Données multimodales pour l'analyse d'image

Matthieu Guillaumin

UNIVERSITÉ DE GRENOBLE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : Mathématiques et Informatique

préparée au Laboratoire Jean Kuntzmann

dans le cadre de l'École Doctorale Mathématiques,
Sciences et Technologies de l'Information, Informatique

présentée et soutenue publiquement

par

Matthieu Guillaumin

le 27 septembre 2010

Exploiting Multimodal Data for Image Understanding

Données multimodales pour l'analyse d'image

Directeurs de thèse : Cordelia Schmid et Jakob Verbeek

JURY

M. Éric Gaussier	*Université Joseph Fourier*	Président
M. Antonio Torralba	*Massachusetts Institute of Technology*	Rapporteur
Mme Tinne Tuytelaars	*Katholieke Universiteit Leuven*	Rapporteur
M. Mark Everingham	*University of Leeds*	Examinateur
Mme Cordelia Schmid	*INRIA Grenoble*	Examinatrice
M. Jakob Verbeek	*INRIA Grenoble*	Examinateur

Abstract

This dissertation delves into the use of textual metadata for image understanding. We seek to exploit this additional textual information as weak supervision to improve the learning of recognition models. There is a recent and growing interest for methods that exploit such data because they can potentially alleviate the need for manual annotation, which is a costly and time-consuming process.

We focus on two types of visual data with associated textual information. First, we exploit news images that come with descriptive captions to address several face related tasks, including *face verification*, which is the task of deciding whether two images depict the same individual, and *face naming*, the problem of associating faces in a data set to their correct names. Second, we consider data consisting of images with user tags. We explore models for automatically predicting tags for new images, *i.e. image auto-annotation*, which can also used for keyword-based image search. We also study a *multimodal semi-supervised learning* scenario for image categorisation. In this setting, the tags are assumed to be present in both labelled and unlabelled training data, while they are absent from the test data.

Our work builds on the observation that most of these tasks can be solved if perfectly adequate similarity measures are used. We therefore introduce novel approaches that involve metric learning, nearest neighbour models and graph-based methods to learn, from the visual and textual data, task-specific similarities. For faces, our similarities focus on the identities of the individuals while, for images, they address more general semantic visual concepts. Experimentally, our approaches achieve state-of-the-art results on several standard and challenging data sets. On both types of data, we clearly show that learning using additional textual information improves the performance of visual recognition systems.

Keywords

Face recognition • Face verification • Image auto-annotation • Keyword-based image retrieval • Object recognition • Metric learning • Nearest neighbour models • Constrained clustering • Multiple instance metric learning • Multimodal semi-supervised learning • Weakly supervised learning.

Résumé

La présente thèse s'intéresse à l'utilisation de méta-données textuelles pour l'analyse d'image. Nous cherchons à utiliser ces informations additionelles comme supervision faible pour l'apprentissage de modèles de reconnaissance visuelle. Nous avons observé un récent et grandissant intérêt pour les méthodes capables d'exploiter ce type de données car celles-ci peuvent potentiellement supprimer le besoin d'annotations manuelles, qui sont coûteuses en temps et en ressources.

Nous concentrons nos efforts sur deux types de données visuelles associées à des informations textuelles. Tout d'abord, nous utilisons des images de dépêches qui sont accompagnées de légendes descriptives pour s'attaquer à plusieurs problèmes liés à la reconnaissance de visages. Parmi ces problèmes, la *vérification de visages* est la tâche consistant à décider si deux images représentent la même personne, et le *nommage de visages* cherche à associer les visages d'une base de données à leur noms corrects. Ensuite, nous explorons des modèles pour prédire automatiquement les labels pertinents pour des images, un problème connu sous le nom d'*annotation automatique d'image*. Ces modèles peuvent aussi être utilisés pour effectuer des recherches d'images à partir de mots-clés. Nous étudions enfin un scénario d'*apprentissage multimodal semi-supervisé* pour la catégorisation d'image. Dans ce cadre de travail, les labels sont supposés présents pour les données d'apprentissage, qu'elles soient manuellement annotées ou non, et absentes des données de test.

Nos travaux se basent sur l'observation que la plupart de ces problèmes peuvent être résolus si des mesures de similarité parfaitement adaptées sont utilisées. Nous proposons donc de nouvelles approches qui combinent apprentissage de distance, modèles par plus proches voisins et méthodes par graphes pour apprendre, à partir de données visuelles et textuelles, des similarités visuelles spécifiques à chaque problème. Dans le cas des visages, nos similarités se concentrent sur l'identité des individus tandis que, pour les images, elles concernent des concepts sémantiques plus généraux. Expérimentalement, nos approches obtiennent des performances à l'état de l'art sur plusieurs bases de données complexes. Pour les deux types de données considérés, nous montrons clairement que l'apprentissage bénéficie de l'information textuelle supplémentaire résultant en l'amélioration de la performance des systèmes de reconnaissance visuelle.

Mots-clés

Reconnaissance de visage • Vérification de visages • Annotation automatique d'image • Recherche d'image par mots-clés • Reconnaissance d'objet • Apprentissage de distance • Modèles par plus proches voisins • Agglomération de données sous contrainte • Apprentissage de métrique par instances multiples • Apprentissage multimodal semi-supervisé • Apprentissage faiblement supervisé.

Contents

1

Introduction

Recently, large digital multimedia archives have appeared. This is the result of massive digitisation efforts from three main sources. The first source are broadcasting services who are digitising their archives and redistributing content that was previously analog. This includes television channels, major film companies and national archives or libraries, who release their archive data to the public for online consultation. Second, digital data is now produced directly by these services. For instance, news oriented media or movie makers now use digital cameras to capture their work as a digital signal – hence avoiding the loss of quality resulting from the analog-to-digital conversion of the signal – that they can publish online, or in physical formats such as DVD or Blue-ray discs, directly. Finally, with the advent of digital consumer products and media sharing websites, user provided digital content has seen an exponential growth over the last few years, with billions of multimedia documents already available on websites such as Facebook, Dailymotion, YouTube, Picasa and Flickr.[1] In Figure 1.1, we illustrate this growth by showing the increasing number of images under the Creative Common license that were uploaded every month on Flickr between April 2006 and December 2009.[2] As of February 2010, the total number of images on the Flickr website is over 4 billion.

Following this exponential growth, there is an increasing need to develop methods to allow access to such archives in a user-oriented and semantically meaningful way. Indeed, given the speed at which new data is released, the cost of using manual indexing has become prohibitive. There is a recent and large effort (*c.f.* Jégou et al. [2008], Torralba et al. [2008], Fergus et al. [2009], Perronnin et al. [2010]) to develop automatic methods to index and search web-scale data sets of images.

In order to automatically index the archive documents with the goal of providing easy and efficient access to users, it is necessary to automatically extract from the documents the semantic information that is relevant to the users. This supposes to build

[1] URLs are: `htt://www.facebook.com/`, `http://www.dailymotion.com`, `http://www.youtube.com`, `http://www.picasa.com/` and `http://www.flickr.com/`, respectively.

[2] These numbers were obtained from `http://wiki.creativecommons.org/Metrics/License_statistics`.

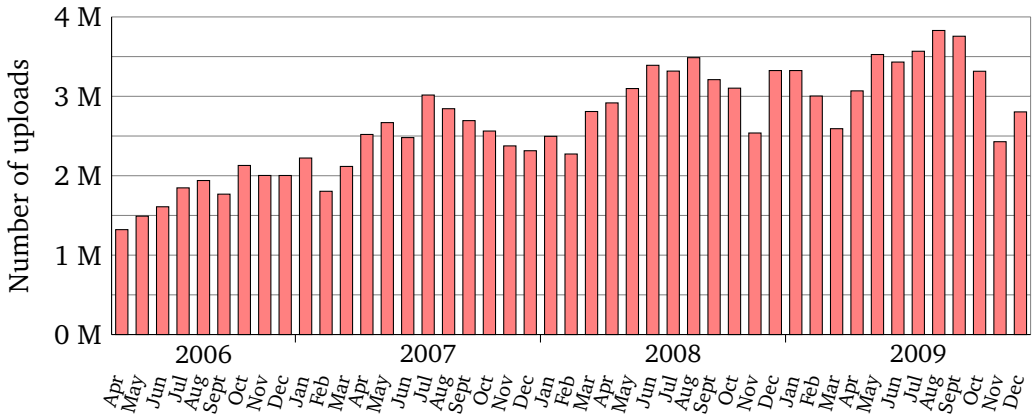


***Figure 1.1**: Bar plot of the number of images under the Creative Common (CC) license uploaded on Flickr between April 2006 and December 2009. The regular increase fluctuates with yearly peaks in the summer months. The total number of CC images in Flickr now exceeds 135 million.*

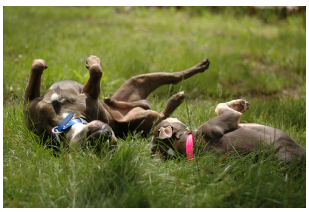
systems that can bridge the semantic gap between low-level features and semantics (Smeulders et al. [2000]), *i.e.* the gap between raw pixel values and the interpretation of the scene that a human is able to make.

To illustrate this fact, let us consider an important computer vision problem, namely image classification. The goal of image classification is the following. Given some images, which are merely two-dimensional arrays of pixel values, the system has to decide whether they are relevant to a specific visual concept, which can range from detecting an object instance to recognising object classes or general patterns. We illustrate the variety of semantic concepts that have to be dealt with in Figure 1.2. The PASCAL VOC challenge, *c.f.* Everingham et al. [2007], and the ImageCLEF Photo Retrieval and Photo Annotation tasks, *c.f.* Nowak and Dunker [2009], are good examples of the wide interest for this topic.

In parallel, it is striking that the huge amount of visual data that is available today is more and more frequently provided with additional information. For instance, this additional information may consist of text surrounding an image in a web page such as technical information on Wikipedia: from Figure 1.3 we can see that it is technically possible to extract hierarchical classification information from such data. We can also find user tags as present in video and photo sharing websites like Youtube and Flickr. These tags, as illustrated in Figure 1.4, are typically assigned by users for indexing purposes, or to provide additional information to visitors (such a camera model, etc.). Finally, captions for news images can be found on aggregation sites like Google News or Yahoo! News. Often, the captions describe the visual content of the image, also referring to the event at the origin of the photo, as shown in Figure 1.5.

Clouds, Plant life, Sky, Tree

Flowers, Plant life



Animals, Dog, Plant life

Car, Female, Male, People, Plant Life, Structures, Transport

Baby, Indoor, Male, People

Clouds, Sky, Structures, Sunset, Transport, Water

***Figure 1.2**: Illustration of the computer vision task of image classification from the MIR Flickr data set: for each of the 24 semantic visual concepts of the data set, systems are built to automatically decide whether images are relevant or not. The manually assigned concepts used for evaluation purposes are specified below the images.*

The growth of such *multimodal* data is especially true on the web, but it is not limited to this source: it is also the case of videos that come with subtitles, scripts and audio feeds. The textual *metadata* typically describes in a very weak and noisy manner the content of the visual data. This motivates our interest in using these sources of information to aid image classification and object recognition in general, and face recognition in particular. Below, we describe in more details the tasks that we have investigated.

1.1 Goals

In this thesis, we will focus on trying to understand the visual content of digital photos. Of particular interest are humans and their identities. Recognising humans and their actions has obviously many applications in surveillance. It can also help automatically organise photo collections. For instance, such techniques could be used to group images based on the identity of the depicted people. There is also a growing interest in systems to efficiently query a data set of images for retrieving images of similar persons, or of a specific person (*c.f.* Sivic et al. [2005a], Ozkan and Duygulu

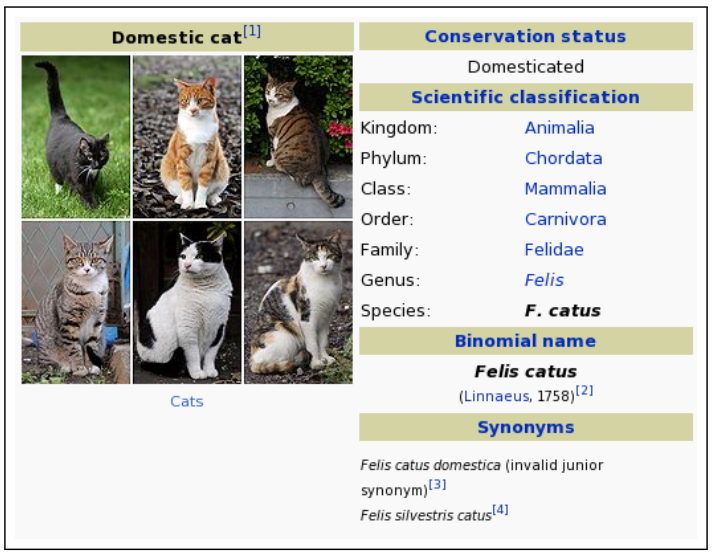
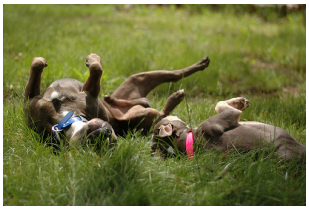


***Figure 1.3**: Examples of multimodal documents formed with images and other sources of information such as text, tags or captions (2/3): An excerpt from the Wikipedia web page on cats showing cat images and taxonomic classification information.*

desert, nature, sky, landscape

rose, pink, bloom, dcdead



israel, pitbull, brindle, nikon, d70, bliss

festivalofspeed, goodwood

baby, girl, newborn, monkey

canal, boat, water, blue, trees, bridge, amsterdam

***Figure 1.4**: Examples of multimodal documents formed with images and other sources of information such as text, tags or captions (1/3): The images from Figure 1.2 with (some of) their associated user tags on Flickr.*

In this March 23, 2010 file photo, House Speaker Nancy Pelosi of Calif. clutches a pen used by President Barack Obama to sign the health care bill, in the East Room of the White House in Washington. [...]
(AP Photo/J. Scott Applewhite, File)

***Figure 1.5**: Examples of multimodal documents formed with images and other sources of information such as text, tags or captions (3/3): A news photograph and associated caption from the Associated Press agency.*

[2010]). The recent emergence of such features in consumer products like Picasaweb and iPhoto stresses the wide interest for such tools. In Chapter 2, we therefore study face verification systems that can meet these objectives by deciding whether two face images depict the same individual, a task known as *face verification*.

To obtain large data sets of face images, we exploit the large source of images of people that news photographs form. As we have already discussed, pieces of news containing visual data also come with text. Frequently, the text is a caption that directly describes the image content, especially by naming the important persons in the image, *c.f.* Figure 1.5. In Chapter 3, we are interested in exploiting these captions to associate automatically names to the correct faces in the image. We also seek to adapt the face verification system mentioned above to fit this setting. In this way, the organisation of the photo collection with respect to human faces can be performed without any additional user intervention.

However, the automatic organisation of photo collections for efficient indexing and retrieval and more generally image understanding are problems that go beyond the focus on human faces alone. The retrieval of objects (*c.f.* Hirata and Kato [1993], Sivic and Zisserman [2003], Torralba et al. [2008]) or complex multilingual and multimodal documents (*c.f.* Peters et al. [2008]) are, for instance, much more challenging problems. One of the promising applications of computer vision for information retrieval is to allow to search a data set of images using a query string (*e.g.* as is done

with Google Image search). Text-based queries provide additional challenges: words are often ambiguous, users might be imprecise in they query formulation, and of course a relationship between words and images has to be built. This can be thought of as an automatic translation between image and text (*c.f.* Duygulu et al. [2002]).

Automatically finding which keywords are relevant for images is a difficult open problem for computer vision, but would assist users in annotating and tagging their images, while allowing for more relevant image retrieval from a text query. In Chapter 4, we investigate the use of large data sets of user-tagged images like the ones shown in Figure 1.4 for image auto-annotation and keyword-based image search. Using a visual similarity, it is possible to infer the relevance of tags for a target image, for instance by looking at the tags of the images in the data set that are most similar to the target.

Finally, we want to show that image metadata is useful for generic image categorisation, and we want to measure the practical performance gap between the different forms of supervision: the fully supervised setting where manual labels are required, semi-supervised learning where the training set is only partially labelled, and finally the weakly supervised approach where labels are imperfect. This is the aim of the study in Chapter 5.

1.2 Context

From its origins, computer vision has aimed at automatic image understanding. To comprehensively understand an image, it is interesting, as already mentioned, to be able to decide whether a specific object appears in it. However, this is not sufficient. It is also necessary to locate the objects, and explain the relationships between them.

By looking at the picture in Figure 1.6, we can realise that high-level knowledge is required to appreciate an image at its full extent. For instance, as humans, we are able gather a huge amount of information from this image, even about things that are not shown in the photo. First, there are three humans on a stage, they all play the guitar and from the relative positions of their guitars we can tell that they are right-handed. The foremost one appears to be the lead singer of the band. We are able to locate the guitars although their shape and colours vary, and we can also recognise three microphones, only one of each is currently used. This information is obtained from the proximity of the object to one of the humans displaying an open mouth, despite the absence of physical contact. Finally, it is not difficult to say that there is also a drum in the background, and probably an audience not far but outside of the photo.

In order to have computers acquire such an intelligence, there is a need for representing this knowledge. For instance, the constellation models (Fischler and Elschlager

***Figure 1.6**: A comprehensive understanding of this image requires advanced knowledge about human pose, actions and social behaviour (i.e. humans form bands to play music on stage), object appearances and usages (i.e. guitars have many different colours and shape and are manipulated by humans in a very specific way, whereas a microphone relates to neighbouring mouths without physical contact, etc.). Modelling and acquiring this knowledge is one of the goal of computer vision.*

[1973]) are an early and influential framework for representing complex objects. These models combine a rough spatial relationship between the object parts, *i.e.* authorising for some flexibility in the layout, with an appearance model that tries to locate the parts, as illustrated in Figure 1.7 (left). For instance, a face is composed of two eyes, two ears, a mouth and a nose with a specific arrangement. Similarly, a human body has a head, a torso, two arms and two legs, etc., but the variations in pose can be much wider (*c.f.* Figure 1.7, right). Although successful, the part-based models were often complex and computationally expensive to exploit.

To temporarily alleviate the problem of image representation, the computer vision community has first addressed simpler tasks such as digit and face recognition. Significant attempts at face recognition appeared in the late 1980s and early 1990s (*c.f.* Turk and Pentland [1991]) using controlled settings: the objects were well centred, easily segmented from the background, and did not offer large variations in appearance, pose, lighting, etc. As a consequence, the images themselves, or simple sketches, could be used to represent the objects. However, this approach did not generalise well for images with background clutter and occlusions.

Local descriptors have therefore been proposed to describe only local regions of the objects. They are hence robust to occlusions of other parts of the object and background clutter. They can describe either the local shape (*e.g.*, see Belongie et al.

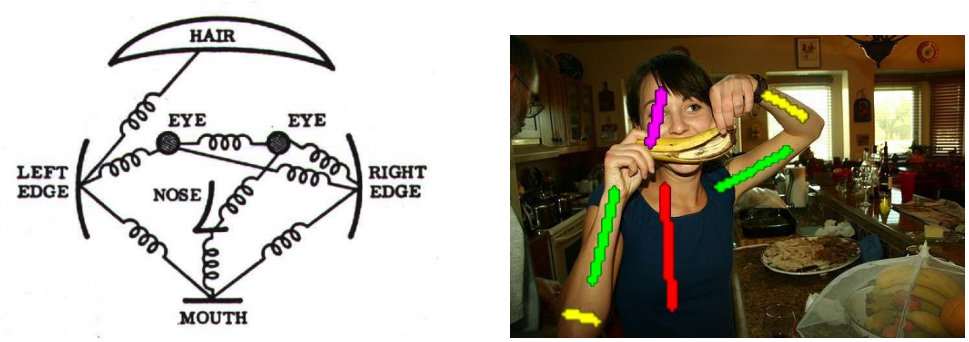


Figure 1.7: *Illustration of a constellation model for facial features and human pose estimation. Left, the sketch, courtesy of Fischler and Elschlager [1973], illustrates the idea of constellation models with elastic constraints between parts. Right, using the part-based method of Eichner and Ferrari [2009], human upper body pose estimation is performed despite the unusual human pose.*

[2002]) or local appearance (Schmid and Mohr [1997]) of the objects. The features were progressively improved and are now frequently composed of histograms of oriented gradients, such as SIFT (Lowe [1999], see Figure 1.8, left), or Haar-wavelet responses, such as SURF (*c.f.* Bay et al. [2006]). To represent an image or an image region, it has been proposed by Csurka et al. [2004] to collect such local descriptors into unordered sets, called "bags-of-features", as illustrated in Figure 1.8 (right). With the improvements in local feature extraction, part-based models have recently regained much interest. For instance, Felzenszwalb et al. [2010] exploits such models to locate objects in images.

With the elaboration of these complex image representations and the availability of larger sets of images, it appeared more and more crucial to analyse the data using mathematical models. Since these models have proved impossible to tune by hand, *machine learning* systems for vision have been designed to generalise the knowledge obtained from a set of examples provided by a human annotator. These methods are called "supervised", because they require human supervision before any learning can be performed.

With the proximity of the bag-of-features representation with the "bag-of-words" approach from the textual analysis domain, a large effort has been devoted to adapting the techniques introduced for textual analysis to solve related vision problems. For instance, the bag-of-words approach had proved very successful for text categorisation (*e.g.* Joachims [1998]). Similar success was hence obtained for object category recognition (Csurka et al. [2004]) using adequate machine learning tools such as Support Vector Machines (SVM, Vapnik [1998]).

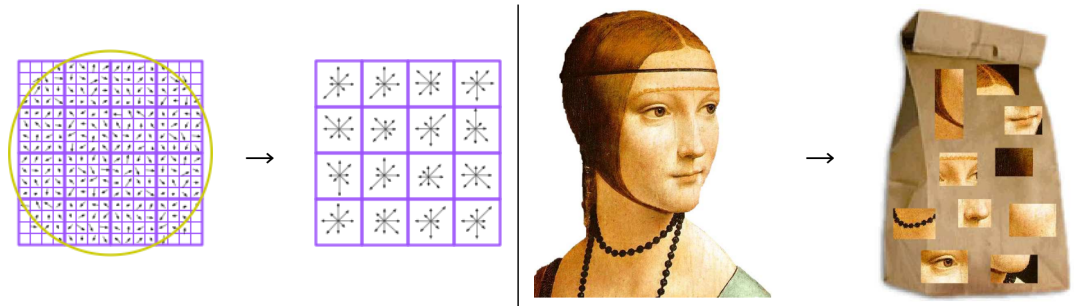


***Figure 1.8**: Illustration of a local appearance descriptor, namely SIFT (Lowe [1999]), left, which consists of a grid of histograms of oriented gradients. The bag-of-features approach (courtesy of L. Fei-Fei), shown right, consists of an unordered set of local appearance descriptors.*

The success of the supervised methods has helped the computer vision community solve more and more difficult, high-level and semantically-rich tasks, from texture classification, object instance detection, object category classification, to action recognition in videos. For instance, detectors for faces (*c.f.* Viola and Jones [2004]) and other structured objects like pedestrians (Dalal and Triggs [2005]) are now available as open-source software[3] and have reached consumer markets.

Typically, the performance of those systems increases as more manual annotation is provided. However, manually annotating examples becomes expensive if many concepts have to be learnt on large training data sets. As a result, there is a growing interest in approaches that rely less on manual annotations. At the extreme, "unsupervised" systems (*c.f.* Hinton and Sejnowski [1999], Ghahramani [2004]) do not rely at all on manual annotation to analyse the visual content but will automatically use the structure or the distribution of the data to identify groups of visually similar data (or modes in the distribution). For instance, topic detection in text documents (Hofmann [1999]) and unsupervised discovery of visual object classes (*c.f.* Sivic et al. [2005b], Quelhas et al. [2007]) have been considered, using Latent Semantic Analysis (PLSA, Hofmann [2001]), an unsupervised method. We can note that the family of machine learning techniques that have been applied to more specific face-related tasks does not differ strongly from the ones used for more generic problems, both in visual and textual analysis, *e.g.* see Jain et al. [2007] who use PLSA for people-oriented topic detection.

As intermediate approaches, "semi-supervised" learning (*c.f.* Chapelle et al. [2006], Li et al. [2007], Fergus et al. [2009]) considers training scenarios with only partially

[3] OpenCV, the open computer vision library: `http://willowgarage/`.

labelled data sets, and "weakly supervised" methods use data sets that are imperfectly labelled, either by humans or machines, *e.g.* using Google Image Search (Fergus et al. [2005]).

The metadata that we consider in this thesis typically describes in a very weak and noisy manner the content of visual data, which makes them particularly suited for weak supervision. Although imperfect, these annotations are very interesting to use as supervision for one major reason: they can be obtained at very low cost and therefore a huge amount of data can be collected. Leveraging this huge amount of data could outweigh the disadvantage of using weak supervision to train visual models. There is currently a broad line of research to exploit this idea. These work address a diversity of vision problems such as image annotation (Blei and Jordan [2003], Barnard et al. [2003]), clustering (Bekkerman and Jeon [2007]) or landmark classification (Li et al. [2009b]).

Our work is also encouraged by the recent attempt by Berg et al. [2004a] (see also Pham et al. [2010]) to automatically name the faces detected in news images, using the captions as the pool of names that can be associated to the faces, and the work of Everingham et al. [2006] to automatically identify characters in videos. The typical face processing pipeline that these work use is shown in Figure 1.9. Other human-related tasks are also considered such as sign language recognition in videos (Buehler et al. [2009]). Potentially, automatically labelled data, either images or videos, can be used to train face recognition systems, but the labels did not appear to be very reliable when building a large data set of face images (*c.f.* Huang et al. [2007b]).

In this context, we believe that it is possible to build on these recent advances that concern both feature extraction and machine learning to achieve our goals of improved face recognition in uncontrolled settings, but also to use similar weakly supervised settings for image auto-annotation and object recognition.

1.3 Contributions

In this thesis, our main contributions are the following:

- For the task of face verification in uncontrolled settings, we introduce a supervised metric learning algorithm, Logistic Discriminant-based Metric Learning (LDML). Additionally, we extend nearest neighbour approaches to classify pairs of examples whose respective classes are not present in the training set. This method, Marginalised k-Nearest Neighbour classification (MkNN), combined with metric learning methods to define the neighbourhood of a face image, helps improve the accuracy of verification. Using the two methods described

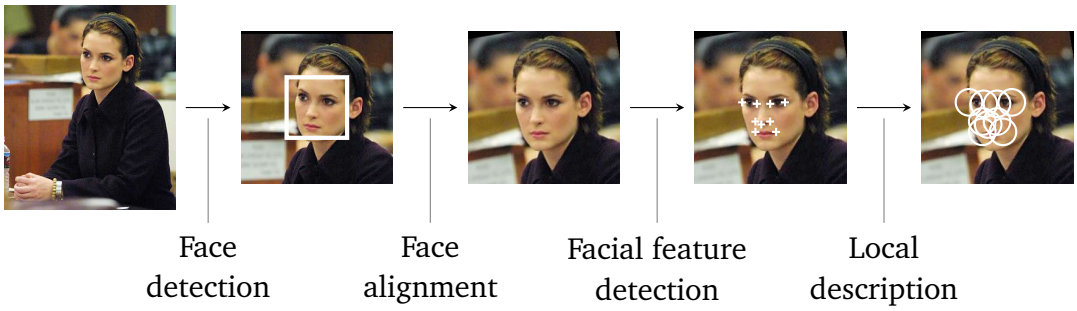


***Figure 1.9**: Illustration of the face processing pipeline of Everingham et al. [2006]. After detecting faces in the images, face images are aligned and given to a facial feature detector. Then, local appearance descriptors of the located points are extracted and concatenated to obtain a vectorial representation of the faces.*

above, we obtain state-of-the-art results on a challenging data set of face images in uncontrolled settings, with application to unconstrained clustering of faces and recognition from a single example. This work was published in Guillaumin et al. [2009b] with improvements for low rank regularisation in Guillaumin et al. [2010a], and we present it in Chapter 2.

- For the problem of associating names extracted from captions to faces detected in the corresponding images, a task usually known as *face naming*, we present a graph-based approach for constrained clustering and retrieval. To approximate the solution of the resulting NP-hard problem, we resort to a form of local optimisation where we express local constraints as a bipartite graph matching problem which can be solved exactly and efficiently. We evaluate our approach on a data set of around 20000 news images that we have manually annotated and made freely available online. This work is published in Guillaumin et al. [2008] using the Euclidean distance to compare faces, and later improved in Guillaumin et al. [2010a] using learnt metrics. We present them in Chapter 3.

- We also propose to exploit news images with captions directly to learn metrics for face recognition. To this end, we introduce a Multiple Instance Learning (MIL) formulation (illustrated in Figure 1.10) of metric learning, Multiple Instance Logistic Discriminant-based Metric Learning (MildML), that tries to optimise a metric without explicitly assigning a name to each face. We show that it is possible to learn meaningful metrics from such automatic supervision and that the MIL formulation outperforms metrics learnt from automatically named faces. As expected, fully supervised metrics further improve over MIL metrics. This work is published in Guillaumin et al. [2010c] and presented in Chapter 3.

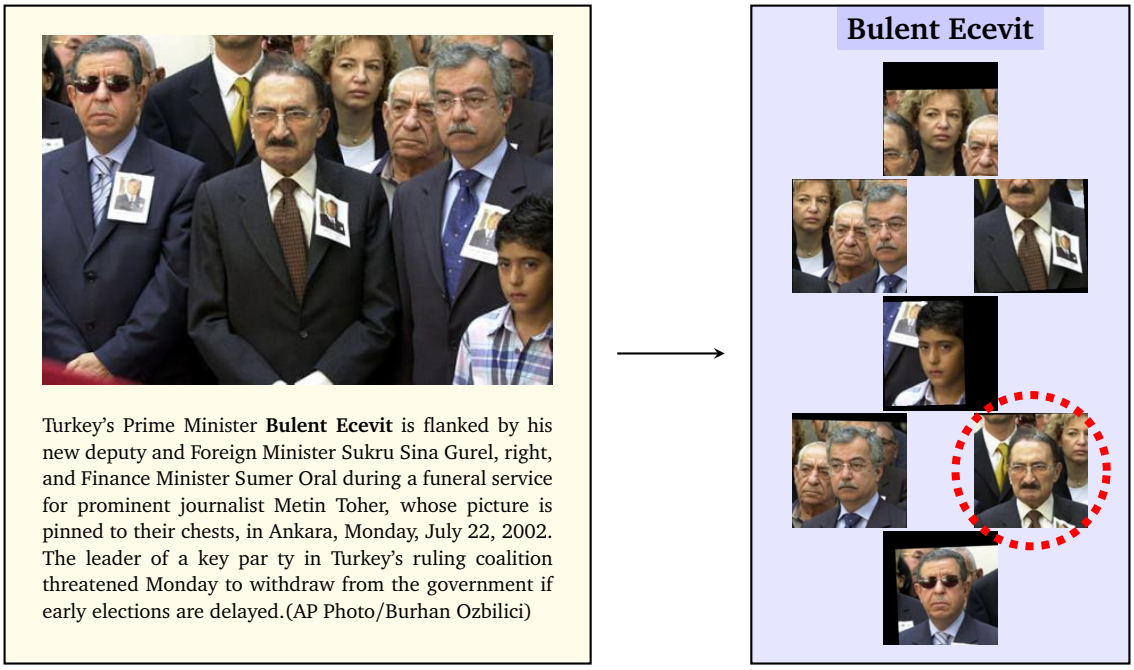


***Figure 1.10**: Multiple instance point of view of news documents as used in our proposed Multiple Instance multiple label Logistic Discriminant-based Metric Learning (Mild-ML) algorithm. The image is considered as an unordered bag of face images where at least one instance is positive for the bag labels, obtained from extracting named entities from the caption. The positive instance in the bag is highlighted with a red contour.*

- For the problem of image auto-annotation, we propose to annotate an image using the tags of its most similar images. We introduce TagProp, short for Tag Propagation, a probabilistic model for weighted nearest neighbour tag prediction, as illustrated in Figure 1.11. TagProp differs from other local approaches in that it automatically sets the neighbourhood size to use and is able to optimise the linear combination of several image representations to obtain the neighbours that best predicts the image tags. For both image auto-annotation and keyword-based retrieval, we show state-of-the-art performance on several challenging data sets. This work was published in Guillaumin et al. [2009a] and has led to excellent results in the ImageCLEF 2009 competition (*c.f.* Douze et al. [2009]). We present it in Chapter 4.

- Finally, we address the more general problem of visual object class recognition. Using a data set of images with associated tags, we study the training of visual models from the weak supervision provided by image tags. We also consider a semi-supervised scenario where we assume presence at train time of multimodal data, but only partially manually labelled, while only images are present

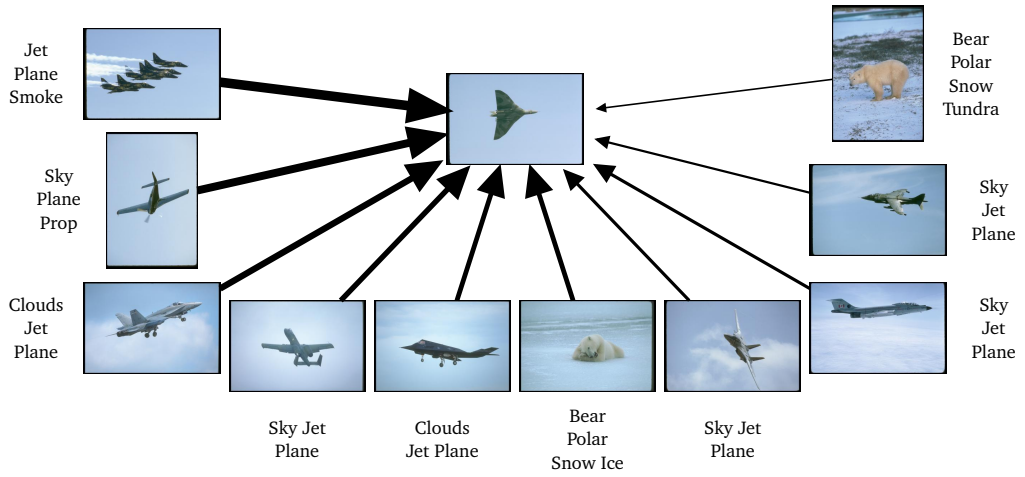


***Figure 1.11**: Weighted nearest neighbour tag prediction model in TagProp. The probability of relevance of a tag for an image can be viewed as the weighted average of presence of this tag in the image neighbourhood.*

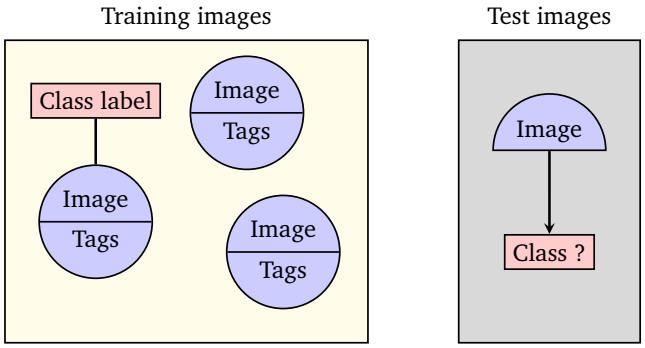


***Figure 1.12**: Overview of the multimodal semi-supervised setting that we consider in Chapter 5. Training images come with tags, and only a subset is labelled. The goal is to predict the class label of test images without tags.*

at test time. A schematic overview of this *multimodal semi-supervised learning* framework is given in Figure 1.12. For dealing with such a setting, the idea is to use the stronger classifiers that can be learnt from multimodal data to label unlabelled data and augment the training set for the visual classifier. Specifically, we use a non-linear visual kernel to regress the score function obtained using the Multiple Kernel Learning (MKL) framework (Lanckriet et al. [2004]) to combine visual and textual kernels. We evaluate the performance of our approaches with varying levels of supervision, as published in Guillaumin et al. [2010b]. This work in presented in Chapter 5.

2

Metric learning for face recognition

Contents

2.1 Introduction

In this chapter, we address the problem of face recognition in uncontrolled settings. This work was published in Guillaumin et al. [2009b] and Guillaumin et al. [2010a]. Recently there has been considerable interest for face verification. Compared to other object categories like cars, aeroplanes, or other rigid objects, human faces are particularly challenging due to their muscular flexibilities: the Facial Action Coding System by Ekman and Friesen [1978] identifies 46 muscles in an attempt to build a taxonomy of facial expressions. Combined with pose changes, the possible variations in appearance are very strong. The quantity of available resources, data sets and published methods highlights how important the topic is, not only for the computer vision community, but also neuroscientists and psychologists. For an extensive and up-to-date list of publications, one can for instance refer to the Face Recognition Homepage[1].

Since we are interested in exploiting the quantity of data available online, it is important to realise first that the face images on the Internet are quite different from those

[1] URL: `http://www.face-rec.org/`

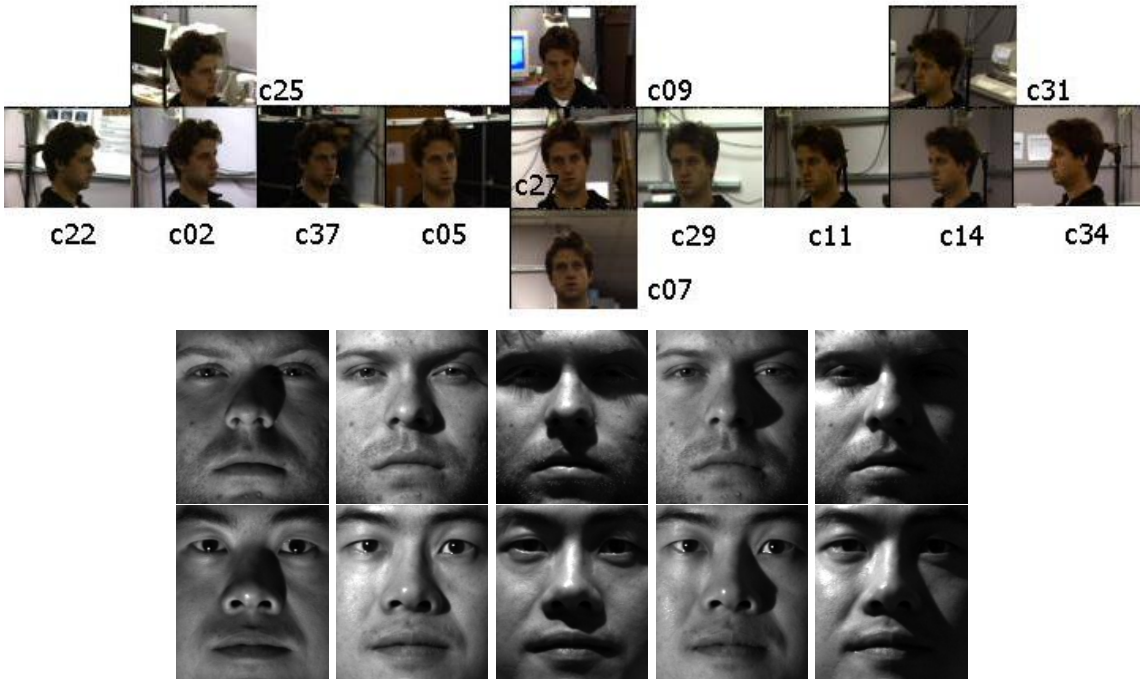


***Figure 2.1**: PIE (top) and YaleB (bottom) data sets respectively show high pose and illumination variations, but are controlled settings.*

found in usual face recognition data sets. Common face data sets include FERET[2], Yale B[3] or CMU PIE[4] and they typically have constrained environments: no or little background clutter, no or little pose and expression changes, controlled light, and, above all, medium to high resolution images, as shown in Figure 2.1. To help build systems that could potentially be applied to any image on the web, and for instance allow automatic tagging of faces in photographs on social websites such as Facebook, we have to resort to data sets of uncontrolled face images. This is the case of the recent the *Labeled Faces in the Wild* data set (LFW, Huang et al. [2007b]). Example images from LFW can be seen in Figure 2.2, and they show large variations in pose, expression, lighting and occlusions, and also changes in scale, although smaller. Obviously, these changes make the recognition task even harder.

Originally, the *Labeled Faces in the Wild* data set was designed to evaluate face verification systems. Face verification is a binary classification problem over pairs of face images: we have to determine whether or not the same person is depicted in both images. More generally, visual verification refers to the problem of deciding whether or not two images depict objects from the same class. The confidence scores, or *a pos-*

[2]URL: `http://face.nist.gov/colorferet/`
[3]URL: `http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html`
[4]URL: `http://www.ri.cmu.edu/research_project_detail.html?project_id=418`

***Figure 2.2**: Illustration of face images of the Labeled Faces in the Wild data set. Images of the same person show large variations in pose, expression, lighting and occlusions.*

teriori class probabilities, for the visual verification problem can be thought of as an object-category-specific similarity measure between instances of the category. Ideally it is 0 for images of different instances, and 1 for images of the same object. Importantly, similarity scores can also be applied for other problems such as visualisation, recognition from a single example (Li et al. [2006]), retrieval (Mensink and Verbeek [2008], Ozkan and Duygulu [2010]), associating names and faces in images (Berg et al. [2004a], Pham et al. [2010]) or video (Everingham et al. [2006], Sivic et al. [2009]), or people-oriented topic models (Jain et al. [2007]). This is a direct consequence of those systems relying on good face similarities. Moreover, face similarities are likely to be at the heart of face recognition technologies in applications such as Picasa or iPhoto.

To address the problem of visual verification, we propose two approaches. The first consists in using metric learning techniques to optimise the distance measure between data points to match the semantic similarity. The intuition is that we want the distance between similar objects to be smaller than the distance between different objects. For faces, the metric should suppress differences due to pose, expression, lighting conditions, clothes, hair style, sun glasses while retaining the information relevant to identity. Specifically, we introduce an algorithm to optimise the metric based on logistic discriminant that we refer to as LDML, for Logistic Discriminant-based Metric Learning. We also propose to extend nearest neighbour classification for the classification of pairs of data whose class is not present in the training set. We do this by marginalising over which class it might be, leading to MkNN, for Marginalised k-Nearest Neighbour.

In Section 2.2, we review the current state of the art in visual verification and metric learning, and we present in Section 2.3 our approaches for solving the visual verification problem using metric learning and nearest neighbour classification. Then, in Section 2.4, we describe in more details the *Labeled Faces in the Wild* data set and present a feature extraction procedures for face description. We evaluate those procedures and our methods in Section 2.5 for face verification, then apply the similarity

scores to face recognition from one exemplar, and face clustering problems, and we conclude in Section 2.6.

