Action Recognition by Graph Embedding and Temporal Classifiers

Ehsan Zare Borzeshi

Faculty of Engineering and Information Technology

University of Technology, Sydney

A dissertation submitted for the degree of

Doctor of Philosophy

May 2014

Certificate of Authorship and Originality

Title: **Action Recognition by Graph Embedding and Temporal Classifiers**
Author: **Ehsan Zare Borzeshi**
Date: **May 1, 2014**
Degree: **PhD**

I certify that the work in this dissertation has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the dissertation has been written by me. Any help that I have received in my research work and the preparation of the dissertation itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the dissertation.

Signiture of author

I would like to dedicate this dissertation to my beloved wife

Shima

for her unending love and support.

Acknowledgements

Many people have contributed in various ways to make this PhD study an exciting and memorable journey. Just to name a few:

I would like to acknowledge the lively and stimulating atmosphere in our group. The lunches, discussions at the coffee shops and evenings out: I have enjoyed all of them. Several persons deserve a special mention. I gratefully acknowledge all of my current and former group members: Dr. Oscar Perez Concha, Ava Bargi, Shaukat Abidi, Jaime Andres Garcia, Dr. Richard Yi Da Xu; my friends: Prof Federico Girosi and Dr. Majid Nazem; and members of the iNext research centre at UTS. Furthermore, I owe sincere gratitude to all members of the Center for Research in Computer Vision at UCF: Afshin Dehghan, Shayan Modiri and Amir Roshan Zamir; and Dr. Kasper Riesen at University of Applied Sciences and Arts Northwestern Switzerland.

I would like to express my utmost gratitude to my external advisors, Professor Mubarak Shah and Professor Horst Bunke for all their invaluable help and advice during my studies.

I would like to acknowledge ARC and my colleagues in the "Airports of the Future" project. This work has been supported by the Australian Research Council (ARC) under Linkage Projects Scheme "LP0990135".

I would like to thank my dissertation reviewers for finding the time to read and comment on this work.

I would like to thank my parents, Mohammad and Marzieh, for giving birth to me at the first place, teaching me that it is no disgrace to work hard (which has proven to be a useful lesson) and supporting me spiritually throughout my life. I also thank them for their faith in me and allowing

me to be as ambitious as I wanted. It was under their watchful eye that I have gained so much drive and ability to tackle challenges head on. Furthermore, I thank my sister and brother, Elaheh and Amirhossein, for their love and support. I am so lucky to have all of you as my family.

I would also like to thank my wife's parents, Hossein and Tahereh, for first letting me take her hand in marriage, and for their extreme kindness and invaluable support to date. They are my best friends and beloved family and their interest and support has been tremendously valuable and appreciated indeed.

A huge thanks to my supervisor Professor Massimo Piccardi for supporting me during my PhD, for being patient and for being critical about my work. I admire his brilliance, determination, and hunger for knowledge. I have very much appreciated his pleasant way of supervising, even though I have never mentioned this explicitly. He has been an invaluable source of insights that any PhD student would love to know but which have never been written down. Massimo has supported me academically and emotionally through the rough road to finish this thesis. I feel deeply indebted to Massimo and a huge portion of the knowledge and confidence that I have today is due to him.

This list would be incomplete without expressing the most wholehearted gratitude to my life angel and love, Shima. Her support has been unconditional all these years; she has given up many things for me to finish my study; she has cherished with me every great moment and supported me whenever I needed it. Her positive spirit, unwavering love, patient endurance and tolerance of my occasional vulgar mood throughout the last ten years deserve so much more than just a "thank you". Not only now, but also in the years to come.

Abstract

With the improved accessibility to an exploding amount of video data and growing demand in a wide range of video analysis applications, video-based action recognition becomes an increasingly important task in computer vision. Unlike most approaches in the literature which rely on bag-of-feature methods that typically ignore the structural information in the data, in this monograph we incorporate the spatial relationship and the time stamps in the data in the recognition and classification processes.

We capture the spatial relationships in the subject performing the action by representing the actor's shape in each frame with a graph. This graph is then transformed into a vector of real numbers by means of prototype-based graph embedding. Finally, the temporal structure between these vectors is captured by means of sequential classifiers. The experimental results on a well-known action dataset (KTH) show that, although the proposed method does not achieve accuracy comparable to that of the best existing approaches, these embedded graphs are capable of describing the deformable human shape and its evolution over time.

We later propose an extended hidden Markov model, called the hidden Markov model for multiple, irregular observations (HMM-MIO), capable of fusing spatial information provided by graph embedding and the textural information of STIP descriptors. Experimental results show that recognition accuracy can be significantly improved by combining the spatio-temporal features with the structural information obtaining higher accuracy than from either separately. Furthermore, HMM-MIO is applied to the task of joint action segmentation and classification over a concatenated version of the KTH action dataset and the challenging CMU multi-modal activity dataset. The achieved accuracies proved comparable to or higher

than state-of-the-art approaches and show the usefulness of the proposed model also for this task.

The next and most remarkable contribution of this dissertation is the creation of a novel framework for selecting a set of prototypes from a labelled graph set taking class discrimination into account. Experimental results show that such a discriminative prototype selection framework can achieve superior results, not only for the task of human action recognition, but also in the classification of various structured data such as letters, digits, drawings, fingerprints compared to other well-established prototype selection approaches.

Lastly, we change our focus from the forementioned problems to the recognition of complex event, which is a recent area of computer vision expanding the traditional boundaries of visual recognition. For this task, we have employed the notion of concept as an alternative intermediate representation with the aim of improving event recognition. We model an event by a hidden conditional random field and we learn its parameters by a latent structural SVM approach. Experimental results over video clips from the challenging TRECVID MED 2011 and MED 2012 datasets show that the proposed approach achieves a significant improvement in average precision at a parity of features and concepts.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Visual recognition of human actions in video clips has been an active field of research for several years and there exists a vast body of literature on the subject. The fundamental approach is to extract features from video that can be informative about the structure in the data. However, most published methods rely on bag-of-words approaches that typically ignore the spatial and temporal structure in the data. Conversely, the potential benefit of incorporating the structural information in the recognition process has been raised recently [45, 96]. As such, in this monograph we will discuss and propose different approaches to involve the spatial relationship and the time stamps in the data in the recognition process.

For capturing the spatial structure, we first represent the actor's shape in each frame by a graph. The main reason for this choice is that graphs possess a strong representational power for structured objects and as such they might be promising for human action recognition. In order to use conventional statistical classifiers, we then embed each graph into a finite set of distances from prototype graphs. The prototype graphs are an intermediate representation which spans across the different spatial structure in the training set. To capture the temporal structure, we make use of sequential classifiers such as hidden Markov models and hidden conditional random fields.

The main results of this research unit are a) an extended hidden Markov model allowing to jointly leverage the embedded graphs and local spatio-temporal features for

action recognition and b) novel discriminative approaches for the selection of the prototype graphs. In particular, the proposed discriminative prototype selection approaches have permitted a significant improvement in classification accuracy compared to the state-of-the-art methods. As such, we have decided to dedicate a second research unit to expand these methods and explore how they would perform on other types of structured data such as letters, digits, molecules, fingerprints and many others. Results from this second unit of work show that discriminative prototype selection is a very general and effective approach, and likely the single most remarkable outcome of my PhD.

Finally, in the last part of this dissertation we decided to address recognition of more challenging spatio-temporal patterns such as *complex events*. Complex events are entities of higher-level semantics than single actions, often involving multiple actors and objects, and including occurrences as articulated as "a cruise ship departing from port", "a wedding", "a birthday party" et cetera. Here, we have employed the notion of "concept" as an alternative intermediate representation with the aim of improving event recognition. We model an event by a hidden conditional random field and we learn its parameters by a latent structural SVM approach. This last part of my thesis was developed while I was an intern at the Center for Research in Computer Vision (CRCV) at the University of Central Florida in the US under co-guidance from Professor Mubarak Shah and in collaboration with his group.

1.2 Research Questions

The main research questions addressed in this monograph are:

1. Can graph embedding by prototype selection prove a suitable approach to represent shape in human action recognition?

2. Can graph embedding by prototype selection also prove an effective approach for other types of structured data such as as letters, digits, drawings, molecules, fingerprint types and others?

3. Can other intermediate representations such as concepts entail useful temporal structure for recognition of more complicated spatio-temporal patterns such as complex events?

1.3 Outline of the Dissertation

The dissertation is organised as follows:

Chapter 2 provides a brief overview of the fundamental concepts required for a thorough understanding of this monograph. The chapter mainly addresses graphs and graph embedding, graphical models and sequential classifiers, and the structural support vector machine.

Chapter 3 presents a study which uses graphs to represent the actor's shape and graph embedding to then convert the graph into a suitable feature vector. Experiments on the popular KTH dataset [124] show that the embedded graphs are capable of describing the deformable human shape and its evolution over time.

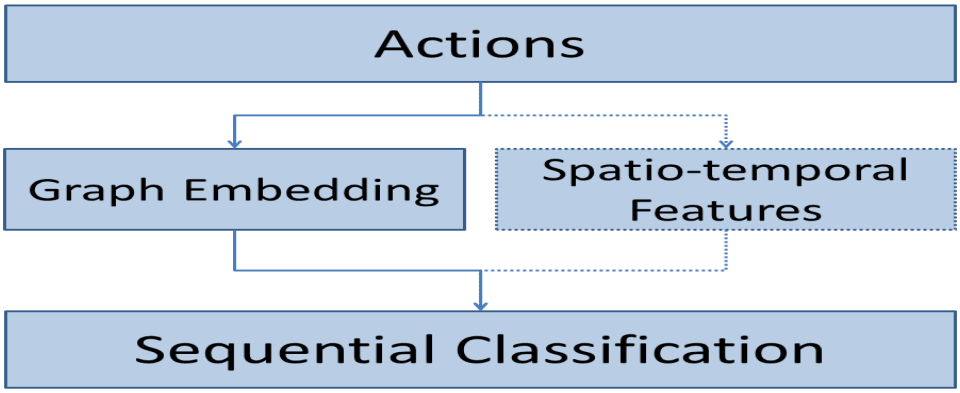
Chapter 4 proposes a way of fusing the information of both graphs and local spatio-temporal descriptors leveraging the strong representational power of both types of descriptors. We first present a joint action segmentation and classification approach based on an extended hidden Markov model, named hidden Markov model for multiple, irregular observations (HMM-MIO), and then employ this model for the fusion of structural information provided by graph embedding and appearance descriptors centred around spatio-temporal interest points (STIPs) [70]. Figure 1.1(a) shows a sketch of this approach.

Chapter 5 introduces a novel framework for selecting a set of prototypes for graph embedding from a labelled graph set. This framework exploits the notion of discriminative selection by using objective functions that simultaneously take into account within- and between-class properties. Experimental results over a variety of structured data show that such a framework can achieve superior results in classification compared to other well-established prototype selection approaches and is very general. Figure 1.1(b) shows a sketch of this approach.

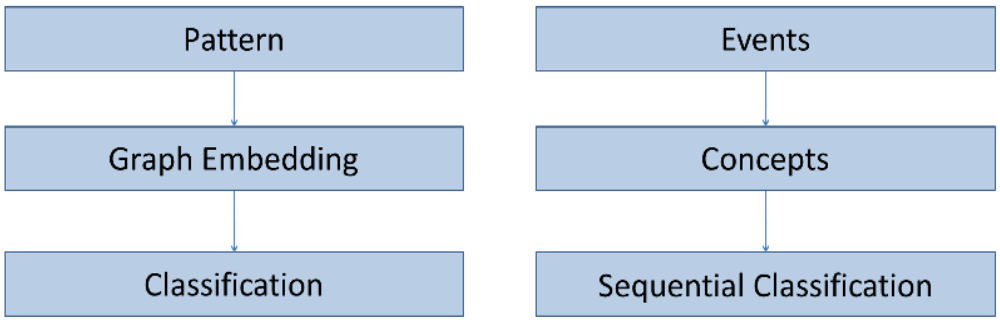
Chapter 6 addresses recognition of complex events exploiting the scores of concept detectors as measurements in a temporal model. This model (known as hidden conditional random field) leverages a latent state chain that jointly decodes the concept scores and provides event recognition. For training, we have employed a maximum-margin approach given its strong reputation for experimental accuracy [138]. Results over the very challenging TRECVID Multimedia Event Detection 2011 and 2012 datasets show the accuracy of the proposed approach. This unit of work was carried

out during my internship at the Center for Research in Computer Vision (CRCV) at the University of Central Florida in the US under the co-supervision of Professor Mubarak Shah. Figure 1.1(c) provides a sketch of this approach.

We conclude the dissertation in Chapter 7.



(a) Chapters 3 and 4



(b) Chapter 5

(c) Chapter 6

Figure 1.1: A visual sketch of the approaches presented in the various chapters.

1.4 Publications

The research in this thesis has resulted in the following publications:

Chapter 3:

- EHSAN ZARE BORZESHI, RICHARD Y. D. XU, AND MASSIMO PICCARDI. *Automatic human action recognition in videos by graph embedding*. In 2011 International Conference on Image Analysis and Processing (ICIAP), pages 19-28, 2011.
- EHSAN ZARE BORZESHI, MASSIMO PICCARDI, AND RICHARD Y. D. XU. *A discriminative prototype selection approach for graph embedding in human action recognition*. In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pages 1295-1301, 2011.

Chapter 4:

- EHSAN ZARE BORZESHI, OSCAR P. CONCHA, AND MASSIMO PICCARDI. *Human action recognition in video by fusion of structural and spatio-temporal features*. In Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR), 7626, pages 474-482. 2012.
- EHSAN ZARE BORZESHI, OSCAR P. CONCHA, RICHARD Y. D. XU, AND MASSIMO PICCARDI. *Joint action segmentation and classification by an extended hidden Markov model*. Signal Processing Letters (SPL), IEEE, 20[12]:1207-1210, 2013.

Chapter 5:

- EHSAN ZARE BORZESHI, MASSIMO PICCARDI, KASPAR RIESEN, AND HORST BUNKE. *Discriminative prototype selection methods for graph embedding*. Pattern Recognition (PR), 46[6]:1648-1657, 2013.

Chapter 6:

- EHSAN ZARE BORZESHI, AFSHIN DEHGHAN, MASSIMO PICCARDI, MUBARAK SHAH. *Complex event recognition by latent temporal models of concepts*. Submitted to International Conference on Image Processing (ICIP), IEEE, 2014.

Chapter 2

Literature Review

2.1 Action Recognition

Human action recognition is an important but still largely unsolved problem in computer vision with many potential useful applications, including video surveillance, human-computer interaction, content-based video retrieval, multimedia, and others. Recognising human actions is challenging since actions are complex patterns which take place over the time. Due to the nature of human physiology and the varying environmental constraints, different people may perform the same action in pronouncedly different ways in both spatial extent and temporal progression. In addition to this intrinsic, high intra-class variance, low inter-class distance in terms of subject and scene appearance, motion, viewing positions and angles, as well as action duration pose great challenges.

The main steps of a generic action recognition system are:

- extracting a set of informative measurements (feature set) from the image sequence depicting the action;
- learning statistical models from the extracted measurements and using those models to detect and classify new actions;
- possibly, segmenting streams of motions into single action instances that are consistent with the set of pre-defined actions (action segmentation).

Human action recognition is a part of a broader research area, *human motion analysis* from images and videos. For the abstract level of movement recognition, different taxonomies have been proposed in the literature [3, 89, 103, 139]. We adopt the hierarchy proposed by Moeslund *et al.* in [89] that has also been exploited by Poppe [103]. In this taxonomy, the human motion is categorised in three levels:

1. *Action primitive*: human movement at the limb level; e.g. "left hand up".
2. *Action*: a combination of action primitives at whole body level; e.g. "walking".
3. *Activity*: an interpretable sequence of actions; e.g. "jumping hurdles" which consists of starting, jumping and running actions.

It is important to know that recognising human actions requires processing the full body movement, unlike other motion analysis such as face or gesture recognition that relates to only one body part.

2.1.1 Challenges

Due to the nature of human physiology every individual performs each action in a variable manner over different instances, both in space and time. As such, it creates a main problem of high intrinsic intra-class variability. This issue worsens when increasing the number of action classes and decreasing inter class distances, as more overlap will be likely to occur.

Furthermore, the various environments in which the action is performed and its recording settings cause totally different visual appearances of the individual performing the action. Moreover, the same action, observed from different viewpoints, can easily lead to very different image observations.

Adding to the challenge, the number of samples available for training and validation is limited compared to the earlier mentioned variations, preventing a "brute force" training approach. Finally, ground-truth labeling of action videos is a challenging task and universal agreement over ground-truth labels is still a controversial issue.

2.1.2 Feature Extraction

The first step in any recognition system is the process of extracting representative information from data. This process is called *feature extraction* and it defines a set of features, or data characteristics, in order to most efficiently or meaningfully represent the information that is important for analysis and classification. It is obvious that an ideal feature set for human action recognition should be action discriminative, and theoretically, not too sensitive to small variations in human appearance, background, viewpoints and action performance.

One of the main challenges in action recognition is how to consider the temporal information of action execution, and to that end, any recognition system has to choose a feature set that either includes or excludes the temporal information. In general, the feature set for action recognition can be categorised into two main groups [103]:

Global features: These features encode the region of interest (ROI) in a holistic manner and are often applied jointly with background subtraction or tracking. These representations are powerful since they represent much of the information. However, they strongly rely on accurate localization, background subtraction or tracking which is hard to have in realistic conditions. Furthermore, they are more sensitive to viewpoint, noise and occlusions. Some of these features exploited for action recognition are silhouette-based [14, 88, 149], contour [49, 75], projection histograms [33, 53, 58, 143], optical flow [4, 5, 42], and space-time volumes [13, 159, 161].

Local features: Unlike global features, local features represent the ROI as a collection of independent patches. They do not require background subtraction or tracking and are also less sensitive to noise and partial occlusions. In order to extract such features, spatio-temporal interest points are detected first, and then local patches are calculated around these points. The final features are made of combinations of these patches. Some examples of local features employed for action recognition include the space-time interest points (STIP) [70], space-time cuboids [71], histogram of oriented gradients (HOG) and histogram of oriented flow (HOF) [72, 73], the extension of HOG to 3D [67], scale invariant feature transform (SIFT) [83] and its extension to 3D [125], speeded-up robust features (SURF) [7] and their extension to 3D [155].

2.1.3 Action Detection and Classification

Armed with the extracted feature set from the ROI, the human action recognition problem becomes an action detection or an action classification task. If a is a specific action and A is the total number of different classes of actions, in the action detection we would like to just detect single action classes ($a \in \{0,1\}^A$) while in the action classification we assign an instance to one of the existing action classes ($a \in \{0 \dots A\}$). In other words, the action detection task can be described as a binary action classification: one specific class versus anything else. Various approaches have been proposed and followed for these tasks in the literature and we can categorise them into three main groups [103]:

Direct classification: These approaches, e.g. [36, 72, 73], employ either a discriminative classifier, e.g. the support vector machine (SVM) [124], or a k-Nearest Neighbor (kNN) classifier to directly classify the extracted feature set. With these approaches, one needs to extract a feature set which is able to capture the spatial and also temporal nature of the action of interest and is of a fixed size for any video. While these approaches, e.g. [36, 72], have proved capable of high classification accuracies on challenging action datasets such as KTH [124] and HOHA [72], they do not seem to pay adequate attention to the temporal duration of human actions which is known to stretch in a non-linear, local way over different samples of the same action class.

Temporal graphical models: These models consist of a set of states representing various stages during action execution connected by edges where each edge represents probabilities between states, and between states and observations. Temporal state-space models are either generative or discriminative. In the former approach the target is training a model for a certain action class through maximising the likelihood of all the training data for that class. The hidden Markov models (HMMs) [106] is the main generative approach for temporal graphical models. Contrary to the generative approaches, the discriminative methods do not train one model per action. Instead, the target is to discriminate between various action classes by using all the samples of various actions. So, they focus on differences between classes and try to maximise the conditional likelihood of all the samples. The main representative of the discriminative approach is the conditional random fields (CRFs) [133] (more details in section 2.3.3).

2.1.4 Joint Segmentation and Classification

In the last two sections, we have explained approaches that extract visual features from video streams and combine them in space and in time for making a decision on what actions are present in the video. In most cases, those approaches are demonstrated with results obtained using segmented video clips, each showing a single action from start to finish, both for training and testing. Another interesting problem for human action recognition is to jointly tackle the problems of action segmentation and classification. The methods for this task can be classified into three categories: boundary detection, sliding windows and grammar concatenation [153]. The first category uses generic techniques to first just detect the action boundaries and then classify each interval separately [87, 108, 145]. In the second category, the motion sequence is divided into multiple overlapping regions by means of a sliding window. The classification task can then be easily applied to each window [65, 165]. The last category uses genuine graphical models capable of modeling the actions as well as the transitions between them simultaneously [15, 90, 102, 127]. The last category makes neither the assumptions of the boundary detection, nor do they require heavy evaluations such as the sliding window. The segmentation is elegantly and efficiently solved using dynamic programming techniques.

2.2 Graph Theory

One of the main instruments used in this thesis is the adoption of graphs for representing the human posture as a holistic feature. Therefore, this section briefly conveys the key terminology and some concepts of graph theory used in this thesis.

2.2.1 Graph

Different definitions for a graph can be found in the literature based on the considered applications. The following provides a versatile definition of graph g which is sufficiently flexible for a large variety of tasks. A graph g is defined as a four-tuple $g = (V, E, \alpha, \beta)$, where

- V is the finite set of vertices (or nodes),

- $E \subseteq (V \times V)$ is the set of edges,
- $\alpha : V \rightarrow L_V$ is the vertex labeling function, and
- $\beta : E \rightarrow L_E$ is the edge labeling function.

L_V and L_E are finite or infinite label sets of vertices and edges, respectively.

The labeling functions (α and β) in this definition are unconstrained, thus they can easily handle arbitrarily structured graphs. For instance, the vertices and edges of the graph g can get labels from the set of integers $L = \{1, 2, \ldots\}$, the vector space $L = \mathbb{R}^n$, or a set of symbolic labels $L = \{\rho, o, \kappa, \ldots\}$. Given that the vertices and/or the edges are labelled, the graphs are referred to as *attributed graphs*. Similarly, *non-attributed graphs* (or *data graphs*) are obtained by assigning the same label ϵ to all vertices and edges. Moreover, a graph is *directed* if E is a set of ordered pairs of vertices and *undirected* if E is composed of unordered vertex pairs.

2.2.2 Graph Matching

With a graph-based object representation, the concept of similarity in pattern recognition turns into that of graph (dis)similarity. Evaluating the (dis)similarity of a pair of graphs is commonly referred to as *graph matching* (for an extensive review of graph matching techniques and application, the reader is referred to [29]). Based on the definition proposed by [29], "Graph matching is the process of finding a correspondence between the nodes and the edges of two graphs that satisfies some (more or less stringent) constraints ensuring that similar substructures in one graph are mapped to similar substructures in the other".

The two main types of graph matching are: *exact* and *in-exact* graph matching. Exact graph matching is only applicable to a very small range of real-world problems. In other words, the requirement that a significant number of node and edge labels in two graphs must be identical for a positive match is not realistic in application on graph extracted from real world data [29]. In-exact, or error-tolerant, graph matching methods offer a wider range of models for structural matching. One of the most widely used methods for error-tolerant graph matching is the *graph edit distance* (GED). Actually, GED is an important way to find graph dissimilarity between two graphs, in an error-tolerant manner [17].

2.2.2.1 Graph Edit Distance

The graph edit distance is recognized as one of the most flexible and universal matching methods. It measures dissimilarity (similarity) of arbitrarily structured and arbitrarily labelled graphs and it is flexible thanks to its ability to cope with any kind of structural errors [29, 50]. The main idea of graph edit distance is to find the dissimilarity (similarity) of two graphs by the minimum amount of distortion that is required to transform one graph into the other [50]. In the first step, the underlying distortion models (or *edit operations*) are defined as an insertion, a deletion and a substitution operation for both nodes and edges. Based on the definition of graph edit distance, every graph can be transformed to the other graph by applying a sequence of edit operations (called *edit path*). For example, a valid and obvious edit path can always be constructed by first removing all nodes and edges from the first graph, and then inserting all nodes and edges of the second graph. Thus, every pair of graphs has many edit paths and the one that best represents the matching of two graphs is used to define their similarity. In order to find the best edit path, an edit cost function is introduced. The idea is to assign a cost to each edit operation, reflecting the strength of the associated distortion. For example, changing the label of an edge from 0.7 to 0.1 should usually have a higher cost than changing the label from 0.6 to 0.65.

Regarding the above discussion, a sequence of edit operations $(e_1, \ldots, e_K)$ that transforms g_1 into g_2 is called an edit path from g_1 to g_2. Figure 2.1 shows an example of an edit path between g_1 and g_2 consisting, in step order, of three edge deletions, one node deletion, one node insertion, two edge insertions, and two node substitutions.

Figure 2.1: An example edit path between g_1 and g_2 (node labels are represented by different shades of gray). Image courtesy of Kaspar Riesen [111].

Based on the above definition, every graph can be transformed into another graph by applying a sequence of edit operations or an edit path. Clearly, for every pair of graphs, there exits an infinite number of different edit paths transforming one graph into the other. Thus, to select the best edit path between each pair of graphs, an edit cost function is introduced to assign a cost to each edit operation. Then, given a set of

edit paths and an edit cost function, the dissimilarity of a pair of graphs is defined as the minimum-cost edit path in the set.

Let $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ be a pair of graphs in a set. The graph edit distance of such graphs is defined as:

$$d(g_1, g_2) = \min_{(e_1,\ldots,e_K)\in E(g_1,g_2)} \sum_{k=1}^{K} C(e_k), \tag{2.1}$$

where $E(g_1, g_2)$ denotes the set of edit paths between the two graphs, C denotes the edit cost function and e_k denotes the individual edit operation.

Based on equation 2.1, given an edit cost function (which can be assigned heuristically or learned from a set of sample graphs [93, 94]), the dissimilarity between each pair of arbitrary structured and arbitrarily labelled graphs (e.g. directed, undirected, node and/or edge labelled from any finite or infinite domain, unlabelled) can be measured by means of the graph edit distance. Furthermore, a certain degree of robustness against various graph distortions can be expected.

2.2.2.2 Probabilistic Graph Edit Distance

Among various methods to define a graph edit distance [17, 95, 114, 130], in this thesis we have used the *probabilistic graph edit distance* (P-GED) approach proposed in [92, 94] to automatically find the cost function from a labelled sample set of graphs. First, the P-GED assumes that any graph can be transformed into any other graph by iteratively applying six basic edit operations (node and edge insertion, deletion and substitution) [50]. Then, a probability, or probability density function, is defined for each type of edit operation over its label or labels. For instance, for edit operation "node substitution", between a node from g_1 with label l_1 and a node from g_2 with label l_2, we have $p_{ns}(l_1, l_2)$ as the probability for node substitution. Given a probability distribution on edit operations, the probability of an edit path $e = (e_1, \ldots, e_k)$ is defined as

$$p(e_1, \ldots, e_k) = \prod_{k=1}^{K} p(e_k), \tag{2.2}$$

assuming that the edit operations are independent one another. The probability of two graphs, g_1 and g_2, is defined as

$$\widehat{p}(g_1, g_2) = \max_{(e_1,\ldots,e_K)\in E(g_1,g_2)} p(e_1, \ldots, e_k). \tag{2.3}$$

where $E(g_1, g_2)$ is the set of all edit paths between g_1 and g_2. In practice, it is not possible to create this entire set and we resort to random generation to generate a subset. If we assume that the structural similarity of two graphs can be expressed by their probability $p(g_1, g_2)$, we obtain a dissimilarity measure on graphs by using:

$$d(g_1, g_2) = -\log p(g_1, g_2). \tag{2.4}$$

With this assumption, the problem of learning the graph edit distance can therefore be understood as learning the probability distribution $p(g_1, g_2)$, which is given by the sum of the probabilities of any paths leading from g_1 to g_2. As a final objective is to allocate low distances to graphs from the same class and high distance to graphs from different classes, the authors introduced the edit operations in such a way to increase intra-class probabilities and decrease inter-class probabilities in a controlled way [94]. Overall, the authors introduced a model for the distribution of each edit operation, and then trained the model to reach high intra-class probabilities and finally derived edit costs from the model. The main advantage of this model is that it is able to cope with large samples of graphs with huge distortion between samples of the same class [50, 94].

2.2.2.3 Bipartite Graph Edit Distance

Another approach to compute the graph edit distance is based on the fast *bipartite* optimization procedure mapping local substructures of one graph to local substructures of another graph [114]. In the bipartite graph edit distance:

- a version of Munkres' algorithm [91] is used to find a minimum cost assignment of the nodes of g_1 to the nodes of g_2 in polynomial time;
- in the assignment, the costs of edges operations are also taken into account, but only as lower bounds;

- after the assignment, the costs of the implied, actual edge operations are added.

This algorithm returns only an approximate distance since the edge costs are not evaluated sequentially. As such, this method is much less computationally demanding than other approaches which are based on combinatorial search procedures (e.g. [17]).

2.3 Graphical model based learning

This section provides a brief summary of standard machine learning algorithms as required for the rest of this thesis. Further background, as well as more detailed accounts, can be found in [11, 62, 97, 146].

2.3.1 Inference and Learning

Graphical models are a marriage between probability theory and graph theory. They provide a natural tool to compactly represent complex, real-world phenomena. These models have enjoyed a surge of interest in the last two decades, due to both the flexibility and power of the representation. The most fundamental tasks in graphical models (and yet highly non-trivial) are their *inference* and *learning*. Here we present a short recap on these fundamental tasks [62].

Inference: Given a discrete random variable y representing a class (state, index etc) and x a measurement, statistical inference is given by conditional probability $p(y|x)$. "Inference" is often assimilated with *decision* i.e. choosing the best value for y based on $p(y|x)$ and some decision rule. The main approaches to make this decision are:

- *Maximum a posteriori* (MAP) approach which tries to make as few misclassifications as possible (zero-one loss):

$$y^* = \underset{y}{\operatorname{argmax}} \;\; p(y|x) \tag{2.5}$$

- *Minimum risk* approach which tries to minimizes the expected loss i.e. errors

weighted by their cost $\Delta(y', y)$:

$$y^* = \underset{y}{\operatorname{argmin}} \left[\sum_{y'} \Delta(y', y) p(y'|x) \right] \tag{2.6}$$

Based on the above definition, we can break the classification stage into two separate stages, the inference stage in which we use training data to learn conditional probability $p(y|x)$, and the subsequent decision stage in which we use these posterior probabilities to make optimal class assignments [11].

