}^{(l)}(G^{(l)}) = \left(\eta_0(G^{(l)}, \sigma_{01}^{(l)}), \ldots, \eta_0(G^{(l)}, \sigma_{0|\Sigma_0^{(l)}|}^{(l)}), \ldots, \eta_h(G^{(l)}, \sigma_{h1}^{(l)}), \ldots, \eta_h(G^{(l)}, \sigma_{h|\Sigma_h^{(l)}|}^{(l)})\right)$$

$$\phi_{(h)}^{(l)}(G'^{(l)}) = \left(\eta_0(G'^{(l)}, \sigma_{01}^{(l)}), \ldots, \eta_0(G'^{(l)}, \sigma_{0|\Sigma_0^{(l)}|}^{(l)}), \ldots, \eta_h(G'^{(l)}, \sigma_{h1}^{(l)}), \ldots, \eta_h(G'^{(l)}, \sigma_{h|\Sigma_h^{(l)}|}^{(l)})\right)$$

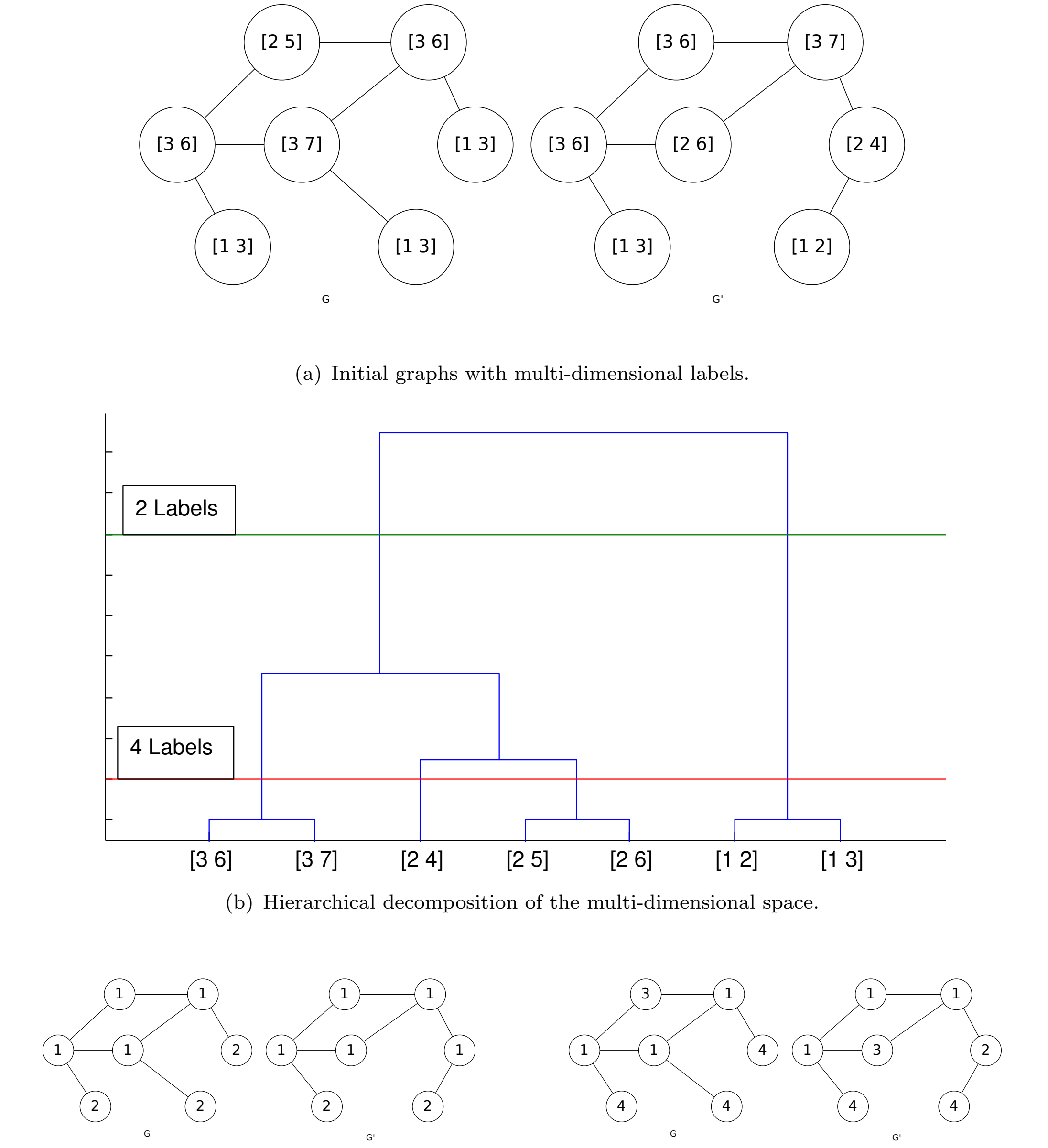
and

$$\mathcal{I}\left(\phi_{(h)}^{(l)}(G^{(l)}), \phi_{(h)}^{(l)}(G'^{(l)})\right) = \sum_{i=0}^{h} \sum_{j=1}^{|\Sigma_i^{(l)}|} \min\left(\eta_i(G^{(l)}, \sigma_{ij}^{(l)}), \eta_i(G'^{(l)}, \sigma_{ij}^{(l)})\right) \tag{3.7}$$

Note that the features $\phi_{(h)}^{(l)}$ are histograms of the original and compressed labels, *i.e.* histograms of subtree patterns of depths $0, \ldots, h$ for a given quantization level l, while the intersection Weisfeiler-Lehman subtree kernel counts the overlap of features $\phi_{(h)}^{(l)}$ between two graphs $G^{(l)} = (V, E, \mathcal{L}^{(l)})$ and $G'^{(l)} = (V', E', \mathcal{L}'^{(l)})$ which match at the given quantization level l. Note that the intersection Weisfeiler-Lehman kernel for a given binning resolution l is a positive-definite similarity function [Odone et al., 2005].[4]

An example of the quantization step of the pyramid quantized Weisfeiler-Lehman graph representation is illustrated in Figure 3.3. Given the graphs $G = (V, E, \mathcal{L}), G' = (V', E', \mathcal{L}')$ with continuous vector labels in their nodes, *i.e.* $\mathcal{L} : V \rightarrow \mathbb{R}^d$ and $\mathcal{L}' : V' \rightarrow \mathbb{R}^d$, shown in Figure 3.3(a), the first step is to determine the hierarchical decomposition of the labeled space given, for example, by the data guided binning strategy. The red and green line in Figure 3.3(b) depict the thresholds for achieving two different quantization resolutions with two and and four discrete labels, respectively. For these two quantization resolutions we end up with a sequence of discretized labels for each

[4] We could also use the linear kernel over the subtree patterns, but the histogram intersection kernel has been shown to give better results [Odone et al., 2005].



(a) Initial graphs with multi-dimensional labels.

(b) Hierarchical decomposition of the multi-dimensional space.

(c) Relabeled graphs of the quantization level 1 with $2^1 = 2$ number of discrete labels.

(d) Relabeled graphs of the quantization level 2 with $2^2 = 4$ number of discrete labels.

FIGURE 3.3: An illustration of the quantization step of the pyramid quantized Weisfeiler-Lehman graph representation. Figure 3.3(a) shows the two given multi-dimensional labeled graphs under comparison $G = (V, E, \mathcal{L})$ and $G' = (V', E', \mathcal{L}')$, where $\mathcal{L} : V \to \mathbb{R}^2$, $\mathcal{L}' : V' \to \mathbb{R}^2$ and the label of each node v is defined as its degree of node $d(v)$ and the summation of the degree of nodes of its neighborhoods N_v, *i.e.* $\mathcal{L}(v) = [d(v), \sum_{u \in N_v} d(u)]$. Figure 3.2(b) shows the hierarchical decomposition of the multi-dimensional labeled space given by the data guided binning strategy (see Section 3.3.1.2). With the green and red line are depicted the thresholds for achieving two different quantization resolutions with two and and four discrete labels respectively. Figure 3.3(c) shows the two relabeled discretized graphs $G^{(1)} = (V, E, \mathcal{L}^{(1)})$ and $G'^{(1)} = (V', E', \mathcal{L}'^{(1)})$, where $\mathcal{L}^{(1)} : V \to \Sigma_0^{(1)}$, $\mathcal{L}'^{(1)} : V' \to \Sigma_0^{(1)}$ and $|\Sigma_0^{(1)}| = 2$ is the coarser resolution level $l = 1$ with two labels, while Figure 3.3(d) shows the two relabeled discretized graphs $G^{(2)} = (V, E, \mathcal{L}^{(2)})$ and $G'^{(2)} = (V', E', \mathcal{L}'^{(2)})$, where $\mathcal{L}^{(2)} : V \to \Sigma_0^{(2)}$, $\mathcal{L}'^{(2)} : V' \to \Sigma_0^{(2)}$ and $|\Sigma_0^{(2)}| = 4$ is the finer resolution level $l = 2$ with four labels.

graph

$$G = (V, E, \mathcal{L}) \quad \underset{l \in \{1,2\}}{\overset{Q^{(l)} \circ \mathcal{L}}{\approx}} \quad \left(G^{(1)}, G^{(2)}\right) = \left((V, E, \mathcal{L}^{(1)}), (V, E, \mathcal{L}^{(2)})\right)$$

and

$$G' = (V', E', \mathcal{L}') \quad \underset{l \in \{1,2\}}{\overset{Q^{(l)} \circ \mathcal{L}'}{\approx}} \quad \left(G'^{(1)}, G'^{(2)}\right) = \left((V', E', \mathcal{L}'^{(1)}), (V', E', \mathcal{L}'^{(2)})\right)$$

also shown in Figure 3.3(c) and in Figure 3.3(d), where $\mathcal{L}^{(l)} : V \to \Sigma_0^{(l)}$ and $\mathcal{L}'^{(l)} : V' \to \Sigma_0^{(l)}$, while $|\Sigma_0^{(1)}| = 2$ is the coarser resolution with two labels and $|\Sigma_0^{(2)}| = 4$ is the finer resolution with four labels. For each level l of the graph sequence the Weisfeiler-Lehman algorithm is applied to produce the $\phi_{(h)}^{(l)}(G^{(l)})$, while for each pair $\{G^{(l)}, G'^{(l)}\}$ for all quantization levels l the intersection Weisfeiler-Lehman subtree kernel will determine the similarity between the two graphs for that level of quantization.

3.3.3 The monotonicity property of the pyramid quantized Weisfeiler-Lehman kernel

We may show the following monotonicity property of the pyramid quantized Weisfeiler-Lehman kernel described in Section 3.3.2 :

Theorem 3.3. Monotonicity property of the pyramid quantized Weisfeiler-Lehman kernel with the granularity of the node labeling
The Weisfeiler-Lehman algorithm for a given height h of subtree patterns produces histograms whose intersection are monotonically decreasing in the granularity of the graph node labeling:

$$\forall l, G^{(l)}, G'^{(l)} \left[\mathcal{I}\left(\phi_{(h)}^{l}(G^{(l)}), \phi_{(h)}^{l}(G'^{(l)})\right) \geq \mathcal{I}\left(\phi_{(h)}^{l+1}(G^{(l+1)}), \phi_{(h)}^{l+1}(G'^{(l+1)})\right)\rangle\right], \tag{3.8}$$

where $\phi_{(h)}^{l}(G^{(l)})$ is the histogram of subtree patterns of height h computed at pyramid level l, and level $l+1$ is more granular.

Proof. We first note that the number of subtree patterns of a given depth, h, of the Weisfeiler-Lehman algorithm is dependent only on the topology of the graph, and not on the graph labeling:

$$\|\phi_{(h)}^{l}(G^{(l)})\|_1 = v \cdot (h+1) \ \forall l \tag{3.9}$$

where v is the number of vertices in the graphs. We next note that the number of vertex labels is strictly monotonic in the pyramid level, $|\Sigma^l| < |\Sigma^{l+1}|$, and that for each label $\sigma \in \Sigma^l$ at level l, there exist a non-empty set of labels $\mathcal{S} \subset \Sigma^{l+1}$ at level $l+1$ such that

$\mathcal{L}^{l+1}(u) \in \mathcal{S} \iff \mathcal{L}^{l}(u) = \sigma$. To complete the proof, we observe that

$$\begin{aligned}&\forall G^{(l)}\ \forall l \left[\left(\|\phi^{l}_{(h)}(G^{(l)})\|_1 = \|\phi^{l+1}_{(h)}(G^{(l+1)})\|_1\right) \wedge \left(\|\phi^{l}_{(h)}(G^{(l)})\|_0 < \|\phi^{l+1}_{(h)}(G^{(l+1)})\|_0\right)\right] \\ &\implies \forall G^{(l)}, G'^{(l)}\ \forall l \left[\mathcal{I}\left(\phi^{l}_{(h)}(G^{(l)}), \phi^{l}_{(h)}(G'^{(l)})\right) \geq \mathcal{I}\left(\phi^{l+1}_{(h)}(G^{(l+1)}), \phi^{l+1}_{(h)}(G'^{(l+1)})\right)\right],\end{aligned} \tag{3.10}$$

where $\|\cdot\|_0$ is the ℓ_0 pseudo norm. □

As the Weisfeiler-Lehman algorithm with $h = 0$ specializes to the bag of words model, we have as a result that our graph kernel for continuous vector valued node labels strictly generalizes the pyramid match kernel [Grauman and Darrell, 2007a].

3.4 Exploring the pyramid quantized Weisfeiler-Lehman features

Given a set of vector labeled graphs $\mathbf{G} = \{G_i = (V_i, E_i, \mathcal{L}_i)\}_{1 \leq i \leq n}$ where $\mathcal{L} : V \to \mathbb{R}^d$ and a classification label $\mathcal{Y}_i$ for each graph G_i, we want to classify them through the information created by the Weisfeiler-Lehman algorithm after the quantization of each the vector labeled graph into a sequence of discrete labeled graphs with increasing granularity as described in Section 3.3. In order to maximize their classification performance and explore better the pyramid quantized Weisfeiler-Lehman features we examine two different approaches, one through the combination of the intersection Weisfeiler-Lehman kernel of the different pyramid levels in Section 3.4.1 and another through the evaluation of each individual subtree pattern $\phi^{(l)}_{(h)}$ of the Weisfeiler-Lehman algorithm in Section 3.4.2.

3.4.1 The pyramid quantized Weisfeiler-Lehman kernel

Applying the intersection Weisfeiler-Lehman subtree kernel (see Definition 3.2) for each pair of graphs $G^{(l)}, G'^{(l)}$ for all the pyramid levels from Equation 3.4, we end up with a sequence of intersection Weisfeiler-Lehman kernels for a given height h of subtree patterns :

$$\left(K^{(0)}_{(h)}(G^{(0)}, G'^{(0)}), \ldots, K^{(L)}_{(h)}(G^{(l)}, G'^{(L)})\right) \tag{3.11}$$

where $K^{(l)}_{(h)}(G^{(l)}, G'^{(l)}) = \mathcal{I}\left(\phi^{(l)}_{(h)}(G^{(l)}), \phi^{(l)}_{(h)}(G'^{(l)})\right)$. Since we have a sequence of Weisfeiler-Lehman kernels from the different quantization levels and taking also into consideration the observation by [Lanckriet et al., 2004] that using multiple kernels instead of a single one can enhance the interpretability of a decision function and improve its performance, we would like to combine this sequence of kernels into a single one. A convenient approach is to consider that the kernel $K(G, G')$ is actually a convex combination of *basis* kernels:

$$K_{(h)}(G, G') = \sum_{l=0}^{L} d_l K_{(h)}^{(l)}(G^{(l)}, G'^{(l)}), \text{ with } d_l \geq 0, \sum_{l=0}^{L} d_l = 1. \tag{3.12}$$

For determining the weights d_l we consider two different approaches a automatic way through the framework of *multiple kernel learning* and another through *a fixed weight scheme*. Remember that for a single kernel, where $\{x_i, y_i\}_{i=1}^{n}$ is the learning set with x_i belongs to some input space $\mathcal{X}$ and $y_i \in \mathcal{Y}$ is the target value for pattern x_i the solution of the learning problem has the form :

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) + \beta \tag{3.13}$$

3.4.1.1 Multiple kernel learning

The automatic determination of the weights d_l of the linear combination of our multiple kernels $K_{(h)}^{(l)}(G^{(l)}, G'^{(l)})$ as well as the coefficients α_i, β in a single optimization problem is known as the multiple kernel learning (MKL) problem [Zien and Ong, 2007, Sonnenburg et al., 2006, Lanckriet et al., 2004, Rakotomamonjy et al., 2008]. The multiple kernel learning approach addresses the problem through a weighted ℓ_2 norm regularization formula. In addition, a ℓ_1 norm is posed as a constraint on the kernel weights a_i. This additional constraint encourages sparse set of basis kernels as an inherited property from the ℓ_1 norm. The primal MKL problem is defined as

$$\begin{aligned} \min_{\{f_l\}, \beta, \xi_i, d_l} & \frac{1}{2} \sum_{l=0}^{L} \frac{1}{d_l} ||f_l||_{\mathcal{H}_l}^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t. } & y_i \sum_{l=0}^{L} f_l(G_i^{(l)}) + y_i \beta \geq 1 - \xi_i \forall i \\ & \xi_i \geq 0 \ \forall i \\ & \sum_{l=0}^{L} d_l = 1, \ d_l \geq 0 \ \forall l \end{aligned} \tag{3.14}$$

where each function f_l belongs to a different RKHS $\mathcal{H}_l$ associated with a kernel $K_{(k)}^{(l)}$. Note that the smaller the d_l is, the smoother f_l should be. When $d_l = 0$, $||f_l||_{\mathcal{H}_l}$ is also equal to zero to yield a finite objective value. MKL transforms the problem into a smooth and convex optimization problem and uses a gradient descent approach to solve it.

3.4.1.2 Fixed weight kernel

Due to the constraint of the ℓ_1 norm on the weight the multiple kernel learning leads to sparse solutions, which sometimes could result in poor performances when all the pyramid level contain approximately equal important information. In order to overcome this problem we additionally consider a fixed weight scheme for combing the different pyramid levels of the quantized Weisfeiler-Lehman kernel, and more specifically an equal weight. That mean that the final kernel is defined as

$$K_{(h)}(G,G') = \sum_{l=0}^{L} d_l K_{(h)}^{(l)}(G^{(l)}, G'^{(l)}), \text{ where } d_l = \frac{1}{L+1}. \tag{3.15}$$

3.4.1.3 Visualization

As the pyramid quantized Weisfeiler-Lehman kernel is defined to be a linear combination of the intersections of the histograms of subtree patterns up to a given height h and as the intersection kernel can be considered as a “quasi-linear” kernel [Vedaldi et al., 2009], we may use these properties to develop visualizations that approximate the learned discriminant functions. We note that a discriminant function for class c has the form:

$$f_c(G) = b + \sum_{l=0}^{L} d_l \sum_{j} \alpha_{lj} \mathcal{I}\left(\phi_{(h)}^{(l)}(G_j^{(l)}), \phi_{(h)}^{(l)}(G^{(l)})\right) \approx b + \sum_{l=0}^{L} d_l \langle w_l, \phi_{(h)}^{(l)}(G^{(l)})\rangle \tag{3.16}$$

where l indexes the levels of the pyramid and j indexes over the samples in the training set. At each pyramid level l there is exactly one subtree pattern of height h rooted at each vertex of the graph. We may generate for each vertex v a visualization of the function by coloring each vertex by the weight in w_i corresponding to the subtree pattern rooted at that node. We may additionally sum the weights over all levels of the Weisfeiler-Lehman iterations. Due to the bias term, b, the visualizations show only the relative contribution of a region of the graph to the discriminant function.

3.4.2 Elastic net on the pyramid quantized Weisfeiler-Lehman subtree features

In order to explore the contribution of each quantized Weisfeiler-Lehman subtree feature $\phi_{(h)}^{(l)}(G^{(l)})$ for all the quantized pyramid levels l and for all depths up to depth h, we make use of the statistical estimator Elastic Net, which is described in detail in Section 5.1.2. We note that the Elastic Net combines ℓ_1 with ℓ_2 regularization in order to appropriately trade off sparsity with a low variance estimator in the case of correlated signals. Formally, if $\phi^{(l)}_{(h)}(G^{(l)}_i)$ is a feature vector of subtree pattern up to height h for a given quantization level l for a graph G_i, the elastic net computes

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \frac{1}{n} \sum_{i=1}^{n} \left(\langle \beta, \phi^{(l)}_{(h)}(G^{(l)}_i) \rangle - y_i \right)^2, \tag{3.17}$$

where $\lambda_1,\ \lambda_2 \geq 0$ are scalar regularization parameters. This objective includes the lasso [Tibshirani, 1996] and ridge regression [Tikhonov, 1963] as special cases by setting λ_2 or λ_1 to zero, respectively. We can see this as a tradeoff between the high degree of sparsity achieved by the lasso, and the low variance estimates achieved by ridge regression.

Visualization Visualizations of the learned discriminant functions is easy to obtain when the Elastic Net is employed over the pyramid quantized Weisfeiler-Lehman subtree patterns. Each voxel is associated with a $L \times h$ subtree patterns, where L is the number of quantization levels of the label space and h is the height of subtree patterns created by the Weisfeiler-Lehman algorithm, and we just need to sum up their respective weights b_i as provided by the Elastic Net.

Chapter 4

Applications of the pyramid quantized Weisfeiler-Lehman graph representation in neuroimaging and shape classification

As we have seen in Section 1.1, many problem domains can be naturally represented with graphs. The widespread use of graphs requires the development of efficient methods for representation and comparison. Although many algorithms have been developed the last decades for graph comparison between either unlabeled or discrete labeled graphs, as we have seen in Section 2, the efficient and expressive representation and comparison of graphs with continuous and/or high-dimensional vectors labels remains an open research problem. In Chapter 3 we tackled the graph comparison problem with continuous or high-dimensional vector labels with the introduction of the *pyramid quantized Weisfeiler-Lehman graph representation*. In this chapter, we evaluate this representation using real data from two different domains. The first evaluation, described in Section 4.1, comes from the fMRI analysis area and its objective is to discriminate between cocaine abusers and healthy control subjects, while in Section 4.2, we use two datasets with 3D shape meshes. For the first dataset the objective is to discriminate between healthy and patients subjects that suffer from a neuromuscular dystrophy, while in the second dataset we tackle the problem of multiclass object classification.

4.1 The pyramid quantized Weisfeiler-Lehman graph representation in fMRI analysis

4.1.1 Introduction

In this section we evaluate the pyramid quantized Weisfeiler-Lehman graph representation in an fMRI data analysis problem. The functional magnetic resonance imaging (fMRI) is a wide spread, non-invasive modality used in the field of neuroimaging that measures brain activity by detecting associated changes in blood flow. The goal of fMRI data analysis is to detect relationships between brain activation and the designed task the subject performs during the scan. Depending on the specifics of the problem under investigation this goal can be translated as different objectives, such as localizing regions of the brain that participate in the specific task, or determining connectivity networks that correspond to brain function or even making predictions about psychological or disease states.

A number of discriminative learning approaches have been applied to fMRI analysis including the wide spread generalized linear model [Bartels et al., 2007, Bartels and Zeki, 2004a], support vector machines [Song et al., 2011, LaConte et al., 2005], independent component analysis [Bartels and Zeki, 2004b, 2005] and kernel canonical correlation [Hardoon et al., 2007, Blaschko et al., 2009, 2011]. All these methods have to deal with (*a*) data that lie in a high-dimensional space, with ten of thousands of voxels, (*b*) a small number of samples, due to the high cost and time consuming nature of the fMRI acquisition procedure, and (*c*) high levels of noise that arise from different sources, such as system noise and random neural activity. In order to overcome the problem of curse of dimensionality, some approaches select features either by a predefined set of regions of interest (ROIs) using either prior knowledge [Demirci et al., 2008, Wang et al., 2003], or statistical methods [Mitchell et al., 2004, Tahmasebi et al., 2012] such as a t-test [Mitchell et al., 2004], analysis of variance (ANOVA) [Cox and Savoy, 2003]. The main disadvantages in the use of ROIs are (*a*) such regions are frequently defined within a reference space, which raises the issue of misregistrations, (*b*) in practice people might perform “double dipping” [Kriegeskorte et al., 2009] in the data in order to find the set of ROIs and hence significantly skew the results and (*c*) in the case of absence of prior knowledge they are undefined. Therefore, fully exploratory methods are preferred.

fMRI analysis is particularly suited to sparsity regularization due to the intrinsic high dimensional nature of fMRI data and the expense of collecting large numbers of samples. Moreover, sparsity regularization methods do not require a predefined set of ROIs, are fully exploratory and are also mathematical appealing. Previous works that have explored sparsity regularization in fMRI include [Carroll et al., 2009, Ng et al., 2012b].

Although the aforementioned methods perform well in analyzing fMRI data, they treat the fMRI prediction as a linear combination of functions over individual voxels, ignoring either the 3D structure of the brain and they cannot capture potentially complex interactions between voxels. On the other hand, graph-theoretic methods can model such information through the rich representations of networks of data, and are consequently a promising representation for neural populations. The most common use in the fMRI analysis is modeling the network of brain connectivity [Supekar et al., 2008, Liu et al., 2008], under both healthy conditions (*e.g.* age-related changes [Fair et al., 2009, Supekar et al., 2009]) and diseases (*e.g.* Alzheimer's [Supekar et al., 2008] or Schizophrenia [Liu et al., 2008]), and the network's analysis, including modularity, small-worldness and the existence of highly connected network hubs. Graph kernel methods have also been used in fMRI connectivity graphs for brain decoding [Mokhtari and Hossein-Zadeh, 2013].

In this section, we approach the fMRI analysis by representing fMRI recordings as graphs, and we use the *pyramid quantized Weisfeiler-Lehman graph representation* to learn from the interconnections between voxels. Our approach has an enriched capacity to model such dependencies by considering interconnections between voxels which may be functionally important.

The remaining of the section is organized as follows: in Section 4.1.2 we present the data that we use in this study, Section 4.1.3 is dedicated to the methodology, in Section 4.1.4 we show the experimental setting and the results, and we conclude in Section 4.1.5 with a discussion over the obtained results and the perspectives of this work.

4.1.2 Cocaine Addiction Dataset

The cocaine addiction dataset consists of the contrast maps from 16 cocaine addicted individuals and 17 control subjects performing a neuropsychological experiment, called a drug Stroop experiment [Goldstein et al., 2009]. The drug Stroop experiment has a block design, that included six sessions, with each of them having different conditions. The two varying conditions are the monetary reward (50¢, 25¢ and 0¢) and the cue shown (drug words, neutral words). The session consists of an initial screen displaying the monetary reward and then presenting a sequence of forty words in four different colors (yellow, blue, red or green). The subject was instructed to press one of four buttons matching the color of the word they had just read. The subjects were rewarded for correct performance depending on the monetary condition. The fMRI data were acquired a 4Tesla whole-body Varian/Siemens system. The blood-oxygen-level dependent

(BOLD) responses were measured as a function of time using a T2*-weighted single-shot gradient-echo EPI sequence (TE/TR=20/1600ms, 4mm slice thickness, 1mm gap, typically 33 coronal slices, 20cm FOV, 64×64 matrix size, 3.1×3.1mm in-plane resolution, 90° flip angle, 200kHz bandwidth with ramp sampling, 128 time points and 4 dummy scans to be discarded to avoid non-equilibrium effects in the fMRI signal). Padding was used to minimize subject motion, which was also monitored immediately after each fMRI run [Honorio et al., 2012].

The subjects that complied to the following requirements: motion $<$ 2mm translation, $< 2^\circ$ rotation and at least 50% performance in an unrelated task [Goldstein et al., 2009], where include in this study. The Statistical Parametric Mapping (SPM2) toolbox [Friston et al., 2007] was used to preprocess the imaging data and to produce the contrast maps before the analysis with the regularization methods. The preprocessing included a six-parameter rigid body transformation (3 rotations, 3 translations) for image realignment and to correct for head motion, spatially registration to the standard Talairach frame using a voxel size of $3 \times 3 \times 3\text{mm}^3$, an 8mm full-width half-maximum Gaussian kernel to smooth the data in order to reduce the amount of spatial noise as well as the impact of small inaccuracies in the spatial registration across subjects.