2.2 Related work on verification and metric learning

Face recognition is a very promising application of computer vision, and therefore a wide subject of research. Faces in uncontrolled settings are involved in many other work, which are related to our approaches. For instance, Holub et al. [2008] addresses the problem of clustering images of celebrities found on the web, while Everingham et al. [2006] perform character naming in videos, and Mensink and Verbeek [2008] and Kumar et al. [2008] focus on finding the faces of people in large data sets.

Generally, the first task is to obtain a good face descriptor. An example of successful descriptor for constrained face recognition (*c.f.* Ahonen et al. [2004]) is the Local Binary Pattern (LBP, Ojala et al. [2002]), which we describe later in Section 2.4.2.

Similar descriptors have been used for unconstrained verification. For instance, in Wolf et al. [2008], the authors extended LBP and adapted linear discriminant analysis to score a pair of faces. In their work, the binary task of face pair classification is seen as two binary classifications from one example, one for each face of the pair. A linear discriminant-like classifier is learnt using either face as the only positive face and using the entire training data as the negative class, and this system scores the left-out image. The two scores are averaged to give the image pair a confidence score for the verification task.

Another and more recent approach by Kumar et al. [2009] uses 65 attribute classifiers, the outputs of which form the face descriptor. The attributes are specific to faces, such as hair colour, gender, age, hair style, etc., and the classifiers are learnt on an external data set collected from Google Image and manually annotated using the Amazon Mechanical Turk interface. Although this system performs excellently on the *Labeled Faces in the Wild* data set, we note that the use of external data does not allow for a totally fair comparison with most of the other published methods.

Pinto et al. [2009], who proposes to use advanced machine learning techniques such as Multiple Kernel Learning (MKL) on simple visual features inspired by biological vision (V1-like features), also obtains state-of-the-art results.

A common line of approach though, which is also ours, is to identify a lower dimensional space for representing the faces, like in the seminal work of Turk and Pentland [1991], which used principal components analysis (PCA), and Belhumeur et al. [1996], which used linear discriminant analysis (LDA), or later work such as Sirovich and Kirby [1987], He et al. [2005]. In the approach of Chopra et al. [2005] for face

verification, a convolutional network-based projection is learnt discriminatively such that the distances between the projections of faces of the same person are minimised.

For the general problem of visual verification, Nowak and Jurie [2007] proposed an original method for learning the discriminative features and a binary classifier to score the image pairs. The idea is to sample random patches in the images and to match them to similar patches in similar regions in the other image of the pair, hence obtaining patch pairs from image pairs. Positive image pairs provide positive patch pairs, and respectively for negative image pairs. The patches are described using SIFT, and the local differences between the descriptors of the patches of a pair are then quantised using an extremely randomised forest. An image pair is then represented as the binary histogram of quantised patch pairs obtained from the patch sampling procedure. The histograms are then fed to a binary SVM (Vapnik [1995]) for classification, *i.e.* the output score of the SVM is used as a similarity measure between the images of the pair. The approach is very general and has shown state-of-the-art performance for cars, faces and other object instances.

Similarly, Jain et al. [2006] and Ferencz et al. [2008] try to locate and learn the features that are most discriminative for the particular task at hand. Like Nowak and Jurie [2007], these works can handle any visual verification task. The main drawback of these methods is the time-consuming learning process that results from the optimisation of the global verification pipeline.

Distance functions, because of their functional form $\mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^+$, are a natural approach for binary classification tasks in vectorial spaces $\mathbb{R}^D$. Although more general similarities are also investigated (*c.f.* Qamar et al. [2008]), metric learning has recently received a lot of attention, for recent work in this area see *e.g.* Xing et al. [2002], Bar-Hillel et al. [2005], Davis et al. [2007], Frome et al. [2007], Globerson and Roweis [2005], Weinberger et al. [2005]. Most methods learn a Mahalanobis metric based on an objective function defined by means of a labelled training set, or from sets of positive (same class) and negative (different class) pairs. The underlying idea is to find a distance function such that positive pairs have smaller distance values than negative ones, as illustrated in Figure 2.3. These metrics can be designed in an ad-hoc fashion, set heuristically, or learnt from manually annotated data.

The difference between the methods mainly lies in their objective functions, which are designed for their specific tasks (clustering for Xing et al. [2002], kNN classification for Weinberger et al. [2005], etc.). Some methods explicitly need all pairwise distances between points (Globerson and Roweis [2005]), making large scale applications (say more than ten thousand data points) more difficult. Among the existing methods for learning metrics, large margin nearest neighbour (LMNN, Weinberger et al. [2005]) and information theoretic metric learning (ITML, Davis et al. [2007]) are state-of-the-art.

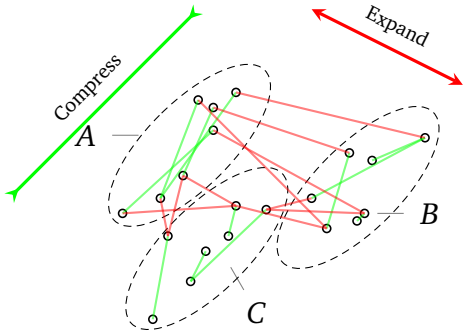


***Figure 2.3**: Illustration of learning a metric with the goal of having smaller distances for positive pairs of data (points that have the same class A, B or C, with some pairs shown in green) and larger distances for negative pairs (shown in red). The learnt metric should compress the direction shown with the green arrow while expanding the direction indicated by the red arrow.*

Below, in Section 2.2.1, we first highlight some well-known properties of Mahalanobis metrics. Then, we provide some technical details about the following techniques: unsupervised metrics such as PCA in Section 2.2.2, and, in Section 2.2.3, supervised techniques such as LDA, LMNN and ITML.

2.2.1 Mahalanobis metrics

Given the vectorial representation $\mathbf{x}_i \in \mathbb{R}^D$ of the data points (indexed by i), we seek to design good metrics for verification. As stated above, the optimisation of the metric is typically done over a parametrised class of distances called Mahalanobis metrics that generalises the Euclidean distance. The (squared) Mahalanobis distance between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \tag{2.1}$$

$$= \sum_{k=1}^{D}\sum_{l=1}^{D} \mathbf{M}_{kl}(\mathbf{x}_{ik} - \mathbf{x}_{jk})(\mathbf{x}_{il} - \mathbf{x}_{jl}) \geq 0 \tag{2.2}$$

where $\mathbf{M} \in \mathbb{R}^{D\times D}$ is a symmetric positive semi-definite matrix that parametrises the distance.

The set of symmetric positive semi-definite matrices of size D, noted $\mathcal{S}_D^+$, is well-known to be a cone in the space of $D \times D$ symmetric matrices. Therefore, the constraint $\mathbf{M} \in \mathcal{S}_D^+$ is convex (Boyd and Vandenberghe [2004]). Importantly, Mahalanobis metrics also correspond to the Euclidean distance after a linear transformation of the input

space. Clearly, from a linear mapping represented by a matrix $\mathbf{U} \in \mathbb{R}^{D\times D}$, we can write:

$$||\mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j||^2 = (\mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j)^\top(\mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j) \tag{2.3}$$
$$= (\mathbf{x}_i - \mathbf{x}_j)^\top\mathbf{U}^\top\mathbf{U}(\mathbf{x}_i - \mathbf{x}_j) \tag{2.4}$$
$$= d_{\mathbf{U}^\top\mathbf{U}}(\mathbf{x}_i, \mathbf{x}_j), \tag{2.5}$$

which is well-defined because $\mathbf{U}^\top\mathbf{U} \in \mathcal{S}_D^+$.

Conversely, using the eigenvalue decomposition of $\mathbf{M}$, we can write

$$\mathbf{M} = \mathbf{V}^\top\mathbf{\Lambda}\mathbf{V} \tag{2.6}$$

where $\mathbf{V}$ is orthonormal and $\mathbf{\Lambda}$ is diagonal with positive diagonal values. By setting

$$\mathbf{U} = \mathbf{\Lambda}^{1/2}\mathbf{V} \tag{2.7}$$

we see that any semi-definite positive matrix corresponds to a linear mapping $\mathbf{U}$ such that $d_\mathbf{M}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j||^2$. Learning the Mahalanobis distance can be equivalently performed by optimising $\mathbf{U}$ or $\mathbf{M}$ directly.

Equivalently, any method for learning linear projections of the data can be considered as metric learning. For instance, Principal Components Analysis (PCA, Pearson [1901]) or Linear Discriminant Analysis (LDA, Fisher [1936]) are well-known methods that are forms of unsupervised and supervised metric learning, respectively. We describe them below in Section 2.2.2 and Section 2.2.3, respectively.

2.2.2 Unsupervised metrics

A natural baseline metric is obtained by fixing $\mathbf{M}$ to be the identity matrix. This results simply in the Euclidean distance (L_2) between the vectorial representations of the faces.

A more elaborate method sets $\mathbf{U}$ using PCA, which is very common in data analysis. The basic idea is to find a linear projection $\mathbf{U}$ that retains the highest possible amount of the data variance. This unsupervised method can improve the performance of face recognition by making the face representation more robust to noise. These new representations are typically much more compact then the original ones, hence making easier the use of methods, like metric learning techniques, that would otherwise scale with the square of the data dimensionality. The pre-processing of the data using a PCA projection to lower dimensional space acts as a rank (*i.e.*, subspace) constraint on the metric, effectively regularising the metric parameters.

There are several alternate ways to define and derive PCA (Bishop [2006]), such as finding the projections that minimise the square reconstruction error, but let us focus on finding an optimal projection vector $\mathbf{v} \in \mathbb{R}^D$ such that the highest quantity of data variance is retained.

Given the mean $\overline{\mathbf{x}}$ and covariance matrix $\mathbf{S}$ of N data points $\mathbf{x}_i$ defined with:

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{2.8}$$

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top, \tag{2.9}$$

we seek to maximise the variance of the projected data as expressed by:

$$\frac{1}{N}\sum_{i=1}^{N}(\mathbf{v}^\top\mathbf{x}_i - \mathbf{v}^\top\overline{\mathbf{x}})^2 = \mathbf{v}^\top\mathbf{S}\mathbf{v}. \tag{2.10}$$

By limiting the search to unit vectors, *i.e.* $\mathbf{v}^\top\mathbf{v} = \|\mathbf{v}\|^2 = 1$, because we are only interested in projection directions, we can use the positive definiteness of $\mathbf{S}$ and denote by λ_1 the largest eigenvalue of $\mathbf{S}$ to obtain the following inequality:

$$0 \leq \mathbf{v}^\top\mathbf{S}\mathbf{v} \leq \lambda_1. \tag{2.11}$$

And the maximum is therefore achieved for $\mathbf{v}_1$ such that the equality ${\mathbf{v}_1}^\top\mathbf{S}\mathbf{v}_1 = \lambda_1$ holds, which is the case when $\mathbf{v}_1$ is an eigenvector associated to the eigenvalue λ_1. More generally, the first d PCA projections are obtained from d unit eigenvectors associated to the d largest eigenvalues $\lambda_1, \ldots, \lambda_d$ of $\mathbf{S}$. These projections, for face data, are known as eigenfaces (Turk and Pentland [1991]).

The main drawback of PCA is that the dimensions of large data variance are not necessarily the ones corresponding to *important* variations for verification. Therefore, the resulting compression could discard important discriminative information by focusing on data variance. An alternative to PCA consists in effectively using the class labels to find projections that help discriminate between the different classes. Those methods are, by nature, supervised.

2.2.3 Supervised metric learning

Instead of trying to retain as much data variance as possible when projecting the data as in PCA, the idea of supervised methods is to find projections that best separate

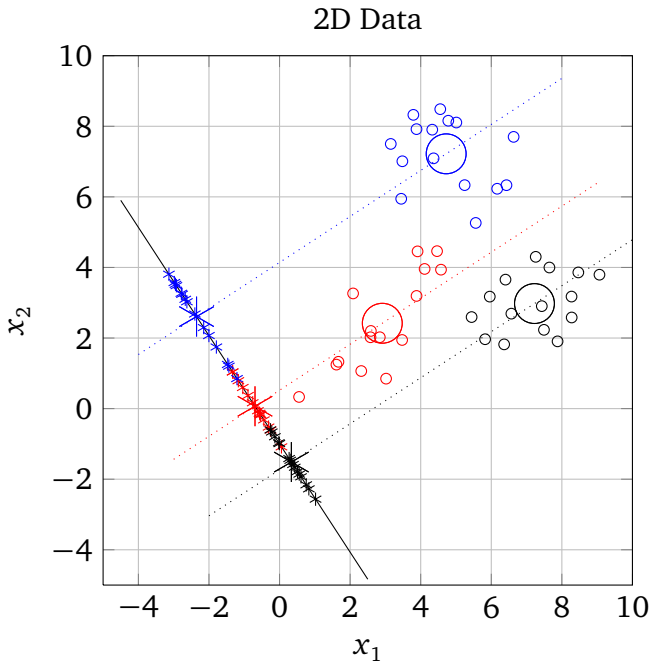


***Figure 2.4**: Illustration of Fisher's discriminant analysis for supervised dimensionality reduction. The 2D data points (represented as circles) of three classes (colour coded) are projected such that the variances within each class is minimised while the variance between the class centres (represented with larger markers) is maximised. The 1D projected points are represented as stars on the projection line.*

the different classes present in the data, as provided by manual annotation. One of those methods is the Linear (or Fisher's) Discriminant Analysis (LDA). The projection should maximise the distance between the classes, chosen as the distance between projected class means, and minimise the variance within each class, as illustrated in Figure 2.4.

For each of the classes $c \in \{1, \ldots, C\}$, we denote by Ω_c the set of points belonging to class c, N_c its number of points, $\overline{\mathbf{x}}_c$ its mean and $\mathbf{S}_c$ the intra-class variation, with the following definitions:

$$\overline{\mathbf{x}}_c = \frac{1}{N_c} \sum_{i \in \Omega_c} \mathbf{x}_i \tag{2.12}$$

$$\mathbf{S}_c = \sum_{i \in \Omega_c} (\mathbf{x}_i - \overline{\mathbf{x}}_c)(\mathbf{x}_i - \overline{\mathbf{x}}_c)^\top. \tag{2.13}$$

We then can measure the *within-class* covariance $\mathbf{S}_W$ and *between-class* covariance $\mathbf{S}_B$ the following way:

$$\mathbf{S}_W = \sum_{c=1}^{C} \mathbf{S}_c \tag{2.14}$$

$$\mathbf{S}_B = \sum_{c=1}^{C} N_c (\overline{\mathbf{x}}_c - \overline{\mathbf{x}})(\overline{\mathbf{x}}_c - \overline{\mathbf{x}})^\top \tag{2.15}$$

The objective is to maximise the *between-class* covariance of the projected data while minimising the *within-class* covariance of the same projected data. To this end, Fisher proposed to minimise:

$$J(\mathbf{v}) = \frac{\mathbf{v}\mathbf{S}_B\mathbf{v}^\top}{\mathbf{v}\mathbf{S}_W\mathbf{v}^\top}, \tag{2.16}$$

which leads to a generalised eigenvalue problem (see also: Fukunaga [1990]). The resulting projections correspond to the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ corresponding to the largest eigenvalues. These projections, for face data, are known as Fisherfaces (Belhumeur et al. [1996]).

In the following paragraphs, we describe other options to model the problem and find discriminative projections $\mathbf{U}$ or metrics $\mathbf{M}$ in a supervised fashion. This includes large margin nearest neighbours (LMNN, Weinberger et al. [2005]) and information theoretic metric learning (ITML, Davis et al. [2007]). For image i, let us denote by y_i its class label. Therefore, images i and j form a positive pair if $y_i = y_j$, and a negative pair otherwise.

Large Margin Nearest Neighbour Metrics

Recently, Weinberger et al. [2005] introduced a metric learning method, that learns a Mahalanobis distance metric designed to improve results of k-nearest neighbour (kNN) classification. A good metric for kNN classification should make for each data point the k nearest neighbours of its own class closer than points from other classes. To formalise, we define target neighbours of $\mathbf{x}_i$ as the k closest (according to a given metric) points $\mathbf{x}_j$ with $y_i = y_j$, let $\eta_{ij} = 1$ if $\mathbf{x}_j$ is a target neighbour of $\mathbf{x}_i$, and $\eta_{ij} = 0$ otherwise. Furthermore, let $\rho_{ij} = 1$ if $y_i \neq y_j$, and $\rho_{ij} = 0$ otherwise.

The objective function is:

$$\varepsilon(\mathbf{M}) = \sum_{i,j} \eta_{ij} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i,j,l} \eta_{ij}\rho_{il} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_l)\right]_+, \tag{2.17}$$

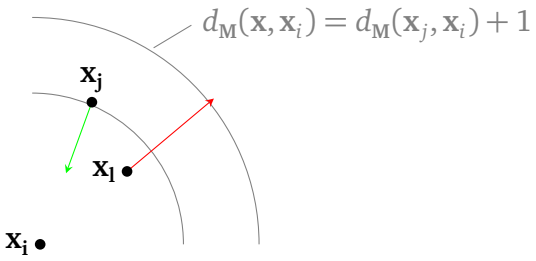


***Figure 2.5**: Target neighbours of $\mathbf{x}_i$ are the k closest points of the same class, here $\mathbf{x}_j$. Those target neighbours are pursued to be closer to $\mathbf{x}_i$ than points from another class (shown as green arrow) by a margin of at least one unit distance (shown by the concentric circles). The constraints of LMNN only apply for a couple of neighbours $(\mathbf{x}_j, \mathbf{x}_l)$ such that $\mathbf{x}_j$ is a target neighbour of $\mathbf{x}_i$ and $\mathbf{x}_l$ is of a different class but not further by more than one unit distance from $\mathbf{x}_i$ than $\mathbf{x}_j$. The impostor neighbour $\mathbf{x}_l$ is pushed away from $\mathbf{x}_i$ (shown as red arrow).*

where $[z]_+ = \max(z, 0)$.

The first term of this objective minimises the distances between target neighbours, whereas the second term is a hinge-loss that encourages target neighbours to be at least one distance unit closer than points from other classes. See an illustration of the objective in Figure 2.5. The objective is convex in $\mathbf{M}$ and can be minimised using sub-gradient methods under the constraint that $\mathbf{M}$ is positive semi-definite, and using an active-set strategy for efficiently handling the large number of constraints.

Rather than requiring pairs of images labelled positive or negative, this method requires labelled triples (i, j, l) of target neighbours (i, j) and points which should not be neighbours (i, l). In practice this method[5] uses labelled training data $(\mathbf{x}_i, y_i)$, and implicitly uses all pairs although many never appear as active constraints.

The cost function is designed to yield a good metric for kNN classification, and does not try to make all positive pairs have smaller distances than negative pairs. Therefore, directly applying a threshold on this metric for visual verification might not give optimal results but they are nevertheless very good.

Information Theoretic Metric Learning

Davis et al. [2007] have taken an information theoretic approach to optimise $\mathbf{M}$ under a wide range of possible constraints and prior knowledge on the Mahalanobis distance. This is done by regularising the matrix $\mathbf{M}$ such that it is as close as possible

[5] Code is available at `http://www.cse.wustl.edu/~kilian/code/code.html`.

to a known prior $\mathbf{M}_0$. This closeness is interpreted as a Kullbach-Leibler (KL) divergence between the two multivariate Gaussian distributions corresponding to $\mathbf{M}$ and $\mathbf{M}_0$: $p(\mathbf{x};\mathbf{M})$ and $p(\mathbf{x};\mathbf{M}_0)$. The KL divergence between two probability distributions P and Q, defined over set X and with respective densities p and q, is defined by:

$$\mathrm{KL}(P||Q) = \int_X p(x) \log \frac{p(x)}{q(x)} dx. \tag{2.18}$$

The constraints that can be used to drive the optimisation include those of the form: $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq u$ for positive pairs and $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq l$ for negative pairs, where u and l are constant values. Scenarios with unmanageable constraints are avoided by introducing slack variables $\boldsymbol{\xi} = \{\xi_{i,j}\}$. The objective function has the following form:

$$\begin{aligned} \underset{\mathbf{M}\geq 0, \xi}{\text{minimise}} \quad & \mathrm{KL}\left(p(\mathbf{x};\mathbf{M}_0)||p(\mathbf{x};\mathbf{M})\right) + \gamma \cdot f(\boldsymbol{\xi}, \boldsymbol{\xi}^0) \\ \text{subject to} \quad & d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \xi_{i,j} \quad \text{for positive pairs} \\ \text{or} \quad & d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq \xi_{i,j} \quad \text{for negative pairs,} \end{aligned} \tag{2.19}$$

where f is a loss function between $\boldsymbol{\xi}$ and target $\boldsymbol{\xi}^0$ that contains $\xi^0_{i,j} = u$ for positive pairs and $\xi^0_{i,j} = l$ for negative pairs, and γ is a regularisation multiplier that controls the trade-off between satisfying the constraints and using $\mathbf{M}_0$ as metric.

The parameters $\mathbf{M}_0$ and γ have to be provided, although it is also possible to resort to cross-validation techniques. Usually, $\mathbf{M}_0$ can be set to the identity matrix to regularise $d_{\mathbf{M}}$ to the Euclidean distance.

The proposed algorithm[6] scales with $O(CD^2)$ where C is the number of constraints on the Mahalanobis distance. From the labelled data, we define N^2 constraints $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq b$ for positive pairs and $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq b$ for negative pairs, and we set $b = 1$ as the decision threshold. The complexity is therefore $O(N^2D^2)$.

2.3 Our approaches for face verification

For the problem of visual verification, we propose two methods based on learning Mahalanobis metrics over a given vectorial feature space. The first method, LDML, uses logistic discriminant to learn a metric from a set of labelled image pairs. Its objective is to find a metric such that positive pairs have smaller distances than negative pairs, as already illustrated in Figure 2.3. LDML is presented in Section 2.3.1. The second method, MkNN, uses a set of labelled data, and is based on marginalising a

[6] Code is available at `http://www.cs.utexas.edu/users/pjain/itml/`.

k-nearest neighbour (kNN) classifier for both data points of a pair. The MkNN classifier computes the marginal probability that the two points are of the same class, *i.e.* marginalising over which one that exactly is. This enables to classify pairs of data of potentially unseen classes. For this second method we also use a learnt metric, albeit one that is optimised for kNN classification, LMNN, by Weinberger et al. [2005], already detailed in the previous section. We present MkNN in Section 2.3.2. This work, applied to face verification, was published in Guillaumin et al. [2009b] and LDML is available online.[7]

2.3.1 Logistic discriminant-based metric learning

In this section, we proposed a method, similar in spirit to Davis et al. [2007], that learns a metric from labelled pairs. The model is based on the intuition that we would like the distance between images in positive pairs, *i.e.* images i and j such that $y_i = y_j$ (we also denote this property with $t_{ij} = 1$), to be smaller than the distances corresponding to negative pairs ($t_{ij} = 0$). Using the Mahalanobis distance between two images as given by Equation 2.1, we define the probability p_{ij} that they contain the same object as:

$$p_{ij} = p(t_{ij}|\mathbf{x}_i, \mathbf{x}_j; \mathbf{M}, b) = \sigma(b - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)), \tag{2.20}$$

where b is a bias term and the sigmoid function σ is defined by:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}. \tag{2.21}$$

Interestingly for the visual identification task, the bias directly works as a threshold value and is learnt together with the distance metric parameters. Notably, b is the distance value which leads to the probability $\sigma(0) = 0.5$ that the two images are of the same class.

The objective of logistic discriminant-based metric learning (LDML) is the maximum likelihood estimation for the parameters $\mathbf{M}$ and b for a set of training pairs (i, j). The log-likelihood function $\mathcal{L}$ of such data is given by:

$$\mathcal{L}(\mathbf{M}, b) = \sum_{i,j} t_{ij} \log p_{ij} + (1 - t_{ij}) \log(1 - p_{ij}) \tag{2.22}$$

[7] Our code is available at `http://lear.inrialpes.fr/software/`

The gradients of which, with respect to $\mathbf{M}$ and b respectively, are given by:

$$\nabla \mathcal{L}(\mathbf{M}) = \sum_{i,j} (p_{ij} - t_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{2.23}$$

$$\nabla \mathcal{L}(b) = \sum_{i,j} (t_{ij} - p_{ij}) \tag{2.24}$$

By decomposing the expression of the distance value as in Equation 2.2, the linearity of $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ with respect to the components of $\mathbf{M}$ appears clearly. Taking into account the symmetry of $\mathbf{M}$, we can use the mappings in Equation 2.25 and Equation 2.26 to rewrite p_{ij} as in Equation 2.27.

$$\mathbf{w}_u = \mathbf{M}_{kl} \quad \text{for } k \geq l \text{ and } u = k + l(l-1)/2 \tag{2.25}$$

$$\phi_u^{ij} = \begin{cases} (\mathbf{x}_{ik} - \mathbf{x}_{jk})^2 & \text{for } k = l, \textit{ i.e. } u = k(k+1)/2, \\ 2(\mathbf{x}_{ik} - \mathbf{x}_{jk})(\mathbf{x}_{il} - \mathbf{x}_{jl}) & \text{for } k > l \text{ and } u = k + l(l-1)/2 \end{cases} \tag{2.26}$$

$$p_{ij} = \sigma(b - \mathbf{w}^\top \boldsymbol{\phi}^{ij}) \tag{2.27}$$

The maximum likelihood estimation of $\mathbf{M}$ and b is therefore a standard logistic discriminant model. This also highlights that from a linear decision on the distance measure, we obtain a quadratic decision in the space of data difference, and with the explicit mapping in Equation 2.26, a linear classifier in a higher $D(D+1)/2$-dimensional space, as shown in Figure 2.6.

Logistic discriminant models are well-known to have a convex objective function for maximum likelihood estimation, so, together with the convexity of $\mathcal{S}_D^+$, the optimisation problem is convex and can be solved using interior-point methods (Boyd and Vandenberghe [2004]) or projected gradient ascent (Bertsekas [1976], Bertsekas et al. [1995]).

Intuitively, the projected gradient ascent is required because a simple gradient ascent on $\mathbf{M}$ would not guarantee to yield a semi-definite matrix. During the gradient ascent, it is therefore required to project $\mathbf{M}$ orthogonally on the SDP cone $\mathcal{S}_D^+$. This is typically done by decomposing $\mathbf{M}$ using the eigenvalue decomposition (*c.f.* Equation 2.6), which writes:

$$\mathbf{M} = \mathbf{V}^\top \mathbf{\Lambda} \mathbf{V} \tag{2.28}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with sorted eigenvalues, and $\mathbf{V}$ is a unitary matrix. If we define the diagonal matrix $\mathbf{\Lambda}^+$ such that:

$$\Lambda_{ii}^+ = \max(0, \Lambda_{ii}), \tag{2.29}$$

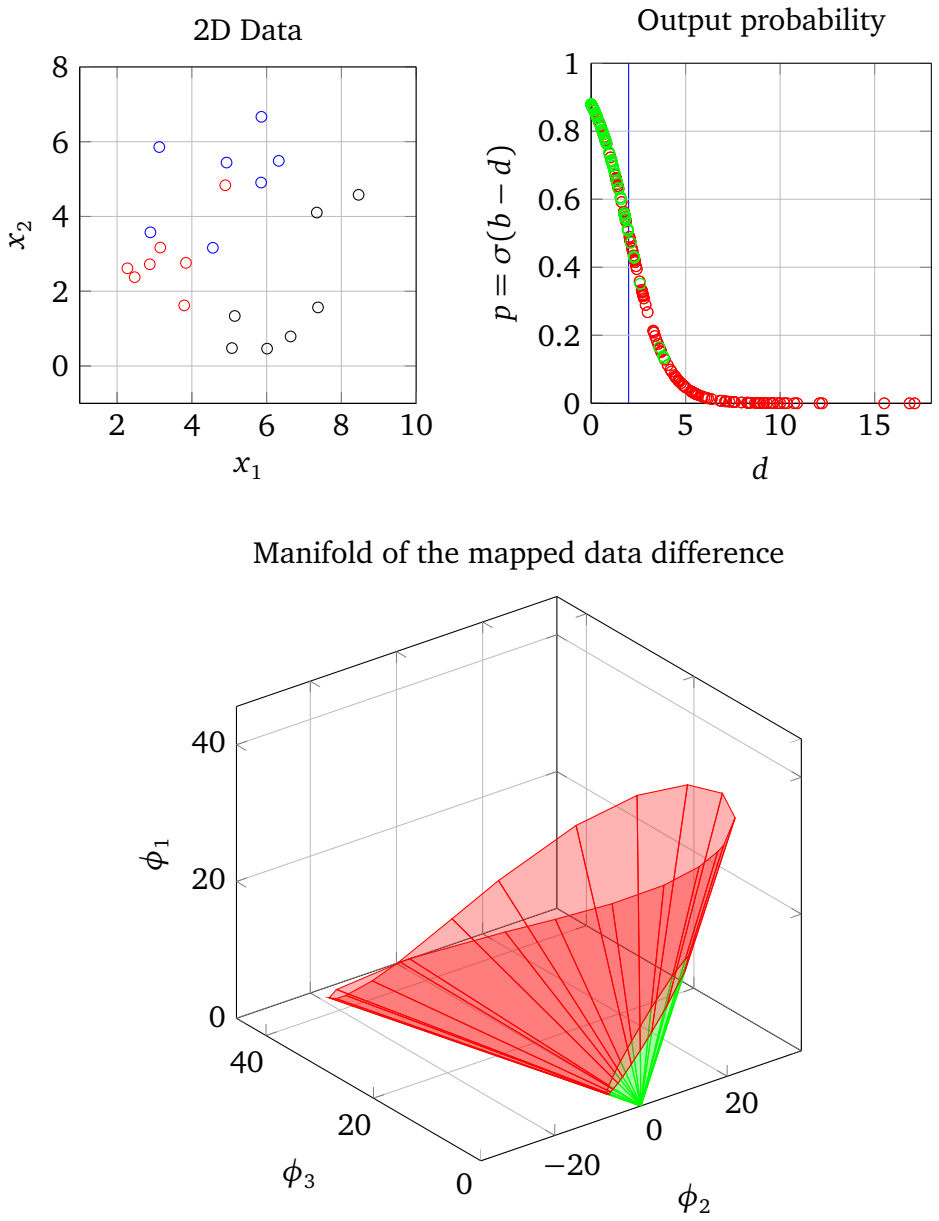


***Figure 2.6**: In the plot in the top left, we consider 21 data points in $\mathbb{R}^2$ from 3 classes as shown by colour. In the space of data difference (not shown), LDML is interpreted as a quadratic (hyper-ellipsoidal) decision boundary for the binary identification task. In the bottom figure, the same decision boundary is interpreted as a plane splitting the manifold of the mapped data difference (ϕ_1, ϕ_2, ϕ_3) (as given in Equation 2.26), with colours showing whether data pairs will be considered positive (green) or negative (red) depending on their position on the manifold. In the top right, we show the corresponding output probabilities from our logistic discriminant model with respect the learnt pairwise distance values, with colours showing the ground-truth labels (positive pairs in green, negative pairs in red).*

Then $\mathbf{\Lambda}^+$ is the projection of $\mathbf{\Lambda}$ on the PSD cone and the orthogonal projection $\mathcal{P}_{\mathcal{S}_D^+}(\mathbf{M})$ of $\mathbf{M}$ on $\mathcal{S}_D^+$is simply given by:

$$\mathcal{P}_{\mathcal{S}_D^+}(\mathbf{M}) = \mathbf{V}^\top \mathbf{\Lambda}^+ \mathbf{V}. \tag{2.30}$$

Since the eigenvalue decomposition of PSD matrices scales as $O(D^3)$ with the data dimensionality, the projected gradient ascent can only be considered for reasonably-sized data spaces. The computation of the gradient itself scales with $O(MD^2)$ where M is the number of observed pairs. When all the pairs from a data set of size N are considered, it is possible to compute this gradient more efficiently than $O(N^2D^2)$, by using the following algebra:

$$\nabla \mathcal{L}(\mathbf{M}) = \sum_{i,j}(p_{ij} - t_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \tag{2.31}$$

$$= \sum_{i,j}(p_{ij} - t_{ij})(\mathbf{x}_i\mathbf{x}_i{}^\top + \mathbf{x}_j\mathbf{x}_j{}^\top - \mathbf{x}_i\mathbf{x}_j{}^\top - \mathbf{x}_j\mathbf{x}_i{}^\top) \tag{2.32}$$

$$= 2\sum_{i,j}(p_{ij} - t_{ij})(\mathbf{x}_i\mathbf{x}_i{}^\top - \mathbf{x}_i\mathbf{x}_j{}^\top) \tag{2.33}$$

by symmetry of p_{ij} and t_{ij},

$$= 2\sum_i \mathbf{x}_i \left(\left(\sum_j t_{ij} - p_{ij} \right) \mathbf{x}_i{}^\top - \sum_j (t_{ij} - p_{ij})\mathbf{x}_j{}^\top \right) \tag{2.34}$$

$$= 2\sum_{i,j} \mathbf{x}_i (w_{ij} + p_{ij} - t_{ij})\mathbf{x}_j{}^\top \tag{2.35}$$

using $w_{ii} = \sum_j t_{ij} - p_{ij}$ and $w_{ij} = 0$ for $j \neq i$,

$$= 2\mathbf{X}\mathbf{H}\mathbf{X}^\top \tag{2.36}$$

where

$$\mathbf{X} = [\mathbf{x}_i] \in \mathbb{R}^{D\times N} \tag{2.37}$$

$$\mathbf{H} = [w_{ij} + p_{ij} - t_{ij}] \in \mathbb{R}^{N\times N}. \tag{2.38}$$

As already stated, the expression in Equation 2.31 can be computed with complexity $O(N^2D^2)$, whereas the matrix-form expression of Equation 2.36 is $O((N+D)ND)$. Using the already computed decomposition of $\mathbf{M}$ given in Equation 2.6 and having linearly transformed the data (which is $O(ND^2)$), the matrix $\mathbf{H}$ can be computed in $O(N^2D)$. The overall complexity is therefore $O((N+D)ND+D^3)$ instead of $O(N^2D^2+D^3)$. This is especially interesting considering the typical order of magnitudes for our data sets ($N \approx 10^4, D \approx 10^3$).

Rank regularisation for supervised dimensionality reduction

In the event of data separability with the quadratic decision boundary, the logistic discriminant model may suffer from severe over-fitting. Considering that we have $D(D+1)/2$ parameters, this event is quite likely. In practice, though, the gradient ascent is usually not performed until full convergence, but only with a limit for the maximum number of line searches allowed. However, regularisation as presented in Guillaumin et al. [2010c] is a better answer than this early stopping solution.

Among different possible regularisations for the metric, we will focus on rank regularisation. This means that the search for the optimal metric will be performed on a subset of $\mathcal{S}_D^+$, namely the set of matrices of rank lower or equal to d. A first option to obtain such a regularisation is to restrict the metric to the d-dimensional PCA subspace of the data. This method is applicable to any metric learning algorithm and also helps reducing their computational complexity, from D^2 to d^2.

An alternative solution is to obtain the maximum likelihood estimation of a low-rank matrix $\mathbf{M}$. Unfortunately, the rank constraint is *not* convex. Directly enforcing the rank of $\mathbf{M}$ with, for instance, eigenvalue decomposition is not theoretically sound to use together with gradient ascent. Instead, we can make use of the decomposition of $\mathbf{M}$ and learn $\mathbf{U}$ directly.

The advantage of learning $\mathbf{U}$ is that the semi-definite positiveness is naturally enforced, and therefore a simple gradient ascent will suffice. The drawback is that objective function is not concave anymore, and has many local maxima. Notably, the objective function is defined up to an isometry in the projection space. This isometry is not relevant in our setting, so in practice there is no particular problem. To avoid bad local maxima, we resort to multiple random initialisations.

The gradient of $\mathcal{L}$ with respect to $\mathbf{U}$ can be computed with the same algebra as in Equation 2.36, and is now:

$$\nabla \mathcal{L}(\mathbf{U}) = 4\mathbf{U}\mathbf{X}\mathbf{H}\mathbf{X}^\top. \tag{2.39}$$

Additionally, $\mathbf{U} \in \mathbb{R}^{d \times D}$ need not be a square matrix, which enforces that the resulting metric is low-rank, and in this case supervised dimensionality reduction is performed. The final complexity for computing the gradient is $O((N+D)Nd)$ where it was $O((N+D)ND + D^3)$ when learning the full-rank metric with projected gradient ascent. In Figure 2.7, we show the data distribution after projection on a 2D plane, comparing supervised dimensionality reduction and unsupervised PCA. As we can see, supervised dimensionality reduction is a powerful tool to grasp in low-dimensional spaces the important discriminative features useful for the identification task.

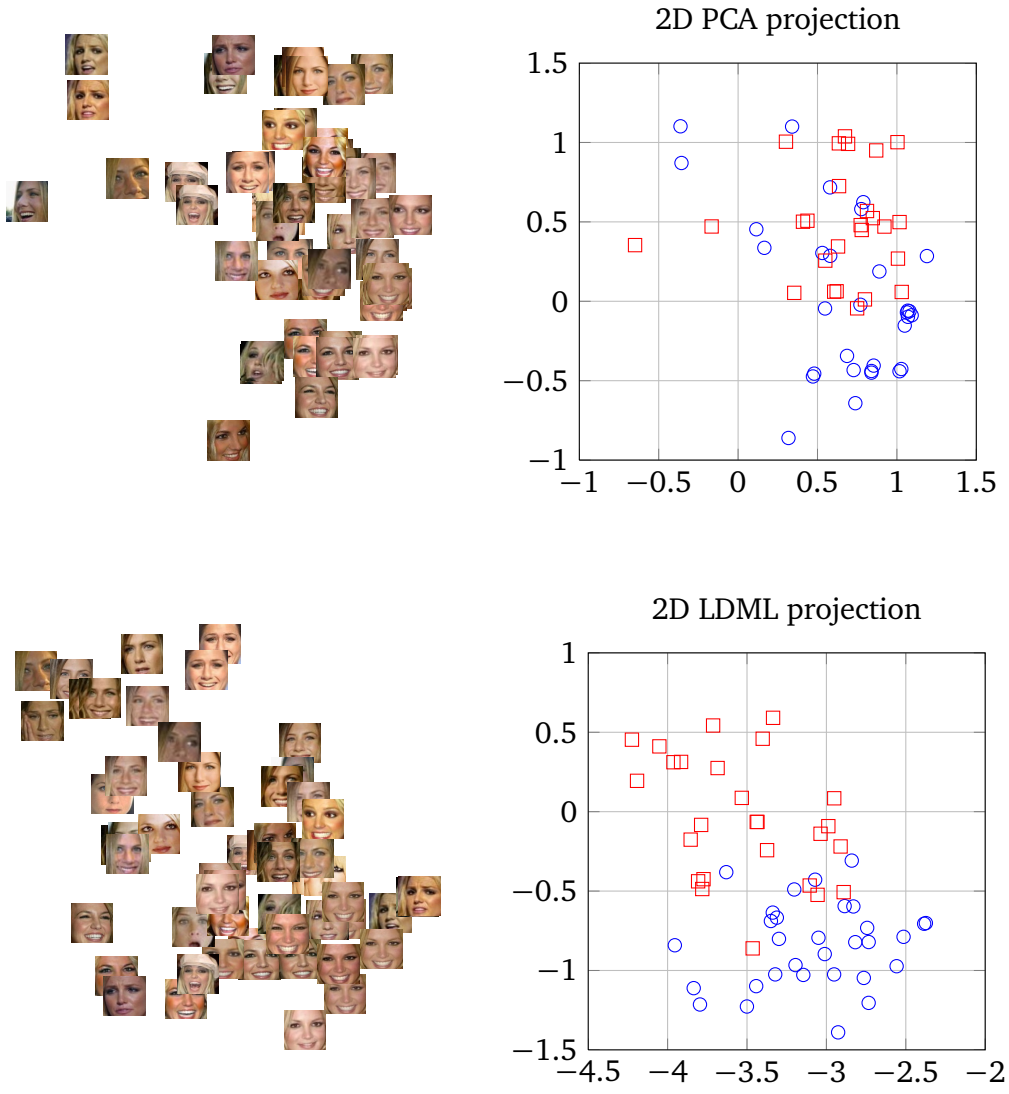


***Figure 2.7**: Comparison of PCA and LDML for 2D projection of data from the Labeled Yahoo! News data set (c.f. Section 3.4). For clarity, the data of only two persons are shown: Britney Spears and Jennifer Aniston. The cleaner separation of the identities obtained with LDML will improve many face related tasks such as the retrieval and naming.*

Kernelization

Note that without loss of generality, the rows of $\mathbf{U}$ can be restricted to be in the span of the columns of $\mathbf{X}^\top$. This is possible since this is true for the gradient (Equation 2.39) and since the Mahalanobis distance over the training data is invariant to perturbations of $\mathbf{U}$ in directions outside the span of $\mathbf{X}$. Hence, using $\mathbf{U} = \mathbf{A}\mathbf{X}^\top$, we can write the Mahalanobis distance in terms of inner products between data points, which allows us to use kernel functions to perform non-linear LDML like was done in Globerson and Roweis [2005]:

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{X}\mathbf{A}^\top \mathbf{A}\mathbf{X}^\top (\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{k}_i - \mathbf{k}_j)^\top \mathbf{A}^\top \mathbf{A}(\mathbf{k}_i - \mathbf{k}_j), \tag{2.40}$$

where $\mathbf{k}_i \in \mathbb{R}^N$ is the vector of inner products between $\mathbf{x}_i$ and the training data $\mathbf{X}$. Straightforward algebra shows that to learn the coefficient matrix $\mathbf{A}$ we simply replace $\mathbf{X}$ with the kernel matrix $\mathbf{K}$ in the learning algorithm, which will then output the optimised matrix $\mathbf{A}$. In the experiments, Section 2.5, we only report results using linear LDML; preliminary results using polynomial kernels did not show improvements over linear LDML for our data set of faces in uncontrolled settings.

2.3.2 Marginalised k-nearest neighbour classification

In the previous sections we presented methods to learn Mahalanobis metrics for visual identification, which are always linear transformations of an original space. With this limitation to hyper-ellipsoid decision boundaries on data difference (*c.f.* Figure 2.6), it may be impossible to separate positive and negative pairs, as appearance variations for a single person might be non-linear and larger than the inter-person variations for a similar pose and expression. In this section, we show how k-nearest neighbour classification can be used for visual identification. The resulting non-linear, high-capacity classifier implicitly uses all pairs that can be generated from the labelled data.