A *generative classifier* is a classifier where $p(y|x)$ is factorised as $p(y|x) \propto p(x|y)p(y)$. Inference for generative and discriminative classifiers is therefore

$$\begin{aligned} Generative &: y^* = \underset{y}{\operatorname{argmax}} \ (p(x|y)p(y)) \\ Discriminative &: y^* = \underset{y}{\operatorname{argmax}} \ (p(y|x)) \end{aligned} \tag{2.7}$$

with the zero-one loss.

An alternative would be to solve both inference and decision together and simply learn a function that maps input x directly into decisions. Such a function is called a *discriminant* function [11].

$$Discriminant : y^* = f(x) \tag{2.8}$$

It is obvious that if a loss function different from the zero-one loss needs to be used, the decision rule for 2.7 has to be as in the minimum risk. For 2.8, the loss function has to be suitably accounted for in $f(x)$.

Learning: Learning algorithms build on the inference algorithms and allow the model to be estimated from data. If the model consists of parameterization θ of a given function, it is called *parameter estimation*. Please note that names such as learning, training, parameter estimation are often used interchangeably in the literature. The main approaches to select the best model parameters θ^* are [11]:

- *Maximum a posteriori estimation* (MAPE) approach which chooses the best pa-

rameters θ^* such that:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \;\; p(\theta|X) \tag{2.9}$$

- *Maximum likelihood estimation* (MLE) approach uses *Bayes' theorem* to express $p(\theta|X) \propto p(X|\theta)p(\theta)$ and also assumes $p(\theta)$ is uniform, so it can choose θ^* as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \;\; p(X|\theta) \tag{2.10}$$

where X is a set of N different measurements ($X = \{x_i\}, i = 1 \ldots N$) [11].

We can also categorize the learning approaches based on the provided training set. If we have a labelled training set of pairs $(Y, X) = \{y_i, x_i\}$, where y_i is the class or output in general and x_i is the measurement, we can perform *supervised* learning (what we had noted as X, the data, here becomes (Y, X)). The other learning approach is *unsupervised* learning where we only have the measurements without any label, $X = \{x_i\}$ (the labels can be considered as hidden variables).

Figure 2.2: A simple directed graphical model

For example, figure 2.2 shows a simple directed graphical model with the joint probability of $p(x, y) = p(x|y)p(y)$ where y is a label and x is the measurement. In unsupervised learning, the label is unknown (e.g. the component in the mixture distribution) and the best model parameters can be learned by MLE (equation 2.10) over the measurement alone. In supervised scenario (e.g. Bayesian classifier), as we have the labels for the measurements during training, we can find the best model parameters by

using MLE as expressed in equation 2.11 [11].

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \;\; p(X, Y|\theta) \tag{2.11}$$

2.3.2 Directed and Undirected Graphical Models

The two main types of graphical models are:

Directed graphical models, also known as *Bayesian networks*, specify the family $p(y)$ with y a set of random variables by means of a directed acyclic graph [1] $G = (V, E)$ and the factorization of $p(y)$ as

$$p(y) = \prod_{i \in V} p(y_i|y_{pa_G(i)}) \tag{2.12}$$

where each $p(y_i|y_{pa_G(i)})$ is a conditional probability distribution, and $pa_G(i)$ denotes the set of parents of node $i \in V$.

Undirected graphical models, also known as Markov random fields (MRF), define a family of joint probability distribution by mean of an undirected graph $G = (V, E)$ as factorization

$$p(y) = \frac{1}{Z} \prod_{c \in C(G)} \psi_c(y_c) \tag{2.13}$$

where $C(G)$ denotes the set of all cliques [2] of G. By y_c we denote the sub-set of variable that are indexed by c. The normalizing constant Z is given by

$$Z = \sum_{y \in Y} \prod_{c \in C(G)} \psi_c(y_c) \tag{2.14}$$

and is known as partition function. The functions $\psi_c \to \mathbf{R}_+$ are the so called *potential functions* or *factors*. Each factor ψ_c defines an interaction between one or more variables but in contrast to Bayesian networks it is not a conditional probability but an arbitrary non-negative function.

[1] A directed acyclic graph (DAG), is a directed graph with no directed cycles.

[2] Given $G = (V, E)$, a sub-set $W \subseteq V$ of the vertices is a clique if for any $i, j \in W$ we have $\{i, j\} \subseteq E$, that is there exist an edge for any pair of vertices in W.

2.3.3 Probabilistic Graphical Models for Sequential Data

If data from each class are assumed drawn from a single generating distribution, independently of each other (i.i.d assumption), we can express the likelihood function $p(X|\theta)$ as the product over all data points of probability distribution evaluated at each data point (equation 2.15).

$$p(X|\theta) = p(x_1, \ldots, x_N) = \prod_{n=1}^{N} p(x_n|\theta) \tag{2.15}$$

For many applications, however, this assumption will be a poor one. Here we introduce other types of datasets where we should deal with *sequential* data instead of i.i.d data. With these data, considering the measurement's order in the sequence during inference and learning is crucial. These types of measurements often arise through measurement of time series: for instance, the image features at successive time frames used for action recognition. Sequential measurements can also arise in other contexts such as space (e.g. nucleotide pairs along a strand of deoxyribonucleic acid (DNA)). Here we briefly introduce the main generative and discriminative models which can deal with sequential data and their hidden variables. In both models, Hidden Markov model (HMM) and Hidden Conditional Random Field (HCRF), we have a sequence of observations (measurements, or emissions), $X = \{x_1, \ldots, x_t, \ldots, x_T\}$, where T is its length, and a corresponding sequence of hidden states (or classes), $Y = \{y_1, \ldots, y_t, \ldots, y_T\}$; and each sample x_t may be generated out of a different distribution. Each state of an HMM/HCRF can take value in a discrete set with N symbols $\{s_1, \ldots, s_N\}$, while the observation can have either discrete or continuous values. Please note that HMMs and HCRFs are the natural extension of *Markov Random Fields* (MRFs) [66] and *Conditional Random Fields* (CRFs) [133] while they are augmented with latent states.

The Hidden Markov Model (HMM) is the main generative approach for sequential data. It is a temporal graphical model in which the modeled system has observed outputs and a set of hidden states (figure 2.3).

The HMMs have two fundamental hypotheses:

1. Markov state transitions: the value of state at time t, y_t, only depends on the

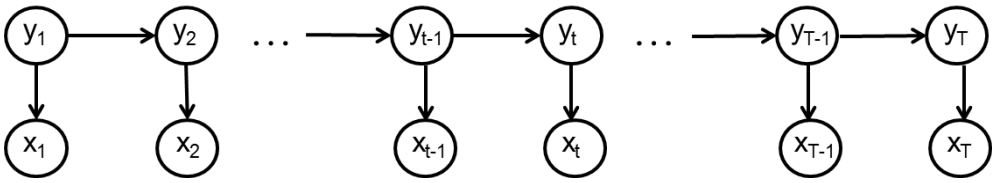


Figure 2.3: Graphical model for the HMM

value of state at previous time, y_{t-1}, and is independent of the other previous variables ($p(y_t|y_{t-1}, x_{t-1}, \ldots, y_1, x_1) = p(y_t|y_{t-1})$).

2. Independence of each observation given its state: the value of observation at time t, x_t, only depends on the value of state at time t, y_t, that means the independence of each observation given its state ($p(x_t|y_T, x_T, \ldots, y_1, x_1) = p(x_t|y_t)$).

An HMM is represented by a set of parameters, $\theta = \{A, B, \pi\}$. A is the $N \times N$ state transition probability matrix, π is a vector of N initial state probabilities and B represents the parameters of the observation probabilities for each state. There are three "canonical" problems for an HMM, each of them with an exact solution:

1. **Evaluation:** given X and θ, measure $p(X|\theta)$. The solution of this problem is the *forward-backward* algorithm [41, 106].

2. **Decoding:** given X and θ, find the best sequence of states, Y which explains X. The *Viterbi* algorithm can solve this problem [41, 106].

3. **Unsupervised estimation:** given X, find θ that maximises $p(X|\theta)$. The *Baum-Welch* re-estimation algorithm [10, 106] is exploited to solve this density estimation problem.

4. **Supervised estimation:** given X and Y, find θ that maximises $p(Y, X|\theta)$.

A much cited tutorial on the HMM can be found in [106].

Classification with HMM is the case in which the entire observation sequence corresponds to a single, pre-segmented action. As such, each action class is in correspondence with one HMM. The learning of the HMM parameters for each class is

achieved by the Baum-Welch re-estimation algorithm [106] and classification of an unseen observation sequence, X_{new}, is obtained by maximum-likelihood (ML) classification. In other words, let us denote as $A = \{a_1, \ldots, a_k, \ldots, a_K\}$ the set of K different action classes; $\theta = \{\theta_1, \ldots, \theta_k, \ldots, \theta_K\}$, the set of HMM parameters associated with each action class in A; $\mathbb{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_k, \ldots, \mathcal{X}_K\}$, the set of K different groups of observation sequences, one per class; and, eventually, each $\mathcal{X}_k = \{X_k^1, \ldots, X_k^{N_k}\}$ as the group of N_k observation sequences for action class k. Then, parameters θ_k^*, $k = 1 \ldots K$, are estimated with maximum likelihood as:

$$\theta_k^* = \arg\max_{\theta_k}\left(\prod_{e=1}^{N_k} p(X_k^e|\theta_k)\right). \tag{2.16}$$

After training of the θ parameters, the action class, a_k^*, for an unseen sequence, X_{new}, can be chosen by maximum likelihood as:

$$a_k^* : k^* = \arg\max_k(p(X_{new}|\theta_k)), \quad k = 1 \ldots K. \tag{2.17}$$

where the likelihood of X_{new} in action class k, $p(X_{new}|\theta_k)$, can be efficiently evaluated by the forward or backward algorithm [106].

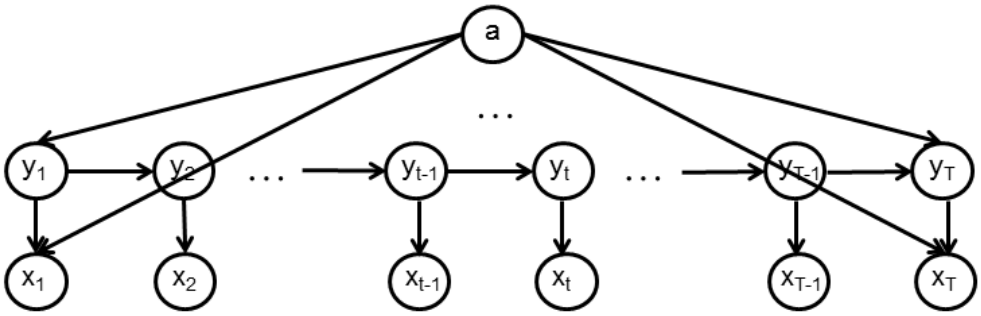


Figure 2.4: Graphical model for Classification with HMM

The Linear-chain Conditional Random Field (CRF) is a discriminative undirected probabilistic graphical model [69, 133]. The conditional model of CRF is $p(y_{1:T}|x_{1:T})$ and it is represented in figure 2.5. The learning of a CRF model is supervised (class labels are known during learning); and the inference and decoding can

be carried out using standard graphical model algorithms equivalent to those for the HMM.

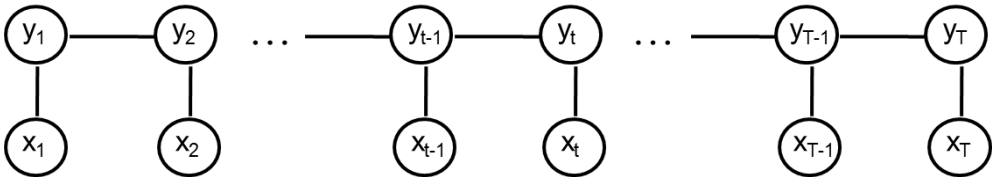


Figure 2.5: Graphical model for the Linear-chain CRF

The Hidden Conditional Random Field (HCRF) is the main representative of discriminative models which can deal with hidden states. An HCRF uses intermediate latent variables to model the hidden structure of the input data. It defines a joint distribution over the class label and latent state labels conditioned on the observations, with dependencies between the latent states; the hidden states and the observations are expressed by an undirected graph (figure 2.6). The conditional model is represented in equation 2.18. The inference and learning for an HCRF can be carried out using standard graphical model algorithms. More details can be found in [105, 133].

$$p(a|x_{1:T}) \propto \sum_{y_{1:T}} p(a, y_{1:T}|x_{1:T}) \tag{2.18}$$

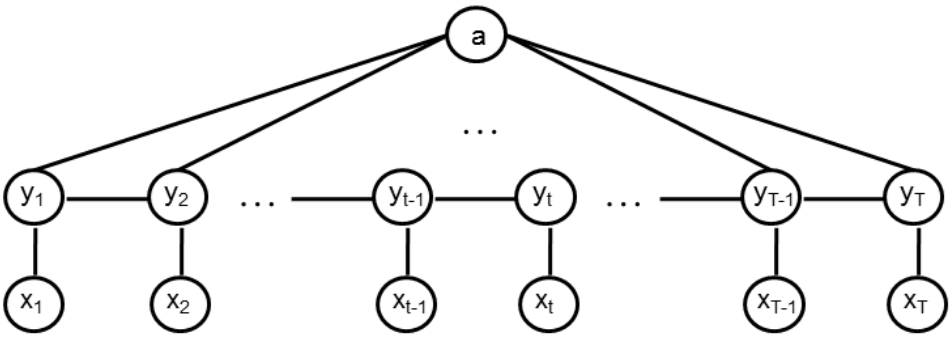


Figure 2.6: Graphical model for the HCRF

2.4 Support Vector Machine

Support Vector Machines (SVMs) are among the best (and believed by many to be the best) "off-the-shelf" supervised learning algorithms. These machines were developed by Cortes and Vapnik [31] for the sake of binary classification. The main properties of the basic SVMs are:

- **Class separation:** basically, we are looking for the optimal separating hyperplane between the two classes by maximizing the *margin* between the closest points of the classes. The points lying on the boundaries are called *support vectors*, and the middle of the margin is our optimal separating hyperplane (figure 2.7);
- **Overlapping classes:** data points on the "wrong" side of the discriminant margin are weighted down to reduce their influence ("*soft margin*", figures 2.8, 2.9);
- **Nonlinearity:** when we cannot find a *linear* separator, data points are projected into a (usually) higher-dimensional space where data points effectively become linearly separable (this projection is realised via *kernel* techniques);
- **Problem/Solution:** the whole task can be formulated as a convex optimization problem, and so any local solution is also a global optimum.

If we express the two class labels as $y = \{+1, -1\}$ and also assume the arbitrary scale of W and b to be such that the *implicit equation* for the separating hyperplane is $W^T X + b = 0$ and the closest points (or support vectors) lie on $W^T X + b = 1$ and $W^T X + b = +1$, the objective function of SVM can be expressed as:

$$\begin{aligned} W^*, b^* &= \underset{W,b}{\operatorname{argmin}} \ \frac{1}{2}\|W\|^2 \\ s.t. \quad & y_i(W^T x_i + b) \geq +1 \quad \forall x_i \end{aligned} \tag{2.19}$$

This objective function is a quadratic subject to linear inequality constraints. The inference of the class (*aka* prediction, classification) for a new point, x_{new}, is given by:

$$y^* = \underset{y}{\operatorname{argmax}} \ \ y(W^T x_{new} + b). \tag{2.20}$$

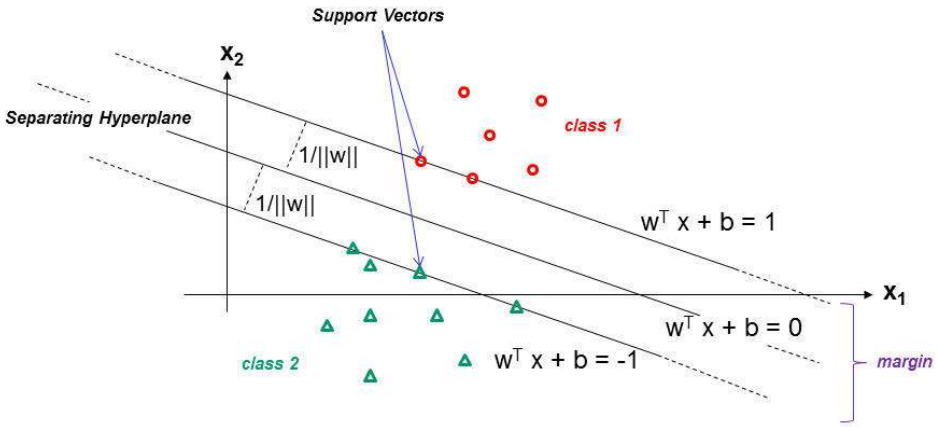


Figure 2.7: Example of SVM classification (linear separable case)

For learning, we must maximise the margin between the closest samples of the two classes. This can be done by minimizing the objective function in equation 2.19 subject to inequality constraints with the following *Lagrangian* equation:

$$
\begin{aligned}
L(W, b, \alpha_{1:N}) &= \frac{1}{2}\|W\|^2 - \sum_{i=1}^{N} \left[y_i (W^T x_i + b) - 1 \right] \\
p^* &= \min_{W,b} \max_{\alpha_{1:N} \geq 0} \; L(W, b, \alpha_{1:N})
\end{aligned}
\tag{2.21}
$$

This problem is known as the primal problem; p^* is the sought constrained minimum and $w^*, b^*, \alpha^*_{1:N}$ are the arguments of L where it occurs. It can be proven that the same maximum, p^*, and the same argmax, $w^*, b^*, \alpha^*_{1:N}$, can be obtained by solving the following *dual* problem:

$$
d^* = \max_{\alpha_{1:N} \geq 0} \min_{W,b} \; L(W, b, \alpha_{1:N}). \tag{2.22}
$$

Learning using the dual problem has various advantages (e.g. it allows us to use kernels and simplifies the treatment of the non-separable case). More details can be found in [23].

In some cases, maximising the margin between the closest points of the two classes may not be an ideal strategy because a single "outlier" point can significantly affect the separating hyperplane significantly (figure 2.8).

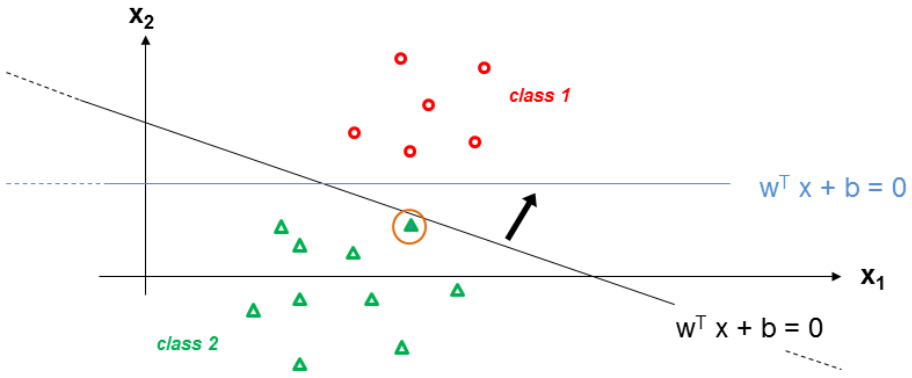


Figure 2.8: A single outlier point can significantly affect the separating hyperplane significantly

In such cases, one can modify the constraints so as to tolerate a few points that do not meet the "≥ 1" constraint, called *soft margin* SVMs. For each such a point, a penalty is accrued to the objective. An updated objective compared to equation 2.19 becomes a trade-off between:

- the maximum margin, which is a promise of future performance (low generalisation error);
- the minimum error on the training set (low empirical risk) by minimising an upper bound for it (equation 2.23).

This justifies the reference to the SVMs as a minimiser for the empirical risk with a regularisation term (regularised minimum empirical risk).

$$
\begin{aligned}
W^*, b^* = \underset{W,b}{\operatorname{argmin}} & \left[\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N} \xi_i \right] \quad s.t. \\
& y_i(W^T x_i + b) \geq 1 - \xi_i \\
& \xi_i \geq 0, \quad i = 1 \ldots N
\end{aligned}
\tag{2.23}
$$

C is an arbitrary weight. The ξ_i are called *slack variables*. It is also possible to use ${\xi_i}^2$ as penalty to discourage large individual errors. Exactly the same constrained objective (equation 2.23) can be used for the much more realistic case of non-separable classes: classes for which there exist no hyperplane that can separate all points from both classes (figure 2.9). In this case, the penalty is added for all violating points. The heavier the violation, the larger is ξ_i.

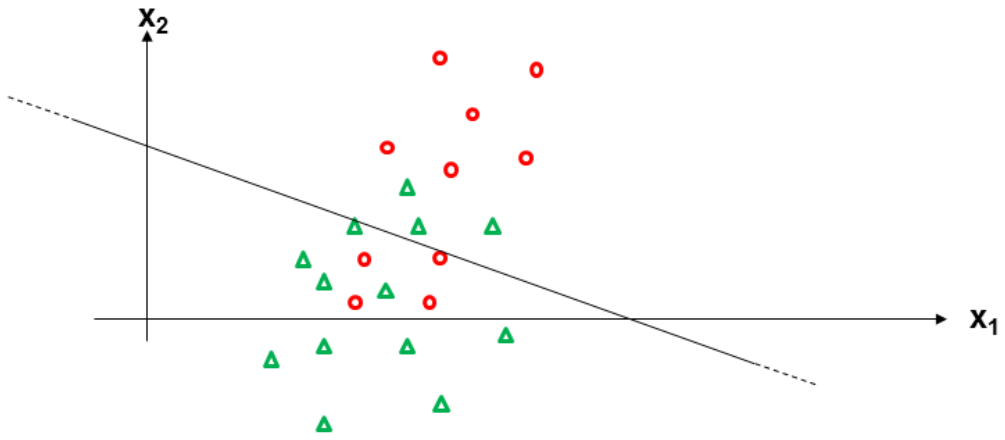


Figure 2.9: Non-separable classes

The standard SVM is a binary classifier. Now, what should be done if one wants to the extend standard SVMs to multi-class problems? The first approach is to create many binary SVMs with various combinations of classes. The main two techniques are known as “one vs all” and “one vs one” [116] (Please note that these techniques can be used to combine any binary classifier, not just the SVM). In “one vs all”, one

trains K binary classifiers, of the type class 1 vs not class 1, ... class K vs not class K. To classify a new sample, x_{new}, all classifiers are applied. The classification is not necessarily consistent: the sample may be assigned to more than one class, or none. Noting the class as y, the classification rule is then:

$$y^* = \underset{y=1...K}{\operatorname{argmax}} \, (W_y x_{new} + b_y). \tag{2.24}$$

In "one vs one", one trains $\frac{K(K-1)}{2}$ binary classifiers, of the type class 1 vs class 2, class 1 vs class 3, ...class $K-1$ versus class K. To classify a new sample, all classifiers are applied and the class that gets the highest number of votes is selected.

The second approach is to do a genuine multi-class extension. Multi-class SVM offers a consistent way to classify a sample into K classes. This approach was first proposed by Weston and Watkins [154] and an alternative formulation was next given by *Crammer and Singer* [32] (equation 2.25):

$$\begin{aligned} W^*, b^* = \underset{W,b}{\operatorname{argmin}} & \left[\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N} \xi_i \right] \quad s.t. \\ & (W_{y_i}^T x_i + b_{y_i}) - (W_k^T x_i + b_k) \geq 1 - \xi_i \\ & \forall k \neq y_i, \;\; \xi_i \geq 0, \;\; i = 1 \ldots N \end{aligned} \tag{2.25}$$

where W is the concatenation of the individual class W_k's, $W^T = \left[W_1^T \ldots W_k^T\right]$, and b is the concatenation of all b_k's. We can also create larger margins with the classes of most undesirable misclassification, which is called *margin-rescaled multi class SVMs*, by changing the constraints in equation 2.25 with the following equation [138]:

$$(W_{y_i}^T x_i + b_{y_i}) - (W_k^T x_i + b_k) \;\geq\; \Delta(y_i, k) - \xi_i. \tag{2.26}$$

Yet now, consider the case of multi-class SVMs with huge numbers of classes. For example, in classifying the states of an HMM, there are easily a million possible different sequences of states. The number of their combinations is exponential in T, the length of the sequence. In other similar problems, instead of a chain of states you want to obtain a tree of states, or a lattice or a graph. In all such cases, the possible classes are a huge number and one cannot apply a standard multi-class SVM. This case is technically known as *structured-output* SVMs (or *structured* SVMs or *struc-*

tural SVMs) because the output from the classifier is not just one label, but many, and there are edges between them, meaning that they need to be classified together and the combinations are exponential (figure 2.10).

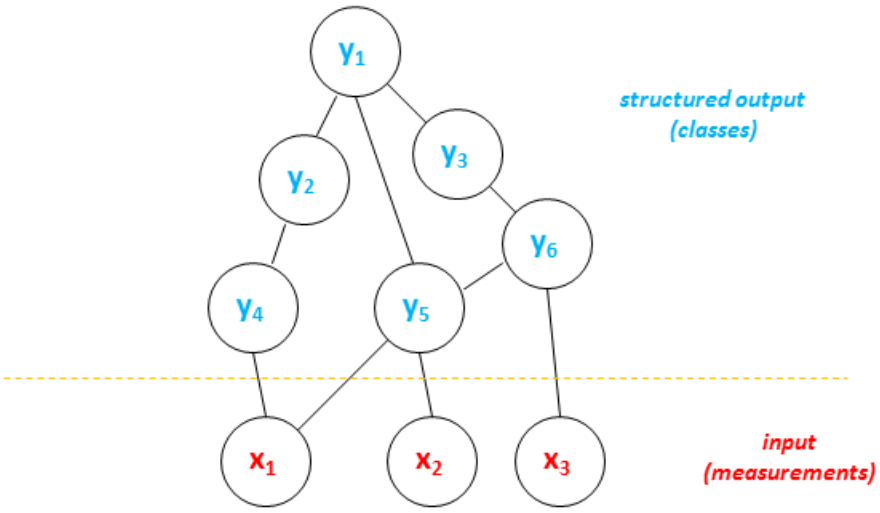


Figure 2.10: Structured-output SVMs

The strategy for tackling this problem is to use the dual form of the SVM with a sub-set of constraints. If we choose the sub-set wisely, we will obtain an approximate solution that differs from the exact solution only by some ϵ that can be quantified. The choice of the sub-set to use is called "constraint generation" and is the fundamental step of structured-output SVM. The constraint generation in brief requires computing, for every measurement x_i:

- the score of x in any possible class y ($score(x_i, y)$);
- the loss that y causes compared to the ground-truth class for x ($\Delta(y_i, y)$, equation 2.27).

$$y_i^* = \underset{y=1...K, y \neq y_i}{\operatorname{argmax}} \; (\Delta(y_i, y) + score(x_i, y)) \tag{2.27}$$

If the violation is $\geq$ current $\xi_i + \epsilon$, we can add the constraint to the working set and solve the objective function with an updated constraint set. This task is repeated

recursively until no new constraint is added to the working set.

Depending on the structure over the classes and measurements, the score function can take significantly different forms. It is therefore convenient to introduce a common notation: we call W the entire vector of parameters and $\Psi(x, y)$ an arrangement of the classes and the measurements such that the score is simply given by $W^T\Psi(x, y)$. A simple example for multi-class SVMs is presented in figure 2.11. In this way, $W^T\Psi(x, y) = W_k^T x + b_k$.

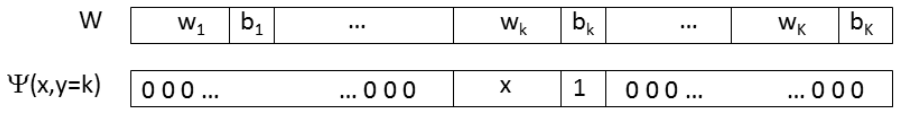


Figure 2.11: Multi-class SVMs

Considering the above definitions, the violated constraint (aka augmented inference) for margin rescaling is:

$$\begin{aligned} & W^T\Psi(x_i, y_i) - W^T\Psi(x_i, y) \;\geq\; \Delta(y_i, y) - \xi_i \quad \forall y \\ & \rightarrow\; \xi_i \;\geq\; -W^T\Psi(x_i, y_i) + W^T\Psi(x_i, y) + \Delta(y_i, y) \quad \forall y \\ & \rightarrow\; \xi_i \;\geq\; \max_y \left(-W^T\Psi(x_i, y_i) + W^T\Psi(x_i, y) + \Delta(y_i, y)\right) \\ & \textbf{NB:} \quad \operatorname*{argmax}_y \left(-W^T\Psi(x_i, y_i) + W^T\Psi(x_i, y) + \Delta(y_i, y)\right) = \\ & \operatorname*{argmax}_y \left(W^T\Psi(x_i, y) + \Delta(y_i, y)\right) \end{aligned} \tag{2.28}$$

In many structured prediction tasks, there is useful modeling information that is not available as part of the training data. This missing information even if not observable, is crucial for expressing high-fidelity models for these tasks and as such it is important to include them in the model as latent variables. We have already seen the extension of some well known generative and discriminative approaches which can deal with hidden variables, namely HMMs and HCRFs. A similar extension has been done for the structural SVM framework by Yu and Joachims [163] in order to include latent variables. In *latent structural SVM*, some of the output variables are unsupervised

(hidden or unknown), i.e. we do not know their value during training (figure 2.12).