In order to compute contrast maps for each subject, experimental condition and session, a general linear model (GLM) with box-car design convolved with a canonical hemodynamic response function (HRF), low-pass filters (HRF) and high-pass filters (cut-off frequency: 1/520s) was used. The GLM contained a single regressor for each of six sessions corresponding to one of three monetary reward conditions (50¢, 25¢, 0¢) and one of two cues (drug words, neutral words). In addition, six motion regressors (3 rotations, 3 translations) were included for all event related tasks. In order to compute a single contrast map for each subject and experimental condition, the contrast maps that were produced by the GLMs (per subject, experimental condition and session) were averaged. After computing these average contrast maps and before using them in our pipeline, grand mean scaling [Friston et al., 2007] was applied independently per subject and experimental condition, since scale between different subjects can significantly differ. Note that in our experiments, we use only one image per subject and experimental condition. In this study, we focus on the monetary conditions only, and more specifically the session of 50¢ following [Honorio et al., 2012] and the discriminative task is to classify the subject as cocaine addicted or control.

Algorithm 4.1 The statistical learning pipeline for fMRI analysis with sparse subgraph statistics.

```
Require: Training set $\mathcal{D} = \{(v_i, y_i), i = 1, \ldots, n\}$.
Compute $\hat{\beta}_{\text{lin}}$ from the objective in Equation (5.4) with $x_{\text{lin}} \equiv \operatorname{vec} v$.
Construct $k$-nearest neighbor graphs for all training samples from the voxels associated with non-zero $\hat{\beta}_{\text{lin}}$
for each level in the quantization pyramid do
    Label the nodes of all graphs according to the quantization of the voxel value.
    Compute the Weisfeiler-Lehman statistics for the given quantization level over all graphs and aggregate them into the feature vector $\phi_{\text{graph}}(v)$.
end for
Compute $\hat{\beta}_{\text{graph}}$ from the objective in Equation (5.4) with $x_{\text{graph}} = \phi_{\text{graph}}(v)$.
```

4.1.3 Methodology

Our approach for fMRI analysis enriches the capacity to model non-linear dependencies between voxels, through the representation of an fMRI recording as a graph. The statistical learning pipeline of our approach can be seen in Algorithm 4.1. In order to make use of a rich graph representation several design choices must be made: (i) the learning algorithm, (ii) the graph construction, (iii) the node labeling and (iv) the graph statistics employed as a feature representation, which we address in the following paragraphs.

Sparsity Regularization As our statistical estimator, we have made use of the Elastic Net (for more details see Section 5.1.2). The Elastic Net combines ℓ_1 with ℓ_2 regularization in order to appropriately trade off sparsity with a low variance estimator in the case of correlated signals. This method is particularly appropriate in fMRI where nearby voxels are likely to be correlated, and regions responsible for a given function or behavior distributed across multiple voxels. Furthermore, it is typical that the majority of voxels in the brain are not discriminative of a specific output. Note that one could use the k-support norm as a regularizer as we consider in a later section of this thesis, but we have used a more established statistical approach in this section. We make use of the Elastic Net twice in our learning pipeline (see Algorithm 4.1). In the first instance, we use the Elastic Net on the raw voxel values to determine a subset of voxels on which we build a graph representation, specifically those with non-zero $\hat{\beta}_{\text{lin}}$. Our model selection step has typically chosen approximately 10^3 voxels for this stage. We subsequently compute subgraph statistics over this graph to generate a feature vector, $\phi_{\text{graph}}(v)$. Finally, we use the Elastic Net on these subgraph statistics in order to determine our final prediction function, with a model selection step to determine appropriate values for λ_1 and λ_2.

Graph Construction To construct the graph representation, we have made use of k-nearest neighbor graphs on the voxels that were selected by an initial training of the Elastic Net (see Line 1 in Algorithm 4.1). We symmetrize the k-nn relationship by considering the edges to indicate an undirected graph structure. While other models of connectivity are of interest [Sporns, 2010, Wee et al., 2011], we have found that the use of k-nearest neighbors to determine the graph topology yields good performance in general. Furthermore, the subtree statistics considered here implicitly account for longer distance connections for sufficiently deep subtree patterns. We set $k = 5$ in all experiments.

Continuous node labels To enrich our graph representations of the fMRI contrast maps, we take advantage of the activation information. At each voxel selected by the Elastic Net for the construction of the graph, we label it with its activation. Since the activation has continuous values, our graph representation is transformed to a continuous labeled graph.

Graph statistics Since the fMRI contrast maps are represented as graphs with continuous labels on the vertices, we explore the *pyramid quantized Weisfeiler-Lehman graph representation* introduced in Chapter 3. We quantized the continuous activation labels with the fixed-binning strategy (see Section 3.3.1.1), ending with a sequence of discretely labeled graphs with increasing granularity. Through the efficient Weisfeiler-Lehman algorithm, we aggregate statistics of subtree patterns of different depth h for all the levels of quantization. Finally, we control the complexity of our prediction while modeling non-linear interactions between voxels by adding a sparsity regularizer (the Elastic Net) over the statistics of subtree patterns (see Line 7 in Algorithm 4.1).

We are able to learn in a fully exploratory fashion without restricting our prediction, e.g., to a pre-defined region of interest or a connected component. Overall, we represent fMRI data as graphs over voxels, and compare the resulting graphs with a novel method that combines elements of the Weisfieler-Lehman graph kernel [Shervashidze et al., 2011] and the pyramid match kernel [Grauman and Darrell, 2007a], a method that achieves the computational advantages of efficient graph kernels while extending the representation to continuous node labels.

Table 4.1: Mean accuracy over the hold-out data of 50 trials of the *pyramid quantized Weisfeiler-Lehman graph representation* for four different subtree pattern depths, $h \in \{0, 1, 2, 3\}$. Maximum performance is achieved with subtree patterns up to depth two.

Pyramid Quantized Weifeiler-Lehman				
h	**0**	**1**	**2**	**3**
Accuracy	54.00%	57.14%	**64.28%**	63.42%

4.1.4 Results

We use the same experimental setup, a random splitting scheme with 50 trials, to estimate the classification performance of *pyramid quantized Weisfeiler-Lehman graph representation* and the baseline method on the cocaine addiction dataset. In each trial, a random selection of 80% of the data are used for training, while the remaining 20% are used to estimate the performance.

In Table 4.1 we show the performance of the *pyramid quantized Weisfeiler-Lehman graph representation* for four different depths of subtree patterns (see Chapter 3). Our approach achieves a mean accuracy of 64.28% for subtree patterns up to depth two. We also compare our proposed technique with three other methods on the same dataset: (i) Gaussian kernel ridge regression, (ii) the Elastic Net with raw voxels as features, and (iii) the Elastic Net with raw voxels and *pyramid quantized Weisfeiler-Lehman* subtree features concatenated in a joint feature vector. In Figure 4.1 we show the mean accuracy of the final system and the standard error. *Pyramid quantized Weisfeiler-Lehman graph representation* outperforms the rest of the methods. With a Wilcoxon signed rank test between the Elastic Net with raw voxels and the *pyramid quantized Weisfeiler-Lehman graph representation* we determine that our proposed method is statistically significantly better ($p = 0.02$). Additionally, a reduction of over 14% in classification error is recorded between the Elastic Net on the raw voxels and our method.

Intermediate Accuracies In order to explore the behavior of the *pyramid quantized Weisfeiler-Lehman graph representation* for different combinations of quantization and depths of subtree patterns, we estimated the mean accuracies per quantized pyramid level and per subtree pattern depth, as shown in Figure 4.2. Figure 4.2 gives insight into the effect of the depth of the subtree patterns on the degree of quantization that gives maximal performance. With subtree patterns of depth $h = 0$, the method reduces to a simple pyramid bag of words model, and a relatively high granularity quantization works best. With subtree patterns of depth two or greater, accuracies are highest with a very coarse quantization and information appears to be represented primarily in the

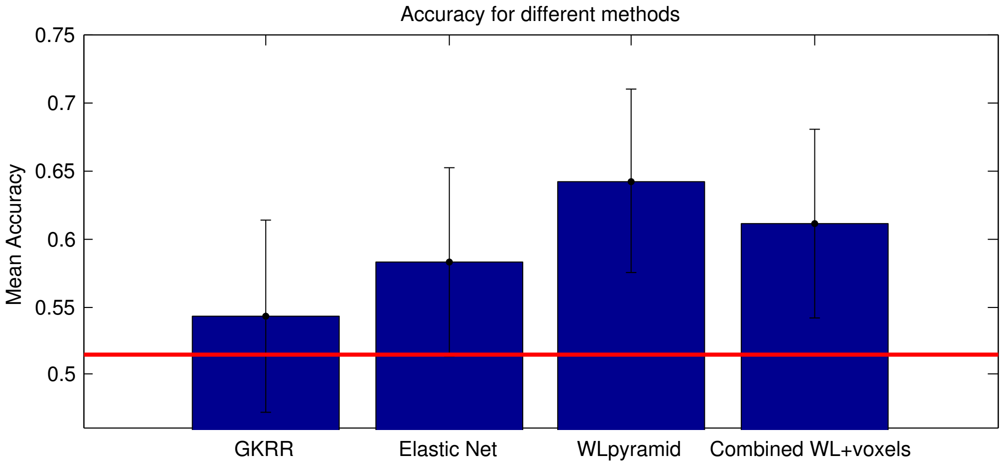


FIGURE 4.1: Mean accuracy and standard error on the cocaine addiction dataset. The compared methods are (left to right) Gaussian kernel ridge regression (GKRR), the Elastic Net on raw voxels, *pyramid quantized Weisfeiler-Lehman* (WLpyramid), and the Elastic Net with a concatenation of the raw voxels and the *pyramid quantized Weisfeiler-Lehman* features (Combined EN+WL). The horizontal red line indicates chance performance. The *pyramid quantized Weisfeiler-Lehman* features perform better than Gaussian kernel ridge regression and the Elastic Net on raw voxels with statistical significance.

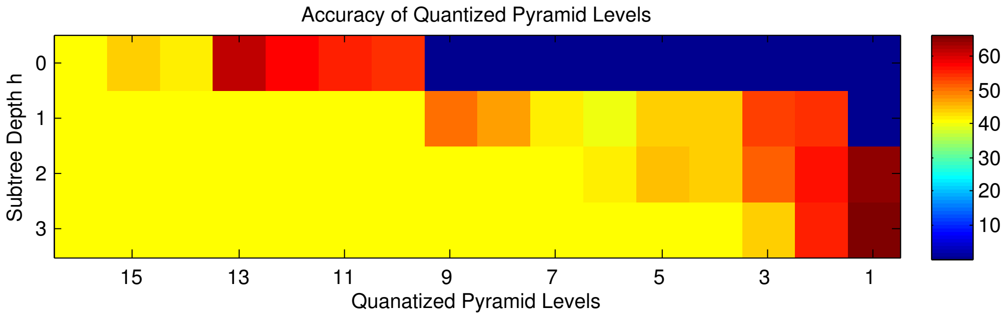


FIGURE 4.2: A heat map representation of the intermediate mean accuracies over the hold-out data of 50 trials for all the quantized pyramid levels and for four different depths of the subtree patterns, $h \in \{0, 1, 2, 3\}$. This figure shows that in the bag of words model, we need a large vocabulary, while as the depth of the Weisfeiler-Lehman algorithm increases, accuracies are highest for low granularity quantization. The final algorithm learns across all depths and quantization levels automatically. (Figure best viewed in color.)

relationships between voxels. We note, however, that the results in Table 4.1 and in Figure 4.1 are computed with the concatenation of features computed from all quantization levels, and an appropriate combination of subtree features across all quantization levels and depths of subtree patterns was selected by the Elastic Net.

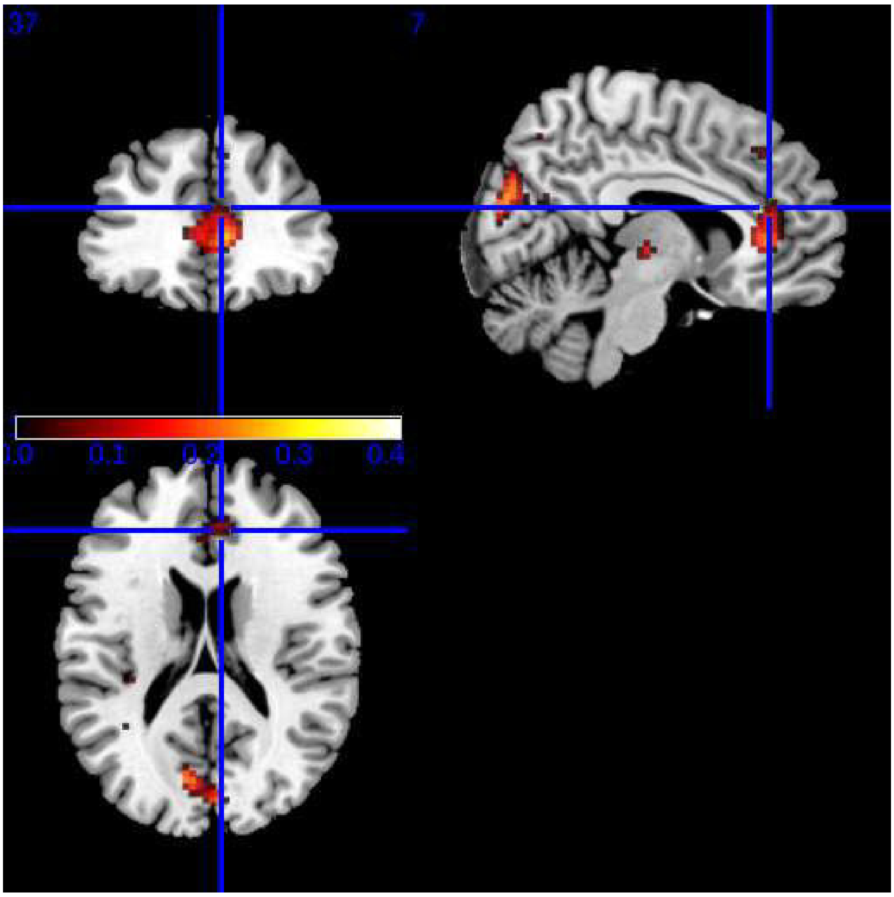
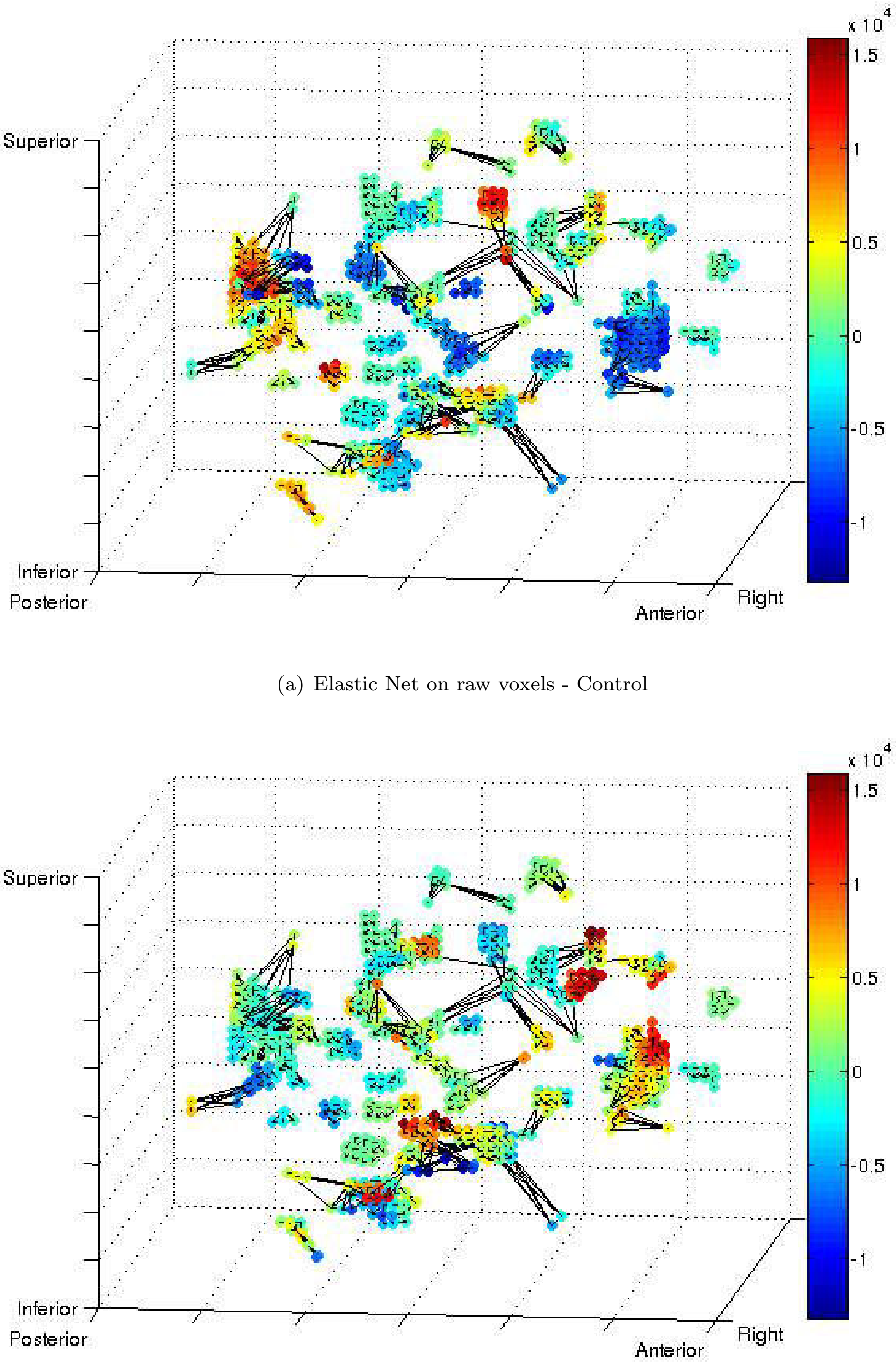


FIGURE 4.3: A visualization of the areas of the brain selected by Elastic Net. The selected regions correspond to areas previously implicated as being related to addiction [Goldstein et al., 2009].

4.1.5 Discussion

Figure 4.3 shows the areas selected by the Elastic Net, while Figure 4.4 and Figure 4.5 show the visualizations of the learned functions for the Elastic Net on raw voxels and the *quantized Weisfeiler-Lehman graph representation* respectively. Note that Elastic Net on the raw voxels was able to select the rostral anterior cingulate cortex (rostral ACC), an important region as our neuroscientist mentioned (for more details see Section 5.2.4).

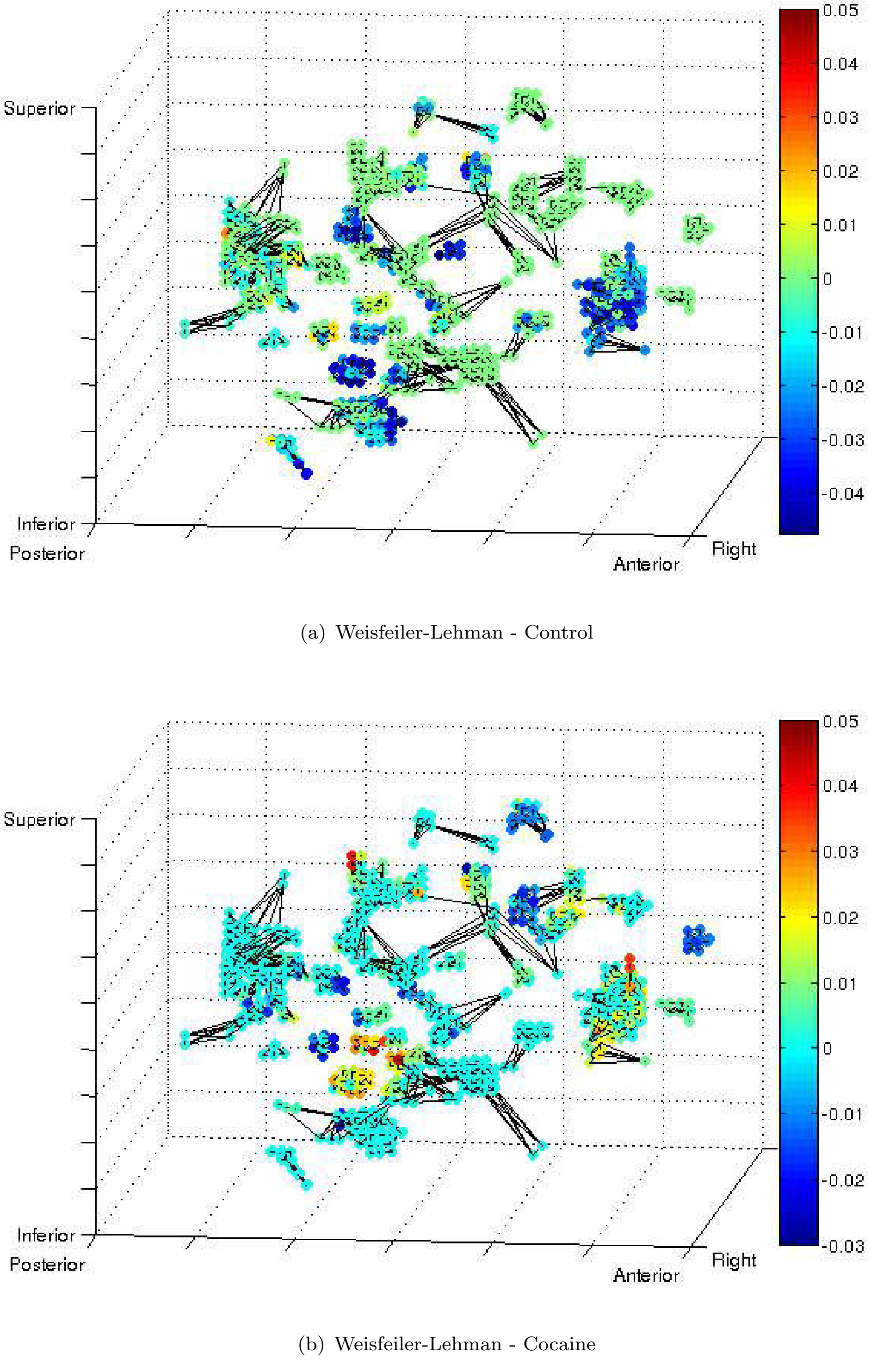
Although our method works in an implicitly high dimensional space, we empirically observe that Elastic Net regularization controls the complexity at each stage of the pipeline. The first learning step selects approximately 1100 voxels. Using the *pyramid quantized Weifeiler-Lehman graph representation*, we generate a feature vector of length 6×10^5, but with a sparsity of $\sim 2\%$. The second application of Elastic Net selects only $\sim$ 2K dimensions. In each step, the method retains complexity much lower than a “simple” linear function over tens of thousands of voxels as has been proposed in previous works.



(a) Elastic Net on raw voxels - Control

(b) Elastic Net on raw voxels - Cocaine

Figure 4.4: A visualization of the function learned by Elastic net for control and cocaine subjects over the raw voxels. The visualization is illustrated over a graph, whose construction is described in Section 4.1.3, just for comparison with Figure 4.5.



(a) Weisfeiler-Lehman - Control

(b) Weisfeiler-Lehman - Cocaine

Figure 4.5: A visualization of the function learned by the pyramid quantized Weisfeiler-Lehman graph representation applied to control and cocaine addicted subjects.

Several broad observations are apparent from our quantitative results. From Table 4.1, we note that subtree patterns up to depth two seem to perform best, and that deeper subtree patterns begin to reduce average performance. This indicates that the big-$\mathcal{O}$ complexity of the graph representation is only slightly higher than using a simple linear function. The proposed method performs significantly better than the Gaussian kernel ridge regression and the Elastic Net baselines (see Table 4.1 and Figure 4.1). In our final experiment of combining the raw voxel values with the subtree pattern features, we found that performance decreased slightly from that of only considering subtree pattern features.

In this work, we have presented a fully automated, statistically sound method for classification of brain states with graph representations, using the *pyramid quantized Weisfeiler-Lehman graph representation*. The method was evaluated on a real world dataset and outperformed other machine learning techniques with statistical significance, including kernel ridge regression and the Elastic Net. This validates the primary hypothesis of this work: that the interconnections between voxels can contain additional information about brain structure that is not apparent in a linear function on the raw voxel values.

4.2 3D shape classification

4.2.1 Introduction

Three-dimensional objects are extensively used in a numerous areas, such as computer games, biomedical research studies, CAD models and cultural heritage. Examples with applications that use 3D objects can been seen in Figure 4.6. Their widespread incorporation in many areas generates the need to store, classify and retrieve them automatically and efficiently. 3D surface models, also known as 3D shapes, represent a 3D object by a finite set of surface points in 3D space, connected by various geometric entities such as triangles, curved surfaces, *etc.*

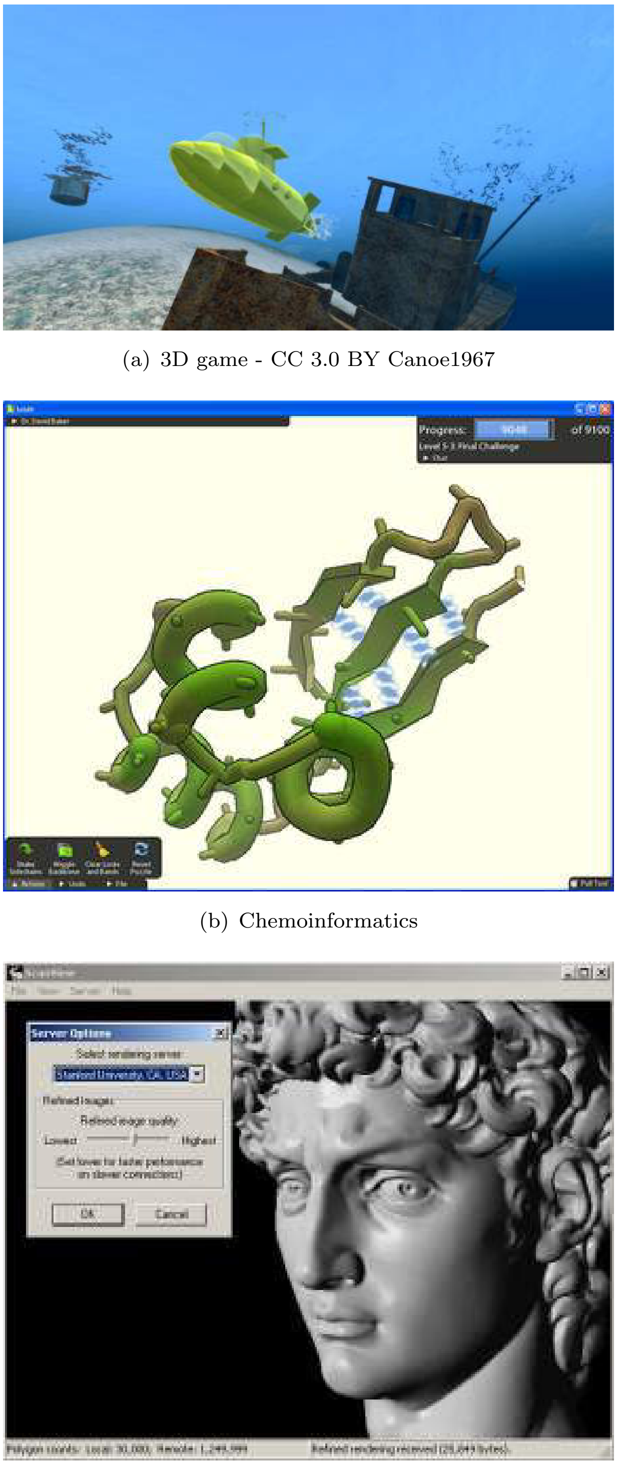
In previous proposed methods, the three-dimensional objects are commonly associated with a 3D descriptor. There are three wide categories of 3D descriptors (*a*) feature-based methods, (*b*) view-based methods, and (*c*) graph-based methods. Feature-based methods represent objects as histograms of statistics of global features [Elad et al., 2002, Mahmoudi and Sapiro, 2008, Kokkinos et al., 2012], such as volume, moments and geodesic distance, or local features [Lee et al., 2005, Castellani et al., 2008], such as curvature and normals. The advantage of feature-based methods is that they are computationally efficient, as they represent a potentially complex 3D object by only a few dimensions. On the other hand, view-based methods use multi-viewpoint projections to produce a number of rendered images, the combination of which forms a global object descriptor [Ohbuchi and Furuya, 2010]. Finally, graph-based methods use only the topological properties of the 3D object in its representation, such as Reeb graphs [Hilaga et al., 2001] and skeleton graphs [Sundar et al., 2003]. The disadvantage of these methods is that are computationally expensive and can be sensitive to small topological changes.

In our approach, we denote that the 3D surface models can be viewed as graphs $G(V, E)$, where the finite set of points in the 3D space will represent the vertices V and the connection between two points in order to form triangles or curved surfaces will represent the edges E. This perspective specifies the topology of a graph, but does not explicitly encode relative vertex positions or other geometric properties. Therefore, we extend the notion of the graph to incorporate node labels that encode properties, such as local curvature of the surface. In order to incorporate this representation in a statistical learning framework, we interpret 3D shapes as continuous vector labeled graphs and use the *pyramid quantized Weisfeiler-Lehman kernel*, introduced in Chapter 3, to learn the classification functions. We overcome the problem of computational inefficiency and oversensitivity to topological changes by representing graphs using statistics of subtree patterns.