Normally, kNN classification is used to assign single data points $\mathbf{x}_i$ to one of a fixed set of k classes associated with the training data. The probability of class c for $\mathbf{x}_i$ is:

$$p(y_i = c|\mathbf{x}_i) = \frac{n_c^i}{k}, \tag{2.41}$$

where n_c^i is the number of neighbours of $\mathbf{x}_i$ of class c among the first k. Here, we have to predict whether a pair of images $(\mathbf{x}_i, \mathbf{x}_j)$ belongs to the same class, regardless of which class that is, and even if the class is not represented in the training data. To

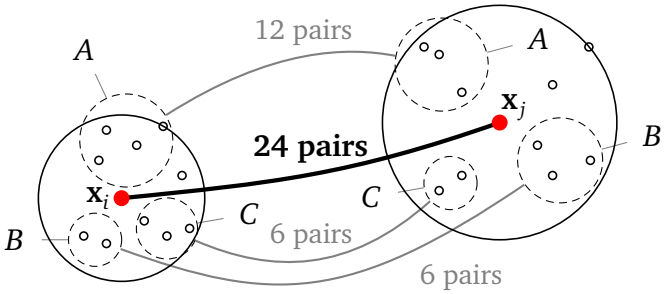


***Figure 2.8**: Schematic representation of $k = 10$ neighbours for $\mathbf{x}_i$ and $\mathbf{x}_j$, and the 24 neighbour pairs (out of 100) that have the same name and contribute to the score.*

answer this question we compute the marginal probability that we assign $\mathbf{x}_i$ and $\mathbf{x}_j$ to the same class using a kNN classifier, which equals:

$$p(y_i = y_j|\mathbf{x}_i, \mathbf{x}_j) = \sum_c p(y_i = c|\mathbf{x}_i)p(y_j = c|\mathbf{x}_j) \tag{2.42}$$

$$= k^{-2} \sum_c n_c^i n_c^j. \tag{2.43}$$

Alternatively, we can understand this method directly as a nearest neighbour classifier in the implicit binary labelled set of N^2 pairs. In this set, we need a measure to define neighbours of a pair. One choice to do so for a pair $(\mathbf{x}_i, \mathbf{x}_j)$ is to take all the k^2 pairs we can make using one of the k neighbours of $\mathbf{x}_i$ and one of the k neighbours of $\mathbf{x}_j$. The probability for the positive class given by this classifier for a pair is then determined by the number of positive $n_+^{ij} = \sum_c n_c^i n_c^j$ neighbour pairs, and is precisely given by Equation 2.43 except for a fixed constant factor k^{-2}.

Either way, the score of our Marginalised kNN (MkNN) binary classifier for a pair of images $(\mathbf{x}_i, \mathbf{x}_j)$ is based on how many positive neighbour pairs we can form from neighbours of $\mathbf{x}_i$ and $\mathbf{x}_j$. In Figure 2.8 we illustrate the procedure with a simple example. We expect this method to benefit most from an LMNN base metric to define the neighbours, as it is designed to improve kNN classification.

Using this approach, we profit from the amount of available data and flexibly model non-linearities, at the expense of a higher computational cost at test time. Note that this method is not "local" in the sense of usual kNN classifiers or other "local learning" methods (Bottou and Vapnik [1992], Frome et al. [2007]), as MkNN measures the correspondence between two distinct local neighbourhoods. It implicitly uses all pairs we can generate from the labelled faces.

2.4 Data set and features

In this section, we present the data set used for our experiments and our feature extraction procedure for describing faces in a vector space, as required by metric learning techniques. We also discuss several alternative face descriptors from the literature.

2.4.1 *Labeled Faces in the Wild*

The *Labeled Faces in the Wild* data set (Huang et al. [2007b]) originates from a multimodal data set named *Yahoo! News*, which was collected in 2002–2003 and introduced by Berg et al. [2004a]. The *Yahoo! News* data set consists of images and accompanying captions. On this large collection, a face detector was applied, as well as a named entity detector. Documents with no detected faces or names were removed, leading to a database with roughly 15000 image and caption pairs, with 15280 detected named entities and 22750 detected faces.

Using these documents as sets of faces and names, a constrained clustering technique was proposed to automatically assign names to the depicted faces. However, it yield a high percentage of errors, and some pictures appeared to be duplicates. In Chapter 3, we consider alternative approaches for this clustering task. To obtain a labelled data set of face images, Huang et al. [2007b] have taken the approach of manually annotating a subset of the face images using the captions as an aid for naming the person, using a unique identifier for each individual. In the end, the *Labeled Faces in the Wild* data set contains 12233 face images with identity labels. In total 5749 people appear in the images, 1680 of them appear in two or more images.

The faces show a big variety in pose, expression, lighting, etc., see Figure 2.2 for some examples. An aligned version of all faces is available, referred to as "funnelled", which we use throughout our experiments. This data set can be viewed as a partial ground-truth for the *Yahoo! News* data set. *Labeled Faces in the Wild* has become the *de facto* standard data set for face verification, with an active monitoring of the literature for improvements. As of today, more than 20 participants have submitted results to this challenge, even though the data has been available for less than 2 years.

The data set comes with a division in 10 parts called *folds* that can be used for cross validation experiments. The folds contain between 527 and 609 different people each, and between 1016 and 1783 faces. From all possible pairs, a small selection of 300 positive and 300 negative image pairs are provided for each fold. Using only these pairs for training is referred to as the "image-restricted" paradigm; in this case the identity of the people in the pairs cannot be used. The "unrestricted" paradigm is

used to refer to training methods that can use all available data, including the identity of the people in the images.

Performance is measured using a ROC curve on the 6000 selected pairs with their classification scores obtained from classifiers when their fold is excluded from training. A ROC curve plots the true positive rate versus the false positive rate. Since the positive and negative classes are balanced in the test set, we will also report the classification performance when there are as many false positive as false negative. This corresponds to the operating point of the ROC curve with equal misclassification, or equal error rate. We therefore refer to this measure as the ROC-EMC accuracy:

$$v = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}, \tag{2.44}$$

where TP is the number of true positives, TN the true negatives, P the true number of positives and N the true number of negatives. The recommended measures for comparing methods on the *Labeled Faces in the Wild* data set are the following: mean accuracy μ and standard deviation σ over the folds, as given by:

$$\mu = \frac{\sum_{i=1}^{10} v_i}{10} \quad \text{and} \quad \sigma = \sqrt{\frac{\sum_{i=1}^{10} (v_i - \mu)^2}{9}}, \tag{2.45}$$

where v_i is the accuracy for fold i alone.

2.4.2 Face descriptors

In this section, we first describe existing descriptors for faces then present the one we used in our experiments.

Historically, a vectorial representation of faces is obtained from a holistic description of the face image. This includes recent successful approaches like Local Binary Patterns (LBP, Ahonen et al. [2004]) and later extensions – *e.g.* TPLBP and FPLBP (Wolf et al. [2008]).

As the name suggests, the idea behind LBP is to extract a binary code from the pattern locally surrounding a pixel. From the comparison between a pixel value and its eight neighbour values, a 8-bit code is obtained, as illustrated in Figure 2.9. The robustness of LBP with illumination changes comes from the invariance of the comparison to monotonic transformations of the greyscale pixel values. Notably, Gamma correction ($x \mapsto x^\gamma$), brightness and contrast transformations are monotonic.

We also consider the extension of LBP from Wolf et al. [2008], namely Three-Patch and Four-Patch LBP (TPLBP and FPLBP, respectively). As illustrated in Figure 2.10,

57	65	80
73	68	66
95	53	33

$\rightarrow$

0	0	1
1	↻	0
1	0	0

$\rightarrow \quad c = 10001100$

***Figure 2.9**: A local binary pattern (LBP) is a binary code formed by comparing a pixel value with its neighbours (left). The 8 comparisons are encoded with 0 and 1 (centre), and the pattern is serialized into a 8-bit code (right).*

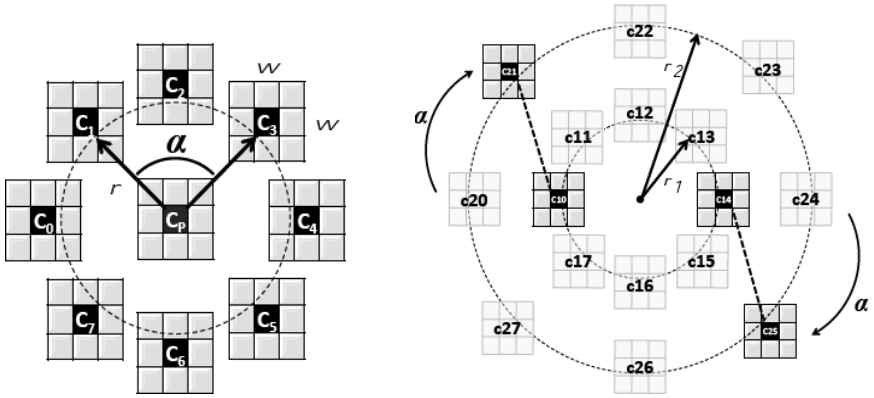


***Figure 2.10**: TPLBP (left), respectively FPLBP (right), are extensions of LBP. The drawings, from Wolf et al. [2008], show how codes are constructed from comparing pixel values on three, respectively four, patches, with the associated parameters w (patch size), r (radius) and α (angle).*

they consist in sampling several patches in the neighbourhood of pixels and perform comparison between several pixel values in order to obtain binary codes.[8] Notably, these descriptors have many parameters to set. We will use the recommended parameter values provided by the authors of the original work.

Other global descriptors exist. For instance GIST (Oliva and Torralba [2001]) and Histogram of Oriented Gradient (HOG, Dalal and Triggs [2005]). Although they are not specifically designed for human face recognition, it is interesting to use them in combination of machine learning techniques like metric learning for the verification task. The work of Funes Mora [2010] shows that HOG descriptors can perform comparably for our SIFT descriptor described below, but the face images have to be correctly aligned first.

Another type of face description relies on facial feature detection. Facial feature detection, also known as fiducial point localization, has the goal of localizing in face images specific points such as corners of eyes, mouth, nose, eyebrows. This is an im-

[8] Code available at: `http://www.openu.ac.il/home/hassner/projects/Patchlbp/`

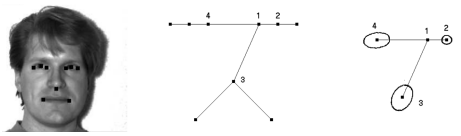


***Figure 2.11**: On the left, the facial features are shown. The sketch is the middle illustrates the tree-like constellation model. On the right, the position uncertainty for parts 2, 3 and 4 are shown once part 1 is fixed. Courtesy of Felzenszwalb and Huttenlocher [2005].*

portant tool for face recognition as it can intervene in several processing steps. The first is data alignment: it is straightforward to infer an affine transformation between two faces if at least three fiducial points are matched (see also Urschler et al. [2009]). Second, due to the flexibility of individual localizations, the facial features can help build face descriptors that are invariant to non-linear transformations. Finally, if confidence scores are obtained for the localization of the fiducial points, it is possible to estimate the probability that the points are incorrectly localized, using for instance outlier detection techniques. An incorrectly localized feature could also be caused by an occlusion occurring on the corresponding part of the face image. Systems can therefore build upon facial feature detection for handling occlusions in a robust manner, see *e.g.* Funes Mora [2010].

State-of-the-art facial feature detection include Felzenszwalb and Huttenlocher [2005]. It is based on a constellation model which combines discriminative local appearance models and a generative model for spatial regularization. The system is made computationally efficient by using a tree-structured Gaussian mixture for the joint position of the features, as illustrated in Figure 2.11, and this structure is learned together with the appearance models at train time. Three mixture components are used and they correspond approximately to frontal faces and faces oriented towards both sides. For the appearance models, score functions are learned from the training set using an AdaBoost framework, with weak classifiers operating on Haar features.

Everingham et al. [2006] improved the efficiency of the appearance model evaluation, and trained the system on detecting 9 facial features: (1) Left corner of the left eye, (2) Right corner of the left eye, (3) Left corner of the right eye, (4) Right corner of the right eye, (5) Left corner of the nose, (6) Tip of the nose, (7) Right corner of the nose, (8) Left corner of the mouth, and (9) Right corner of the mouth. Having detected the facial features, their description is usually performed by extracting features from the pixel neighbourhood using local appearance descriptors, as illustrated in Figure 2.12.

Everingham et al. [2006] compared two descriptors. First they used the pixel values of surrounding circular patches, including an alignment procedure, normalisation and noise reduction. The resulting descriptor is illustrated in Figure 2.13. Greyscale values are serialized and concatenated into a 1937D descriptor. Second, they tried using

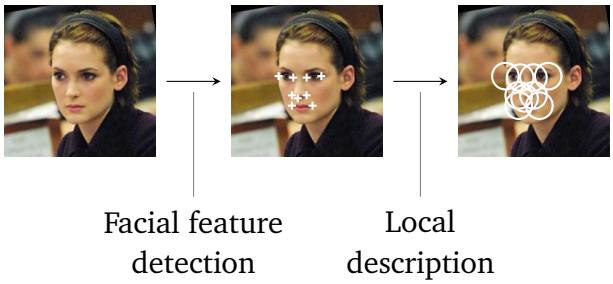


***Figure 2.12**: Illustration of the facial feature-based processing pipeline of Everingham et al. [2006]. After detecting faces in the images, face images are aligned and given to the facial feature detector. Then, local appearance descriptors of these fiducial points are extracted and concatenated to obtain a vectorial representation of the faces.*

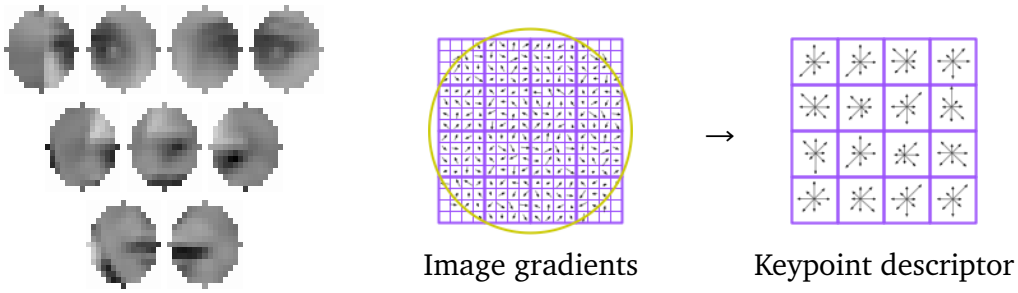


***Figure 2.13**: On the left, the circular patches describe the 9 facial features in the 1937D descriptor of Everingham et al. [2006]. In the centre and on the right, illustration of SIFT features (128D each, Lowe [1999]) for describing the patches.*

SIFT descriptors (Lowe [1999]). SIFT features have proved very successful in many computer vision tasks such as image stitching, object recognition, robotic navigation, and action recognition in videos. The SIFT descriptor is composed of a grid of $4 \times 4 = 16$ histograms of 8 bins, and is therefore 128D. Each bin represents the magnitude for a particular orientation of the gradient in the cell begin considered. This magnitude is weighted by a Gaussian function centred on the keypoint. Using the Euclidean distance between descriptors, the authors did not find the SIFT features to bring any improvement.

Relying on the success of using facial features for recognition, we will use the detector of Everingham et al. [2006] which is available on the web.[9] In a setting where a keypoint provides a scale, the SIFT description is performed at that particular scale.

[9] At: `http://www.robots.ox.ac.uk/~vgg/research/nface/`.

***Figure 2.14**: Illustration of our SIFT-based face descriptor. SIFT features are extracted at 9 locations and 3 scales shown on the top. Each row represents a scale at which the patches are extracted: the top row is scale 1, the middle row is scale 2 and the bottom row is scale 3. The first column shows the locations of the facial features, and the remaining nine columns show the corresponding patches. Our descriptor is the 3456D concatenation of the* 3×9 *SIFT features describing the patches.*

Here, although the face detector gives an approximate scale for the face, it is not precise, and the facial feature detector does not provide any scale either.

To overcome this potential issue, we propose (*c.f.* Guillaumin et al. [2009b]) to use multiscale SIFT descriptors to describe the patches. Setting scale $\sigma = 1$ to represent a 16×16 patch in the 250×250 face images, we extract SIFT features at multiple scale for $\sigma \in \{1, 2, 3\}$. Our 3456D SIFT-based face descriptor is the concatenation of the $9 \times 3 = 27$ SIFT features, as illustrated in Figure 2.14.

2.5 Experiments

In this section, we present our experimental study of the different descriptors and metrics. First, in Section 2.5.1, we compare different descriptors using basic metrics. Then, in Section 2.5.2, we study the influence of the parameters of our descriptor and that of the metric algorithms. In Section 2.5.3 we evaluate the classification performance of the MkNN approach, and in Section 2.5.4 we compare to the state-of-the-art on the *Labeled Faces in the Wild* data set. Finally, in Section 2.5.5 and Section 2.5.6, we consider two applications of learnt metrics, namely clustering and recognition from one exemplar.

Name	ID	Definition
Euclidean	L2	$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (\mathbf{x}_i - \mathbf{y}_i)^2}$
Hellinger	Hellinger	$d(\mathbf{x},\mathbf{y}) = \sum_i (\sqrt{\mathbf{x}_i} - \sqrt{\mathbf{y}_i})^2$
Chi-square	χ^2	$d(\mathbf{x},\mathbf{y}) = \sum_i (\mathbf{x}_i - \mathbf{y}_i)^2 / (\mathbf{x}_i + \mathbf{y}_i)$

***Table 2.1**: Names and definitions of three base metrics for comparing distributions or histograms (i.e. with all components being positive): the Euclidean distance, the Hellinger distance which is the Euclidean distance on square-rooted data, and the Chi-square pseudo-distance.*

2.5.1 Comparison of descriptors and basic metrics

Here, we use basic metrics to compare different descriptors. Namely, we compare the Euclidean distance, the Hellinger distance and the Chi-square pseudo-distance. Their definitions are given in Table 2.1.

As can be noted, the Hellinger distance corresponds to the Euclidean distance on vectors whose components have been square-rooted, and is related to the Bhattacharyya distance. For comparing histograms, considering this square-root transformation has proved to bring small, yet tangible, improvements for classification tasks (*e.g.*, see Wolf et al. [2008], Guillaumin et al. [2009b], Perronnin et al. [2010], Funes Mora [2010]). Similarly, the Chi-square distance is a robust distance measure for multinomial probability distributions. It can be understood as a weighted Euclidean distance where differences in infrequent components are stressed over differences in large-valued components.

In Table 2.2, we show the classification performance on *Labeled Faces in the Wild* for state-of-the-art face descriptors using those metrics. Results are obtained by simply thresholding standard metrics, and since there is not training involved, they are valid for both restricted and unrestricted settings. Perhaps surprisingly, the results in Table 2.2 show that all descriptors and distances lead to comparable ROC-EMC of 66% to 69%. The SIFT-based descriptor performs slightly better than the others on all three metrics, just ahead of the original LBP descriptor, and we observed a similar hierarchy on preliminary results using learnt metrics. In the following, we use the SIFT descriptor to compare learnt metrics.

2.5.2 Metric learning algorithms

In this section we analyse the performance of our face descriptor with respect to its main parameters. Evaluation on the *Labeled Faces in the Wild* data set is done on the

Descriptor	L2	Hellinger	χ^2
Patch	66.58 ± 0.5	67.12 ± 0.7	67.07 ± 0.7
LBP	67.65 ± 0.7	68.13 ± 0.7	68.33 ± 0.6
TPLBP	66.90 ± 0.4	66.82 ± 0.4	66.58 ± 0.2
FPLBP	66.52 ± 0.5	67.37 ± 0.4	67.10 ± 0.5
SIFT	$\mathbf{67.78 \pm 0.6}$	$\mathbf{68.50 \pm 0.5}$	$\mathbf{68.77 \pm 0.4}$

Table 2.2: *ROC-EMC classification results for L2, Hellinger and χ^2 distances for different descriptors: patch pixel values (Patch) from Everingham et al. [2006], LBP and variants from Wolf et al. [2008], and finally our SIFT-based face descriptor.*

"unrestricted" setting, where the faces and their identities are used to form all the possible negative and positive pairs. The Equal Error Rate of the ROC curve over the ten folds is then used as performance measure, see Equation 2.44.

The following parameters are studied:

1. *The scales of the descriptor*. We compare the performance of each individual scale (see Figure 2.14) independently, and their combination. The results are shown in Figure 2.16 and are discussed below.

2. *The dimensionality of the descriptor*. Except for the Euclidean distance, using more than 500 dimensions is impractical, since metric learning involves algorithms that scale as $O(D^2)$ where D is the data dimensionality. Moreover, we can expect to over-fit when trying to optimise over a large number of parameters. Therefore, we compare in Figure 2.15 the performance of metric learning algorithms on data pre-processed with PCA to keep respectively 35, 55, 100, 200 and 500 dimensions. LDML is also able to learn metrics with this reduced dimensionality directly.

3. *Metrics for the descriptor*. We compare the following measures: Euclidean distance (L2), Euclidean distance after PCA (PCA-L2), LDML metric after PCA (PCA-LDML), LMNN metric after PCA (PCA-LMNN), ITML metric after PCA (PCA-ITML), and finally Euclidean distance after low-rank LDML projection (LDML-L2).

In Figure 2.15, we present the performance on *Labeled Faces in the Wild* of the different metrics for each individual scales of the descriptor, as a function of the data dimensionality. As a first observation, we note that all the learnt metrics perform much better than the unsupervised metrics like L2 and PCA-L2. The difference of performance between learnt metrics is smaller than the gap between learnt metrics and unsupervised ones.

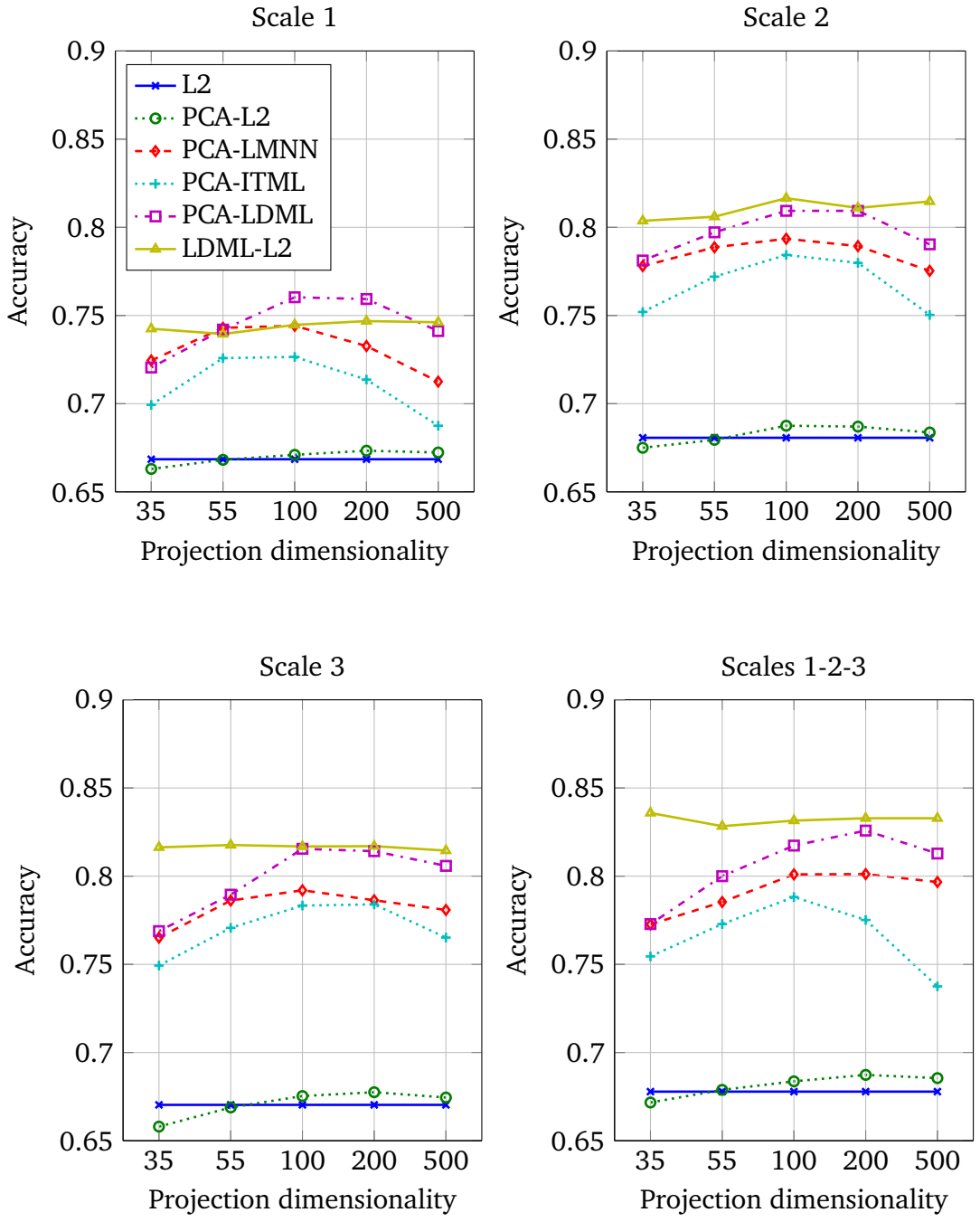


***Figure 2.15**: Comparison of methods for the three scales of the face descriptor and the concatenated descriptor of all three scales. We show the accuracy of the projection methods with respect to the dimensionality, except for L2 where it is irrelevant. Scales 2 and 3 appear more discriminative than scale 1 using learnt metrics, and the concatenation brings an improvement. Except for scale 1, LDML-L2 performs best on a wide range of dimensionalities.*

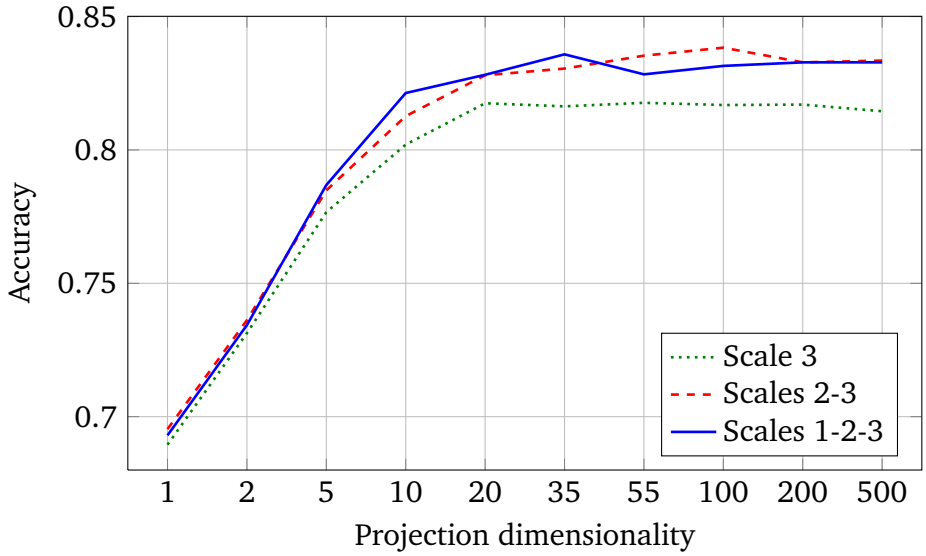


***Figure 2.16**: Accuracy of LDML projections over a wide range of space dimensionalities, for scale 3, the combination of scale 2 and 3, and the three scales.*

When comparing performance obtained with the different scales, we see that scales 2 and 3 perform similarly, and better than scale 1. The combination of the scales brings an improvement over the individual scales.

From Figure 2.15, we also observe that metric learning methods benefit from preprocessing with larger PCA dimensionalities up to 200 dimensions. For low dimensionalities, the methods are limited by the weak discriminative power of PCA. We can observe a hierarchy of methods: PCA-LDML performs better than PCA-LMNN, which itself performs better then PCA-ITML. But the difference is rarely more than 2% between PCA-ITML and PCA-LDML below 200 dimensions. Performances seem to decrease when the data dimensionality is above 200, which might be due to overfitting. For ITML, the drop can be explained by unoptimised code which required early stopping in the optimisation. Keeping 100 to 200 PCA dimensions appears as a good trade-off between dimensionality reduction and discriminative power. When using LDML for supervised dimensionality reduction, the performance is maintained at a very good level when the dimension is reduced, and typically LDML-L2 is the best performing method in low dimensions.

The performance of LDML-L2 for dimensionalities ranging from 1 to 500 can be seen in Figure 2.16, with an illustration already shown in Figure 2.7. We show the influence of target space dimensionality on performance for the best scale (the third), the two best scales (second and third) and all three scales together. We can clearly observe that combining scales benefits the performance, at the expense of a higher dimensional input space. Notably, adding scale 1 does not seem to have any significant effect on performance.

ITML	LDA-based	LMNN	LDML	MkNN
80.5 ± 0.5	79.3 ± 0.3	80.5 ± 0.5	$\mathbf{83.8 \pm 0.6}$	$\mathbf{83.1 \pm 0.5}$

***Table 2.3**: Comparison of ROC-EMC classification results for methods in the unrestricted setting (SIFT). LDML is used with 100D, while ITML, LMNN and MkNN use 200D PCA pre-processing. Compared to the previous section, ITML is here trained on 90000 pairs but more optimisation iterations are performed.*

We can observe a plateau of performance starting at 20 dimensions, and even 10 for the full descriptor. This shows that the relevant linear variability in the data for this task is particularly low-dimensional, and is sufficient to distinguish people with around 82% to 83% accuracy. Additional performance can be expected with non-linear methods or combination with additional visual cues.

In the rest of the experiments in this chapter, we will use the descriptor composed of all three scales. In the next chapter, we employ methods that would over-fit with this large descriptor, so we will be using only the last two scales since this can be done without any loss of performance.

2.5.3 Nearest-neighbour classification

Using the labels of the unrestricted setting, we can employ LMNN and our MkNN approach. For LMNN we used a PCA projection of the data to 200 dimensions; using less dimensions gave slightly worse results, and using more dimensions gave slightly better results at the cost of much higher training times. We used 5 target neighbours to learn the LMNN metric; using between 3 and 20 target neighbours gave similar performance, other values gave slightly worse results. This resulted in a best performance of 80.5%.

We, then, applied the MkNN classifier using L2, LMNN, and LDML as base metrics. In the case of L2 and LDML as base metric, the MkNN classifier did not give as good results as the base metric. However, when using LMNN, designed for kNN classification, as a base metric, the MkNN classifier performs better when between 100 and 200 neighbours are used: 83.1% instead of 80.5%, see Figure 2.17. We also considered a variant where a weighted sum of the base metric and the class probability is learnt using a logistic discriminant classifier. This combination brings an insignificant improvement from LDML, a small improvement over the base metric from 67.8% to 69.2% for L2, from 80.5% to 83.5% for LMNN. Furthermore, for LMNN, this improvement is consistent over all neighbourhood sizes, as shown in Figure 2.17. We summarise the comparison with the results of other methods in Table 2.3

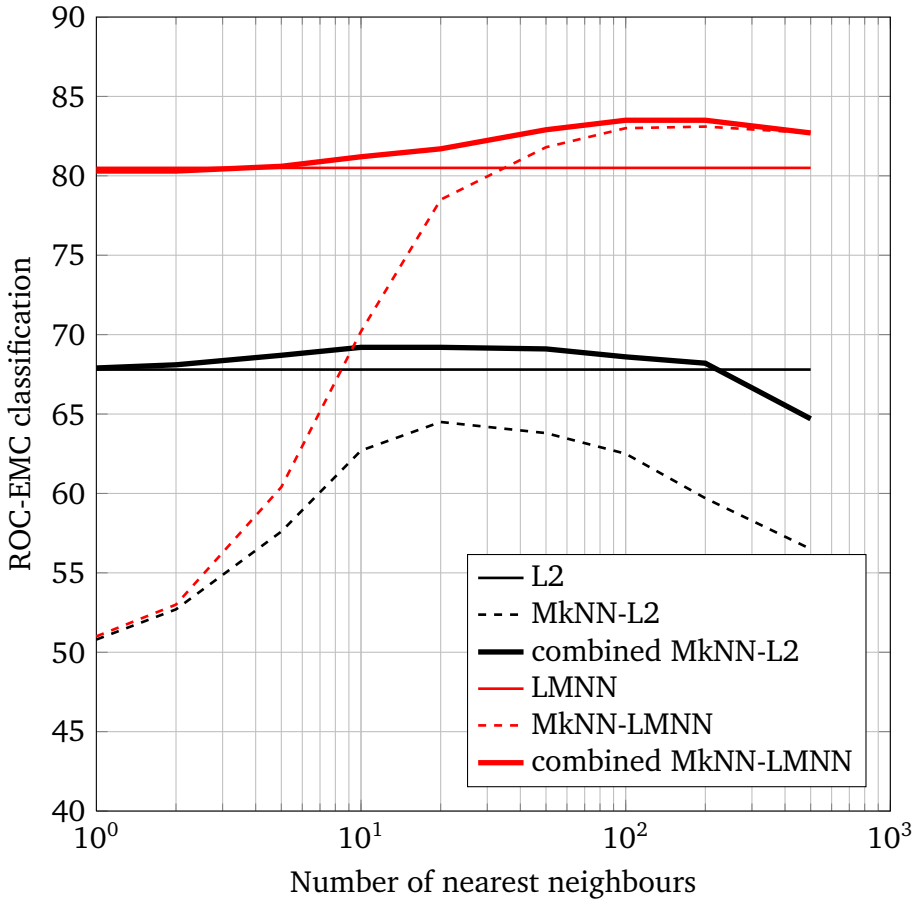


***Figure 2.17**: ROC-EMC performance using the MkNN classifier, with L2 and LMNN as base metrics.*

Figure 2.18 shows some of the examples that were incorrectly classified using the LMNN metric, but were correctly classified using the MkNN classifier. The benefit of the MkNN classifier can be seen most for pairs with large pose and or expression changes.

2.5.4 Comparison to the state-of-the-art

In this section, we compare with previously published results on *Labeled Faces in the Wild* (Huang et al. [2008], Nowak and Jurie [2007], Wolf et al. [2008]). We used the strict protocol to calculate the ROC curve and accuracy for our method. Note that each published result combines its own feature extraction with its own machine learning technique, making any conclusion harder to draw than in the previous sections. Since our original publication (Guillaumin et al. [2009b]), several other methods have been proposed and show improvements over these figures (*e.g.*, see Kumar et al. [2009],

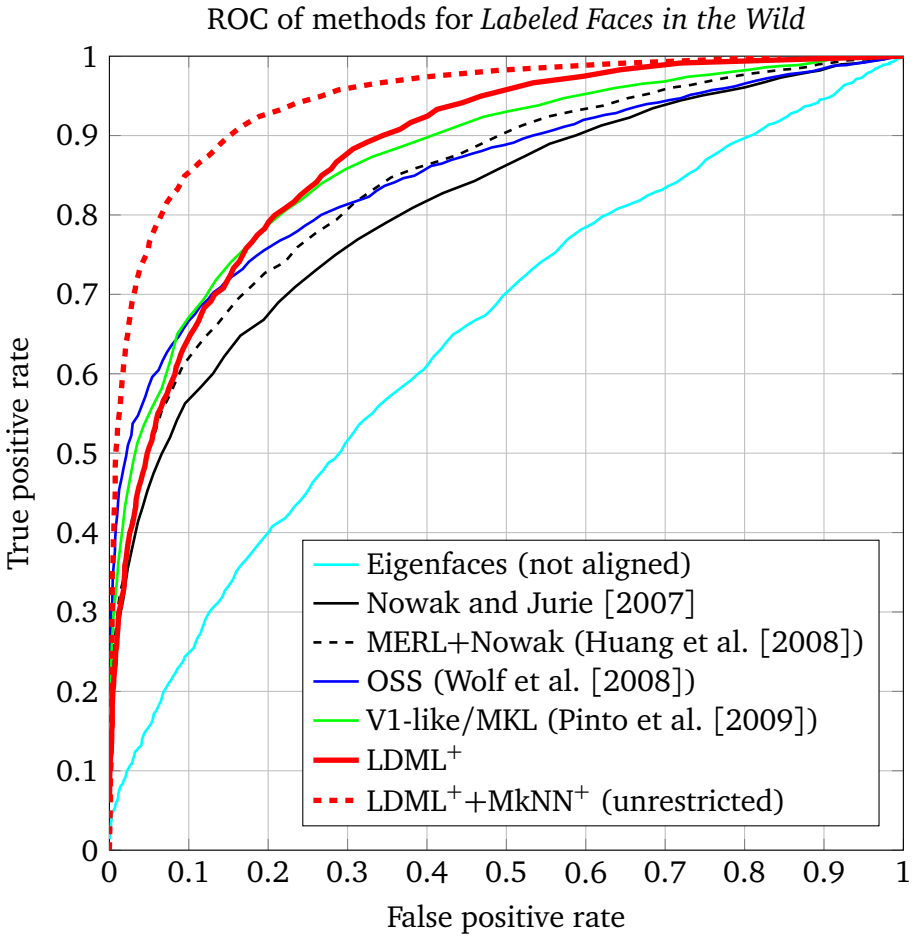
***Figure 2.18**: Examples of positive pairs correctly classified using the MkNN classifier with LMNN as a base metric, but wrongly classified using the LMNN metric alone.*

Wolf et al. [2009]), but the improvements are obtained either using external data or a better face alignment, which could also improve our method.

Following recent work (*e.g.*, see Varma and Ray [2007], Wolf et al. [2008]), we have linearly combined different scores to improve classification performance. In the restricted setting, we combine 4 descriptors (LBP, TPLBP, FPLBP and SIFT, *c.f.* Table 2.2) with the LDML metrics on the original data and its square root (8 scores). Remember that the restricted setting limits to 600 the number of pairs to use in each fold. In the unrestricted setting, we combine the same inputs with the LDML, LMNN, and MkNN metrics (24 scores), and limited the number of training pairs to 10000 per fold for computational efficiency. The linear combinations are learnt using a logistic discriminant model for each fold independently. In Figure 2.19 and until the end of the chapter, we refer to these combined methods as LDML$^+$ and LDML$^+$+MkNN$^+$, with a "$^+$" sign to stress that there are combination of scores.

The ROC curves in Figure 2.19 show that LDML$^+$ is close to the method of Pinto et al. [2009] in terms of performance, and largely better than earlier methods like Nowak and Jurie [2007] and Huang et al. [2008]. The best result reported by (Wolf et al. [2008]) attains 78.47% accuracy in the restricted setting also by combining several descriptors. LDML$^+$ on the restricted setting obtains an accuracy of 79.27%, while Pinto et al. [2009] reports 79.53%.

When considering the unsupervised setting, our combined model significantly outperforms all published methods on the funnelled data. LDML$^+$+MkNN$^+$ obtains a performance of 87.50%, showing the benefit of our metric learning approaches when using more training data. Notably, in Guillaumin et al. [2009a], we showed that the competing method of Wolf et al. [2008] did not benefit from this additional training data. Taigman et al. [2009] later obtained even better results in the unrestricted setting, but they were reported using a commercial face alignment software, and therefore are not directly comparable to our results.



Method	Accuracy
Eigenfaces, not aligned	60.02 ± 0.8
Nowak and Jurie [2007], restricted	73.93 ± 0.5
MERL+Nowak, restricted (Huang et al. [2008])	76.18 ± 0.6
Hybrid descriptor-based (OSS), restricted (Wolf et al. [2008])	78.47 ± 0.5
V1-like/MKL, restricted (Pinto et al. [2009])	**79.35 ± 0.6**
LDML+, restricted	**79.27 ± 0.6**
LDML+MkNN+, unrestricted	**87.50 ± 0.4**

***Figure 2.19**: Comparison of our results with best results on the funnelled LFW data: ROC curves, and average accuracy and standard error.*

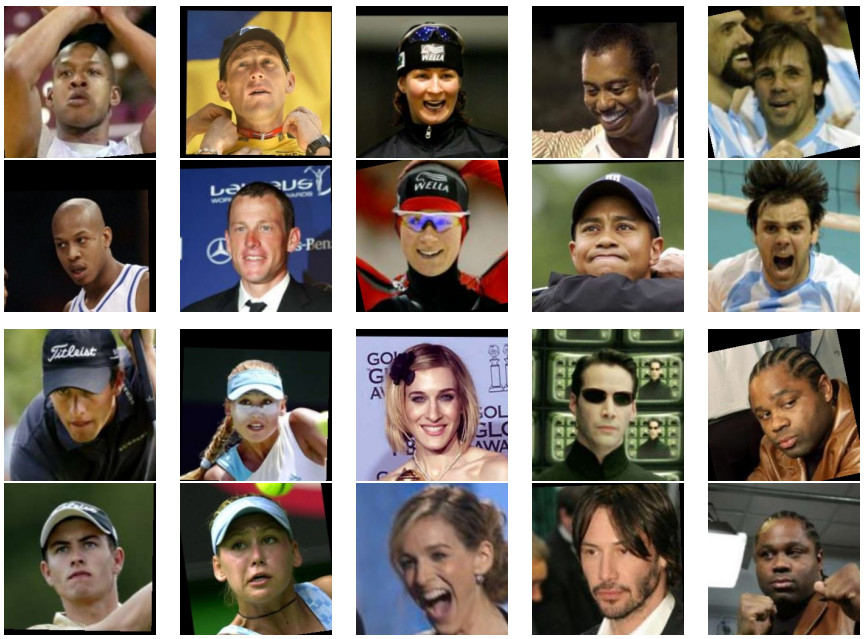


***Figure 2.20**: Several examples face pairs of the same person from the Labeled Faces in the Wild data set. We show pairs that were correctly (top) and incorrectly (bottom) classified with our LDML$^+$+MkNN$^+$ method combining features and metrics.*

Finally, we observe that a combination of descriptors and metrics improves over using only one metric and one descriptor, highlighting the complementarity of our two approaches. We refer to Figure 2.20 for classification examples using our combined method.

2.5.5 Face clustering

Here we show the merit of learnt metrics for a first application: unsupervised clustering of face images. We learn our metrics exactly as in the previous Section 2.5.4, on 90000 pairs of 9 folds (unrestricted setting), and apply them to faces in the held-out fold. The test fold contains 1369 faces from 601 people. In the following experiments, we focus on the 17 most frequent people (411 faces). We compare L2 and LDML on SIFT, and LDML$^+$+MkNN$^+$, *i.e.* combining several scores using logistic discriminant.