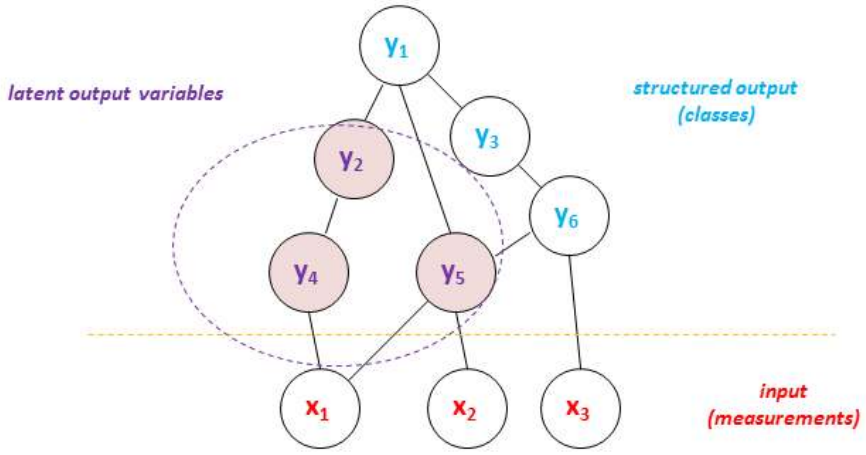


Figure 2.12: Latent structured-output SVMs

Let $\{y, h\}$ denote the output variables where y is known during training and h is unknown during training. The formulation of the constraint in equation 2.28 becomes:

$$W^T \Psi(x_i, y_i, h_i) - W^T \Psi(x_i, y, h) \;\geq\; \Delta((y_i, h_i), (y, h)) - \xi_i \tag{2.29}$$

The main problem in the above equation is that h_i is unknown. In probability theory (e.g. HMMs and HCRFs), we solve this problem by marginalizing h. But the operation of marginalization is a gift of the specific properties of probability. Since latent structural SVM is not a probabilistic model, h cannot be marginalized and is instead assigned with a "best" value (instead of marginalizing h, we maximise in it). In brief, the strategy of Yu and Joachims [163] is akin to a clustering algorithm alternating a step of optimization (equation 2.30) with one of assignment (equation 2.31) until convergence to some local minimum is reached.

$$
\begin{aligned}
&W^* = \underset{W}{\operatorname{argmin}} \left[\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N}\xi_i \right] \quad s.t. \\
&W^T\Psi(x_i, y_i, h_i^*) - W^T\Psi(x_i, y, h) \;\geq\; \Delta((y_i),(y,h)) - \xi_i \\
&\forall y \neq y_i, h \neq h_i^*
\end{aligned}
\tag{2.30}
$$

$$
h_i^* = \underset{h}{\operatorname{argmax}}\; W^{*T}\Psi(x_i, y_i, h_i^*) \tag{2.31}
$$

Once the model is trained, inference is as usual (equation 2.32), with all the output variables, $\{y, h\}$, inferred and then h discarded (unless you are interested in retaining them for some reasons).

$$
y^*, h^* = \underset{y,h}{\operatorname{argmax}}\; W^T\Psi(x, y, h) \tag{2.32}
$$

Additional information on this topic can be found in [31, 32, 97, 138, 163].

Chapter 3

Action Recognition by Graph Embedding

The problem of human action recognition has received increasing attention in recent years for its importance in many applications. Yet, the main limitation of current approaches is that they do not capture well the spatial relationships in the subject performing the action. This chapter presents a study which uses graphs to represent the actor's shape and graph embedding to then convert the graph into a suitable feature vector. In this way, we can benefit from the wide range of statistical classifiers while retaining the strong representational power of graphs. This chapter shows that, although the proposed method does not yet achieve accuracy comparable to that of the best existing approaches, the embedded graphs are capable of describing the deformable human shape and its evolution over time. This confirms the interesting rationale of the approach and its potential for future performance.

3.1 Prior Work and Our Contributions

Many approaches have been proposed for human action recognition to date, including bag of words (or features) [37, 70], dynamic time warping [12], hidden Markov models [158] and conditional random fields [105]. A recent survey has offered a systematic review of these approaches [103]. The problem of incorporating the structural information in the recognition process has been raised in [45]. However, the problem

of a suitable feature set which can well encapsulate the deformable shape of the actor is still partially unresolved. As an alternative, graphs offer a powerful tool to represent structured objects and as such are promising for human action recognition. Ta *et al.* in [134] have recently used graphs for activity recognition. However, to assess the similarity of two instances, they directly compare their graphs and this is prone to significant noise. An alternative to the direct comparison of action graphs is offered by graph embedding: in each frame, the graph representing the actor's shape can be converted to a finite set of distances from prototype graphs, and the distance vector is then used with conventional statistical classifiers. Graph embedding has been successfully used in the past for fingerprint and optical character recognition [112]. To the best of our knowledge, this is the first work proposing to employ graph embedding for human action recognition. Such an extension is significant since feature vectors need to prove discriminatory along the additional dimension of time.

In this chapter, we propose to extract spatial feature points from each frame and use them as nodes of a graph describing the actor's shape. With an adequate prototype set, we convert the graph to a set of distances based on the probabilistic graph edit distance (P-GED) of Neuhaus and Bunke [94]. P-GED is a sophisticated edit distance capable of learning edit costs directly from a training set and weighing each edit operation individually. The feature vectors of each frame are then composed into a sequence and analysed by means of a conventional sequential classifier. The recognition accuracy that we obtain is not yet comparable to that of the best methods from the literature; however, results show unequivocally that the embedded vectors are capable of representing the human posture as it evolves along the time and setting the basis for potential future improvements.

3.2 Proposed Methods

In this section, we first provide a recall of graph embedding. We then describe the methodology proposed in our work to incorporate graph embedding into an action recognition approach on the popular KTH action dataset [124]. Finally, we present and discuss an experimental evaluation of the proposed approach on the selected human action dataset.

3.2.1 Graph Embedding

In the literature, "graph embedding" refers interchangeably to the embedding of a graph as a whole into a point in vector space, or the embedding of its set of nodes into a set of corresponding points in vector space. In this work, we assume the former meaning, although similar embedding techniques can be applied in the two cases and for other types of non-vectorial objects such as strings or trees [110]. The embedding assumes that a set of objects is given alongside distance values between any two objects in the set. The goal is that of converting the set of objects into a set of points in a vector space of given dimensionality while ensuring certain properties or constraints. Well-known embedding techniques include Laplacian eigenmaps, commute times, symmetric polynomials, and kernel principal component analysis, amongst others [8, 104, 122, 157]. After the embedding of the initial set of objects, it is also possible to embed new, out-of-sample objects, although this is not always straightforward. An alternative embedding approach is to make use of a given set of "prototype" objects (or prototypes, for short) which can equally embed in-sample and out-of-sample data, in a way that is not unlike that of eigenvectors in principal component analysis. Let $G = \{g_1, g_2, \ldots, g_m\}$ be a set of graphs, $P = \{p_1, p_2, \ldots, p_n\}$ be a set of prototype graphs with $m > n$, and d be a dissimilarity measure. For embedding any graph $g_j \in G$ by way of P, the dissimilarity measure $d_{ji} = d(g_j, p_i)$ of graph g_j to prototype $p_i \in P$ is computed $\forall i$. Then, an n-dimensional vector $(d_{j1}, \ldots, d_{jn})$ is assembled from all the n dissimilarities. With this procedure, any graph can be individually transformed into a vector of real numbers. Formally, the mapping $t^P : G \to \mathbb{R}^n$ is defined as the following function:

$$t^P(g) \to (d(g, p_1), \ldots, d(g, p_n)) \tag{3.1}$$

where $d(g, p_i)$ is a dissimilarity measure between graph g and prototype p_i [112], [100]. Prototype-based embedding is certainly the simplest and fastest embedding approach and for these reasons is adopted hereafter.

3.2.2 Feature Extraction

The approach used for extracting informative features consists of the following main steps:

Step 1: Dataset As human action dataset, we have used the KTH human action dataset [124] for its widespread past utilisation. The KTH human action dataset contains six different human actions: walking, jogging, running, boxing, hand-waving and hand-clapping, all performed at various times over homogeneous backgrounds by 25 different actors in four different scenarios: outdoors, outdoors with zooming, outdoors with different clothing and indoors (figure 3.1). This dataset contains 2391 sequences, with each sequence down-sampled to the spatial resolution of 160×120 pixels and a length of four seconds on average. While this dataset consists of simplified actions, it is challenging in terms of illumination, camera movements and variable contrasts between the subjects and the background. KTH has been a de-facto benchmark in the last few years and many results are available for comparison.

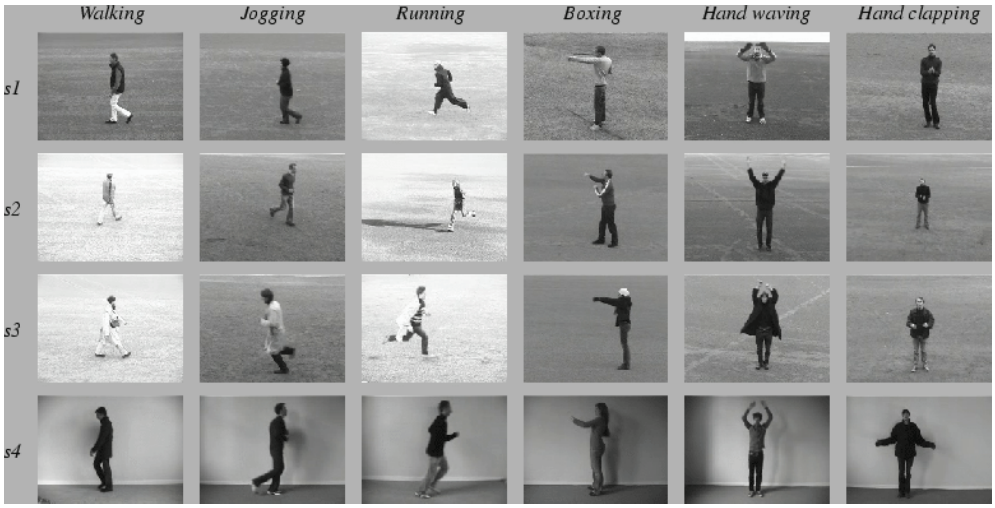


Figure 3.1: KTH human action database: examples of sequences corresponding to different types of actions and scenario [124].

Step 2: Graph Building As a preliminary step, a modified tracker is used to extract a bounding box of each actor in each frame [25]. Over the dataset at hand, the tracker performs really well, providing bounding boxes which almost invariably contain the actor in full size. As the next step, a number of *scale invariant feature transform* (SIFT) keypoints [83] are extracted within the actor's bounding box in each

video frame using the software of Vedaldi and Fulkerson [142]. Based on the chosen threshold, their number typically varies between 5 and 8. Example results of this step are illustrated in figure 3.2. After extraction, the location of each SIFT keypoint, (x, y), is expressed relatively to the actor's centroid and employed as a node label for an attributed graph describing the human's shape. In a preliminary study, we found that graphs with only labeled nodes granted comparable accuracy to graphs with both labeled nodes and labeled edges, yet resulted in faster processing. We therefore decided to employ graphs consisting only of labeled nodes.

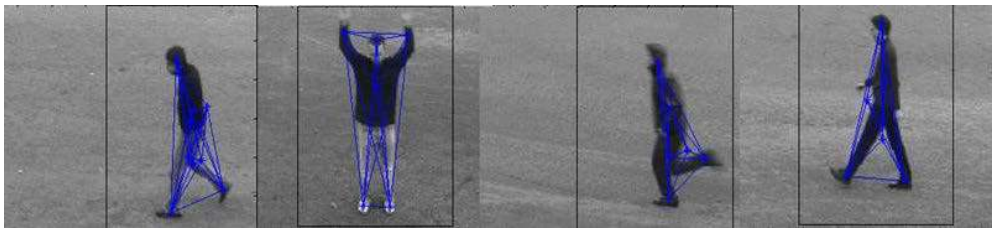


Figure 3.2: Bounding box generated from a modified tracker [25] using the KTH action dataset and the extracted SIFT keypoints composed into a graph.

Step 3: Posture Set In order to identify a prototype set which could lead to meaningful feature vectors in the embedded space, a number of different reference postures was chosen to describe all human shapes in the action dataset. For KTH, we carried out a manual analysis of about 400 frames from the six classes and manually chose a set of 16 different reference postures across all human actions (running, walking, boxing, jogging, hand-waving, hand-clapping). Such selected postures should prove adequate for also recognising human actions in any other dataset where the actors are approximately in full view such as UCF Sports [119] and MuHAVi [129]. For training purposes, we manually selected a number of different frames varying in scenario (e.g. outdoor, outdoor with different clothes, indoor), action (e.g. hand waving, hand clapping, jogging) and actor (e.g. person01, person25, person12) (see figure 3.3). We have then trained a single probabilistic graph edit distance (P-GED) for each posture set, $\{PGED_1, \ldots, PGED_{16}\}$.

Step 4: Prototype Selection As stated in subsection 3.2.1, an appropriate choice

Figure 3.3: Examples of selected postures from the KTH action dataset.

of the prototype set, P, plays a critical role in this approach as it impacts the classification accuracy. Given that we avail ourselves of a labelled training set, we have decided to employ class-based approaches for prototype selection. The details of prototype selection methods can be found in subsection 3.2.3.

Step 5: Feature Vector The embedding of a graph by any of the above prototype selection methods leads to a 16-dimensional feature vector describing the shape of a single actor in a frame. Time series of such vectors may prove action-discriminative. Yet, we decided to augment the feature vector by some basic information about the actors' global motion and location relative to the bounding box. We thus added the horizontal displacement between the bounding boxes of two successive frames (which is proportional to the horizontal velocity) and the location of the actor's centroid relative to the bounding box. This leads to an overall 19-dimensional feature vector with information about the shape, motion and location of the actor in a frame. Figure 3.4 shows time series of the feature vector for a boxing action in KTH. An analysis of the individual contributions of the shape, motion and location information is presented in 3.2.4.

3.2.3 Prototype Selection Techniques

In the following, we describe three popular, existing approaches and one of the approaches proposed in our work.

Class-based Center Prototype Selection (c-cps) In this method, a prototype set, $P = \{p_1, \ldots, p_n, \ldots, p_N\}$, is generated from a labeled training set, $C = \{C_1, \ldots, C_n, \ldots, C_N\}$, with each p_n prototype located in, or near, the "center" of the graphs from the n-th

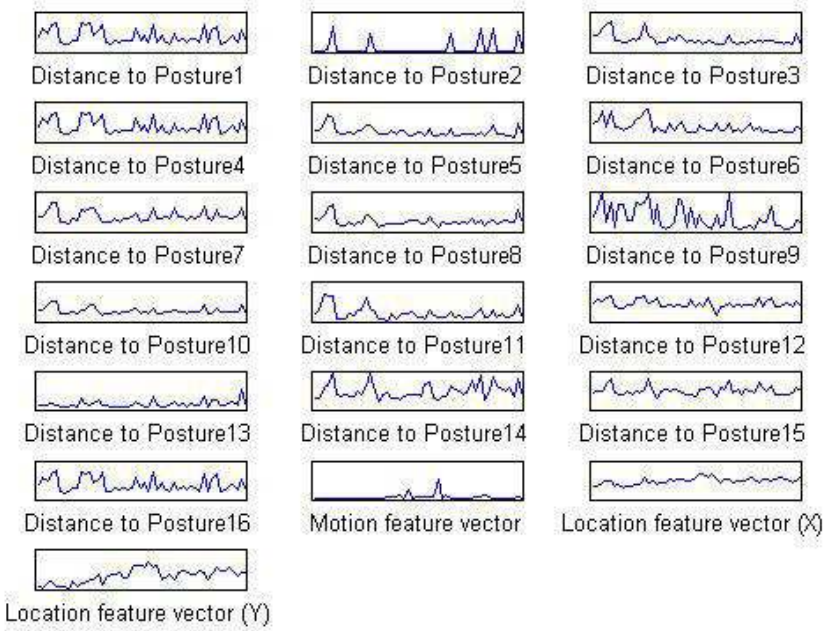


Figure 3.4: The time-sequential values of a 19-dimensional feature vector obtained from graph embedding based on the $c - dps$ for one action (boxing) performed by one subject in the KTH action dataset.

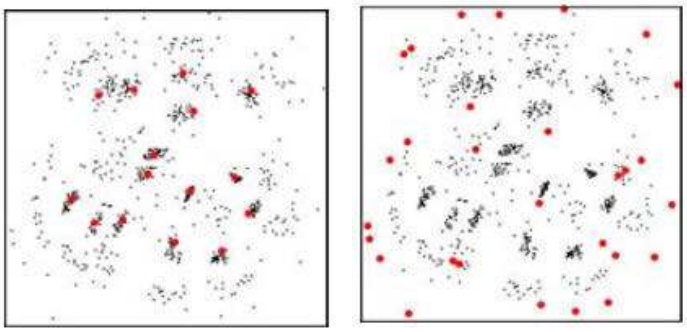
class, C_n (figure 3.5(a)). To implement the notion of center, we select the median graph from sample set $C_n = \{g_{n1}, \ldots, g_{nj}, \ldots, g_{nN_n}\}$, defined as the g_{nj} graph such that the sum of distances between g_{nj} and all other graphs in C_n is minimal [112]:

$$p_n = \arg \min_{g_{nj} \in C_n} \sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni}). \tag{3.2}$$

Class-based Border Prototype Selection (c-bps) This approach chooses the prototype set, P, with each p_n prototype situated at the "farthest border" of its class, C_n (figure 3.5(b)). Again, the notion of border is vague in class domain. The rationale for this selection is that of having prototypes which are at maximum distance from the training graphs and generate feature vectors with the largest values. To implement it, we select the marginal graph from the sample set of class $C_n =$

$\{g_{n1}, \ldots, g_{nj}, \ldots, g_{nN_n}\}$, defined as the g_{nj} graph such that the sum of distances between g_{nj} and all other graphs in C_n is maximal [112]:

$$p_n = \arg \max_{g_{nj} \in C_n} \sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni}). \tag{3.3}$$



(a) c-cps (b) c-dps

Figure 3.5: Illustration of the different prototype selectors applied to the training set. The number of prototypes is defined by N = 30. The prototypes selected by the respective selection algorithms are shown with red dots. Image courtesy of Kaspar Riesen [111].

Class-based Random Prototype Selection (c-rps) Given the relative arbitrariness of the above selections, a random choice of the class prototype is a plausible alternative. In c-rps, each p_n prototype is randomly selected from class C_n with uniform probability [112]:

$$p_n = g_{nj} \in C_n, j \sim p(k = 1 \ldots N_n) = \frac{1}{N_n}. \tag{3.4}$$

Class-based Discriminative Prototype Selection (c-dps) All of the above selection approaches choose the class' prototype based solely on the graphs in the class. This is in a way reminiscent of generative classifiers, where a class' parameters are estimated based on only the samples from that class. Discriminative classifiers, instead, choose parameters based on the information from multiple classes at once, maximizing objective functions such as the class margin, Fisher discriminants and others, and often proving more accurate than their generative counterparts. Inspired by discriminative

approaches, we propose herewith a class-based discriminative prototype selection approach, where each p_n prototype is chosen as the graph g_{nj} that minimizes the ratio between the sum of distances between g_{nj} and all other graphs in C_n and the sum of distances between g_{nj} and all graphs in the other classes, $\overline{C_n}$:

$$p_n = \arg \min_{g_{nj} \in C_n} \frac{\sum_{g_{ni} \in C_n, g_{ni} \neq g_{nj}} d(g_{nj}, g_{ni})}{\sum_{g_{ni} \in \overline{C_n}} d(g_{nj}, g_{ni})}. \tag{3.5}$$

This selection approach is analogous to minimizing the ratio between the within-class and between-class scatter matrices in vector spaces. Discriminative prototype selection methods is a fundamental contribution of this thesis and we explain various approaches revolving around the same idea in Chapter 5.

3.2.4 Classification

We have evaluated the recognition accuracy of the proposed method with the following classification approaches:

The Bag of Words Paradigm The past decade has seen the growing popularity of Bag of Words (BoW) approach to many computer vision tasks including image classification, action recognition, texture recognition. BoW approach is characterized by the use of an orderless collection of extracted features. Lacking any structure or spatial information, it is perhaps surprising that this choice of image representation would be powerful enough to match or exceed state-of-the-art performance in many of the applications to which it has been applied. Due to its simplicity and performance, the Bag of Words approach has become well-established in the field [98]. Here, we describe the procedure for generating a fixed size feature vector for an image with BoW approach.

1. Learn the "vocabulary": Extract features from all samples in a training set and quantize, or cluster, these features into a "visual vocabulary", where each cluster represents a "visual word". In some works, the vocabulary is called the "visual codebook". Words in the vocabulary are the codes in the codebook.

2. Quantize features of a new image using the visual vocabulary: Extract features

from a novel image and assign each feature to the closest visual word in the vocabulary.

3. Represent samples by frequencies of "visual words": Record the counts of each visual word that appears in the sample to create a normalized histogram representing a "vector". This vector is the Bag of Words representation of the sample. This approach can be applied to images, videos and various types of features.

For the classification task over the extracted vectors we can easily employ an SVM classifier or any other classifier.

HMM with maximum conditional likelihood training In addition to the positions above, let $Y = \{Y_1, \ldots, Y_k, \ldots, Y_K\}$ be the set of K different groups of ground-truth labels for the observation sequences in each class; and each $Y_k = \{y_k^1, \ldots, y_k^{N_k}\}$ be the group of ground-truth labels for the N_k observation sequences of action class k. Each such a label takes value in A, the set of action classes. Here, the availabilty of the ground-truth labels allows defining a different objective function, known as *conditional likelihood*, for the setting of the θ parameters [133]:

$$\mathcal{L}(\theta; Y, X) = \prod_{k=1}^{K} \prod_{e=1}^{N_k} p(y_k^e | X_k^e, \theta). \tag{3.6}$$

Parameters $\theta = \{\theta_1, \ldots, \theta_k, \ldots, \theta_K\}$ are then selected to maximize the conditional likelihood as in:

$$\theta^* = \arg\max_{\theta}(\mathcal{L}(\theta; Y, X)). \tag{3.7}$$

The parameters estimated by maximizing (3.6) are more promising for classification than those estimated with the conventional likelihood since conditional likelihood $p(y'|X', \theta')$ for a given class, y', and measurement, X', is, with different wording, the posterior probability of class y' given measurement X'. In essence, training the parameters with the conditional likelihood target maximizes the posterior probability of the correct class labels over the entire training set. As such, it is an example of *maximum score* training [127].

However, maximizing the conditional likelihood for the HMM is not trivial. Therefore, in this work we resort to an approximation: at each iteration of the Baum-Welch

algorithm (which is guaranteed to increase the conventional likelihood), we evaluate (3.6) and store the parameters. Upon convergence of Baum-Welch, the value of the parameters corresponding to the largest conditional likelihood encountered during the iterations are selected.

Hidden Conditional Random Field The HCRF is a powerful discriminative approach which can model time series data considering hidden measurements. In previous HMM-based approaches, the model for each action is trained separately while we can jointly train a single model for all actions using a *multi-class* HCRF. In other words, because an HCRF is trained by maximizing conditional probability, all parameters are jointly and discriminatively optimized. A decoder is then used to find the most likely action, a_k^*, given a new sequence of measurements, X_{new}, and the trained model parameters, θ. It is obvious that we can also train k different number of HCRF models following "one vs all" or "one vs one" techniques.

3.3 Experimental Results

In this section, we evaluate the recognition accuracy of the proposed method. We first evaluate various choices of feature vectors and then compare our approach based on the best feature vector with the state of the art. All of these experiments were performed on a computer with an Intel(R) Core(TM)2 Duo CPU (E8500, 3.16GHz) and 4GB RAM using Matlab R2009b.

3.3.1 Evaluation of the feature vectors

The 19-dimensional feature vector described in section 3.2.2 contains shape, motion and location features jointly. In order to assess the individual contribution of these different types of features, we have conducted experiments with feature vectors containing only shape, motion or location features in isolation. To this aim, we have used *leave one (actor) out cross validation* (LOOCV) reporting a correct classification rate (CCR) for each feature vector, the c-cps as a prototype selector and standard HMM with maximum likelihood training as a classifier. It can be seen that none of the individual type of features was capable of achieving high accuracy in isolation; in all

cases, recognition accuracy was below 50% (table 3.1). However, these features show interesting complementarity: for instance, the motion features report good accuracy in recognising the Jogging class, but a rather low performance on the Boxing class (which is mainly a stationary class). Conversely, the graph-embedded shape features report good accuracy on the Boxing class, but cannot discriminate well between classes such as Jogging and Running where the articulated shape is similar, yet speed of execution varies remarkably. This complementarity is at the basis of the higher performance achieved by the joint vector which jumps to 70.00%, as shown by table 3.2.

Table 3.1: The average CCRs on the KTH action dataset based on separate feature vectors for motion, location and shape.

validation technique	motion	location	shape
LOOCV-CCR	49.34%	45.67%	47.63%

Table 3.2: Action confusion matrix (%) for the proposed method based on the LOOCV test approach on the KTH action dataset. The average CCR is 70.00%.

	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	80	9	8	1	1	1
Clapping	10	59	25	1	2	2
Waving	8	22	66	1	0	3
Jogging	0	0	0	56	21	23
Running	0	0	0	17	74	9
Walking	0	0	0	8	7	85

3.3.2 Comparison to the state of the art

Accuracy measurements on the KTH database have been performed with different methods by different papers in the literature. For easier comparison, in this section we have used the test approach presented by Schuldt *et al.* in [124]. With this test

approach, all sequences are divided into 3 different sets with respect to actors: training (8 actors), validation (8 actors) and test (9 actors). The classifier is then tuned using the first two sets (training and validation sets), and the accuracy on the test set is measured by using the parameters selected on the validation set, without any further tuning.

We have first evaluated the 19-dimensional feature vectors obtained from four different prototype selectors: c-dps, c-cps, c-bps and c-rps. We have used standard HMM with maximum likelihood training (HMM_{ml}) and maximum conditional likelihood training (HMM_{mcl}) as classifiers. Table 3.3 shows the recognition accuracy with these approaches for each prototype selector. The proposed discriminative prototype selector, c-dps, achieves the highest accuracy. In addition, the proposed conditional likelihood training permits higher accuracy than conventional likelihood training in most cases.

Table 3.3: Classification accuracy of HMM_{ml} and HMM_{mcl} applied to feature vectors from different prototype selectors (c-dps, c-cps, c-bps and c-rps).

Schuldt's validation		
Prototype Selector	HMM_{ml} CCR(%)	HMM_{mcl} CCR(%)
c-dps	**67.80**	**70.35**
c-cps	66.75	68.85
c-bps	64.05	64.15
c-rps	65.50	65.50

Discriminative classifiers (e.g. SVM, HCRF) generally report higher accuracy compared to generative approaches (e.g. HMM) in the literature. As such, we have selected the best extracted feature vectors (19-dimensional feature vector from class-based discriminative prototype selector) and tried standard SVM and HCRF as classifiers. For SVM, we ignore the sequential information and follow the bag-of-words approach and use k-means clustering with N = {50,100,200,400,600} clusters for quantization and an SVM classifier with different kernels (Linear, RBF, Chi Square and Histogram Intersection) for classification. As a software, we have employed the LIBSVM package [22]. Similarly to HMM, we consider the sequential information

with HCRF. We have used the library developed by Morency *et al* [90] and set Q = {6,8,10,20}, $Windowsize$ = {0, 1} and initialisation to *Gaussian*. Table 3.4 shows the CCR on the test set, using the parameter's values that scored the best accuracy on the validation set. Results show that, with these features, the performance of SVM and HCRF is not better than HMM.

Table 3.4: The average CCRs (%) on the KTH action dataset based on the 19-dimensional feature vectors from c-dps and SVM and HCRF as classifiers.

19D Feature Vectors from c-dps		
Validation technique	SVM	HCRF
Schuldt	58.7%	67.11%

We now compare the best overall accuracy achieved from graph embedding approach with the results reported in the literature (table 3.5). Our overall accuracy is 70.35%. This result is not comparable with the best accuracies reported in the literature: it is not far from the accuracy reported by Schuldt*et al.* [124], but much lower than that reported by Guo *et al.* in [54] and many others. These approaches mostly leverage spatio-temporal descriptors. Given that HMM as a classifier seems to perform satisfactorily, we have to conclude that this feature set is not sufficient to prove action-discriminative with the KTH dataset. In the following chapter, we will therefore analyse the performance of a feature set augmented with local features.

Table 3.5: Average class accuracy on the KTH action dataset.

Method	**Ours**	Schuldt *et al.* [124]	Laptev *et al.* [72]	Guo *et al.* [54]
Schuldt	**70.35%**	71.70%	91.80%	97.40%

3.4 Discussion and Conclusions

In this chapter, we have presented a novel approach for human action recognition based on graph embedding. To this aim, an attributed graph is used to represent the actor's

shape in each frame and then graph embedding is used to convert the graph into a feature vector in order to have access to the wide range of current classification methods. Although this method does not match the accuracy of existing approaches, it generates a novel methodology for human action recognition based on graph embedding, and may outperform existing methods in conjunction with other features. Furthermore, we have shown the ability of this approach to encapsulate the global structural information for human action recognition. Based on our judgment, the main difficulty faced by the proposed approach is the extraction of a reliable set of keypoints in each frame. Due to noise and variable appearance, the extracted set changes significantly over the frames (see figure 3.6). This leads to heavy changes to the structure of the graphs and, likely, the embedded vectors. Another possible limitation is that it ignores texture information (unlike features such as STIP, HOG, HOF). In the next chapter we investigate the possibility of the fusion of global spatial relationships provided by graph embedding and textural information.

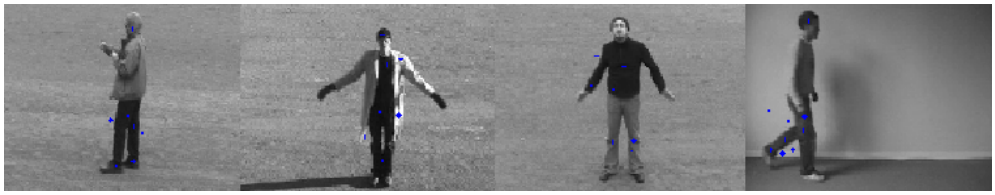


Figure 3.6: Instance images illustarte that the SIFT keypoints are not able to capture the body shape sufficiently well to be used as a shape descriptor.

Chapter 4

Fusion of Texture and Structural Features for Action Recognition

In the previous chapter of this monograph, we have shown the ability of the graph embedding approach to encapsulate the actor's shape in each frame into a finite set of distances from prototype graphs. We have also shown that this distance vector can be used as a feature vector with any conventional statistical classifiers. However, we did not consider any local descriptors in the mentioned approach. These descriptors and, in particular, appearance descriptors centred around spatio-temporal interest points (STIPs) [70], have gained increasing popularity for human action recognition since they describe salient points in space and time and have demonstrated strong recognition performance. Nevertheless, spatio-temporal features may fail when the activities become complex since they are unable to capture the global spatial relationships in the subject performing the action [96]. As such, the necessity of a systematic approach for the fusion of global spatial relationship and the local spatio-temporal information can be seen in the literature. Such an approach enjoys the strong representational of both graphs and local descriptors.

For this reason, in this chapter we present a method capable of fusing the information of both graphs and local features. We first present a joint action segmentation and classification approach based on an extended hidden Markov model, named hidden Markov model for multiple, irregular observations (HMM-MIO), which is capable of processing sparse local features. This requires:

- processing measurements which are irregular in space and time;
- mollifying issues deriving from high dimensionality;
- dealing with heavy-tailed distributions and outliers.

By employing the proposed model, we have introduced a novel framework for the fusion of structural information provided by graph embedding and the textural information of STIP descriptors.