The remainder of this section is structured as follows: In Section 4.2.2 we present the two 3D shape datasets, a dataset from medical imaging and a dataset from the semantic shape classification tasks, in Section 4.2.3 we introduce the local features of the 3D shapes that are used as continuous vector labels in their graph representation, in Section 4.2.4 we present an overview of the pipeline strategy used in this problem, in Section 4.2.5 we report the experimental results on two aforementioned datasets and we conclude with a discussion in Section 4.2.6.

4.2.2 3D shapes datasets

We evaluate the *pyramid quantized WeisfeThe obtained volumes had a size of 64 × 64 × 20 voxels and a voxel resolution of 3.125mm × 3.125mm × 7mm. T1- and T2-weighted MR images were acquired at the same time. As a consequence, the image volumes are naturally co-registered. iler-Lehman kernel* on two 3D surface shape categorization tasks. In the first task, we address the problem of categorizing shapes extracted from segmented medical images of calf muscle. The discriminative task is to determine the presence or absence of neuromuscular dystrophy. In the second task, we address the problem of semantic shape categorization based on the SHREC 2013 data set.

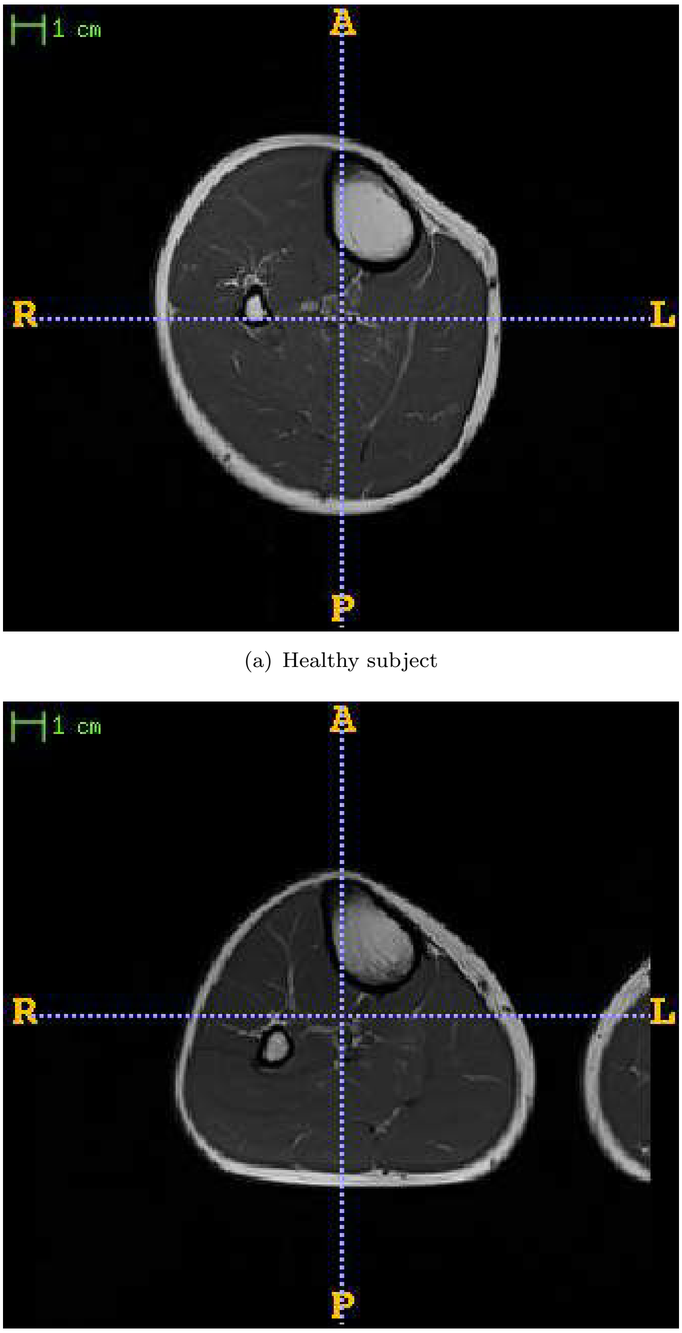

Progress: 9048 of 9100
Level 5-3: Final Challenge
Server Options
Select rendering server:
Stanford University, CA, USA
Refined images:
Refined image quality
Lowest
Highest
(Set lower for faster performance on slower connections)
OK
Cancel
Polygon counts: Local: 30,000, Remote: 1,249,999
Refined rendering received (25,049 bytes).

(a) 3D game - CC 3.0 BY Canoe1967

(b) Chemoinformatics

(c) Cultural heritage

FIGURE 4.6: Examples of application that use 3D objects. Figure 4.6(a) show an screen-shot from the video game *Second life* that stimulates a virtual 3D world. Figure 4.6(b) show a screen-shot from the on-line puzzle video game *Foldit* that uses 3D protein structure to understand how proteins fold for the use of drug development. Figure 4.6(c) shows the *Digital Michelangelo* project from Standford that aims to digitize cultural artifacts for cataloging, conservation and restoration.



(a) Healthy subject

(b) Patient

FIGURE 4.7: T1-weighted MR images of the calf from a healthy and patient subject. On the top, Figure 4.7(a) shows a slice of the MR image from a healthy subject, while on the bottom, Figure 4.7(b) shows a slice of the MR image from a patient with a neuromuscular disease.

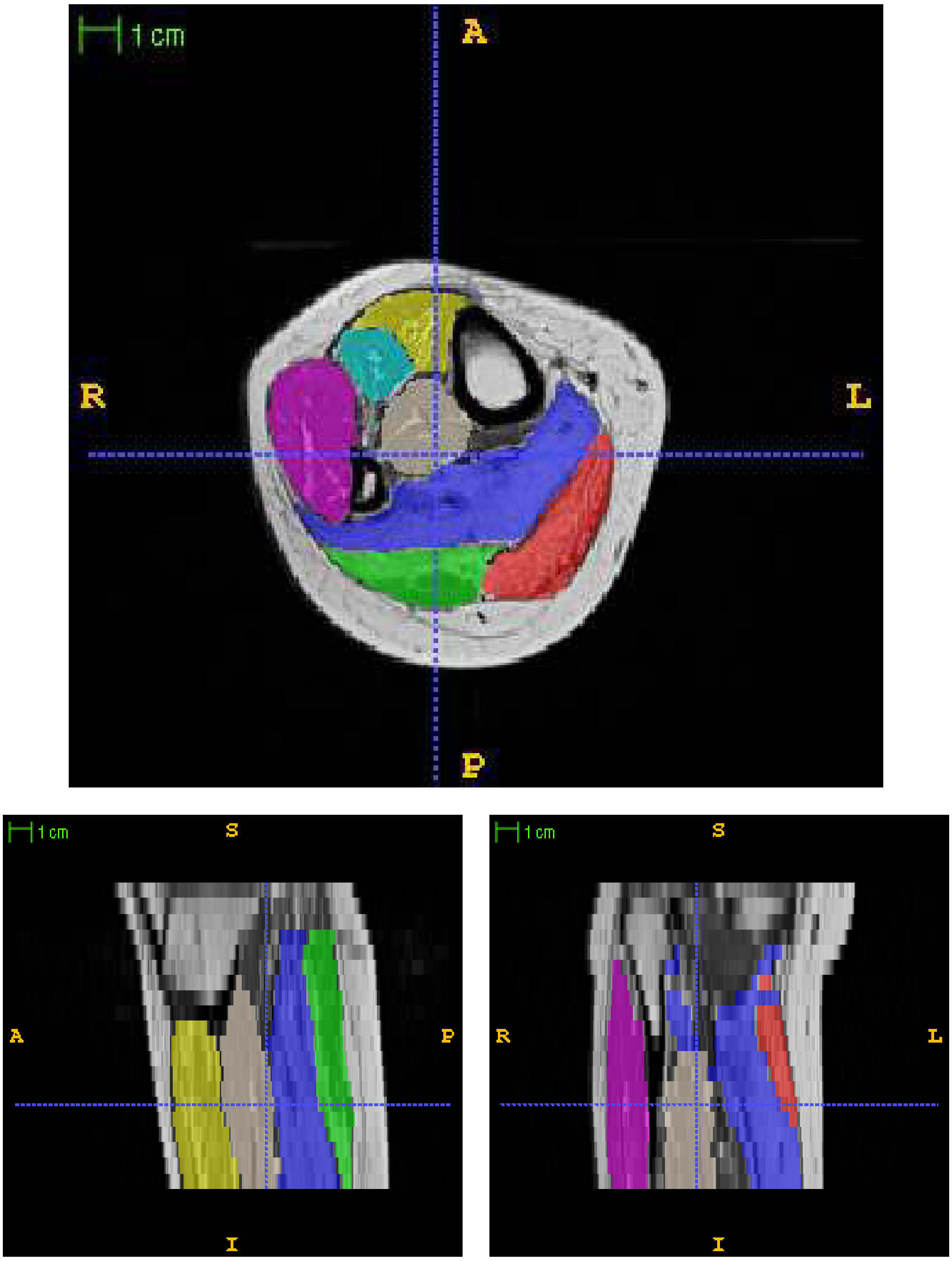
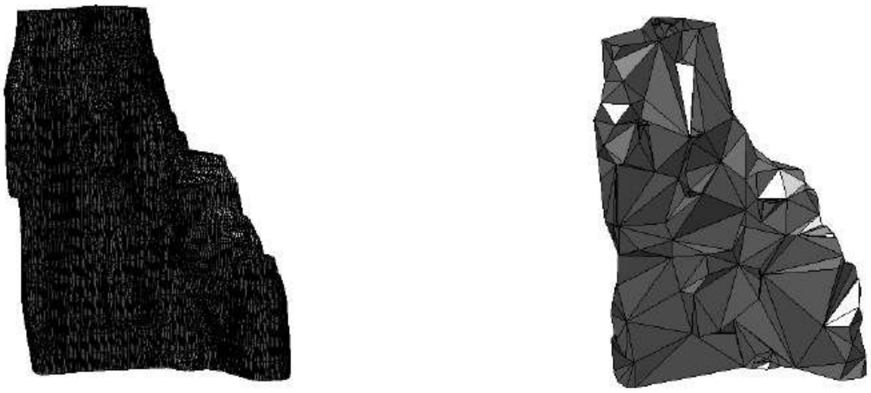


Figure 4.8: An example of an T1 weighted MR image with the seven segmented muscles of the calf. Each color represents a single muscle. Yellow represents the anterior tibialis, cyan the extensor digitorum longus, magenta the peroneous longus, white the posterior tibialis, blue the soleus, green the lateral gastrocnemius, and red the edial gastrocnemius. (Figure best viewed in color.)



(a) Without decimation

(b) With decimation

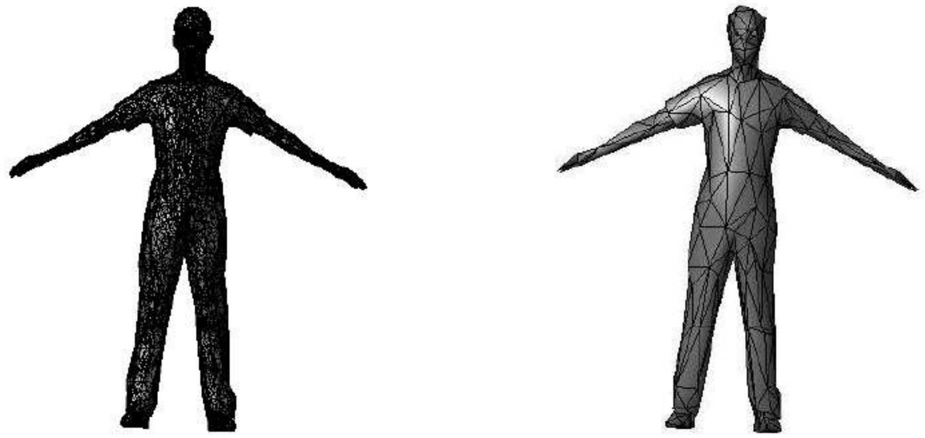
Figure 4.9: An example from the neuromuscular dystrophy dataset for soleus muscle before and after the decimation pre-processing step. On the left, Figure 4.9(a) shows the soleus muscle before the decimation procedure, while on the right Figure 4.9(b) show the soleus muscle after the decimation procedure.

4.2.2.1 Neuromuscular Dystrophy Dataset

The neuromuscular dystrophy dataset consists of 41 subjects: 27 are affected by a neuromuscular dystrophy (either facioscapulohumeral muscular dystrophy or myotonic muscular dystrophy type 1), while the remaining 14 subjects are healthy. In a clinical context, this a large sample size. The subjects were imaged in the calf using a 1.5 T MRI scanner. The obtained volumes had a size of $64 \times 64 \times 20$ voxels and a voxel resolution of 3.125mm $\times$ 3.125mm $\times$ 7mm. An example of the T1-weighted MR images of the calf from a healthy and patient subject can be seen in Figure 4.7. It is not immediately apparent from these images whether zero, one, or both subjects have a neuromuscular dystrophy even to experts; a confirmation is achieved by an invasive muscle biopsy. The T1 weighted MR images were manually segmented by an expert separating 7 important calf muscle groups: 1) soleus (SOL), 2) lateral gastrocnemius (LG), 3) medial gastrocnemius (MG), 4) posterior tibialis (TP), 5) anterior tibialis (AT), 6) extensor digitorum longus (EDL), and 7) peroneous longus (PL). An example of the segmented muscle can be seen in Figure 4.8. It is planned to automate this process in future work. In the meantime, the overall approach provides a strategy to avoid an invasive biopsy.

Each segmented muscle is then transformed into a 3D surface mesh using the itk-snp program.[1] Consequently, the exported 3D meshes consist of a huge number of vertices and edges and require a decimation pre-processing step (for more details see in paragraph "Decimation preprocessing" in Section 4.2.2.3 and Figure 4.9). Figure 4.11 shows an example of the seven segmented muscles of the calf as 3D surfaces meshes from a healthy

[1] http://www.itksnap.org/pmwiki/pmwiki.php



(a) Without decimation

(b) With decimation

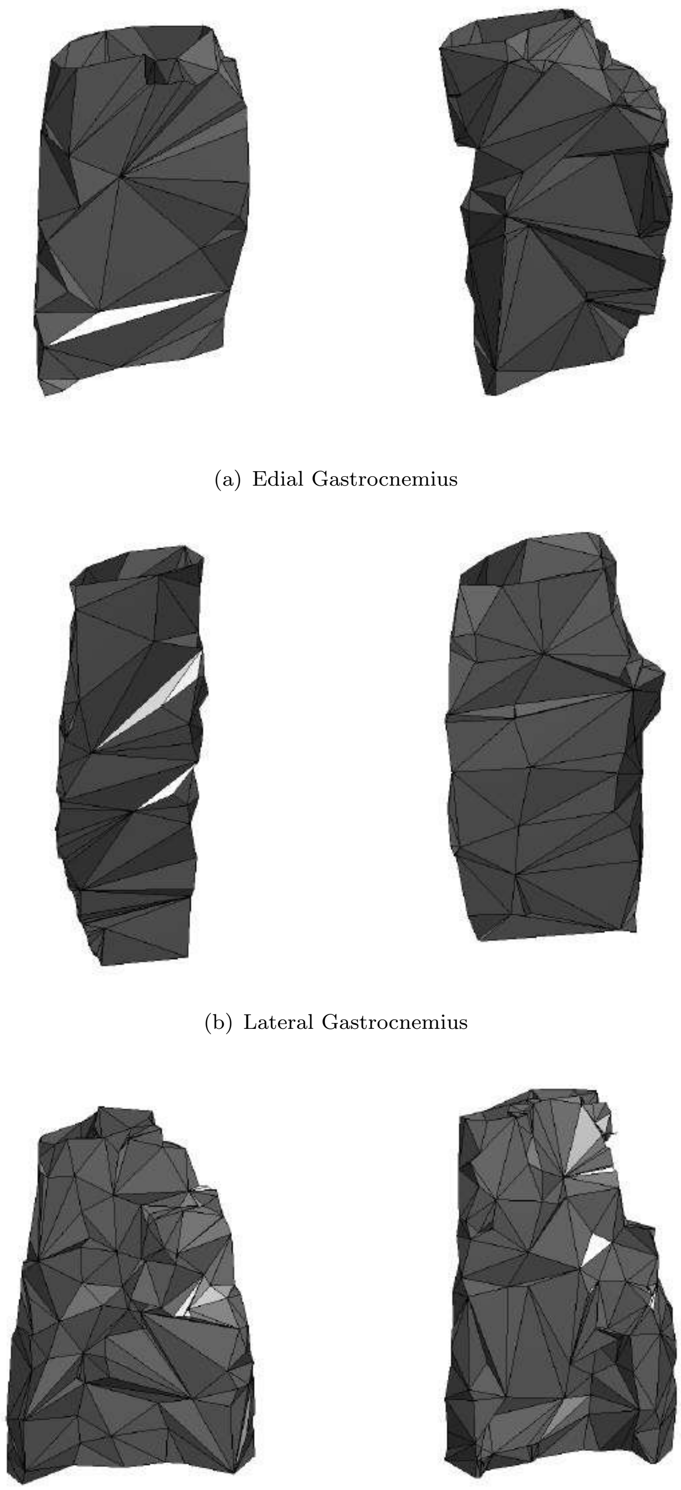
Figure 4.10: An example from the SHREC 2013 dataset for the biped class before and after the decimation pre-processing step. On the left, Figure 4.10(a) shows the biped 3D shape before the decimation procedure. while on the right 4.10(b) shows the shape biped 3D shape after the decimation procedure.

subject, on the left, and a patient, on the right respectively, after the prepocessing procedure of decimation is applied. Finally, we should note that the discriminative task for this dataset is to distinguish between patients of neuromuscular dystrophy and healthy subjects.

4.2.2.2 SHREC 2013 dataset

The SHREC 2013 dataset was selected from the SHREC 2013 Contest *"Large-Scale Partial Shape Retrieval Track Using Simulated Range Images"* track.[2] Although the initial dataset consists of a target set and a query set, that contains full large-scale models and partial views of the models respectively, we focus only on the target set where a ground-truth was easily accessible. The dataset consists of 20 classes of generic objects, which are in alphabetical order: 1) bed, 2) bicycle, 3) biped, 4) biplane, 5) bird, 6) bottle, 7) car, 8) cellphone, 9) chair, 10) cup, 11) desklamp, 12) fish, 13) floorlamp, 14) insect, 15) monoplane, 16) mug, 17) phone 18) quadruped, 19) sofa and 20) wheelchair. Each class contains 18 different large-scale models, resulting in a total of 360 objects. Examples from each class of the SHREC 2013 dataset after the preprocessing decimation step (for more details see paragraph "Decimation preprocessing" in Section 4.2.2.3) is

[2] http://dataset.dcc.uchile.cl/

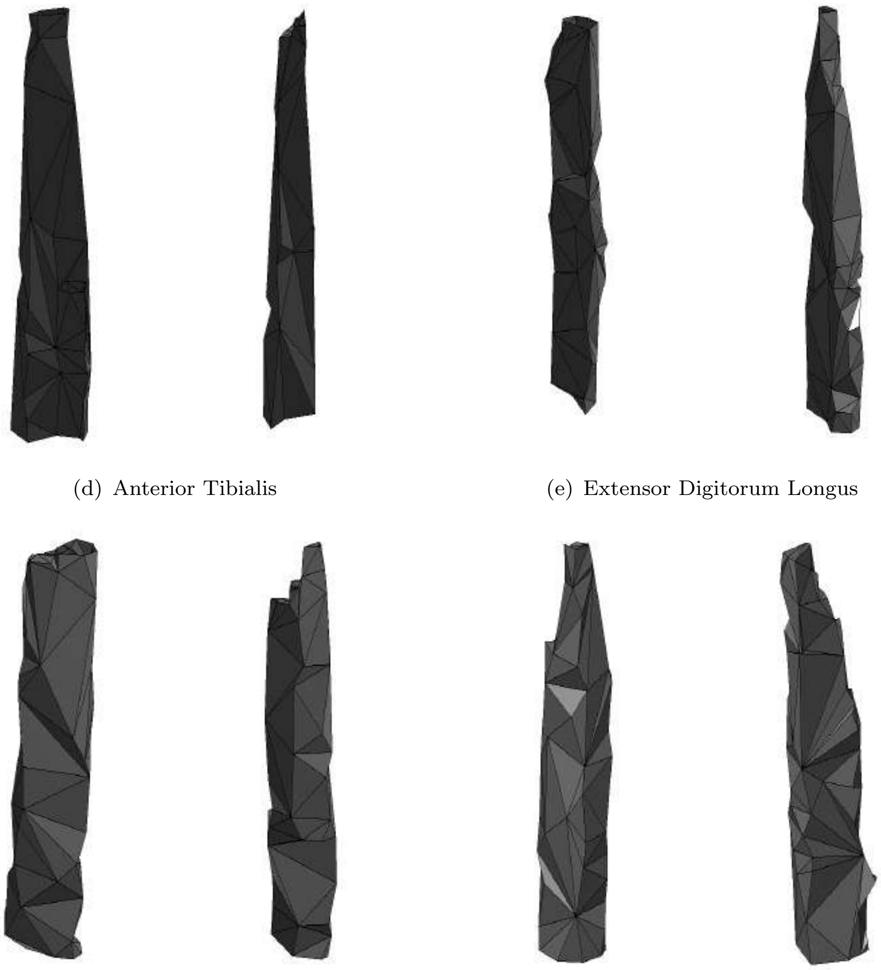


(a) Edial Gastrocnemius

(b) Lateral Gastrocnemius

(c) Soleus

FIGURE 4.11: An example of the seven segmented muscles of the calf as 3D surface meshes from a healthy subject on the left and from a patient with neuromuscular disease on the right (continued).



(d) Anterior Tibialis

(e) Extensor Digitorum Longus

(f) Peroneous Longus

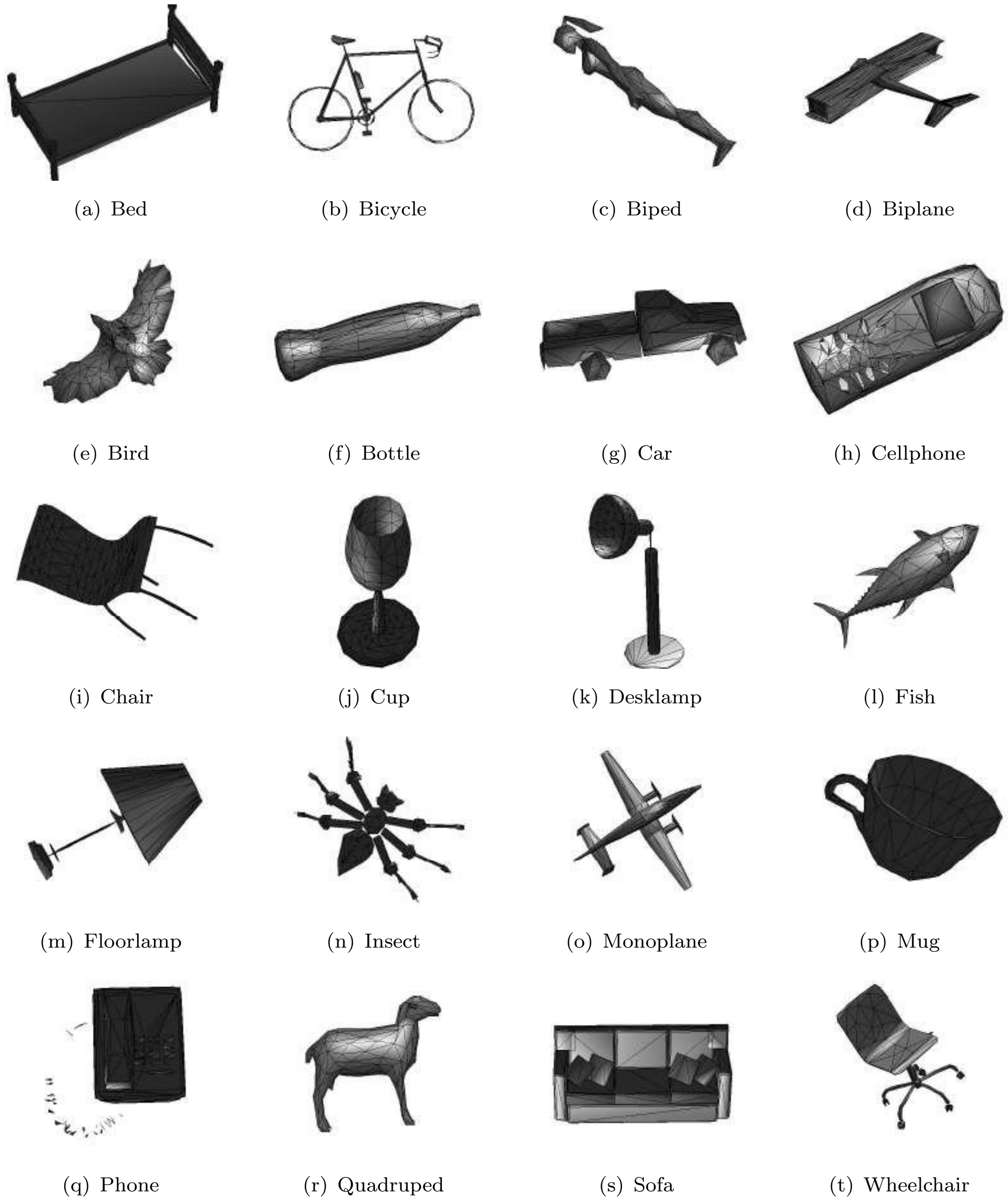
(g) Posterior Tibialis

Figure 4.11: An example of the seven segmented muscles of the calf as 3D surface meshes from a healthy subject on the left and from a patient with neuromuscular disease on the right.

illustrated in Figure 4.12. Finally, we should note that the task is to categorize each object among the 20 different semantic classes using a one-vs-rest approach.

4.2.2.3 Decimation preprocessing

Both the neuromuscular dystrophy dataset and the SHREC 2013 dataset consist of 3D meshes with a very big number of vertices and edges. For example for the neuromuscular dystrophy dataset for soleus muscle the average number of vertices is 25430 and the average number of edges is 76291, while for the SHREC2013 dataset the average number of vertices is 10446 and the average number of edges is 29920. Since the Weisfeiler-Lehman algorithm is quadratic to the number of edges and linear to number of vertices (see paragraph "Complexity" in Section 3.1), a larger number of vertices and edges



(a) Bed (b) Bicycle (c) Biped (d) Biplane

(e) Bird (f) Bottle (g) Car (h) Cellphone

(i) Chair (j) Cup (k) Desklamp (l) Fish

(m) Floorlamp (n) Insect (o) Monoplane (p) Mug

(q) Phone (r) Quadruped (s) Sofa (t) Wheelchair

FIGURE 4.12: Examples of each of the 20 classes of the SHREC 2013 dataset after the decimation preprocessing step.

could make the computation infeasible. For these reasons a decimation preprocessing was performed to simplify the 3D shapes in both datasets. Additionally, in the context of the *pyramid Weisfeiler-Lehman graph representation*, decimation can be viewed as an important source of regularization. For the neuromuscular dystrophy dataset the 3D meshes were simplified with the decimation algorithm incorporated in the itk-snap program, keeping on average 4308 number of vertices and 13598 number of edges. For the SHREC 2013 dataset the 3D meshes were simplified to 500 faces using the qslim program [Garland and Heckbert, 1997, 1998], keeping on average 350 vertices and 819 edges. Examples of the 3D meshes before and after the decimation process are shown in Figure 4.9 and in Figure 4.10 for the neuromuscular dystrophy dataset and the SHREC 2013 dataset, respectively. Overall, this preprocessing step increases the speed of the algorithm and works also a regularizer on the pyramid quantized Weisfeiler-Lehman graph representation.

4.2.3 Node Labels' description

As we denoted above, we view the 3D surface meshes as labeled graphs $G = (V, E, \mathcal{L})$, where $\mathcal{L} : V \to \mathbb{R}^d$ is the label function and the label of each vertex is defined as a concatenation of a number of local properties of the 3D surface mesh. In this section we present the local properties used as labels on the vertices.