We cluster the faces using complete-linkage hierarchical clustering. This method yields a hierarchy of clusters by varying the maximum distance with which clusters can be merged. To compare clustering results we define a cost that reflects the labelling effort needed for a user to label the faces, *e.g.* for a personal photo album. We assume the user has two buttons: one to assign a single label to all faces in a cluster, and one to assign a label to a single face. The most efficient way to label all faces in

Metric	L2	LDML	LDML$^+$+MkNN$^+$
Total correct	57.8%	70.3%	75.7%
Faces of the 17 targets			
correctly recognised (b)	14.0%	38.8%	53.3%
wrongly recognised	27.2%	20.6%	10.2%
wrongly rejected (c)	58.9%	40.6%	36.5%
Faces of other people			
correctly rejected	75.8%	83.3%	85.0%
wrongly accepted (d)	24.2%	16.7%	15.0%

***Table 2.4**: Comparison of one-exemplar classification performances. The 1352 test faces are broken down over the five possible situations. The labels (b)–(d) refer to the example images shown in Figure 2.23.*

a cluster is to first label the cluster with the name of the most frequent person, and then to correct the errors. For a cluster c of n^c faces, the cost is $1+n^c-\max_i n_i^c$, where n_i^c denotes the number of faces of person i in the cluster. The cost to label all faces is then the sum of the costs to label the faces in each cluster. The optimal clustering has cost 17, a trivial over-clustering with a cluster for each face yields a cost of 411, while using a single cluster of all faces yields a cost of 341 as we have 71 images of the most frequent person. Formal definitions and derivation of bounds on the labelling cost are given in Appendix A.

In Figure 2.22 we show the costs as a function of the number of clusters using the L2 and LDML metrics on the SIFT data, the LDML$^+$+MkNN$^+$ combination, the average for random clustering, and the minimum and maximum costs that can be obtained. Clearly, LDML yields much better clustering results than L2 for a wide range of number of clusters. For LDML the minimum cost of 109 is obtained with only 25 clusters, most of which are fairly pure. If we label the faces in each cluster by the identity of the most frequent person in that cluster then 75% of the faces are correctly classified. For the L2 metric the minimum cost is 233 for 135 clusters (92% correct but over-clustered). Combining the different descriptors with LDML leads to a decreased cost of 88 with 28 clusters (85% correct), and with LDML$^+$+MkNN$^+$ the cost drops to 71 with 29 clusters (90% correct). In Figure 2.21 we show three example clusters from this clustering. Note that the clustering is successful despite big changes in expression, pose, and lighting.

2.5.6 Recognition from one exemplar

Here, we perform multi-class face recognition using a single random training exemplar for each of the 17 people. We test classification accuracy on the remaining $1369-17=1352$ faces. A test face is assigned to the exemplar with the best score,

Figure 2.21: *Three example clusters obtained using LDML$^+$+MkNN$^+$ scores. The top two clusters are pure, and only few faces of these persons are assigned to incorrect clusters. The last cluster is typical, it contains a few faces of other people (the last 2).*

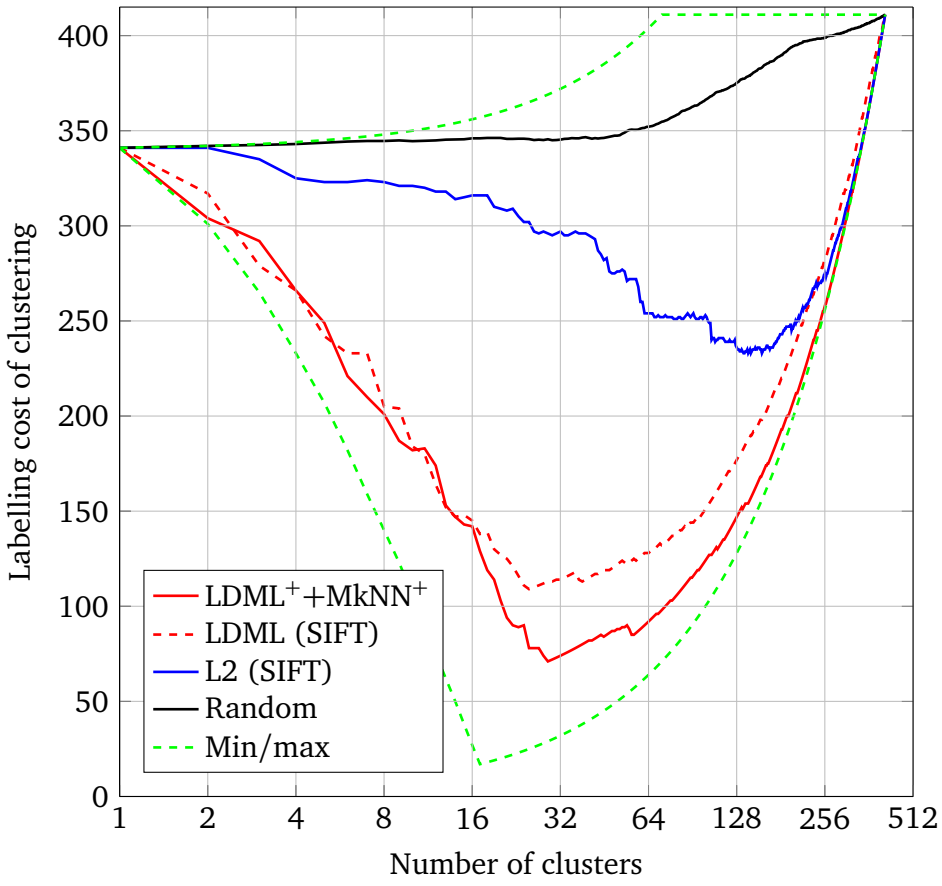


***Figure 2.22**: Labelling cost of clusterings using different metrics.*

or rejected if all scores are below a certain threshold. From the 1352 test faces only $411 - 17 = 394$ should be accepted as one of the 17 classes, and the remaining 958 should be rejected. We measure performance using precision at equal error rate, where the number of wrongly rejected faces equals the number of wrongly accepted faces. In Table 2.4 we present quantitative results showing that, as with the clustering, LDML leads to significantly better performance than L2 on the SIFT features: 39% of the accepted faces are correctly recognised, compared to only 14%. The LDML$^+$+MkNN$^+$ combination boosts precision to 53%. In Figure 2.23 we show classification examples for LDML$^+$+MkNN$^+$.

2.6 Conclusion

We have introduced two new methods for visual verification: Logistic Discriminant Metric Learning (LDML), and Marginalised kNN classification (MkNN). We note that LDML can be trained from labelled pairs as provided in the restricted paradigm of *La-*

***Figure 2.23**: Illustration of face recognition of 7 (out of 17) people using one training exemplar, with one person in each column. For each person we show: (a) the exemplar image, (b) a correctly recognised face of that person, (c) a non-recognised face of that person, and (d) another failure: an erroneously accepted face of another person.*

beled Faces in the Wild, whereas MkNN requires labelled training data and implicitly uses all pairs. With its log loss, LDML is a robust technique to learn a Mahalanobis metric in a supervised fashion. Additionally, LDML can perform dimensionality reduction and is kernelizable. The MkNN classifier is conceptually simple, but in practice it is computationally expensive as we need to find nearest neighbours in a large set of labelled data. This computational cost can be alleviated by using efficient and/or approximate nearest neighbour search techniques.

LDML in combination with our descriptors yields a classification accuracy of 79.3% on the restricted setting of *Labeled Faces in the Wild* data set, where the best reported results so far were 78.5% and 79.5% using the funnelled data (Wolf et al. [2008] and Pinto et al. [2009]) and 86.8% using improved alignment and background information (*c.f.* Wolf et al. [2009]). LDML and MkNN yield comparable accuracies on the unrestricted setting, above 83%. Remarkably, the gain when using the unrestricted setting is not observed with other state-of-the-art methods such as Wolf et al. [2008]. We were the first to present results on the *Labeled Faces in the Wild* data that follow and make good use of the unrestricted paradigm. Combining our methods, the accuracy is further improved to 87.5%. Later work by Taigman et al. [2009] obtained 89.5%, which is the current best for the unrestricted setting.

We also showed that metric learning leads to great improvements as compared to a simple L2 metric for applications of face similarities like clustering and recognition from a single exemplar.

For face recognition, looking at the examples of failure cases of our method in Figure 2.20, pose changes remain one of the major challenges to be tackled in future work. Explicit modelling of invariance due to pose changes using techniques like those in Cao et al. [2010] is an option worth exploring.

Concerning metric learning, we plan to explore learning directly the optimal metric for MkNN classification. Metric learning is more generally a promising technique to use for other computer vision tasks, for instance for retrieval. We intend to extend our study to other data sets and tasks, especially using weak settings.

3

Caption-based supervision for face naming and recognition

Contents

3.1 Introduction

In this chapter, we consider a first type of multimodal data: news images with captions. This data is typically published by news media agencies such as Agence France Presse, Associated Press, Reuters, or the Belga News Agency. The published documents consist of a short text describing a piece of news, and an image illustrating the event, as shown in Figure 3.1. Although they are not specifically created with this purpose, the textual parts of those documents describe the visual content of the images to some extent.

Using such data sets, we can consider many computer vision applications, as long as the image caption is sufficiently informative about the visual task at hand. Obviously, the origin of the data puts a strong bias towards political, sport and social events. Many news stories concern people and their actions: President Obama addresses the press, Roger Federer wins a tennis match, etc...

An Iranian reads the last issue of the Farsi-language Nowruz in Tehran, Iran Wednesday, July 24, 2002. An appeals court on Wednesday confirmed the sentence banning Iran's leading reformist daily Nowruz from publishing for six months and its publisher, Mohsen Mirdamadi, who is President Mohammad Khatami's ally, from reporting for four years. Mirdamadi is head of the National Security and Foreign Policy Committee of the Iranian parliament. (AP Photo/Hasan Sarbakhshian)

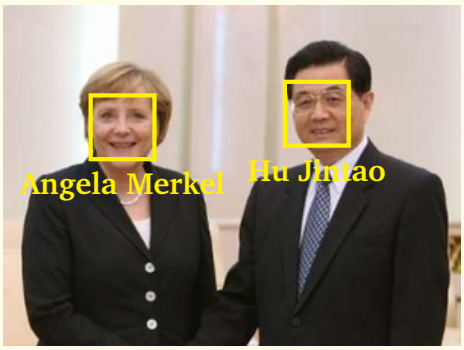
Chanda Rubin of the United States returns a shot during her match against Elena Dementieva of Russia at the Hong Kong Ladies Challenge January 1, 2003. Rubin beat Dementieva 6-4 6-1. (REUTERS/Bobby Yip)

***Figure 3.1**: Illustration of two multimodal documents from news agencies: they consist of a text illustrated by an image. The caption therefore partially describes the visual content of the image, especially in terms of human identities and actions.*

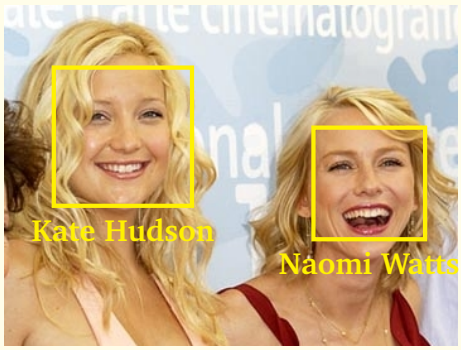
It is quite natural to try to leverage the quantity of information that these data sets represent for tasks that relate to human properties such as: their identity, their pose and appearance, or their actions. The applications are numerous, and given the convergence of the quality of amateur photography as found on Flickr and Facebook towards professional photography, it is expected that the systems for face and action recognition that we today train on news data will soon be available directly as tools for users of media sharing websites.

In Chapter 2, we showed how we can use a similar type of images with manual labels to train face recognition systems in uncontrolled settings. Here, we explore how we can exploit the weak supervision that the captions provide instead of using manual annotations. Specifically, we study the following two tasks.

First, we address the task of face naming. Face naming is the task of finding the correct associations between names that appear in the captions to faces that appear in the corresponding images, as illustrated in Figure 3.2. By obtaining a good naming of the faces, the burden of building a manually labelled data set of faces can be greatly alleviated. Equivalently, for the same annotation effort, much larger data sets can be obtained. We show that given a simple model for documents, face naming is a constrained clustering problem, and propose adapted algorithms to perform it



German Chancellor **Angela Merkel** shakes hands with Chinese President **Hu Jintao** (...)



Kate Hudson and **Naomi Watts**, Le Divorce, Venice Film Festival - 8/31/2003.

***Figure 3.2**: Examples of typical image-caption pairs in the Yahoo! News data set, and the result of automatic face naming.*

efficiently. We introduce a graph-based method to solve this task and compare to previously proposed generative approaches.

Second, we try to learn metrics for verification as in Chapter 2 but using captions as weak supervision. As a baseline, we can use the automatic naming of faces as supervision to train traditional metric learning algorithms. The errors that will be made by the naming process will penalise the predictive performance of the metric. Therefore, we also explore an alternative approach to learn useful metrics from noisily annotated faces, or, more exactly, noisily annotated sets of faces. We formulate the problem in the Multiple Instance Learning (MIL) framework and introduce MildML for learning metrics in such a setting. MildML stands for Multiple Instance Logistic Discriminant Metric Learning. As we will see, with this approach, the learnt metrics for face recognition outperform the ones obtained from using automatically named faces, without any user intervention.

In Section 3.2 we review work related to face naming and MIL metric learning. Then, in Section 3.3, we present generative and graph-based approaches for face naming, which are published in Guillaumin et al. [2008] and Guillaumin et al. [2010a]. In the same section, we also study the different means to learn a metric for face recognition automatically from the weak supervision of news documents, including MildML which was published in Guillaumin et al. [2010c]. In Section 3.4, we describe techniques to extract names from captions and features for faces in images, and we also present the *Labeled Yahoo! News* data set which is a subset of *Yahoo! News* that we have manually annotated. In Section 3.5, we evaluate and discuss the performance of the different methods for face naming and weakly supervised metric learning. We conclude in Section 3.6.

3.2 Related work on face naming and MIL settings

Our setting using caption-based supervision for naming and recognition is a particular case of using weaker forms of supervision for learning semantic relations. This is currently an active and broad line of research (Satoh and Kanade [1997], Pan et al. [2004], Fergus et al. [2005], Everingham et al. [2006], Li et al. [2007], Zhao et al. [2008], Luo et al. [2009]). Work along these lines include learning correspondence between keywords and image regions (Lazebnik et al. [2003], Verbeek and Triggs [2007]), and learning image retrieval and auto-annotation with keywords (Barnard et al. [2003], Grangier et al. [2006]). In these approaches, images are labelled with multiple keywords per image, requiring resolution of correspondences between image regions and semantic categories. Supervision from even weaker forms of annotation are also explored, *e.g.* based on images and accompanying text or video with scripts and subtitles.

For instance, Jain et al. [2007] use images with captions to associate textual topics to different people appearing in the news, while Quattoni et al. [2007] use similar data to learn embeddings for visual data. Bressan et al. [2008] propose to automatically illustrate travel blogs with relevant photographs. For videos, Everingham et al. [2006], Laptev et al. [2008] and Gaidon et al. [2009] use subtitles and transcripts of movies, the first to automatically identify characters, the second to train human action classifiers, the third to help mine human action sequences in videos. Finally, Zhao et al. [2008] propose to use images from the Web to learn classifiers for celebrities to perform recognition in videos while Ikizler-Cinbis et al. [2009] have a similar approach for actions.

The crux of those systems is to exploit the relations between different media, such as the relation between images and text, and between video and subtitles combined with scripts (Satoh et al. [1999], Sivic et al. [2009]). The correlations that can be automatically detected are typically less accurate – *e.g.* images and text associated using a web search engine like Google (Fergus et al. [2005], Berg and Forsyth [2006], Holub et al. [2008]) or other text-based retrieval (Mensink and Verbeek [2008], Ozkan and Duygulu [2010]) – than supervised information provided by explicit manual efforts. However, the important difference is that the former can be obtained at a lower cost, and therefore from much larger amounts of data, which may in practice outweigh the higher quality of supervised information.

The earliest work on automatically associating names and faces in news photographs is probably the PICTION system of Srihari [1991]. This system is a natural language processing system that analyses the caption to help the visual interpretation of the picture. The main feature of the system is that identification is performed only using face locations and spatial constraints obtained from the caption. No face similarity, description or characterisation is used, although weak discriminative clues (like male

vs. female) were included. Similar ideas have been successfully used in, for instance, the Name-it system (Satoh et al. [1999]), although their work concerned face-name association in news videos. The name extraction is done by localising names in the transcripts and video captions, and, optionally, sound track. Instead of simple still images, they extract face sequences using face tracking, so that the best frontal face of each sequence can be used for naming. These frontal faces are described using Eigenfaces method of Turk and Pentland [1991]. The face-name association can then be obtained with additional contextual cues, *e.g.* candidate names should appear just before the person appears on the video, because speeches and interviews are most often introduced by an anchor person.

Related work considering associating names to faces in a data set includes the generative mixture model of Berg et al. [2004a,b, 2007], where a mixture component is associated with each name for modelling the facial features in the data set. The main idea of this approach is to perform a constrained clustering, where constraints are provided by the names in a document, and the assumption that each person appears at most once in each image, which rules out assignments of several faces in an image to the same name. While in practice some violations of this assumption occur, *e.g.* people that stand in front of a poster or mirror that features the same person, there are sufficiently rare to be ignored. Additionally, the names in the document provide a constraint on which names may be used to explain the facial features in the document. A Gaussian distribution in a facial feature space is associated with each name. The clustering of facial features is performed by fitting a Gaussian mixture model (GMM) to the facial features with the expectation-maximisation (EM) algorithm (Dempster et al. [1977]), and is analogous to the constrained k-means clustering approach of Wagstaff and Rogers [2001].

The generative models in Berg et al. [2007] and also Luo et al. [2009] incorporate more information from the caption but, by leaving this out, all the methods discussed here use the same information and we can compare directly with our graph-based method described in Section 3.3.3. These work use cues such as word position, relative distance to closest punctuation or verb, etc., to build more complex models for the caption. In Jain et al. [2007] the caption is treated as a bag-of-words using a variant of latent Dirichlet allocation (Blei and Jordan [2003]). However, their main focus was to obtain people-specific distributions over words; for face naming the model was reported to perform worse than that of Berg et al. [2004a]. Similar caption features can be incorporated in graph-based methods by introducing additional weight terms that favour names of people who are likely to appear in the image based on textual analysis, although we did not explore this in our current work. To obtain a fair comparison between methods, we decided to leave these advanced models of captions out.

Note that, just as more complex models are possible for modelling the caption, it is also possible to have more complex models for the images. Notably, spatial relation-

ships (*c.f.* Gallagher and Chen [2009]) or body pose, as done by Luo et al. [2009], can be modelled to recognise social relations or actions, respectively.

Rather than learning a mixture model over faces constrained by the names in the caption, the reverse was considered by Pham et al. [2008]. They clustered face descriptors and names in a pre-processing step, after which each name and each face are both represented by an index in a corresponding discrete set of cluster indices. The problem of matching names and faces is then reduced to a discrete matching problem, which is solved using probabilistic models. The model defines correspondences between name clusters and face clusters using multinomial distributions, which are estimated using an EM algorithm. Other multimodal clustering techniques (*e.g.* Bekkerman and Jeon [2007]) can achieve similar goals.

Notably, the numerous clustering approaches that have been considered do not integrate metric learning, except Bilenko et al. [2004]. In this chapter, we propose to deploy our logistic discriminant metric learning approach (LDML) for the face naming task. Metric learning aims at finding a metric in a feature space that approximates a task-specific notion of semantic distance, and is one of the numerous types of methods that can provide robust similarity measures for the problem of face and, more generally, visual verification, as shown in Chapter 2. We want to use the particular form of weak supervision that comes from news documents. However, there is currently more work on metric learning from semi-supervised settings (Bilenko et al. [2004], Wang and Zhang [2008]) than from noisy and weak supervision (Yang et al. [2005]).

A way of looking at the data is to form small groups of instances that we call bags. Each bag represents the faces detected in an image. Labels for the bag come from the caption, where we detect putative names for the faces. An illustration of the MIL point of view is given in Figure 3.3.

Bags appear naturally in several computer vision settings: for instance, an image can be viewed as a bag of several regions or segments (Zhou and Zhang [2007]) – each of which is described by a feature vector – or a video sequence as a bag of frames (Yang et al. [2005]). Multiple instance learning (MIL, Dietterich et al. [1997]) refers precisely to the class of problems where data instances appear in bags, and each bag contains at least one instance for each label associated with the bag. MIL settings are particularly interesting because bag labelling is often easier to obtain than instance labelling. To support this idea, it is indeed easier to label a video sequence with the identities of the people that appear in it than labelling each detected face in each frame of the sequence.

The work closest related to ours is that of Jin et al. [2009], where the authors learn a metric from MIL data for image auto-annotation. To achieve this, the authors resort to constrained clustering. They use several clusters for each class, and, for each bag labelled with this class, assign one instance to the closest cluster. The metric is optimised to push cluster centres of different classes apart while minimising the

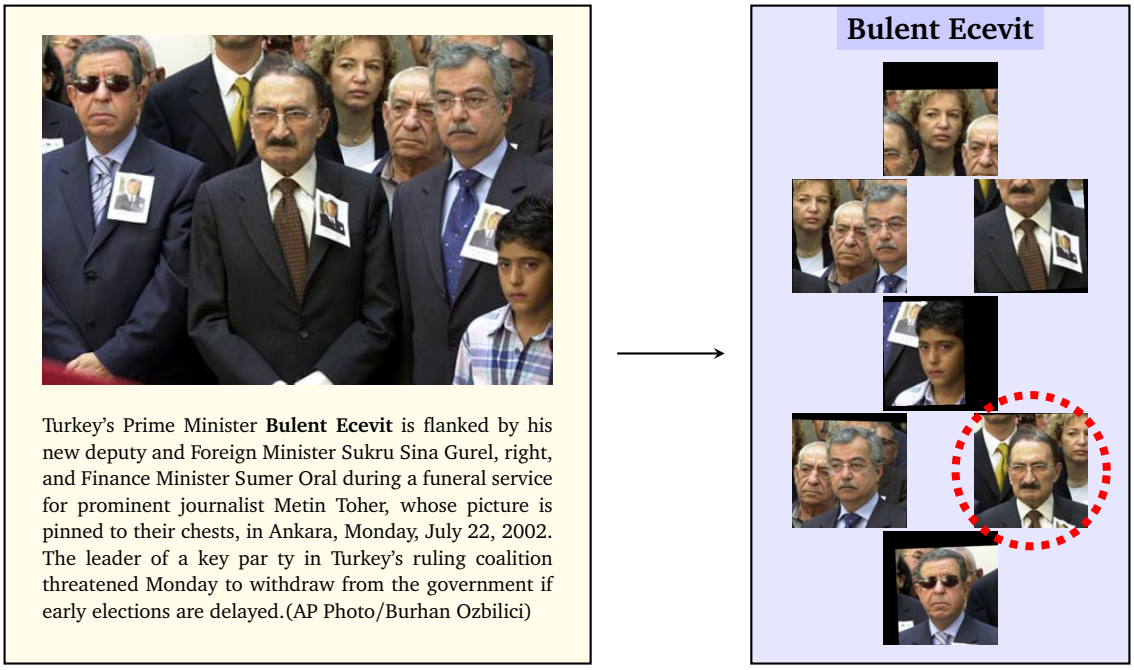


Figure 3.3: *Viewing news images with captions as a Multiple Instance Learning problem. The label "Bulent Ecevit" is assumed to be valid for at least one face in the face bag. The correct face image for Bulent Ecevit is highlighted in red.*

distances from instances to their associated cluster centres, with a objective function that relates to Fisher's discriminant analysis (*c.f.* Section 2.2.3). This setting is close to our proposed extension of LDML, which also uses clustering, but without resorting to cluster prototypes. Compared to their setting, though, we will also investigate the performance of metric learning when bag labels are noisy, which means that the underlying assumption of MIL will not be true in general: a bag may be assigned a label for which none of the instances is relevant.

3.3 Automatic face naming and recognition

In this section, we present different methods to perform face naming and metric learning from bag-level supervision. First, in Section 3.3.1, we highlight the three types of constraints that arise from our document model. Then we present a generative model (Section 3.3.2) and our graph-based method (Section 3.3.3) that deal with these constraints to associate names and faces. To obtain the solution efficiently, we propose a form of local optimisation in Section 3.3.4. In Section 3.3.5, we show how it is possible to learn a metric similar to LDML while performing the naming, and in Section 3.3.6, we present our MIL metric learning technique, MildML.

The St. Louis Cardinals from top left are Fernando Vina, J.D. Drew, Jim Edmonds, Albert Pujols, Tino Martinez, Edgar Renteria. Bottom from left are Miguel Cairo, Mike Matheny, Matt Morris, Chuck Finley, Andy Benes and manager Tony La Russa. (AP Photo/file)

***Figure 3.4**: This news document from the Yahoo! News data set illustrates the problem arising with "large" documents, which contain many names and many faces. Without any constraints, there are $2^{N\times F}$ possible assignments for this document, where N is the number of names, and F the number of faces. In this example with 12 names and 12 faces, there would be in the order of 10^{43} possible assignments. With the two additional constraints of unique usage of names and faces, the number shrinks to 10^{10}.*

3.3.1 Document-constrained clustering

The underlying idea behind face naming using news documents is to find the "best" assignment of names and faces in each document. From the limited set of names in each caption and the large amount of data where those people appear, we hope to be able to estimate for each document the most probable assignment of names and faces. For documents with N detected names and F detected faces, considering name ambiguity and possible multiple instances of each person, each name-pair is possible and should be considered. The number of possible assignments for the document is therefore $2^{N\times F}$.

When the document is large, this number will be too large to exhaustively search for the best solution. For instance, in Figure 3.4 we show a document with 12 names and 12 faces. This would suggest around $2^{144} \approx 10^{43}$ assignments to consider, which is intractable. Instead, the three following constraints can be used to reduce this number. These constraints come from assumptions that we make about news documents.

Constraint (i): Faces can only be assigned to names detected in the caption

Now explicit, this first constraint has in fact been implicitly assumed in the previous paragraphs. It makes the problem of naming much easier, because the system only has

Pictures of the Bali bombing suspects clockwise from top include: Umar (also known as Patek), Dulmatin (a.k.a. Amar Usman), M. Ali Imron (a.k.a. Alik), Idris (a.k.a. Jhoni Hendrawan, a.k.a. Gembrot), Umar (a.k.a. Wayan) and Imam Samudra. REUTERS/Handouts from Indonesian Police.

***Figure 3.5**: This news document from the Yahoo! News data set illustrates the challenges behind the ambiguity of names for people.*

to distinguish co-occurring people, that is up to a dozen, and not among thousands of different people that appear in the data set. However, if the caption is incorrectly analysed and a the correct name for a face is missed, this constraint will prevent, as a consequence, from associating the correct name to it. This can also happen if a person is simply not mentioned in the caption although his name is present elsewhere in the data set. Therefore, constraint (i) inherently prevents the systems to perform perfectly even if it is the key to the success of our methods. Note that the subset of names occurring in the caption is shared by all the faces in this image.

Constraint (ii): Faces can only be assigned to at most one name

As the previous constraint, stating that a face should correspond to only one name is not always accurate. To illustrate this, we refer to Figure 3.5. In this example, we see that several names in the caption are valid for naming a single face. This assumption more commonly breaks when a person is the focus of a story, and the journalist chooses to refer to this person using different named entities across the caption (for instance, *Barack Obama* later referred to as *President Obama* or *the Presisent of the United States of America*). Doing so, the journalist conveys more information about the subject, and also improves the style by avoiding repetitions. Importantly, using this constraint, associating faces and names in such a data set is therefore a constrained clustering problem: it states that faces can be assigned only to a unique name, *i.e.* to a single cluster.

***Figure 3.6**: Two examples where the image contains multiple faces of the same person (left: Edmund Stoiber, right: Keanu Reeves), hence breaking the assumption behind constraint (iii). The image on the right shows an extreme case where a single name should be assigned to dozens of faces in the image. Simple lower and upper limits on the face sizes are often an adequate solution.*

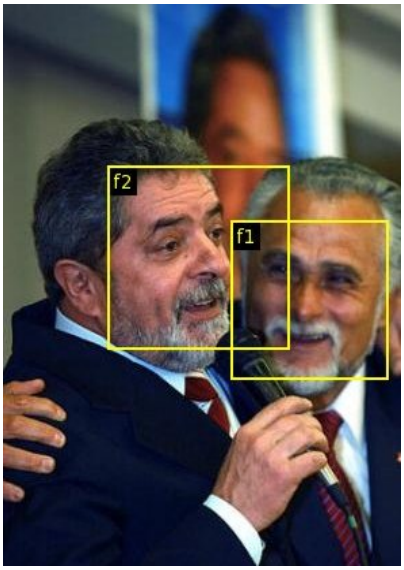
Constraint (iii): Names can only be assigned to at most one face in an image

Constraint (iii) assumes that it is impossible that a given person appears several times in the same image. Equivalently, it means that several faces in the same image cannot be assigned to the same cluster. Although it sounds sensible and also helps making the problem easier, this assumption is sometimes inconsistent with the observed data. Posters, mirrors, and computer generated images are common causes for this inconsistency, as shown in Figure 3.6.

***Null*-assignments and constrained clustering**

There are also cases where it is impossible to assign a face to a name because of the constraints or when the system estimates that none of the names are suitable for a face. A way to model this situation is to allow for an additional cluster, the *null cluster*. Its purpose is to collect all the faces that are not assigned to any named cluster, it is therefore not subject to constraint (iii). When a face is assigned to the *null* cluster, we will also call this situation a *null*-assignment for the face.

We can now summarise modelling of news documents by explicitly showing in Figure 3.7 the admissible assignments for a typical document under the three constraints



Brazilian presidential candidate Luiz Inacio Lula da Silva and the candidate for governor of the state of Sao Paulo, Jose Genoino, both of the Workers Party (PT) embrace during a news conference in Sao Paulo, Brazil on Monday Oct. 7, 2002. Unable to win a weekend election outright, the former labour boss is headed for an Oct. 27 showdown with the second-place finisher, candidate Jose Serra.(AP Photo/Dario Lopez-Mills).

Detected names are: *Luiz Iniacio Lula da Silva* (n_1), *Jose Serra* (n_2) and *Jose Genoino* (n_3).

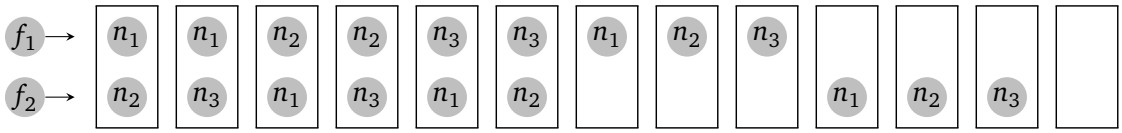


***Figure 3.7**: For a document with 2 detected faces and 3 names, shown on the top, there are 13 admissible assignments, shown below.*

described above. The enumeration of possible assignments $\mathcal{V}(N,F)$ for a document with N names and F faces leads to the following formula:

$$\mathcal{V}(N,F) = \sum_{p=0}^{\min(N,F)} p! \binom{N}{p} \binom{F}{p}. \tag{3.1}$$

This formula is interpreted as follows: for a number of name-face assignments p, pick p faces among F and p names among N, then pick one permutation (out of $p!$) of the names to associate to the faces. For most documents, this number is sufficiently small to allow for an exhaustive search for the best assignment. For the largest documents, it is still impractically large, but efficient techniques are proposed below in Section 3.3.4.

Many clustering algorithms can be adapted to handle such constraints. For instance, in bottom-up hierarchical clustering, we can propagate cannot-link constraints to unions of clusters as they are agglomerated. EM algorithms like K-means and mixtures of Gaussians can also be adapted by constraining the E-step to select a single admissible assignment. Below, we first detail how this can be done in the case of a Gaussian Mixture Model (GMM, Section 3.3.2), then we present in Section 3.3.3 our graph-based method.

3.3.2 Generative Gaussian mixture model

Previous work on naming faces in news images, which includes that of Berg et al. [2004a], use a constrained mixture model approach based on vectorial representations for the faces. To compare our graph-based approach to these methods, we propose to use the following model.

We associate a Gaussian density $\mathcal{N}(\mu_n, \Sigma_n)$ in the feature space with each name n, and an additional Gaussian is associated with *null*. The parameters of the latter will be fixed to the mean and variance of the ensemble of all faces in the data set, while the former will be estimated from the data. We denote with γ an assignment, which is a constrained set of face-name pairs $(\mathbf{x}, n)$.

The model for an image with faces $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_F\}$ is the following:

$$p(\mathbf{X}) = \sum_{\gamma} p(\gamma) p(\mathbf{X}|\gamma) \tag{3.2}$$

$$p(\mathbf{X}|\gamma) = \prod_{i=1}^{F} p(\mathbf{x}_i|\gamma) \tag{3.3}$$

$$p(\mathbf{x}_i|\gamma) = \mathcal{N}(\mathbf{x}_i; \mu_n, \Sigma_n) \tag{3.4}$$

where n is the name (or *null*) as given by the assignment $(\mathbf{x}_i, n) \in \gamma$. Given the assignment we have assumed the features $\mathbf{x}_i$ of each face f_i to be independently generated from the associated Gaussian.

The prior on γ influences the preference of *null* assignments. Using parameter $\theta \in \mathbb{R}$, we define:

$$p(\gamma) = \frac{\exp(-n_\gamma \theta)}{\sum_{\gamma'} \exp(-n_{\gamma'} \theta)} \propto \exp(-n_\gamma \theta) \tag{3.5}$$

where n_γ is the number of *null* assignments in γ.

For $\theta = 0$, the prior is uniform over the admissible assignments. As θ increases, the prior leans towards assigning more faces to a name, *i.e.* the prior probability for assignments with many null-assignments tends to zero. At the opposite, when θ is negative and decreases, assignments with many null-assignments are more likely. It our experiments, we study the performance of this model for values of θ ranging from large negative values (very few faces are named in the entire data set) to large positive values (every possible face is named as allowed by the constraints).

We use Expectation-Maximisation to learn the maximum likelihood parameters μ_n and Σ_n from the data. This requires computing the posterior probability $p(\gamma|\mathbf{X})$ for each possible assignment γ for each image in the E-step, which is intractable. Instead,

we constrain the E-step to selecting the assignment with maximum posterior probability. This procedure does not necessarily lead to a local optimum of the parameters, but is guaranteed to maximise a lower bound on the data likelihood (Neal and Hinton [1998]).

Considering that clusters will have very different sizes, one can question the robustness of using a full matrix for Σ_n. Instead, it is possible to restrict the parameter estimation in the M step to diagonal or even isotropic covariance matrices. The former is often a good trade-off between modelling flexibility and robustness of parameter estimation.

Additionally, one can consider sharing parameters among clusters. The underlying idea is that having observed many appearance variations of one specific person, it is possible to generalise this knowledge to other people. Assuming that this intra-class variation originates from a Gaussian distribution, only the mean of each cluster would be estimated independently.

3.3.3 Graph-based approach

We now propose a graph-based approach to address the same naming problem. A graph $G = (V, E)$ is defined such that each face f_i is represented as a vertex in V, and edges in E linking those vertices are weighted by a corresponding face similarity w_{ij}. Notably, this similarity is not restricted to metrics between vectorial face representations.

In this setting, the constrained clustering problem consists in identifying sub-graphs $\{Y_n\}$ of G, with one sub-graph for each name n. Consistently with the documents constraint (i), faces can only be assigned to the sub-graphs corresponding to the names in the caption of the corresponding document, and to at most one (constraint (ii)). Moreover, cannot-link constraints from constraint (iii) enforce sub-graphs not to overlap. In Figure 3.8 we illustrate these constraints for a simple example document.

The objective of the clustering is to maximise the sum $\mathcal{L}$ of sub-graph inner similarities under those sub-graph constraints:

$$\text{maximise} \quad \mathcal{L}(\{Y_n\}) = \sum_n \sum_{i \in Y_n} \sum_{j \in Y_n} w_{ij}. \tag{3.6}$$

Note that when $w_{ii} = 0$ this criterion does not differentiate between empty clusters and clusters with a single face. To avoid clusters with a single associated face, for which there are no other faces to corroborate the correctness of the assignment, we set w_{ii} to small negative values. The sub-graphs Y_n are obtained concurrently by directly maximising Equation 3.6, while preserving the document constraints. However,

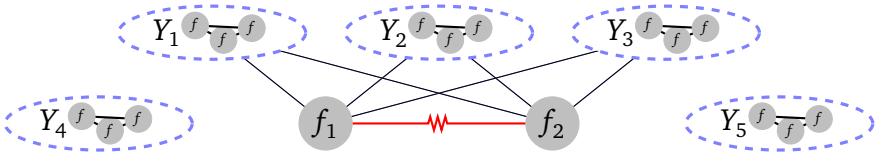


***Figure 3.8**: Example of a document with faces f_1 and f_2, and three names corresponding to sub-graphs Y_1, Y_2 and Y_3. Sub-graphs Y_4 and Y_5 cannot be used because of constraint (i). Sub-graphs do not overlap (constraint (ii)). Faces f_1 and f_2 cannot be assigned to the same sub-graph due to constraint (iii) shown as a red spring between the faces.*

finding the optimal global assignment is computationally intractable, and we thus resort to the approximate optimisation algorithm described below in Section 3.3.4.

In the case a vectorial representation of data with Euclidean distance as dissimilarity measure, this objective has a natural link to other clustering techniques like K-means. In K-means, clusters are represented by centroids μ_n which are the mean observation for each cluster n. The objective $D(\{Y_n\})$ of K-means can be written as:

$$\text{maximise} \quad D(\{Y_n\}) = -\sum_n \sum_{i \in Y_n} \|\mathbf{x}_i - \mu_n\|^2 \tag{3.7}$$

$$= -\sum_n \sum_{i \in Y_n} \sum_{j \in Y_n} \sum_{k \in Y_n} \frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_k)}{|Y_n|^2} \tag{3.8}$$

$$= -\sum_n \frac{1}{|Y_n|} \sum_{\substack{i \in Y_n \\ j \in Y_n}} \left(\|\mathbf{x}_i\|^2 - \mathbf{x}_j^\top \mathbf{x}_i \right) \tag{3.9}$$

$$= -\sum_n \frac{1}{2|Y_n|} \sum_{\substack{i \in Y_n \\ j \in Y_n}} \|\mathbf{x}_i - \mathbf{x}_j\|^2, \tag{3.10}$$

such that setting $w_{ij} = -\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2$ makes the relation to Equation 3.6 clear. However, we note the presence of a weighing term $|Y_n|^{-1}$ that accounts for the sizes of the clusters. Transferring this weight in Equation 3.6 would correspond to the optimisation of sub-graph densities. As explained in Guillaumin et al. [2008], optimising sub-graph densities leads to poor results. The reason is that the number of examples for each person varies greatly, from just a few to several hundreds. Using the sum of the densities tends to assign an equal number of faces to each name, as far as allowed by the constraints, and therefore does not work well for very frequent and rare people. Reciprocally, optimising Equation 3.6 without constraints to cluster data typically leads to unbalanced clusters, which might not be desirable in many applications (such as, *e.g.*, codebook construction).

3.3.4 Local optimisation at document-level

The relation between the objective of our graph-based approach and K-means also shows that the global optimisation problem is NP-hard. This can also be deduced from the link with the MAXCUT problem. Given the intractability of the problem, we seek approximate solutions. A first option would be to use spectral relaxation, which is a common technique for solving MAXCUT problems. In our case, it would consist in relaxing the problem to a semi-definite program. Considering the binary matrix $\mathbf{Y} \in \mathbb{B}^{F \times N}$ encoding the clustering (with $\mathbf{Y}_{in} = 1 \Leftrightarrow f_i \in Y_n$), this means to search for a real-valued solution for $\mathbf{Y}$ instead of a binary one. Then, solving the dual problem using Lagrange multipliers is a convex problem. Finally, the obtained real-valued solution needs to be cast back into a binary one. This projection does not necessarily yield the optimal binary solution.

However, the very strong structure in our constraints suggests us to exploit them directly. This structure is obvious: all the constraints appear at document-level. When looking at only one document, the constraints (i), (ii) and (iii) can be easily enforced, and the optimisation can be done exactly. It is therefore natural to iterate over documents and perform optimisation of Equation 3.6 per document. The approximation in the solution comes from the fact that the documents will not be optimised together.

To increase the stability of the solution, the sub-graphs are initialised with all faces that could be assigned, thus temporarily relaxing constraint (ii) and (iii), but keeping (i). Constraint (ii) and (iii) are progressively enforced as a consequence of the fully constrained document-level optimisation. Consequently, after a full iteration over images, constraints (i), (ii) and (iii) are correctly enforced. The iteration continues until a fixed-point is reached, which takes in practice often less than 10 iterations.

We now detail how the optimisation can be done exactly for a document. As already stated, for a document with N names and F faces, the number of assignments satisfying all the constraints is $\mathcal{V}(N,F) = \sum_{p=0}^{\min(N,F)} p! \binom{N}{p} \binom{F}{p}$, see Equation 3.1. The exhaustive search is reasonable for small documents. But for large ones, this number is still impractically large. For instance, the largest document in the *Yahoo! News* data set has $F = 12$ faces and $N = 7$ names, and $\mathcal{V}(N,F)$ is in the order of 10^7.

Given the fact that assignments share many common sub-assignments – the underlying structure is a lattice – a large efficiency gain can be expected by not re-evaluating the shared sub-assignments. We therefore introduce a reduction of the optimisation problem to a well-studied minimum cost matching in a weighted bipartite graph (*c.f.* Cormen et al. [2001]). This modelling takes advantage of the underlying structure and can be implemented efficiently. Its use is limited to objectives that can be written as a sum of "costs" $c(f,n)$ for assigning face f to name n, *i.e.* to cluster Y_n, which is still very general.