4.1 Prior Work and Our Contributions

Over the last decade, action recognition approaches have vastly leveraged on the notion of local spatio-temporal features [70]. Extracting such features consists of a detection and a description stage. The first stage requires detecting all the points in the actor's bounding box where significant "spatio-temporal change" occurs. The second stage consists of collecting a local descriptor for each detected point that summarizes its local spatio-temporal appearance. Actions as diverse as an elbow bending while picking a cup or a reclining head tend to generate specific descriptor values. As an example, Fig. 4.1 shows the points detected in the video of a person walking outdoors using the spatio-temporal interest points (STIPs) of Laptev *et al*. [72]. As shown in the sequence of frames, the detected points are irregular in both space (i.e. area in the frame) and time (i.e. number of points per frame). The same irregular nature in space and time is also shared by other, more specialized detectors such as the recently proposed poselet detectors [86]. While it is possible to collect descriptors over regular grids [147, 148], descriptors at interest points are computationally much lighter and suitable for certain applications. In addition, descriptors are also typically high dimensional, affected by outliers and characterized by long-tailed statistics [60].

The baseline approach for action classification is known as "bag-of-words" [52, 68, 72, 124]. As we mentioned in Chapter 3, in this approach the multi-dimensional descriptors are first quantized based on a learned codebook. Then, for each action instance, a histogram is computed over its quantized descriptors and used as input for a supervised classifier. Notwithstanding its simplicity, this approach has proved capable of remarkable recognition accuracy [52, 68, 72, 124, 148]. Extension to segmentation

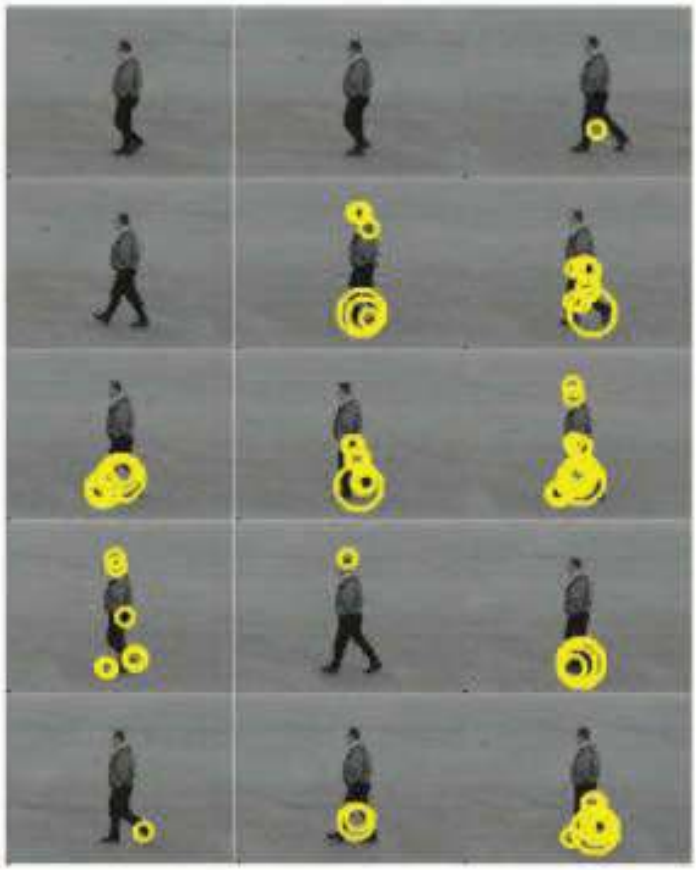


Figure 4.1: Example of the spatio-temporal interest points from [72] in a video from the KTH action dataset. Frames are displayed in row-major order. The radius of circles is proportional to the scale at which change is detected. Note the variable number of points appearing in subsequent frames.

can be obtained by simply splitting the video into overlapping windows and repeating classification for each window [40]. Yet, the size of the window and the overlap between windows is arbitrary, with possible impact on accuracy and temporal resolution. Temporal graphical models offer a more principled approach to the segmentation problem [57, 76, 140]. For this reason, in this chapter we adopt an extended hidden Markov model - named hidden Markov model for multiple, irregular observations

(HMM-MIO) hereafter - capable of providing classification and time segmentation over a) observations which are irregular in time and space, b) high-dimensional observation spaces, and c) outliers and heavy-tailed distributions. This model was recently proposed in [28] and is extended in this thesis over spatial regions. Experiments are performed over the KTH dataset, a "stitched" version of the popular KTH dataset [124] where individual actions have been collated into uninterrupted sequences, and the challenging CMU multi-modal activity dataset (CMU-MMAC) which displays scenes of cooking actions [34]. The achieved accuracies show that HMM-MIO is capable of competitive performance in action recognition and joint action segmentation and classification.

Another limitation of spatio-temporal features is that they may fail when the activities become complex, since they are unable to capture the global spatial relationships in the subject performing the action [96]. Conversely, graphs are a powerful tool for representing structured objects and as such have been used for action recognition in a recent work from Ta *et al* [134]. Nevertheless, in [134] graphs are directly compared to assess the similarity of two action instances, a procedure that is prone to significant noise. An efficient alternative to the direct comparison of action graphs is offered by graph embedding: in each frame, the graph representing the actor's shape can be converted to a finite set of distances from prototype graphs, and the distance vector can then be used as a feature vector with any conventional statistical classifiers. Other approaches leveraging on a graphical representation of the actor are based on models akin to Pictorial Structures [47]. Such models were originally proposed for limb motion tracking and require higher resolution imagery to ensure accurate fitting. In all cases, purely structural approaches do not take advantage of the useful information offered by spatio-temporal appearance descriptors. In this chapter we have employed HMM-MIO in a novel framework for the fusion of the structural information provided by graph embedding and the spatio-temporal information given by STIP descriptors, thus benefiting from both powerful representations and overcoming their respective limitations.

4.2 Proposed Methods

In this section, we first present the generative model of the extended HMM, called HMM-MIO, for classification and time segmentation; and also its evaluation on the KTH, Stitched KTH and CMU-MMAC datasets. The proposed approach using HMM-MIO for the fusion of structural and spatio-temporal features is then described and evaluated on the KTH action dataset.

4.2.1 Classification and Time Segmentation

HMM offers a natural model for action classification and segmentation as shown in Fig 4.2. In the case of joint classification and segmentation (figure 4.2(a)), each state is assumed to be the action at that time frame. Given an observation sequence $x_{1:T}$, state decoding, i.e. $y^*_{1:T} = \text{argmax}_{y_{1:T}} \; p(y_{1:T}|x_{1:T})$, retrieves the most probable action sequence. Model estimation is typically performed with supervised states, assuming ground-truth knowledge of a labeled training set of action sequences. Conversely, in the case in which the entire observation sequence corresponds to a single, pre-segmented action, HMM can be used with a different semantic. In this case the target is a single action label, $a^* = \text{argmax}_a \; p(a|x_{1:T})$, obtained from Bayes' inversion rule and marginalization of the state sequence (equation 4.1). The sequence of states represents the dynamic within an action rather than between the actions as in the previous case. The graphical model is shown in figure 4.2(b).

$$p(a|x_{1:T}) \propto \sum_{y_{1:T}} p(x_{1:T}, y_{1:T}|a) \; p(a) \tag{4.1}$$

4.2.1.1 HMM-MIO

In action recognition, typical local features such as STIP descriptors are irregular in space and time and characterized by high dimensionality. In addition, their empirical distributions tend to exhibit heavy tails and outliers [60]. Therefore, the conventional observation model of HMM requires extensions that we provide as follows with HMM-MIO:

- To deal with space irregularity, the video frame is partitioned over a uniform grid

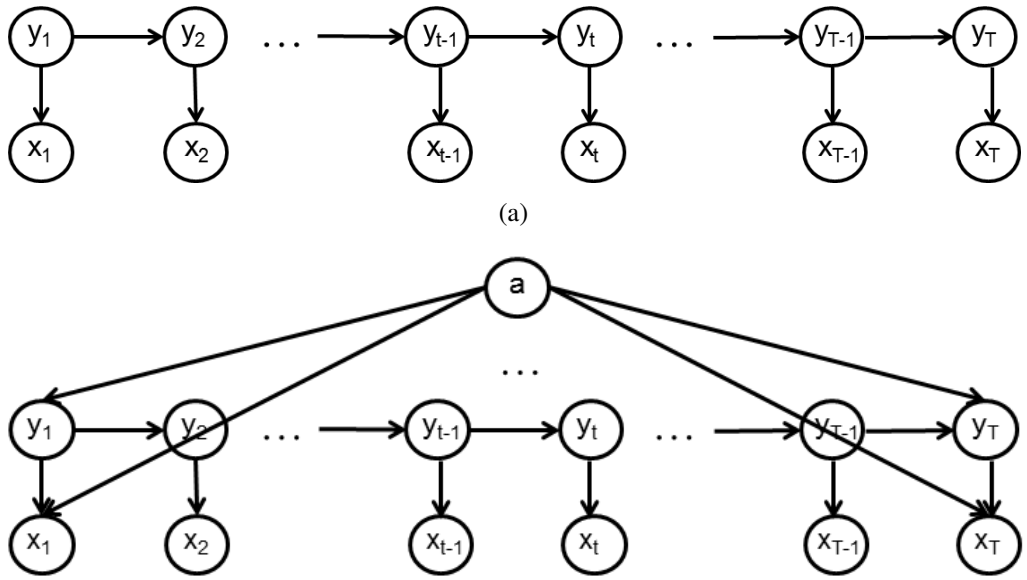


Figure 4.2: (a) Decoding the state sequence, $y_{1:T}$, of an HMM provides joint action classification and segmentation from observations $x_{1:T}$; (b) decoding variable a by Bayes' inversion rule and marginalization of $y_{1:T}$ provides a single action label for the entire sequence $x_{1:T}$.

with a small number of cells (typically, 1 to 4; see figure 4.3), and a separate observation density is modelled for the observations of each cell. The cell index, S, is a fully observed variable for each observation.

- The model is extended to multiple observations per frame (including none) as described in the rest of this section.

- High dimensionality is mollified by adopting the probabilistic principal component framework [137]. With this approach, the covariance matrix of each observation density, Σ, is constrained to decompose as $W^T W + \sigma^2 \mathbb{I}$, where W is a matrix of limited vertical size. This constraint equates to modelling the observations over a lower dimensional space with spherical Gaussian noise.

- Both heavy-tailed statistics and outliers are taken into account by modelling the observation densities by a long-tailed distribution such as the Student's t [24].

- Possible multimodality of the observation densities is accounted for by using mixture distributions.

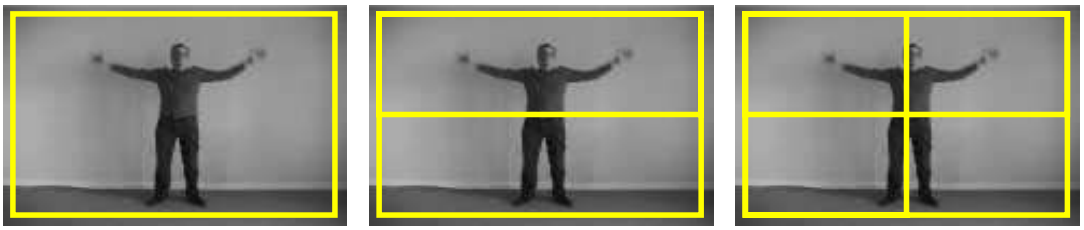


Figure 4.3: The uniform grid over the actor's area.

We note each observation as x_t^n, with t the frame index and $n = 1 \dots N_t$ the observation index within the frame. Each observation consists of the pair, $x_t^n = \{d_t^n, s_t^n\}$, of the descriptor, d_t^n, and the cell index where it occurs, s_t^n. Observations probabilities are assumed to factorize as $p(x_t^n) = p(d_t^n|s_t^n)p(s_t^n)$ and $p(s_t^n)$ is assumed uniform. For the multiple observations in a frame, we posit:

$$\begin{aligned} p(x_t^{1:N_t}) \equiv p(x_t^1, \dots x_t^{N_t}) &= \prod_{n=1}^{N_t} p(x_t^n|y_t), \text{if } N_t \geq 1 \\ &= 1, \text{if } N_t = 0 \end{aligned} \tag{4.2}$$

Posing $p(x_t^{1:N_t})$ = 1 in the case of no observations is equivalent to a missing observation and has neutral effect in the chain evaluation of the HMM. The generative model of HMM-MIO:

$$\begin{aligned} p\left(x_{1:T}, y_{1:T}\right) &\equiv p\left(\{x_t^{1:N_t}, y_t\}_{t=1}^T\right) \\ = \; p(x_1^{1:N_1}, y_1, \dots x_t^{1:N_t}, y_t, \dots x_T^{1:N_T}, y_T) \end{aligned} \tag{4.3}$$

is shown in Fig. 4.4.

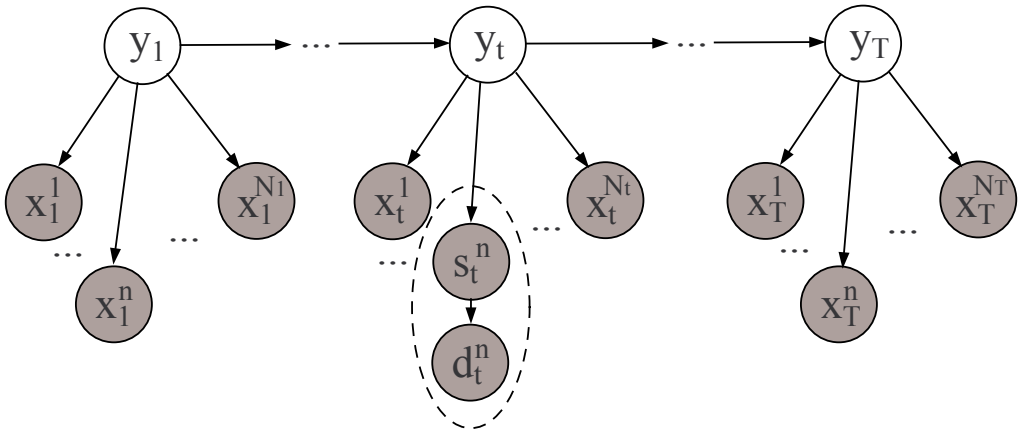


Figure 4.4: The generative model of HMM-MIO.

4.2.1.2 Scale of the observation probabilities in HMM-MIO

A side effect of introducing multiple observations into equation (4.3) is that the scale of the probability for all the observations in a frame, $p(x_t^{1:N_t})$, may vary considerably with their number, N_t. This is an undesirable effect since the number of features such as STIPs varies significantly along the frame sequence and cannot be regarded as an indicator of the reliability of the measurement process. We therefore impose that the scale of probability $p(x_t^{1:N_t})$ be the same at each frame, irrespectively of the number of the observations, by normalizing the probability as:

$$p^g(x_t^{1:N_t}|y_t) = \sqrt[N_t]{\prod_{n=1}^{N_t} p(x_t^n|y_t)}. \tag{4.4}$$

In logarithmic scale, the above normalization corresponds to the average of the observation log-probabilities:

$$p^g(x_t^{1:N_t}|y_t)) = \frac{1}{N_t}\sum_{n=1}^{N_t} \ln p(x_t^n|y_t). \tag{4.5}$$

Given that the model is evaluated in logarithmic scale during expectation-maximization, the canonical estimation algorithms can be simply updated by replacing the single ob-

servation log-probability for each frame with the average of multiple log-probabilities.

4.2.1.3 Forward and backward formulas for HMM-MIO

The forward and backward formulas for the traditional HMM have been changed to accommodate the multiple observations of HMM-MIO. Following notations in [10], the forward formula, i.e., $\alpha_i(t)$, is now changed to $\alpha_i^g(t)$:

$$\alpha_i^g(t) = p^g(x_{1:t}, y_t = i|\theta). \tag{4.6}$$

The recursion in the forward algorithm is then specified as:

$$1. \quad \alpha_i^g(1) = \pi_i b_i^g(x_1^{1:N_t}) \tag{4.7a}$$

$$2. \quad \alpha_i^g(t) = \left[\sum_{j=1}^{R} \alpha_j^g(t-1)a_{ji}\right] b_i^g(x_t^{1:N_t}) \tag{4.7b}$$

$$3. \quad p(x_{1:T}|\theta) = \sum_{i=1}^{R} \alpha_i^g(T) \tag{4.7c}$$

where a_{ij} and π_i indicate the transition probabilities between any two states, and the initial probabilities, respectively. In the above equations R refers to the number of possible hidden states. Like the forward formula, the backward algorithm is changed from the usual $\beta_i(t)$ to $\beta_i^g(t)$:

$$\beta_i^g(t) = p^g(x_{t+1:T}|y_t = i, \theta). \tag{4.8}$$

The corresponding recursion in the backward algorithm is then formulated as:

$$1. \quad \beta_i^g(T) = 1 \tag{4.9a}$$

$$2. \quad \beta_i^g(t) = \sum_{j=1}^{R} a_{ij} b_j^g(x_{t+1}^{1:N_{t+1}}) \beta_j^g(t+1) \tag{4.9b}$$

$$3. \quad p(x_{1:T}|\theta) = \sum_{i=1}^{R} \pi_i b_i^g(x_1^{1:N_1}) \beta_i^g(1) \tag{4.9c}$$

For the purpose of parameter estimation with the Baum-Welch algorithm, we replace the expression for the state posterior at time t, $\gamma_i(t)$, given in [10] with $\gamma_i^g(t)$, obtaining:

$$\gamma_i^g(t) = p^g(y_t = i|x_{1:T}, \theta) = \frac{\alpha_i^g(t)\beta_i^g(t)}{\sum_{j=1}^{R} \alpha_j^g(t)\beta_j^g(t)}. \tag{4.10}$$

However, the posterior probability for the mixture component generating an observation must still be computed individually for each observation. Therefore, the following holds:

$$\gamma_{il}(x_t^n) = p(y_t = i, C_{it}^n = l|x_{1:T}, \theta) = \gamma_i^g(t)\frac{c_{il}b_{il}(x_t^n)}{b_i(x_t^n)}. \tag{4.11}$$

where C_{it}^n is a random variable indicating the mixture component for observation x_t^n for state i, c_{il} notes the component's weight in the mixture, and $b_{il}(x_t^n)$ is the probability of observation x_t^n in the l-th mixture component for state i, $l = 1 \ldots M$.

4.2.1.4 A brief comparison with discriminative sequential models

In recent years, linear-chain conditional random fields (CRFs) have gained attention as an alternative to hidden Markov models [69]. The main advantage offered by CRFs is their discriminative training, either as a probabilistic model or in a maximum-margin framework [138]. Their accuracy has been repeatedly reported as higher than that of corresponding HMMs (e.g., [140, 150]). However, there are two standing limitations which prevent extending a conditional random field with the features of HMM-MIO. The first limitation is that a principal component framework requires a log-quadratic model (for terms of the form $w_i w_j x_i x_j$) for which standard estimation algorithms are unsuited. The second limitation is the short tails of the exponential family on which CRFs are based. Conversely, the density of the Student's *t* is not exponential and enjoys an asymptotic value of $O(x^{-\nu-1})$ that can be modulated by the degree of freedom parameter, ν, to properly account for long tails and outliers. These considerations explain why a generative model like HMM-MIO offers complementary advantages to CRF for action recognition from local features.

4.2.1.5 Experimental Results

We have evaluated the accuracy of the proposed method for the task of action classification over the KTH dataset and joint action classification and segmentation over the Stitched KTH and CMU-MMAC datasets. All of these experiments were performed on a computer with an Intel(R) Core(TM)2 Duo CPU (E8500, 3.16GHz) and 4GB RAM using Matlab R2009b.

KTH Dataset: A first experiment was performed to evaluate HMM-MIO over a task of action classification from pre-segmented actions instances. The action dataset used is the KTH dataset described in Chapter 3. Although in recent years KTH has become saturated with results reporting high accuracies, it still offers the widest comparative platform [51]. As features, we have used the STIP descriptors from Laptev *et al.* [72] with a combination of HOG and HOF components for an overall dimensionality of 145 dimensions per observation.

The experiments were conducted comparing various observation densities such as Gaussian mixture models (GMM), mixtures of probabilistic principal component analyzers (MPPCA) and mixture of t distribution subspaces [24]. For both MPPCA and the mixture of t distribution subspaces, we have carried out trials over a range of reduced dimensions ($D = \{36,18,9\}$). For the mixture of t distribution subspaces we have manually selected different values of the degrees of freedom parameter, ν, and tested with different grid sizes ($S = \{1,2,4\}$). For evaluation, we have followed the procedure proposed by Schuldt *et al.* in [124]: the KTH sequences were grouped into three sets, namely, training, validation, and test, comprising of specific actors from the dataset in the number of 8, 8, and 9, respectively. An HMM-MIO for each action class was trained on the training set, and the validation set was used to select the best number of states and mixture components. Finally, the parameters selected from the validation set were used over the test set to provide the final accuracy results.

Table 4.1 shows the results for the various combinations of observation probabilities and main parameters. The first comment is that accuracies are rather high in general, showing that HMM-MIO can utilize individual STIP descriptors as its observations despite their sparsity in space and time. The best result, 87.1%, is obtained with the mixture of t distribution subspaces over a grid with two cells. This accuracy is 7.4 percentage points higher than a standard HMM. Yet, this best accuracy is still lower

Method		Valid. accuracy (%)	Test accuracy (%)
GMM	Σ=full	87.3	79.7
	Σ=diag.	81.8	74.3
	Σ=spher.	79.7	72.9
MPPCA	*S=1*, *D=9*	84.3	76.6
Mt-ss	*S=1*, *D=9*, ν=*3*	91.2	85.7
Mt-ss	*S=2*, *D=9*, ν=*3*	90.8	87.1
Mt-ss	*S=4*, *D=9*, ν=*3*	90.5	86.9
Laptev *et al.* [72]		-	91.8
Dollár *et al.* [36]		-	81.2
Schuldt *et al.* [124]		-	71.7

Table 4.1: Accuracy (%) of HMM-MIO over the KTH dataset with different observation probabilities: GMM (full, diagonal, spherical), MPPCA and mixture of t distribution subspaces (Mt-ss). Other results are shown for comparison [36, 72, 124].

than that reported in Laptev *et al.* [72], 91.8% on the test set, with a bag-of-words approach. Yet, such results are not directly comparable as the actual set of descriptors used is different. Results with HMM-MIO are also higher than others obtained with other local spatio-temporal descriptors [36, 124]. While we cannot conclude that a sequential classifier over the dynamic within an action is preferable to a bag-of-words approach, result are interesting.

Stitched KTH Dataset: For a first experiment on joint segmentation and classification, we have created a "stitched" version of the well-known KTH dataset by simply concatenating individual action instances into sequences. Each sequence depicts a single actor in a homogeneous scenario (indoor, outdoor etc) performing a succession of 24 action instances for a total duration of approximately 2,000 frames. The actions were picked randomly, alternating between the two groups of {walking, jogging, running} and {boxing, hand-waving and hand-clapping} to emphasize action boundaries. A total of 64 such sequences were used for training and 36 for testing. The parameters selected over the training set were used unchanged for the test set.

Comparative experiments have been performed using HMM-MIO, classification of single frames and a bag-of-words approach. The number of reduced dimensions, D, and the number of components in each observation mixture, M, which were made vary

over interval (3,30). The degrees of freedom of the t distribution, ν, which were made vary over $\{3,6,9\}$ and the number of cells, S, over 1,2,4. To implement single-frame classification, we have used a version of HMM-MIO with uniform transition probabilities. This equates to classifying frames solely based on the observation model, ignoring sequentiality in decoding. Frames with no observations were arbitrarily assigned to the first class in appearance order. For bag-of-words, we have used *k*-means clustering with $N = \{128,256,512\}$ clusters for quantization and an SVM classifier with RBF kernel for classification. In the test sequences, each window of $W = 32$ frames has been assigned a single action label, sliding the window forward one frame at a time. We have also tried 16 and 64 frames for W. The accuracy is approximately on par for $W = 16$ and $W = 32$. With $W = 64$, the accuracy starts to visibly decrease, certainly due to the presence of shorter action instances in the dataset. As features, we have extracted STIPs with the public software from [72], with the default descriptors of 162 dimensions each.

Method	Parameters	Test accuracy (%)
HMM-MIO	$D = 30$, $M = 18$, $\nu = 3$, $S = 2$	71.2
Single-frame classification	$D = 30$, $M = 18$, $\nu = 3$, $S = 2$	41.8
Bag-of-words	$N = 256$, $W = 32$	61.8

Table 4.2: Frame-based accuracy (%) for joint classification and segmentation over a stitched version of the KTH dataset. S: number of cells; D: number of reduced dimensions, M: number of components per mixture, ν: degrees of freedom; N: number of clusters; W: window size.

Table 4.2 shows the results on the test set in terms of frame-based accuracy, using the parameters' values that scored the best accuracy on the training set. The highest accuracy for the three compared models was achieved by HMM-MIO (71.2%). The importance of using a sequential model for segmentation is evidenced by the comparison with single-frame classification: the drop in accuracy is almost 30 percentage points. This drop is caused by both the arbitrary classification of frames without observations and the dismissal of the sequential context. The accuracy achieved by bag-of-words (61.8%) proved more than 9 percentage points lower than that achieved by HMM-MIO. The sensitivity to the parameters' values is not very pronounced: the range of accuracies for HMM-MIO is {66.5%-71.2%}, {38.8%-41.8%} for single-frame clas-

sification, and {55.3%-61.8%} for bag-of-words.

CMU-MMAC Dataset: For a more probing and realistic experiment, we have tested our approach on a sub-set of the CMU Multi-Modal Activity Database (CMU-MMAC) containing multimodal measurements of the activity of forty subjects cooking five different recipes: brownies, pizza, sandwich, salad, and scrambled eggs [34]. For this experiment, we have selected the video clips of twelve different subjects making "brownies" from a dry mix box. The subjects attended to the preparation in a spontaneous way, without receiving instructions on how to perform each task; therefore, the action instances vary greatly in time span and manner of execution. Each video depicts a person performing a sequence of actions, with each action belonging to one of 14 classes including pouring, spraying, stirring, and others (see Fig. 4.5 for the complete list). The average duration of a video is approximately 15,000 frames while the average length of an action instance is approximately 230 frames, with a minimum length of 3 frames and a maximum of 3,269. As video source, we have used the view from static camera "7151062" which offers a side view of the scene (see Fig. 4.5). As action labels, we have used the annotations provided for the wearable camera mounted atop the subject's head, albeit only loosely synchronized with the static camera. For the experiment, we have used 12-fold cross-validation with a validation set, selecting eight subjects for training, three for validation and one for testing in each fold on a rotating basis. As features, we have again extracted STIPs with the public software from [72], but sub-sampling them one in ten in appearance order so as to limit the overall data size. The compared algorithms include HMM-MIO, single-frame classification, and the bag-of-words approach. For bag-of-words, we have extended the parameter search to {128,256,512,1024} for the number of clusters and {16,32,64} for the window size.

Method	Average accuracy	Standard deviation
HMM-MIO	38.4	6.1
Single-frame classification	11.7	1.6
Bag-of-words	35.2	2.3

Table 4.3: Frame-based accuracy (%) for joint classification and segmentation over a sub-set of the CMU-MMAC dataset.

Table 4.3 shows the results from this experiment in terms of average accuracy and

Figure 4.5: Examples of actions for preparation of "brownies": (from left to right, column wise) *close, crack, none, open, pour, put, read, spray, stir, switch-on, take, twist-off, twist-on and walk.*

standard deviation over the folds. As it is far more realistic and challenging, accuracies are generally much lower. The best average accuracy is achieved by HMM-MIO (38.4%) and is noticeably higher than that of bag-of-words (35.2%). However, HMM-MIO is more sensitive to the training fold as it reports a much higher standard deviation (6.1 vs. 2.3). The drop in accuracy with single-frame classification (minus 26.7 percentage points from HMM-MIO) is proportionally more marked than in the previous experiment, giving evidence about the importance of the sequential structure at a parity of observation model.

From these two experiments, we can conclude that a sequential classifier can outperform Window-based bag-of-words approaches in joined the task of segmentation and classification.

4.2.2 Feature Fusion

We now present an extention of HMM-MIO for the fusion of structural and spatio-temporal features. For this reason, we first define the feature set used in our framework. The proposed approach is then described in the next section. We finally present an experimental evaluation of the proposed method on the KTH action dataset.

4.2.2.1 Features

The structural and spatio-temporal features provided by graph embedding and typical descriptors are:

- Structural features: We have used the 16-dimensional feature vectors described in Chapter 3.
- Spatio-temporal features: We have used a combination of HOG and HOF for an overall dimensionality of 145 (STIP descriptors [70]).

4.2.2.2 Fusion graphical model

As classifier, we have used the hidden Markov model for multiple, irregular observation (HMM-MIO, described in section 4.2.1.1) capable of dealing with sequences of observations that include outlier, high-dimensional, and sparse measurement typical of action recognition with S = 1. In the proposed model for feature fusion, the probability for all the observations in a frame, t, is calculated by the fusion of two likelihoods which model two types of measures:

- Spatio-Temporal Texture or Appearance Observations ($x_{a,t}$) provided by the STIP descriptors: the different numbers of STIP points per frame introduced a scale problem in the resulting probability that is solved in HMM-MIO by means of equation 4.4.
- Structural Observations ($x_{s,t}$) provided by graph embedding: In our experiments, the embedding of a graph with 16 different selected prototypes leads to a 16-dimensional feature vector describing the shape of a single actor in each frame. This feature vector is modelled statistically by likelihood $P(x_{s,t}|y_t)$.

The combination of the two likelihoods (equation 4.12) is performed as a weighted sum of weights W_a and W_s, such that $W_a + W_s = 1$.

$$P(x_t|y_t) = W_a \cdot P_a(x_{a,t}^{1:N_t}|y_t) + W_s \cdot P(x_{s,t}|y_t) \tag{4.12}$$

The graphical model for the modified HMM-MIO can be seen in Figure 4.6. The generative model is then obtained using equation 4.3.