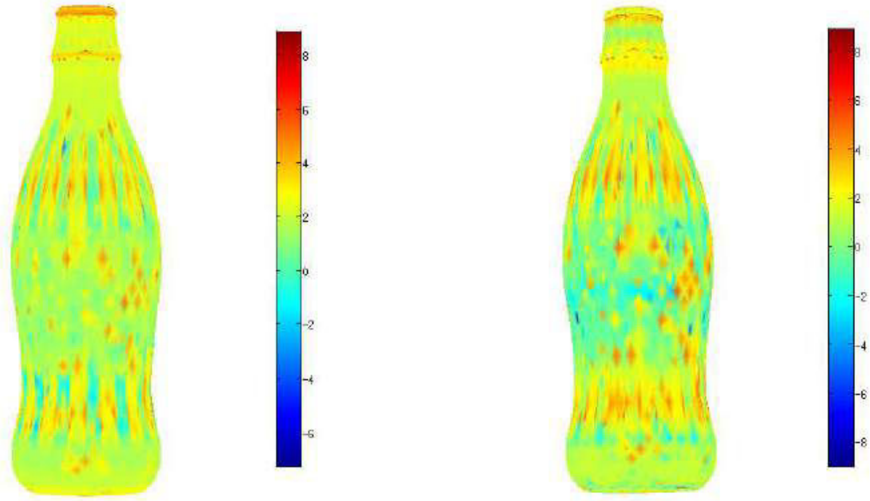
4.2.3.1 Curvature

The first attributes we select are the two principal curvatures k_1 and k_2 of each vertex of the 3D surface mesh, which are attached as a 2D continuous vector features. The normal curvature k_n of a surface in some direction is defined as the reciprocal of the radius of the circle that best approximates a normal slice of surface in that direction. Specifically, the normal curvature is defined as

$$k_n = \begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} \tag{4.1}$$

where k_1 and k_2 are the principal curvatures and $\begin{pmatrix} s & t \end{pmatrix}$ is the unit-length vector in the local tangent plane that express the principal directions, i.e. the directions in which the normal curvature reaches its minimum and maximum. We estimate the value at each vertex as a weighted average over the principal curvature features of the immediately adjacent triangulated faces [Rusinkiewicz, 2004]. Examples of the two principal curvatures on each node are shown in Figure 4.13 for a bottle object from the SHREC2013 dataset, while Figure 4.14 shows the minimum curvature feature on the muscle soleus of the calf from the neuromuscular dataset for a patient with a neuromuscular disease and a healthy subject.

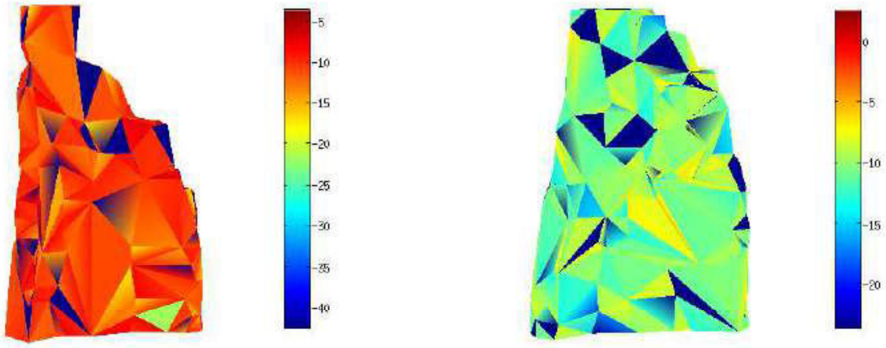
Apart from the principal curvatures that are used as node attributes in both datasets, in the SHREC2013 dataset we also used as node attributes local multi-viewpoint rendering features as defined in the following paragraph, but only for the 3D surface mesh structure that is enclosed within a given radius of the current node.



(a) Minimum Curvature

(b) Maximum Curvature

FIGURE 4.13: An example of the two principal curvatures on a bottle object from the SHREC2013 dataset in a logarithmic scale. On the left, Figure 4.13(a) shows the minimum curvature, while the Figure 4.13(a) on the right shows the maximum curvature. (Figure best viewed in color.)



(a) Disease

(b) Healthy

FIGURE 4.14: An example of the minimum curvature on the muscle soleus of the calf from the neuromusclar dystrophy dataset. On the left, Figure 4.14(a) shows the minimum curvature for a patient with a neuromuscular disease, while on the right Figure 4.14(b) shows the minimum curvature for a healthy subject. (Figure best viewed in color.)

4.2.3.2 Multi-viewpoint rendering descriptors

A successful method for 3D shape classification that has been previously proposed, is based on rendering shapes from multiple viewpoints and developing kernels based on these rendered images [Ohbuchi and Furuya, 2010]. We complement our continuous vector labeled graph representation with multi-viewpoint rendering features for the more complicated SHREC 2013 dataset. As we cannot assume a canonical basis for specifying

Algorithm 4.2 The statistical learning pipeline for 3D shape with pyramid quantized Weisfeiler-Lehman kernel.

```
Require: Training set of $D = \{(G_i^{'}, y_i), i = 1, \ldots, n\}$ where $G_i^{'} = (V_i^{'}, E_i^{'})$ is a graph.
1: Decimate the graphs $G_i^{'} = (V_i^{'}, E_i^{'})$ into $G_i = (V_i, E_i)$ where $|V_i^{'}| \leq |V_i|$ and $|E_i^{'}| \leq |E_i|$.
2: Calculate the curvature and/or multi-viewpoint rendering descriptor for each vertex and label the graphs $G_i = (V_i, E_i, \mathcal{L}_i)$, where $\mathcal{L}_i : V_i \rightarrow \mathbb{R}^d$ .
3: for each level in the quantization pyramid do
4:     Label the nodes of all graphs according to the data guided quantization of the vector label.
5:     Compute the Weisfeiler-Lehman statistics for the given quantization level over all graphs and calculate the intersection kernel $k_{i-WLsubtree}^{(h)}$.
6: end for
7: Combine the kernels $k_{i-WLsubtree}^{(h)}$ across all levels given a predefine weighted scheme or multiple kernel learning.
```

the 3D coordinates of the surface control points, we use a principal component analysis step to determine one. The multi-viewpoint rendering descriptor for a given vertex on the graph is calculated for a given percentage of the radius on the graph. For comparison, we also develop a multi-viewpoint rendering baseline on the whole graph. Similarly, we use a principal components analysis step to determine a basis. We then render images in these canonical bases and compute (non-)linear kernels. We have explored linear, polynomial of 2nd and 3rd degree and Gaussian kernels. As the third and the second degree polynomial kernel performed best for the muscle and SHREC2013 dataset, respectively, we use these rendering baselines in Section 4.2.5.

4.2.4 Method

As we already mentioned above, we view the 3D surface models as graphs $\mathcal{G}(V, E)$, where V is the finite set of points in the 3D space and E is the set of connections between two points in order to form triangles or curved surfaces. We further annotate each vertex using the local features, defined in Section 4.2.3, in order to take advantage of shape's information. Before we incorporate local features as labels on the vertices, we simplify the mesh, as noted in Section 4.2.2.3, due to the large sizes of the graphs. Since we end up with a continuous vector labeled graphs, we use the *pyramid quantized Weisfeiler-Lehman graph representation* with a data guided binning scheme (see Section 3.3.1.2) to create subtree statistics over the graphs for comparison. For all the levels of quantization, we calculate the intersection kernel over the previous calculated subtree statistics (see Section 3.3.2), resulting in a number of kernels, one per pyramid level as in Equation 3.11. To combine the kernels from all pyramid levels into one, we

	WLpyramid	pyramid BoW	Rendering	Combined
Accuracy	78.00%	73.00%	75.50%	82.93%
AUC	0.6410	0.6361	0.6300	0.6648

Table 4.2: The mean accuracy and the mean area under the ROC curve (AUC) on the neuromuscular dystrophy dataset. The compared methods are (left to right) the *pyramid quantized Weisfeiler-Lehman kernel* (WLpyramid), the pyramid bag of words model (pyramid BoW), the multi-viewpoint rendering images procedure (Rendering) and a combination of the multi-viewpoint rendering procedure with the *pyramid quantized Weisfeiler-Lehman kernel* (Combined). Note that the chance is 65.5% accuracy.

follow two different approaches, one for each dataset in order to maximize their performance. For the neuromuscular dystrophy dataset we use a equal fixed weight strategy (see Section 3.4.1.2), while for the SHREC 2013 dataset we use a multiple kernel learning approach (see Section 3.4.1.1). An overview of the pipeline for the 3D shape dataset with the *pyramid quantized Weisfeiler-Lehman kernel* is shown in Algorithm 4.2.

4.2.5 Results

For both datasets we use the same experimental setup, a double cross-validation procedure. The inner 5 fold cross-validation procedure is used for parameter selection, while the outer 10 fold cross-validation procedure is used for evaluating the performance. We only report results from the outer 10 fold cross-validation procedure. We also compare the *pyramid quantized Weisfeiler-Lehman kernel* with two other methods on both datasets (a) a pyramid bag of words model, which is the *pyramid quantized Weisfeiler-Lehman kernel* for depth $h = 0$, and (b) a multi-viewpoint rendering procedure (see details in paragraph "Multi-viewpoint rendering descriptor" in Section 4.2.3.2) following the same experimental setup. We also present the results obtained from the combination of the best multi-viewpoint rendering representation with the best *pyramid quantized Weisfeiler-Lehman kernel*.

The performance for the neuromuscular dystrophy dataset of the *pyramid quantized Weisfeiler-Lehman kernel*, as well as the three other methods is shown in Table 4.2 and in Figure 4.15. The performance is evaluated as the mean accuracy and the mean area under the Receiver Operating Characteristic curve (AUC) in Table 4.2 over a 10 fold cross validation procedure. The respective ROC curves can be seen in Figure 4.15. The *pyramid quantized Weisfeiler-Lehman kernel* outperforms both the pyramid bag of words method and the multi-viewpoint image rendering procedure. The overall best performance is achieved when we combined the *pyramid quantized Weisfeiler-Lehman kernel* with the multi-viewpoint rendering approach with a mean accuracy of approximately 83% and a mean AUC of 0.6648.

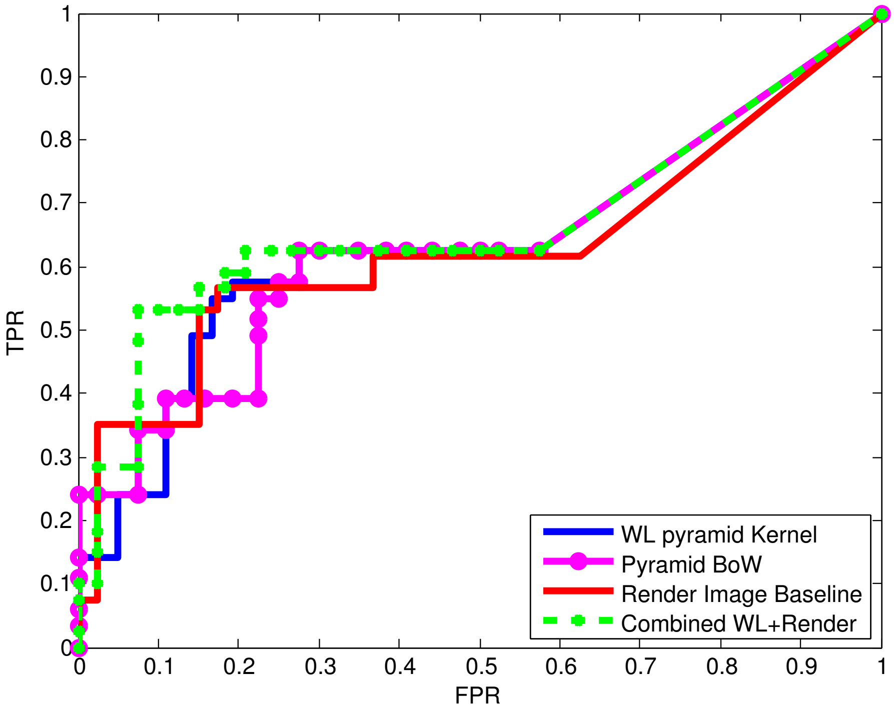
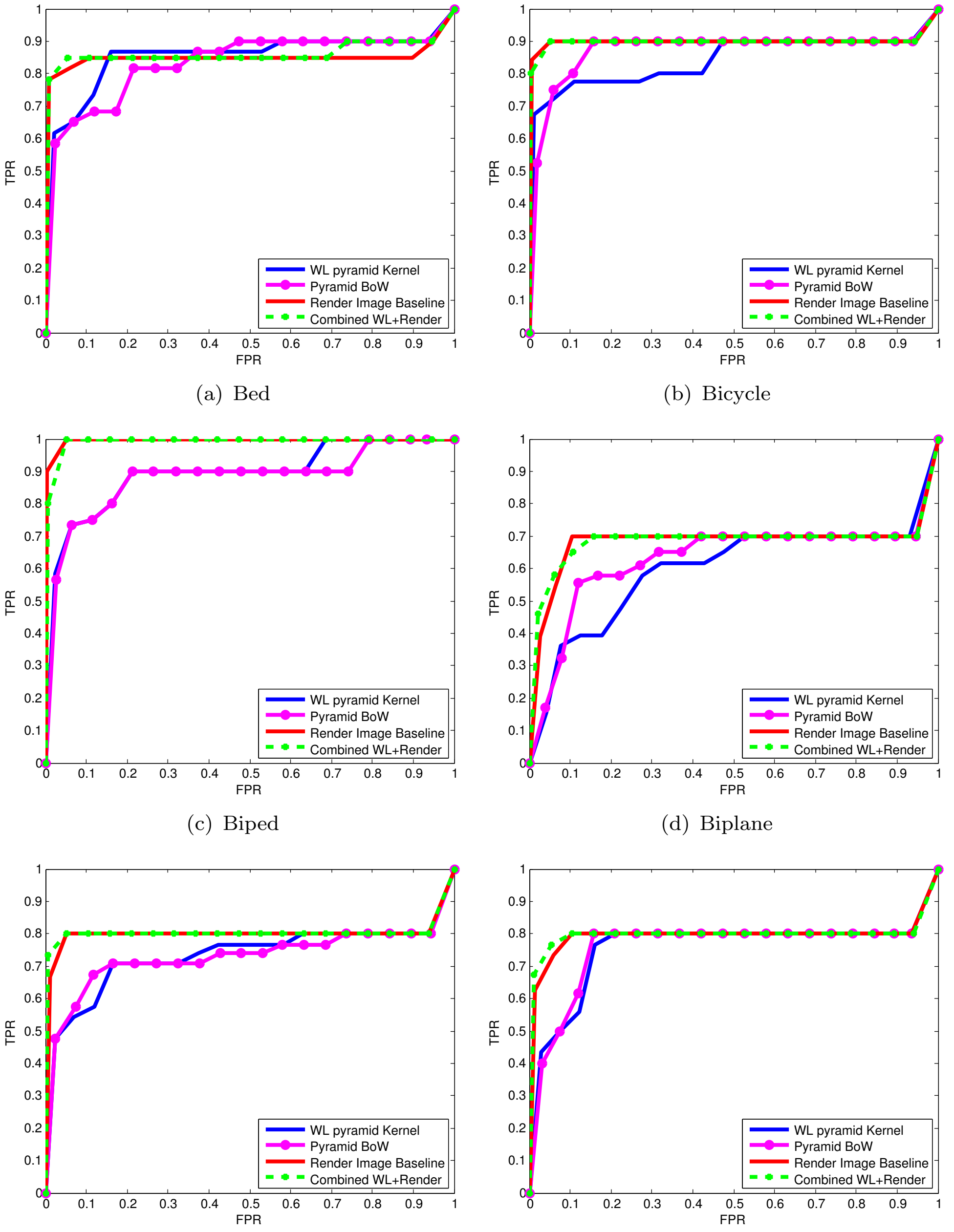


FIGURE 4.15: The mean area under the ROC curve for the neuromuscular dystrophy dataset over 10 fold cross-validation procedure. The compared methods are the *pyramid quantized Weisfeiler-Lehman kernel* (WLpyramid) in blue, the pyramid bag of words model (pyramid BoW) in magenta, the multi-viewpoint rendering images procedure (Rendering) in red and a combination of the multi-viewpoint rendering procedure with the *quantized Weisfeiler-Lehman pyramid kernel* (Combined) in green.

The performance for the SHREC 2013 dataset of the *pyramid quantized Weisfeiler-Lehman kernel*, as well as for the other methods is shown in Table 4.3 and in Figure 4.16. The performance is evaluated as the mean area under the Receiver Operating Characteristic curve (AUC of ROC curve) over a 10 fold cross-validation procedure. The overall best performance is achieved when we combined the *pyramid quantized Weisfeiler-Lehman kernel* with the multi-viewpoint rendering images with a mean AUC of approximately 0.85 across all 20 classes. A Wilcoxon signed-rank test showed that the combined method performed better than all other methods with high statistical significance ($p < 10^{-3}$).

We further show the learned weight of the *pyramid quantized Weisfeiler-Lehman kernel* for the SHREC 2013 dataset in Figure 4.17 and in Figure 4.18. Figure 4.17 shows an example of the learned weights on a 3D object of the bird class for three different subtree depths ($h \in \{0, 1, 2\}$) of the Weisfeiler-Lehman algorithm, while Figure 4.18 shows the learned weight over all levels of the *pyramid quantized Weisfeiler-Lehman kernel* for depth $h = 1$ for all one-vs-rest classifiers. Note that the values of the learned weights increase as the color changes from blue to red.



(a) Bed

(b) Bicycle

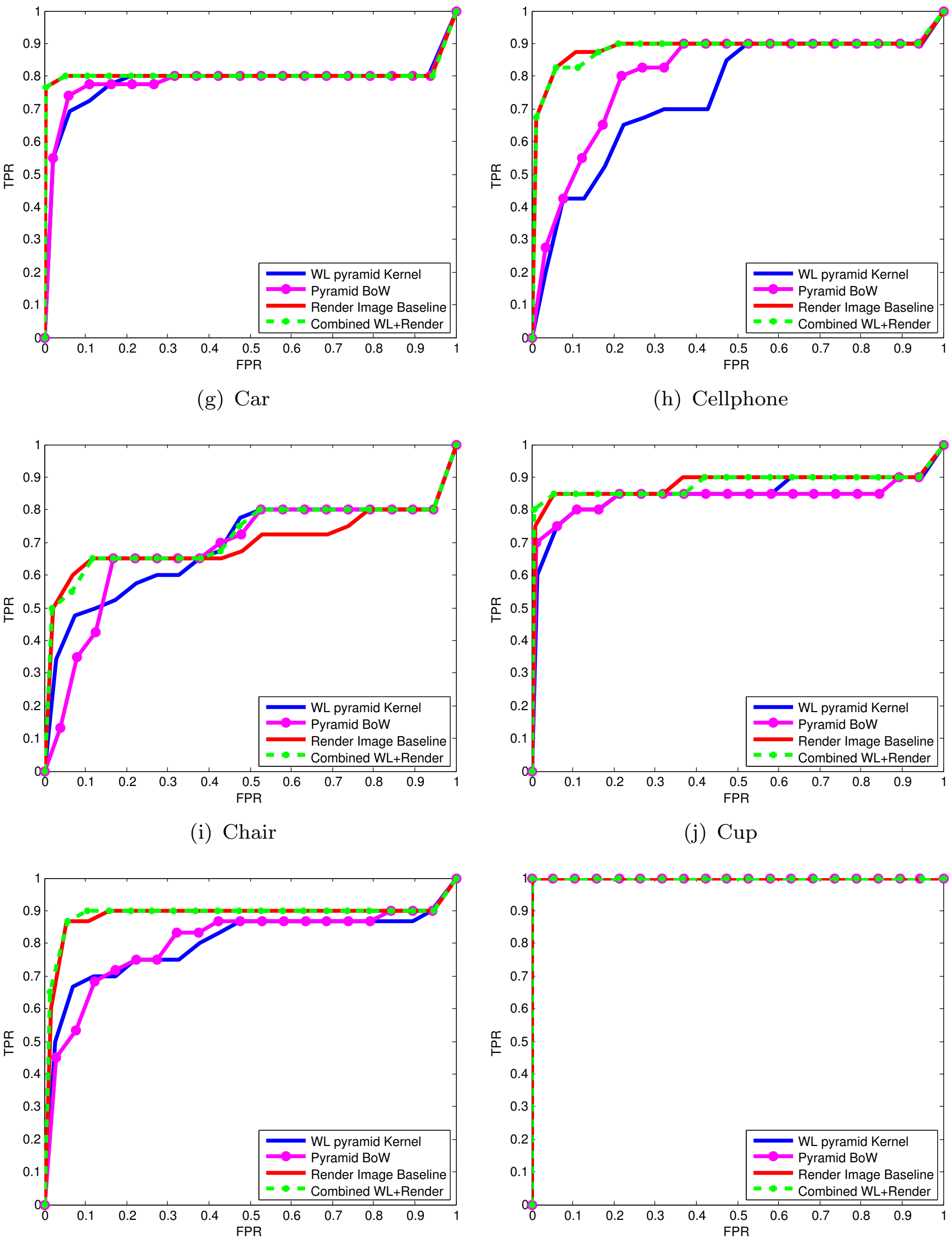
(c) Biped

(d) Biplane

(e) Bird

(f) Bottle

Figure 4.16: The Receiver Operating Characteristic Curves for all one-vs-rest classifiers of the SHREC2013 dataset over a 10 fold cross-validation procedure. In blue is the *pyramid quantized Weisfeler-Lehman kernel* (WL pyramid Kernel), in magenta a pyramid bag of words approach (Pyramid Bow), in red is the Render Image descriptor (Render Image Baseline) and in green the combination of the *pyramid quantized Weisfeler-Lehman kernel* with the Render Image descriptor (Combined WL+Render). (Best viewed in color) (continued)



(g) Car

(h) Cellphone

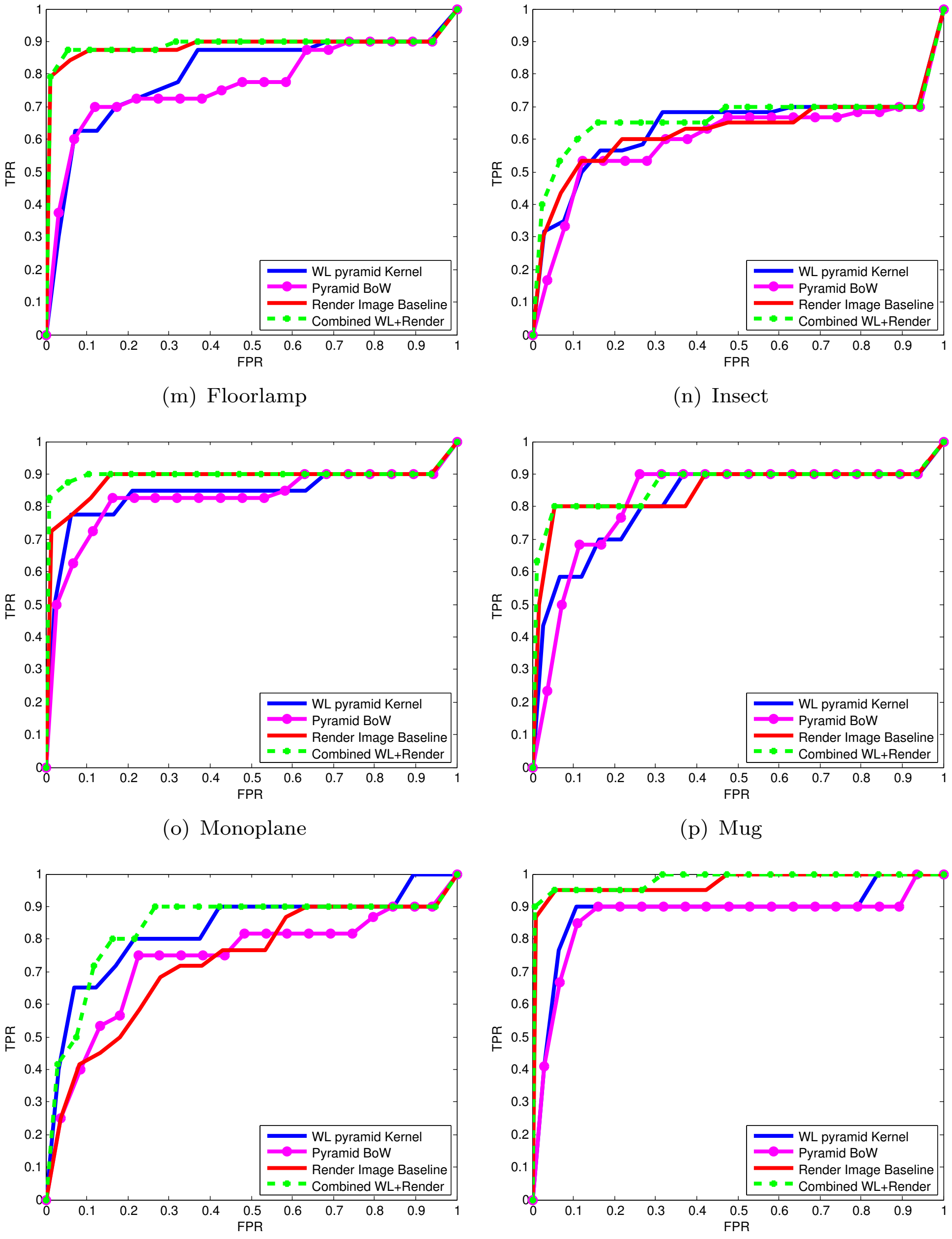
(i) Chair

(j) Cup

(k) Desklamp

(l) Fish

Figure 4.16: The Receiver Operating Characteristic Curves for all one-vs-rest classifiers of the SHREC2013 dataset over a 10 fold cross-validation procedure. In blue is the *pyramid quantized Weisfeler-Lehman kernel* (WL pyramid Kernel), in magenta a pyramid bag of words approach (Pyramid Bow), in red is the Render Image descriptor (Render Image Baseline) and in green the combination of the *pyramid quantized Weisfeler-Lehman kernel* with the Render Image descriptor (Combined WL+Render). (Best viewed in color) (continued)



(m) Floorlamp

(n) Insect

(o) Monoplane

(p) Mug

(q) Phone

(r) Quadruped

Figure 4.16: The Receiver Operating Characteristic Curves for all one-vs-rest classifiers of the SHREC2013 dataset over a 10 fold cross-validation procedure. In blue is the *pyramid quantized Weisfeler-Lehman kernel* (WL pyramid Kernel), i magenta a pyramid bag of words approach (Pyramid Bow), in red is the Render Image descriptor (Render Image Baseline) and in green the combination of the *pyramid quantized Weisfeler-Lehman kernel* with the Render Image descriptor (Combined WL+Render). (Best viewed in color) (continued)

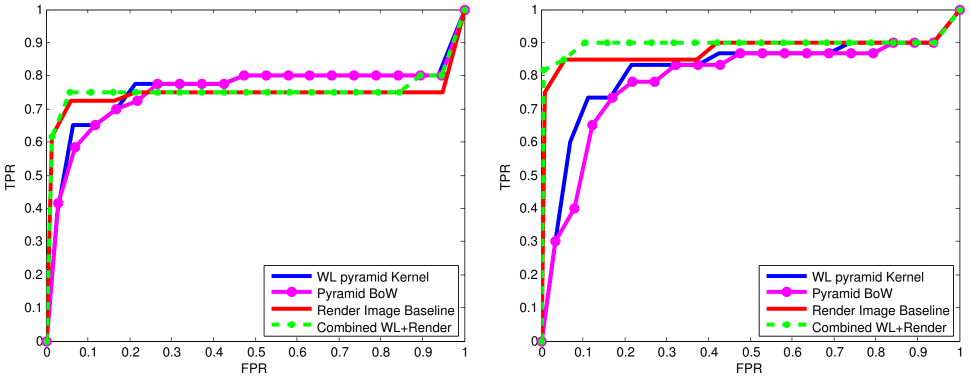
Class	WLpyramid	pyramid BoW	Rendering	Combined
Bird	0.85	0.83	0.85	**0.86**
Bicycle	0.84	0.87	0.90	0.90
Biped	0.89	0.88	0.99	0.99
Biplane	0.60	0.63	0.68	**0.69**
Bird	0.73	0.73	0.80	0.80
Bottle	0.76	0.76	0.79	**0.80**
Car	0.78	0.79	0.80	0.80
CellPhone	0.74	0.80	0.88	**0.89**
Chair	0.69	0.68	0.70	**0.72**
Cup	0.85	0.84	0.88	0.88
Desklamp	0.80	0.80	0.88	**0.89**
Fish	1.00	1.00	1.00	1.00
Floorlamp	0.80	0.77	0.89	0.89
Insect	0.64	0.60	0.62	**0.66**
Monoplane	0.84	0.82	0.88	**0.90**
Mug	0.82	0.82	0.85	**0.87**
Phone	0.83	0.74	0.72	**0.83**
Quadruped	0.89	0.86	0.97	**0.98**
Sofa	0.76	0.75	0.74	**0.75**
Wheelchair	0.81	0.79	0.88	**0.90**
Average	0.80	0.79	0.84	0.85

Table 4.3: The mean area under the curve on the SHREC 2013 dataset over 10 fold cross-validation procedure. The compared methods are (left to right) The *pyramid quantized Weisfeiler-Lehman kernel* (WLpyramid), the pyramid bag of words model (pyramid BoW), the multi-viewpoint rendering procedure (Rendering) and a combination of the multi-viewpoint rendering procedure with the pyramid quantizes Weisfeiler-Lehman kernel (Combined). The Area under the curve is given for all one-vs-rest classifiers as well as the average across all classifiers. In bold are the one-vs-rest classifier where the combined classifier outperforms the multi-viewpoint rendering procedure. A Wilcoxon signed-rank test showed that the combined method performed better than all other methods with high statistical significance ($p < 10^{-3}$).