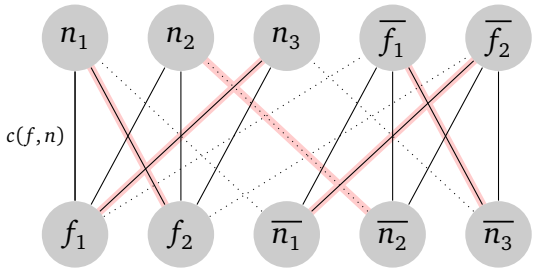


***Figure 3.9**: Example of the weighted bipartite graph corresponding to a document with two faces and three names. For clarity, costs are not indicated, and edges between vertices and their null copies are dotted. An example of a matching solution is given with the highlighted lines, it is interpreted as assigning face f_1 to name n_3, f_2 to n_1, and not assigning name n_2.*

The names and faces problem differs from usual bipartite graph matching problem because we have to take into account *null*-assignments, and this *null* value can be taken by any number of faces in a document. This is handled by adding as many *null* nodes as there are faces and names in the graph. A face f can be paired with any name or its own copy of *null*, which is written $\overline{f}$, and reciprocally, a name n can be paired with any face or its own copy of *null*, written $\overline{n}$. The presence of an edge between f and n is echoed by the one between $\overline{n}$ and $\overline{f}$ to ensure that $n+f$ assignments are always possible. A corresponding graphical representation is shown in Figure 3.9.

The weights of the pairings are simply the costs of assigning a face f_i to the sub-graph Y_n, *i.e.*:

$$c(f_i, n) = -\sum_{j \in Y_n} w_{ij}. \tag{3.11}$$

A bipartite graph matching problem is efficiently solved using the Kuhn-Munkres algorithm (also known as the Hungarian algorithm, Kuhn [1955]) which directly works on a cost matrix. The cost matrix modelling our document-level optimisation is a squared matrix with $n+f$ rows and columns where the absence of edge is modelled with infinite cost. The rows represent faces and *null* copies of names, while columns represent names and *null* copies of faces. See Figure 3.10 for the cost matrix modelling the matching problem of Figure 3.9. It is then straightforward to obtain the minimum cost and the corresponding assignment, as highlighted in the example matrix.

In Figure 3.11 we show how the processing time grows as a function of the number of admissible assignments in a document for the Kuhn-Munkres algorithm compared to an exhaustive ("brute-force") search over all admissible assignments. For reference,

$$\begin{bmatrix} c(f_1,n_1) & c(f_1,n_2) & c(f_1,n_3) & c(f_1,\overline{f_1}) & \infty \\ c(f_2,n_1) & c(f_2,n_2) & c(f_2,n_3) & \infty & c(f_2,\overline{f_2}) \\ c(\overline{n_1},n_1) & \infty & \infty & c(\overline{n_1},\overline{f_1}) & c(\overline{n_1},\overline{f_2}) \\ \infty & c(\overline{n_2},n_2) & \infty & c(\overline{n_2},\overline{f_1}) & c(\overline{n_2},\overline{f_2}) \\ \infty & \infty & c(\overline{n_3},n_3) & c(\overline{n_3},\overline{f_1}) & c(\overline{n_3},\overline{f_2}) \end{bmatrix}$$

***Figure 3.10**: Example of the* 5×5 *cost matrix representing the bipartite graph matching formulation of document-level optimisation for the Kuhn-Munkres algorithm, for a document with two faces and three names. The costs* $c(f_i,n_j)$ *are set to the negative sum of similarities from* f_i *to vertices in the sub-graph* Y_{n_j}*,* $c(f_i,\overline{f_i})$ *are set to a constant threshold value* θ*, and* $c(\overline{n_j},\cdot)$ *are set to zero. For* $c(\overline{n_j},n_j)$*, this is because we do not model any preference for using or not certain sub-graphs. Infinite costs account for absence of vertex. The same solution as in Figure 3.9 is highlighted.*

we also include the min-cost max-flow algorithm of Guillaumin et al. [2008], but it is slower than Kuhn-Munkres because the solver is more general than bipartite graph matching.

The overall approximate optimisation algorithm for our constrained clustering problem is given in Algorithm 1. For the computation of the cost matrix, we have the option of using the fixed clustering $\mathbf{Y}'$ of the previous iteration or the current clustering $\mathbf{Y}$ that is being updated as we go over the images. Our experiments do not show any major influence of this choice. We also observe that the convergence is fast, typically in less than 10 iterations of the outer loop.

Algorithm 1: Face naming under document constraints.

Input: Documents d as sets of faces $\mathbf{f}_d$ and names $\mathbf{n}_d$, Cost matrix function.
Output: Assignment matrix $\mathbf{Y}\in\mathbb{B}^{F\times N}$.

foreach *name* n **do** /* *Initialise clusters* */
  $\mathbf{Y}(n)\leftarrow\left\{f \;\middle|\; \exists d,(f,n)\in\mathbf{f}_d\times\mathbf{n}_d\right\}$
end
repeat
  $\mathbf{Y}'\leftarrow\mathbf{Y}$
  foreach *document* d **do** /* *Document-level optimisation* */
    $\mathbf{M}\leftarrow$ CostMatrix($\mathbf{f}_d$,$\mathbf{n}_d$,$\mathbf{Y}$ or $\mathbf{Y}'$)
    $\mathbf{Y}(\mathbf{f}_d,\mathbf{n}_d)\leftarrow$ KuhnMunkres($\mathbf{M}$)
  end
until $\mathbf{Y}'=\mathbf{Y}$

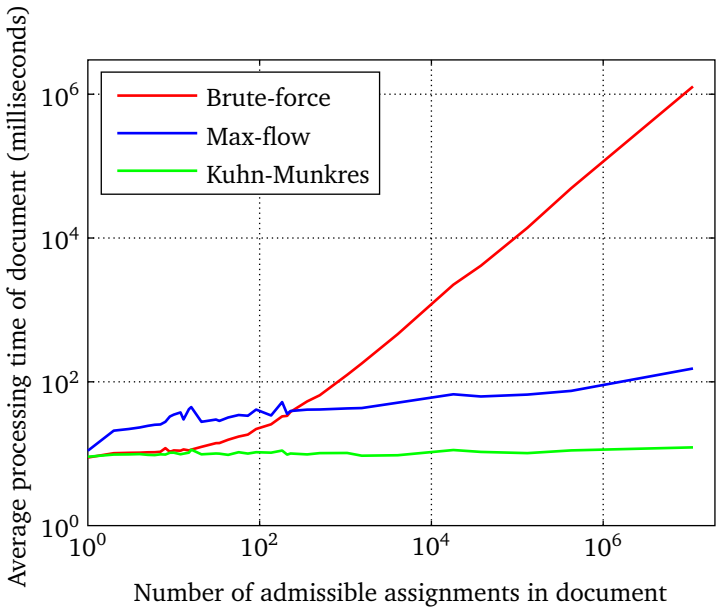


***Figure 3.11**: Average processing time of the three algorithms as a function of the number of admissible assignments in documents. The average is computed over 5 runs of randoms costs, and over all documents that have the same number of admissible assignments. The Kuhn-Munkres algorithm combines low overhead and slow growth with document complexity. Note the log scales on both axes.*

Interestingly, the likelihood of the generative GMM from Section 3.3.2 can also be written in the form of costs $c(f,n)$ that should be minimised:

$$c(f_i,n) = -\ln \mathcal{N}(\mathbf{x}_i;\mu_n,\Sigma_n) \tag{3.12}$$

$$= \frac{1}{2}\left(d_{\Sigma_n^{-1}}(\mathbf{x}_i,\mu_n) + \ln|\Sigma_n| + c_n\right), \tag{3.13}$$

where d is the Mahalanobis distance from Equation 2.1, and $c_n = D\ln(2\pi)$ is a constant cost common to all clusters (including the *null* cluster), which can be removed in the corresponding line in the cost matrix. The maximum *a posteriori* maximisation with a prior distribution on assignments is also easily incorporated into the bipartite graph matching formulation by adding θ of Equation 3.5 to each $c(f,\overline{f})$. One θ cost is added for each *null*-assigned face, so the cost is $n_\gamma\theta$ for the selected assignment γ, which is precisely the negative log-prior of Equation 3.5.

3.3.5 Joint metric learning and face naming from bag-level labels

In this section we now consider learning a metric for face recognition directly from the data consisting of news images with captions. In such a setting, we have partial

knowledge of the labels of the instances in a bag $\mathbf{X}_d$ of faces appearing in document d, given by a set of labels which indicate that at least one example of class n is in d. This setting is also known as Multiple Instance Multiple Label Learning (MIML). Below, we adapt LDML to explicitly estimate the labels of the instances in each bag from the bag-level annotation.

To learn an LDML metric in this setting, we optimise the objective in Equation 2.22 jointly over the metric parametrised by $\mathbf{L}$ and over the label matrix $\mathbf{Y}$ subject to the label constraints given by the bag-level labelling, which means we will make the same assumptions as described in Section 3.3.1:

$$\underset{\mathbf{Y},\mathbf{L},b}{\text{maximise}} \quad \mathcal{L} = \sum_{i,j} (y_i^\top y_j) \log p_{ij} + (1 - y_i^\top y_j) \log(1 - p_{ij}). \tag{3.14}$$

Unfortunately, the joint optimisation is intractable. For fixed $\mathbf{Y}$, it is precisely the optimisation problem discussed in Section 2.3.1. When optimising $\mathbf{Y}$ for fixed $\mathbf{L}$ and b, we can rewrite the objective function as follows:

$$\mathcal{L} = \sum_{i,j} (y_i^\top y_j)(\log p_{ij} - \log(1 - p_{ij})) + c = \sum_{i,j} w_{ij} (y_i^\top y_j) + c, \tag{3.15}$$

where

$$c = \sum_{ij} \log(1 - p_{ij}), \text{and} \tag{3.16}$$

$$w_{ij} = b - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \text{ are constants.} \tag{3.17}$$

This optimisation problem is NP-hard, and we therefore have to resort to approximate optimisation techniques.

Observing that the only non-constant terms in Equation 3.15 are those for data points in the same class, we can rewrite the objective for a particular instantiation of $\mathbf{Y}$ as

$$\underset{\mathbf{Y}}{\text{maximise}} \sum_{n=1}^{C} \sum_{i \in Y_n} \sum_{j \in Y_n} w_{ij}, \tag{3.18}$$

where Y_n is the set of indices of instances that are assigned to class n, *i.e.* $Y_n = \{i | y_i^{(n)} = 1\}$. Equation 3.18 reveals that we are solving a constrained clustering problem which is exactly the one discussed in the previous section (Section 3.3.3) with a choice of weights that originate from the specific objective function of LDML. We have to assign the instances to clusters corresponding to the classes so as to maximise the sum of intra-cluster similarities w_{ij}, as in Equation 3.6.

To obtain an approximate solution for $\mathbf{Y}$ we naturally resort to the same approximate optimisation detailed in Section 3.3.4 and summarised in Algorithm 1. The label optimisation is initialised by assigning all instances in a bag to each permissible class according to the bag label. We then maximise $\mathcal{L}$ with respect to the labels of the instances in each bag in turn, also enforcing that each instance is assigned to exactly one class. The optimisation at bag level is done exactly and efficiently using bipartite graph matching.

3.3.6 Multiple instance metric learning

Here, we still consider another way to learn a metric from the same type of data. Instead of supervision on the level of single instances, or pairs of instances, we assume here that the supervision is provided at the level of pairs of bags of examples. This naturally leads to a multiple-instance learning (MIL) formulation of the metric learning problem, which we refer to as MildML, for Multiple Instance Logistic Discriminant Metric Learning.

Let us denote a bag of examples as $\mathbf{X}_d = \{\mathbf{x}_1^d, \mathbf{x}_2^d, \ldots, \mathbf{x}_{N_d}^d\}$, where N_d is the number of examples in the bag. The supervision is given by labels $t_{de} \in \{0, 1\}$ that indicate whether for a pair of bags $\mathbf{X}_d$ and $\mathbf{X}_e$ there is a pair of examples $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbf{X}_d \times \mathbf{X}_e$ such that $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to the same class. If there is such a pair of examples then $t_{de} = 1$, and $t_{de} = 0$ otherwise.

The objective in Equation 2.22 is readily adapted to the MIL setting by extending the definition of the distance to compare bags (Jin et al. [2009]) with:

$$d_{\mathbf{M}}(\mathbf{X}_d, \mathbf{X}_e) = \min_{\mathbf{x}_1 \in \mathbf{X}_d, \mathbf{x}_2 \in \mathbf{X}_e} d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2), \tag{3.19}$$

which leads naturally to the following optimisation:

$$\underset{\mathbf{M}, b}{\text{maximise}} \quad \mathcal{L} = \sum_{d,e} t_{de} \log p_{de} + (1 - t_{de}) \log(1 - p_{de}), \tag{3.20}$$

where $p_{de} = \sigma(b - d_{\mathbf{M}}(\mathbf{X}_d, \mathbf{X}_e))$.

This objective makes bags that share a label closer, and pushes bags that do not share any label apart. For a negative pair of bags, all the pairs of instances that can be made from these two bags are eventually pushed apart since the pair of examples with minimum distance is.

We optimise the objective iteratively by alternating (i) the pair selection by the min operator for a fixed metric, and (ii) the optimisation of the metric for a fixed selection

of pairs. The optimisation in the second step is exactly of the same form as the optimisation of the low-rank version of LDML presented in the Section 2.3.1. For a given selection of pairs, we perform only one line search in the direction of the negative gradient, such that the pair selection is performed for each computation of the gradient. This way we do not spend many gradient steps optimising the metric for a selection of pairs that might be inaccurate. This optimization method relates to sub-gradient methods.

Note that MildML, contrary to the adapted version of LDML presented in the previous section, does not try to specifically assign labels to instances, and instead for each pair of bags only a single pair of instances is used to learn the metric. The benefit, illustrated in Figure 3.12, is that this single pair is robust to noise in the data, but the drawback is that many pairs of examples are lost, especially the negative ones occurring inside a bag, which may impact the quality of the learnt metric.

3.4 Data set

Before describing the data we use in our experiments, let us first provide some details about the challenges and the general context concerning the analysis of text and captions.

3.4.1 Processing of captions

Behind the extraction of names of individuals in the captions of our documents, there is the natural language processing technique of *named entity recognition* (NER). The goal is to segment and label accordingly the words of the caption, composed of several sentences. This problem of segmentation on sequences also arises in other scientific field, such as speech recognition or computational biology. Hidden Markov models (HMM, Hammersley and Clifford [1971]) and stochastic grammars have been successfully applied in those fields (*c.f.* Rabiner [1989], Durbin et al. [1998]). Because they are generative models, they explicitly try to estimate the density over the entire parameter space by generating all possible observation sequence.

To overcome these shortcomings, Lafferty et al. [2001] introduced a discriminative model named Conditional Random Field (CRF). CRFs are a type of undirected graphical models. In graphical models, each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. For text analysis, linear-chained models are commonly considered.

Since 2001, many research groups around the world have developed very efficient Part-Of-Speech (POS) taggers, topic segmenters, and NER systems, and provided

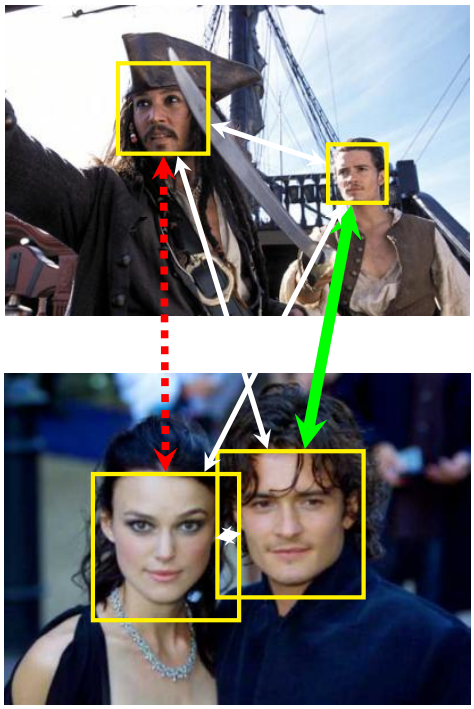


***Figure 3.12**: Illustration of the potential benefit of MildML compared to the naming baseline. In this pair of documents, two labels are shared ("Johnny Depp" and "Orlando Bloom"). However, only one of these two persons effectively appears in both images. Therefore, if the naming procedure incorrectly assigns both names in both images, then an incorrect positive pair (shown in dashed red) is used during training. On the contrary, since MildML uses only the pair of faces that are most similar, the selection of this pair is likely to be correct, and the pairs used for training are error-free. However, MildML then ignores the five negative pairs that can be formed from this pair of images (shown by white arrows in addition to the red arrow).*

them freely online, with pre-trained models. Specifically, we have used the CRF implementation of NER from Deschacht and Moens [2006]. Even in the event of perfect named entity recognition in the caption, the analysis of the names is difficult and challenging. This is the consequence of the ambiguity of names. Indeed, the same person can be named using different text strings: "George Bush" and "President of the United States of America", for instance, denote the same individual, while "George Bush" could refer to either "George H.W. Bush", the father, or "George W. Bush", the son. By looking back at Figure 3.5, we can see that the ambiguity problem can appear in a single document, as several names and aliases are used to identify the people appearing in the image.

In the following, we will focus on techniques to exploit the detected names in a caption as a set of names. At this point, it is worth noting that more complex models have

Paul McCartney (R) with Heather Mills (L) arrive at the 74th annual Academy Awards in Hollywood, March 24, 2002. REUTERS/Fred Prouser

***Figure 3.13**: This news document from the Yahoo! News data set illustrates the possibility of using additional textual cues to improve the face naming process. Here, the relative positions are clearly indicated by "(L)" and "(R)" in the caption.*

been proposed in the literature (see Berg et al. [2007], Luo et al. [2009]). This originates from the fact that captions of news images often possess additional cues that could help the naming process. For instance, in Figure 3.13, the text includes spatial indications such as "(L)", "(R)" to help the reader associate the nearby names to the correct people in the image, according to their relative position and their absolute position in the image. Among the possible textual cues to use, one can mention:

- the Part-Of-Speech (POS) tag of the word immediately before and immediately after the name,
- the location of the name in the caption, simply measured by the word count from the beginning of the caption,
- the distance to the nearest punctuation or spatial indication,
- action verbs relating to the detected name.

These cues are used as additional textual features associated to the name. They are used to estimate the probability of the corresponding name to appear as a detected face in the image. However, these additional cues must be handled with care. The main reason for this is the imperfection of detectors, but visual confusion is also possible. For instance, in Figure 3.13, the woman standing between (but also behind) the two main characters of the picture is indeed detected by state-of-the-art face detectors. Therefore, as a face, it complies with the indications of being on the left of Paul

McCartney, so could be named as Heather Mills, especially if the correct instance face for Heather Mills is missed. Complex models, although able to profit from more textual and visual clues to estimate their parameters, will also require careful tuning to avoid pitfalls, and therefore are not good candidates to compare naming algorithms. Instead, in order to ensure fair comparison, we will use the simple document model consisting of unordered sets of names and faces.

3.4.2 *Labeled Yahoo! News*

The *Yahoo! News* database was first introduced by Berg et al. [2004a], and was gathered in 2002–2003. It consists of news images and their captions describing the event appearing in the image. On this large collection, a face detector was applied, as well as a named entity detector. Documents with no detected faces or names were removed, leading to a database with roughly 15000 image-caption pairs, with 15280 detected named entities and 22750 detected faces. There are wide variations in appearances with respect to pose, expression, and illumination, as shown in two examples in Figure 3.1. Ultimately, the goal was to automatically build a large data set of annotated faces, so as to be able to train face recognition systems on it.

Unfortunately, this data set is not publicly available and does not possess any ground-truth labelling. Therefore, the face naming methods based on variants of this *Yahoo! News* "data set" that were published (Berg et al. [2004a, 2007], Guillaumin et al. [2008, 2010a,c], Ozkan and Duygulu [2006, 2010], Mensink and Verbeek [2008], Pham et al. [2008, 2010]) have not been directly compared on a fair basis.

With growing efforts towards systems that can efficiently query data sets for images of a given person, or use the constraints given by documents to help face clustering, it has become important for the community to be able to compare those systems with a standardised data set. We have therefore introduced the *Labeled Yahoo! News* data set, which is freely available online for download.[1] From the original *Yahoo! News* data, we have applied the OpenCV implementation of the face detector of Viola and Jones [2004] and removed documents without detections.

We then applied the named entity detector of Deschacht and Moens [2006] described in the previous section to gather the set of names that appear in the data set and added the names from the *Labeled Faces in the Wild* data set that were missed by the detector. Finally, we filtered the captions of document in the search for presence of these names.

We manually annotated the 28204 filtrate documents for the correct association of detected faces and detected names. For faces detections that are not matched to a name, the annotation indicates which of the three following possibilities is the case:

[1] `http://lear.inrialpes.fr/data/`

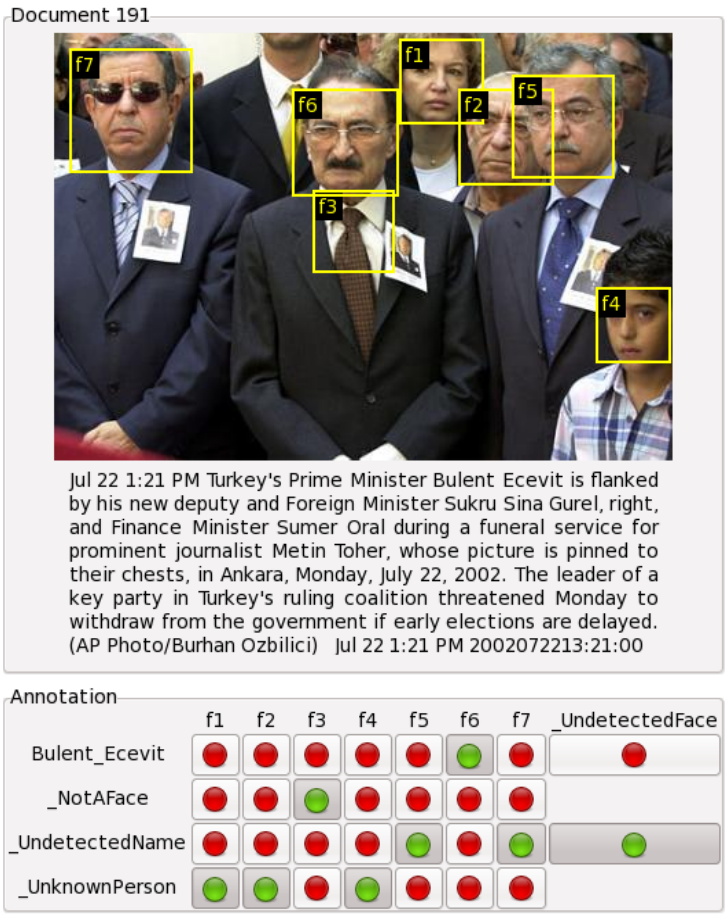


***Figure 3.14**: Example of a Labeled Yahoo! News document annotation in our annotation tool. This complex document contains faces of unknown persons, a false face detection, missed names and missed faces of Metin Toher, whose name is also undetected.*

1. it is a false detection (not a face),

2. it depicts a person whose name is not in the caption, or

3. it depicts a person whose name was missed by the name detector.

Likewise, for names that are not assigned to a face, the annotation contains the information of a possibly missed face. Finally, we also annotate the document when an undetected face matches an undetected name. Although this information is not used in our system, it allows for a very efficient update of the ground-truth annotations if we were to change the face detector or named entity detector. An example of annotation is shown in Figure 3.14.

This annotation is an extension to the *Labeled Faces in the Wild* data set with more faces, more names, and additional information about the document structure of the data.

In order to divide this data set into completely independent training and test sets, we have proceeded the following way. Given the 23 person queries used by Ozkan and Duygulu [2006, 2010], Guillaumin et al. [2008, 2010a] and Mensink and Verbeek [2008], the subset of documents containing these names is determined. This set is extended with documents containing "friends" of these 23 people, where friends are defined as people that co-occur in at least one image (following Mensink and Verbeek [2008]). This forms set A. From the remaining set of documents we discarded the 8133 ones that contain a name or a face from any person appearing in set A, such that it is now completely disjoint of set A.

Set A contains 9362 documents, 14827 faces and 1072 different names in the captions: because of the specific choice of queries, it has a strong bias towards politicians. Set B contains 10709 documents, 16320 faces and 4801 different people, relatively many athletes. The average number of face images for each person is significantly different between the two sets. Due to these differences between the two sets, we report performance for verification by averaging the results obtained from training on either set and testing on the other.

3.4.3 Feature extraction

Before extracting features from the face images, we apply a face alignment procedure. Alignment compensates for the imperfect location and scale estimation of the faces by the face detector. We use the same technique used for the *Labeled Faces in the Wild* data set, called funnelling (Huang et al. [2007a]). In this way, the full face description pipeline is exactly the same as in Chapter 2. Funnelling consists in iteratively transforming the image to lower its entropy according to a distribution field obtained from (unsupervised) training data. The distribution field is not taken over pixel values but each pixel is modelled as a mixture of prototype SIFT descriptors, learnt from the data using a GMM. Other alignment procedure exist (*c.f.* Lucas and Kanade [1981], Cox et al. [2009]), and for instance facial features can also be used to estimate a transformation to a model face (*c.f.* Urschler et al. [2009], Funes Mora [2010]).

We have already extensively discussed the description of faces in Section 2.4. We will therefore use the same SIFT-based descriptors as in Chapter 2 for the detected faces. Since we are considering a generative model where the robustness of parameter estimation is crucial, we restrict ourselves to scales 2 and 3 of the descriptor. It is 2304D instead of 3456D, without any loss of performance for the identification task, as was observed in Figure 2.16.

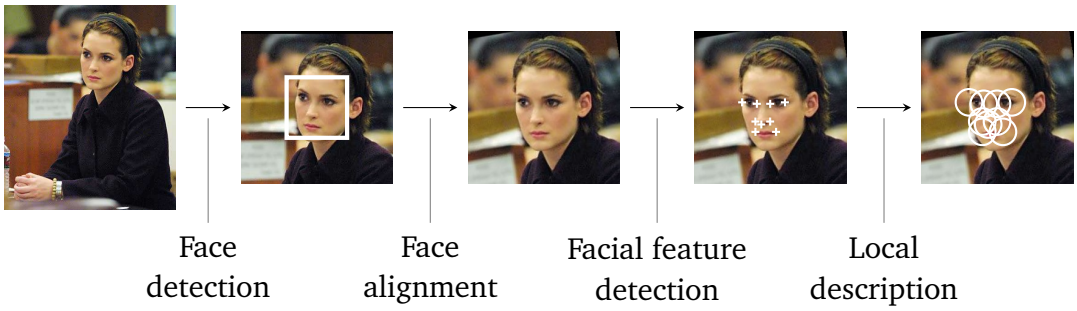


***Figure 3.15**: Illustration of the full face processing pipeline. After detecting faces in the images, face images are aligned and given to the facial feature detector. Then, local appearance descriptors of these fiducial points are extracted and concatenated to obtain a vectorial representation of the faces.*

Graph-based approaches, contrary to the Gaussian mixture model, do not rely on cluster prototypes nor covariance estimates, so they can be used with other types of face similarities (*c.f.* Ozkan and Duygulu [2006], Guillaumin et al. [2008]) than vectorial representations using a Mahalanobis distance. In order to fairly compare our graph-based method to the generative approach, we will only use weights that are derived from the Mahalanobis distance between the vectorial representation of two faces, and we will use the same face descriptors.

3.5 Experiments

We present our experimental results in three parts. In the first, we evaluate our different methods to associate names and faces. In these experiments, we also consider the impact of using learnt metrics, optimised on independent training data. In the second section, we evaluate the metrics learnt from images with captions for verification, and in the third, for face naming.

3.5.1 Face naming with distance-based similarities

In this section, we first present our experiments on associating names to faces in a data set of images with captions with a given face representation.

Experimental protocol

We use the two sets of the *Labeled Yahoo! News* data set to allow learning of metrics on independent training data. The learning is performed on set B, and the naming

algorithms are run on set A after projecting the data. We learn the similarity measures using LDML and PCA, and also compare to the Euclidean distance. Then, we apply on the test set the generative and graph-based methods described in Section 3.3.2 and Section 3.3.3 respectively and measure their performance. We call the performance measure we use the "naming precision". It measures the ratio between the number of correctly named faces over the total number of named faces. Recall that some faces might not be named by the methods (*null*-assignments). Those faces are ignored in the measure.

To plot the results, we vary a parameter that influences, for each of the naming algorithms, the number of effectively named faces. For both approaches, we denote by θ this parameter, noting that we already discussed it for the GMM approach (*c.f.* Equation 3.5). For a choice of this parameter, we obtain a specific number of name-face associations and the corresponding naming precision. To obtain an interesting range of number of name-face associations, we explore a large range of parameter values in order to find points in 50 (for the generative approach) to 100 (for the graph-based approach) segments uniformly dividing the target range. This exploration is performed using a divide-and-conquer mechanism that divides the parameter range by 2 at each refinement step until all segments are attained. For this matter we use the assumption that the variation of the number of named faces is monotonous with respect to the parameter. We stop the refinement when the parameter changes less than 10^{-3}, such that certain segments may remain unattained. This mechanism is a straightforward generalisation of dichotomic search.

For the generative approach, we have a parameter $\theta \in \mathbb{R}$ in the definition (*c.f.* Equation 3.5) of the prior distribution on admissible assignments that influences the preference for *null*-assignments. For the graph-based approach, we need to define weights w_{ij} between two face images. We consider four definitions of weights: "hard weights", "soft weights", "hinge weights" and "log weights". They are formally given in Table 3.1 and illustrated in Figure 3.16. All definitions use a threshold parameter $\theta \in \mathbb{R}^+$ such that the following properties hold for each weight definition:

$$w_{ij} > 0 \text{ when } d(\mathbf{x}_i, \mathbf{x}_j) < \theta, \tag{3.21}$$

$$w_{ij} = 0 \text{ when } d(\mathbf{x}_i, \mathbf{x}_j) = \theta, \tag{3.22}$$

$$w_{ij} \leq 0 \text{ when } d(\mathbf{x}_i, \mathbf{x}_j) > \theta. \tag{3.23}$$

The θ parameter of the weights also influences the preference for *null*-assignments.

Experimental results

In Figure 3.17, we show the naming precision using the original descriptor and compare the generative approach (GMM) with the four definitions of weights for the

Weight name	Weight definition
Hard	$w_{ij} = \left[d(\mathbf{x}_i, \mathbf{x}_j) < \theta \right] \in \{0, 1\}$
Soft	$w_{ij} = \theta - d(\mathbf{x}_i, \mathbf{x}_j)$
Hinge	$w_{ij} = [\theta - d(\mathbf{x}_i, \mathbf{x}_j)]_+ \quad$ where $[\cdot]_+ = \max(0, \cdot)$
Log	$w_{ij} = \log\left(1 + \exp\left(\theta - d(\mathbf{x}_i, \mathbf{x}_j)\right)\right) - \log 2$

Table 3.1: *Definitions of the graph weights we consider in our experiments. The corresponding functions are plotted in Figure 3.16.*

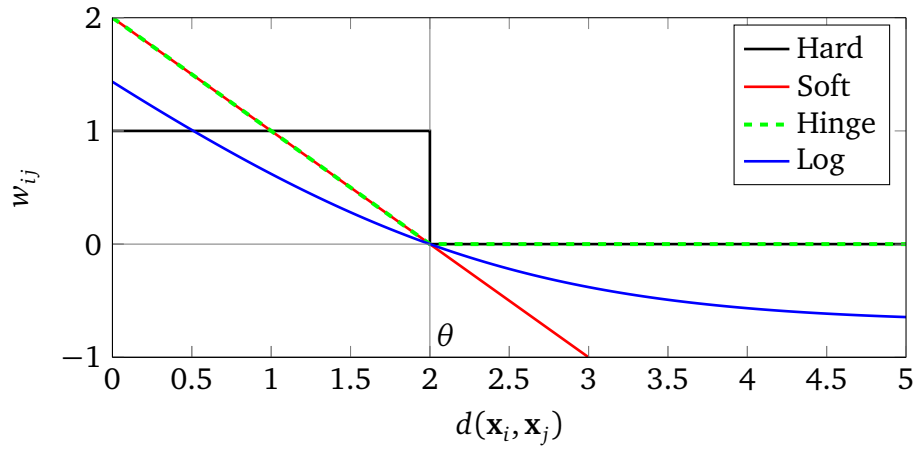


Figure 3.16: *Graph weights as a function of the pairwise distance $d(\mathbf{x}_i, \mathbf{x}_j)$ for the different definitions in Table 3.1. There are shown for the choice of $\theta = 2$.*

graph-based method. Using the original descriptor means that we are using the Euclidean distance for computing those weights. We observe that graph-based methods outperform the generative model when the parameter value is conservative. This means that only a small amount of faces is associated with names, and therefore the parameter estimation of the generative method is poor. On the contrary, when naming many faces, the graph-based method is penalised by imperfect similarities that add incorrect examples in the clusters.

We find that soft versions of weights (Soft or Log) yield more stable results than the harder weights obtained using Hard or Hinge. This is easily understood because the Hard thresholding process erases the differences between values if they fall on the same side of the threshold. The Hinge weights performs similarly for a similar reason, although the erase is done only on one side of the threshold. Similar observations hold for all our other experiments, but for clarity, we will report results only for Soft and Log in the following experiments.

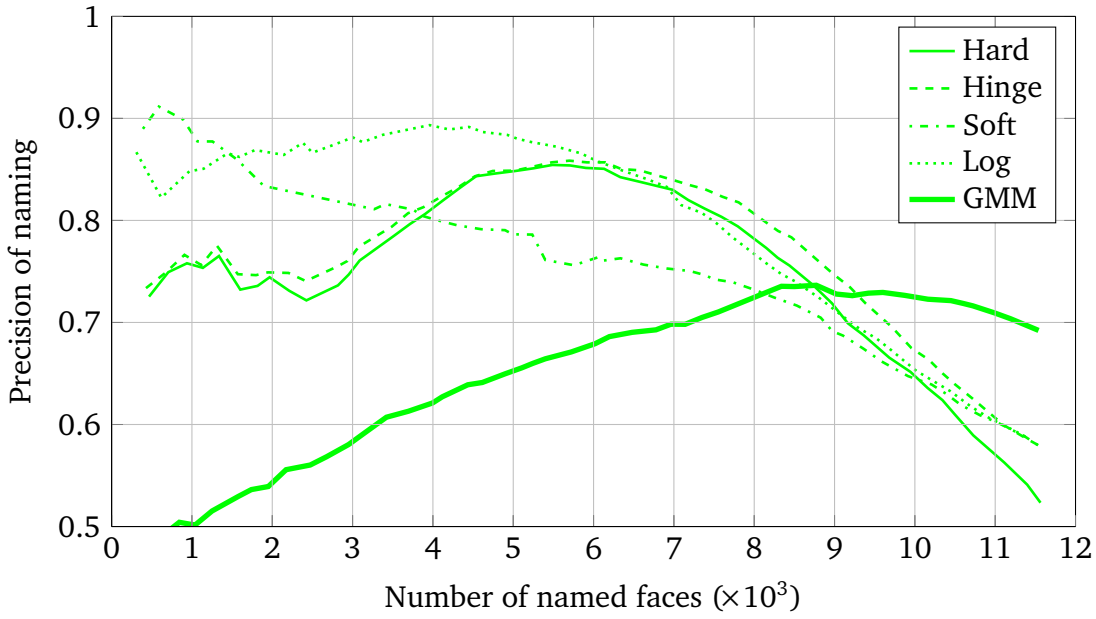


Figure 3.17: *Plot of the naming precision of the different approaches using the original descriptor and the Euclidean distance.*

In Figure 3.18, we show the performance of the generative approach for different projections of the descriptor, and compare with the unprojected one. We first observe that projecting the data performs better than the original descriptor. By reducing the number of parameters to 100D, and further to 20D, the estimation of the parameters is made more robust and therefore performance increases, especially when fewer faces are named and therefore used for the estimation. A major improvement is obtained when moving from PCA to LDML: LDML clearly outperforms PCA by more than 10 points over a large range of number of named faces.

For the graph-based method using Soft weights, as shown in Figure 3.19, we can first observe that PCA is comparable to the Euclidean distance for the graph-based approach. This is expected since PCA effectively tries to minimise the data reconstruction error. With only 20D, PCA performs slightly worse than L2, certainly because the reconstruction is coarse. With more dimensions, PCA slightly outperforms L2. PCA shows benefits in reducing noise in the data. Again, there is a great improvement in using LDML instead of PCA. Approximately 15 points in naming precision are gained when around 8000 faces are named by the algorithm. For LDML, the influence of the dimensionality is limited here. This corresponds to the observation of Figure 2.16 where we saw that above 20D, there was a plateau of performance for LDML.

We now show the same plot using the Log weights for the graph-based method in Figure 3.20. Again, PCA appears as a good approximation of L2 when enough dimensions are used. With 20D, performance drops significantly. Using LDML, we observe

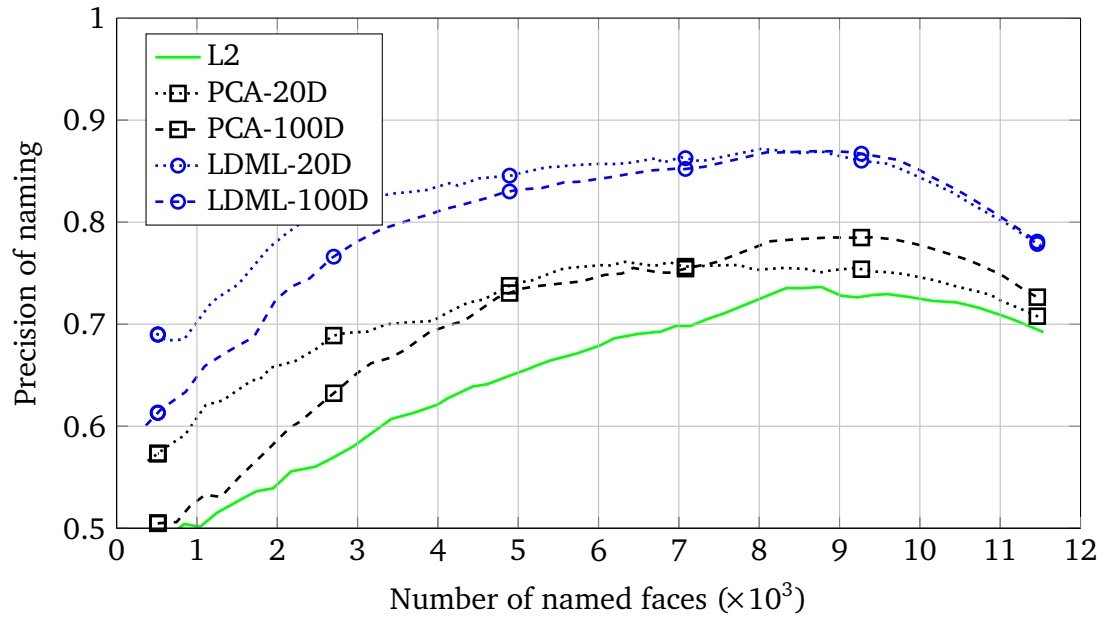


***Figure 3.18**: For the generative approach, we compare L2, PCA and LDML metrics, the latter two with 20D and 100D.*

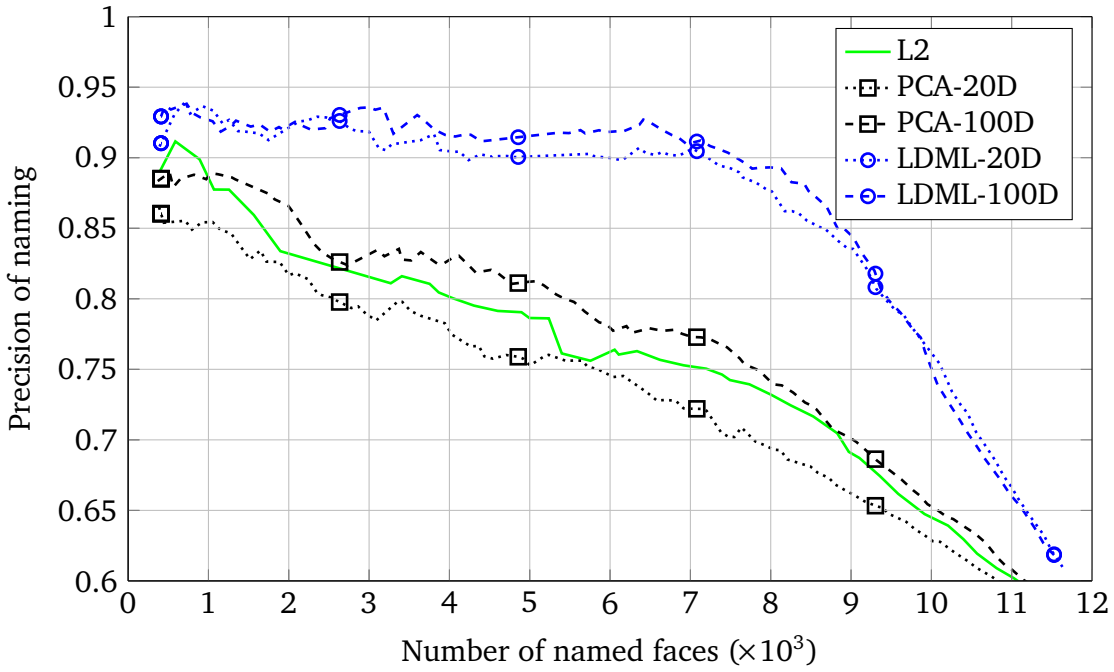


***Figure 3.19**: For the graph-based method using Soft weights, we compare L2, PCA and LDML metrics, the latter two with 20D and 100D.*

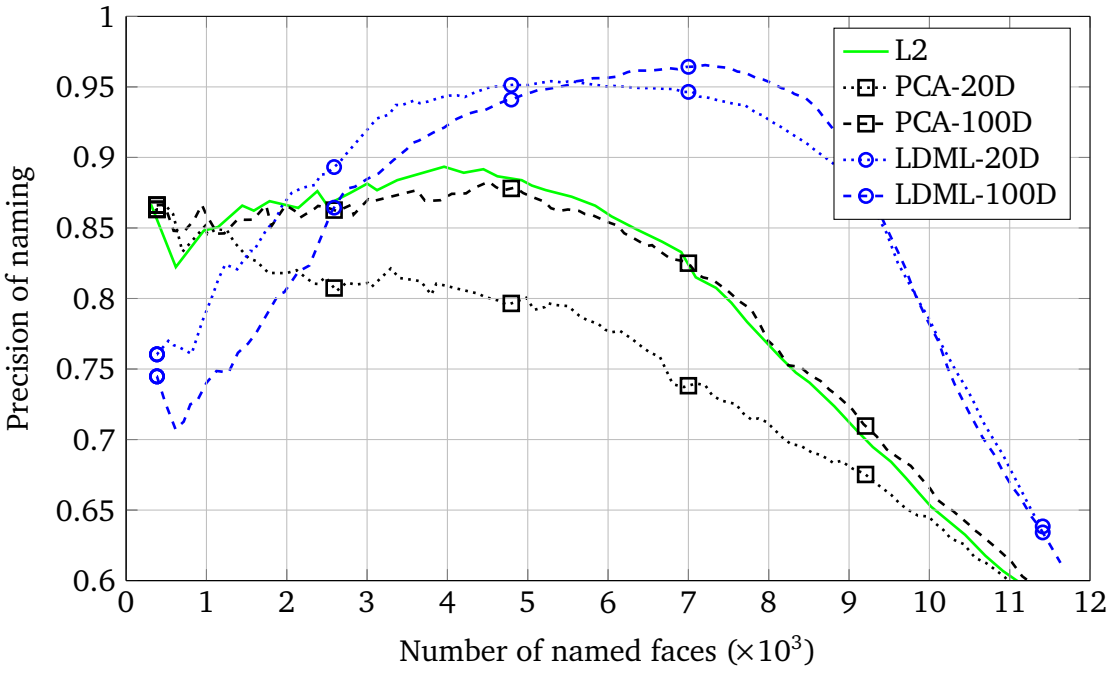


***Figure 3.20**: For the graph-based method using Log weights, we compare L2, PCA and LDML metrics, the latter two with 20D and 100D.*

a drop of performance when less than 2000 faces are named. This can be explained from two facts. First LDML metrics and PCA do not have similar scales. In our data set, distances between 20D data range from 0 to less than 10, while for LDML it is ten times more. Second, with a low θ, Log weights are simply a smooth version of Hinge weights when distance values are large. Therefore, the performance of Log weights for LDML have similar shape as the Hinge weights, while PCA is not affected. When larger numbers of faces are named, Log weights perform extremely well with LDML. The corresponding curve for LDML-20D shows a bump between 3500 and 7500 named faces with more than 94% of faces that are correctly named, and up to 96.5% for LDML-100D.