4.2.2.3 Experimental Results

This section provides the evaluation of the proposed approach and shows the advantages of combining the structural information provided by graph embedding with the

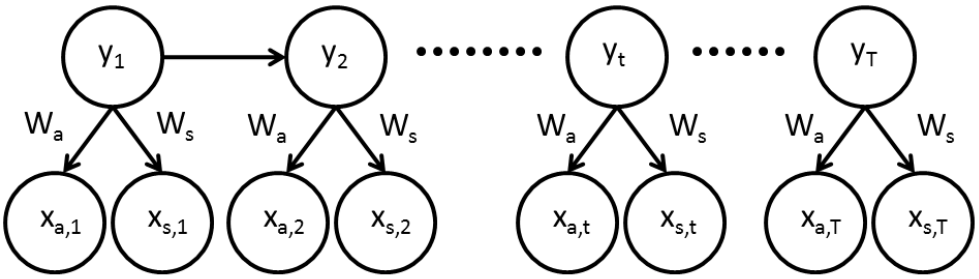


Figure 4.6: Modified HMM-MIO (hidden Markov model with multiple, independent observations); x_t are the observations at time t (appearance observations provided by the STIP descriptors, x_a, and the structural observation provided by graph embedding, x_s); y_t is the corresponding hidden state; W_a and W_s are the two weights for computing the total observation probability $P(x_t|y_t) = W_a \cdot P_a(x_{a,t}^{1:N_t}|y_t) + W_s \cdot P(x_{s,t}|y_t)$; $W_a + W_s = 1$.

spatio-temporal information provided by STIPs. In order to assess the individual contribution of the features and show the advantages of the proposed fusion, we have conducted experiments with different weights (Table 4.4). A value of (W_a, W_s)=(1,0) means that only appearance features are used, whereas structural features are solely utilised when (W_a, W_s)=(0,1). As shown by Table 4.4, recognition accuracy is significantly improved by combining the structural information with the spatio-temporal features, reaching its maximum when (W_a, W_s)=(0.5,0.5).

4.3 Discussion and Conclusions

In this chapter, we have first presented an approach to joint action segmentation and classification based on an extended HMM capable of exploiting local spatio-temporal features. Such measurements are irregular in space and time, high dimensional and characterized by heavy-tailed distributions and outliers. The extended model, HMM-MIO (hidden Markov model with multiple, irregular observations), effectively tackles these issues and provides significant accuracy. The results show that sequential generative classifiers can be capable of significant action recognition accuracy, provided they are endowed with likelihood models that are well suited to typical visual measurements.

Table 4.4: Accuracy (%) of our approach over the KTH dataset with variable weights over the appearance and structural components.

W_a	W_s	Test accuracy (%)
1.0	0.0	*85.7* [28]
0.9	0.1	85.9
0.8	0.2	86.8
0.7	0.3	87.9
0.6	0.4	88.9
0.5	**0.5**	**89.8**
0.4	0.6	87.9
0.3	0.7	85.8
0.2	0.8	82.2
0.1	0.9	77.9
0.0	1.0	*48.7*

We have then used this model to present a novel approach for human action recognition based on the fusion of structural and spatio-temporal information. To this aim, the structural information provided by graph embedding and the local spatio-temporal information provided by STIP descriptors are jointly modelled by a modified hidden Markov model with multiple, independent observations (HMM-MIO) [28]. Although our approach does not yet equal state-of-the-art accuracy, it shows that structural and spatio-temporal features can be fused constructively to obtain higher accuracy than from either separately.

Chapter 5

Discriminative Prototype Selection

As we have seen in the previous chapters, graphs have shown a remarkable representational power for action recognition. In particular, the discriminative prototype selection we proposed in section 3.2.3 is a major innovation over conventional selection techniques. Motivated by this rationale, in this chapter, we introduce a novel framework for selecting a set of prototypes from a labelled graph set taking their discriminative power into account and test it over a variety of structured data including letter, digit, molecular, fingerprint. Experimental results show that such a discriminative prototype selection framework can achieve superior results in classification compared to other well-established prototype selection approaches.

5.1 Prior Work and Our Contributions

Although the vast majority of pattern recognition algorithms rely on vectorial data representations, more and more effort is now rendered in various research fields on graph based representations [29]. Unlike the vectorial representation which ignores the dependencies between observations, graphs preserve these dependencies and relations. To phrase it more generally, the main merits of a graph-based representation are:

- the number of nodes and edges in the graph is not fixed a priori; rather, it adjusts to the complexity of the target object;
- graphs are capable of encapsulating the object's structure not merely by storing

the object's features, but by also explicitly modeling the relations amongst such features (beyond simple co-statistics).

Leveraging on these appealing properties, many approaches have used graphs in, for instance, human action recognition [16], bioinformatics and chemoinformatics [85, 107, 144], web content analysis and data mining [30, 120, 121], classifying images from various fields [6, 55, 84], symbol and character recognition [81, 118, 132] and computer network analysis [18, 35].

However, object representations given in terms of graphs suffer from a number of severe drawbacks when compared to feature vectors. One major limitation is the significantly increased complexity of many algorithms. For instance, the comparison of two feature vectors for identity can be accomplished in linear time with respect to the length of the two vectors. For the equivalent operation on graphs, i.e. testing two graphs for isomorphism, only exponential algorithms are known to date. Another major drawback of graph-based representations is that even basic mathematical operations such as sums and products cannot be performed on graphs, making them unsuited for conventional pattern recognition approaches. As a consequence of these general limitations, the lack of algorithmic tools for graph-based pattern recognition appears obvious.

The way we have chosen to circumvent this problem is to embed the graphs in a real vector space. By this approach, we can benefit from the wide range of statistical pattern recognition methods while retaining the universality of graphs for pattern representation. To date, many approaches have been proposed in the literature to embed graphs in a vector space. In [84], for instance, features derived from the eigendecomposition of graphs are exploited. Another approach uses an "edit distance" to compute the matrix of distances between any two graphs in a set and then uses it to embed the graphs into a vector space by means of multidimensional scaling [156]. In [157], the authors turn to the spectral decomposition of the Laplacian matrix. They show how the elements of the spectral matrix for the Laplacian can be used to construct symmetric polynomials. In order to encode a graph as a vector, the coefficients of these polynomials are used as graph features. Another approach for graph embedding has been proposed in [117]; the authors use the relationship between the LaplaceBeltrami operator and the graph Laplacian to embed a graph onto a Riemannian manifold.

The present work follows another approach of graph embedding where a graph is embedded into a point of the vector space by means of a template set and a dissimilarity measure. This approach is primarily based on the idea proposed in [100, 101] where a dissimilarity representation over vectors was first introduced, and then extended in [131] to map strings onto vector spaces and finally generalized to graphs in [19, 112]. In the literature, graph embedding by means of prototype selection has reported higher classification accuracy than alternative methods such as K-NN classification directly in the graph domain and SVM classification over similarity kernels [19, 20]. The key idea of this graph embedding approach is to convert a graph into an n-dimensional feature vector by way of a set of "prototype" graphs P and a dissimilarity measure such that the feature vector consists of the dissimilarity between the graph and each prototype. Intuitively, the prototype set should be distributed over the graph domain in a uniform way. However, in principle this is difficult to ensure since uniformity over a graph domain is a vague concept.

Let us assume to be given a training set, $\mathbf{C}$, of class-labelled graphs from N different classes, $C_1, \ldots, C_N$. Various approaches have been proposed to date for selecting informative prototypes from $\mathbf{C}$. In [19], all available elements from the training set are used as prototypes $P = \mathbf{C}$, and then feature extraction algorithms, e.g. *principal component analysis* (PCA) [61], is applied to the embedded graphs in vector space. Although by this approach the authors bypass the difficult problem of selecting adequate prototypes, it is obvious that it may prove computationally too expensive for large datasets and that its run-time cost on a new graph may be too high. To overcome this limitation, in [115], the authors proposed different heuristic approaches based on the distances between the graphs in $\mathbf{C}$. The authors distinguish between *unlabelled* and *labelled* selection. The unlabelled selection is executed over the whole training set at once ignoring the class labels, while the labelled selection selects prototypes separately for each of the N classes, $C_1, \ldots, C_N$. In general, labelled approaches have reported higher classification accuracy than unlabelled methods.

Labelled prototype selection can be likened to the training of class likelihoods in generative classifiers, where each likelihood is estimated based on only the samples from that class. Conversely, discriminative classifiers choose parameters based on the information from multiple classes at once, maximizing objective functions such as the class margin, Fisher discriminants, mutual information and others, and often prov-

ing more accurate than their generative counterparts. Inspired by discriminative approaches, in this chapter we propose various *discriminative prototype selection* methods where the prototype set is chosen by weighing intra-class compactness and inter-class separation and demonstrate their ability to outperform previous methods. We have also recently become aware of another proposal for discriminative prototype selection from Raveaux *et al.* [109]. In [109], the authors propose to conduct the search for prototypes in the exponential space of possible selections by way of a genetic algorithm. While this is certainly an attractive strategy, we believe that the tradeoffs we propose between intra-class compactness and inter-class separation offer a more immediate interpretation.

5.2 Proposed Methods

In this section, we describe the proposed approaches.

5.2.1 Prototype selection

Selecting informative prototypes from the underlying graph domain plays a vital role in graph embedding [151]. In order to obtain a meaningful as well as class-discriminative vector representation in the embedding space, a set of selected prototypes $P = \{p_1, p_2, \ldots, p_n\}$ should be adequately distributed over the whole graph domain, at the same time avoiding redundancies in terms of selection of similar graphs [56, 100, 112].

Let us assume that the graphs in the graph domain can be classified into N different classes, $c_1, \ldots, c_N$. Given a labelled training set, $\mathbf{C}$, we note as $C_1, \ldots, C_N$ the N subsets spanned by the classes, such that $\mathbf{C} = \bigcup_{n=1}^{N} C_n$. We then categorise the prototype selection methods into labelled and unlabelled approaches. In the former, the selection is performed individually for each class, while the latter determines all prototypes from the whole training set, $\mathbf{C}$, ignoring the class label information. As shown in [115], labelled selection approaches tend to deliver higher classification accuracy than corresponding unlabelled approaches. Yet, existing labelled selection methods choose the class' prototypes based solely on the graphs in the class. In this thesis, we instead propose *discriminative* approaches for the selection of prototypes which consider graphs from all classes at once maximising a function of the inter- and intra-class distances. In

this way, we elicit prototype selection strategies imposing that the selected prototypes for the class be well-distributed within the class, yet being discriminative with respect to the graphs in the remaining classes.

5.2.2 Learning discriminative prototypes

The ultimate goal of prototype selection for classification is to identify the most discriminative graphs in the training set , **C**. In our work, each of the selection algorithms in section 5.2.3 is learned in two different ways. If the prototypes are chosen for a class to discriminate well against all other classes, they form a *one-vs-all* prototype set. Similarly, if the selection strategy tries to maximize this discrimination between the selected class and the closest class, it obtains a *one-vs-nearest* prototype set.

Let $C_n = \{g_{n1}, \ldots, g_{ni}, \ldots, g_{n|C_n|}\}$ and $C_m = \{g_{m1}, \ldots, g_{mj}, \ldots, g_{m|C_m|}\}$ be the subsets of **C** for classes c_n and c_m, respectively. We adopt the following definition for the class distance between c_n and c_m:

$$d_{class}(c_n, c_m) = \frac{\sum_{i=1}^{|C_n|} \sum_{j=1}^{|C_m|} d(g_{ni}, g_{mj})}{|C_n||C_m|} \tag{5.1}$$

where $d(u, v)$ is the distance between graphs u, v and $|C_n|$ and $|C_m|$ are the total number of graphs in C_n and C_m, respectively. Alternative definitions are also possible, but they are not the focus of the following work.

Based on equation 5.1, the nearest class $c_{n_{near}}$ to class c_n is the class which has the minimum class distance to c_n, formally defined as:

$$c_{n_{near}} = \underset{\overline{n}=1\ldots N, \overline{n}\neq n}{\operatorname{argmin}} \; d_{class}(c_n, c_{\overline{n}}). \tag{5.2}$$

5.2.3 Discriminative prototype selection algorithms

Based on the graph embedding definition, an appropriate choice of the prototype set, P, plays a critical role in graph embedding as it impacts the classification accuracy. The six deterministic algorithms used to select the discriminative prototypes in this work are described below. In the selection of these discriminative prototypes, different objective functions are proposed which not only provide high intra-class compactness,

but also consider inter-class separation. The part influencing the intra-class compactness is weighted by a weight, W_c, and the part controlling the inter-class separation is weighted by W_s where $\{W_c, W_s\} \in [0, 1]$ and $W_c + W_s = 1$. Each of these algorithms is an extension of an existing labeled algorithm. All these objective functions allow selecting an arbitrary number, K, of prototypes from each class.

5.2.3.1 Discriminative Center Prototype Selection

In *discriminative center prototype selection* (d-cps), a prototype set, $P_n = P_{n(1:K)} = \{p_{n1}, \ldots, p_{nk}, \ldots, p_{nK}\}$, is generated from each C_n subset, with each p_{nk} prototype simultaneously located near the "center" of the graphs from C_n, and away from the graphs of the remaining classes, $\overline{C_n}$. Prototypes are selected incrementally, with each prototype p_{nk} determined as a graph $g_{nj} \in C_n$ which is not already selected as prototype and such that the difference between the sum of distances between g_{nj} and all other graphs in its class, excluding the already selected prototypes, and the sum of distances between g_{nj} and all other graphs in $\overline{C_n}$ is minimal:

$$p_{nk} = \underset{g_{nj} \in C_n, g_{nj} \notin P_{n(1:k-1)}}{\operatorname{argmin}} \left[W_c \cdot \sum_{\substack{g_{ni} \in C_n, i \neq j, \\ g_{ni} \notin P_{n(1:k-1)}}} d(g_{nj}, g_{ni}) - W_s \cdot \sum_{g_{\overline{n}i} \in \overline{C_n}} d(g_{nj}, g_{\overline{n}i}) \right] \tag{5.3}$$

This objective function promotes class discrimination. However, it may suffer from redundancy as it tends to select multiple prototypes from the center of the class. Moreover, it should be noted that because the number of graphs in C_n is usually much lower than that in $\overline{C_n}$, the objective function in (5.3) usually takes negative values. However, this has no impact on the minimisation.

5.2.3.2 Discriminative Border Prototype Selection

The idea of *discriminative border prototype selection* (d-bps) is to choose the prototype set, P_n, such that each p_{nk} prototype be situated near the border of its class. The rationale for this selection is that of having prototypes which are simultaneously mutually spread apart and distant from the graphs in the other classes. Prototypes are

selected incrementally, with each prototype p_{nk} determined as a graph $g_{nj} \in C_n$ which is not already selected as prototypes and such that the total sum of the sum of distances between g_{nj} and all other graphs in C_n, excluding the already selected prototype, and $\overline{C_n}$ is maximal:

$$p_{nk} = \underset{g_{nj} \in C_n, g_{nj} \notin P_{n(1:k-1)}}{\operatorname{argmax}} [\, W_c \cdot \sum_{\substack{g_{ni} \in C_n, i \neq j, \\ g_{ni} \notin P_{n(1:k-1)}}} d(g_{nj}, g_{ni}) + W_s \cdot \sum_{g_{\overline{n}i} \in \overline{C_n}} d(g_{nj}, g_{\overline{n}i}) \,] \tag{5.4}$$

In contrast to the previous prototype selector, where many prototypes could be structurally similar, this selection procedure prevents redundancy. However, it lacks prototypes from the inner region of the class and this may lead to poorly discriminative embedded vectors for graphs located in such regions.

5.2.3.3 Discriminative Repelling Prototype Selection

To overcome the inherent limitations of both previous approaches, *discriminative repelling prototype selection* (d-rps) chooses the set of prototypes of each C_n sub-set based on the following procedure: the first prototype, p_{n1}, is selected by means of d-cps. To select any additional prototype, $p_{nk}, k = 2 \ldots K$, we pick a graph g_{nj} from the class' graphs not already selected as prototypes to minimize the following equation:

$$\begin{aligned} p_{nk} = \underset{g_{nj} \in C_n, g_{nj} \notin P_{n(1:k-1)}}{\operatorname{argmin}} & [\, W_c \cdot \sum_{\substack{g_{ni} \in C_n, i \neq j, \\ g_{ni} \notin P_{n(1:k-1)}}} d(g_{nj}, g_{ni}) \; - \\ & W_s \cdot (\sum_{g_{\overline{n}i} \in \overline{C_n}} d(g_{nj}, g_{\overline{n}i}) \cdot \sum_{g_{ni} \in P_{n(1:k-1)}} d(g_{nj}, g_{ni})) \,] \end{aligned} \tag{5.5}$$

This objective function is similar to that in (5.3), but encourages p_{nk} to also be distant from all previously selected prototypes, $P_{n(1:k-1)} = \{p_{n1}, \ldots, p_{n(k-1)}\}$ ("repelling" component). This favors mutual separation amongst the class' prototypes

and their more uniform distribution within the class.

5.2.3.4 Discriminative Spanning Prototype Selection

Along a similar rationale, *discriminative spanning prototype selection* (d-sps) selects each prototype with the following iterative procedure: the first prototype, p_{n1}, is selected by d-cps. Each additional prototype, $p_{nk}, k = 2 \dots K$, is the graph in C_n that preserves the following conditions: be the farthest graph from the already selected prototypes, $P_{n(1:k-1)}$, as well as all graphs in the other classes, $\overline{C_n}$:

$$\begin{aligned} p_{nk} = & \underset{g_{nj} \in C_n, g_{nj} \notin P_{n(1:k-1)}}{\operatorname{argmax}} \\ & [W_c \cdot \sum_{g_{ni} \in P_{n(1:k-1)}} d(g_{nj}, g_{ni}) + W_s \cdot \sum_{g_{\overline{n}i} \in \overline{C_n}} d(g_{nj}, g_{\overline{n}i})] \end{aligned} \tag{5.6}$$

Compared to (5.5), this objective function ignores the compactness term and composes the other two terms in an additive rather than multiplicative scale.

5.2.3.5 Discriminative Targetsphere Prototype Selection

In *discriminative targetsphere prototype selection* (d-tps), the first and second prototypes, $\{p_{n1}$,$p_{n2}\}$, for C_n are selected by means of d-cps and d-bps, respectively. These two prototypes represent the center and farthest boundary of the class. Then, the distance between these two prototypes, $d_{max} = d(p_{n1}, p_{n2})$, is computed and each other prototype, $p_{nk}, k = 3 \dots K$, is selected as the graph closest to a distance of $(k-2)d_{max}/(K-1)$ from p_{n1} and furthest away from the graphs in the other classes, $\overline{C_n}$. This procedure is called "targetsphere" as it is reminiscent of the evenly-spaced divisions of a shooting target circle:

$$\begin{aligned} p_{nk} = & \underset{g_{nj} \in C_n, g_{nj} \notin P_{n(1:k-1)}}{\operatorname{argmin}} \\ & [W_c \cdot \left| d(g_{nj}, p_{n1}) - (k-2) \cdot \frac{d_{max}}{(K-1)} \right| - W_s \cdot \sum_{g_{\overline{n}i} \in \overline{C_n}} d(g_{nj}, g_{\overline{n}i})] \end{aligned} \tag{5.7}$$

5.2.3.6 Discriminative k-Center Prototype Selection

They key idea of *discriminative k-center prototype selection* (d-$k$cps) is to select the prototypes of each class by a procedure similar to k-medoids clustering, at the same time maintaining separation from the graphs in the remaining classes, $\overline{C_n}$ [63]. The six steps of this method are:

1. Select an initial set of K prototypes, $P_{n_{initial}} = \{p_{n1}, \ldots, p_{nk}, \ldots, p_{nK}\}$, by means of any of the previous prototype selectors.

2. Construct K sets, with each set containing one of the initial prototypes: $S_1 = \{p_{n1}\}, \ldots, S_k = \{p_{nk}\}, \ldots, S_K = \{p_{nK}\}$.

3. For each other graph $g \in C_n$, $g \notin P_{n_{initial}}$, find its nearest prototype in terms of a distance between elements and add g to the corresponding set. As a result of this, we attain a partition on C_n with K disjoint subsets and $C_n = \bigcup_{k=1}^{K} S_k$.

4. For each set S_k, find its center graph c_k by means of d-cps. This retains the discriminative aspect of the selection.

5. For each set S_k, if its center c_k is not equal to prototype p_{nk}, replace p_{nk} by c_k.

6. If any replacement has occurred, return to step 2; otherwise, select the centers of the K disjoint sets, $\{c_1, \ldots, c_K\}$, as the set of prototypes, P_n.

5.3 Experimental Results

This section provides the evaluation of the proposed methods and shows the benefits of using discriminative prototype selection approaches compared to existing methods. To this aim, several classification tasks are carried out over a wide number and variety of datasets including letters, digits, drawings, fingerprints and more.

5.3.1 Dataset

For extensive testing of the proposed approaches, we have chosen a total of 10 different graph datasets from the publicly available IAM graph database repository for

graph based pattern recognition and machine learning [113]. Databases from the IAM repository have been a de-facto benchmark in the last few years and many results are available for comparison. A summary of these graph data sets together with some characteristic properties is reported in Table 5.1. For the sake of accuracy evaluation, each of the datasets used in the chapter is divided into three disjoint subsets which are used for training, validation and testing. Each of these datasets is presented in greater detail in the following subsections.

Dataset	Size(tr, va, te)	# classes	$\varnothing\,\lvert V\rvert$	$\varnothing\,\lvert E\rvert$	max$\lvert V\rvert$	max$\lvert E\rvert$	Balanced
Letter low	750, 750, 750	15	4.7	3.1	8	6	Y
Letter medium	750, 750, 750	15	4.7	3.2	9	7	Y
Letter high	750, 750, 750	15	4.7	4.5	9	9	Y
Digit	1000, 500, 2000	10	8.9	7.9	17	16	Y
GREC	836, 836, 1628	22	11.5	12.2	25	30	Y
Fingerprint	500, 300, 2000	4	5.4	4.4	26	24	N
AIDS	250, 250, 1500	2	15.7	16.2	95	103	N
Mutagenicity	500, 500, 1000	2	30.3	30.8	417	112	Y
Protein	200, 200, 200	6	32.6	62.1	126	149	Y
Webpage	780, 780, 780	20	186.1	104.6	834	596	N

Table 5.1: Summary of graph data set characteristics, e.g. the size of the training (tr), the validation (va) and the test set (te), the number of classes (# classes), the average and max number of nodes and edges ($\varnothing\,\lvert V\rvert$, max$\lvert V\rvert$, $\varnothing\,\lvert E\rvert$, max$\lvert E\rvert$), and whether the graphs are uniformly distributed over the classes or not (balanced).

5.3.1.1 Letter datasets

Each graph in the letter dataset represents a distorted letter drawing. This dataset considers the 15 capital letters of the Roman alphabet which consist of straight lines only (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z) to build 15 different classes. For each class, a prototype line drawing is manually constructed. These prototype drawings are then converted into prototype graphs by representing lines as undirected edges and ending points of lines as nodes. Each node is labeled with a two-dimensional attribute giving its position relative to a reference coordinate system. Edges are undirected and unlabeled. In order to test classification under different conditions, three sets of this dataset are obtained by applying three levels of distortion (namely *low*, *medium* and *high*) on each original graph. The training, validation and test sets are of size 750 each. The graphs are uniformly distributed over the 15 classes. Figure 5.1 shows the original graph and a graph instance for each distortion level representing letter “A”.

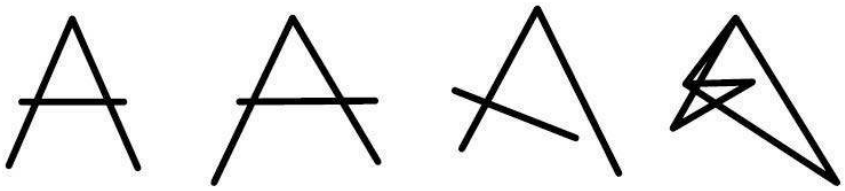


Figure 5.1: Examples of letter A: Original and distortion levels low, medium and high (from left to right)

5.3.1.2 Digit dataset

This dataset contains graphs representing different handwritten digits [48]. In each graph, nodes represent line segments of a handwritten digit. More formally, the sequences of (x, y) coordinates are converted into graphs by grouping coordinate points forming sub-paths of similar length. These sub-paths are represented by nodes labeled with their starting and ending position relative to a reference coordinate system (i.e. the first and last (x, y) coordinates from the respective sub-path). Successive sub-paths are connected by undirected and unlabeled edges. Finally, the derived graphs are normalized such that each corresponding digit has equal width and height.

The dataset used in this work comprises a randomly selected sub-set of totally 3500 digits (1000, 500 and 2000 samples used for training, validation and testing respectively) which are uniformly distributed over the 10 classes. Figure 5.2 illustrates a graph instance of each of the ten digit classes.

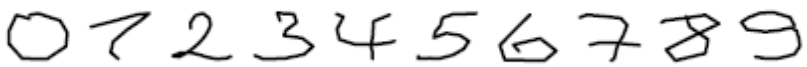


Figure 5.2: A graph example of each of the ten digit classes

5.3.1.3 GREC dataset

The GREC dataset consists of graphs representing symbols from 22 classes of architectural and electronic drawings [38]. The image can occur at five distortion levels.

An example of each distortion level can be seen in Figure 5.3. Based on the distortion level, a morphological operation (e.g. erosion, dilation) is applied and the result is thinned to get lines of one pixel width. Finally, graphs are extracted from the resulting denoised images by tracing the lines from end to end and detecting intersections as well as corners. Ending points, corners, intersections and circles are represented by nodes and labeled with a two-dimensional attribute giving their position. The nodes are connected by undirected edges which are labeled as *line* or *arc*. An additional attribute specifies the angle with respect to the horizontal direction or the diameter in case of arcs. For an adequately sized set, the five graphs per distortion level are individually distorted 30 times to obtain a dataset containing 3300 graphs uniformly distributed over the 22 classes. The resulting set is split into a training, a validation and a test set of size 836, 836 and 1628 samples respectively.

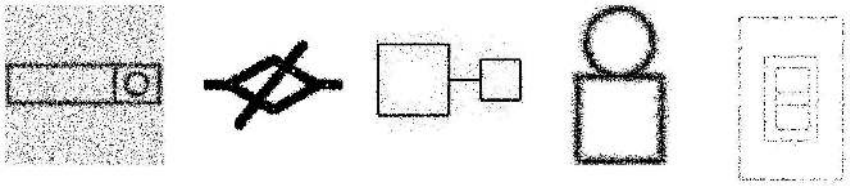


Figure 5.3: An instance image of each distortion level

5.3.1.4 Fingerprint dataset

The graphs in the Fingerprint dataset are built by binarizing the relevant regions and then applying a noise removal and thinning procedure. Thus, a skeletonized representation of the interest regions is extracted. Ending points and bifurcation points of the skeletonized regions are represented by nodes. Additional nodes are inserted of regular intervals between ending points and bifurcation points. Finally, undirected edges are inserted to link nodes that are directly connected through a ridge in the skeleton. Each node is labeled with a two-dimensional attribute giving its position. The edges are attributed with an angle denoting the orientation of the edge with respect to the horizontal direction.

The dataset used in this work is based on the NIST-4 reference database of fingerprints [152] and it consists of 2800 fingerprint images (500 samples for training, 300

samples for validation and 2000 samples for testing) distributed over four different classes *arch*, *left*, *right* and *whorl* from the Galton-Henry classification system. Note that in this dataset only a four-class problem of fingerprint classification is considered, i.e. the fifth class *tented arch* is merged with the class arch. Thus, class arch consists of about twice as many graphs as the other three classes. Figure 5.4 shows an instance of each of the four fingerprint classes.

Figure 5.4: Instances of fingerprint classes: left, right, arch and whorl (from left to right)

5.3.1.5 AIDS data set

The AIDS dataset is based on the "AIDS Antiviral Screen Database of Active Compounds" [39]. This data set consists of two classes (active, inactive), which represent molecules with activity against HIV or not. The molecules are converted into graphs in a straightforward manner by representing atoms as nodes and the covalent bonds as edges. Nodes are labeled with the number of the corresponding chemical symbol and edges by the valence of the linkage. Figure 5.5 represents one molecular compound of both classes where the shades of gray show various chemical symbols, i.e. node labels. The dataset used in this work consists of a total number of 2000 samples (1600 inactive elements and 400 active elements) where the samples are split into a training and a validation set of size 250 each and a test set of size 1500.

5.3.1.6 Mutagenicity dataset

The mutagenicity dataset follows the same approach used in the AIDS dataset to convert molecular compounds into attributed graphs [64]. It contains two classes *mutagen*

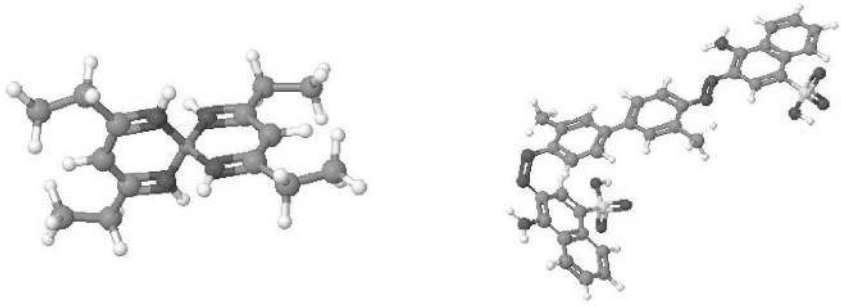


Figure 5.5: A molecular compound of both classes: active and inactive (from left to right)

and *nonmutagen* with 1250 elements each. All elements are split into three sets: training, validation and test set with the size of 250, 250 and 1500 respectively.

5.3.1.7 Protein dataset

The protein data set consists of graphs representing proteins [144]. The graphs are constructed from the Protein Data Bank [9] and labeled with their corresponding enzyme class labels from the BRENDA enzyme database [123]. Based on the six enzyme commission top level hierarchy (EC classes), this dataset consists of six classes (EC1, EC2, EC3, EC4, EC5, EC6) to represent proteins from various EC classes. Figure 5.6 illustrates an example protein of each class.

The proteins are converted into graphs by representing the structure, the sequence, and chemical properties of a protein by nodes and edges. Nodes represent secondary structure elements (SSE) within the protein structure, labeled with their type (helix, sheet, or loop) and their amino acid sequence. Every pair of nodes is connected by an edge if they are neighbors along the amino acid sequence (sequential edges) or if they are neighbors in space within the protein structure (structural edges). Every node is connected to its three nearest spatial neighbors. In case of sequential relationships, the edges are labeled with their length in amino acids, while in case of structural edges a distance measure in Angstroms is used as a label. The total number of 600 proteins are evenly distributed on six classes to build a training, a validation and a test set of size 200 each.

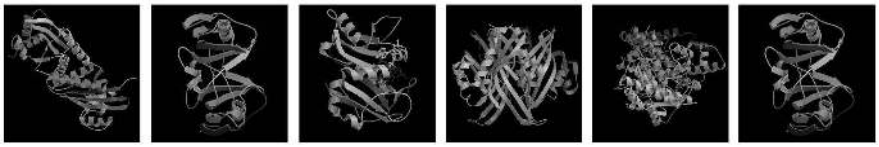


Figure 5.6: An example of each class: EC1, EC2, EC3, EC4, EC5 and EC6 (from left to right)

5.3.1.8 Webpage dataset

Amongst the various approaches for building graphs from web documents, the following procedure is applied to build graphs for the webpage dataset [120]. First, all words occuring in the web document (excluding stop words) are converted into nodes in the resulting web graph. Each node is then attributed with its corresponding word and frequency in the document. Next, the document is divided in the following sections: *title* which contains the text related to the document's title; *link* which is text in any clickable hyperlink; and *text* which comprises all the ordinary text. If a pair of words, w_i and w_{i+1}, are consecutive words in any of the above sections, a directed edge from the node corresponding to word w_i to the node of word w_{i+1} is inserted in the web graph. These edges are labeled with the corresponding section label. Finally, only the most frequently used words (nodes) are kept in the graph and the terms are conflated to the most frequently occurring forms.