4.2.6 Discussion

In the neuromuscular dystrophy dataset, the *pyramid quantized Weisfeiler-Lehman kernel* performs substantially better than both the pyramid Bag of Words approach as well as the multi-viewpoint rendering technique. These techniques clearly contain complementary information as the combined method performs best.

We have further confirmation from the SHREC 2013 dataset that the *pyramid quantized Weisfeiler-Lehman kernel* contains complementary information to the multi-viewpoint rendering baseline. The combined approach gives the best performance in 13 out of the 20 classes, while never performing worse than the baselines. A Wilcoxon signed-rank test also showed that the combined method performed better than all other methods with high statistical significance ($p < 10^{-3}$).



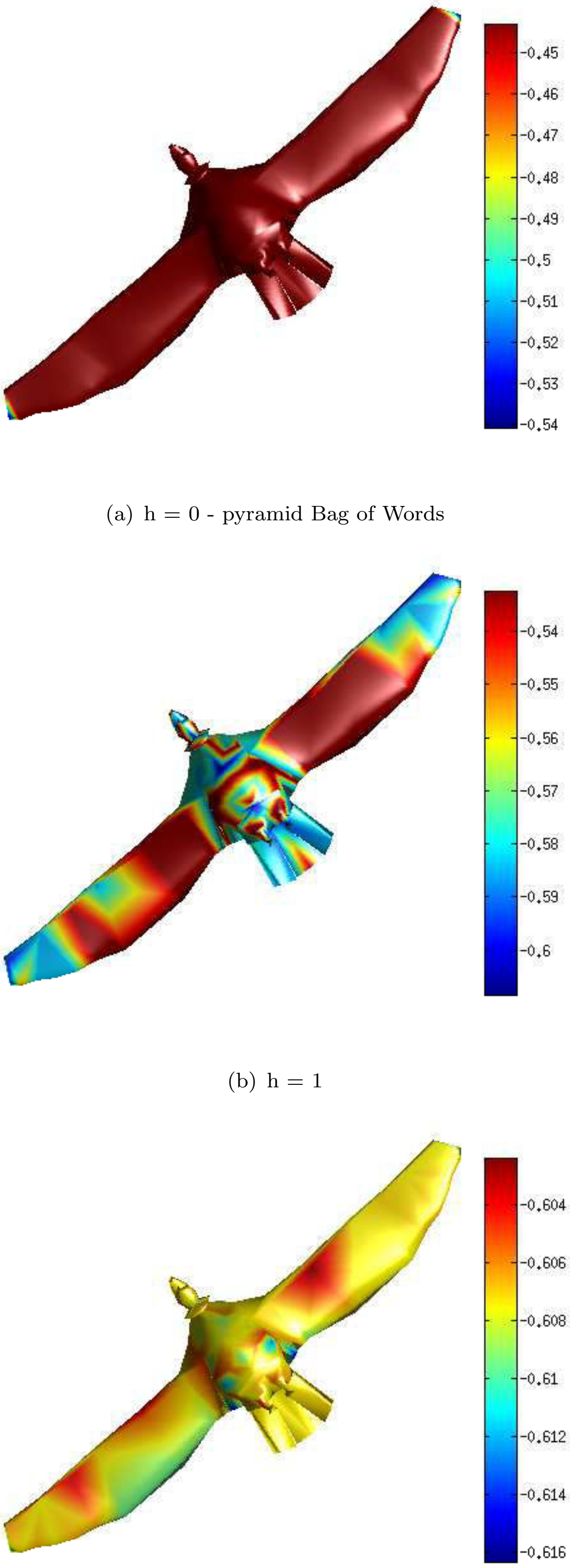
(s) Sofa

(t) Wheelchair

Figure 4.16: The Receiver Operating Characteristic Curves for all one-vs-rest classifiers of the SHREC2013 dataset over a 10 fold cross-validation procedure. In blue is the *pyramid quantized Weisfeler-Lehman kernel* (WL pyramid Kernel), in magenta a pyramid bag of words approach (Pyramid Bow), in red is the Render Image descriptor (Render Image Baseline) and in green the combination of the *pyramid quantized Weisfeler-Lehman kernel* with the Render Image descriptor (Combined WL+Render). (Best viewed in color)

As shapes are commonly represented by surface meshes, a natural approach is to use these graphs for categorization and retrieval. In this section we have shown two such applications, one on medical image analysis and one on generic 3D shape classification. The *pyramid quantized Weisfeiler-Lehman kernel* is a flexible and efficient method for learning from graphs with continuous, vector-valued node labels, such as annotations of local curvature. Furthermore, visualizations of the learned discriminant function is feasible, providing rich information about the discriminative power of each 3D shape.

In this work, we have not directly incorporated any features in our node labels capturing surface reflectance, color, or texture. This is an interesting area for future research. Learned shape retrieval by discriminative training of a Mahalanobis metric [Weinberger and Saul, 2009] is another interesting possible future direction. Finally, in this work, we have made three main contributions (a) we developed a novel framework for shape classification based on the interpretation of shape meshes as annotated graphs, (b) we applied a generalization of the Weisfeiler-Lehman graph kernel to continuous node labels, the *pyramid quantized Weisfeiler-Lehman kernel*, and (c) we performed experiments on medical imaging and semantic shape classification tasks, showing that the *pyramid quantized Weisfeiler-Lehman kernel* contains complementary information to baseline methods and that the best results are achieved by a combination of information sources.

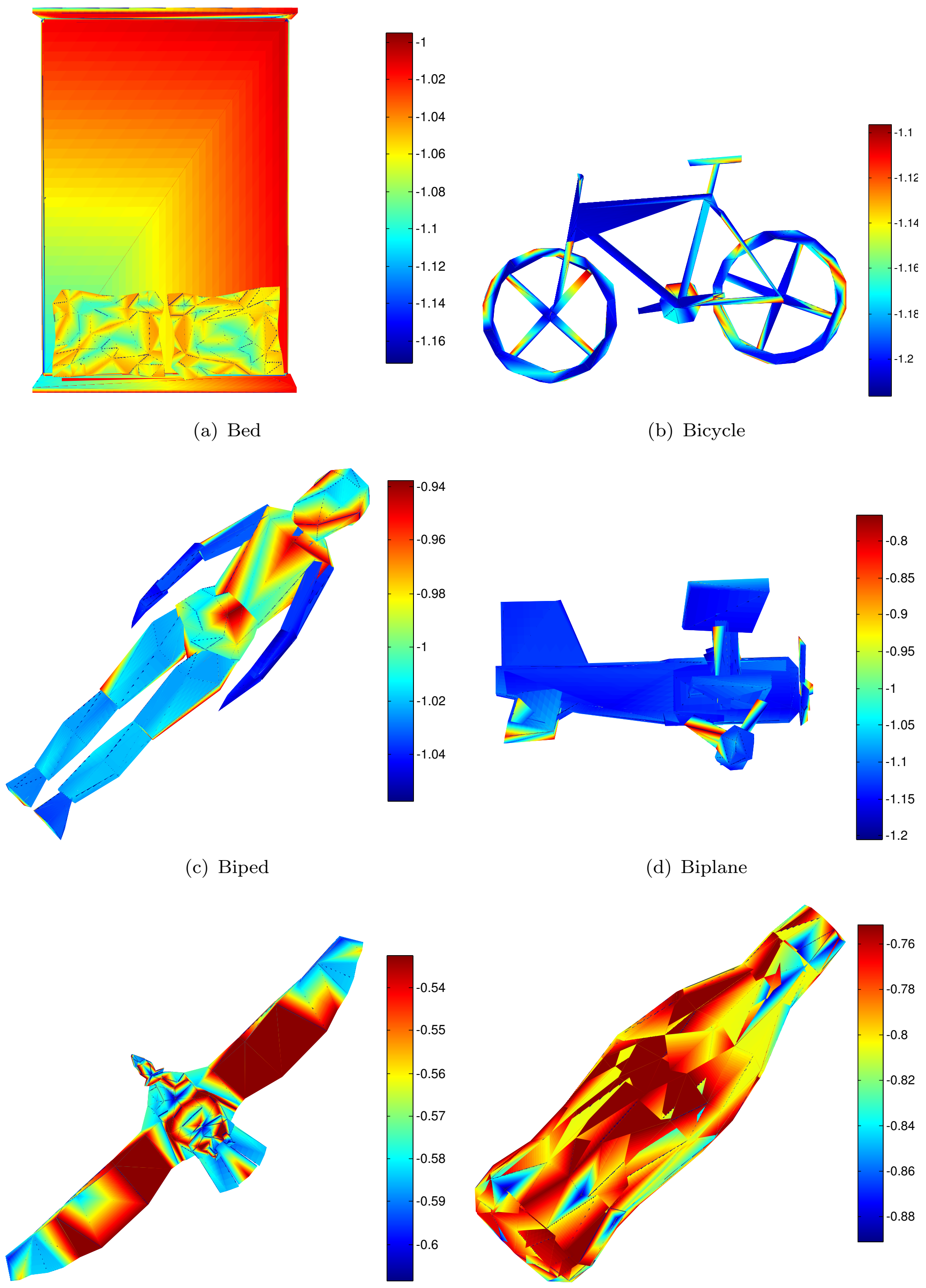


(a) h = 0 - pyramid Bag of Words

(b) h = 1

(c) h = 2

FIGURE 4.17: An example of the learned weights of the *pyramid quantized Weisfeiler-Lehman Kernel* on a 3D object of the class bird from the SHREC 2013 dataset for three different subtree depths ($h \in \{0, 1, 2\}$). The values of the learned weights increase as the color changes from blue to red. See Section 3.4.1.3 for details of the visualization. (Figure best viewed in color).



(a) Bed

(b) Bicycle

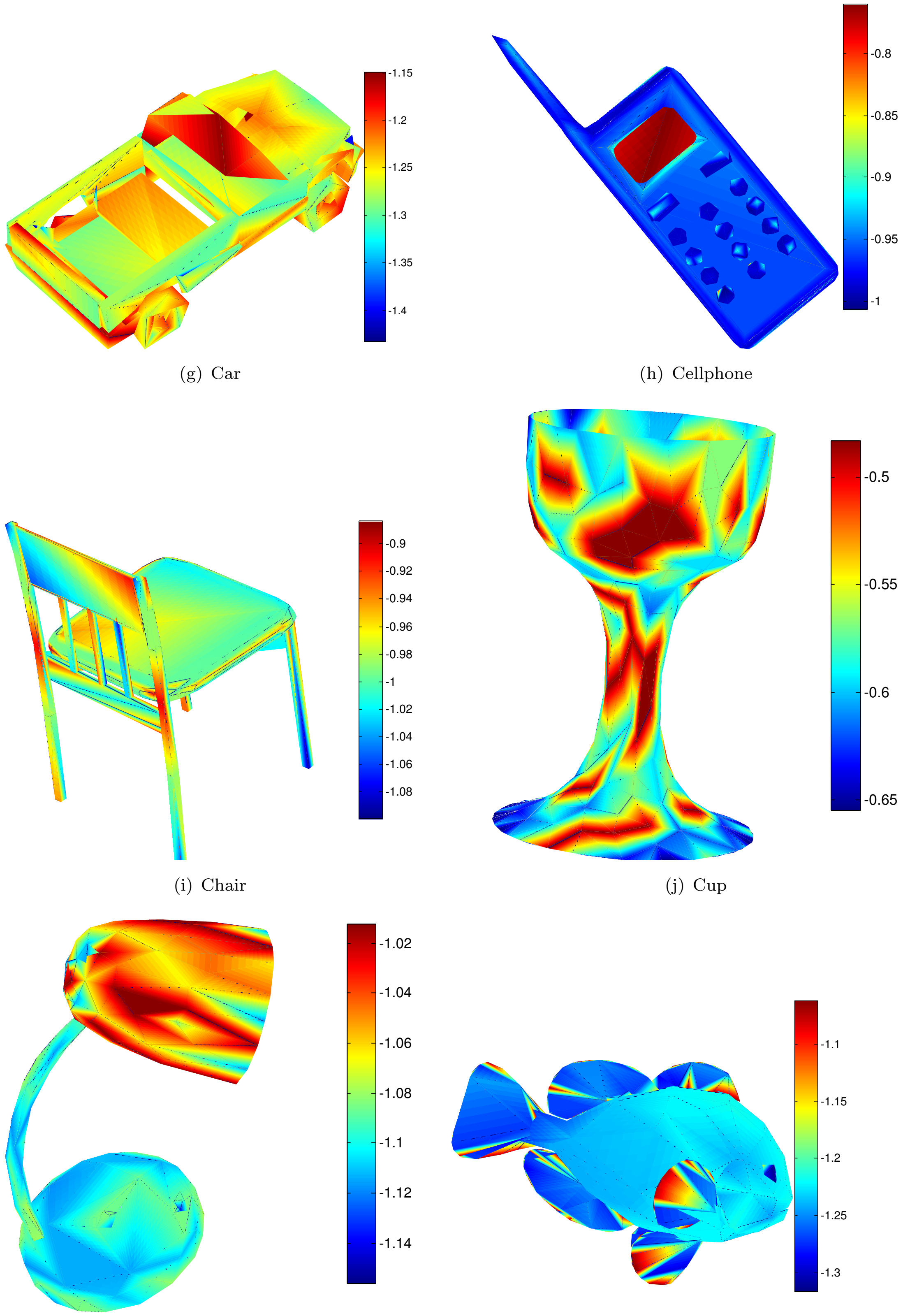
(c) Biped

(d) Biplane

(e) Bird

(f) Bottle

FIGURE 4.18: Visualization of the learned weights of the *pyramid quantized Weisfeiler-Lehman kernel* of subtree patterns with depth $h = 1$ for each vertex on the 3D surface mesh per Class-vs-Rest Classifier for the SHREC2013 dataset. The values of the learned weights increase as the color changes from blue to red. The evaluation of the weight per vertex is derived in Section 3.4.1.3.(Figure best viewed in color.) (continued)



(g) Car

(h) Cellphone

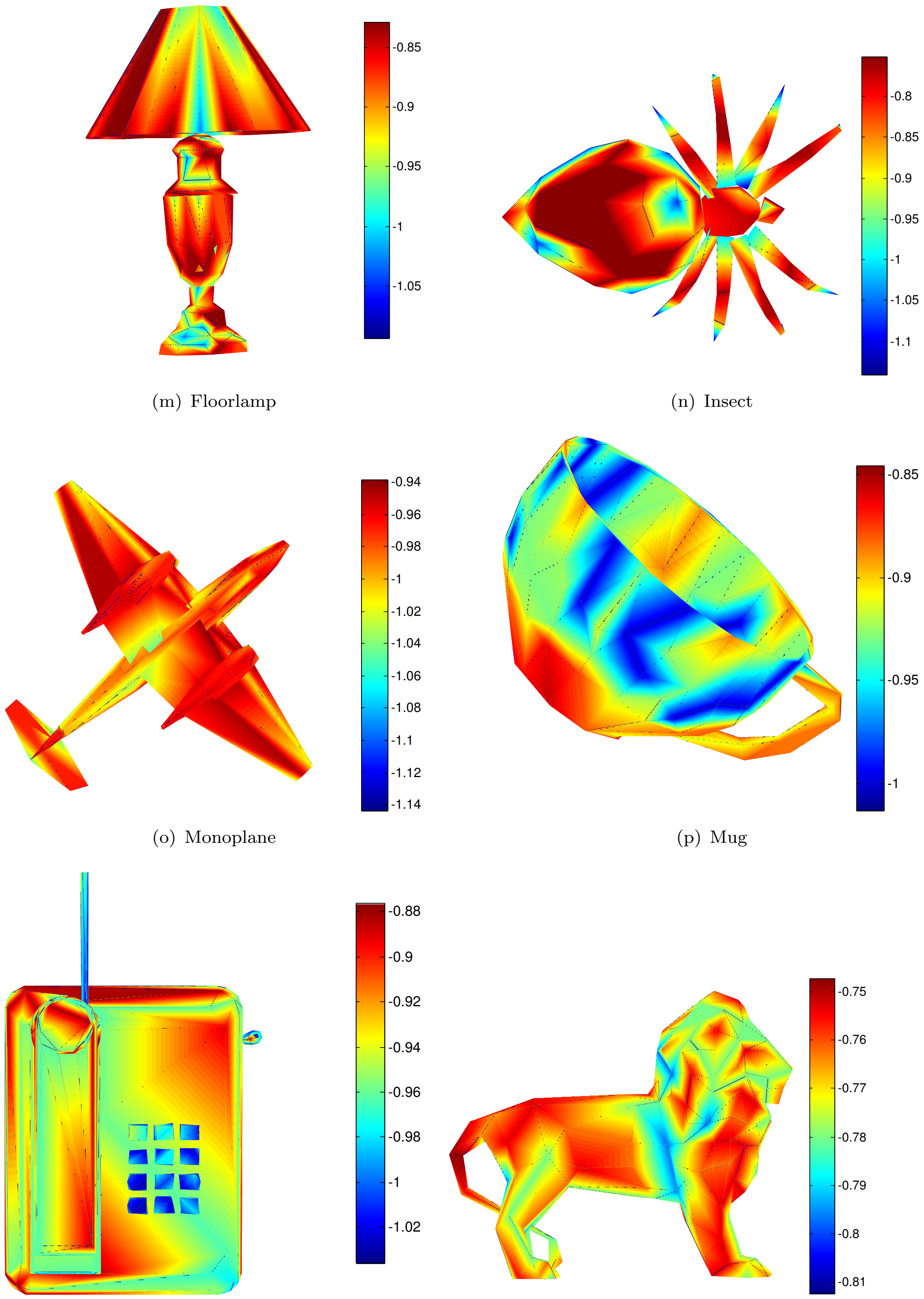
(i) Chair

(j) Cup

(k) Desklamp

(l) Fish

FIGURE 4.18: Visualization of the learned weights of the *pyramid quantized Weisfeiler-Lehman kernel* of subtree patterns with depth $h = 1$ for each vertex on the 3D surface mesh per Class-vs-Rest Classifier for the SHREC2013 dataset. The values of the learned weights increase as the color changes from blue to red. The evaluation of the weight per vertex is derived in Section 3.4.1.3.(Figure best viewed in color.) (continued)



(m) Floorlamp

(n) Insect

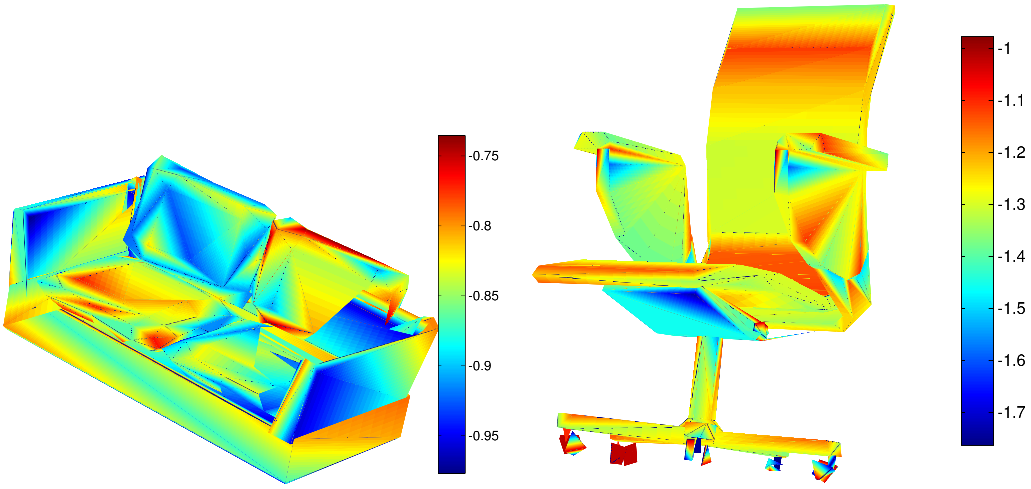
(o) Monoplane

(p) Mug

(q) Phone

(r) Quadruped

FIGURE 4.18: Visualization of the learned weights of the *pyramid quantized Weisfeiler-Lehman kernel* of subtree patterns with depth $h = 1$ for each vertex on the 3D surface mesh per Class-vs-Rest Classifier for the SHREC2013 dataset. The values of the learned weights increase as the color changes from blue to red. The evaluation of the weight per vertex is derived in Section 3.4.1.3.(Figure best viewed in color.) (continued)



(s) Sofa

(t) Wheelchair

Figure 4.18: Visualization of the learned weights of the *pyramid quantized Weisfeiler-Lehman kernel* of subtree patterns with depth $h = 1$ for each vertex on the 3D surface mesh per Class-vs-Rest Classifier for the SHREC2013 dataset. The values of the learned weights increase as the color changes from blue to red. The evaluation of the weight per vertex is derived in Section 3.4.1.3.(Figure best viewed in color.)

Chapter 5

Regularization methods for analyzing Magnetic Resonance Imaging data

Magnetic resonance imaging (MRI) is a widespread medical imaging technique used to visualize internal structures of the body in detail. MRI makes use of the nuclear magnetic resonance (NMR) property to image nuclei of atoms inside the body. The NMR property allows the atomic nuclei in the body to be aligned when powerful magnetic field is applied. Applying different radio frequencies causes the nuclei to produce a rotating magnetic field that is detected by the MRI scanner and this information is recorded to construct the image of the scanned area of the body. Magnetic field gradients cause nuclei at different locations to rotate at different speeds, which allows spatial information to be recovered using Fourier analysis of the measured signal. MRI provides good contrast between the different soft tissues of the body, which makes it especially useful in imaging the brain, muscles, the heart, and cancers compared with other medical imaging techniques such as computed tomography (CT) or X-rays. Unlike CT scans or traditional X-rays, MRI does not use ionizing radiation [Berger, 2002].

In clinical practice, MRI is usually used to distinguish pathological tissue, such as a brain tumor, from normal tissue. There are a number of different types of MRI scans with different properties, but in this thesis we focus our analysis in the following MRI scans (a) T_1-weighted MRI (b) T_2-weighted MRI (c) diffusion MRI and (d) functional MRI. Detailed description for the functional MRI type is provided in Section 5.2.2, while descriptions for the T_1-weighted MRI, the T_2-weighted MRI and the diffusion MRI are provided in Section 5.3.2. Independent of the specific properties of each MRI type, when one analyzes clinical data sets with MRI scans has to deal with the fact that the number of observations is substantially smaller that the dimensionality of the data. Under these circumstances the use of sparsity regularizers is a powerful tool for improving the predictive performance and avoiding overfitting. The main methods considered here are the LASSO [Tibshirani, 1996], the Elastic Net [Zou and Hastie, 2005], and the k-support norm [Argyriou et al., 2012]. The former two are frequently applied sparsity regularizers developed in the statistics literature, while the latter is a recently introduced method that is mathematically related to the Elastic Net. A detailed description of these methods are presented in Section 5.1. We then evaluate the performance of these three different regularization methods on a fMRI analysis task in Section 5.2, while in Section 5.3 we evaluate the k-support norm method with the use of two different loss functions over a disease classification problem.

5.1 The regularization methods

Regularization, in mathematics and statistics, refers to a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting. This information is usually of the form of a penalty for complexity, such as restrictions for smoothness or bounds on the vector space norm. For the rest of this chapter, we assume that we have a sample of paired labeled training data $(X, Y) = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in \left(\mathbb{R}^d \times \mathbb{R}\right)^n$ where x_i is an independent variable and y_i is a dependent variable that we would like to be able to predict. All regularization methods can be expressed as follows:

$$\arg\min_{w \in \mathbb{R}^d} \lambda\Omega(w) + \mathcal{L}(w, X, Y) \tag{5.1}$$

where $\mathcal{L}(\cdot, \cdot, \cdot)$ is some loss function, $\lambda > 0$ is a scalar parameter controlling the degree of regularization and $\Omega(\cdot)$ is a scalar valued function monotonic in a norm of the learned weights $w \in \mathbb{R}^d$. Sparsity regularization is a key family of priors over linear functions that prevents overfitting, and aids interpretability of the resulting models by retaining a subset of the predictors and discarding the rest in a more continuous framework than typical subset selection procedures. Key to the mathematical understanding of sparsity regularizers is their interpretation as convex relaxations to quantities involving the ℓ_0 norm, which simply counts the number of non-zero elements of a vector.

In the following sections, we examine the well known LASSO and Elastic Net in Section 5.1.1 and in Section 5.1.2 respectively and the newly introduced k-support norm in Section 5.1.3.

5.1.1 LASSO

LASSO [Tibshirani, 1996] stands for the *Least Absolute Shrinkage and Selection Operator* and is a regularized version of the ordinary least squared method (OLS). The LASSO is defined by setting as loss function the squared error of the scalar output and as regularization term $\Omega(\cdot)$ the ℓ_1 norm of w, so Equation 5.1 is transformed as follows:

$$\arg\min_{w\in\mathbb{R}^d} \lambda\|w\|_1 + \frac{1}{n}\sum_{i=1}^{n}(\langle w, x_i\rangle - y_i)^2 \tag{5.2}$$

where the ℓ_1 norm is defined as the sum of the absolute values of the vector

$$\|w\|_1 = \sum_{j=1}^{d}\|w_j\| \tag{5.3}$$

and has an important interpretation as convex relaxation of the ℓ_0 norm, meaning it is the tightest sparsity norm that retains convexity. Note that as λ increases in Equation 5.2 that will cause more coefficients w_j to be exactly zero, providing a more sparse solution.

The LASSO constraint makes the solutions nonlinear in the y_i and there is no closed form expression as in least squares. Computing the LASSO solution is a quadratic programming problem and an efficient algorithm for computing the entire path of solutions as λ varies exist - it is a variation of the Least Angle Regression (LAR) called LAR-LASSO - making the LASSO computationally attractive. The LAR can be viewed as a forward stepwise regression where instead of adding one variable at a time, the algorithm adds "as much" of a predictor as it deserves. The idea of LAR is that it identifies variables, called the *active set*, that are most correlated with the response and moves their coefficients continuously toward the least-squared value. It increases the active set by one variable at the time, but controlling its influence by its coefficient. This process is continued until all variables are in the model, ending at the full least-squares fit. The LAR algorithm has at most $min(n-1, d)$ steps, the same order of computations as a single OLS fit. The LAR-LASSO variation allows a variable to be dropped from the active set of variables if its coefficient hits zero.

5.1.2 Elastic Net

The Elastic Net [Zou and Hastie, 2005] is a regularization and variable selection method that was introduced by Zou and Hastie in 2005. Elastic Net is defined by setting as loss function the squared error of the scalar output and as regularization term $\Omega(\cdot)$ a linear combination of the ℓ_1 and squared ℓ_2 norms of w, so Equation 5.1 is transformed as follows:

$$\arg\min_{w\in\mathbb{R}^d} \lambda_1\|w\|_1 + \lambda_2\|w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}(\langle w, x_i\rangle - y_i)^2 \tag{5.4}$$

where the ℓ_1 norm is defined in Equation 5.3 and the squared ℓ_2 norm is defined as

$$\|w\|_2^2 = \sum_{j=1}^{d} w_j^2. \tag{5.5}$$

This objective is a regularized version of the ordinary least squares and includes both ridge regression [Tikhonov, 1963] and LASSO [Tibshirani, 1996] as special cases by setting λ_1 or λ_2 to zero, respectively.

It should be noted that the Elastic Net is looser than the convex hull of a norm that combines ℓ_2 regularization with sparsity. Moreover, if there is a group of variables among which the pairwise correlations are very high, the LASSO (see Section 5.1.1) tends to select only one variable from the group and does not care which one is selected, ending with a higher variance predictor. Elastic Net on the contrary may overcome this disadvantage by selecting multiple variables from a group of correlated ones if that improves the prediction accuracy, assigning approximately equal coefficients to highly correlated variables.