Finally, since we have observed that LDML-20D performs best for almost all settings and methods, we show in Figure 3.21 the performance for all approaches and weights using these projections. The conclusions are similar to the case of L2 distance: graph-based methods perform best when smaller amounts of faces are named, and the generative approach benefits from larger amounts of data. As we increase the number of named faces, Soft weights performs best at first, then Log weights have very good precision between 3000 and 9000 named faces, after which the generative method performs best.

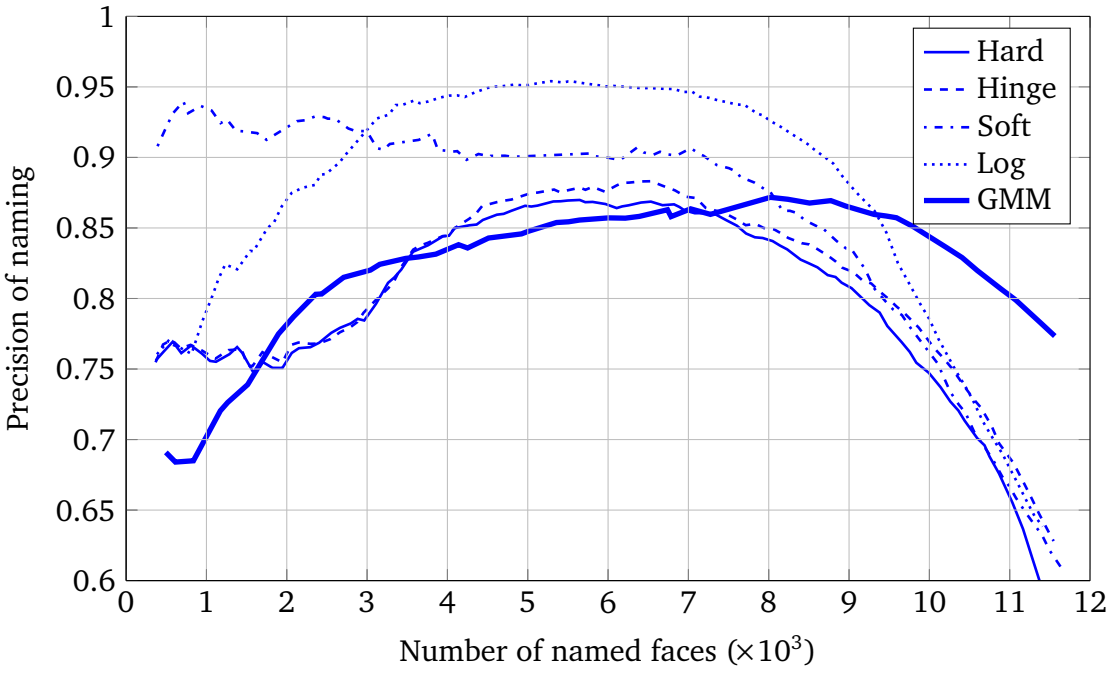


***Figure 3.21**: For the LDML-20D projection, we compare the different naming approaches.*

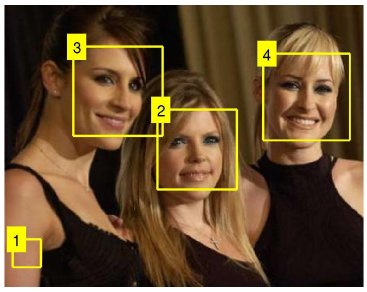
In Figure 3.22, we show qualitative results for the comparison between LDML-100D and PCA-100D for our graph-based naming procedure. These examples show how LDML helps detecting null-assignments and performs better than PCA for selecting the correct association between faces and names. The quantitative evaluation of the same comparison is provided in Table 3.2.

We conclude this first set of experiments by stressing that our graph-based method is very competitive compared to previously proposed Gaussian mixture models. Although our graph-based approaches are outperformed when most faces are named, they obtain the best performance when fewer faces are named, up to 9000. For all approaches, using a supervised metric instead of PCA improves face naming significantly.

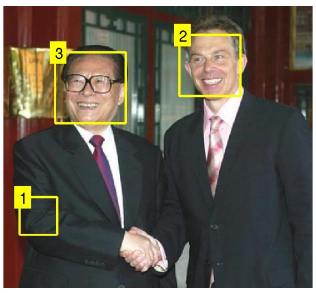
3.5.2 Metric learning from caption-based supervision

In this section we present our experimental setup to compare our different methods to learn metrics from weak supervision. Here, the metrics are evaluated for the task of face verification.

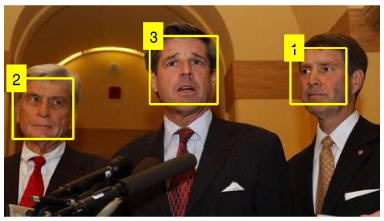
LDML	1. George W. Bush 2. **null** 3. **Tony Blair**
PCA	1. George W. Bush 2. Junichiro Koizumi 3. **Tony Blair**



LDML	1. **null** 2. **Natalie Maines** 3. **Emily Robison** 4. **Martie Maguire**
PCA	1. **null** 2. **Natalie Maines** 3. Martie Maguire 4. Emily Robison



LDML	1. **null** 2. **Tony Blair** 3. **Jiang Zemin**
PCA	1. David Kelly 2. **Tony Blair** 3. **Jiang Zemin**



LDML	1. Saddam Hussein 2. **John Warner** 3. **Paul Bremer**
PCA	1. **Bill Frist** 2. Paul Bremer 3. Saddam Hussein

***Figure 3.22**: Four document examples with their naming results for LDML-100D and PCA-100D when the maximum number of correctly associated names and faces is reached. The correct associations are indicated in bold. On these examples, the names that can be used for association with the faces are all shown: they were used by LDML or PCA, or both. Typically, LDML is better at detecting null-assignments and is more precise when associating a face to a name.*

	PCA-100D	LDML-100D
Graph-based		
Correct: name assigned	6585	7672
Correct: no name assigned	3485	4008
Incorrect: not assigned to name	1007	1215
Incorrect: wrong name assigned	3750	1932
Generative model		
Correct: name assigned	8327	8958
Correct: no name assigned	2600	2818
Incorrect: not assigned to name	765	504
Incorrect: wrong name assigned	3135	2547

***Table 3.2**: Summary of names and faces association performance obtained by the different methods when the maximum number of correctly associated names and faces is reached.*

Experimental Protocol

In the face verification task we have to classify a pair of faces as representing the same person or not. Using our *Labeled Yahoo! News* data set, and following the evaluation protocol of Huang et al. [2007b], we sampled 20000 face pairs of both sets A and B, approximately half of which are positives and half are negatives. Note that we cannot use the same test set as in *Labeled Faces in the Wild* because of overlap between test and train sets in this case. We measure average precision (AP) obtained at ten evenly spaced values of recall, and compute the mean AP (mAP) of the two following settings:

1. training the metric on set A and testing on set B's pairs, and
2. training the metric on set B to classify the pairs of set A.

Experimental Results

We study the performance of metric learning for different levels of supervision as described in Section 3.3.5 and Section 3.3.6. Since both optimise an objective function based on LDML, differences in performance will be due to the strategy to leverage the bag-level labels: either by selecting a single pair of instances for each pair of bag (MildML) or by inferring instance level labels (LDML).

The approach for LDML is similar to the one presented in Jin et al. [2009]. That work also infers instance level labels to perform metric learning in a MIL setting. However,

it is based on estimating prototypes, or cluster centres, for each of the classes. The objective then tries to ensure that for each bag and each class label of the bag, there is at least one instance of the bag close to one of the centres of the class. A second term in the objective function forces the centres of different classes to be maximally separated. The optimisation scheme is relatively complex, as in each iteration it involves minimising a non-convex cost function. Due to this complexity and the fact that we are mainly interested in comparing the different strategies to leverage the bag-level annotation, we do not include Jin et al. [2009] in our experimental evaluations.

As baseline methods we consider the L2 metric in the original space and after applying PCA to reduce the dimensionality. We compare the following settings for learning metrics:

(a) Instance-level manual annotations. This setting is only applicable to LDML, which should be an upper-bound on performance.

(b) Bag-level manual annotations. This setting is applicable directly to MildML, and indirectly to LDML, using instance-level annotations obtained by applying constrained clustering using the L2 metric to define the face similarities.

(c) Bag-level automatic annotations. Here, the data comes in a similar structure as setting (b) but the labels are noisy, since the names in the caption do not necessarily correspond to faces detected in the images.

In Figure 3.23, we report the performance of PCA, LDML and MildML for a wide range of dimensionalities and for the three settings (a), (b) and (c). As we increase the rank from $d = 4$ to $d = 128$, we can see that the different methods reach a plateau of performance. For LDML with the instance-level annotations (a), the plateau is attained approximately at $d = 32$, with a performance of 88.4% of mAP, which is substantially above L2 and PCA metrics (77.9%).

When learning from manual bag-level annotations (b), we can still learn effective metrics: MildML and LDML are still substantially better than the L2 and PCA metrics. Moreover, MildML matches the performance of the fully supervised LDML on the entire range of metric ranks, with at most 0.6% of improvement for $d = 4$ and 0.2% of decrease at $d = 8$. Notably, MildML outperforms the constrained clustering version of LDML using the same annotation (b), also over the full range of metric ranks, by around 2 points.

When using the fully automatic annotation (c), performance drops for both methods, which is understandable since the labels are now noisy. For $d \geq 16$, the performance is still better than L2 and PCA. Also in this setting MildML performs best, reaching 84.3% for 128 dimensions. This score is closer to the fully supervised LDML (89.0%) than to the Euclidean distance (77.8%) or PCA (77.9%) for the same rank. Still, there is a

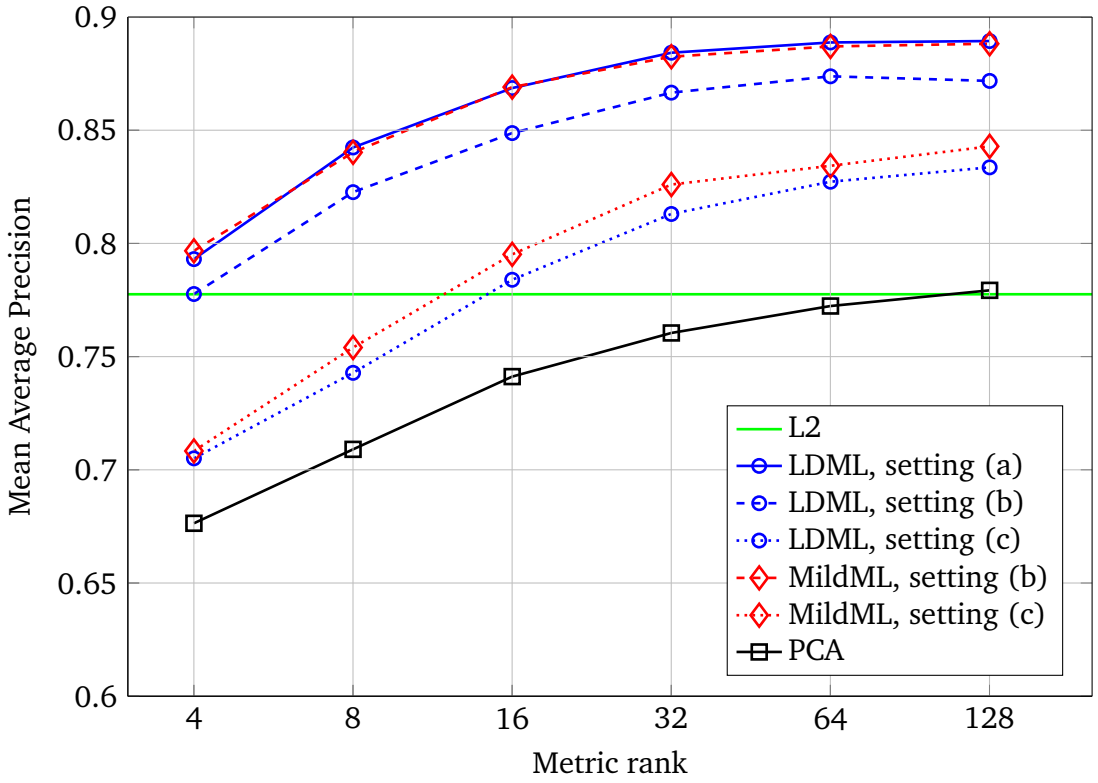


***Figure 3.23**: Mean average precision for L2, PCA, LDML and MildML for the three settings (a), (b) and (c) described in the text, when varying the metric rank.*

significant gap between the supervised learning and the learning from automatically generated labels, and it appears that this gap narrows from low to higher dimensions: from 8.8% at $d = 4$ to 4.5% at $d = 128$ between the two levels of supervision for MildML.

Finally, we also considered a variant of LDML which re-estimates the instance level labels using the current metric, and iterates until convergence. We refer to this variant as LDML*. As shown in Table 3.3, it has little influence on performance with the manual bag-level annotation of setting (b), at the cost of a much higher training time. On setting (c), the performance drops consistently by around 2%. We conclude that the noisy annotations penalise the clustering significantly. Remarkably, Jin et al. [2009] also relies on data clustering while MildML does not.

3.5.3 Naming with metrics using various levels of supervision

In our third set of experiments, we assess the quality of the learnt metrics for constrained clustering, including MildML learnt from bag-level supervision. Considering

Setting (b) Rank	4	8	16	32	64	128
LDML	77.8%	82.3%	84.9%	86.7%	87.4%	87.2%
LDML*	76.6%	82.4%	84.8%	86.5%	87.0%	87.0%

Setting (c) Rank	4	8	16	32	64	128
LDML	70.5%	74.3%	78.4%	81.3%	82.7%	83.4%
LDML*	68.1%	73.0%	76.9%	79.2%	80.8%	81.3%

***Table 3.3**: Comparison of mean average precision on the Labeled Yahoo! News data set for LDML and LDML* metrics. The two tables correspond to annotation settings (b) and (c), respectively. Please refer to the text for more details.*

the results in Section 3.5.1 showing that Soft weights perform very well for a large range of parameter, and considering that Soft weights are used in the optimisation of LDML from bag-level supervision (*c.f.* Equation 3.17), we restrict the following study to the graph-based approach using those weights.

Experimental Protocol

We use the clustering algorithm described in Algorithm 1 on one set of *Labeled Yahoo! News* after learning a metric on the other set. Note, the threshold b in Equation 3.17 is exactly the θ parameter of the Soft weights from Section 3.5.1 of the experiments. Therefore, we can measure the precision (*i.e.* the ratio of correctly named faces over total number of named faces) of the clustering procedure for various numbers of named faces by varying the value of b. As before, the curve is approximated on a reasonable range of named faces using a dichotomic search on the threshold value to obtain 50 approximately evenly spaced points.

Experimental Results

We study the performance of metric learning for different levels of supervision as described in Section 3.3.5 and Section 3.3.6, while varying the parameter of the clustering algorithm. As a baseline method we consider PCA with 128 dimensions, which performs comparably to the L2 metric. In addition to PCA, we compare the following two learnt metrics:

1. The fully supervised 128D LDML (which is comparable in performance to the 128D MildML learnt from manual bag-level supervision).

2. The 128D MildML learnt from automatically labelled bags of faces.

In Figure 3.24, we show the naming precision of those three metrics for the two sets: (a) for clustering faces of set A, and (b) for set B. First, we notice that the clustering algorithm which associates each instance with a label is efficient, and is able to name several thousand faces with a precision above 80%. Second, there is a large increase of performance using learnt metrics on both sets. LDML performs better than MildML, but the difference (of max. 6.0% between the two curves over the two sets) is smaller than the benefit of using MildML compared to PCA (up to +12.2%).

3.6 Conclusion

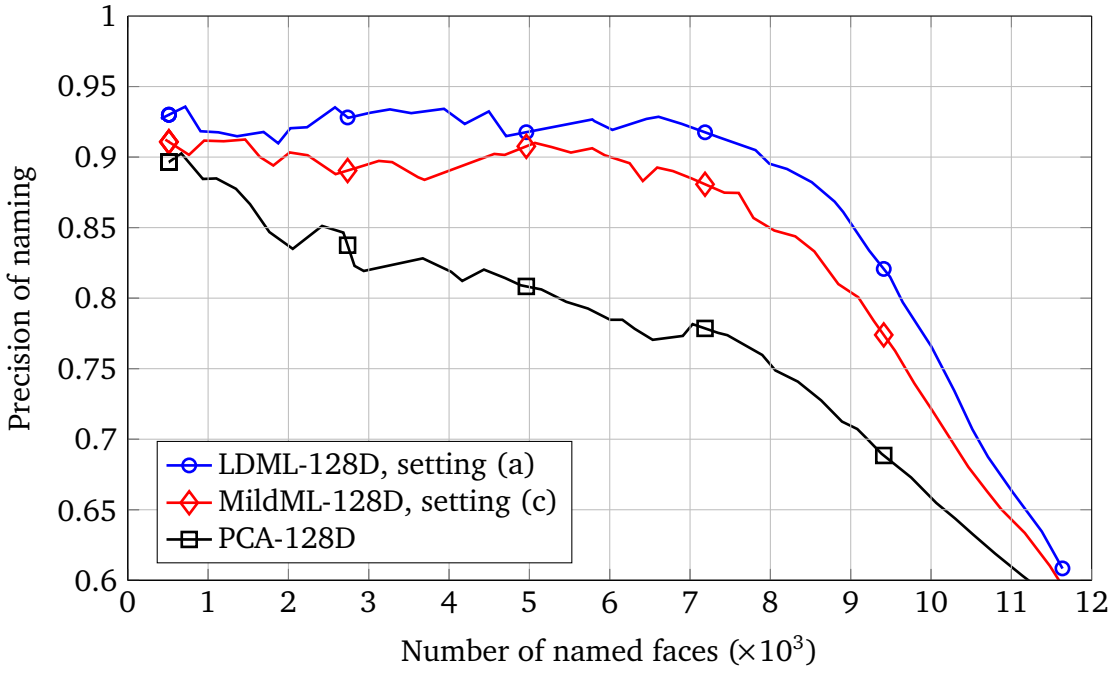
In this chapter, we have proposed a graph-based approach for automatically naming faces in images with names appearing in associated captions. In order to measure the performance, we have fully annotated a data set of around 20000 documents with more than 30000 faces. This data set is publicly available for fair and standardised future comparison with other approaches.

We have evaluated our graph-based approach using a collection of different possible definitions for the weights, and shown that Soft and Log weights achieve the best naming precision when associating up to two thirds of the faces. For larger numbers, the parameter estimation of the previously proposed generative Gaussian mixture model is robust enough to outperform the graph-based approach. A hybrid approach that would combine graph-based and GMM approaches is a direction worth exploring in the future.

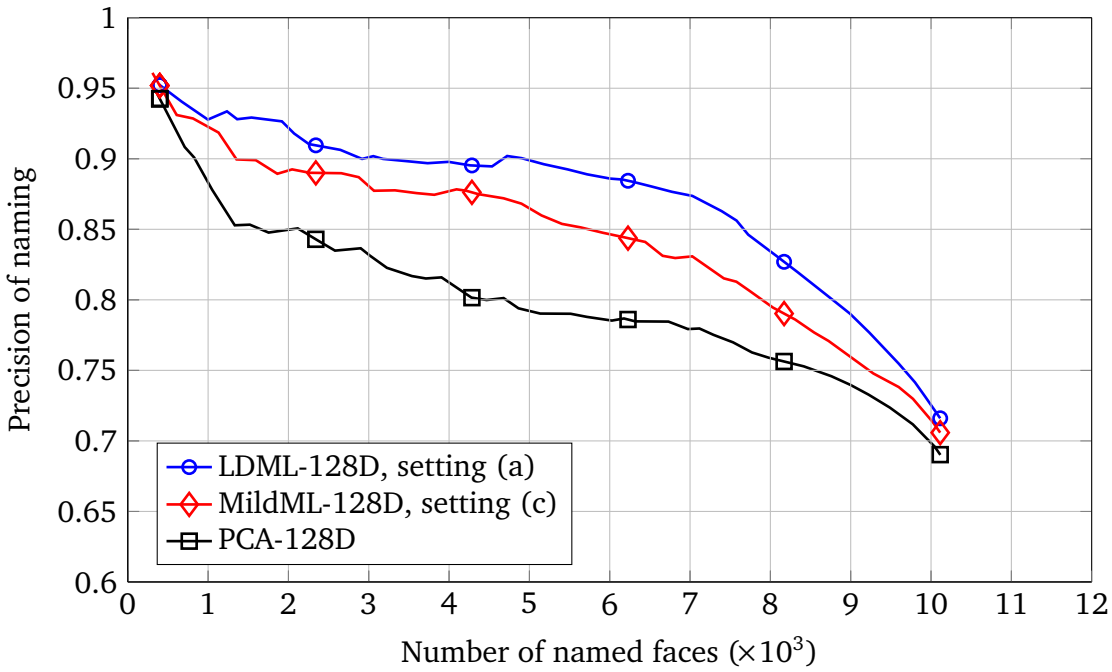
We have shown that metric learning improves both graph-based and generative approaches for face naming. We have attained precision levels above 90% with the graph-based approach, and around 87% for the generative approach, which is in both cases 6 points above the best score obtained using PCA. Since these maxima are attained for different numbers of named faces, the generative approach is in fact able to correctly name a larger number of faces, up to almost 9000 faces.

We have also proposed a Multiple Instance Learning (MIL) formulation of metric learning to allow metric learning from data coming in the form of labelled bags. We refer to it as MildML, for multiple instance logistic discriminant metric learning. We have also shown that it is possible to extend LDML, an instance-level metric learning method, to learn from the same labelled bags using constrained clustering.

On the *Labeled Yahoo! News* data set, we showed that our proposed MildML approach leads to the best results for face verification when using bag-level labels. When the bag-level labels are noise-free, the results are comparable to the case where instance



(a) Precision curve for clustering set A after learning metrics on set B.



(b) Precision curve for clustering set B after learning metrics on set A.

***Figure 3.24**: Precision of the clustering algorithm on set A (top) and B (bottom) for three metrics of rank $d = 128$ with the parameter varied, corresponding to a certain percentage of named faces. PCA is an unsupervised metric and performs worst, which confirms our first set of experiments. LDML is fully supervised at instance-level and performs best. MildML is learnt from automatically labelled bags and achieves performance close to the fully supervised metric.*

level labels are available. When using noisy bag labels, performance drops, but remains significantly better than that of the alternative methods. It appears that performing clustering to obtain instance-level labels and then learning LDML on the labelled examples does not perform well. The (costly) LDML* procedure that iterates metric learning and instance label assignment does not remedy this problem.

In conclusion, we have shown that effective metrics can be learnt from automatically generated bag-level labels, underlining the potential of weakly supervised methods. In future work we will consider learning algorithms that scale linearly with the number of data points, allowing learning from much larger data sets. Using larger data sets we expect the difference in performance between weakly supervised and fully supervised learning methods to diminish further.

4

Nearest neighbour tag propagation for image auto-annotation

Contents

4.1 Introduction

In this chapter, we consider a second type of multimodal data: images with keywords, as illustrated in Figure 4.1. Such user tags differ from manual annotations in that they have high ambiguity and unregularity, and the hypothetical corresponding visual concepts are unprincipled. There is an explosive growth of such data, with billions of tagged images already found on websites such as Flickr or Picasaweb. Here, we are interested in image auto-annotation and keyword-based image retrieval, for two tasks: to propose a list of most-relevant keywords to assist human annotation, and to perform keyword-based image search on a data set where not all images were tagged. Both tasks imply to predict the relevance of a given tag for a given image.

Image auto-annotation is an active subject of research (*e.g.*, see Grangier et al. [2006], Grangier and Bengio [2008], Mei et al. [2008], Li and Wang [2008], Li et al. [2009a], Liu et al. [2009b]). The goal is to develop methods that can predict for a new image the relevant keywords from an annotation vocabulary. Such methods are becoming more and more important given the growing collections of user-provided visual

arctic
den
fox
grass

iguana
lizard
marine
rocks

(a) Images and tags from the *Corel 5000* data set

box
brown
square
white

blue
cartoon
man
woman

(b) Images and tags from the *ESP Game* data set

glacier
mountain
people
tourist

landscape
lot
meadow
water

(c) Images and tags from the *IAPR TC-12* data set

***Figure 4.1**: Examples of images with their associated tags. In (a), (b) and (c), images are from the Corel 5000, ESP Game and IAPR TC-12 data sets, respectively. There is a large diversity between the data sets, more details on their peculiarities are given in Section 4.4. Note that the associated tags, which we consider as ground truth annotations, do not always contain all relevant tags (e.g. "water" is missing for the Corel 5000 image on the right), and sometimes contain tags for which one can argue whether they are relevant or not (e.g. "tourist" for the left image of IAPR TC-12).*

content, *e.g.* on photo or video sharing sites, and desktop photo management applications. These large-scale collections feed the demand for automatic retrieval and annotation methods. Since the amount of images with more or less structured annotations is also increasing, this allows the deployment of machine learning techniques to leverage this potential by estimating accurate tag prediction models.

Although the general problem is a difficult one, progress has been made in the research community by evaluations on standardised annotated data sets. In the next section we will detail the related work that is most closely linked to ours.

The main shortcomings of existing work are twofold. First, models are often estimated to maximise generative likelihood of image features and annotations, which might not be optimal for tag prediction. Second, many parametric models are not rich enough to accurately capture the intricate dependencies between image content and annotations. Non-parametric nearest neighbour approaches have been found to be quite successful for tag prediction, *e.g.* Feng et al. [2004], Jeon et al. [2003], Lavrenko et al. [2003], Makadia et al. [2008], Pan et al. [2004], Zhang et al. [2006], but also for many other computer vision problems such as image segmentation (*c.f.* Liu et al. [2009a]), scene completion (*c.f.* Hays and Efros [2007]) and object recognition (*c.f.* Malisiewicz and Efros [2009]).

This is mainly due to the high "capacity" of such models: they can adapt flexibly to the patterns in the data as more data is available. However, existing nearest neighbour type methods do not integrate the learning of the similarity that defines the nearest neighbours in order to maximise the predictive performance of the model. Either a fixed metric (Feng et al. [2004], Zhang et al. [2006]) or ad-hoc combinations of several metrics (Makadia et al. [2008]) are used, despite many recent work showing the benefits of metric learning for many computer vision tasks such as image classification (Jin et al. [2009]), image retrieval (Hertz et al. [2004]), or visual identification (Guillaumin et al. [2009b]).

In this chapter, we present TagProp, short for Tag Propagation, a new nearest neighbour type model that predicts tags by taking a weighted combination of the tag absence/presence among neighbours. This work has been published in Guillaumin et al. [2009a]. Our contributions are the following. First, the weights for neighbours are either determined based on the neighbour rank or its distance, and set automatically by maximising the likelihood of annotations in a set of training images. With rank-based weights the k-th neighbour always receives a fixed weight, whereas distance-based weights decay exponentially with the distance. Our tag prediction model is conceptually simple, yet outperforms the current state-of-the-art methods using the same feature set. Second, contrary to earlier work, our model allows the integration of metric learning. This enables us to optimise *e.g.* a Mahalanobis metric between image features – or, less costly, a combination of several distance measures – to define the neighbour weights for the tag prediction task. Third, TagProp includes word-

specific logistic discriminant models. These models use the tag predictions of the word-invariant models as inputs and are able, using just two parameters per word, to boost or suppress the tag presence probabilities for very frequent or rare words. This results in a significant increase in the number of words that are recalled, *i.e.* assigned to at least one test image.

To evaluate our models and to compare to previous work, we use three different data sets – *Corel 5000*, *ESP Game* and *IAPR TC-12*– and standard measures including precision, recall, mean average precision and break-even point. In Figure 4.1, we show example images with their tags from the three data sets. Although they show large diversity, they also share the common property that each image is associated with a limited number of tags, such that many relevant tags are not used. This observation is taken into account in our models by down-weighing tag absences when learning our models.

The rest of this chapter is organised as follows. In the next section we give an overview of the related work. Then, in Section 4.3, we present our tag prediction models, and how we estimate their parameters. In Section 4.4 we present the data sets, evaluation criteria as well as the image features we use in our experiments. The experimental results are presented in Section 4.5, where we also compare TagProp to the state-of-the-art. In Section 4.6 we present our conclusions and directions for further research.

4.2 Related work and state of the art

In this section we discuss models for image annotation and keyword-based retrieval most relevant for our work. We identify four main groups of methods: those based on topic models, mixture models, discriminatively trained models, and nearest neighbour type models.

4.2.1 Parametric topic models

The first group of methods are based on topic models such as probabilistic latent semantic analysis (PLSA, Hofmann [2001]) and latent Dirichlet allocation (LDA, Blei et al. [2003]), see *e.g.* Barnard et al. [2003], Blei and Jordan [2003], Monay and Gatica-Perez [2003, 2004].

These methods model annotated images as samples from a specific mix of topics, where each topic is a distribution over image features and annotation words. Parameter estimation involves estimating the topic mix for each image, and estimating the data distributions of the topics. Most often, a multinomial distribution over words is

used for the labels, and a Gaussian over visual features for a set of different image regions.

Methods inspired by machine translation (Duygulu et al. [2002]), in this case translating from discrete visual features to the annotation vocabulary, can also be understood as topic models, using one topic per visual descriptor type.

Below we present PLSA and the PLSA-based approach to image auto-annotation by Monay and Gatica-Perez [2004].

PLSA-based image auto-annotation

The PLSA model (Hofmann [2001]) is a generative model which assumes that the observed data x (either text data t or visual data v) in document d is conditioned on a latent variable z. Given this latent variable, the distribution of data occurrences x_i is independent of the document it occurs in, such that the joint probability of the observed variables is written the following way:

$$p(x_i, d_j) = p(d_j)p(x_i|d_j) \tag{4.1}$$

$$= p(d_j)\sum_{k=1}^{K} p(x_i, z_k|d_j) \tag{4.2}$$

$$= p(d_j)\sum_{k=1}^{K} p(x_i|z_k)p(z_k|d_j), \tag{4.3}$$

by marginalisation over the K latent aspects z_k.

The PLSA parameters, which are estimated using the EM algorithm, are divided into two sets. First, there are K multinomial distributions of size D (where D is the size of the data) for modelling $p(x|z_k)$. This can be understood as a $D \times K$ matrix $\mathbf{Y}$ which gives, for each aspect, its data distribution. Second, for each of the N documents, the mixture of the K topics have to be inferred. This can be represented in a $K \times N$ matrix $\mathbf{Z}$ containing $p(z_k|d_j)$. For unseen documents, PLSA requires to estimate new parameters $p(z|d)$, with $p(x|z)$ kept fixed from the training stage.

Alternatively, PLSA can be seen as a low-rank non-negative matrix factorisation technique (*c.f.* Gaussier and Goutte [2005]): the $V \times N$ matrix $\mathbf{X}$ of $p(x_i|d_j)$ is approximated as the product of $\mathbf{Y}$ and $\mathbf{Z}$ described above: $\mathbf{X} \approx \mathbf{YZ}$.

To model multimodal data such as images and words, one could apply PLSA to each modality separately. This would of course lead to two distinct latent spaces which would not take any profit from the co-occurrences of visual and textual cues. To achieve this goal, one must link the two PLSA models, for instance by sharing the same mixture of aspects for each document.

Monay and Gatica-Perez [2004] propose to learn the word-specific PLSA first, since the textual modality is on a much higher semantic level than the visual one. Once both $p(t|z)$ and $p(z|d)$ are learnt, they are kept fixed when inferring $p(v|z)$. Given a new image d' represented by its visual features v', it is possible to use the previously computed $p(v|z)$ to infer $p(z|d')$, and then compute $p(t|d')$ by marginalising over aspects:

$$p(t|d') = \sum_{k=1}^{K} p(t|z_k)p(z_k|d'). \tag{4.4}$$

Concerning the visual and textual features, the caption is represented as a vector of word occurrences using a fixed vocabulary. The image is also represented as a histogram vector, but built over lower-level features: RGB colour values are quantised according to a regular $6 \times 6 \times 6$ grid to yield counts over 216 bins.

4.2.2 Non-parametric mixture models

Although the models described above are conceptually attractive, their expressive power is limited by the number of topics. Typically several hundreds of topics are used, but the number of parameters grows linearly with the number of topics. Thus using more topics might cause over-fitting effects, and Bayesian parameter estimation or other forms of regularisation need to be used. Another option is to use non-parametric approaches like Hierarchical Dirichlet Processes (Teh et al. [2006]), as done by Yakhnenko and Honavar [2008].

A second family of methods uses mixture models to define a joint distribution over image features and annotation tags. To annotate a new image, these models compute the conditional probability over tags given the visual features by normalising the joint likelihood. Sometimes a fixed number of mixture components over visual features per keyword is used (Carneiro et al. [2007]), while other models use the training images as components to define a mixture model over visual features and tags (Feng et al. [2004], Jeon et al. [2003], Lavrenko et al. [2003]). Each training image defines a likelihood over visual features and tags by a smoothed distribution around the observed values. These models can be seen as non-parametric density estimators over the co-occurrence of images and annotations. For visual features Gaussians are used, while the distributions over annotations are multinomials, or separate Bernoullis for each word.

Below, we describe the Multiple Bernoulli Relevance Model (MBRM) of Feng et al. [2004] in more details.

Multiple Bernoulli Relevance Model

The Multiple Bernoulli Relevance model (MBRM) is a generative approach that models the joint probability of observing a set of R image regions $\mathbf{r} = \{r_1, \ldots, r_R\}$ and a set of K keywords $\mathbf{w} = \{w_1, \ldots, w_K\}$ as a mixture over the N images i of the training set:

$$p(\mathbf{r}, \mathbf{w}) = \sum_{i=1}^{N} p(i) \left(\prod_{j=1}^{R} p(r_j|i) \prod_{w \in \mathbf{w}} p(w|i) \prod_{w \notin \mathbf{w}} (1 - p(w|i)) \right). \tag{4.5}$$

This model involves three distinct distributions. First, the distribution of images $p(i)$, which is assumed uniform because of the lack of task-specific information. Second, the distribution $p(r|i)$ of the visual features of one region r given that image i was chosen by the generative process. This distribution is estimated as a Gaussian kernel-based density:

$$p(r|i) = \frac{1}{Z} \sum_{j=1}^{R_i} \exp\left(-\frac{1}{2} (r - r_j^i)^\top \mathbf{\Sigma}^{-1} (r - r_j^i) \right), \tag{4.6}$$

where $Z = R_i (2^D \pi^D \det \mathbf{\Sigma})^{\frac{1}{2}}$, D is the dimensionality of the region descriptor and $\mathbf{r}^i = \{r_1^i, \ldots, r_{R_i}^i\}$ is the set of visual features of image i. The covariance matrix $\mathbf{\Sigma}$ is typically assumed to be a scalar matrix $\lambda \mathbf{I}$, where $\mathbf{I}$ is the identity matrix and λ is empirically estimated from a held-out subset of the training data.

Third, $p(w|i)$ are independent Bernoulli distributions for each of the W possible keywords. This is an important feature of MBRM. It originates from the valid remark that tag prediction is about deciding on the relevance or irrelevance of a keyword. This differs from multinomial distributions over the entire vocabulary which model the prominence of keywords compared to others. Using a Beta prior for each word, which is the conjugate to Bernoulli, with a smoothing parameter μ obtained through cross-validation, the Bayes estimate of $p(w|i)$ is written the following way:

$$p(w|i) = \frac{\mu y_{iw} + N_w}{\mu + N}, \tag{4.7}$$

where $y_{iw} \in \{0, 1\}$ indicates the absence or presence of w in the list of keywords of image i, and N_w is the number of images annotated with w.

To annotate a new image $\mathbf{r}$, a search is performed for the most likely set of keywords given the visual features:

$$\mathbf{w}^\star = \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{w}|\mathbf{r}) = \operatorname*{argmax}_{\mathbf{w}} \frac{p(\mathbf{r}, \mathbf{w})}{p(\mathbf{r})} = \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{r}, \mathbf{w}), \tag{4.8}$$

which can be optimised efficiently.

For visual features, rectangular image regions are used instead of segmentation, which accelerates the processing time significantly, on which a set of 30 different simple features is extracted.

4.2.3 Discriminative methods

Both families of generative models discussed above may be criticised because the estimation of their parameters relies on maximising the generative data likelihood, which is not necessarily optimal for predictive performance. Therefore, discriminative models for tag prediction have also been proposed (Cusano et al. [2004], Grangier and Bengio [2008], Hertz et al. [2004], Li and Wang [2003], Chang et al. [2003], Mei et al. [2008], Gao et al. [2006], Vijayanarasimhan and Grauman [2008]). These methods learn a separate classifier for each tag, and use these to predict for each test image whether it belongs to the class of images that are annotated with each particular tag. Different learning methods have been used, including support vector machines, multiple-instance learning, and Bayes point machines.

Notable is the work of Grangier and Bengio [2008] which also addresses the problem of retrieving images based on multi-word queries. We describe their proposed Passive-Aggressive Model for Image Retrieval (PAMIR) below.

Passive-Aggressive Model for Image Retrieval

Contrary to most approaches, Grangier and Bengio [2008] formalise the problem of keyword-based image retrieval as a ranking problem, and not as an image auto-annotation task.

The system is based on the idea that for a text-based query q, the score function $F(q,\cdot)$ for the relevant images p^+ should be higher than for irrelevant images p^-:

$$\forall q, \forall p^+ \in R(q), \forall p^- \in \overline{R}(q), \quad F(q,p^+) - F(q,p^-) > 0. \tag{4.9}$$

where $R(q)$ and $\overline{R}(q)$ represent the sets of relevant and irrelevant images for query q, respectively. After learning the system's parameters, *i.e.* the parameters of F, the assumption is that this property is likely to hold also for unseen images and new queries.

To obtain the solution that best generalises to test data among all feasible solutions, the objective is designed to maximise the margin $m(F)$ of the score function as defined by:

$$m(F) = \min_{(q,p^+,p^-)} F(q,p^+) - F(q,p^-) > 0. \tag{4.10}$$

The function F is further restricted to a linear classification function of the following form:

$$F(q,p) = q^\top \mathbf{A} p, \tag{4.11}$$

where q represents the query as a vector of $\mathbb{R}^W$, p represents images in $\mathbb{R}^D$, and $\mathbf{A} \in \mathbb{R}^{W \times D}$ is a matrix that can interpreted either as projecting the images in the textual query space, or equivalently projecting the queries in the image space. F is linear with respect to the elements of A and therefore can be written $F(q,p) = \mathbf{a}^\top \phi(p,q)$, using weights $\mathbf{a}$ to combine the features $\phi(p,q)$. The vector $\phi(p,q)$ contains the components of $qp^\top$. Using the definition of the margin in Equation 4.10, Equation 4.9 can then be rewritten as:

$$\mathbf{a}^\top(\phi(q,p^+) - \phi(q,p^-)) \geq m(F). \tag{4.12}$$

Without loss of generality, we can assume $\|\mathbf{a}\| = 1$, such that maximising the margin under the ordering constraints of Equation 4.9 is equivalent to:

$$\underset{\mathbf{w}}{\text{minimise}}\ \|\mathbf{w}\|^2 \quad \text{subject to} \quad \forall (q,p^+,p^-),\ \mathbf{w}^\top(\phi(p,q^+) - \phi(q,p^-)) \geq 1, \tag{4.13}$$

by setting $\mathbf{w} = \frac{1}{m(F)}\mathbf{a}$.

In case of infeasibility, slack variables are introduced to relax the constraints. The objective is then written in a multivariate SVM fashion (*c.f.* Joachims [2005]) using a regularisation hyper-parameter C:

$$\underset{\mathbf{w}}{\text{minimise}}\ \|\mathbf{w}\|^2 + C \sum_{(q,p^+,p^-)} l(\mathbf{w}; q, p^+, p^-), \tag{4.14}$$

where $l(\mathbf{w}; q, p^+, p^-) = [1 - \mathbf{w}^\top \phi(q,p^+) + \mathbf{w}^\top \phi(q,p^-)]_+$ is the loss function and $[z]_+ = \max(0,z)$ is the Hinge function.