The dataset used in this work consists of 2340 documents from 20 categories (*Business, Health, Politics, Sports, Technology, Entertainment, Art, Cable, Culture, Film, Industry, Media, Multimedia, Music, Online, People, Review, Stage, Television,* and *Variety*).These documents were originally hosted at Yahoo as news pages (http://www.yahoo.com) and their number varies from only 23 (Art) up to 500 (Health). The dataset is split into a training, a validation and a test set of equal size (780).

5.3.2 Comparison between the discriminative and labeled approaches

The aim of the evaluation described in this section is to empirically verify the power and applicability of the feature vectors extracted by the discriminative prototype selection approaches compared to those obtained by other methods, e.g. [115]. For the sake of comparison, the following settings are identically applied in all experiments.

For graph embedding, the graph edit distance is computed by means of the suboptimal algorithms introduced in [114]. This approach shows superior performance in time and accuracy compared to other suboptimal algorithms. The classification task of the vector space embedded graphs is carried out by employing the support vector machine [141]. Although any other statistical classifier could be used for this purpose, SVM has a theoretical advantage as well as a remarkable empirical performance [46]. In our experiments, we make use of an SVM with the Radial Basis Function (RBF) kernel [126]. In [112], this kernel was reported more accurate than *linear* and *polynomial* kernels for classifying graphs embedded in a vector space. The number of prototypes per class, n, and the SVM parameters are tuned using a training and a validation set, and the accuracy on the test set is then measured "blindly" by using the parameters selected on the validation set, without any further tuning. All our experiments were performed on a personal computer with an Intel(R) Core(TM)2 Duo CPU (E8500, 3.16GHz) and 4GB RAM using Matlab R2009b. As software, we have used the LIBSVM toolbox for Matlab [22].

In order to assess the individual contribution of the two learning approaches described in section 5.2.2, we have first conducted experiments with feature vectors extracted with different prototype selection methods, learned with one-vs-all and one-vs-nearest approaches (Table 5.2). All results for each dataset are then compared and the best accuracy per dataset is displayed in bold face. According to Table 5.2, 16 out of the top 27 prototype selectors were obtained with the one-vs-all approach rather than the one-vs-nearest. In most cases, the differences are very limited.

We compare the proposed discriminative approaches, existing labeled prototype selectors and, as a term of reference/baseline approach, using all the graphs in the training set as prototypes (Table 5.4).

We first evaluate the discriminative selection approaches (d-cps, d-bps, d-sps, d-tps and d-kcps) in comparison with corresponding labeled prototype selectors (l-cps, l-bps, l-sps, l-tps and l-kcps) [20, 115]. These labeled prototype selectors are defined as 3.2.1-6 with weights $W_c = 1$ and $W_s = 0$. In other word, each of the labeled approaches is equivalent to the corresponding discriminative approach without the inter-class term in its objective function. Table 5.3 shows that the discriminative selection strategy has increased the classification accuracies in 42 out of 50 cases over all datasets.

Next, we report the full accuracy over the various selectors and datasets in Table

	One-Vs-All						One-Vs-Nearest					
Dataset	d-cps	d-bps	d-drps	d-sps	d-tps	d-kcps	d-cps	d-bps	d-rps	d-sps	d-tps	d-kcps
Letter low	99.5	99.5	99.5	99.5	99.5	99.5	99.4	**99.6**	99.4	99.5	99.5	99.4
Letter medium	94.4	**95.6**	94.0	95.4	95.4	95.4	95.4	95.4	95.2	95.2	95.4	95.0
Letter high	92.2	92.8	93.0	**93.4**	93.0	92.8	92.6	92.3	91.8	92.7	92.3	92.8
Digit	98.6	98.6	98.6	**98.7**	98.5	98.6	98.5	98.6	98.4	**98.7**	98.6	98.6
GREC	92.0	92.0	92.0	**92.5**	92.0	92.2	91.9	92.1	92.1	92.1	92.2	92.2
Fingerprint	81.2	80.6	81.1	**81.6**	80.9	81.4	81.2	81.0	**81.6**	**81.6**	80.8	81.5
AIDS	98.0	98.0	98.0	**98.2**	**98.2**	98.1	98.0	98.0	98.0	**98.2**	**98.2**	98.1
Mutagenicity	71.1	71.1	69.9	**71.5**	71.1	70.6	71.1	71.1	69.9	**71.5**	71.1	70.6
Protein	**75.0**	72.0	72.0	73.0	**75.0**	61.0	**75.0**	72.0	72.0	73.0	**75.0**	62.0
Webpage	**82.4**	**82.4**	**82.4**	**82.4**	**82.4**	**82.4**	82.3	82.3	82.3	82.3	**82.4**	**82.4**

Table 5.2: Classification accuracy (%) of SVM-RBF applied to graphs embedded using different learning approaches (One-Vs-All and One-Vs-Nearest). The best result per dataset is displayed in bold face.

Dataset	cps	bps	sps	tps	kcps
Letter low	+0.4	+0.4	0.0	+0.1	+7.3
Letter medium	+0.6	+0.8	+1.0	+0.8	+1.2
Letter high	+0.4	+0.4	+0.8	+0.8	+0.8
Digit	+0.3	+0.1	+0.1	+0.2	+0.1
GREC	+0.8	+0.4	+0.1	+0.8	-0.2
Fingerprint	+0.2	+0.6	+0.8	+1.2	+1.4
AIDS	+0.9	-0.1	0.0	+0.2	0.0
Mutagenicity	+5.1	-0.1	+1.9	+2.8	+0.5
Protein	+4.5	-0.5	0.0	+3.5	+1.5
Webpage	+0.1	+0.1	+0.1	+0.1	+0.1

Table 5.3: Increment of classification accuracy (%) with discriminative prototype selectors

5.4 (best values are highlighted in bold face). We observe that there is only one dataset (AIDS) where the classification accuracy with the best labeled approach is as high as that of the best discriminative approach. For all other datasets, using a discriminative approach significantly outperforms all labeled methods. Moreover, narrowing our comparison to the discriminative prototype selectors alone, we note that d-sps generally outperforms the other methods and achieves the best accuracy in 8 out of 10 datasets.

Studying the optimal number of prototypes, i.e. the dimensionality of the embedding vector space, is also of interest. For the results reported in Tables 5.4 and 5.2, we have set an equal number of prototypes for each class with the balanced datasets and proportionally equal with the unbalanced datasets in all experiments. Then, we have

		Labeled Prototype Selectors					Discriminative Prototype Selectors					
Dataset	All	l-cps	l-bps	l-sps	l-tps	l-kcps	d-cps	d-bps	d-rps	d-sps	d-tps	d-kcps
Letter low	99.4	99.1	99.2	99.5	99.4	99.2	99.5	**99.6**	99.5	99.5	99.5	99.5
Letter medium	95.0	94.8	94.8	94.4	94.6	94.2	95.4	**95.6**	95.2	95.4	95.4	95.4
Letter high	92.3	92.2	92.4	92.6	92.2	92.0	92.6	92.8	93.0	**93.4**	93.0	92.8
Digit	98.4	98.3	98.5	98.6	98.4	98.5	98.6	98.6	98.6	**98.7**	98.6	98.6
GREC	92.1	91.2	91.7	92.4	91.4	92.4	92.0	92.1	92.1	**92.5**	92.2	92.2
Fingerprint	81.0	81.0	80.4	80.8	79.7	80.1	81.2	81.0	**81.6**	**81.6**	80.9	81.5
AIDS	**98.2**	97.1	98.1	**98.2**	98.0	98.1	98.0	98.0	98.0	**98.2**	**98.2**	98.1
Mutagenicity	67.6	66.0	71.2	69.6	68.3	70.1	71.1	71.1	69.9	**71.5**	71.1	70.6
Protein	73.0	70.5	72.5	73.0	71.5	60.5	**75.0**	72.0	72.0	73.0	**75.0**	62.0
Webpage	82.3	82.3	82.3	82.3	82.3	82.3	**82.4**	**82.4**	**82.4**	**82.4**	**82.4**	**82.4**

Table 5.4: Classification accuracy (%) of SVM-RBF applied to graphs embedded using all the graphs in the training set (All), the labeled prototype selectors and the discriminative prototype selectors (One-Vs-All strategy). The best result per dataset is displayed in bold face.

followed the usual training-validation-test protocol to identify the optimal number of prototypes. In addition, Figure 5.7 reports the classification accuracy on the test set as a function of the number of selected prototypes per class. The discriminative approaches almost invariably achieve better accuracy than their labeled counterparts at a parity of number of prototypes, or the same accuracy with fewer prototypes. Thus, from a computational point of view, the proposed selection strategy is also preferable to a labeled strategy as it requires a smaller number of prototypes to deliver equivalent accuracy. This translates into fewer graph edit distances to be computed for transforming each graph, with shorter training and run-time computational times.

Considering the aforementioned definitions and explanations, the labeled approaches are the same as the discriminative ones but with $W_c = 1$ and $W_s = 0$. Thus, studying the optimal value of weights as well as their influence on the classification accuracy is important. In our experiments, a grid search is used to optimize the weights. Based on the definition $W_c + W_s = 1$, thus W_s is the only free parameter, making a grid search easily feasible (the values explored range between 0.01 and 1 in 0.01 step). Table 5.5 shows the W_s value in correspondence with the accuracies reported in Table 5.2. Furthermore, Figure 5.8 presents the classification accuracy on the test set as a function of the value of W_s for two exemplary cases. Figure 5.8(a) shows a desirable case where the cross-validation accuracy is highly insensitive to the tuning of the W_s parameter. Figure 5.8(b) shows instead a case where this empirical dependence is

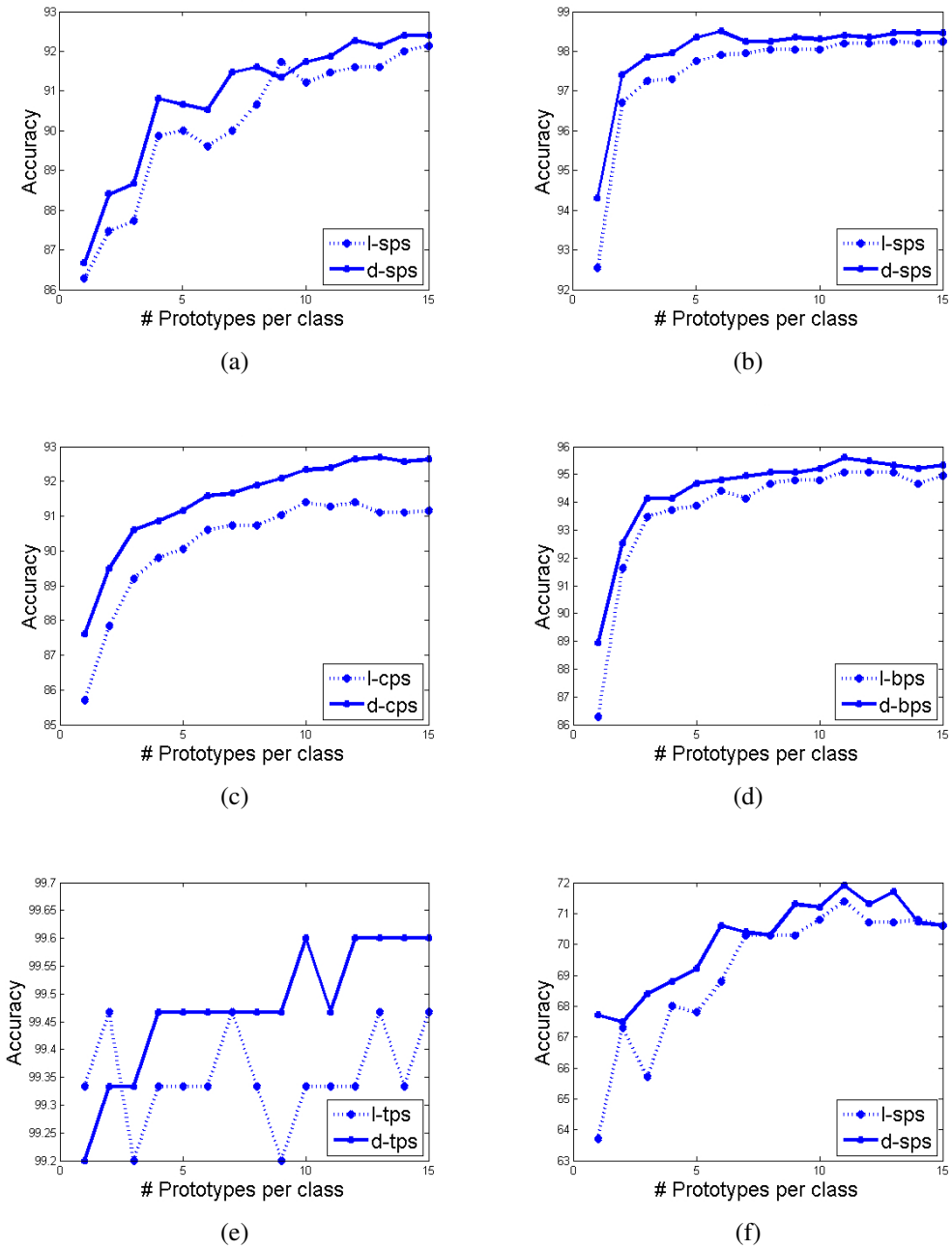
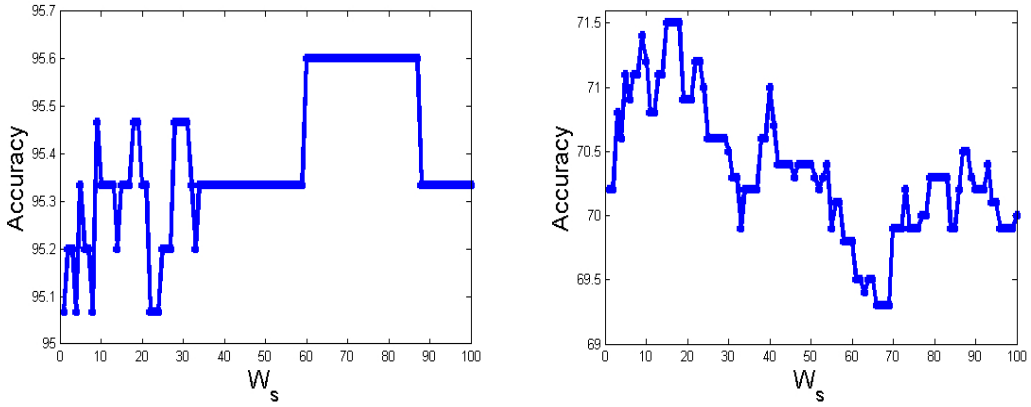


Figure 5.7: Accuracy with various prototype selection approaches and datasets as a function of the number of prototypes per class. (a) Letter High, l-sps vs d-sps; (b) Digit, l-sps vs d-sps; (c) Grec, l-cps vs d-cps; (d) Letter Medium, l-bps vs d-bps; (e) Letter Low, l-tps vs d-tps; (f) Mutagenecity, l-sps vs d-sps.

stronger. However, there still is an interval of values over which the cross-validation accuracy is unaffected.

	One-Vs-All						One-Vs-Nearest					
Dataset	d-cps	d-bps	d-rps	d-sps	d-tps	d-kcps	d-cps	d-bps	d-rps	d-sps	d-tps	d-kcps
Letter low	0.11	0.01	0.10	0.53	0.01	0.06	0.54	**0.22**	0.37	0.01	0.04	0.03
Letter medium	0.10	**0.60**	0.01	0.93	0.05	0.05	0.28	0.31	0.05	0.38	0.16	0.09
Letter high	0.13	0.14	0.01	**0.79**	0.01	0.12	0.39	0.03	0.18	0.01	0.01	0.31
Digit	0.24	0.01	0.06	**0.04**	0.03	0.56	0.81	0.04	0.03	**0.08**	0.03	0.13
GREC	0.24	0.06	0.01	**0.09**	0.03	0.14	0.02	0.17	0.29	0.18	0.08	0.81
Fingerprint	0.02	0.27	0.15	**0.20**	0.02	0.15	0.01	0.22	**0.05**	**0.20**	0.03	0.95
AIDS	0.46	0.46	0.01	**0.42**	**0.01**	0.03	0.46	0.46	0.01	**0.42**	**0.01**	0.03
Mutagenicity	0.82	0.19	0.30	**0.15**	0.01	0.04	0.82	0.19	0.30	**0.16**	0.01	0.04
Protein	**0.04**	0.01	0.05	0.01	**0.01**	0.01	**0.24**	0.01	0.14	0.01	**0.04**	0.03
Webpage	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.01	0.01	0.01	0.01	**0.01**	**0.01**

Table 5.5: The W_s value of the best classification accuracy (%) reported in Table 5.2. The W_s value which returns the best result per dataset is displayed in bold face.



(a) (b)

Figure 5.8: Accuracy with various prototype selection approaches and datasets as a function of the value of W_s (The reported W_s value is multiplied by 100). (a) Letter Medium, d-bps; (b) Mutagenicity, d-sps;.

5.4 Discussion and Conclusions

Many common data types can be seen as special cases of graphs. For example, from an algorithmic perspective both strings and trees are simple instances of graphs. A string is a graph in which each node represents one character, and consecutive characters are connected by an edge. A tree is a graph in which any two nodes are connected by exactly one path. Obviously, also a feature vector $\mathbf{x} \in \mathbb{R}^n$ can be represented as a graph, whereas the contrary, i.e. finding a vectorial description for graphs, is highly non-trivial. In other words, dissimilarity embedding can be applied to any objects for which a distance can be defined, but they are most urgent in the domain of graphs as there are no classifiers directly available, other than nearest neighbors and those based on graph kernels. Therefore, in the case of graphs, the availability of an embedding method is of crucial importance. This motivates and justifies the search for well performing embedding methods and shows that the selection of prototypes is very important for dissimilarity embedding of graphs.

Hence, in this chapter, we have proposed novel, discriminative approaches for selecting prototypes from a class-labeled collection of graphs. The proposed approaches select prototypes based on a trade off between intra-class compactness, intra-class uniform spread and inter-class separation. Experiments were carried out over a range of datasets as diverse as letters, digits, drawings, fingerprints, antiviral compounds, mutagenicity, proteins and web pages. From the experimental results, it is possible to draw the following conclusions:

- the proposed discriminative prototype selectors have increased the classification accuracy over the corresponding labeled prototype selector in 42 out of 50 cases, with increases comprising between 0.1% and 5.1% (Table 5.3);
- the best discriminative prototype selector has outperformed the best compared selector in all cases except one in which they scored equal accuracy, with increases comprising between 0.1% and 2.0% over the range of datasets (Table 5.4);
- training in a one-vs-all manner has achieved higher accuracy than one-vs-nearest training in the majority of cases (Table 5.2);

- the accuracy for the proposed discriminative approaches has proved almost invariably the highest for any tested number of prototypes per class (1 to 15) (Figure 5.7).

The conclusion brought forward by this framework is that prototype selection operating in a class-discriminative manner is an ideal approach for selecting effective prototypes for the ensuing classification task. Application is possible with any type of graph including spatial, structural, temporal, spatio-temporal and others and therefore suits a wide range of classification tasks.

Chapter 6

Complex Event Recognition by Latent Temporal Models of Concepts

In the previous chapters of this thesis, we have explored the use of graph embedding for action recognition and recognition of other structural patterns. In this chapter, we depart from graph embedding to turn our attention to very complex spatio-temporal patterns known in the literature as "events". Complex events are entities of high-level semantics such as a cruise departure, a home renovation or a wedding. Early recognition approaches have been mostly based on bags of low-level spatio-temporal features such as STIP, independent subspace analysis (ISA) and dense trajectory based HOG (DTF-HOG). More recently, the notion of "concept" has emerged as an alternative, intermediate representation with greater descriptive power, and concept detectors have been used to form "bags-of-concepts" for recognition. In these approaches, the temporal structure of the measurements is completely ignored. Yet, as we have seen in Chapters 3 and 4, a human action tends to articulate over a temporal structure and that sequential classification can help action recognition. Based on a similar rationale, in this chapter we argue that concepts in an event tend to articulate over a similar temporal structure and we exploit the scores of concept detectors as measurements in a temporal model. The temporal model leverages a latent state chain that jointly decodes the concept scores and provides event recognition. However, instead of classification with a generative approach, we have employed an equivalent undirected graphical model called HCRF and trained this model in a maximum-margin framework [138]. This ap-

proach has repeatedly reported higher accuracy compared to HMMs (e.g., [140, 150]). Furthermore, various heuristics are proposed to improve the latent state initialization and avail of the time-sparsity of the concepts. Experimental results on a sub-set of the TRECVID MED 2011 and MED 2012 datasets show that the proposed temporal approach achieves a significant improvement in average precision at a parity of features and concepts.

6.1 Prior Work and Our Contributions

Recognition of complex events in video is a current focus of computer vision, with potential application, amongst others, to Web search, multimedia indexing and retrieval, and real-time monitoring of public premises. In this chapter, we particularly refer to multimedia events of complexity such as "renovating a home", "proposing to marry", "meeting at the town hall", and the like. Large samples of these events have increasingly become available to researchers via public repositories such as YouTube and Vimeo or organized collections such as the TRECVID datasets [1, 2]. Understanding the nature of complex events can prove a fundamental requirement for their effective recognition. For instance, it could be argued that certain complex events are defined as a collection of activities or sub-events. Identifying optimal detectors for such structured events is the subject of much current investigation.

Approaches based on bags of low-level features such as Dollár *et al.*'s spatio-temporal cuboids [37], SIFT [83], ISA [74], DTF-HOG [147], STIP [70] and motion boundary histogram (MBH) [147], which have proved highly successful for recognition of primitive actions, have also proved effective for the recognition of complex events as reported in, amongst others, [79, 124, 148]. This result is very important and somehow unanticipated as it shows that, despite their complex nature, many events can be well characterized by features of low-level semantics [59, 135, 160]. In addition, pooling of these low-level features was proposed in [21], and [135] used a combination of seven different low-level features in a Bag-of-Words (BoW) framework reporting good performance on the TRECVID MED 2011 dataset [1].

While low-level features have delivered promising results, a hierarchical approach has also become increasingly popular in recent years where more general "concepts" are first identified and then used as atoms for the characterization and recognition of

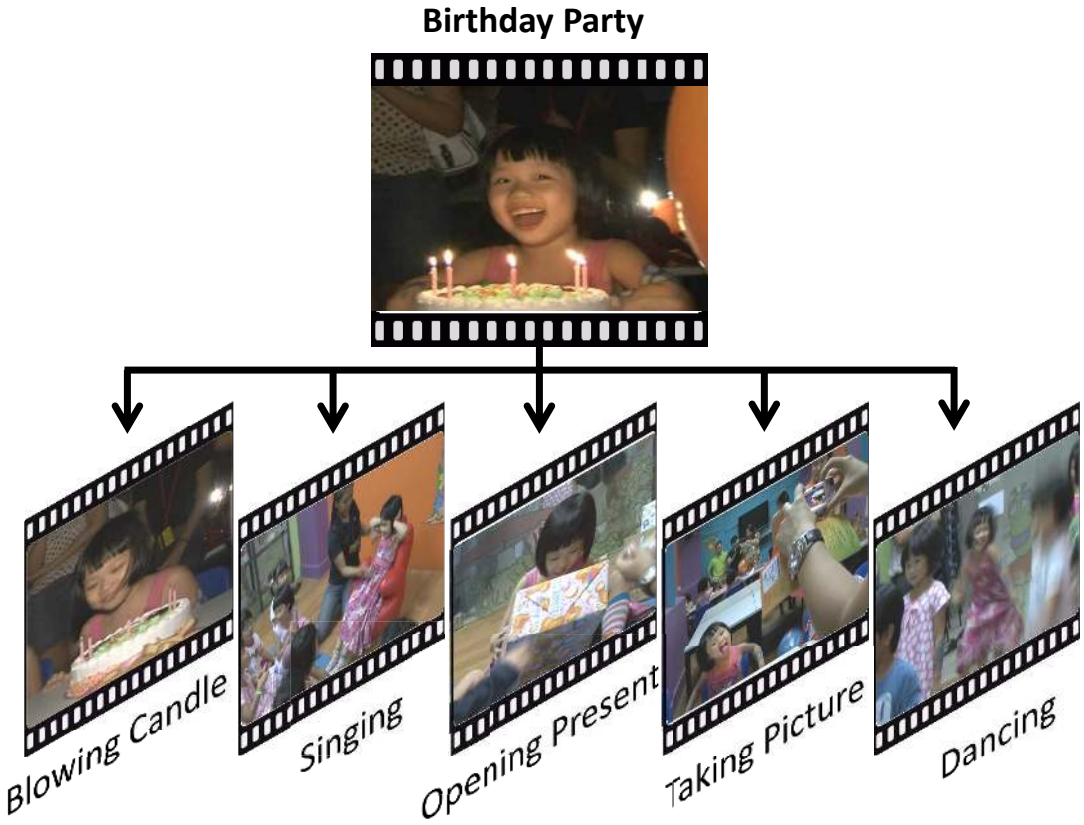


Figure 6.1: A birthday party event and its articulation over concepts.

complex events [59, 80, 82, 160]. Examples include concepts such as person eating, animal approaching and rider standing on top of bike. Fig. 6.1 illustrates a birthday party event and its articulation over concepts. Concept detectors are typically trained in a supervised manner and the use of a hierarchical approach shows analogies with that of attribute-based activity recognition [78]. As the objectives of computer vision grow in semantic scope, intermediate representations are required to fill the semantic gap. For instance, in [82] and [80], a large dataset is collected and used to train concept detectors for a task of video annotation. However, their concepts are not suitable for general videos as they were trained in constrained conditions. Loui *et al*. in [82] collected a benchmark dataset containing 25 general concepts; however, they are based on static images, not videos. Concepts have also been employed for other computer vision problems such as image ranking and retrieval [128]. In those cases, concepts

were used in the form of attributes [44], which can be considered as concepts with small granularity [59]. The most recent works on complex event recognition are [160] and [59]. The former utilized 62 action concepts as well as low-level features in a latent SVM framework, while the latter used deep learning to find data-driven concepts. Data-driven concepts are an interesting idea and have promising performance. However, they are harder to link to a conceptual description of the videos.

Bag-of-Words methods typically ignore the time stamps in the data. Yet, events are occurrences in time and as such they are likely to exhibit some degree of internal dynamics and/or temporal structure. Recently, works such as [136], [77] have demonstrated the importance of temporal structure in complex event recognition. In this chapter, we also propose to combine the use of trained concept detectors with a latent temporal model. We divide an event video into time slices and use the scores of concept detectors as measurements in a hidden conditional random field (HCRF) [105], learning its parameters with latent structural large margin [163]. In other approaches, [136] proposed the use of a variable-duration latent temporal model and learned its parameters with large margin. The main difference with our work is that [136] uses low-level features as measurements whereas we choose to employ the scores of trained concept detectors to benefit fom their higher-level semantics. Based on a similar rationale, [77] has used attribute scores as measurements. However, their temporal model is based on specialized bags of binary dynamics systems (BDSs) whereas in this work we adopt the more general latent structural max-margin framework [163]. Max-margin learning with the structural support vector machine is an efficient approach for training structural models since ϵ-approximate solutions can be obtained by only including the most violated constraints during learning [138]. Moreover, convex-concave algorithms allows extending the solution to models with latent variables [164]. However, the non-convex nature of the objective makes the solution very sensitive to the algorithm's initialization. For this reason, we explore several initialization strategies, showing that they attain a significant improvement in event classification accuracy. Furthermore, moving from the empirical observation that concepts in an event may be sparse in time, we enforce an equivalent sparsity in the latent states. In brief, the main contributions of the proposed approach are:

- The use of concept detector scores as measurements for a latent temporal model with the aim of leveraging on both trained concept detectors and the properties

of latent structural models;

- The introduction of state priors enforcing sparsity in the decoded chain of latent states in order to mirror the time-sparse distribution of concepts in an event;
- The exploration of various state initialization to improve the quality of the latent large margin solution;
- A comparative experimental evaluation against several types of bag-of-features including low-level features, concepts, and combinations of low-level features and concepts.

As datasets, we have utilized a sub-set of the NIST's TRECVID MED 2011 [1] and MED 2012 datasets [2]. These datasets are very probing in terms of event complexity and cover a total of 25 event classes [43]. The experimental results show that the combined use of concept detectors and latent temporal models significantly improves recognition performance at a parity of features and concepts.

6.2 Proposed Methods

In this section, we refer to the graphical model as hidden conditional random field even though we approach its learning by a maximum-margin method. The graphical model is displayed in Fig. 6.2. The learning objective for training the HCRF with maximum margin is defined as:

$$\begin{aligned} &\underset{W}{\operatorname{argmin}} \left(\|W\|^2 + C \sum_{i=1}^{N} \xi_i \right) \\ &s.t. \\ &W^T \Psi(a_i, y^*_{1:T_i,i}, x_{1:T_i,i}) - W^T \Psi(a, y_{1:T_i}, x_{1:T_i,i}) \\ &\qquad \geq 1 - \xi_i \qquad \forall \{a, y_{1:T_i}\} \neq \{a_i, y^*_{1:T_i,i}\} \end{aligned} \tag{6.1}$$

where a is an event label, a_i is the ground-truth label of event sample i, $x_{1:T_i,i}$ is its sequence of measurements and $y_{1:T_i}$ is an assignment for its latent states. The event label is a binary variable taking value 1 for the given event and 0 otherwise. Each latent state, $y_t, t = 1 \dots T_i$, takes values over a discrete range of indices, $\{1 \dots \mathrm{Y}\}$,

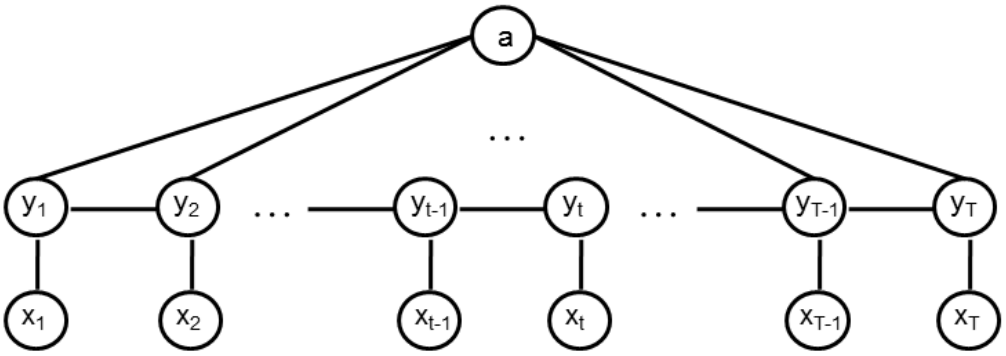


Figure 6.2: The graphical model of the hidden conditional random field. Variable a is the event class, $y_{1:T}$ are the latent states and $x_{1:T}$ are the measurements (output of concept detectors in this work).

representing the internal dynamical state of the HCRF. Each measurement, $x_t, t = 1 \dots T_i$, is an F-dimensional feature vector extracted from the image (in our case, it is the output of F = 93 concept detectors). The parameter vector, W, contains three types of parameters, or weights:

- $Y*Y$ *transition weights*, W^{tr}, scoring the transitions between consecutive states, indexed by the current and previous state values;
- $Y * F$ *emission weights*, W^{em}, indexed by the current state value and the index of the dimension in the measurement;
- $Y * 2$ *compatibility weights*, W^{cmp}, indexed by the current state value and the event value (positive or negative class).