Finally, the Elastic Net retains the computational advantage of the LASSO with the use of a variation of the LAR algorithm, the LAR-EN. The Elastic Net problem can be solved as a LASSO-type one considering the augmented paired data (X^*, Y^*) where

$$X^*_{(n+d)\times d} = (1+\lambda_2)^{-1/2}\begin{pmatrix} X \\ \sqrt{\lambda_2}I \end{pmatrix}, \qquad Y^*_{(n\times d)} = \begin{pmatrix} Y \\ 0 \end{pmatrix},$$

so Equation 5.4 can be written as:

$$\hat{w}^* = \arg\min_{\hat{w}^*\in\mathbb{R}^d} \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}\|\hat{w}^*\|_1 + \frac{1}{n}\sum_{i=1}^{n+d}(\langle \hat{w}^*, x_i^*\rangle - y_i^*)^2 \tag{5.6}$$

and the Elastic Net solution [Zou and Hastie, 2005] is defined as:

$$\hat{w} = \sqrt{(1+\lambda_2)}\hat{w}^*. \tag{5.7}$$

The LAR-EN algorithm takes advantage of the sparse structure of the augmented data when updating the Cholesky factorization of the active set in each iteration. Moreover, an explicit use of the X^* to compute all the quantities of the LAR algorithm can be avoided, by keeping record only the non-zero coefficients and the active set at each LAR-EN step. In the case where $d >> n$, it is not necessary to run the algorithm to the end. If the algorithm is stopped after m steps, it requires $O(m^3 + dm^2)$ operations.

5.1.3 k-support norm

The k-support norm [Argyriou et al., 2012] is a newly introduced regularization penalty that corresponds to the tightest convex relaxation of sparsity combined with the ℓ_2 norm. The k-support norm given an integer $k \in \{1, \dots, d\}$ can be computed as

$$\|w\|_k^{sp} = \left(\sum_{i=1}^{k-r-1} (|w|_i^{\downarrow})^2 + \frac{1}{r+1} \left(\sum_{i=k-r}^{d} |w|_i^{\downarrow} \right)^2 \right)^{\frac{1}{2}} \tag{5.8}$$

where $|w|_i^{\downarrow}$ is the ith largest element of the vector and r is the unique integer in $\{0, \dots, k-1\}$ satisfying

$$|w|_{k-r-1}^{\downarrow} > \frac{1}{r+1} \sum_{i=k-r}^{d} |w|_i^{\downarrow} \geq |w|_{k-r}^{\downarrow}. \tag{5.9}$$

One can see from Equation 5.8 that the norm trades off a squared ℓ_2 penalty for the largest components with an ℓ_1 penalty for the smallest components. Note that means that k-support norm is exactly equivalent to the ℓ_1 norm when $k = 1$, while when $k = d$ k-support norm is equivalent to the ℓ_2 norm. The k-support norm is exactly the convex hull of that hybrid norm and is closely related to the Elastic Net penalty in that it can be bounded to within a constant factor of the Elastic Net, as Argyriou *et al.* have shown in [Argyriou et al., 2012].

In this thesis we employ the k-support norm with two different loss functions, the squared loss and Huber-Hinge approximation loss. If we use the Nesterov's accelerated method, which is a first-order proximal algorithm, for optimization as suggested by [Argyriou et al., 2012], an implementation of the k-support penalty requires apart from the definition of the loss function $\mathcal{L}$, also its gradient $\frac{\partial \mathcal{L}}{\partial w}$ and the Lipschitz constant L for $\frac{\partial \mathcal{L}}{\partial w}$.

5.1.3.1 Squared Loss

The k-support norm was first introduced by Argyriou *et al.* in [Argyriou et al., 2012] using a squared loss function, which is described as follows:

$$\arg\min_{w \in \mathbb{R}^d} \lambda \|w\|_k^{sp} + \|Xw - Y\|^2. \tag{5.10}$$

The gradient of the squared loss function is defined as

$$\frac{\partial \mathcal{L}}{\partial w} = 2X^T Xw - 2X^T Y \tag{5.11}$$

and the Lipschitz constant for $\frac{\partial \mathcal{L}}{\partial w}$

$$L = 2\gamma \tag{5.12}$$

where γ is the largest eigenvalue of $X^T X$. In this case, Equation 5.10 has as special cases the LASSO when $k = 1$ and the ridge regression when $k = d$ respectively.

5.1.3.2 Hinge Loss

Since Hinge loss is not differentiable, we apply a Huber approximation to Hinge loss as in [Chapelle, 2007]. The Huber approximation of the Hinge loss is defined as follows:

$$\mathcal{L}(w, X, Y) = \sum_{i=1}^{n} \begin{cases} 0 & \text{if } y_i \langle w, x_i \rangle > 1 + h \\ \frac{(1+h-y_i\langle w,x_i\rangle)^2}{4h} & \text{if } |1 - y_i\langle w, x_i\rangle| \leq h \\ 1 - y_i\langle w, x_i\rangle & \text{if } y_i\langle w, x_i\rangle < 1 - h \end{cases} \tag{5.13}$$

where h is a parameter to choose, typically between 0.01 and 0.5 The gradient of the Huber-Hinge approximation loss is defined as :

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{i=1}^{n} \begin{cases} 0 & \text{if } y_i \langle w, x_i \rangle > 1 + h \\ \frac{\langle w,x_i\rangle x_i-(1+h)y_i x_i}{2h} & \text{if } |1 - y_i\langle w, x_i\rangle| \leq h \\ y_i x_i & \text{if } y_i\langle w, x_i\rangle < 1 - h \end{cases} \tag{5.14}$$

and the Lipschitz constant for $\frac{\partial \mathcal{L}}{\partial w}$

$$L = \frac{\gamma}{2h} \tag{5.15}$$

where γ is the largest eigenvalue of $X^T X$, as before. The use of the Huber-Hinge approximation loss with the k-support norm can be described as a close approximation to the following optimization problem:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \quad \lambda \|w\|_k^{sp} + \sum_{i=1}^{n} \xi_i \tag{5.16}$$

$$\text{s.t.} \quad y_i \left(\langle w, x_i \rangle + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i, \tag{5.17}$$

which is a modification of the classical Support Vector Machine [Cortes and Vapnik, 1995]. This learning algorithm uses the Huber approximation to the Hinge loss, which arbitrarily closely approximates the Hinge loss as $h \to 0$ while retaining differentiability. Moreover, it employs the k-support norm as a structured sparsity regularizer and we call it the *k-support regularized SVM* ($k$sup-SVM) [Gkirtzou et al., 2013c]. This approach enables the learning algorithm to select a sparse but correlated subset of discriminative variables. $k$sup-SVM has two input parameters, the $\lambda > 0$ regularization parameter and $k \in \{1, \ldots, d\}$, where d is the dimension of the feature space, the parameter that

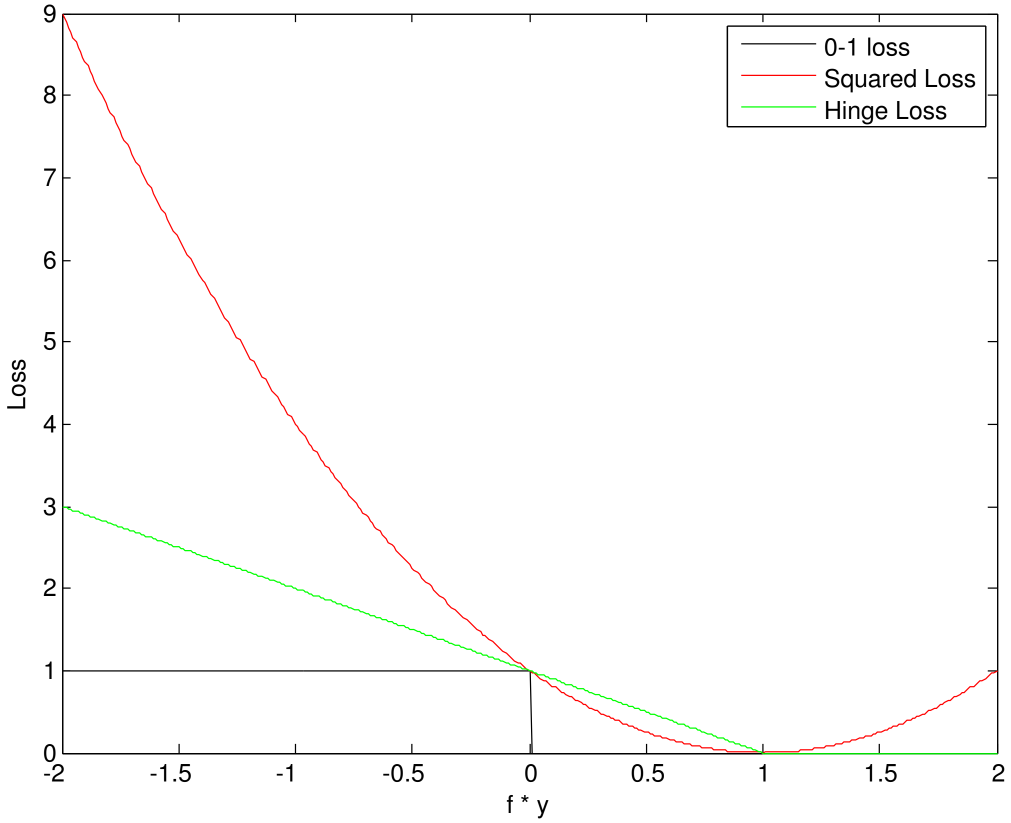


FIGURE 5.1: Loss functions for two class classification, where the response is $y = \pm 1$, the prediction function is f with class prediction $\text{sign}(f)$. The loss function are zero-one loss with black, the squared error loss with red and the hinge loss function with green.

negatively correlates with the sparsity. The limit cases are the support vector machine (SVM) [Cortes and Vapnik, 1995] when $k = d$ and the ℓ_1 regularized SVM [Zhu et al., 2004] when $k = 1$. The $k$sup-SVM can be seen as an alternative to the Elastic Net regularized SVM [Wang et al., 2006], but with a tighter convex relaxation to correlated sparsity.

Figure 5.1 shows both the squared and the hinge loss as a function of the margin $y \cdot f(x)$ for a two class classification problem in which $y \in \{-1, +1\}$. It also shows the zero-one misclassification loss, which gives unit penalty for negative margin values and no penalty for all positive ones. The classification rule used here is $\text{sign}(f(x))$, which implies that observations with positive margin $y_i \cdot f(x_i) > 0$ are classified correctly, where those with negative margin $y_i \cdot f(x_i) < 0$ are misclassified. Both the hinge loss and the squared loss can be viewed as continuous and convex upper bounds to the zero-one misclassification loss. The hinge loss penalizes increasingly negative margin values, while it gives no penalty for all positive ones. Squared loss also penalizes increasingly the negative margin values, but for margin values $y_i \cdot f(x_i) > 1$ it increases quadratically, therefore placing increasing influence on observations that are correclty classified with increasing certainty. Squared loss consequently has two undesirable properties compared with hinge loss (a) it penalizes heavily confident correct classifications and (*b*) the quadratic penalty is not as robust to label noise as the linear penalty. Note, however, that both loss functions lead to a statistically consistent binary classifier [Bartlett et al., 2006].

5.2 fMRI data analysis with the use of regularization methods

5.2.1 Introduction

In this section we evaluate the three regularization methods presented in Section 5.1, the well-known LASSO and Elastic Net and the newly introduced k-support with squared loss, in analyzing fMRI data. Functional magnetic resonance imaging (fMRI) is a non-invasive modality used for studying brain activity by detecting changes in blood oxygenation and flow, that occur as a response to neural activity. The typically small sample size, which is a result of both the high cost and time consuming nature of the fMRI acquisition procedure, and as well as the high dimensional nature of the data, which consists of tens of thousands of voxels, makes the statistical analysis of fMRI data a challenging task.

A number of approaches have been proposed in the literature for analyzing fMRI data, as we have already seen in Section 4.1.1, including machine learning techniques such as support vector machines, independent component analysis, as well as the use of regions of interest (ROIs). Due to the intrinsic high dimensional nature of fMRI data and the cost of collecting large numbers of samples, fMRI analysis is particularly suited for the sparsity regularization framework. Sparsity regularization methods, apart from being mathematical appealing, they are also fully exploratory and they do not require a predefined set of ROIs. Previous works that have explored sparsity regularization in fMRI include [Carroll et al., 2009, Ng et al., 2012b]. To the best of our knowledge, this study is the first to apply the k-support norm to fMRI analysis.

The remaining of the section is organized as follows: in Section 5.2.2 we present the data that we use in this study, in Section 5.2.3 is dedicated to the experimental setting and results and in Section 5.2.4 concludes the overall section with a discussion over the obtained results and the perspectives of this work.

5.2.2 fMRI data description

In order to evaluate the three regularization methods we use two fMRI datasets. The first one, which consists of a healthy subject in a free-viewing framework, is employed for the quantitative evaluation due to its larger sample size, while the second one, which consists of the contrast maps of 16 cocaine addicted subjects and 17 controls, is employed for qualitative evaluation.

5.2.2.1 Free-viewing dataset

The free-viewing dataset consists of fMRI data of a single session from one human volunteer [Bartels and Zeki, 2004b, Bartels et al., 2007]. The session has 350 time slices of 3-dimensional fMRI brain volumes, that was acquired using a Siemens 3T TIM scanner. The time-slices were separated by 3.2 seconds (TR) and each one has a spatial resolution of 46 slices with 64×64 pixels of 3×3mm, resulting in a spatial resolution of $3 \times 3 \times 3$mm. The subject watched one movie, which had labels indicating the continuous content of the movie (i.e. degree of visual contrast, or the degree to which a face was present, etc.). The imaging data were preprocessed using standard procedures using the Statistical Parametric Mapping (SPM5) toolbox before analysis [Friston et al., 2007]. This included slice-time correction to compensate for acquisition delays between slices, a spatial realignment to correct for small head-movements, a spatial normalization to the SPM standard brain space (near MNI), and spatial smoothing using a Gaussian filter of 6 mm full width at half maximum (FWHM). Subsequently, images were skull-and-eye stripped and the mean of each time-slice was set to the same value (global scaling). A temporal high-pass filter with a cut-off of 512 seconds was applied, as well as a low-pass filter with the temporal properties of the hemodynamic response function, in order to reduce temporal acquisition noise[Blaschko et al., 2009, Shelton, 2010].

The label time-series were obtained using two separate methods, using computer frame-by-frame analysis of the movie [Bartels et al., 2007], and using subjective ratings averaged across an independent set of five human observers [Bartels and Zeki, 2004b]. The computer-derived labels indicated luminance change over time (temporal contrast), visual motion energy (i.e. the fraction of temporal contrast that can be explained by motion in the movie). The human-derived labels indicated the intensity of subjectively experienced color, and the degree to which faces and human bodies were present in the movie. In prior studies, each of these labels had been shown to correlate with brain activity in particular and distinct sets of areas specialized to process the particular label in question [Bartels and Zeki, 2004b, Bartels et al., 2007]. The discriminative task is the prediction of a "Temporal Contrast" variable computed from the content of a movie presented to the subject [Blaschko et al., 2011].

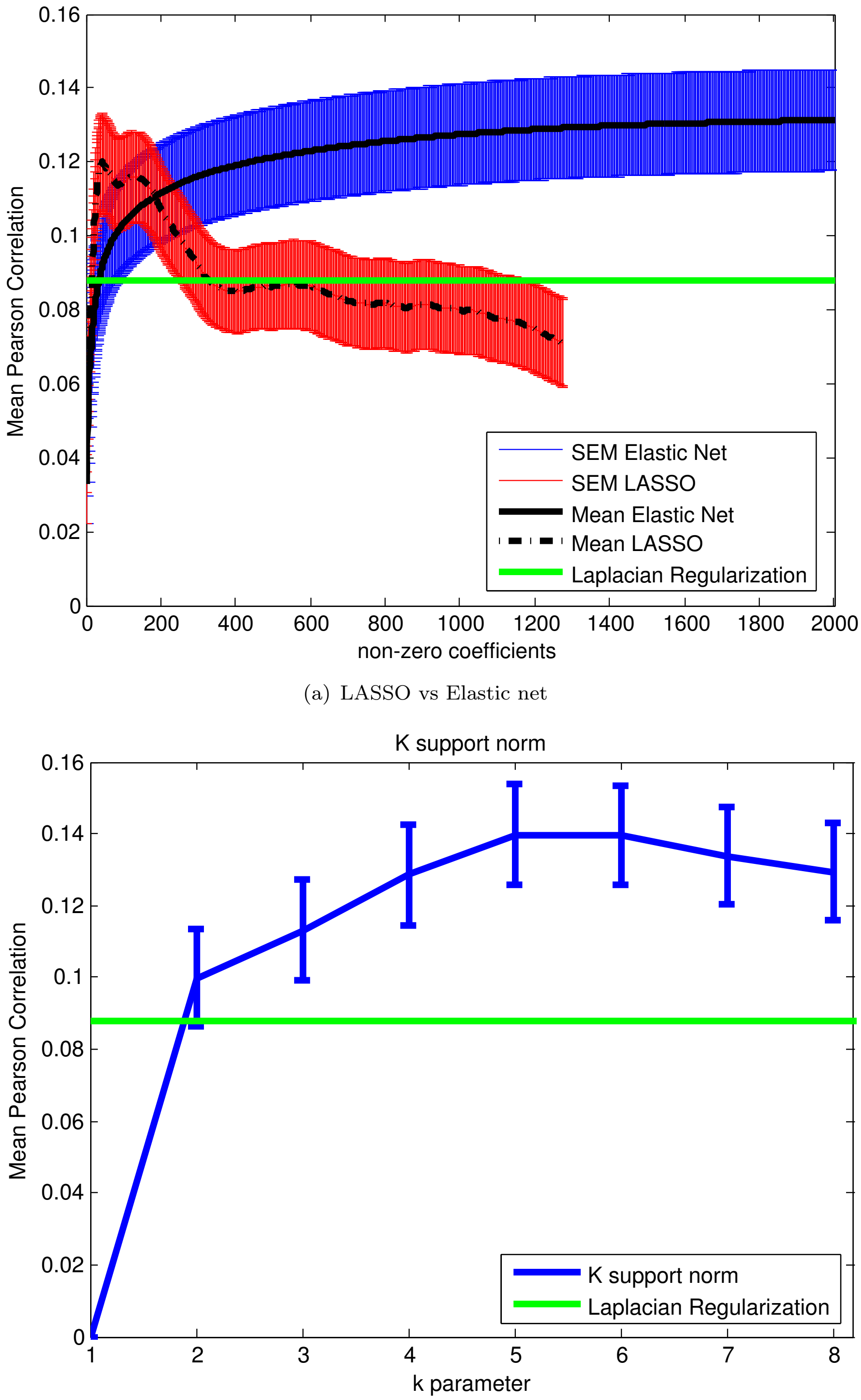
5.2.2.2 Cocaine Addiction Dataset

Our dataset [Goldstein et al., 2009] contains an approximately equal number of cocaine addicted individuals and control subjects performing a neuropsychological experiment of block design, called drug Stroop experiment. The experiment consists of an initial screen displaying the monetary reward and then presenting a sequence of forty words in four different colors. The subject was instructed to press one of four buttons matching the color of the word and was rewarded for correct performance. Each subject performs six sessions with two varying conditions, a monetary reward (50¢, 25¢ and 0¢) and the cue shown (drug-related or a neutral word). We focus here on the 50¢ condition based on previous analysis of the data [Honorio et al., 2012]. The data were preprocessed using statistical parametric mapping SPM2 [Friston et al., 1994] and a contrast map for each subject is produced. A detailed description of the preprocessing procedure is presented in Section 4.1.2. Only the subjects that complied to motion < 2mm translation, $< 2^\circ$ rotation and at least 50% performance of the subject in an unrelated task [Goldstein et al., 2009] were kept. Finally, the task of interest is the classification of a subject as cocaine addicted or control.

5.2.3 Results

The qualitive evaluatation of the different sparse regularization techniques is performed on the free viewing dataset (see Section 5.2.2.1) and their performance is shown in Figure 5.2. The performance is evaluated as the mean correlation over 100 trials of random permutations, where in each trial, 80% of the data are used to train the method, while the remaining 20% are used to evaluate the performance. We evaluate both the Elastic Net and k-support norm for same λ_2 and λ values respectively, where $\lambda, \lambda_2 \in \{10^2, 10^3, 10^4, 10^5, 10^6\}$. For the parameter selection a similar inner scheme of random permutations is used. In this inner scheme the 90% of the training data are used to train the method and rest 10% of the training to select the model's parameter over 20 trials of random permutations.

In Figure 5.2 the performance of the regularization methods is plotted as a function of sparsity. More specifically, Figure 5.2(a) shows the mean correlation between LASSO and Elastic Net against the number of non-zero variables (*i.e.* voxels) that is correlated with the λ parameter in Equation 5.2 and with the λ_1 in Equation 5.4, while Figure 5.2(b) shows the mean correlation for the k-support norm against different k values –which



(a) LASSO vs Elastic net

(b) k-support norm

FIGURE 5.2: Mean Pearson correlations between the label and prediction on the hold-out data over 100 trials for the free-viewing dataset (higher values indicate better performance). Error bars show the standard error of the mean. The LASSO achieves its best performance with a sparsity level substantially lower than the Elastic Net, as it suppresses correlated voxels (Figure 5.2(a)). The k-support norm performs better than the LASSO, Elastic Net, or Laplacian regularization reported in [Blaschko et al., 2011], and is a promising candidate for sparsity in fMRI analysis (Figure 5.2(b)). (Figure best viewed in color.)

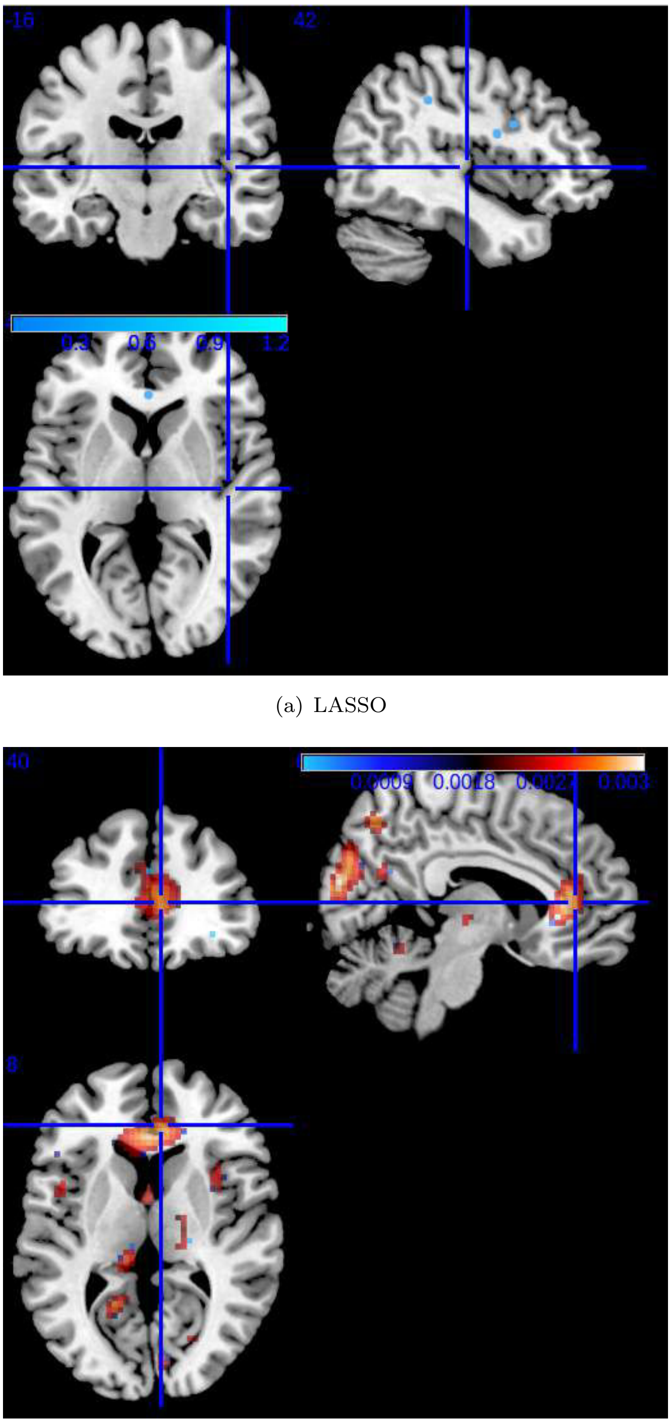
are negative correlated with the number of non-zero coefficients. LASSO achieves a maximum mean correlation of 0.12 for 45 non-zero variables, Elastic Net a maximum mean correlation of 0.1313 for 1950 non-zero variables, while k-support norm a maximum of 0.1398 for $k = 200$. This is substantially higher than was previously reported result in [Blaschko et al., 2011]. According to a Wilcoxon signed rank test, the performance of k-support norm is statistically significantly better than the performance of both LASSO and Elastic Net (p-value was $\ll 0.05$).

We have additionally visualized the brain regions predicted when applying the LASSO and the k-support norm to the cocaine addiction dataset (see Section 5.2.2.2). For each, we have selected slices through the brain that maximize the sum of the absolute values of the weights predicted by the respective methods. These results are presented in Figure 5.3 and discussed in the following section.

5.2.4 Discussion

The main area of activity shown in Figure 5.3(b) is the rostral anterior cingulate cortex (rostral ACC). It has been shown to be deactivated during the drug Stroop as compared to baseline in cocaine users versus controls even when performance, task interest and engagement are matched between the groups [Goldstein et al., 2009] and that its activity is normalized by oral methylphenidate [Goldstein et al., 2010]–which similarly to cocaine blocks the dopamine transporters increasing extracellular dopamine–an increase that was associated with lower task-related impulsivity (errors of commission). This region was responsive (showed reduction in drug cue reactivity) to pharmacotherapeutic interventions in cigarette smokers [Culbertson et al., 2011, Franklin et al., 2011], and may be a marker of treatment response in other psychopathology (e.g., depression). The LASSO, on the other hand, does not show a meaningful sparsity pattern (see Figure 5.3(a)).

In this section, we have investigated the applicability of sparsity regularizers in fMRI analysis. We have shown that the k-support norm can give better predictive performance than the LASSO and Elastic Net, while having favorable mathematical and computational properties. Furthermore, the brain regions implicated in addiction by the k-support norm coincide with previous results on addiction, indicating that the k-support norm is additionally useful for generating sparse, but correlated, regions suitable for interpretation in a medical-research setting. In the specific task of cocaine addiction, the k-support norm has implicated the involvement of the rostral ACC, in line with previous studies, while the LASSO did not lead to an interpretable result. We therefore consider the k-support norm as a promising tool for sparsity in fMRI analysis and believe it merits future study. While initial experiments have shown promising results with the



(a) LASSO

(b) k-support norm

FIGURE 5.3: A visualization of the areas of the brain selected by the LASSO and by the k-support norm applied to the cocaine addiction dataset. The LASSO leads to overly sparse solutions that do not lend themselves to easy interpretation (Figure 5.3(a)), while the k-support norm does not suppress correlated voxels, leading to interpretable and robust solutions (Figure 5.3(b)). A medical interpretation of the result presented in Figure 5.3(b) is given in Section 5.2.4. (Figure best viewed in color.)

k-support norm for a range of machine learning problems [Argyriou et al., 2012], to the best of our knowledge this study is the first to apply the approach to fMRI.

5.3 MRI based markers for neuromuscular disease categorization with k-support norm

5.3.1 Introduction

In this section, we evaluate the k-support norm regularization method with the use of two different loss functions, with the squared loss function and with the Huber-Hinge approximation loss function known as the *k-support regularized SVM* both described in detail in Section 5.1.3, over the problem of disease classification. More specifically, we are interested in discriminating between two different neuromuscular diseases, the facioscapulohumeral muscular dystrophy (FSH) and the myotonic muscular dystrophy type 1 (DM1). Myopathies are neuromuscular diseases that affect the muscles resulting in functional anomalies including fat infiltration, weakness, atrophy, paralysis, loss in muscle strength, *etc.* There are a number of procedures available for disease identification, such as electromyogram (EMG) measurement that measure the muscle weakness through its electrical activity, blood tests that can be carried out for a molecular study of the DNA for inherited myopathies with known genes and mutations and of course muscle tissue biopsy. The problem with all these methods is that are invasive or they require prior knowledge of genes and mutation. Overall the area of non-invasive diagnosis could be explored and evolved. Toward this direction, we pursue here a comparatively non-invasive approach based on MR imaging, such as T1 and T2 weighted images and with particular emphasis on Diffusion Tensor Imaging, which has been successfully used in neuroimaging for myopathy categorization.