The optimisation is performed using an online passive-aggressive approach (*c.f.* Crammer et al. [2006]) to handle the large scale problems that arise from considering pairs of relevant-irrelevant images for every query. Additionally, the authors prove that the learnt weights $\mathbf{w}$ are a linear combination of training images, hence showing that PAMIR is kernelizable.

The bag-of-words framework is adopted for the query representation, as well as a bag-of-features descriptor for describing the image's regions of interest. Additional visual cues are concatenated: uniform local binary patterns and colour histograms.

4.2.4 Local approaches

Given the increasing amount of training data that is currently available, local learning techniques are becoming more attractive as a simple yet powerful alternative to parametric models. Examples of such techniques include methods based on label diffusion over a similarity graph of labelled and unlabelled images (Liu et al. [2009b], Pan et al. [2004]), or learning discriminative models in neighbourhoods of test images as in Zhang et al. [2006].

A simpler ad-hoc nearest neighbour tag transfer mechanism was recently introduced by Makadia et al. [2008], showing state-of-the-art performance, which we describe below.

Joint equal contribution ad-hoc nearest neighbours

Using a given similarity measure between images, Makadia et al. [2008] proposed the following simple ad-hoc transfer mechanism from nearest neighbours. From a query image i and a number n, the goal is to obtain n keywords from the neighbourhood of i:

1. Using query image i, find the K nearest neighbours $i_1, i_2, \ldots, i_K$.
2. Rank the keywords of i_1 according to their frequencies in the data set.
3. Transfer the n highest ranked keywords to i.
4. If i_1 did not have at least n keywords, but $n' < n$, then rank the keywords of $i_2, \ldots i_K$ according to the following factors:
   (a) co-occurrence in the training set with the keywords of i_1,
   (b) frequency in the neighbourhood of i,

   and transfer the $n - n'$ highest ranked keywords to i

The original paper is unclear about the order of the ranking (descending or ascending), but in our re-implementation we found taking the *most* frequent keywords of i_1 and the *least* frequent keywords of the remaining neighbourhood to work best. To our understanding, this is because it combines good precision of the first nearest neighbours with a higher recall obtained from the rest of the neighbourhood.

To define a visual similarity, Makadia et al. [2008] considers two options. The first is a normalised distance combination called *joint equal contribution* (JEC). Several distances computed from different visual features are normalised to have a maximal

value of 1. The scaling terms are computed exactly when the features are normalised in some way (*e.g.* they have unit norm) or can be estimated from the training data set. The rescaled distances are then averaged into a single distance measure.

The authors also considered combining the base distances by learning a binary classifier separating image pairs that have several tags in common from images that do not share any tags. This is performed using a linear classifier with L_1 penalty, such that feature selection is performed. This technique is also known as Lasso (*c.f.* Tibshirani [1996]). Importantly, though, this linear distance combination did not give better results than the JEC. We believe that this is due to the fact that, here, Lasso is not used to optimise a function that relates to the tag transfer mechanism.

4.3 Tag relevance prediction models

In this section, we now describe our proposed model for image auto-annotation and keyword-based image retrieval. These goals can be achieved if we are able to correctly predict the relevance of annotation tags for images. Indeed, given these relevance predictions we can annotate images by ranking the tags for a given image, or perform keyword-based retrieval by ranking images for a given tag or a combination of tags.

Our proposed method is based on a weighted nearest neighbour approach, inspired by the recent successful methods described above (*c.f.* Feng et al. [2004], Jeon et al. [2003], Lavrenko et al. [2003], Makadia et al. [2008]), that propagate the annotations of training images to new images. Our models are learnt in a discriminative manner, rather than using held-out data or using neighbours in an ad-hoc manner. We assume that some visual similarity or distance measures between images are given, abstracting away from their precise definition. In practice, these different distance measure will originate from comparing different feature sets for describing the images.

4.3.1 Nearest neighbour prediction model

To model image annotations, we use Bernoulli models for each keyword, following Feng et al. [2004]. This choice is natural because keywords, unlike natural text where word frequency is meaningful, are either present or absent. The dependencies between keywords in the training data are not explicitly modelled, but are implicitly exploited in our model.

We use $y_{iw} \in \{-1, +1\}$ to denote the absence/presence of keyword w for image i, hence encoding the image annotations. The tag presence prediction $p(y_{iw} = +1)$ for image i is a weighted sum over the training images, indexed by j:

$$p(y_{iw} = +1) = \sum_j \pi_{ij} p(y_{iw} = +1|j), \tag{4.15}$$

$$p(y_{iw} = +1|j) = \begin{cases} 1 - \epsilon & \text{for } y_{jw} = +1, \\ \epsilon & \text{otherwise,} \end{cases} \tag{4.16}$$

where π_{ij} denotes the weight of image j for predicting the tags of image i. To ensure proper probabilities, we require the following:

$$\forall i, j, \quad \pi_{ij} \geq 0 \tag{4.17}$$

$$\forall i, \quad \sum_j \pi_{ij} = 1. \tag{4.18}$$

We use ϵ in Equation 4.16 to avoid zero prediction probabilities, and in practice we set $\epsilon = 10^{-5}$. To estimate the parameters that control the weights π_{ij} we maximise the log-likelihood of the predictions of training annotations. Taking care to set the weight of training images to themselves to zero, *i.e.* $\pi_{ii} = 0$, our objective is to maximise:

$$\mathcal{L} = \sum_{i,w} c_{y_{iw}} \log p(y_{iw}), \tag{4.19}$$

where c_y is a cost that takes into account the imbalance between keyword presence and absence. Indeed, in practice, there are many more tag absences than presences, and absences are much noisier than presences. As already discussed, this is because most tags in annotations are relevant, but often the annotation does not include all relevant tags. We set:

$$c_{+1} = 1/n^+ \tag{4.20}$$

$$c_{-1} = 1/n^- \tag{4.21}$$

where n^+ is the total number of positive labels, and likewise n^- is the total number of negative labels.

In Figure 4.2, we illustrate the task of image auto-annotation as it is modelled using a weighted nearest neighbour approach. The probability of relevance of a tag for an image can be viewed as the weighted average of presence of this tag in the neighbourhood, or the expected prediction of a stochastic neighbour selection process, or again a mixture model using the training images as the mixture components (*c.f.* Equation 4.15). In the nearest neighbour graph, tags "propagate" with a probability that depends on the weights relating two neighbours.

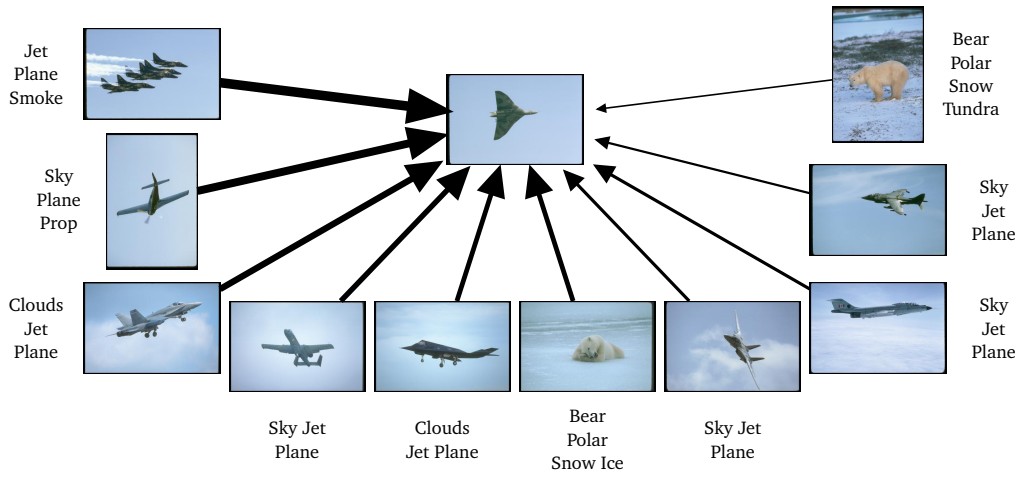


***Figure 4.2**: Weighted nearest neighbour tag prediction model in TagProp. The probability of relevance of a tag for an image can be viewed as the weighted average of presence of this tag in the image neighbourhood.*

4.3.2 Rank-based weights

Our first approach is to assign a given weight to all first nearest neighbours, then another one for all second nearest neighbours, etc. We call this model "rank-based weight". This can be formalised by setting:

$$\pi_{ij} = \gamma_k \tag{4.22}$$

if j is the k-th nearest neighbour of i. To limit the number of parameters, we can choose $K \in \mathbb{N}$, typically between 10 and 1000, and constrain γ_k to be zero for $k > K$. The likelihood of y_{iw} can therefore be written:

$$p(y_{iw}) = \sum_{k=1}^{K} \gamma_k p(y_{iw}|n_{ik}), \tag{4.23}$$

where n_{ik} denotes the index of the k-th neighbour of image i.

The constraints on π_{ij} directly translate into constraints for γ_k:

$$\forall k, \quad \gamma_k \geq 0, \tag{4.24}$$

$$\sum_{k=1}^{K} \gamma_k = 1, \tag{4.25}$$

and are obviously convex, since they express that $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_K]^\top$ lies on the $(K-1)$-simplex.

The data log-likelihood in Equation 4.19 is concave in the parameters $\boldsymbol{\gamma}$ and can therefore be estimated using a projected-gradient algorithm. The derivative of Equation 4.19 with respect to γ_k equals:

$$\frac{\partial \mathcal{L}}{\partial \gamma_k} = \sum_{i,w} \frac{c_{y_{iw}}}{p(y_{iw})} \frac{\partial p(y_{iw})}{\partial \gamma_k} \tag{4.26}$$

$$= \sum_{i,w} \frac{c_{y_{iw}} p(y_{iw}|n_{ik})}{p(y_{iw})}. \tag{4.27}$$

In practice, we use an equivalent but unconstrained formulation. Assuming strict positivity of the weights for the K nearest neighbours, we use the following soft-max definition of the weights:

$$\gamma_k = \frac{\exp(\delta_k)}{\sum_{k'=1}^{K} \exp(\delta_{k'})}. \tag{4.28}$$

The optimisation is performed on the vector $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_K]^\top$ directly. The gradient of the log-likelihood becomes:

$$\nabla \mathcal{L}(\boldsymbol{\delta}) = (\mathrm{diag}(\boldsymbol{\gamma}) - \boldsymbol{\gamma}\boldsymbol{\gamma}^\top) \nabla \mathcal{L}(\boldsymbol{\gamma}), \tag{4.29}$$

where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_K]^\top$.

With this parametrisation, the objective is not concave anymore but the optimisation is unconstrained. The soft-max formulation also highlights the relationship with the approach presented in the following section.

In both case (direct weight or soft-max), the number of parameters equals the neighbourhood size K (although, technically, there are only $K - 1$ free parameters). We refer to this variant as RK, for "rank-based".

Using a fixed distance, the K nearest neighbours can be pre-computed, and the model can be optimised using the $N \times K$ matrix containing indices of nearest neighbours and the $N \times W$ annotation matrix.

In order to make use of several different distance measures between images we can extend the model by introducing a weight for each combination of rank and distance measure. For each distance measure d we define a weight π_{ij}^d that is equal to γ_k^d if j is the k-th neighbour of i according to the d-th distance measure. Again we require all weights to be non-negative and to sum to unity: $\sum_{j,d} \pi_{ij}^d = 1$. The total weight for an image j is then given by the sum of weights $\pi_{ij} = \sum_d \pi_{ij}^d$ obtained using different distance measures. Similarly, the relative weight of a particular distance measure d is obtained by considering its contribution in the neighbourhood $\eta_d = \sum_k \gamma_k^d$. We refer to this variant as MR, for "multiple ranks".

4.3.3 Distance-based parametrisation for metric learning

The other possibility for defining the weights is to express the weights directly as a function of the distance, rather than the rank. This has the advantage that weights will depend smoothly on the distance, which is crucial if the distance is to be adjusted during training.

The weights of training images j for an image i are therefore redefined as

$$\pi_{ij} = \frac{\exp(-d_\theta(i,j))}{\sum_{j'} \exp(-d_\theta(i,j'))}, \tag{4.30}$$

where d_θ is a distance metric with parameters θ that we want to optimise. Note that the weights π_{ij} decay exponentially with distance d_θ to image i.

Choices for d_θ include – and are not limited to – the following:

(a) A fixed distance d with a positive scale factor: $d_\lambda(i,j) = \lambda d_{ij}$,

(b) A positive combination of several base distances: $d_{\mathbf{w}}(i,j) = \mathbf{w}^\top \mathbf{d}_{ij}$, where $\mathbf{d}_{ij}$ is a vector of base distances between image i and j, and $\mathbf{w}$ contains the positive coefficients of the linear distance combination,

(c) A Mahalanobis distance $d_{\mathbf{M}}$ parametrised by a semi-definite matrix $\mathbf{M}$, as defined in Equation 2.1:

$$d_{\mathbf{M}}(i,j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j), \tag{4.31}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are feature vectors for images i and j, respectively.

The Mahalanobis distance is the most general case of the three options above: if the base distances are Euclidean distances, then the single distance case corresponds to $\mathbf{M} = \lambda \mathbf{I}$, and the case of multiple base distances can be written using a block-scalar matrix $\mathbf{M}$ using the components of $\mathbf{w}$ on the diagonal.

Let us first focus on the simple case (a). In this setting, there is only one parameter λ, that controls the decay of the weights with the distance. The gradient of $\mathcal{L}$ with respect to λ equals:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{iw} \frac{c_{y_{iw}}}{p(y_{iw})} \sum_j \left(d_{ij} \pi_{ij} p(y_{iw}) - d_{ij} \pi_{ij} p(y_{iw} = +1 | j) \right), \tag{4.32}$$

$$= \sum_{ij} C_i \left(\pi_{ij} - \rho_{ij} \right) d_{ij}, \tag{4.33}$$

where $C_i = \sum_w c_{y_{iw}}$ and ρ_{ij} denotes the weighted average over all words w of the posterior probability of neighbour j for image i given the annotation:

$$\rho_{ij} = \sum_w \frac{c_{y_{iw}}}{C_i} \frac{\pi_{ij} p(y_{iw}|j)}{p(y_{iw})} = \sum_w \frac{c_{y_{iw}}}{C_i} p(j|y_{iw}). \tag{4.34}$$

We refer to this variant as DT, for "distance-based".

In the second option (b), the number of parameters equals the number of base distances that are combined. This is a straightforward extension to DT, and the gradient is written by replacing the d_{ij} term in Equation 4.33 with the vector $\mathbf{d}_{ij}$ of base distances:

$$\nabla \mathcal{L}(\mathbf{w}) = \sum_{i,j} C_i(\pi_{ij} - \rho_{ij})\mathbf{d}_{ij}. \tag{4.35}$$

We refer to this variant as MD, for "multiple distances". Equivalently, DT is the special case of MD when there is only one base distance.

Finally, using option (c), we can decompose $d_{\mathbf{M}}(i,j)$ as in Equation 4.36 (and already given in Equation 2.2). This shows that the distance can be written as a linear combination of the components of $(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$, which can be seen as individual distance values. Therefore we can re-use Equation 4.35 to obtain the gradient of the likelihood in Equation 4.37:

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{D} \sum_{l=1}^{D} \mathbf{M}_{kl}(\mathbf{x}_{ik} - \mathbf{x}_{jk})(\mathbf{x}_{il} - \mathbf{x}_{jl}) \tag{4.36}$$

$$\nabla \mathcal{L}(\mathbf{M}) = \sum_{i,j} C_i(\pi_{ij} - \rho_{ij})(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top. \tag{4.37}$$

Using similar derivations than for LDML in Equation 2.31 to Equation 2.36, we can write the gradient in the following form:

$$\nabla \mathcal{L}(\mathbf{M}) = \mathbf{X}(\mathbf{H} + \mathbf{H}^\top)\mathbf{X}^\top, \tag{4.38}$$

where

$$\mathbf{X} = [\mathbf{x}_i] \in \mathbb{R}^{D \times N}, \tag{4.39}$$

$$h_{ii} = \sum_j C_i(\rho_{ij} - \pi_{ij}), \tag{4.40}$$

$$h_{ij} = C_i(\pi_{ij} - \rho_{ij}) \qquad \text{for } i \neq j, \tag{4.41}$$

$$\mathbf{H} = [h_{ij}] \in \mathbb{R}^{N \times N}. \tag{4.42}$$

Again, as shown for LDML in Section 2.3.1, we can replace the optimisation of $\mathbf{M}$ with the optimisation of a projection matrix $\mathbf{U}$ such that $\mathbf{M} = \mathbf{U}^\top \mathbf{U}$.

$$\nabla \mathcal{L}(\mathbf{U}) = 2\mathbf{U}\mathbf{X}(\mathbf{H} + \mathbf{H}^\top)\mathbf{X}^\top. \tag{4.43}$$

This expression is promising for learning compact image representations for optimal tag prediction using the Euclidean distance in our weighted nearest neighbour model, using a low-rank matrix $\mathbf{U} \in \mathbb{R}^{d \times D}$. Notably, most indexing and approximate nearest neighbour techniques have been proposed for Euclidean spaces. Equation 4.43 also shows that, similarly to LDML and many other Mahalanobis metric learning algorithms, TagProp for learning a Mahalanobis distance is kernelisable.

In the following, we focus on the DT and MD variants. The first reason is that these variants have only a limited number of parameters, one for each base distance. In our experiments, we have $M = 15$ image representations which account for a total of $D = 32752$ dimensions. The number of parameters would therefore be very large ($\sim 5 \cdot 10^8$) in the case of general Mahalanobis distance. Since our model is a non-linear nearest neighbour classifier, it is already very flexible in its predictions with only a few parameters. Second, computations can be performed very efficiently in the case of MD, by pre-computing the M pairwise distance matrices.

To further reduce the computational cost of training the model, we do not compute all pairwise π_{ij} and ρ_{ij}. Rather, for each i we compute them only over a large set of K neighbours, and assume the remaining π_{ij} and ρ_{ij} to be zero. When only one distance is used, we simply select the K nearest neighbours, since they will not change with the scaling parameter.

When learning a linear combination of several distances it is not clear beforehand which will be the nearest neighbours, as the distance measure changes during learning. Given that we will use K neighbours, we therefore include as many neighbours as possible from each base distance so as to maximise the chance to include all images with large π_{ij} regardless of the distance combination $\mathbf{w}$ that is learnt. After determining these neighbourhoods, our algorithm scales linearly with the number of training images.

For each image, we select the K neighbours the following way. Let us denote with $\mathbf{N}_k^d$ the neighbourhood of size k for distance d. Since the neighbourhoods of size k for the different distances are not disjoint, the union of those will typically have less than $M \times k$ elements. We therefore try to find k such that the union of the M neighbourhoods of size k has cardinality K:

$$k^\star = \operatorname*{argmin}_k \left\{ \left| \bigcup_d \mathbf{N}_k^d \right| \geq K \right\}. \tag{4.44}$$

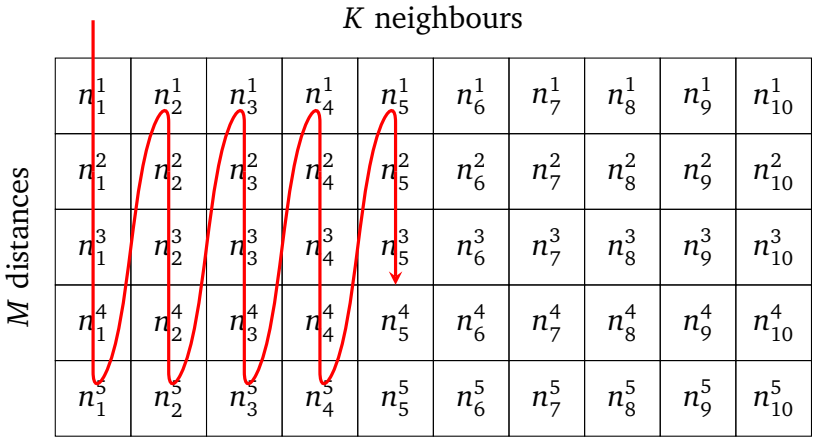


***Figure 4.3**: Illustration of the $M \times K$ nearest neighbour matrix of an image, containing at cell (d,k) the index n_k^d of the k-th nearest neighbour for distance d. To select K neighbours for our tag prediction algorithm, we propose to select the K first unique values found in this matrix when running through it column-wise.*

Alternatively, $k^\star$ can be understood as the maximum $\overline{k} = \min\{k_d\}$, where k_d is the largest neighbour rank for which neighbours 1 to k of base distance d are included among the selected neighbours.

For a given data point and precomputed neighbours for each base distance, there is an efficient algorithm to perform such a selection in linear time. The basic idea is to go through the neighbourhoods, rank by rank, while keeping track of which neighbours are already selected, until K unique neighbours are found. For that, it is sufficient to pre-compute the K neighbours of each of the M distances in a $M \times K$ matrix and to go through it column-wise, as illustrated in Figure 4.3. In practice we use a variation of this idea that also keeps track of the lowest rank (among the distances) at which neighbours are found, such that it is easy to process distances one after the other in an online fashion, with the same overall complexity.

Finally, note the relation of our model to the multi-class metric learning approach of Globerson and Roweis [2005]. In that work, a metric is learnt such that weights π_{ij} as defined by Equation 4.30 are as close as possible in the sense of Kullback-Leibler (KL) divergence to fixed set of target weights ρ_{ij}. The target weights were defined to be zero for pairs from different classes, and set to a constant for all pairs from the same class. In fact, when deriving an EM-algorithm for our model, we find the objective of the M-step to be of the form of a KL divergence between the ρ_{ij} (fixed to values computed in the E-step) and the π_{ij}. For fixed ρ_{ij} this KL divergence is convex in $\mathbf{w}$.

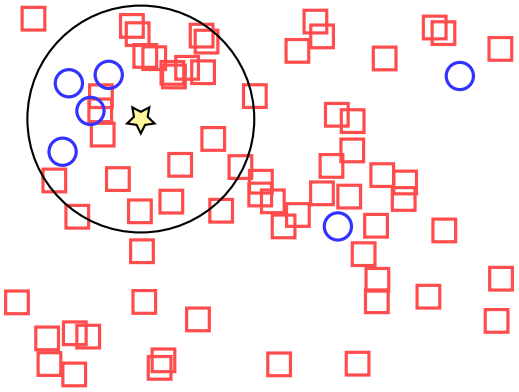


***Figure 4.4**: Illustration of the potential weakness in weighted nearest neighbour models. The star represents the image that is to be annotated, and is surrounded by images tagged with a red square and others tagged by a blue circle. Even if most of the blue circles are included in the neighbourhood (represented as a circle centred on the star) of the target image, its prediction will always be lower than the one for the red squares, since they are densely present in the entire space.*

4.3.4 Sigmoidal modulation of predictions

The weighted nearest neighbour models for tag prediction described above have many advantages. First, the number of parameters is very limited. Second, the parameter estimation is shared among all the keywords. Therefore, the estimated parameters are likely to be very effective also for very rare words that have, say, less occurrences than the number of parameters, M.

However, weighted nearest neighbour approaches tend to have relatively low recall scores, which is easily understood as follows. In order to receive a high probability for the presence of a tag, it needs to be present among most neighbours with a significant weight. This, however, is unlikely to be the case for the rare tags. So, even if we are lucky enough to have a few neighbours annotated with the tag – and this little number might actually account for the majority of the data associated with this tag – we will predict the presence with a low probability. This effect is illustrated in Figure 4.4.

To overcome this, we introduce word-specific logistic discriminant models that can boost the probability for rare tags and decrease it for very frequent ones. The logistic model uses weighted neighbour predictions by defining:

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w), \tag{4.45}$$

$$x_{iw} = \sum_j \pi_{ij} y_{jw}, \tag{4.46}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$. The weighted average of annotations x_{iw} for tag w among the neighbours of i is equivalent to Equation 4.15 up to an affine transformation. This affine transformation is ignored since the logistic model includes a linear transformation of the input. The word-specific models add 2 parameters to estimate for each word, namely (α_w, β_w). The resulting modulated variants are referred to as σRK, σMR, σDT and σMD respectively for RK, MR, DT and MD.

For fixed π_{ij} the model is a logistic discriminant model, and the log-likelihood is concave in $\{\alpha_w, \beta_w\}$, and can be trained for each keyword independently. Using the new model, the gradient of the log-likelihood of the training annotations with respect to the parameters θ that control the weights equals:

$$\nabla \mathcal{L}(\theta) = \sum_{i,w} c_{y_{iw}} \alpha_w p(-y_{iw}) y_{iw} \frac{\partial x_{iw}}{\partial \theta}, \tag{4.47}$$

and for the model based on rank or distances respectively

$$\frac{\partial x_{iw}}{\partial \gamma_k} = y_{n_{ik}w}, \tag{4.48}$$

$$\frac{\partial x_{iw}}{\partial \mathbf{w}} = \sum_j \pi_{ij} \left(x_{iw} - y_{jw} \right) \mathbf{d}_{ij}. \tag{4.49}$$

In practice we estimate the parameters θ and $\{\alpha_w, \beta_w\}$ in an alternating fashion. We observe rapid convergence, typically after alternating the maximisation less than ten times.

4.4 Data sets and features

To evaluate our models, we consider three publicly available data sets that have been used in previous work, and allow for direct comparison for the tasks of image auto-annotation and keyword-based image retrieval. In this section, we first describe each of them, providing basic statistics and some examples images. Then, in Section 4.4.4, we detail the feature sets that we extracted from the images, and the base distances we used for each feature.

4.4.1 *Corel 5000*

This data set was introduced and first used in Duygulu et al. [2002] and is available online[1]. Since then, it has become an important benchmark for keyword based image

[1] At: `http://kobus.ca/research/data/eccv_2002/`

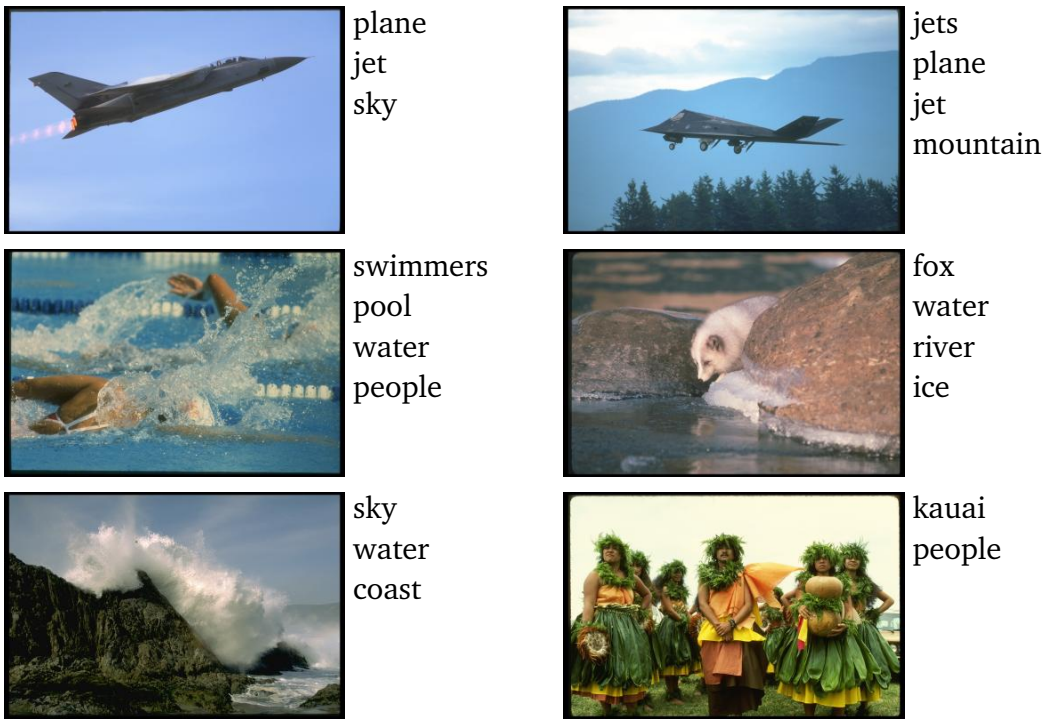


***Figure 4.5**: Examples of images from the Corel 5000 data set, with their associated keywords.*

retrieval and image annotation. It is a subset of around 5000 images of the larger Corel CD set, which is a large collection of manually annotated images. The images, all taken by professional photographers, are have a standard size of 256×384 pixels.

Originally, the collection comes with a high-level structure which indicated roughly the scene type, such as “africa”, “museum”, “insects”, “dogs”, “orchids”, “people”, etc., that can be used for image categorisation. Rather, the *Corel 5000* data set focuses on the manual annotations that have been given to the images in the form of keywords. These keywords describe to some extent the content of the images, especially when objects are present in the images. The annotations consist of one to five keywords, and were assigned for indexing purposes.

The data set is split in training and testing subsets. The former contains 4500 different images, and the later 499. In the training set, there are 371 different keywords, but only 260 also appear in the test set. It is therefore natural to restrict to these 260 words in our vocabulary.

Two example images have already been shown in Figure 4.1, but in Figure 4.5 we give several additional examples. The examples show that the variety in the data is limited, and that the annotations do not cover the visual content comprehensively.

4.4.2 *ESP Game*

The *ESP Game* data set is a recently built data set by Von Ahn and Dabbish [2004]. The images were collected on the Internet, are therefore show a very large diversity. Among the images, we can find logos, drawings, personal photos, web page decorations, etc., sometimes with a very low quality. The data set is therefore challenging.

Moreover, the annotations were obtained from an online game. In this game, two players, that can not communicate outside the game, gain points by agreeing on words when being simultaneously showed the same image. As a consequence, almost all words that are agreed upon describe the image to some extent. Notably, players will easily agree on words that are written in the images, a problem which is beyond the scope of this paper. Using the same image multiple times for several pairs of players help identify the important and meaningful tags for the images. Contrary to the *Corel 5000* data set, these annotations are therefore not specifically intended for indexing or retrieval.

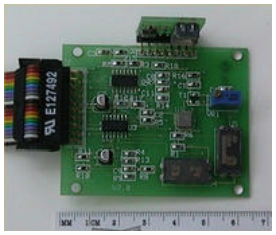
There are 60000 images that are publicly available[2], but, to ensure a fair comparison, we will use the subset of around 20000 images that was used in Makadia et al. [2008], with a vocabulary of 268 words. Similarly, examples have already been provided in Figure 4.1, but we add a few in Figure 4.6.

4.4.3 *IAPR TC-12*

This set of 20.000 images accompanied with descriptions in several languages was initially published for cross-lingual retrieval (Grubinger [2007]). It can be transformed into a format comparable to the other sets by automatically extracting common nouns using natural language processing techniques similar to the ones we already used in Chapter 3 for extracting names of individuals. Given the length of the text, this procedure typically yields a large number of keywords, but they are less descriptive and more noisy. The noun extraction procedure also provides the number of occurrences of each noun in the description, although we do not exploit this additional information. Similar to the *ESP Game* data set, these annotations that we use as ground-truth were not specifically associated with images for the tasks we consider. We use the same resulting annotation as in Makadia et al. [2008], which are publicly available[3], using a vocabulary of 291 words.

The images are touristic photos from South America, and are of a good quality. In this respect, they do not show such a wide variety as what is observed in the *ESP Game* data set, as shown in Figure 4.1 and Figure 4.7.

[2] `http://hunch.net/~learning/`
[3] `http://www.cis.upenn.edu/~makadia/annotation/`



computer
green
board

man
people
red
group
table
window

man
face
film
magazine
cover
river

blue
women
snow

circle
blue
box
war

coin
old
money
round
green

***Figure 4.6**: Examples of images from the ESP Game data set, with their associated keywords.*

In Table 4.1, we summarise different statistics for the three data sets, showing the average and maximum numbers of images assigned with the same keyword, and numbers of keywords associated to images. Below we describe our feature extraction procedure.

4.4.4 Feature extraction

We extract different types of features commonly used for image search and categorisation. They cover all four combinations of global or local features with colour or texture descriptors.

The two types of global image descriptors that we used are Gist features (Oliva and Torralba [2001]) and colour histograms with 16 bins in each colour channel for RGB, LAB, HSV representations. Our local features include SIFT (Lowe [2004]) as well as a robust hue descriptor (van de Weijer and Schmid [2006]), both extracted densely on a multi-scale grid or for Harris-Laplacian interest points. Each local feature descriptor is quantised using k-means with 1000 bins for SIFT and 100 bins for Hue on two million

cloud
mountain
slope
sky

lake
desert
landscape
middle
tourist

lake
mountain
shore
man
woman

woman
wall
house
roof
mountain
city
range

building
street
sky
car
people

sky
door
man
car
desert
roof

***Figure 4.7**: Examples of images from the IAPR TC-12 data set, with their associated keywords.*

	Corel 5000	*ESP Game*	*IAPR TC-12*
Image size	256 × 384	variable	360 × 480
Vocabulary size	260	268	291
Number of training images	4500	18689	17665
Number of test images	499	2081	1962
Average number of words per image	3.4	4.7	5.7
Maximum number of words per image	5	15	23
Average number of images per word	58.6	362.7	347.7
Maximum number of images per word	1004	4553	4999

***Table 4.1**: Statistics of the training sets of the three data sets. Average and maximum image and word counts are also provided. These statistics for the test sets are very similar to those of the training sets, except for the number of images and images per word.*

randomly-sampled descriptors from the training set. Images are then represented as a 'bag-of-words' histogram.

All descriptors but Gist are L1-normalised and also computed in a spatial arrangement (Lazebnik et al. [2006]). We compute the histograms over three horizontal regions of the image, as it roughly corresponds to standard landscape photography, and concatenate them to form a new global descriptor, albeit one that encodes some information of the spatial layout of the image. To limit colour histogram sizes, here, we reduced the quantisation to 12 bins in each channel. Note that this spatial binning differs from segmented image regions, as used in some previous work.

This results in 15 distinct descriptors, namely one Gist descriptor, 6 colour histograms and 8 bag-of-features (2 features types $\times$ 2 descriptors $\times$ 2 layouts). To compute the distances from the descriptors we follow previous work and use L2 as the base metric for Gist, L1 for global colour histograms, and χ^2 for the others.

4.5 Experiments

In this section, we present our experiments for image auto-annotation and keyword-based retrieval. First, in Section 4.5.1, we present the evaluation measures that we consider. Then, in Section 4.5.2, we compare our different base distances and their combination for the tag prediction task, using rank-based or distance-based weights for TagProp. In Section 4.5.3, we study the influence of the sigmoidal modulation and metric learning on performance. Results for complex queries are given in Section 4.5.4. Finally, in Section 4.5.5, we provide qualitative results for TagProp.

4.5.1 Evaluation measures

We evaluate our models with standard performance measures, used in previous work, that evaluate retrieval performance per keyword, and then average over keywords.

Precision and recall for fixed annotation length

Following Duygulu et al. [2002], each image is annotated with the 5 most relevant keywords. Then, the mean precision **P** and recall **R** over keywords are computed.

N+ is used to denote the number of keywords with non-zero recall value. This is the number of words that are effectively used at test time. Note that each image is forced to be annotated with 5 keywords, even if the image has fewer or more keywords in the ground truth. Therefore, even if a model predicts all ground-truth keywords with

a significantly higher probability than other keywords, we will not measure perfect precision and recall.

Precision at different levels of recall

We also evaluate precision at different levels of recall as in Grangier and Bengio [2008]. For each keyword we rank images according to their relevance, compute the precision at different levels of recall and obtain a keyword-specific value. The mean of these values is taken as performance measure.

The break-even point (**BEP**), or R-precision, measures for each keyword w the precision among the top n_w relevant images, where n_w is the number of images annotated with this keyword in the ground truth. The mean average precision (**MAP**) over keywords is found by computing for each keyword the average of the precisions measured after each relevant image is retrieved. These values measure the performance of image retrieval systems.

They can be easily adapted to measure the performance of image auto-annotation methods. This is done by inverting the roles of images and tags. The image-wise break-even point (**iBEP**) measures for each image the precision among the top n_i relevant tags, where n_i is the number of tags of this image in the ground-truth. The mean average precision (**iMAP**) over images is computed accordingly.

4.5.2 Influence of base distance and weight definition

We first study the influence of image features and base distances for the performance of TagProp. For this, we learn TagProp for ranked-based and distance-based weights, using each individual distance separately. In this case, the neighbourhoods do not change as the parameters are estimated, and the distance-based variant has only one parameter. We also combine the 15 distances in the ad-hoc manner of Makadia et al. [2008] as described in Section 4.2.4. As a short reminder, the distances are normalised in scale such that the maximum value is 1, and the 15 normalised distance values are then averaged. We refer to this combined but fixed distance as JEC.

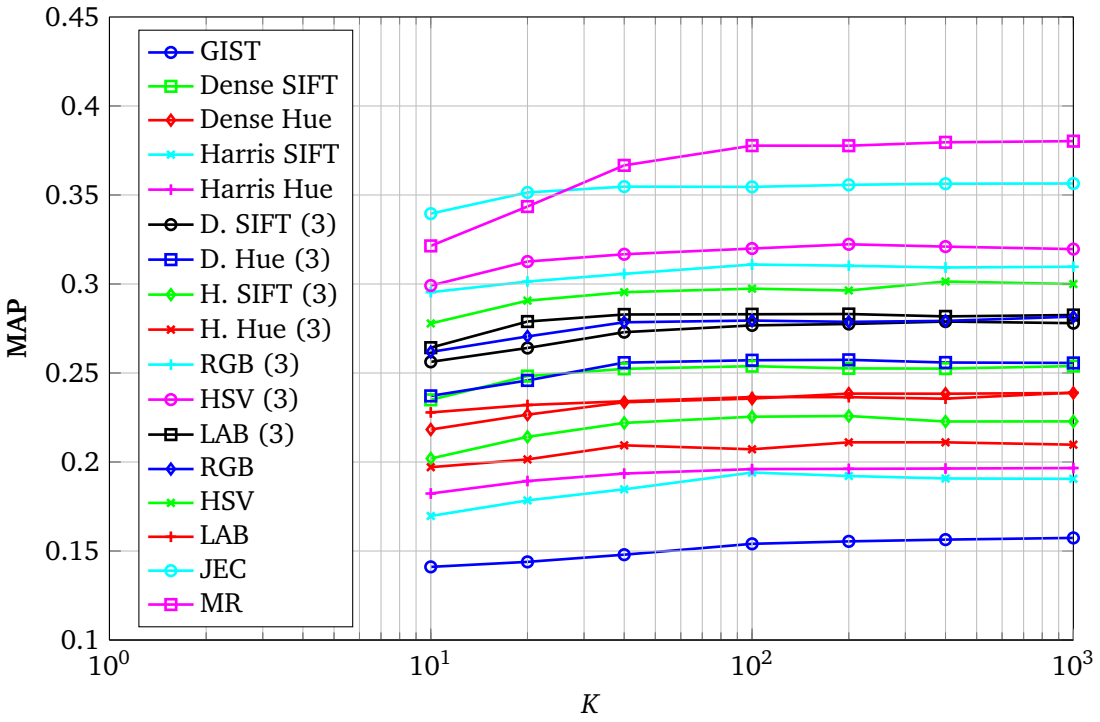
Finally, we also consider combining these 15 distances directly using TagProp. These are the MR (for rank-based weights) and MD (for distance-based weights) variants presented in Section 4.3. To determine the K neighbours for the MD variant, we use not only the 15 base distances but also the JEC combination. In this way, we are more likely to include all the images with high π_{ij} for any combination of the base distances. In total, below, we compare the 16 different fixed distances and the learnt combination.

In Figure 4.8, we show the mean average precision (**MAP**), for each of these 17 options for the *Corel 5000* data set, as a function of the neighbourhood size K. Our first observation is that individual base distances have very different performance. For *Corel 5000*, the colour descriptors typically outperform the texture-based ones, and the densely sampled interest points outperform the Harris detector. Using the spatial grid (referred to as "(3)" because it consists of 3 horizontal regions) consistently improves over the global quantisation. Colour histogram perform best here: the RGB colour space outperforms Lab, and the best individual descriptor to use is the HSV(3) colour histogram with the spatial layout. Its **MAP** is around 30% compared to around 15% for GIST. As we can see from the two plots (rank-based and distance-based, respectively), the neighbourhood size does not have a major influence when only one distance is used.

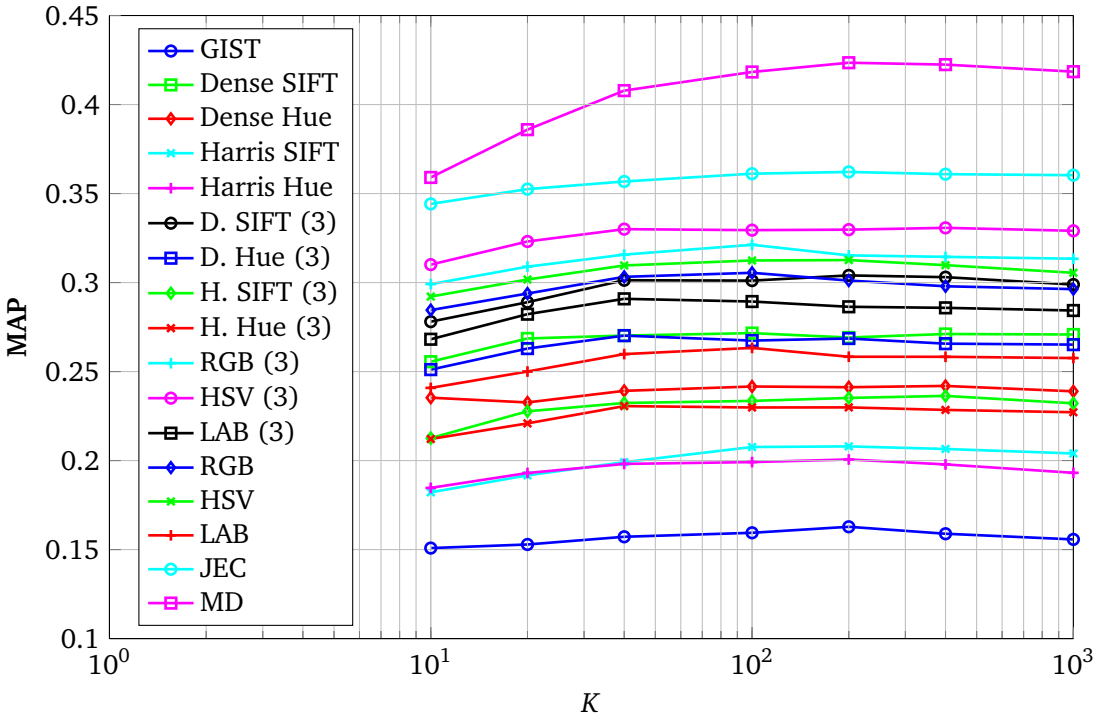
When considering combined distances, we observe, still in Figure 4.8, that even a fixed distance combination brings a significant improvement over using any of the individual distances. For the rank-based weights, there is a consistent improvement of 5 points of **MAP** over HSV(3). Next, when learning the combination, an additional improvement is obtained. For both MR and MD variants, a plateau of performance is attained at around 200 neighbours, which means that in this case the neighbourhood size does have an influence on the performance. For the multiple-rank variant, this can be easily explained. Since we use a fixed K to fairly compare methods that have the same number of parameters, the MR method uses effectively neighbourhoods of size K/M from each distance. When K is small and since $M = 15$, there are effectively very few nearest neighbours of each distance, which impacts the performance. For the distance-based case, the performance is always high above that of JEC: the maximum **MAP** of 42.4% is attained by MD compared to 36.2% for JEC, both for $K = 200$.