Notation $W^T\Psi(a, y_{1:T}, x_{1:T})$ is a compound notation for the HCRF score:

$$W^T\Psi(a, h_{1:T}, x_{1:T}) = \sum_{t=2}^{T} W_{ij}^{tr}\delta\left[y_t = i, y_{t-1} = j\right] + \\ + \sum_{t=1}^{T}\sum_{f=1}^{F} W_{if}^{em} x_{tf}\delta\left[y_t = i\right] + \\ + \sum_{t=1}^{T} W_{ib}^{cmp}\delta\left[y_t = i, a = b\right] \tag{6.2}$$

Given that the states are unsupervised in the training set, their best assignment for sample i must be inferred as

$$y^*_{1:T_i,i} = \underset{y_{1:T_i}}{\operatorname{argmax}}\, W^T\Psi(a_i, y_{1:T_i}, x_{1:T_i,i}). \tag{6.3}$$

This problem can be resolved by an appropriately weighted Viterbi decoder in $O(T)$ time and the solution replaced in the constraints in (6.1) as estimated ground truth. Variable ξ_i is the slack variable for sample i, allowed to take non-negative values so as to let the inequality constraints be violated. The sum of the slack variables over the training set, $\sum_i^N \xi_i$, is an upper bound over the total classification error [138]. One can then see that the objective function in (6.1) pursues a minimization of the empirical error, while regularizing the solution by enforcing the largest possible class margin. Learning of the HCRF is obtained by alternating the solution of (6.1) and (6.3) until convergence.

Due to the exponential number of possible combinations of $a, y_{1:T_i}$ in (6.1), exhaustive verification of the constraints would not be feasible. However, [138] and [163] have shown that it is possible to find ϵ-correct solutions in polynomial time by using only the "most violated" constraints, i.e. the configuration of class and states with the highest sum of score and loss:

$$\bar{a}_i, \bar{y}_{1:T_i,i} = \underset{a, y_{1:T_i} \neq a_i, y^*_{1:T_i,i}}{\operatorname{argmax}} \left(W^T\Psi(a, y_{1:T_i}, x_{1:T_i,i}) + 1\right) \tag{6.4}$$

For the HCRF detector, such a configuration can still be efficiently determined in $O(T)$ time by a 2-best Viterbi decoder.

6.2.1 Latent State Initialization

Due to the presence of the latent variables, learning the HCRF is overall a non-convex problem, whereas the solution of (6.1) is convex in isolation. Learning can be initialized by either an arbitrary vector W in (6.3) or an arbitrary $y^*_{1:T_i,i}$ state sequence in (6.1). Choosing a state sequence could be preferable since it is somehow more confined than selecting a continuous vector, yet learning proves very sensitive to the states' initialization. [99] uses the states returned by an equivalent graphical model trained generatively by expectation-maximization (EM). However, EM requires an arbitrary initialization at its turn. In [136], the initial states are first assigned with a unique label, and then the number of labels is reduced by agglomerative clustering. In this work, we propose initialization strategies inspired by the assumed semantics for the states:

1. *Non-informative assignment (NInf)*: the initial states of each positive sample are all assigned with label 1, while those of negative samples are all assigned with label 2.

2. *Non-informative assignment with overlapping state (NInfOv)*: the initial states of each positive sample are assigned with alternate labels 1 and 2 every other frame. The states of the negative samples are assigned with alternate labels 2 and 3 likewise. This is to enforce an overlapping state across the two classes.

3. *Asymmetric assignment (Asymm)*: given that the negative class is expected to be more spread out (from being the combination of many classes), its states are assigned randomly over a small range of integers, $\{2 \dots \mathrm{Y}\}$. The initial states of the positive examples are still all assigned with label 1.

4. *Asymmetric assignment with neutral state (Sparsity)*: this assignment is similar to the previous, with the addition of a further state meant to represent "no concept". This neutral state is not included in any initial assignments, rather only reserved in anticipation of the learning stage.

6.2.2 Time-Sparsity of Concepts

To illustrate our assumption on time-sparsity of concepts in an event, in Fig. 6.3, top, we show the output of 93 concept detectors for a "Dog show" event. Most detectors

never activate significantly during the sequence (we use a threshold of 0.4 for visualization) and the few that do typically activate for only a few frames at a time. Fig. 6.3, bottom, shows the state trellis for the positive class: state 1 is the "no concept" state, and state 2 activates in loose correspondence with the highest responses from the detectors. This behavior supports the idea that the number of utilized concepts per event is relatively small, and that they tend to be time-sparse.

To leverage this property, we chose to encourage sparsity in the decoded state sequence of the HCRF by favoring transitions towards the neutral state. Given that weights in a support vector machine are akin to log-probabilities, this could be done by adding a prior weight to the neutral state. However, weights are not normalized to any given scale and it is not possible to pre-determine the size of a suitable additive prior. Therefore, we decided to use a multiplicative, positive coefficient, S, on the weights for the transitions towards the neutral state (as $S*W^{tr}_{1j}$) during the computation of both (6.3) and (6.4).

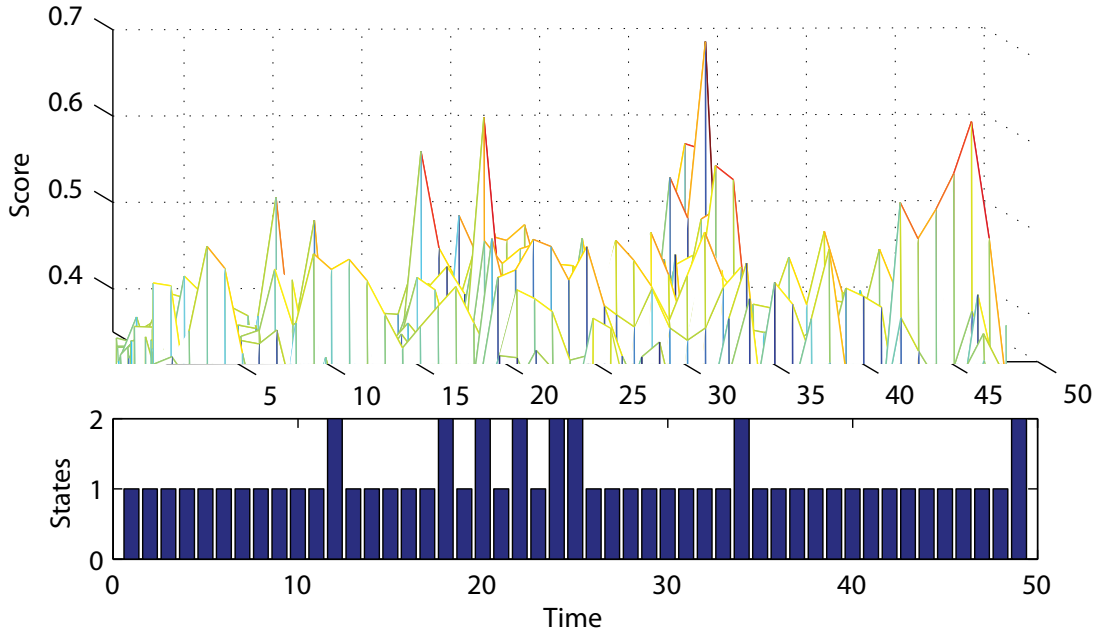


Figure 6.3: Time-sparsity of concepts and states. The top plot shows the output of concept detectors above 0.4 for an event of type "Dog show". The bottom plot shows the corresponding trellis of the states. Sparsity is evident in both the concept detectors' outputs and the states.

To perform our experiments, we have implemented the HCRF on top of the $LatentSVM^{struct}$ package of Yu and Joachims [162]. The implementation includes functions for evaluating the HCRF score (6.2), inferring the latent states (6.3), finding the most violated constraints (6.4) and, eventually, inferring the event class of an unseen example. We have used one model per event and one state variables for every clip in the sequence. The number of values for each state was varied between 5 and 25. We then chose the best value for each state on the training set.

6.3 Experimental Results

We experimented our method with video clips from the TRECVID MED 2011 and TRECVID MED 2012 multimedia datasets. The events collected in these datasets are among the most challenging due to their heavily variable duration (ranging from 30 seconds to 30 minutes), frame rate (from 12 to 30 fps), and resolution (from 320×480 to 1280×2000). As the evaluation metric, we have adopted the *average precision* which is an average of the precision at various levels of recall (equivalent to the area under the precision-recall curve) [59, 135]. As datasets, we have used the TRECVID MED 2011 Event Kit Collection (EC11 hereafter) containing 2062 videos and the TRECVID MED 2012 Event Kit Collection (EC12) consisting of 2000 videos, splitting them into 70% for training and 30% for testing (figure 6.4). The reason for this selection is that we want to be able to directly compare our results with those of [59] that used the same experimental settings but without exploiting temporal structure in the model.

As an important note, both the concept detectors and the HCRF have been trained on the training set alone, and the test videos have been used blindly for testing without any further adjustment of the parameters. A total number of 93 concepts were annotated over a portion of the training data. These concepts were selected based on the description in the TRECVID competition kit and by viewing sample videos. For each concept, at least 50 samples were selected and an SVM model trained using STIP as features [70]. The name of annotated concepts can be found in table 6.1. In order to compute the concepts' scores in a video, we first divide the video into overlapping clips, with a clip length of 180 frames and a step size of 60 frames. Subsequently, the score of each detector is computed for every clip in the video. Finally, 93 normalized

Figure 6.4: Examples from complex video event categories.

scores are collected from each clip leading to an intermediate representation of the video as a multivariate time series. For training our model, we have set the sparsifying coefficient, S, to vary over [2,5000] in logarithmic steps; ϵ was set over [10^{-2},10^{-6}], C to 100 and the linear kernel used as kernel. We compare our approach with the following methods:

- Bag-of-Concepts (BoConcept): we first apply max-pooling on the time series representation of each video, leading to a 93-dimensional vector containing the maximum score of each concept detector in that video. We use an SVM directly on such obtained high-level features, and we refer to this setup in the tables as BoConcept.

- Bag-of-Words (BoW): in this case, we cluster various low-level features (STIP, ISA, and DTF-HOG) to obtain a dictionary. Subsequently, we compute a histogram of word frequency for each feature. We use a codebook size of 10000 for all the features, and min-max normalization for the histograms. We refer to this approach in the tables by the name of the features used in the BoW framework (i.e. STIP, ISA, and DTF-HOG).

- Combinations of the various low-level features and of features and concepts: we use early fusion to combine all the low-level features (All-LL), STIPs and concepts (as the concepts were trained over STIPs; referring to this combination as STIP+HL) and all low-level features and concepts (LL+HL). Direct fusion of

the low-level features and concepts seems imbalanced as their dimensions are 10000 and 93, respectively. We therefore pre-process the low-level features with PCA to reduce their dimensionality from 10000 to 200 to make it comparable with that of the Bag-of-Concepts histograms.

animal eats food from a container	animal grabs offered food and eats	bending metal using a vice
blowing out candles on a cake	bride and groom standing in front of a priest official	brushing smoothing fur
casting	clapping	clipping nails of an animal
cutting and shaping wood	cutting dishing up fillings	cutting fur
cutting metal	cutting ripping	dancing
dancing singing in unison in a group	delivering a speech	exchanging vows
falling	flipping the bike	flipping the board and landing on it
gestures indicating directions	going down on one knee	grinding with the board
group dancing	group marching	group walking
hammering metal	heating metal over a flame	hugging
human holding food in front of an animal	jumping over obstacles	jumping with the bike
jumping with the board	kissing	landing with the board
lifting up body with arms legs	lurching of pole	marking on metal
measuring length	milling around	moving along a rock face
moving in a coordinated fashion	multiple people jumping forward	multiple people running in a race
opening closing parts	opening presents	passing a baton while running
placing fillings on bread	polishing metal	pressing ironing
putting fish in net bucket	raising hands	reeling in
removing fish from hook	removing hubcap wheel	riding bike on one wheel
rolling	running	running next to dog
scaling walls trees	sewing	sewing by hand
slicing cutting bread	sliding the board	slowing pace to a stop
soaping rinsing an animal	somersaulting	spinning the bike
spinning the bike handle	spinning the board	spinning with the board
spreading condiments on bread	standing on the board	taking a tire out
toasting bread	tracing marking	turning wrench unscrewing
unscrewing screwing parts	walking down the aisle	waving signs
attaching pieces of wood together	drilling	exchanging rings
holding out ring	lifting machine parts	making winning gestures
pulling pushing a vehicle	putting ring on finger	scrubbing appliance by hand
standing on top of bike	wiping down an appliance	working on a table top machine

Table 6.1: 93 concept's names used in the experiments.

6.3.1 TRECVID MED 2011 Event Collection

The TRECVID MED 2011 event collection consists of the following 15 events: *Boarding trick (BoT), Feeding animal (FA), Landing fish (LF), Wedding (WD), Woodworking project(WP), Birthday party (BP), Changing tire (CT), Flash mob (FM), Vehicle unstuck (VU), Grooming animal (GA), Making sandwich (MS), Parade (PR), Parkour (PK), Repairing appliance (RA)* and *Sewing project (SP)*. The performance results for the EC11 dataset are presented in Table 6.2 as average precision for each class and overall mean value. In the first four columns, we report the results of all methods based on STIP as low-level feature. Comparing mean values, one can see that BoW with STIP outperforms the Bag-of-Concepts trained over the same feature. This result is somehow expected as a similar relative performance was reported in [26] and [27]. The fusion of STIP and concepts shows only a very slight mean improvement over STIP alone (0.25%). Conversely, the proposed method shows a major improvement over Bag-of-Concepts of 9.25%, showing the importance of exploiting temporal structure over concepts. Moreover, it also reports an improvement of 4.95% over STIP and of 4.70% over the fusion of STIP and concepts.

The remaining columns in Table 6.2 show that the best result is obtained when combining all low-level features and the concepts (LL+HL). While these results offer an interesting perspective about feature complementarity, they cannot be directly compared with our method as they exploit more features. Table 6.2 also reports results from [59] and [160], showing that they are significantly outperformed by the best methods because they use less amount of supervision in certain stages of the training (Concept structure and concept set, respectively).

Table 6.3 shows a comparison of the average precision obtained with the different state initialization methods. For most classes (8 out of 15) and on average, *Sparsity* initialization outperforms the other initializations, confirming our empirical assumption on time-sparsity of the concepts within an event. However, for other event classes, other techniques achieve higher precision. The most notable remark is about the huge fluctuation in performance across the various initializations, proving that learning the objective heavily relies on effective initialization.

Event	STIP	BoConcept	STIP+HL	**Ours**	ISA	DTF-HOG	All-LL	LL+HL	[59]	[160]
BoT	85.82	75.97	81.77	*94.45*	69.89	78.92	85.00	88.18	75.7	~ 78
FA	57.70	71.06	80.60	*82.07*	70.62	67.59	69.70	72.25	56.5	~ 60
LF	86.79	63.40	88.42	*78.49*	94.67	89.20	94.67	90.26	72.2	~ 68
WD	84.69	69.05	74.22	*87.60*	78.42	77.93	90.00	91.61	67.5	~ 76
WP	72.19	60.40	72.08	*85.09*	81.96	71.04	79.21	79.64	65.3	~ 66
BP	81.54	80.99	77.60	*78.04*	82.79	87.10	91.05	88.86	78.2	~ 76
CT	71.00	72.51	71.38	*77.37*	65.48	71.61	75.39	82.95	47.7	~ 56
FM	89.17	85.87	94.48	*95.37*	84.10	83.96	89.19	92.61	91.9	~ 84
VU	75.58	82.27	73.73	*86.24*	74.35	77.71	83.49	84.67	69.1	~ 66
GA	77.50	74.35	69.97	*82.61*	74.08	80.67	82.74	86.90	51.0	~ 52
MS	80.17	83.69	71.21	*82.42*	69.60	84.72	87.84	90.66	41.9	~ 52
PR	88.34	80.94	84.86	*86.54*	82.34	92.68	91.56	93.74	72.4	~ 74
PK	81.30	82.02	86.08	*83.36*	80.04	88.28	89.14	90.11	66.4	~ 82
RA	81.90	69.61	88.60	*83.48*	77.34	69.43	82.17	85.96	78.2	~ 74
SP	80.79	77.82	83.27	*85.44*	69.13	79.12	78.96	77.97	57.5	~ 64
Mean	79.63	75.33	79.88	*84.58*	76.99	80.00	85.67	86.42	66.10	68.20

Table 6.2: The average precision for the EC11 dataset using both high-level features (i.e., concepts) and low-level features. Column 1 reports results for low-level feature STIP; column 2 report results for the concept model; and column 3 shows results for the combination of STIP and concepts. Column 4 reports the average precision achieved by the proposed approach (HCRF) using the concepts as measurements. Columns 5 and 6 show the average precision using ISA and DTF-HOG as low-level features, respectively. Column 7 shows the average precision for the combination of the features of columns 1, 5, and 6. Column 8 shows the results for the combination of the features of columns 1, 2, 5, and 6. Eventually, columns 9 and 10 show the average precision for [59] and [160], respectively.

Event	NInf	NInfOv	Asymm	Sparsity
BoT	93.79	88.84	90.37	**94.45**
FA	77.07	**82.07**	80.26	79.54
LF	73.35	57.89	77.69	**78.49**
WD	85.37	51.94	86.28	**87.60**
WP	78.94	83.95	82.04	**85.09**
BP	72.60	**78.04**	68.27	74.83
CT	71.51	70.23	73.21	**77.37**
FM	**95.37**	45.46	94.00	94.90
VU	84.84	85.23	85.68	**86.24**
GA	66.31	80.43	76.34	**82.61**
MS	**82.42**	69.54	73.24	81.09
PR	83.60	**86.54**	86.03	86.51
PK	83.35	52.31	83.26	**83.36**
RA	**83.48**	74.12	82.93	82.97
SP	80.59	84.37	**85.44**	85.39
Mean	80.84	72.73	81.67	**84.03**

Table 6.3: Comparing initializations for EC11.

6.3.2 TRECVID MED 2012 Event Collection

The TRECVID MED 2012 event collection consists of the following 10 complex events: *Bike trick (BiT), Cleaning appliance (CA), Dog show (DS), Giving direction (GD), Marriage proposal (MaP), Renovating a home (RH), Rock climbing (RC), Town hall meeting (TM), Race winning (RW)*, and *Metal craft project (MeP)*. The same concept detectors are used also here to obtain the time series vector representation for each video. The performance results for the EC12 dataset are reported in Table 6.5 as average precision for each class and overall mean value. Again, in the first four columns we report the performance of all methods based on STIP as low-level feature. Comparing mean values, one can see that the relative rankings are the same as for EC11, although the fusion of STIP and concepts shows a more significant improvement over STIP alone (2.08%). The proposed method reports another remarkable improvement of 8.92% over Bag-of-Concepts, of 6.28% over STIP, and of 4.20% over the fusion of STIP and concepts. This result gives further evidence to the benefit of exploiting temporal structure over the concept detector scores.

The remaining columns in Table 6.5 show the performance of the other single low-level features (ISA and DTF-HOG) and the fusion methods. DTF-HOG proves the best single low-level feature. The proposed method outperforms all single features, their fusion (All-LL), and even achieves a mean precision slightly higher than that of the fusion of all low-level features and concepts (73.81% vs. 73.69%).

Table 6.4 shows a comparison of the average precision obtained with the different state initialization methods. Again, for most classes (7 out of 10) and on average, *Sparsity* initialization outperforms the other initializations. Significant fluctuations in performance across the various initializations are evident also for this dataset.

Event	NInf	NInfOv	Asymm	Sparsity
BiT	63.25	67.72	**70.68**	70.39
CA	70.91	59.91	71.83	**73.76**
DS	59.74	62.56	58.41	**68.18**
GD	65.51	**75.05**	71.59	72.30
MaP	55.29	68.23	65.45	**73.86**
RH	65.68	65.50	67.44	**68.81**
RC	73.80	67.78	74.73	**76.13**
TM	66.91	72.94	69.37	**74.36**
RW	69.79	69.80	**75.36**	70.12
MeP	71.23	74.07	74.97	**79.65**
Mean	66.22	68.36	69.99	**72.76**

Table 6.4: Comparing initializations for EC12.

Event	STIP	BoConcept	STIP+HL	**Ours**	ISA	DTF-HOG	All-LL	LL+HL
BiT	61.78	69.59	67.83	*70.68*	66.48	65.09	72.94	74.22
CA	69.68	67.27	71.27	*73.76*	62.95	64.22	71.11	74.15
DS	47.88	60.09	62.36	*68.18*	62.25	66.87	66.72	68.80
GD	54.27	56.83	48.31	*75.05*	58.77	66.26	62.48	60.98
MaP	77.47	66.61	73.76	*73.86*	65.97	64.74	79.28	79.56
RH	73.06	57.48	68.03	*68.81*	76.27	66.86	73.74	72.70
RC	65.60	65.41	72.83	*76.13*	72.09	80.99	75.41	79.60
TM	69.06	60.16	72.09	*74.36*	69.20	76.90	67.48	71.95
RW	74.90	72.42	77.54	*79.65*	76.97	71.64	75.22	75.74
MeP	81.58	73.09	82.09	*77.58*	65.68	67.82	69.80	79.35
Mean	67.53	64.89	69.61	*73.81*	67.66	69.14	71.42	73.69

Table 6.5: The average precision for the EC12 dataset using both high-level features (i.e., concepts) and low-level features. Column 1 reports results for low-level feature STIP; column 2 report results for the concept model; and column 3 shows results for the combination of STIP and concepts. Column 4 reports the average precision achieved by the proposed approach (HCRF) using the concepts as measurements. Columns 5 and 6 show the average precision using ISA and DTF-HOG as low-level features, respectively. Column 7 shows the average precision for the combination of the features of columns 1, 5, and 6. Eventually, column 8 shows the average precision for the combination of the features of columns 1, 2, 5 and 6.

6.4 Discussion and Conclusions

In this chapter, we have presented an approach to complex event recognition combining a latent temporal model and trained concept detectors. The latent temporal model is a hidden conditional random field that is learned with maximum margin. The measuments for the HCRF are obtained by dividing an event clip into time slices and using the scores of concept detectors as a time series. In the HCRF, the latent state chain allows joint decoding of all the concepts in the event and ultimately supports event recognition. However, learning proves heavily sensitive to state initialization and we have therefore proposed several heuristics for initializing the latent states. In addition, we have given empirical evidence to the time-sparsity of the concept detector scores and proposed a strategy to promote an equivalent sparsity in the HCRF states. Experimental results over video clips from the challenging TRECVID MED 2011 and MED 2012 datasets show that:

- the mean average precision of the proposed method proves 9.25% higher than a Bag-of-Concepts methods based on the same concepts for the TRECVID MED 2011 Event Kit Collection (EC11). In addition, it is also 4.70% higher than that of the best comparable method (combination of STIP and concepts trained on STIP);

- the mean average precision of the proposed method proves 8.92% higher than a Bag-of-Concepts methods based on the same concepts for the TRECVID MED 2012 Event Kit Collection (EC12). Moreover, it is also 4.20% higher than that of the best comparable method (combination of STIP and concepts trained on STIP);

- although our method uses concepts from STIPs alone, its mean average precision proves higher than that of BoW methods employing other low-level features, combinations of multiple low-level features and combinations of low-level features and concepts with the EC12 dataset, and higher than any single low-level feature with EC11. Such results gives further testimony to the interesting performance of the proposed approach.

  Overall, the above results give strong evidence to the benefit of exploiting temporal structure over the output of trained concept detectors.

Chapter 7

Conclusions

The main theme of this monograph is a set of methodologies aiming to incorporate structural information in the recognition and classification of human actions in video. However, its initial scope has also expanded to cover classification of other types of structured data.

In Chapter 3, we have proposed a novel method for human action recognition based on graph embedding. Graphs are a suitable representation for embedding the spatial structure of data, and their embedding into vector spaces allows us to treat graphs as vectors during the classification stage. Based on this rationale, we have used a graph to represent the actor's shape in every frame of an action video and then embedded such a graph into a suitable shape vector. In order to also capture the temporal structure, we have made use of sequential (temporal) classifiers such as hidden Markov models and conditional random fields. We have tested this approach on an action dataset and obtained a number of interesting results. Although this method does not match the accuracy of other, existing approaches, it shows the ability to encapsulate the global structural information in the videos and as such generates a novel methodology for human action recognition based on graph embedding.

The experiments in Chapter 3 were conducted using a feature set based on the embedded graph and simple information about the actor's speed and location. An obvious limitation of such a feature set is that it ignores textural information which is instead the focus of popular spatio-temporal descriptors such as STIP, HOG, HOF and others. Hence, in Chapter 4, we have first presented an extended hidden Markov model - named hidden Markov model for multiple, irregular observations (HMM-MIO) -

which is capable of processing sparse local descriptors. We have initially tested this approach for the task of joint action segmentation and classification over a stitched version of the KTH action dataset and the challenging CMU multi-modal activity dataset and achieved greater accuracy compared to other existing methods. We have then employed HMM-MIO for the fusion of the textural information of STIP descriptors and the spatial structure provided by the embedding graphs, obtaining higher accuracy than from either approach separately. In particular, the accuracy achieved using graph embedding in addition to STIP descriptors was 4% higher than with STIP descriptors alone, showing that embedded shape graphs offer complementary, global information to local features and could be used as an augmented feature set in general application.

In the aformentioned chapters, we have also proposed discriminative prototype selection approaches which have permitted a significant improvement in classification accuracy for human action recognition compared to the state-of-the-art prototype selection methods. As such, in Chapter 5, we have extended this approach and introduced a novel comprehensive framework for selecting prototypes from a class-labeled collection of graphs based on a trade off between intra-class compactness, intra-class uniform spread and inter-class separation. Experiments were carried out over a range of structured data as diverse as letters, digits, drawings, fingerprints, antiviral compounds, mutagenicity, proteins and web pages. Results have shown that this framework is capable of achieving impressive classification accuracy compared to other well-established prototype selection methods.

In Chapter 6 of this dissertation, we have switched our focus from graph embedding and human action recognition to the recognition of complex events in consumer-uploaded Internet videos, captured with real-world settings. This task has very recently emerged as a challenging area of research across both the computer vision and multimedia communities. For this task, we have first identified a set of general "visual concepts" and then used them as an intermediate representation for the characterisation and detection of complex events. For every video, we have collected the output of trained concept detectors and represented the video as a time series of these features. Finally, we have captured the temporal structure by using a graphical model with a latent state chain that jointly decodes the concepts and provides event recognition. Instead of a generative approach, we have trained this model in a structural maximum-margin framework [138]. Experimental results over video clips from the challenging

TRECVID MED 2011 and MED 2012 datasets have shown that the proposed approach can achieve a significant improvement in accuracy (measured as average precision) at a parity of features and concepts.

Lastly, we like to highlight areas that could be subject of future work and extensions. In our judgment, the main difficulty with representing the actor's shape by a graph is the identification of a reliable set of nodes in each frame. Due to noise of various nature and variable appearance, the extracted set of keypoints that we use as nodes can vary abruptly over the frames and lead to major distortions in the graph. A possible way to regularise this behaviour is by adding other information such as contours, splines or region graph-cuts during the formation of the graph. Another interesting area of future investigation could be the joint detection of actions and events based on a combination of the intermediate-level representations used in this thesis: shape graphs and concepts. Human actions, essential visual concepts and their explicit or implicit recognition during the recognition of more complex events may eventually lead to better performing, combined approaches.