T1 and T2 weighted images are wide used MR images that depict the structure of the human body through the difference in the spin-lattice or T1 relaxation time of various tissues within the body in the former or through the difference in the spin-spin or T2 relaxation time of various tissues in the latter. Both techniques can differentiate fat from water — with water appearing darker and fat brighter on T1 weighted images, while fat shows darker and water lighter in T2 weighed images. Finally, Diffusion Tensor Imaging is an imaging modality that captures the diffusion of water in tissues, and along with it, important structural information. It has been widely used in the study of the connectivity of the human brain [Le Bihan et al., 2001]. Nonetheless, it has also been used in different clinical scenarios. Among them, one may cite the study of the human tongue [Gilbert and Napadow, 2005], the heart muscle [Gilbert and Napadow, 2005] and the human calf muscle [Galban et al., 2004]. DTI can capture important structural information in the case of the muscle. This is due to the fact that muscles are highly organized structures that present an architecture of elongated myofibers. Because myopathies affect the muscles, one may expect that the diffusion properties in diseased subjects are also altered [Qi et al., 2008].

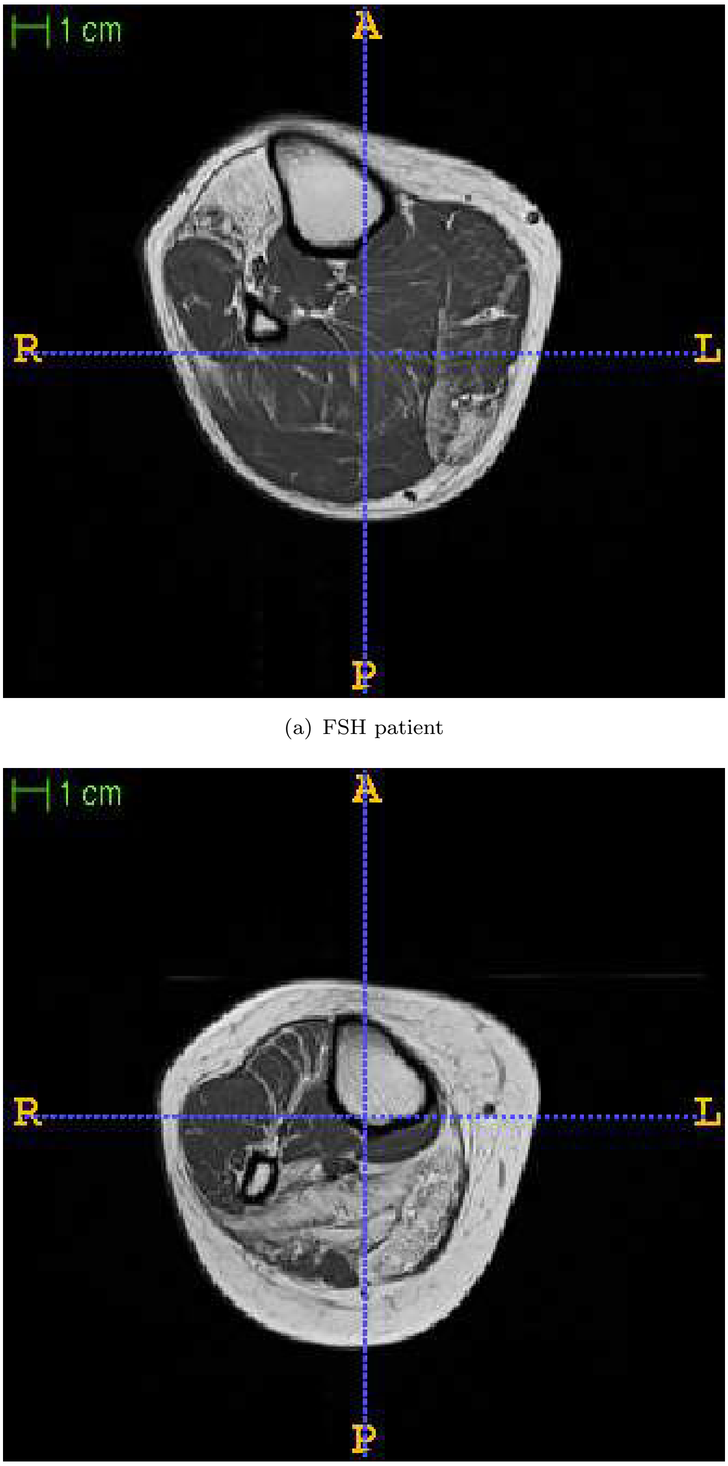
We are interested in verifying that different myopathies alter muscle in distinct ways. Moreover, we investigate the discriminative power of diffusion and structural MR features in distinguishing between diseases. Figure 5.4 shows the T1-weighted MR images of the two myopathies under investigation. The two images are very similar, making the distinction between them a very challenging task. In order to achieve this goal, we develop a strategy that exploits the rich information that is captured by both structural data (T1- and T2-weighted MR images) and DTI data to fuel state-of-the-art machine learning techniques. The use of high dimensional pattern classification in conjunction with DTI information has been previously investigated [Caan et al., 2006, Wang and Verma, 2008, Ingalhalikar et al., 2011]. Nonetheless, it has been mainly applied to distinguish patients from controls in neuroimaging studies. Discriminating between patients poses additional challenges.

The remainder of the section is organized as follows: in Section 5.3.2 we present the data that we use in this study, in Section 5.3.3 is dedicated to the experimental setting and results and in Section 5.3.4 concludes the section with a discussion over the obtained results and the perspectives of this work.

5.3.2 MRI data description

In this study, we use a dataset consisting of twenty five subjects, 10 subjects were affected by FSH and 15 subjects were affected by DM1 [Neji, 2010, Section 4.3.2]. In a clinical context, this is a large sample size. The subjects were imaged in the calf using a 1.5 T MRI scanner. The obtained volumes had a size of 64 × 64 × 20 voxels and a voxel resolution of $3.125mm \times 3.125mm \times 7mm$. T1- and T2-weighted MR images were acquired at the same time. As a consequence, the image volumes are naturally coregistered. Diffusion weighted images were acquired using the following parameters: repetition time (TR)=3600ms, echo time (TE)=70ms, slice thickness of 7mm and b value of $700s/mm^2$ with 12 gradient directions and 13 repetitions. Considering anisotropic diffusion, diffusion tensors $\mathbf{D}$ were estimated with the use of medInria software.[1]

[1] http://med.inria.fr/



(a) FSH patient

(b) DM1 patient

FIGURE 5.4: T1-weighted MR images of the calf from the two neuromuscular diseases. On the top, Figure 5.4(a), a slice of the MR image from a patient with facioscapulohumeral muscular dystrophy (FSH) and on the bottom, Figure 5.4(b), a slice of the MR image from a patient with myotonic muscular dystrophy type 1 (DM1). These diseases are not readily distinguishable by eye.

The images were segmented by an expert in the following 7 classes/muscle groups: 1) soleus (SOL), 2) lateral gastrocnemius (LG), 3) medial gastrocnemius (MG), 4) posterior tibialis (TP), 5) anterior tibialis (AT), 6) extensor digitorum longus (EDL), and 7) peroneous longus (PL). An example of the segmented muscle can be seen in Figure 4.8. It is planned to automate the segmentation process in future work. In the meantime, the approach provides a strategy to avoid an invasive biopsy. Note that a superset that also contained this dataset were also used in Section 4.2, where 3D shapes of the muscles where extracted by the T1 weighted MR images for a shape classification problem between healthy and patient subjects.

For every anatomical region, we extracted features from both the structural and the diffusion data. From the structural data, we extracted for every muscle: 1) the absolute volume, 2) the mean T1 signal, 3) the mean T2 signal, and 4) the Signal to Noise Ration (SNR). For the diffusion data, we calculated for every muscle the mean values of the following scalar measures:

1. the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the diffusion tensor $\mathbf{D}$,

2. the trace of the diffusion tensor $\mathbf{D}$

$$Tr(\mathbf{D}) = \lambda_1 + \lambda_2 + \lambda_3 \tag{5.18}$$

   which is a rotationally independent value and it indicates the average amount of diffusion,

3. the Fractional Anisotropy

$$FA = \frac{\sqrt{3}}{\sqrt{2}} \frac{\sqrt{\left(\lambda_1 - \frac{Tr(\mathbf{D})}{3}\right)^2 + \left(\lambda_2 - \frac{Tr(\mathbf{D})}{3}\right)^2 + \left(\lambda_3 - \frac{Tr(\mathbf{D})}{3}\right)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \tag{5.19}$$

   which is also rotationally invariant, as the trace, takes values between 0 and 1 and measures the anisotropy of the tensor. When FA is close to 0 then the tensor is close in shape to a sphere and there are no privileged directions of diffusion, while when FA is close to 1 the tensor is close to the shape of a "cigar", where the long axis of the ellipsoid is likely to correspond to the local fiber orientation,

4. the volume of the tensor $\mathbf{D}$

$$det(\mathbf{D}) = \lambda_1 \lambda_2 \lambda_3 \tag{5.20}$$

   which is also rotationally invariant,

5. the linear coefficient

$$C_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{5.21}$$

and

6. the planar coefficient

$$P_l = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}. \tag{5.22}$$

Both the linear coefficient and the planar coefficient lie between 0 and 1 and indicate the shape of the diffusion. For example a high value of the linear coefficient indicates a "cigar" shape diffusion, while a high value in the planar coefficient indicates a "flying saucer" diffusion shape.

These scalar measures were estimated with the use of medInria software. The resulted 84 variables were whitened and centered prior to applying the learning algorithms.

5.3.3 Results

In order to better explore the power of k-support norm regularization method with the two different loss functions, we also train a number of supervised learning methods using the same features. More specifically, we examine the k nearest neighbor algorithm [Wang et al., 2008], the support vector machine (SVM) [Cortes and Vapnik, 1995] for a number of different kernels. The k nearest neighbor method ($k$nn) is a simple, non-linear classification algorithm that classifies each new subject according to the majority of the labels of the k nearest neighbors, given a specific distance function. We examine the $k$nn algorithm with Euclidean distance and $k \in \{1, 3, 5, 7, 10\}$. The support vector machine constructs a hyperplane in a high or infinite-dimensional space by maximizing the margin of the different classes, while minimizing the classification error. For the SVM, we examine the following kernel functions, i) linear, ii) polynomial of third degree, and iii) radial basis function (RBF) with a soft-margin parameter $C \in \{10^{-3}, 10^0, 10^3\}$. For the k-support regularized squared loss and for the $k$sup-SVM we examine the following combinations of parameters $\lambda \in \{1, 10, 1000\}$ and $k \in \{1, 10, 20, 40, 80\}$.

To approximate the generalization accuracy of the classification methods using the structural and DTI tensor features, we use a random splitting scheme with 1000 trials. In each trial, a random selection of 80% of the data are used to train the methods, while the remaining 20% are used to evaluate their performance.

Figure 5.5 shows the mean ROC curve across all trials, while Table 5.1 gives the mean classification accuracy and area under the curve over 1000 trials. $k$sup-SVM outperforms the rest of the methods by achieving a mean area under the curve (AUC) of 0.7141 and mean accuracy 77% ± 0.013. The k-support norm regularized squared loss also performs well with a mean AUC of 0.694 and mean accuracy 72% ± 0.006, while $k$nn

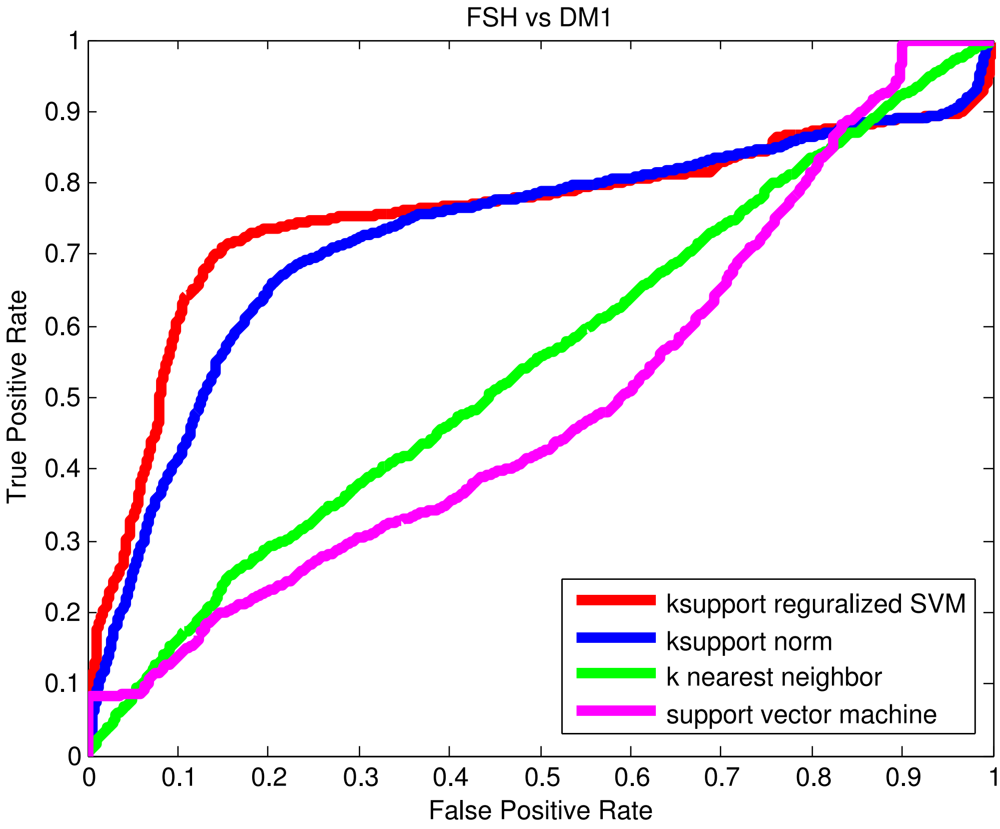


Figure 5.5: Mean ROC curves for each classifier over 1000 trials. $k$sup-SVM, shown in red, outperforms the rest of the methods k-support norm regularized squared error (blue), $k$nn (green) and SVM (magenta). To the best of our knowledge, these are the first results presenting a significant discrimination between FSH and DM1 using MRI based markers. (Figure best viewed in color.)

Table 5.1: Classification mean accuracy (in % ± standard error) and the mean area under the curve of all methods over 1000 trials. Chance is 60%.

Method	Accuracy	Area Under the Curve
$k$sup-SVM	77 ± 0.013	0.756
k-support norm	74 ± 0.006	0.726
$k$nn	61 ± 0.015	0.537
SVM	59 ± 0.015	0.494

and SVM performances are near chance, which is 60%. Moreover, with a Wilcoxon signed rank test we show that the $k$sup-SVM is statistically significantly better than all other methods (all p-values were $\ll 10^{-9}$). Figure 5.7 shows the boxplots of the weights of the structural and DTI tensor features selected by the $k$sup-SVM over 1000 bootstrap trials. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, while the whiskers extend to the most extreme data points that are not considered outliers. Since the MRI features are evaluated for each of the seven muscles of interest, we plot them per muscle. A number of features are systematically assigned zero weight across multiple trials (green line), indicating that

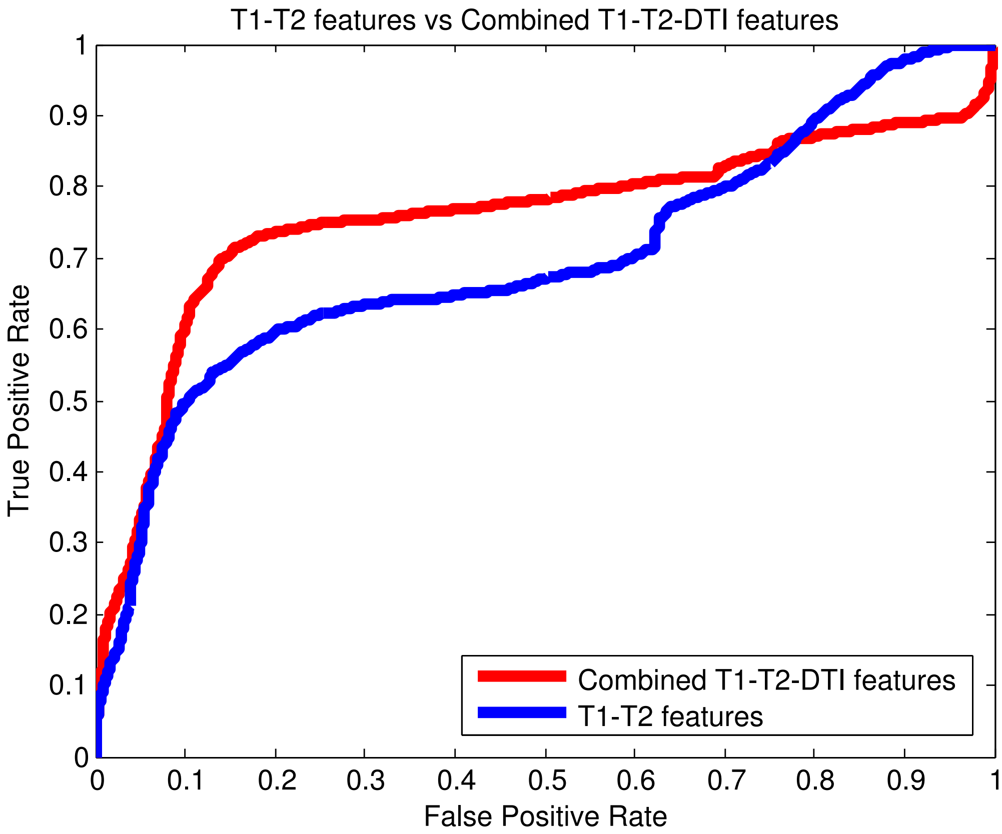


FIGURE 5.6: Mean ROC curves over 1000 trials. In red is the k-support regularized SVM using features both structured and DTI features, while in blue the k-support regularized SVM using only structured features. Using T1 and T2 weighted features, with the combination of DTI features gives the best performance in discriminating the two neuromuscular diseases (Figure best viewed in color).

TABLE 5.2: Classification mean accuracy (in %± standard error) and the mean area under the curve of the best classifier ($k$sup-SVM) using both structured and DTI features and the best classifier ($k$sup-SVM) using only structured features over 1000 trials. Chance is 60%.

Features Used	Accuracy	Area Under the Curve
Combined T1, T2 and DTI features	77 ± 0.013	0.756
T1 and T2 features	73 ± 0.006	0.697

they do not provide useful information for the distinction of the two diseases, while the ones with non-zero weight are considered more informative.

Toward demonstrating the added value of the DTI features, we compared the performance of the $k$sup-SVM when trained only on structural data against the $k$sup-SVM with both structural and DTI features, as presented in the previous paragraph. The same experimental setting as before was used. The results of the comparison are shown in Figure 5.6 and Table 5.2. In this case the $k$sup-SVM with the structural features achieved a mean AUC of 0.697 and mean accuracy 73%±0.006 a drop of 4% in accuracy. According to a Wilcoxon signed rank test, this performance is statistically significantly worse than its previous performance using both structured and DTI features (p-value was $\ll 0.05$).

5.3.4 Discussion

An analysis of variables selected by the sparsity regularizer (Figure 5.7) gives an indication that discrimination varies across muscles as well as features. Increased muscle volume in AT, EDL, MG, and TP was associated with DM1, while increased volume in PL and SOL was associated with FSH. T1 and T2 signal was consistently positively associated with FSH in the EDL muscle. A broad range of statistics were discriminative for the MG muscle, while discriminative features for most other muscles were comparatively sparse.

While MRI markers, and DTI tensor features in particular, have previously been shown to differ between disease and control subjects [Qi et al., 2008, Table 2], we are unaware of previous studies that have shown significant ability to discriminate between disease conditions. Indeed, a high-dimensional analysis of MRI based markers was required to achieve non-random performance in this more challenging task. Sparsity regularization appears to be a more important property of the learning algorithm than non-linearity, as evidenced by the comparatively stronger performance of k-support norm regularized SVM or squared error, as compared to a SVM with non-linear kernels or k nearest neighbors (both non-sparse, non-linear algorithms).

In this section, we have presented several novel methodological and clinical developments related to the use of pattern recognition methods in neuromuscular disease classification. While previous studies have focused on the comparatively easy task of separating disease from healthy subjects, we have approached the more difficult and clinically relevant task of discriminating between diseases. We have shown that a combination of T1- and T2-weighted MR images and Diffusion Tensor Imaging data are discriminative for separating patients with facioscapulohumeral muscular dystrophy and myotonic muscular dystrophy type 1. Our novel machine learning algorithm, the $k$sup-SVM, is an essential machine learning approach for achieving the best performance, with a mean accuracy of $77\% \pm 0.013$.

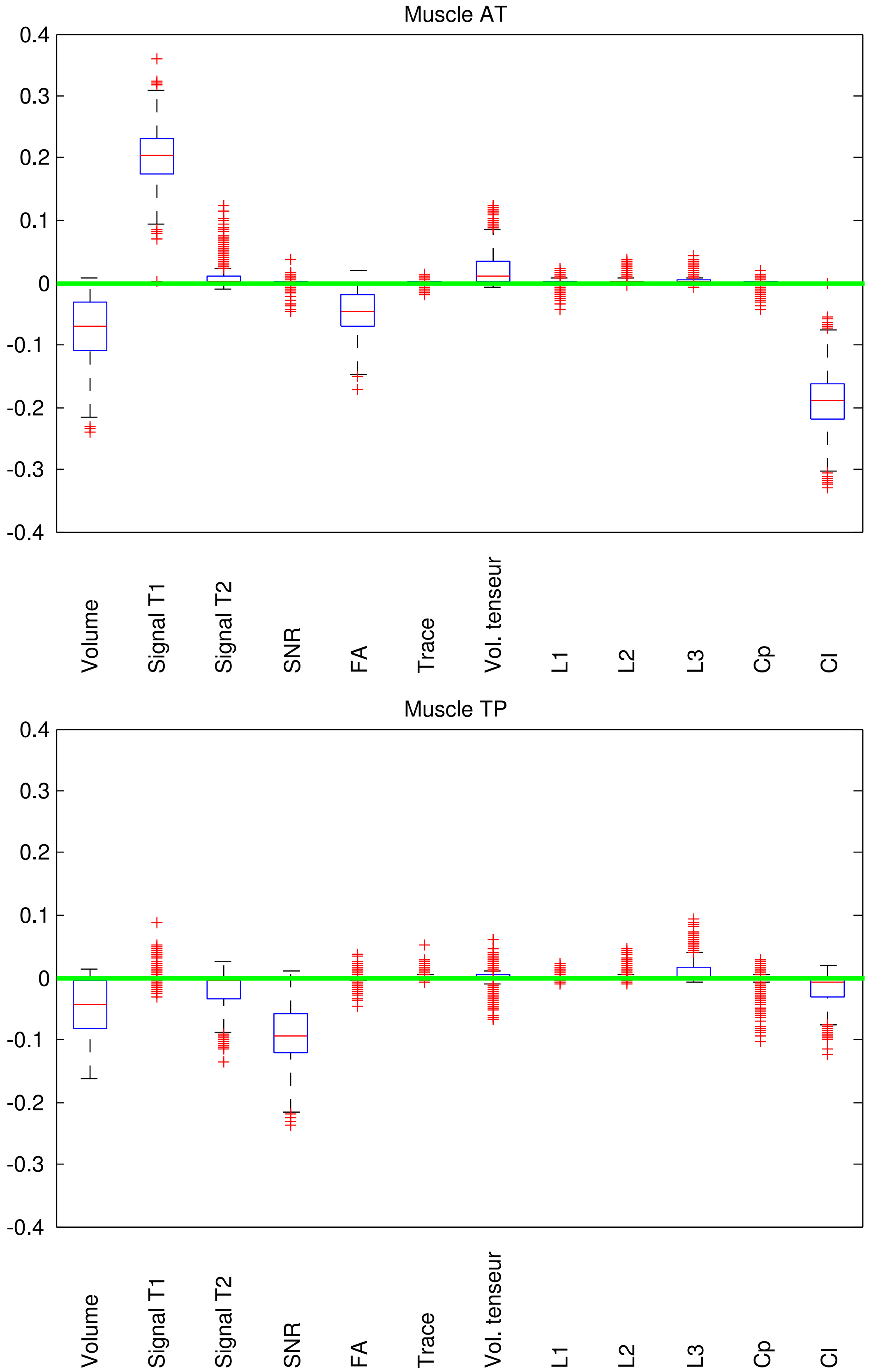


FIGURE 5.7: Boxplot of the weights given to the structural and DTI features of the 7 muscles by the $k$sup-SVM over 1000 trials. Positive values indicate positive association with FSH, while negative values indicate positive association with DM1. Values close to zero are indicative of a lack of discriminative information between the two disease conditions (continue).

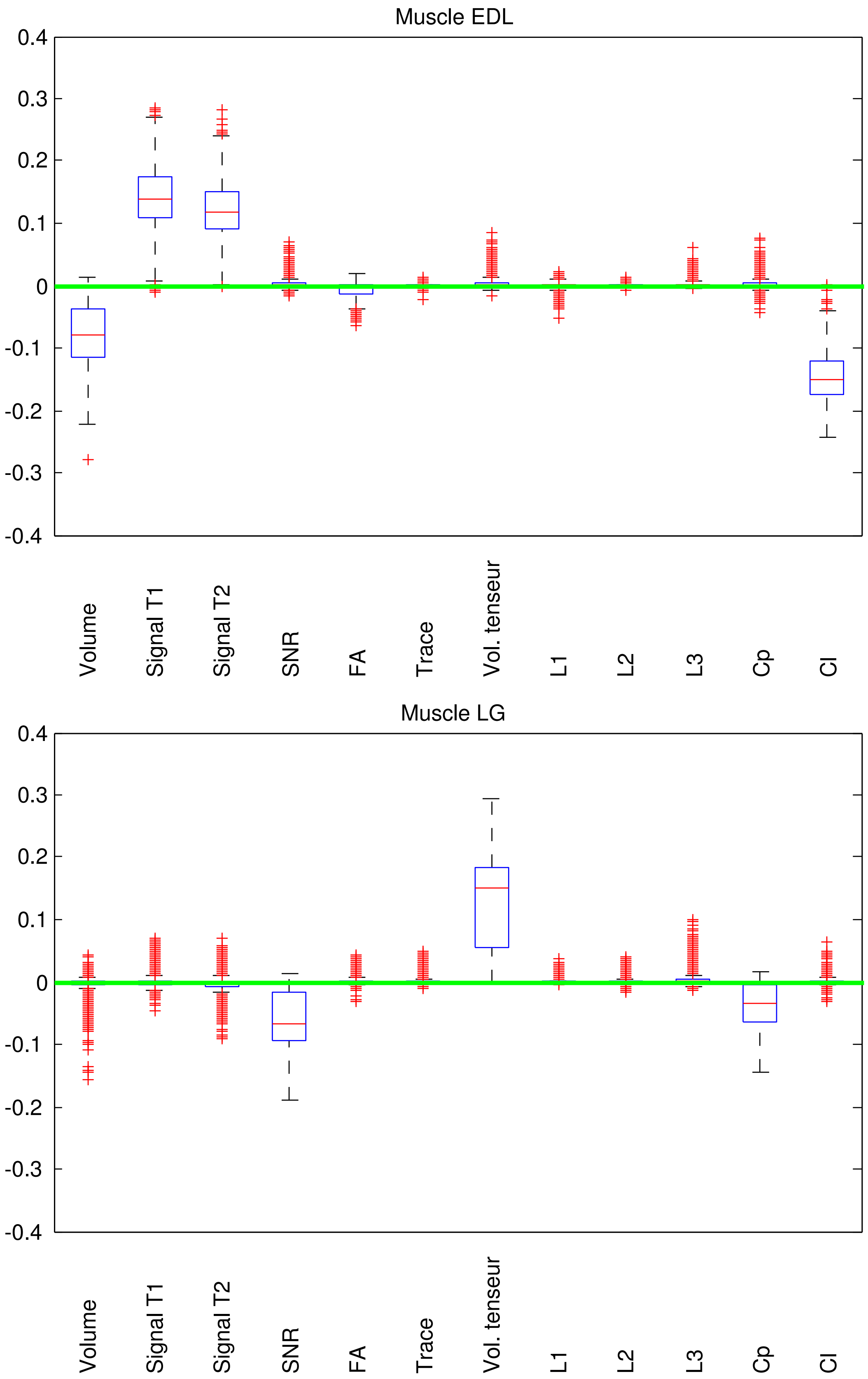


FIGURE 5.7: Boxplot of the weights given to the structural and DTI features of the 7 muscles by the $k$sup-SVM over 1000 trials (continue).