References

[1] TRECVID multimedia event detection track. http://www-nlpir.nist.gov/projects/tv2011/tv2011.html, 2012. 89, 92

[2] TRECVID multimedia event detection track. http://www-nlpir.nist.gov/projects/tv2011/tv2012.html, 2012. 89, 92

[3] JAKE K AGGARWAL AND SANGHO PARK. Human motion: Modeling and recognition of actions and interactions. In *3D Data Processing, Visualization and Transmission (3DPVT), 2004 IEEE International Symposium on*, pages 640–647, 2004. 8

[4] MD ATIQUR RAHMAN AHAD, TAKEHITO OGATA, JOO KOOI TAN, HS KIM, AND SEIJI ISHIKAWA. Motion recognition approach to solve overwriting in complex actions. In *Automatic Face and Gesture Recognition (FG), 2008 IEEE International Conference on*, pages 1–6, 2008. 9

[5] SAAD ALI AND MUBARAK SHAH. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **32**[2]:288–303, 2010. 9

[6] R. AMBAUEN, STEFAN FISCHER, AND HORST BUNKE. Graph edit distance with node splitting and merging, and its application to diatom identification. In *Graph Based Representations in Pattern Recognition*, **2726**, pages 95–106. 2003. 67

[7] HERBERT BAY, TINNE TUYTELAARS, AND LUC VAN GOOL. SURF: Speeded up robust features. In *Computer Vision (ECCV), 2006 European Conference on*, **3951**, pages 404–417. 2006. 9

[8] MIKHAIL BELKIN AND PARTHA NIYOGI. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, **15**[6]:1373–1396, 2003. 35

[9] HELEN M. BERMAN, JOHN WESTBROOK, ZUKANG FENG, GARY GILLILAND, TALAPADY N. BHAT, HELGE WEISSIG, ILYA N. SHINDYALOV, AND PHILIP E. BOURNE. The protein data bank. *Nucleic acids research*, **28**[1]:235–242, 2000. 79

[10] JEFF A. BILMES ET AL. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, **4**[510]:126, 1998. 21, 56, 57

[11] CHRISTOPHER M. BISHOP. *Pattern recognition and machine learning*, **1**. Springer New York, 2006. 16, 17, 18, 19

[12] JARON BLACKBURN AND ERALDO RIBEIRO. Human motion recognition using isomap and dynamic time warping. In *Human Motion - Understanding, Modeling, Capture and Animation*, **4814**, pages 285–298. 2007. 33

[13] MOSHE BLANK, LENA GORELICK, ELI SHECHTMAN, MICHAL IRANI, AND RONEN BASRI. Actions as space-time shapes. In *Computer Vision (ICCV), 2005 IEEE International Conference on*, **2**, pages 1395–1402, 2005. 9

[14] AARON F. BOBICK AND JAMES W. DAVIS. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **23**[3]:257–267, 2001. 9

[15] MATTHEW BRAND AND VERA KETTNAKER. Discovery and segmentation of activities in video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**[8]:844–851, 2000. 11

[16] WILLIAM BRENDEL AND SINISA TODOROVIC. Learning spatiotemporal graphs of human activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 778–785, 2011. 67

[17] HORST BUNKE AND G. ALLERMANN. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, **1**[4]:245–253, 1983. 12, 14, 16

[18] HORST BUNKE, PETER J. DICKINSON, MIRO KRAETZL, AND WALTER D. WALLIS. *A graph-theoretic approach to enterprise network dynamics*, **24**. Birkháuser, 2007. 67

[19] HORST BUNKE AND KASPER RIESEN. Improving vector space embedding of graphs through feature selection algorithms. *Pattern Recognition*, **44**[9]:1928–1940, 2011. 68

[20] HORST BUNKE AND KASPER RIESEN. Towards the unification of structural and statistical pattern recognition. *Pattern Recognition Letters*, **33**[7]:811–825, 2012. 68, 81

[21] LIANGLIANG CAO, YADONG MU, APOSTOL NATSEV, SHIH-FU CHANG, GANG HUA, AND JOHNR. SMITH. Scene aligned pooling for complex video recognition. In *Computer Vision (ECCV), 2012 European Conference on*, **7575**, pages 688–701. 2012. 89

[22] CHIH-CHUNG CHANG AND CHIH-JEN LIN. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2**[3]:27, 2011. 45, 81

[23] OLIVIER CHAPELLE. Training a support vector machine in the primal. *Neural Computation*, **19**[5]:1155–1178, 2007. 25

[24] SOTIRIOS P. CHATZIS, DIMITRIOS I. KOSMOPOULOS, AND THEODORA A. VARVARIGOU. Robust sequential data modeling using an outlier tolerant hidden Markov model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**[9]:1657–1669, 2009. 53, 58

[25] TRISTA P. CHEN, HORST HAUSSECKER, ALEXANDER BOVYRIN, ROMAN BELENOV, KONSTANTIN RODYUSHKIN, ALEXANDER KURANOV, AND VICTOR ERUHIMOV. Computer vision workload analysis: case study of video surveillance systems. *Intel Technology Journal*, **9**[2]:109–118, 2005. xi, 36, 37

[26] HUI CHENG, AMIR TAMRAKAR, SAAD ALI, QIAN YU, OMAR JAVED, JINGEN LIU, AJAY DIVAKARAN, H SAWHNEY, A HAUPTMAN, MUBARAK SHAH, ET AL. Team SRI-Sarnoff's AURORA system @ TRECVID 2011. In *Proceedings of NIST TRECVID, Workshop*, 2011. 101

[27] HUI CHENG, AMIR TAMRAKAR, SAAD ALI, QIAN YU, OMAR JAVED, JINGEN LIU, AJAY DIVAKARAN, H SAWHNEY, A HAUPTMAN, MUBARAK SHAH, ET AL. Team SRI-Sarnoff's AURORA system @ TRECVID 2012. In *Proceedings of NIST TRECVID, Workshop*, 2012. 101

[28] OSCAR P. CONCHA, RICHARD Y. D. XU, ZIA MOGHADDAM, AND MASSIMO PICCARDI. HMM-MIO: an enhanced hidden Markov model for action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Conference on*, pages 62–69, 2011. 51, 65

[29] DONATELLO CONTE, PASQUALE FOGGIA, CARLO SANSONE, AND MARIO VENTO. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, **18**[3]:265–298, 2004. 12, 13, 66

[30] DIANE J. COOK AND LAWRENCE B. HOLDER. *Mining graph data*. Wiley-Interscience, 2006. 67

[31] CORINNA CORTES AND VLADIMIR VAPNIK. Support-vector networks. *Machine learning*, **20**[3]:273–297, 1995. 24, 32

[32] KOBY CRAMMER AND YORAM SINGER. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, **2**:265–292, 2002. 28, 32

[33] RITA CUCCHIARA, COSTANTINO GRANA, ANDREA PRATI, AND ROBERTO VEZZANI. Probabilistic posture classification for human-behavior analysis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, **35**[1]:42–54, 2005. 9

[34] FERNANDO DE LA TORRE, JESSICA K. HODGINS, JAVIER MONTANO, AND SERGIO VALCARCEL. Detailed human data acquisition of kitchen activities: the CMU multimodal activity database (CMU-MMAC). In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009*, 2009. 51, 61

[35] PETER J. DICKINSON, HORST BUNKE, AREK DADEJ, AND MIRO KRAETZL. Matching graphs with unique node labels. *Pattern Analysis and Applications*, **7**[3]:243–254, 2004. 67

[36] PIOTR DOLLÁR, VINCENT RABAUD, GARRISON COTTRELL, AND SERGE BELONGIE. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005 IEEE International Workshop on*, pages 65–72, 2005. 10, 59

[37] PIOTR DOLLÁR, VINCENT RABAUD, GARRISON COTTRELL, AND SERGE BELONGIE. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005 IEEE International Workshop on*, pages 65–72, 2006. 33, 89

[38] PHILIPPE DOSCH AND ERNEST VALVENY. Report on the second symbol recognition contest. In *Graphics Recognition. Ten Years Review and Future Perspectives*, **3926**, pages 381–397. 2006. 76

[39] D.T.P. AIDS antiviral screen. `http://dtp.nci.nih.gov/docs/aids/aids_data.html`, 2004. 78

[40] OLIVIER DUCHENNE, IVAN LAPTEV, JOSEF SIVIC, FRANCIS BACH, AND JEAN PONCE. Automatic annotation of human actions in video. In *Computer Vision (ICCV), 2009 IEEE International Conference on*, pages 1491–1498, 2009. 50

[41] RAKESH DUGAD AND UDAY B. DESAI. A tutorial on hidden Markov models. Technical report, Signal Processing and Artificial Neural Networks Laboratory Department of Electrical Engineering Indian Institute of Technology Bombay Powai, Bombay 400 076, India, 1996. 21

[42] ALEXEI A. EFROS, ALEXANDER C. BERG, GREG MORI, AND JITENDRA MALIK. Recognizing action at a distance. In *Computer Vision (ICCV), 2003 IEEE International Conference on*, pages 726–733, 2003. 9

[43] MARK EVERINGHAM, LUC VAN GOOL, CHRISTOPHER KI WILLIAMS, JOHN WINN, AND ANDREW ZISSERMAN. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, **88**[2]:303–338, 2010. 92

[44] ALI FARHADI, IAN ENDRES, DEREK HOIEM, AND DAVID FORSYTH. Describing objects by their attributes. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1778–1785, 2009. 91

[45] PEDRO F. FELZENSZWALB, ROSS B. GIRSHICK, DAVID MCALLESTER, AND DEVA RAMANAN. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **32**[9]:1627–1645, 2010. 1, 33

[46] ANDREAS FISCHER, KASPER RIESEN, AND HORST BUNKE. An experimental study of graph classification using prototype selection. In *Pattern Recognition (ICPR), 2008 International Conference on*, pages 1–4, 2008. 81

[47] MARTIN A. FISCHLER AND ROBERT A. ELSCHLAGER. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, **100**[1]:67–92, 1973. 51

[48] ANDREW FRANK AND ARTHUR ASUNCION. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2010. 76

[49] HIRONOBU FUJIYOSHI AND ALAN J. LIPTON. Real-time human motion analysis by image skeletonization. In *Applications of Computer Vision (WACV), 1998 IEEE Workshop on*, pages 15–21, 1998. 9

[50] XINBO GAO, BING XIAO, DACHENG TAO, AND XUELONG LI. A survey of graph edit distance. *Pattern Analysis and Applications*, **13**[1]:113–129, 2010. 13, 14, 15

[51] ZAN GAO, MING-YU CHEN, ALEXANDER G. HAUPTMANN, AND ANNI CAI. Comparing evaluation protocols on the KTH dataset. In *Human Behavior Understanding*, **6219**, pages 88–100. 2010. 58

[52] ANDREW GILBERT, JOHN ILLINGWORTH, AND RICHARD BOWDEN. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Computer Vision (ICCV), 2009 IEEE International Conference on*, pages 925–931, 2009. 49

[53] LUTZ GOLDMANN, MUSTAFA KARAMAN, AND THOMAS SIKORA. Human body posture recognition using MPEG-7 descriptors. In *Proceedings of SPIE*, **5308**, pages 177–188, 2004. 9

[54] KAI GUO, PRAKASH ISHWAR, AND JANUSZ KONRAD. Action recognition using sparse representation on covariance manifolds of optical flow. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 IEEE International Conference on*, pages 188–195, 2010. 46

[55] ZAÏD HARCHAOUI AND FRANCIS BACH. Image classification with segmentation graph kernels. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8. Ieee, 2007. 67

[56] GISLI R. HJALTASON AND HANAN SAMET. Properties of embedding methods for similarity searching in metric spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **25**[5]:530–549, 2003. 69

[57] MINH HOAI, ZHEN-ZHONG LAN, AND FERNANDO DE LA TORRE. Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3265–3272, 2011. 50

[58] STEFAN HUWER AND HEINRICH NIEMANN. 2D-object tracking based on projection-histograms. In *Computer Vision (ECCV), 1998 European Conference on*, **1406**, pages 861–876. 1998. 9

[59] HAMID IZADINIA AND MUBARAK SHAH. Recognizing complex events using large margin joint low-level event model. In *Computer Vision (ECCV), 2012 European Conference on*, **7575**, pages 430–444. 2012. 89, 90, 91, 97, 101, 102

[60] YANGQING JIA AND TREVOR DARRELL. Heavy-tailed distances for gradient based image descriptors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 397–405, 2011. 49, 52

[61] IAN JOLLIFFE. *Principal component analysis*, **2**. Wiley Online Library, 2002. 68

[62] MICHAEL I. JORDAN AND YAIR WEISS. Graphical models: Probabilistic inference. *The Handbook of Brain Theory and Neural Networks*, 2002. 16

[63] LEONARD KAUFMAN AND PETER J. ROUSSEEUW. *Finding groups in data: an introduction to cluster analysis*, **39**. Wiley Online Library, 1990. 74

[64] JEROEN KAZIUS, ROSS MCGUIRE, AND ROBERTA BURSI. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, **48**[1]:312–320, 2005. 78

[65] YAN KE, RAHUL SUKTHANKAR, AND MARTIAL HEBERT. Event detection in crowded videos. In *Computer Vision (ICCV), 2007 IEEE International Conference on*, pages 1–8, 2007. 11

[66] ROSS KINDERMANN, JAMES LAURIE SNELL, ET AL. *Markov random fields and their applications*, **1**. American Mathematical Society Providence, RI, 1980. 20

[67] ALEXANDER KLASER, MARCIN MARSZALEK, AND CORDELIA SCHMID. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference (BMVC)*, 2008. 9

[68] ADRIANA KOVASHKA AND KRISTEN GRAUMAN. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2046–2053, 2010. 49

[69] JOHN D. LAFFERTY, ANDREW MCCALLUM, AND FERNANDO C.N. PEREIRA. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Machine Learning (ICML), 2001 International Conference*, pages 282–289, 2001. 22, 57

[70] IVAN LAPTEV. On space-time interest points. *International Journal of Computer Vision*, **64**[2-3]:107–123, 2005. 3, 9, 33, 48, 49, 63, 89, 97

[71] IVAN LAPTEV, BARBARA CAPUTO, CHRISTIAN SCHÜLDT, AND TONY LINDEBERG. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, **108**[3]:207–229, 2007. 9

[72] IVAN LAPTEV, MARCIN MARSZALEK, CORDELIA SCHMID, AND BENJAMIN ROZENFELD. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008. xii, 9, 10, 46, 49, 50, 58, 59, 60, 61

[73] IVAN LAPTEV AND PATRICK PÉREZ. Retrieving actions in movies. In *Computer Vision (ICCV), 2007 IEEE International Conference on*, pages 1–8, 2007. 9, 10

[74] QUOC V. LE, WILL Y. ZOU, SERENA Y. YEUNG, AND ANDREW Y. NG. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011. 89

[75] NING LI AND DE XU. Action recognition using weighted three-state hidden Markov model. In *Signal Processing (ICSP), 2008 IEEE International Conference on*, pages 1428–1431, 2008. 9

[76] WANQING LI, ZHENGYOU ZHANG, AND ZICHENG LIU. Expandable data-driven graphical modeling of human actions based on salient postures. *Circuits*

and Systems for Video Technology, IEEE Transactions on, **18**[11]:1499–1510, 2008. 50

[77] WEIXIN LI, QIAN YU, HARPREET SAWHNEY, AND NUNO VASCONCELOS. Recognizing activities via bag of words for attribute dynamics. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2587–2594, 2013. 91

[78] JINGEN LIU, BENJAMIN KUIPERS, AND SILVIO SAVARESE. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3337–3344, 2011. 90

[79] JINGEN LIU, JIEBO LUO, AND MUBARAK SHAH. Recognizing realistic actions from videos in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1996–2003, 2009. 89

[80] XUELIANG LIU AND BENOIT HUET. Automatic concept detector refinement for large-scale video semantic annotation. In *Semantic Computing (ICSC), 2010 IEEE International Conference on*, pages 97–100, 2010. 90

[81] JOSEP LLADOS AND GEMMA SANCHEZ. Graph matching versus graph parsing in graphics recognition- a combined approach. *International Journal of Pattern Recognition and Artificial Intelligence*, **18**[3]:455–473, 2004. 67

[82] ALEXANDER LOUI, JIEBO LUO, SHIH-FU CHANG, DAN ELLIS, WEI JIANG, LYNDON KENNEDY, KEANSUB LEE, AND AKIRA YANAGAWA. Kodak's consumer video benchmark data set: Concept definition and annotation. In *Workshop on Multimedia Information Retrieval (MIR), 2007 International Workshop on Workshop on Multimedia Information Retrieval*, pages 245–254, 2007. 90

[83] DAVID G. LOWE. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**[2]:91–110, 2004. 9, 36, 89

[84] BIN LUO, RICHARD C. WILSON, AND EDWIN R. HANCOCK. Spectral embedding of graphs. *Pattern recognition*, **36**[10]:2213–2230, 2003. 67

[85] PIERRE MAHÉ, NOBUHISA UEDA, TATSUYA AKUTSU, JEAN-LUC PERRET, AND JEAN-PHILIPPE VERT. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *Journal of chemical information and modeling*, **45**[4]:939–951, 2005. 67

[86] SUBHRANSU MAJI, LUBOMIR BOURDEV, AND JITENDRA MALIK. Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184, 2011. 49

[87] DAVID MARR AND LUCIA VAINA. Representation and recognition of the movements of shapes. *The Royal Society of London: Series B: Biological Sciences*, **214**[1197]:501–524, 1982. 11

[88] FRANCISCO MARTINEZ-CONTRERAS, CARLOS ORRITE-URUNUELA, ELIAS HERRERO-JARABA, HOSSEIN RAGHEB, AND SERGIO A VELASTIN. Recognizing human actions using silhouette-based HMM. In *Advanced Video and Signal Based Surveillance (AVSS), 2009 IEEE International Conference on*, pages 43–48, 2009. 9

[89] THOMAS B. MOESLUND, ADRIAN HILTON, AND VOLKER KRÜGER. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, **104**[2-3]:90–126, 2006. 8

[90] LOUIS-PHILIPPE MORENCY, ARIADNA QUATTONI, AND TREVOR DARRELL. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007. 11, 46

[91] JAMES MUNKRES. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, **5**[1]:32–38, 1957. 15

[92] MICHEL NEUHAUS AND HORST BUNKE. A probabilistic approach to learning costs for graph edit distance. In *Pattern Recognition (ICPR), 2004 International Conference on*, **3**, pages 389–393, 2004. 14

[93] MICHEL NEUHAUS AND HORST BUNKE. Self-organizing maps for learning the edit costs in graph matching. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **35**[3]:503–514, 2005. 14

[94] MICHEL NEUHAUS AND HORST BUNKE. Automatic learning of cost functions for graph edit distance. *Information Sciences*, **177**[1]:239–247, 2007. 14, 15, 34

[95] MICHEL NEUHAUS, KASPAR RIESEN, AND HORST BUNKE. Fast suboptimal algorithms for the computation of graph edit distance. In *Structural, Syntactic, and Statistical Pattern Recognition*, **4109**, pages 163–172. 2006. 14

[96] JUAN CARLOS NIEBLES, CHIH-WEI CHEN, AND LI FEI-FEI. Modeling temporal structure of decomposable motion segments for activity classification. In *Computer Vision (ECCV), 2010 European Conference on*, **6312**, pages 392–405. 2010. 1, 48, 51

[97] SEBASTIAN NOWOZIN AND CHRISTOPH H. LAMPERT. *Structured learning and prediction in computer vision*, **6**. Now publishers Inc, 2011. 16, 32

[98] STEPHEN O'HARA AND BRUCE A DRAPER. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*, 2011. 41

[99] SOBHAN NADERI PARIZI, JOHN G. OBERLIN, AND PEDRO F. FELZENSZWALB. Reconfigurable models for scene recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2775–2782, 2012. 95

[100] ELŻBIETA PEKALSKA AND ROBERT P.W. DUIN. *The dissimilarity representation for pattern recognition: foundations and applications*. Number 64. World Scientific, 2005. 35, 68, 69

[101] ELŻBIETA PEKALSKA, ROBERT P.W. DUIN, AND PAVEL PACLÍK. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, **39**[2]:189–208, 2006. 68

[102] PATRICK PEURSUM, HUNG HAI BUI, SVETHA VENKATESH, AND GEOFF WEST. Human action segmentation via controlled use of missing data in HMMs. In *Pattern Recognition (ICPR), 2004 International Conference on*, **4**, pages 440–445, 2004. 11

[103] RONALD POPPE. A survey on vision-based human action recognition. *Image and Vision Computing*, **28**[6]:976–990, 2010. 8, 9, 10, 33

[104] HUAIJUN J. QIU AND EDWIN R. HANCOCK. Clustering and embedding using commute times. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**[11]:1873–1890, 2007. 35

[105] ARIADNA QUATTONI, SYBOR WANG, LOUIS-PHILIPPE MORENCY, MICHAEL COLLINS, AND TREVOR DARRELL. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**[10]:1848–1852, 2007. 23, 33, 91

[106] LAWRENCE R. RABINER. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**[2]:257–286, 1989. 10, 21, 22

[107] LIVA RALAIVOLA, SANJAY J. SWAMIDASS, HIROTO SAIGO, AND PIERRE BALDI. Graph kernels for chemical informatics. *Neural Networks*, **18**[8]:1093–1110, 2005. 67

[108] CEN RAO, ALPER YILMAZ, AND MUBARAK SHAH. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, **50**[2]:203–226, 2002. 11

[109] ROMAIN RAVEAUX, SÉBASTIEN ADAM, PIERRE HÉROUX, AND ÉRIC TRUPIN. Learning graph prototypes for shape recognition. *Computer Vision and Image Understanding*, **115**[7]:905–918, 2011. 69

[110] KONRAD RIECK AND PAVEL LASKOV. Linear-time computation of similarity measures for sequential data. *Journal of Machine Learning Research*, **9**:23–48, 2007. 35

[111] KASPAR RIESEN AND HORST BUNKE. *Classification and clustering of vector space embedded graphs*. World Scientific, 2010. xi, xii, 13, 40

[112] KASPAR RIESEN, MICHEL NEUHAUS, AND HORST BUNKE. Graph embedding in vector spaces by means of prototype selection. In *Graph-Based Representations in Pattern Recognition*, **4538**, pages 383–393. 2007. 34, 35, 39, 40, 68, 69, 81

[113] KASPER RIESEN AND HORST BUNKE. IAM graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition*, **5342**, pages 287–297. 2008. 75

[114] KASPER RIESEN AND HORST BUNKE. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, **27**[7]:950–959, 2009. 14, 15, 81

[115] KASPER RIESEN AND HORST BUNKE. Graph classification based on vector space embedding. *International Journal of Pattern Recognitionand Artificial Intelligence*, **23**[6]:1053–1081, 2009. 68, 69, 80, 81

[116] RYAN RIFKIN AND ALDEBARO KLAUTAU. In defense of one-vs-all classification. *Journal of Machine Learning Research*, **5**:101–141, 2004. 27

[117] ANTONIO ROBLES-KELLY AND EDWIN R. HANCOCK. A Riemannian approach to graph embedding. *Pattern Recognition*, **40**[3]:1042–1056, 2007. 67

[118] JAIRO ROCHA AND THEO PAVLIDIS. A shape analysis model with applications to a character recognition system. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **16**[4]:393–404, 1994. 67

[119] MIKEL D. RODRIGUEZ, JAVED. AHMED, AND MUBARAK SHAH. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008. 37

[120] ADAM SCHENKER. *Graph-theoretic techniques for web content mining*, **62**. World Scientific, 2005. 67, 80

[121] ADAM SCHENKER, MARK LAST, HORST BUNKE, AND ABRAHAM KANDEL. Classification of web documents using graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, **18**[3]:475–496, 2004. 67

[122] BERNHARD SCHOLKOPF, ALEXANDER SMOLA, AND KLAUS-ROBERT MULLER. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**[5]:1299–1319, 1998. 35

[123] IDA SCHOMBURG, ANTJE CHANG, CHRISTIAN EBELING, MARION GREMSE, CHRISTIAN HELDT, GREGOR HUHN, AND DIETMAR SCHOMBURG. BRENDA, the enzyme database: updates and major new developments. *Nucleic acids research*, **32**[1]:431–433, 2004. 79

[124] CHRISTIAN SCHULDT, IVAN LAPTEV, AND BARBARA CAPUTO. Recognizing human actions: a local SVM approach. In *Pattern Recognition (ICPR), 2004 International Conference on*, **3**, pages 32–36, 2004. xi, 3, 10, 34, 36, 44, 46, 49, 51, 58, 59, 89

[125] PAUL SCOVANNER, SAAD ALI, AND MUBARAK SHAH. A 3-dimensional Sift descriptor and its application to action recognition. In *Multimedia, 2007 International Conference on*, pages 357–360, 2007. 9

[126] JOHN SHAWE-TAYLOR AND NELLO CRISTIANINI. *Kernel methods for pattern analysis*. Cambridge university press, 2004. 81

[127] QINFENG SHI, LI WANG, LI CHENG, AND ALEX SMOLA. Discriminative human action segmentation and recognition using semi-Markov model. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008. 11, 42

[128] BEHJAT SIDDIQUIE, ROGERIO S. FERIS, AND LARRY S. DAVIS. Image ranking and retrieval based on multi-attribute queries. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 801–808, 2011. 90

[129] SANCHIT SINGH, SERGIO A. VELASTIN, AND HOSSEIN RAGHEB. MuHAVi: A multicamera human action video dataset for the evaluation of action recogni-

tion methods. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 IEEE International Conference on*, pages 48–55, 2010. 37

[130] SEBASTIEN SORLIN AND CHRISTINE SOLNON. Reactive tabu search for measuring graph similarity. In *Graph-Based Representations in Pattern Recognition*, **3434**, pages 172–182. 2005. 14

[131] BARBARA SPILLMANN, MICHEL NEUHAUS, HORST BUNKE, ELŻBIETA PEKALSKA, AND ROBERT P.W. DUIN. Transforming strings to vector spaces using prototype selection. In *Structural, Syntactic, and Statistical Pattern Recognition*, **4109**, pages 287–296. 2006. 68

[132] PONNUTHURAI N. SUGANTHAN AND HONG YAN. Recognition of handprinted Chinese characters by constrained graph matching. *Image and vision computing*, **16**[3]:191–201, 1998. 67

[133] CHARLES SUTTON AND ANDREW MCCALLUM. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, **93**:142–146, 2007. 10, 20, 22, 23, 42

[134] ANH-PHUONG TA, CHRISTIAN WOLF, GUILLAUME LAVOUE, AND ATILLA BASKURT. Recognizing and localizing individual activities through graph matching. **0**, pages 196–203, 2010. 34, 51

[135] AMIR TAMRAKAR, SAAD ALI, QIAN YU, JINGEN LIU, OMAR JAVED, AJAY DIVAKARAN, HUI CHENG, AND HARPREET SAWHNEY. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3681–3688, 2012. 89, 97

[136] KEVIN TANG, LI FEI-FEI, AND DAPHNE KOLLER. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1250–1257, 2012. 91, 95

[137] MICHAEL E TIPPING AND CHRISTOPHER M BISHOP. Probabilistic principal component analysis. *The Royal Statistical Society: Series B (Statistical Methodology)*, **61**[3]:611–622, 1999. 53

[138] IOANNIS TSOCHANTARIDIS, THORSTEN JOACHIMS, THOMAS HOFMANN, AND YASEMIN ALTUN. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, **6**:1453–1484, 2005. 3, 28, 32, 57, 88, 91, 94, 108

[139] PAVAN TURAGA, RAMA CHELLAPPA, V. S. SUBRAHMANIAN, AND OCTAVIAN UDREA. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, **18**[11]:1473–1488, 2008. 8

[140] DOUGLAS L. VAIL, MANUELA M. VELOSO, AND JOHN D. LAFFERTY. Conditional random fields for activity recognition. In *Autonomous Agents and Multi-Agent Systems (AAMAS), 2007 International Conference on*, page 235, 2007. 50, 57, 89

[141] VLADIMIR N. VAPNIK. Statistical learning theory. 1998. 81

[142] ANDREA VEDALDI AND BRIAN FULKERSON. 37

[143] ROBERTO VEZZANI, DAVIDE BALTIERI, AND RITA CUCCHIARA. HMM based action recognition with projection histogram features. In *Recognizing Patterns in Signals, Speech, Images and Videos*, **6388**, pages 286–293. 2010. 9

[144] S.V.N. VISHWANATHAN, NICOL N. SCHRAUDOLPH, RISI KONDOR, AND KARSTEN M. BORGWARDT. Graph kernels. *The Journal of Machine Learning Research*, **11**:1201–1242, 2010. 67, 79

[145] SHIV N. VITALADEVUNI, VILI KELLOKUMPU, AND LARRY S. DAVIS. Action recognition using ballistic dynamics. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008. 11

[146] MARTIN J. WAINWRIGHT AND MICHAEL I. JORDAN. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, **1**[1-2]:1–305, 2008. 16

[147] HENG WANG, ALEXANDER KLÄSER, CORDELIA SCHMID, AND CHENG-LIN LIU. Action recognition by dense trajectories. In *Computer Vision and Pattern*

Recognition (CVPR), 2011 IEEE Conference on, pages 3169–3176, 2011. 49, 89

[148] HENG WANG, MUHAMMAD M. ULLAH, ALEXANDER KLASER, IVAN LAPTEV, AND CORDELIA SCHMID. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC)*, 2009. 49, 89

[149] LIANG WANG AND DAVID SUTER. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007. 9

[150] SY BOR WANG, ARIADNA QUATTONI, LOUIS-PHILIPPE MORENCY, DAVID DEMIRDJIAN, AND TREVOR DARRELL. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, **2**, pages 1521–1527, 2006. 57, 89

[151] YU WANG, IGOR V. TETKO, MARK A. HALL, EIBE FRANK, AXEL FACIUS, KLAUS F.X. MAYER, AND HANS W. MEWES. Gene selection from microarray data for cancer classificationa machine learning approach. *Computational Biology and Chemistry*, **29**[1]:37–46, 2005. 69

[152] C. WATSON AND C. WILSON. Nist special database 4. *Fingerprint Database, National Institute of Standards and Technology*, **17**, 1992. 77

[153] DANIEL WEINLAND, REMI RONFARD, AND EDMOND BOYER. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, **115**[2]:224–241, 2011. 11

[154] JASON WESTON AND CHRIS WATKINS. Support vector machines for multi-class pattern recognition. In *Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 1999 European Symposium on*, pages 61–72, 1999. 28

[155] GEERT WILLEMS, TINNE TUYTELAARS, AND LUC VAN GOOL. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Computer Vision (ECCV), 2008 European Conference on*, **5303**, pages 650–663. 2008. 9

[156] RICHARD C. WILSON AND EDWIN R. HANCOCK. Levenshtein distance for graph spectral features. In *Pattern Recognition (ICPR), 2004 International Conference on*, **2**, pages 489–492, 2004. 67

[157] RICHARD C. WILSON, EDWIN R. HANCOCK, AND BIN LUO. Pattern vectors from algebraic graph theory. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1112–1124, 2005. 35, 67

[158] JUNJI YAMATO, JUN OHYA, AND KENICHIRO ISHII. Recognizing human action in time-sequential images using hidden Markov model. In *Computer Vision and Pattern Recognition (CVPR), 1992 IEEE Conference on*, pages 379–385, 1992. 33

[159] PINGKUN YAN, SAAD M KHAN, AND MUBARAK SHAH. Learning 4D action feature models for arbitrary view action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–7, 2008. 9

[160] YANG YANG AND MUBARAK SHAH. Complex events detection using data-driven concepts. In *Computer Vision (ECCV), 2012 European Conference on*, **7574**, pages 722–735. 2012. 89, 90, 91, 101, 102

[161] ALPER YILMAZ AND MUBARAK SHAH. A differential geometric approach to representing the human actions. *Computer Vision and Image Understanding*, **109**[3]:335–351, 2008. 9

[162] CHUN-NAM YU AND THORSTEN JOACHIMS. Latent SVM Struct package. `http://www.cs.cornell.edu/~cnyu/latentssvm/`, 2009. 97

[163] CHUN-NAM YU AND THORSTEN JOACHIMS. Learning structural SVMs with latent variables. In *Machine Learning (ICML), 2009 International Conference*, pages 1169–1176, 2009. 30, 31, 32, 91, 94

[164] ALAN L. YUILLE AND ANAND RANGARAJAN. The concave-convex procedure. *Neural Computation*, **15**[4]:915–936, 2003. 91

[165] HUA ZHONG, JIANBO SHI, AND MIRKÓ VISONTAI. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2004 IEEE Conference on*. 11