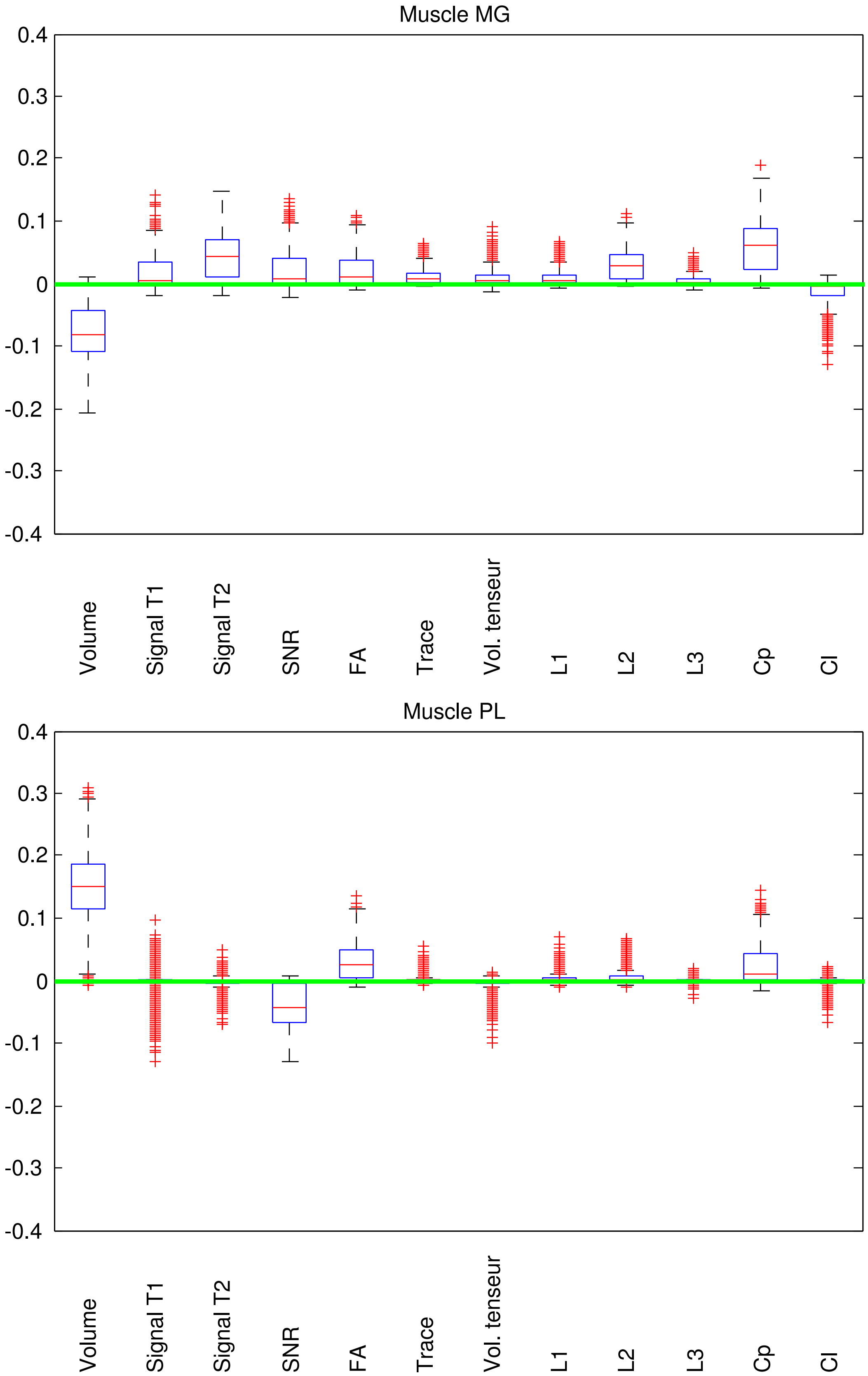


Figure 5.7: Boxplot of the weights given to the structural and DTI features of the 7 muscles by the *k*sup-SVM over 1000 trials (continue).

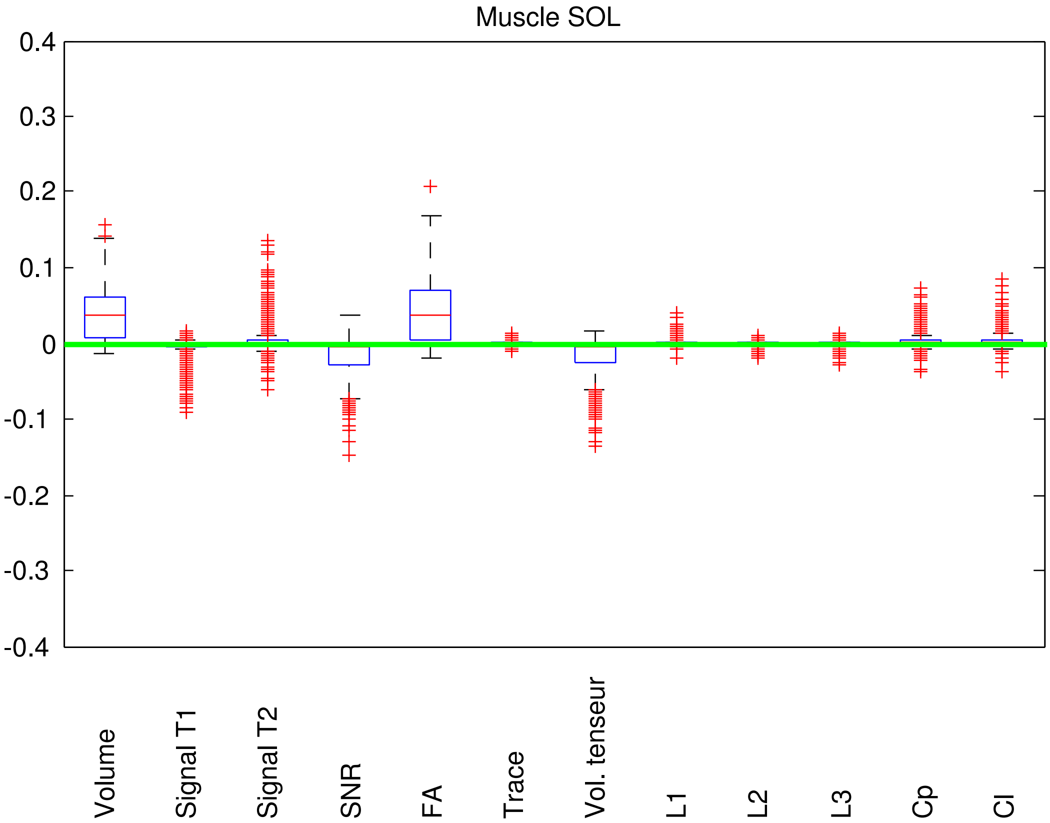


Figure 5.7: Boxplot of the weights given to the structural and DTI features of the 7 muscles by the $k$sup-SVM over 1000 trials.

Chapter 6

Conclusions

6.1 Contributions

Through this thesis, we investigated the use of novel combinations of regularizers and prediction functions for analyzing medical images. We introduced two novel learning algorithms: *a*) the *pyramid quantized Weisfeiler-Lehman graph representation* - an algorithm for comparing graphs with continuous or vector labeled nodes and/or edges - and *b*) the *k-support regularized SVM* - a sparse regularized version of the Support Vector Machine.

6.1.1 The pyramid quantized Weisfeiler-Lehman graph representation

In Chapter 3 we introduced a novel algorithm, the *pyramid quantized Weisfeiler-Lehman graph representation* that tackles the problem of graph comparison for continuous or vector labeled graphs. To the best of our knowledge this novel algorithm is the first attempt to approach the graph comparison problem with continuous or vector labels. The *pyramid quantized Weisfeiler-Lehman graph representation* considers statistics of subtree patterns based on the Weisfeiler-Lehman algorithm [Weisfeiler and Lehman, 1968], which has recently been employed in the construction of a graph kernel [Shervashidze et al., 2011] for discrete labeled graphs. The key advantage of these subtree statistics, which are tree structures constructed recursively from each node in the graph up to a predefined depth h, is that they are linear-time in the number of edges in the graph. Moreover, they make use of an efficient hashing scheme in order to only enumerate the relevant (non-zero) dimensions of an exponentially sized feature space. In addition to these computational benefits, Weisfeiler-Lehman graph kernels have been shown to perform comparably to or better than a number of more computationally complex kernels [Shervashidze et al., 2011].

In order to take advantage of this efficient scheme when extending to the problem of graph comparison with continuous or vector labels, we propose a pyramid quantization strategy inspired by [Grauman and Darrell, 2007a] to determine a logarithmic number of discrete labelings. This approximates a graph representation with continuous or vector valued labels as a sequence of graphs with increasingly granular discrete labels. For each level of quantization, we run the Weisfeiler-Lehman algorithm and get the respective subtree statistics. To construct our pyramid quantization, we recursively partition the label space as seen in Section 3.3.1. In this manner, we have a series of nested vector quantizations of increasing granularity. This can be performed efficiently in high dimensions using, *e.g.* k-d trees, for the fixed-binning scheme or with a hierarchical clustering algorithm for data-guided binning scheme. The complexity of the resulting quantization is bounded by $\mathcal{O}(d \max(|V|, k)L)$, where d is the dimension of the label space, $|V|$ is the number of vertices to be quantized, k is the maximum histogram index value in a single dimension, and L is the number of pyramid levels [Grauman and Darrell, 2007a]. With an appropriate implementation we may constrain $k \leq |V|$, and we may ensure that L is logarithmic in $|V|$, resulting in a simplified complexity of $\mathcal{O}(d|V| \log |V|)$.

We evaluate the *pyramid quantized Weisfeiler-Lehman graph representation* in two different tasks with real datasets, on a fMRI analysis task and on a 3D shape classification one in Chapter 4. In the fMRI analysis task (for more details see Section 4.1), we approach the problem by representing fMRI contrast maps from a cocaine addiction dataset as graphs with continuous labels and we use our *quantized Weisfeiler-Lehman pyramid graph representation* to compare the respective graphs. Our primary hypothesis is that the interconnections between voxels can contain additional information about the brain structure and the problem under investigation, that could not explored under a linear consideration of the data. Our proposed method validated this hypothesis and outperforms other machine learning techniques that are used in fMRI analysis with statistical significance.

In the evaluation of the *pyramid quantized Weisfeiler-Lehman* graph representation on the task of 3D shape classification we used two real datasets, one from the medical imaging and one from the semantic shape domain (for more details see Section 4.2). We view the 3D shape models as graphs, but because the topology would not explicitly encode sufficient geometric properties of the respective shapes, we enrich the graphs with continuous labels on the nodes with local features, such the local curvature of the surface. From both datasets, we conclude that the *pyramid quantized Weisfeiler-Lehman* *graph representation* performs substantially better that the pyramid Bag of Words approach, showing that the spatial structure is important for improving the classification performance. Moreover, the results show that our approach contains complementary information to the multi-viewpoint rendering technique, since the combination of both approaches performs better with statistical significance than each method individually.

Overall, as graphs are fundamental mathematical objects, the *pyramid quantized Weisfeiler-Lehman kernel* is potentially applicable to a wide range of application domains in computer vision, analysis of medical images or data mining. Finally, source code for the *pyramid quantized Weisfeiler-Lehman kernel* is publicly available at `http://gitorious.org/wlpyramid`.

6.1.2 The k-support regularized SVM

In Section 5.1.3.2 we introduced a novel regularized SVM algorithm, the *ksup-SVM*. This algorithm extends the ℓ_1 regularized SVM and the classical SVM algorithms to a mixed norm case. This enables the use of a correlated sparsity regularizer in the classification setting. We showed that using this technique we substantial improved the performance on a neuromuscular dystrophy classification task (see Section 5.3). Moreover, we showed that features extracted from DTI images provide significant information in the discrimination between neuromuscular conditions, extending the use of DTI beyond its usual application in neuroimaging. Finally, code for the new *k-support regularized SVM* is publicly available at `http://gitorious.org/ksp-svm`.

6.2 Future perspectives

Despite the big breakthrough in the graph comparison problem with the introduction of graph kernels, it still remains an open problem in the graph community. Especially the design of efficient algorithms for labeled graphs with complex labels, such as continuous, strings or even discrete but non categorical labels. In this thesis, we introduced the *pyramid quantized Weisfeiler-Lehman graph representation* for comparing graphs with continuous or vector labels, through the discretization of the labeled space and the aggregation of subtree pattern statistics calculated by the Weisfeiler-Lehman algorithm [Weisfeiler and Lehman, 1968] on the described graphs. Although we were able to compare graphs with continuous or vector labels with linear-time to the number of edges and the depth of subtree patterns, we are discarding information concerning the relation between two labels. For example, consider a graph with discrete labels corresponding to ratings on some scale. Then a label 10 and 12 resemble each other more that label 2 and 10. Many applications, such as document classification in data mining or biological network analysis, could benefit more if efficient algorithm that incorporates information concerning the similarity between labels is designed. An other perspective that derives from the previous problem when considering subtree patterns is the graph comparison through inexact matching of the subtree patterns. Solving this problem would benefit applications where graphs tend to be noisy or incomplete or the neighborhood of a node is to big for an exact match.

Bibliography

Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., 1995.

Trevor J. Hastie, Robert John Tibshirani, and Jerome H. Friedman. *The elements of statistical learning : data mining, inference, and prediction.* Springer series in statistics. New York, N.Y. Springer, 2009.

Reinhard Diestel. *Graph Theory.* Springer, 2010.

Alan Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1985.

D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. 1989.

Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 596–609, 2008.

Bernard Ng, Ghassan Hamarneh, and Rafeef Abugharbieh. Modeling brain activation in fMRI using group MRF. *IEEE Transactions on Medical Imaging (IEEE TMI)*, 31 (5):1113–1123, 2012a.

Josna Rao, Rafeef Abugharbieh, and Ghassan Hamarneh. Adaptive regularization for image segmentation using local image curvature cues. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference in Computer Vision*, volume 6314, pages 651–665. 2010.

Adrian A. Canutescu, Andrew A. Shelenkov, Roland L. Dunbrack, and Jr. A graph-theory algorithm for rapid protein side-chain prediction. *PROTEIN SCI*, 12:2001–2014, 2003.

Andreas Wagner and David A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1478): 1803–1810, September 2001.

Ira P. Goldstein. The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11(1):51 – 77, 1979.

Thomas Hapke. Chemoinformatics. a textbook. *Auskunft . - Nordhausen : Bautz*, 25 (1):171–173, 2005.

L. Kaufman and A. Neumaier. PET regularization by envelope guided conjugate gradients. *IEEE Transactions on Medical Imaging*, 15(3):385–389, 1996.

M. W. Woolrich, T. E. J. Behrens, C. F. Beckmann, and S. M. Smith. Mixture models with adaptive spatial regularization for segmentation with an application to FMRI data. *IEEE Transactions on Medical Imaging*, 24(1):1–11, 2005.

A.N. Tikhonov. On the stability of inverse problems. *Doklady Akademii nauk SSSR*, 39 (5):195–198, 1943.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320, 2005.

Katerina Gkirtzou, Jean Honorio, Dimitris Samaras, Rita Goldstein, and Matthew B. Blaschko. MRI Analysis with Sparse Weisfeiler-Lehman Graph Statistics. In *4th International Workhop on Machine Learning in Medical Imaging*, Nagoya, Japan, September 2013a.

Katerina Gkirtzou, Jean Honorio, Dimitris Samaras, Rita Goldstein, and Matthew B. Blaschko. FMRI analysis of cocaine addiction using k-support sparsity. In *International Symposium on Biomedical Imaging*, San Francisco, USA, January 2013b.

Katerina Gkirtzou, Jean-François Deux, Guillaume Bassez, Aristeidis Sotiras, Alain Rahmouni, Thibault Varacca, Nikos Paragios, and Matthew B. Blaschko. Sparse classification with MRI based markers for neuromuscular disease categorization. In *4th International Workhop on Machine Learning in Medical Imaging*, Nagoya, Japan, September 2013c.

Francis R. Bach. Graph kernels between point clouds. In *Proceedings of the 25th international conference on Machine learning*, International Conference on Machine Learning '08, pages 25–32, 2008.

Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis Applications*, 13(1):113–129, January 2010.

David Haussler. Convolution kernels on discrete structures. Technical report, 1999.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143. Springer Berlin Heidelberg, 2003.

S. Vichy N. Vishwanathan, Nicol N. Schraudolph, Risi Imre Kondor, and Karsten M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, November 2011.

P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Extensions of marginalized graph kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, pages 552–559, 2004.

Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society.

Liva Ralaivola, Sanjay Joshua Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005.

Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 158–167, 2004.

N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics*, 2009.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262, 2010.

Jan Ramon and Thomas Gaertner. Expressivity versus efficiency of graph kernels. In *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

Pierre Mahé and Jean-Philippe Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1):3–35, 2009.

Karsten M. Borgwardt. *Graph Kernels.* PhD thesis, Computer Science, Ludwig-Maximilians-University Munich, 2007.

H. L. Morgan. The Generation of a Unique Machine Description for Chemical Structures - A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5(2):107–113, May 1965.

Zaïd Harchaoui and Francis Bach. Image classification with segmentation graph kernels. In *Computer Vision and Pattern Recognition*, 2007.

Boris Weisfeiler and A.A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9): 12–16, 1968.

J.-Y. Cai, M. Furer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, SFCS '89, pages 612–617, Washington, DC, USA, 1989. IEEE Computer Society.

Zellig Harris. Distributional structure. *Word*, 10(3):146–162, 1954.

Youngjoong Ko. A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 1029–1030, 2012.

G. Qiu. Indexing chromatic and achromatic patterns for content-based. *Pattern Recognition, year = 2002, volume = 35, pages = 1675–1686.*

Li Fei-fei. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, pages 524–531, 2005.

Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, May 2007a.

K. Grauman and T. Darrell. Approximate Correspondences in High Dimensions. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007b.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, March 1963.

F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180, February 2005.

Gert R. G. Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, 2007.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machice Learning Research*, 7:1531–1565, December 2006.

Alain Rakotomamonjy, Francis R. Bach, St&#233;phane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, November 2008.

Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *International Conference in Computer Vision*, pages 606–613, 2009.

A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Doklady*, volume 4, pages 1035–1038, 1963.

A. Bartels, S. Zeki, and N.K. Logothetis. Natural vision reveals regional specialization to local motion and to contrast-invariant, global flow in the human brain. *Cereb. Cortex*, 2007.

A. Bartels and S. Zeki. Functional brain mapping during free viewing of natural scenes. *Human Brain Mapping*, 21(2):75–85, 2004a.

S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao. Comparative study of svm methods combined with voxel selection for object category classification on fmri data. *PLoS One*, 6(2):e17191, 2011.

Stephen LaConte, Stephen Strother, Vladimir Cherkassky, et al. Support vector machines for temporal classification of block design fmri data. *NeuroImage*, 26(2):317, 2005.

A. Bartels and S. Zeki. The chronoarchitecture of the human brain–natural viewing conditions reveal a time-based anatomy of the brain. *NeuroImage*, 22(1):419 – 433, 2004b.

A. Bartels and S. Zeki. Brain dynamics during natural viewing conditions–a new guide for mapping connectivity in vivo. *NeuroImage*, 24(2):339–349, 2005.

D.R. Hardoon, J. Mourão-Miranda, M. Brammer, and J. Shawe-Taylor. Unsupervised analysis of fMRI data using kernel canonical correlation. *NeuroImage*, 37(4):1250 – 1259, 2007.

M.B. Blaschko, J.A. Shelton, and A. Bartels. Augmenting feature-driven fMRI analyses: Semi-supervised learning and resting state activity. In *Advances in Neural Information Processing Systems*. 2009.

Matthew B. Blaschko, Jacquelyn A. Shelton, Andreas Bartels, Christoph H. Lampert, and Arthur Gretton. Semi-supervised kernel canonical correlation analysis with application to human fmri. *Pattern Recognition Letters*, 32(11):1572–1583, 2011.

O. Demirci, V.P. Clark, and V.D. Calhoun. A projection pursuit algorithm to classify individuals using fMRI data: Application to schizophrenia. *Neuroimage*, 39(4), 2008.

Xuerui Wang, Rebecca Hutchinson, and Tom M. Mitchell. Training fMRI classifiers to discriminate cognitive states across multiple subjects. In *Advances in Neural Information Processing Systems*, 2003.

Tom M. Mitchell, Rebecca Hutchinson, Radu S. Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

Amir M. Tahmasebi, Eric Artiges, Tobias Banaschewski, Gareth J. Barker, Ruediger Bruehl, Christian Büchel, Patricia J. Conrod, Herta Flor, Hugh Garavan, Jürgen Gallinat, Andreas Heinz, Bernd Ittermann, Eva Loth, Klara Mareckova, Jean-Luc Martinot, Jean-Baptiste Poline, Marcella Rietschel, Michael N. Smolka, Andreas Ströhle, Gunter Schumann, Tomáš Paus, and The IMAGEN Consortium. Creating probabilistic maps of the face network in the adolescent brain: A multicentre functional mri study. *Human Brain Mapping*, 33(4):938–957, 2012.

D. D. Cox and R. Savoy. fMRI 'brain reading': detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2):261–270, 2003.

Nikolaus Kriegeskorte, Kyle W. Simmons, Patrick S. Bellgowan, and Chris I. Baker. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience*, 12(5), May 2009.

M.K. Carroll, G.A. Cecchi, I. Rish, R. Garg, and A.R. Rao. Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, 44(1):112 – 122, 2009.

B. Ng, V. Siless, G. Varoquaux, J.-B. Poline, B. Thirion, and R. Abugharbieh. Connectivity-informed sparse classifiers for fMRI brain decoding. In *Pattern Recognition in Neuroimaging*, 2012b.

Kaustubh Supekar, Vinod Menon, Daniel L. Rubin, Mark A. Musen, and Michael D. Greicius. Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS Computational Biology*, 4(6), 2008.

Yong Liu, Meng Liang, Yuan Zhou, Yong He, Yihui Hao, Ming Song, Chunshui Yu, Haihong Liu, Zhening Liu, and Tianzi Jiang. Disrupted small-world networks in schizophrenia. *Brain*, 131(4), April 2008.

Damien A. Fair, Alexander L. Cohen, Jonathan D. Power, Nico U. F. Dosenbach, Jessica A. Church, Francis M. Miezin, Bradley L. Schlaggar, and Steven E. Petersen. Functional brain networks develop from a "local to distributed" organization. *PLoS Computational Biology*, 5(5), 2009.

Kaustubh Supekar, Mark Musen, and Vinod Menon. Development of large-scale functional brain networks in children. *PLoS biology*, 7(7), July 2009.

Fatemeh Mokhtari and Gholam-Ali Hossein-Zadeh. Decoding brain states using backward edge elimination and graph kernels in fMRI connectivity networks. *Journal of Neuroscience Methods*, 212(2):259–268, 2013.

R.Z. Goldstein, N. Alia-Klein, D. Tomasi, J.H. Carrillo, T. Maloney, P.A. Woicik, R. Wang, F. Telang, and N.D. Volkow. Anterior cingulate cortex hypoactivations to an emotionally salient task in cocaine addiction. *Proceedings of the National Academy of Sciences*, 106(23):9453, 2009.

J. Honorio, D. Tomasi, R. Goldstein, H.C. Leung, and D. Samaras. Can a single brain region predict a disorder? *IEEE Transactions on Medical Imaging*, 2012.

K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny, editors. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.

O. Sporns. *Networks of the Brain*. MIT Press, 2010.

Chong-Yaw Wee, Pew-Thian Yap, Wenbin Li, Kevin Denny, Jeffrey N. Browndyke, Guy G. Potter, Kathleen A. Welsh-Bohmer, Lihong Wang, and Dinggang Shen. Enriched white matter connectivity networks for accurate identification of {MCI} patients. *NeuroImage*, 54(3):1812 – 1822, 2011.

Michael Elad, Ayellet Tal, and Sigal Ar. Content based retrieval of vrml objects: an iterative and interactive approach. In *Proceedings of the sixth Eurographics workshop on Multimedia 2001*, pages 107–118, 2002.

Mona Mahmoudi and Guillermo Sapiro. Three-dimensional point cloud recognition via distributions of geometric distances, 2008.

Iasonas Kokkinos, Michael M. Bronstein, Roee Litman, and Alexander M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *Computer Vision and Pattern Recognition*, pages 159–166, 2012.

Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, July 2005.

U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, April 2008.

Ryutarou Ohbuchi and Takahiko Furuya. Distance metric learning and feature combination for shape-based 3d model retrieval. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR '10, pages 63–68, 2010.

Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 203–212, 2001.

H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of the Shape Modeling International 2003*, SMI '03, pages 130–, 2003.

Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, 1997.

Michael Garland and Paul S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of the conference on Visualization '98*, VIS '98, pages 263–269, 1998.

Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.

K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

Abi Berger. Magnetic resonance imaging. *British Medical Journal*, 324(7328):35, January 2002.

A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In *Advances in Neural Information Processing Systems*. 2012.

Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19:1155–1178, 2007.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, 16(2):589–616, 2006.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Jacquelyn A Shelton. Semi-supervised subspace learning and application to human functional magnetic brain resonance imaging data. Master's thesis, Universität Tübingen, Dept. of Computer Science, and Max Planck Institute for Biological Cybernetics, Dept. Schölkopf, 2010.

K. J. Friston, A. P. Holmes, K. J. Worsley, J. P. Poline, C. D. Frith, and R. S. J. Frackowiak. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, 2(4):189–210, 1994.

Rita Z Goldstein, Patricia A Woicik, Thomas Maloney, Dardo Tomasi, Nelly Alia-Klein, Juntian Shan, Jean Honorio, Dimitris Samaras, Ruiliang Wang, Frank Telang, Gene-Jack Wang, and Nora D Volkow. Oral methylphenidate normalizes cingulate activity in cocaine addiction during a salient cognitive task. *Proceedings of the National Academy of Sciences*, 107(38):16667–72, 2010.

Christopher S. Culbertson, Jennifer Bramen, Mark S. Cohen, Edythe D. London, Richard E. Olmstead, Joanna J. Gan, Matthew R Costello, Stephanie Shulenberger, Mark A Mandelkern, and Arthur L Brody. Effect of bupropion treatment on brain activation induced by cigarette-related cues in smokers. *Archives of General Psychiatry*, 68(5):505–515, 2011.

Teresa R. Franklin, Ze Wang, Yin Li, Jesse J. Suh, Marina Goldman, Falk W. Lohoff, Jeffrey Cruz, Rebecca Hazan, Will Jens, John A. Detre, Wade Berrettini, Charles P. O'Brien, and Anna Rose Childress. Dopamine transporter genotype modulation of neural responses to smoking cues: confirmation in a new cohort. *Addiction Biology*, 16(2):308–322, 2011.

Denis Le Bihan, Jean-François Mangin, Cyril Poupon, Chris A Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. Diffusion tensor imaging: concepts and applications. *Journal of magnetic resonance imaging*, 13(4):534–546, 2001.

Richard J Gilbert and Vitaly J Napadow. Three-dimensional muscular architecture of the human tongue determined in vivo with diffusion tensor magnetic resonance imaging. *Dysphagia*, 20(1):1–7, 2005.

Craig J Galban, Stefan Maderwald, Kai Uffmann, Armin de Greiff, and Mark E Ladd. Diffusive sensitivity to muscle architecture: a magnetic resonance diffusion tensor imaging study of the human calf. *European journal of applied physiology*, 93(3):253–262, 2004.

Jing Qi, Nancy J Olsen, Ronald R Price, Jason A Winston, and Jane H Park. Diffusion-weighted imaging of inflammatory myopathies: Polymyositis and dermatomyositis. *Journal of Magnetic Resonance Imaging*, 27(1):212–217, 2008.

MWA Caan, KA Vermeer, LJ Van Vliet, CBLM Majoie, BD Peters, GJ den Heeten, and FM Vos. Shaving diffusion tensor images in discriminant analysis: A study into schizophrenia. *Medical Image Analysis*, 10(6):841–849, 2006.

Peng Wang and Ragini Verma. On classifying disease-induced patterns in the brain using diffusion tensor images. *Medical Image Computing and Computer Assisted Intervention*, pages 908–916, 2008.

Madhura Ingalhalikar, Drew Parker, Luke Bloy, Timothy PL Roberts, and Ragini Verma. Diffusion based abnormality markers of pathology: Toward learned diagnostic prediction of asd. *Neuroimage*, 57(3):918–927, 2011.

Radhouène Neji. *Diffusion Tensor Imaging of the Human Skeletal Muscle : Contributions and Applications.* PhD thesis, Ecole Centrale Paris, March 2010.

Peng Wang, Ruben Gur, and Ragini Verma. A novel framework for identifying dti-based brain patterns of schizophrenia. *International Society for Magnetic Resonance in Medicine*, pages 3–9, 2008.