For the *ESP Game* and *IAPR TC-12* data sets, the same conclusions hold, except that the ordering of the individual descriptors changes. Only the *Corel 5000* data set seems to prefer colour descriptors, while the two other prefer Dense-SIFT(3) best. The advantage of using TagProp to learn the distance combination appears clearly: TagProp effectively performs feature selection.

To illustrate this fact, we show in Figure 4.9 the partial sums of rank-based weights that originate from each distance, and the learnt combination of distances in the distance-based weights (they are given relative to the JEC combination in order to suppress the scales of the distances in the comparison). As we see, the learnt combinations for both MR and MD are really dataset-specific, such that there is no descriptor that can be ignored beforehand. Using distance-based weights, the combinations are typically sparser. For instance, for the *ESP Game* data set, only 6 out the 15 distances have nonzero weights, with GIST being given the highest relative importance.



(a) Rank-based TagProp on *Corel 5000*



(b) Distance-based TagProp on *Corel 5000*

***Figure 4.8**: Performance of TagProp as measured using* **MAP** *as a function of the neighbourhood size K, for the 17 different distances described in the text. In the plot on the top are shown results for rank-based weights, while distance-based weights are used in the bottom plot.*

In the following, we will report results only using combined distances. Specifically, in the RK and DT variants, the JEC combination is used as it outperforms all individual distances.

4.5.3 Sigmoidal modulations

We now compare the different variants of TagProp, including the ones with sigmoid modulations, and compare them to some baselines and other published results. The closest related work is that of Makadia et al. [2008], that we refer to as as ahNN for "ad-hoc nearest neighbour". In their work, they introduce the JEC distance combination for a tag transfer mechanism. We report their published results (ahNN) and also use their method with our own features, *i.e.* using our own JEC combination, as a first baseline (ahNN*).

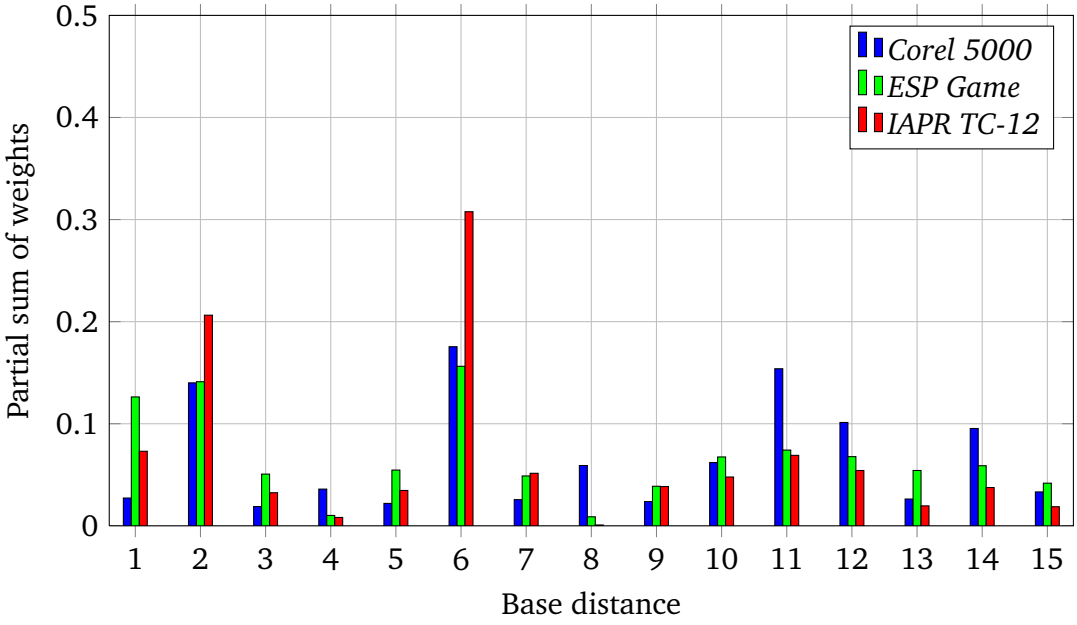
Our second baseline is called σSVM. It consists in using the JEC distance combination in RBF kernel-based SVMs. The SVMs are learnt for each word independently, and sigmoidal modulations are then fitted to obtain prediction probabilities.

In Table 4.2 we show the results for the *Corel 5000* data set. We can make several observations. First, using the tag transfer method proposed in Makadia et al. [2008] with our own features (ahNN*) we obtain results very similar to the original work (ahNN). Thus, other performance differences obtained using our methods must be due to the tag prediction method.

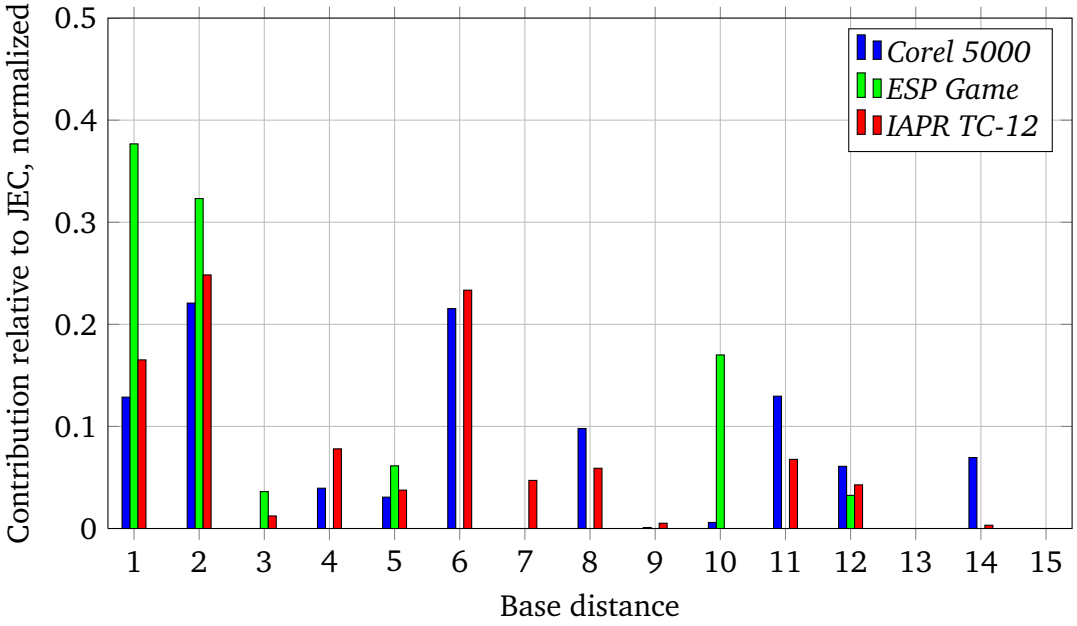
Second, the σSVM baseline obtains relatively good precision and rather poor recall. This observation is specific to the *Corel 5000* data set, and is related to the distribution of words in the data set. Indeed, about half of the words appear only in less than five images. Learning word-specific SVMs with less than five positive images is likely to yield poor classifiers.

Third, our models that use this fixed distance combination to define weights (either directly in DT or using ranks in RK) perform comparably to ahNN*. Among these results, the ones of the sigmoidal model using distance-based weights (σDT) are the best, and they show a modest improvement over the results obtained with the more ad-hoc ahNN*.

More importantly, using our models that combine ranks from different distances (MR and σMR) or integrate metric learning (MD and σMD), much larger improvements are obtained, in particular using the σMD variant. Compared to the current state-of-the-art method using the same features, we obtain marked improvements of 5% in precision, 9% in recall, and count almost 20 more words with positive recall. This result shows clearly that nearest neighbour tag prediction can benefit from metric learning. Interestingly, earlier efforts to exploit metric learning did not succeed (*c.f.* Maka-



(a) Partial sums of rank-based weights for each distance



(b) Relative contribution of each distance for distance-based weights

***Figure 4.9**: In the top are shown, for each data set, the partial sums of rank-based weights grouped by originating distance in the MR variant. Below, the combination of distances as learnt by the MD variant is shown, relative to the JEC combination. In this way, the weights do not account for the scales of the distances. The 15 distances are given in the same order as in the legend of Figure 4.8.*

	P	**R**	**N+**	Reference
Previously published results				
CRM	16	19	107	Lavrenko et al. [2003]
InfNet	17	24	112	Metzler and Manmatha [2004]
NPDE	18	21	114	Yavlinsky et al. [2005]
SML	23	29	137	Carneiro et al. [2007]
MBRM	24	25	122	Feng et al. [2004]
TGLM	25	29	131	Liu et al. [2009b]
ahNN	27	32	139	Makadia et al. [2008]
Baselines (with our features)				
ahNN*	**28**	**33**	**140**	
σSVM	23	17	79	
TagProp				
RK	27	32	136	Rank-based
σRK	24	34	139	
MR	30	32	131	
σMR	30	38	145	
DT	29	31	131	Distance-based
σDT	27	35	145	
MD	**34**	38	146	
σMD	33	**42**	**158**	

***Table 4.2**: Overview of performance on the Corel 5000 data set in terms of* **P**, **R**, *and* **N+** *of our models (using $K = 200$), our baselines and those reported in a selection of earlier work. The ahNN* baseline refers to our implementation of Makadia et al. [2008] using our 15 distances. σSVM uses a JEC-based RBF kernel, and fits sigmoids on the SVMs output. We show results for our variants: RK and DT using the JEC equal distance combination, MR which combines ranks of several distances and MD which integrates metric learning, as well as their modulated extensions (σRK, σDT, σMR and σMD, respectively).*

		ESP Game			*IAPR TC-12*		
		P	**R**	**N+**	**P**	**R**	**N+**
MBRM		18	19	209	24	23	223
ahNN		22	25	224	28	29	250
ahNN*		24	19	222	29	19	211
σSVM		44	24	228	48	25	227
TagProp	DT	**48**	17	203	48	21	219
	σDT	39	24	230	42	30	257
	MD	**48**	19	210	**49**	25	226
	σMD	39	**26**	**236**	46	**34**	**262**

***Table 4.3**: Comparison of performance in terms of* **P**, **R**, *and* **N+** *on the ESP Game and IAPR TC-12 data sets of the state-of-the-art methods reported by Makadia et al. [2008] to our TagProp variants with distance based weights, using $K = 200$ neighbours.*

dia et al. [2008] and Section 4.2). The key to our successful use of metric learning is its integration in the prediction model.

In Table 4.3, we show the results for the *ESP Game* and *IAPR TC-12* data sets. Note that less methods have been reported on these data sets in the literature. We restrict ourselves to the distance-based variants, which performs best. As a first remark, we notice that there is an unexplained discrepancy between the recall levels of ahNN reported by Makadia et al. [2008] and those that we obtained with our implementation and features (ahNN*). Second, the σSVM baseline performs very well here, and outperforms ahNN approaches in precision by 20%. As already mentioned, this is because keywords in these data sets have, on average, much more positive data to train on, such that SVMs are used at their full power. Notably, contrary to ahNN, TagProp is still able to match the performance σSVM in precision when not using the σ modulation (+4% on *ESP Game*, +1% on *IAPR TC-12*), or outperform it in recall when using the modulation (+2% on *ESP Game*, +9% on *IAPR TC-12*).

In Figure 4.10, we show the precision and the recall values on the three data sets when varying the neighbourhood size. We compare distance-based variants with and without the sigmoidal modulation. From these results we conclude that consistently over all number of neighbours, with or without the σ modulation, the metric learning finds distance combinations that outperform the equally weighted combination. Furthermore, it confirms that the σ modulation has a major impact on the **P** and **R** measure. While the recall is increased, the precision decreases. Therefore, choosing either variant is a trade-off between precision and recall.

When the neighbourhood increases, the performance increases significantly for the MD variants. This can be explained by the fact that, in MD, the ranking of neighbour images change with the learnt metric. Thus, in order to ensure that all useful training

images are included in the initial neighbourhoods (computed from the base distances, *c.f.* Section 4.3), these sets should be large enough.

In Figure 4.11 we further analyse which words benefit most from the improved recall in σ variants. For this, we use our distance-based variants MD, with $K = 200$ neighbours. As expected, the improvement is higher for rare words and only the few most frequent words are penalised in terms of recall, but even in this situation the performance remains high and many relevant images are still retrieved.

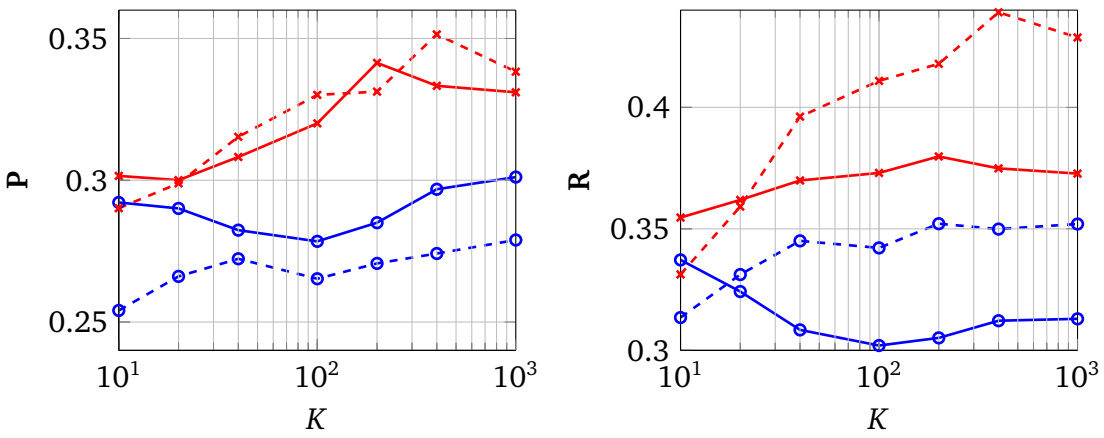
On the other measures, the impact of the sigmoidal modulation is less important, with a slight advantage to the model without it. Since the improvement is higher for recall than the drop of performance in precision, we report performance for variants including the σ modulation in Figure 4.12 using the other measures (**BEP**, **MAP**, **iBEP** and **iMAP**). These results confirm that there is a great benefit in optimising the distance combination within TagProp.

4.5.4 Image retrieval from multi-word queries

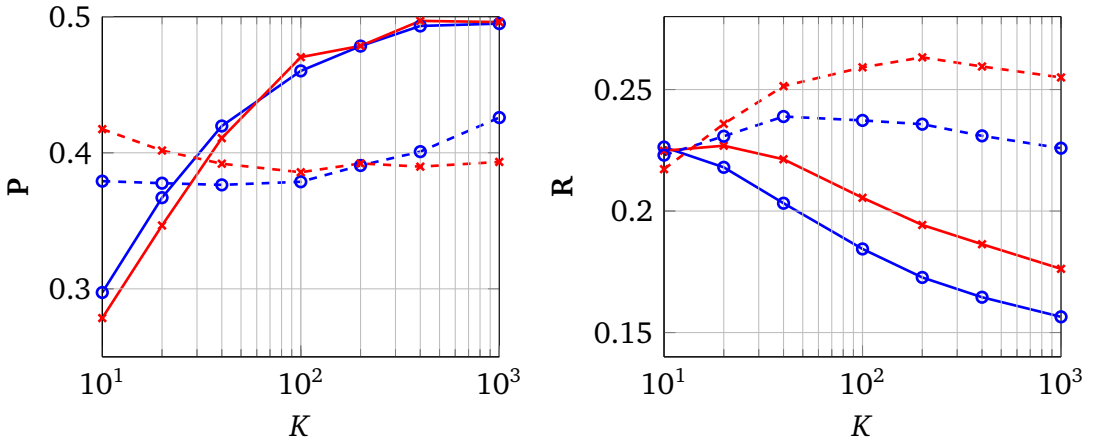
Up to this point, we have focused on image retrieval performance for single word queries. Most existing work also concentrate on this problem, as it is difficult as such. However, any realistic image retrieval system should support multi-word queries as well. Here we present performance in terms of **BEP** and **MAP** on the Corel data set that include multi-word queries.

To allow for direct comparison, we follow the setup of Grangier and Bengio [2008], which uses a subset of 179 words of the 260 annotation words of the *Corel 5000* data set that appear at least twice in the test set. Images are considered relevant for a query when they are annotated with all words, and we consider all 2241 queries composed of one or more words such that the test set includes at least one image that is relevant. Further, the queries are divided into "difficult" ones (1820) for which there are only one or two relevant images, and "easy" ones (421) with three or more relevant images.

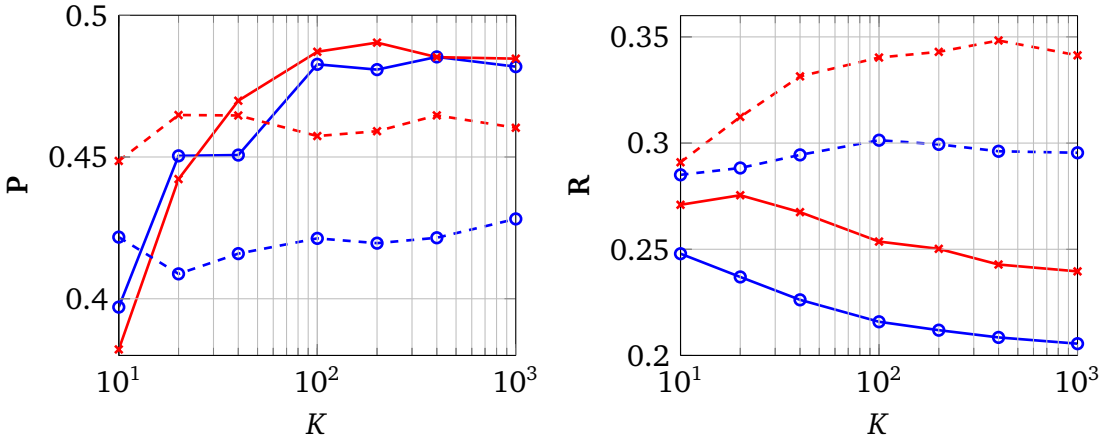
To predict relevance of images for a multi-word query we compute the probability according to our model to observe all keywords in the query. Due to the probabilistic output of TagProp, this is easily done by taking the product over the single keyword probabilities, as our model does not explicitly account for dependencies between words. In Table 4.4 we summarise our results, and compare to those of PAMIR (Grangier and Bengio [2008]). Our results improve by about 10% the **MAP** performance over all query types. Also in terms of **BEP** we gain 10% compared to PAMIR, which was found by Grangier and Bengio [2008] to outperform a number of alternative approaches.



(a) Precision and recall curves for the *Corel 5000* data set

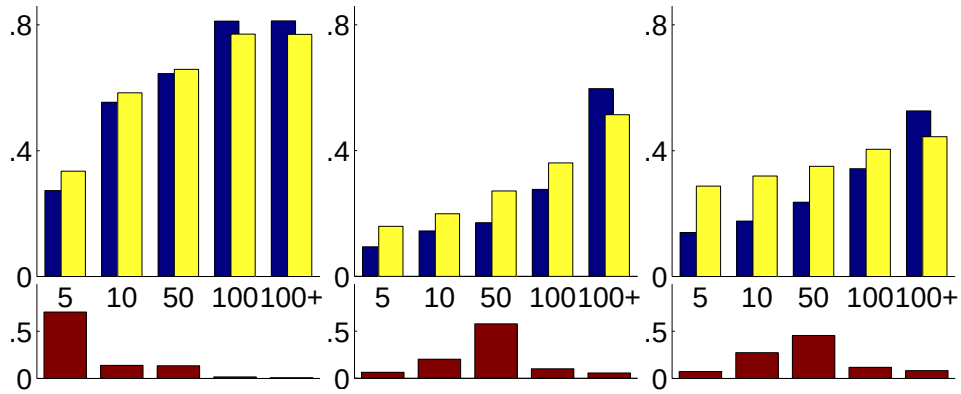


(b) Precision and recall curves for the *ESP Game* data set



(c) Precision and recall curves for the *IAPR TC-12* data set

Figure 4.10: *In these precision* **P** *and recall* **R** *curves plotted as a function of the neighbourhood size K, blue curves with circle markers correspond to the DT variant, while the red curves with crosses correspond to the MD variant. The solid curves are without the sigmoid modulation while the dashed ones include it.*



(a) *Corel 5000* (b) *ESP Game* (c) *IAPR TC-12*

***Figure 4.11**: Mean recall of words in the three data sets (from left to right: Corel 5000, ESP Game, and IAPR TC-12) for MD (blue) and σMD (yellow), grouped with respect to their frequency in the data set. Keywords are binned based on how many images they occur in: the first bin groups words with less then 5 relevant images, the second bin words with between 6 and 10 images, and so on. The lower bars show the fraction of keywords in each bin, the upper bars show the average recall for the words in a bin. The Corel 5000 data set has many keywords with only a few relevant images. The figures are reported for $K = 200$.*

		MAP	**MAP** (Single)	**MAP** (Multi)	**MAP** (Easy)	**MAP** (Difficult)	**BEP**
PAMIR		26	34	26	43	22	17
TagProp	DT	32	40	31	49	28	24
	σDT	31	41	30	49	27	23
	MD	**36**	43	**35**	53	**32**	**27**
	σMD	**36**	**46**	**35**	**55**	**32**	**27**

***Table 4.4**: Comparison of TagProp variants (using $K = 200$) and PAMIR (Grangier and Bengio [2008]) in terms of* **MAP** *and* **BEP**. *The* **MAP** *performance is also broken down over single-word and multi-word queries, easy and difficult ones. Only the 179 words that appear at least twice in test images are used, as in Grangier and Bengio [2008].*

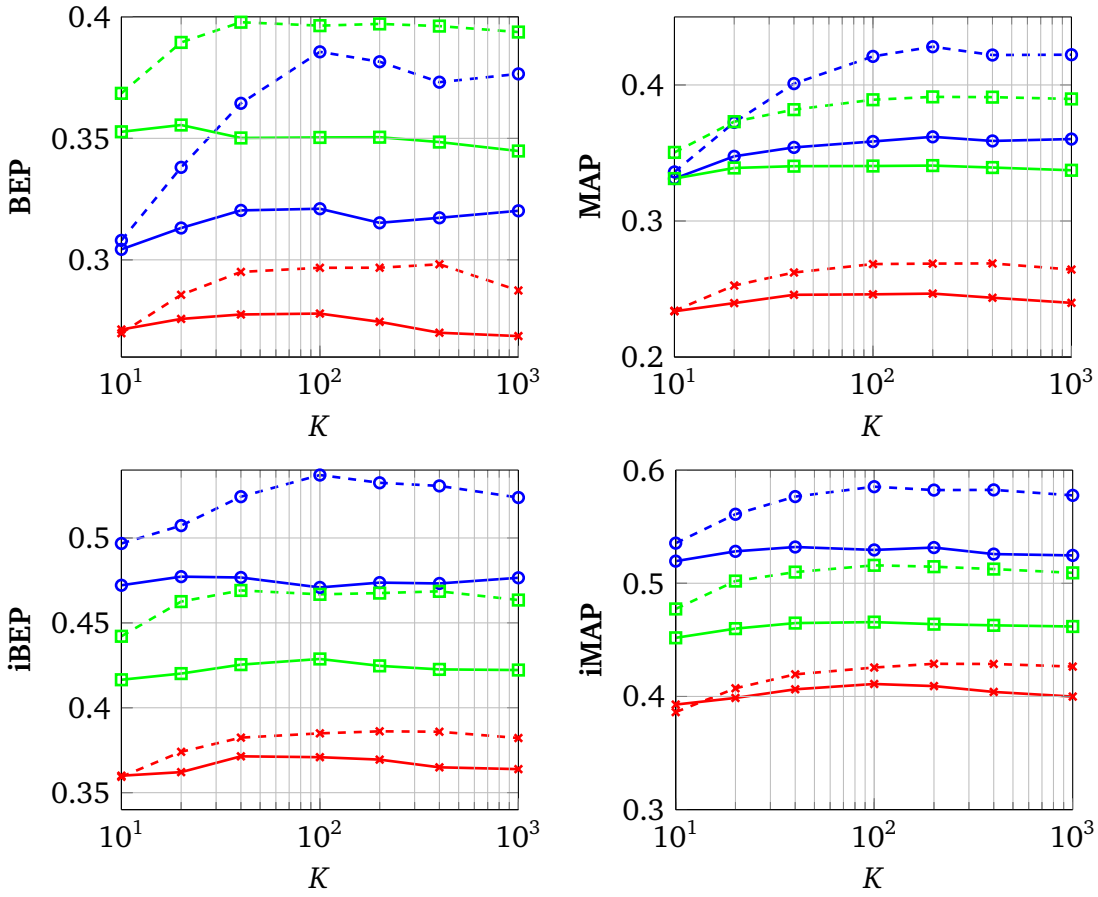


***Figure 4.12**: Performance in terms of* **BEP**, **MAP**, **iBEP** *and* **iMAP** *as a function of the neighbourhood size for all three data sets (Corel 5000: blue with circles; ESP Game: red with crosses; IAPR TC-12: green with square), comparing the JEC (solid) and learnt combination (dashed).*

4.5.5 Qualitative results

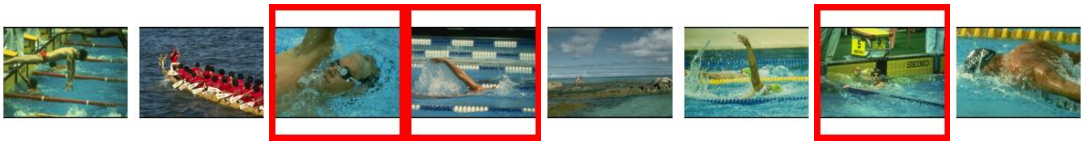
Below, we show qualitative results for image retrieval from keyword-based queries in Figure 4.13, and for image auto-annotation in Figure 4.14. All the qualitative results can be obtained from our online demo.[4]

It appears from these results that it is relatively frequent that visually relevant images are retrieved but considered wrong by the ground-truth annotation. This is related to the observation that the images are not associated with all the relevant tags, but only a subset. Similarly, for annotation, many relevant concepts are identified but are absent from the ground-truth labelling. This certainly disturbs the performance evaluation.

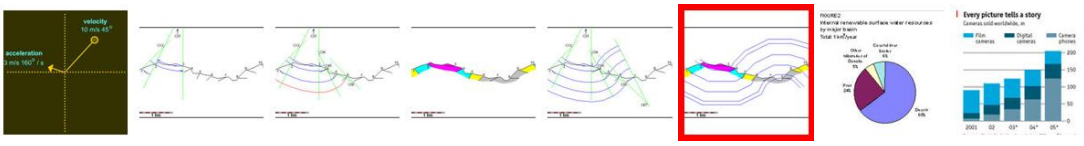
[4]At: `http://pascal.inrialpes.fr/local/tagprop/`.

Corel 5000, query **tiger**: for the 10 relevant images, **BEP**=100.00%.

Corel 5000, query **water & people**: for the 20 relevant images, **BEP**=60.00%.



ESP Game, query **chart**: for the 38 relevant images, **BEP**=68.42%.

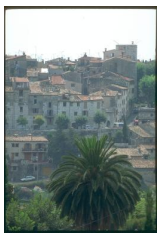


ESP Game, query **plant**: for the 48 relevant images, **BEP**=33.33%.

IAPR TC-12, query **cyclist**: for the 45 relevant images, **BEP**=80.00%.

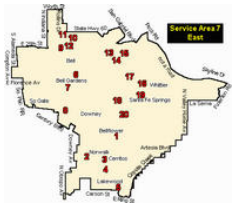
IAPR TC-12, query **backpack**: for the 18 relevant images, **BEP**=5.56%.

***Figure 4.13**: In the top two rows, retrieval examples from the Corel 5000 data set using queries "tiger" and the combined query "water & people". In the middle two rows, similar examples from the ESP Game data set for queries "chart" and "plant". Below, examples for the IAPR TC-12 data set using "cyclist" and "backpack". Irrelevant images according to the ground-truth are highlighted with a red border. As can be seen from these qualitative examples, it is frequent that relevant images are considered irrelevant from the absence of the corresponding tag.*



From *Corel 5000*
Ground-truth : **buildings** *(0.99)*, **tree** *(0.98)*, hills *(0.46)*, town *(0.35)*
Predictions : village *(1.00)*, **buildings** *(0.99)*, house *(0.99)*, **tree** *(0.98)*
BEP: 50%

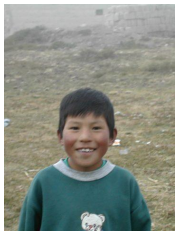
From *Corel 5000*
Ground-truth : **mountain** *(1.00)*, **tree** *(0.99)*, sky *(0.98)*, clouds *(0.94)*
Predictions : hillside *(1.00)*, **mountain** *(1.00)*, valley *(0.99)*, **tree** *(0.99)*
BEP: 50%



From *ESP Game*
Ground-truth : **map** *(0.99)*, **red** *(0.89)*, yellow *(0.79)*, black *(0.79)*, numbers *(0.23)*
Predictions : **map** *(0.99)*, white *(0.97)*, **red** *(0.89)*, chart *(0.85)*, blue *(0.85)*
BEP: 40%

From *ESP Game*
Ground-truth : **wave** *(0.99)*, **girl** *(0.99)*, flower *(0.97)*, black *(0.93)*, america *(0.11)*
Predictions : people *(1.00)*, woman *(1.00)*, **wave** *(0.99)*, group *(0.99)*, **girl** *(0.99)*
BEP: 40%

From *IAPR TC-12*
Ground-truth : **sky** *(0.99)*, terrain *(0.03)*
Predictions : cloud *(1.00)*, **sky** *(0.99)*
BEP: 50%



From *IAPR TC-12*
Ground-truth : **pullover** *(1.00)*, boy *(1.00)*
Predictions : cap *(1.00)*, **pullover** *(1.00)*
BEP: 50%

***Figure 4.14**: Automatic image annotation examples (2 from Corel 5000, 2 from ESP Game and 2 from IAPR TC-12). The ground-truth keywords are shown with their TagProp predictions (σMD, $K = 200$). Below, the same number of top TagProp predictions are shown, in bold if they are correct, with their score and the* **BEP** *measure (i.e., the precision of the shown predictions).*

4.6 Conclusion

In this chapter, we have introduced new models for image annotation and keyword based image retrieval. These models combine a weighted nearest-neighbour approach with metric learning capabilities in a discriminative framework. We add word-specific logistic discriminant modulation to deal with the varying word frequencies in a data-driven manner.

We reported extensive experimental results on three data sets, using several performance measures. From these results we conclude that our σML sigmoidal variant of TagProp that uses distance-based weights and integrates metric learning performs best. It combines high recall with good precision over all data sets. It gives significant improvements over the same model applied to uniformly combined distances, σDT. This contrasts with earlier attempts to use learnt distance combinations for tag prediction, see *e.g.* Makadia et al. [2008], that were unsuccessful because metric learning was not integrated in the prediction model. Our word specific modulation significantly improves recall for rare words as well as the overall performance. On all three data sets, and all seven evaluation measures, our model achieves performance significantly above all previously published results.

In future work, we will consider using approximate nearest neighbour techniques and learning an explicit data embedding using the low-rank Mahalanobis metric learning capabilities of TagProp, with very large scale applications in mind. We will also study the extension of the model to assign tags to image regions, in order to address tasks such as image region labelling, which would relate to image segmentation and scene parsing as performed in Liu et al. [2009a], and object detection from image-wide annotations. TagProp can also easily be adapted for multi-class classification. We plan to develop this alternative and evaluate its performance on standard multi-class problems that are numerous in computer vision and machine learning.

5

Multimodal semi-supervised learning for image classification

Contents

5.1 Introduction

The goal of image classification is to decide whether an image belongs to a certain category or not. Different types of categories have been considered in the literature, *e.g.* defined by presence of certain objects, such as cars or bicycles (Everingham et al. [2007]), or defined in terms of scene types, such as city, coast, mountain, etc... (Lazebnik et al. [2006]). To solve this problem, a binary classifier can be learnt from a collection of images manually labelled to belong to the category or not. Increasing the quantity and diversity of hand-labelled images improves the performance of the learnt classifier, however, labelling images is a time-consuming process. Although it is possible to label large amounts of images for many categories for research purposes (*c.f.* Deng et al. [2009]), this is often unrealistic, *e.g.* in personal photo management applications. This motivates our interest in using other sources of information that can aid the learning process using a limited amount of labelled images.

In this chapter, and as in the previous chapter, we consider a scenario where the training images have associated keywords or tags, such as found on photo sharing websites

Tags: desert, nature, landscape, sky
Labels: clouds, plant life, sky, tree

Tags: rose, pink
Labels: flower, plant life

Tags: india
Labels: cow

Tags: aviation, airplane, airport
Labels: aeroplane

***Figure 5.1**: Example images from MIR Flickr (top row) and VOC'07 (bottom row) data sets with their associated tags and class labels.*

like Flickr. Here, our goal is to learn a classifier for images alone, but we will use the tags associated with labelled and unlabelled images as additional features to improve the classifier using a semi-supervised approach. Image tags tend to be noisy in the sense that they might not directly relate to the image content, and typically only a few of many possible tags have been added to each image, as shown in Figure 5.1. Despite the noisy relation between tags and image content, they have been found a useful additional feature for fully supervised image categorisation (*c.f.* Li et al. [2009b], Wang et al. [2009]).

We propose a semi-supervised learning approach to leverage the information contained in the tags associated with unlabelled images in a two-step process. First, we use the labelled images to learn a strong classifier that uses both the image content and tags as features. We use the multiple kernel learning (MKL) framework (Lanckriet et al. [2004]) to combine a kernel based on the image content with a second kernel that encodes the tags associated with each image. This MKL classifier is used to predict the labels of unlabelled training images with associated tags. In the second step we use both the labelled data and the output of the classifier on unlabelled data to learn a second classifier that uses only visual features as input. Our work, pub-

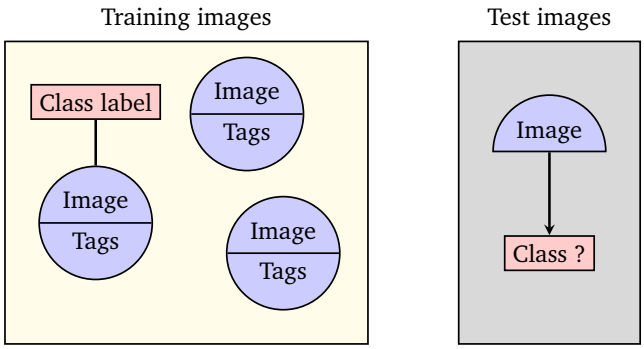


***Figure 5.2**: Overview of multimodal semi-supervised classification. Training images come with tags, and only a subset is labelled. The goal is to predict the class label of test images without tags.*

lished in Guillaumin et al. [2010b], is different from most work on semi-supervised learning as our labelled and unlabelled data have additional features that are absent for the test data. A schematic overview of the approach is given in Figure 5.2.

We perform experiments using the PASCAL VOC'07 (Everingham et al. [2007]) and MIR Flickr data sets (Huiskes and Lew [2008]) that were both collected from the Flickr website and for which user tags are available. The image sets have been manually annotated for 20 and 38 categories respectively. We measure performance using average precision on these manual annotations. In our experiments we confirm that the tags are beneficial for categorisation, and that our semi-supervised approach can improve classification results by leveraging unlabelled images with associated tags. We also consider a weakly-supervised scenario where we learn classifiers directly from the images tags, and do not use any manual annotation. Also in this case our approach can improve the classification performance by identifying images that are erroneously tagged.

In the next section we discuss the most relevant related work, and in Section 5.3 we present our method in detail. In Section 5.4 we present the data sets we used in our experiments and the feature extraction procedure. The experimental results follow in Section 5.5, and we conclude in Section 5.6.

5.2 Related work

Given the increasing amount of images that are currently available on the web with weak forms of annotation, there has been considerable interest in the computer vision community to leverage this data to learn recognition models. Examples are work on filtering images found using web image search, or images found on photo sharing

sites using keyword-based queries, see Berg and Forsyth [2006], Fergus et al. [2004, 2009], Schroff et al. [2007]. In Chapter 3, we have already discussed how image captions can be used to learn face recognition models without manual supervision (see also Berg et al. [2004a]), or to learn low dimensional image representations by predicting caption words that can be transferred to other image classification problems, *c.f.* Quattoni et al. [2007]. A related approach was taken by Wang et al. [2009] where classifiers were learnt to predict the membership of images to Flickr groups, and the difference in class membership probabilities were used to define a semantic image similarity.

Two recent papers that use tagged images to improve image classification performance are closely related to our work. In Li et al. [2009b] image tags were used as additional features for the classification of touristic landmarks. We also use image tags to improve the performance of our classifiers, but we do not assume their availability for test images. Wang et al. [2009] use a large collection of up to one million tagged images, to obtain a textual representation of images without tags. This is achieved by assigning an image the tags associated with its visually most similar images in the set of tagged images. Separate classifiers were learnt based on the visual and textual features, and their scores were linearly combined using a third classifier.

Our approach differs in that we do not construct a new textual image representation. Rather, we use the strength of classifiers that have access to images and associated tags to obtain additional examples to train a classifier that uses only visual features, thus casting the problem as a semi-supervised learning problem. There is a large literature on semi-supervised learning techniques. For sake of brevity, we discuss only three important paradigms, and we refer to Chapelle et al. [2006] for a recent book on the subject.

Generative approach

When using generative models for semi-supervised learning, a straightforward approach is to treat the class label of unlabelled data as a missing variable, see *e.g.* Baluja [1998], Nigam et al. [2000]. The class conditional models over the features can then be iteratively estimated using the EM algorithm. In each iteration the current model is used to estimate the class label of unlabelled data, and then the class conditional models are updated given the current label estimates. These methods are known to work well in cases where the model fits the data distribution well, but can be detrimental in cases where the model has a poor fit.

Current state-of-the-art image classification methods are discriminative ones that do not estimate the class conditional density models, but directly estimate a decision function to separate the classes. However, using discriminative classifiers, the EM

method of estimating the missing class labels used for generative models does not apply: the EM iterations immediately terminate at the initial classifier.

Coaching variables

This idea can be extended to our setting where we have variables that are only observed for the training data (Tibshirani and Hinton [1998]). The idea, referred to as "coaching", is to jointly predict the class label y and the missing text features z for the test data, and then marginalise over the unobserved text features:

$$p(y|x) = \int p(y,z|x)dz. \tag{5.1}$$

Assume for instance that y and z are conditionally independent given x, such that $p(y,z|x) = q(y|x)q(z|x)$, and that the two distributions $q(y|x)$ and $q(z|x)$ share some parameters. Then $p(y|x)$ is estimated as $q(y|x)$ and the benefit from using z comes from the estimation of the shared parameters.

Alternatively, Equation 5.1 can be written the following way:

$$p(y|x) = \int p(y|x,z)p(z|x)dz, \tag{5.2}$$

which leads to "mixture coaching" models. Benefit can be expected in these models if $p(z|x)$ varies with x. In our setting, $p(z|x)$ would be an image annotation model: it predicts the relevant tags for an image; and $p(y|x,z)$ is a combined classifier.

Co-training

Co-training (Blum and Mitchell [1998]) is a semi-supervised learning technique that also applies to discriminative classifiers, and is designed for settings like ours where the data is described using several different feature sets.

The idea is to learn a separate classifier using each feature set, and to iteratively add labelled training examples based on the output of the other classifier. In particular, in each iteration the examples that are most confidently classified, positively or negatively, with each classifiers are added as labelled examples to the training set.

A potential drawback of the co-training is that it relies on the classifiers over the separate feature sets to be accurate, at least among the most confidently classified

examples. In our setting we found that for most categories one of the two feature sets is significantly less informative than the other. Therefore, using the classifier based on the worse performing feature set might provide erroneous labels to the classifier based on the better performing feature set, and its performance might be deteriorated.

In the next section we present a semi-supervised learning method that uses both feature sets on the labelled examples, and we compare it with co-training in our experiments.

5.3 Multimodal semi-supervised learning

In this section we first present the supervised classification setup (Section 5.3.1), which forms the basis for the semi-supervised approach (Section 5.3.2).

5.3.1 Supervised classification

For our baseline image classification system we follow state-of-the-art image categorisation methods (Everingham et al. [2007]), and use support vector machines (SVM) with non-linear kernels based on several different image features. The kernel function $k(\cdot,\cdot)$ can be interpreted as a similarity function between images and is the inner product in an induced feature space. The SVM is trained on labelled images to find a classification function of the form:

$$f(x) = \sum_i \alpha_i k(x, x_i) + b. \tag{5.3}$$

For a test image, the class label $y \in \{-1, +1\}$ is predicted as $\text{sign}(f(x))$.

In order to combine the visual and textual representations we adopt the multiple kernel learning (MKL) framework (more precisely, the simpleMKL method of Rakotomamonjy et al. [2008]). Denoting the visual kernel by $k_v(\cdot,\cdot)$ and the textual kernel by $k_t(\cdot,\cdot)$, we can define a combined kernel as a convex combination of these: $k_c(\cdot,\cdot) = d_v k_v(\cdot,\cdot) + d_t k_t(\cdot,\cdot)$, where $d_v, d_t \geq 0$ and $d_v + d_t = 1$. The MKL framework allows joint learning of the kernel combination weights d_v, d_t and the parameters $\{\alpha_i\}$ and b of the SVM based on the combined kernel. The parameters are found by solving a convex, but non-smooth objective function.[1] Below, we will use f_v, f_t, and f_c to differentiate between classification functions based on the different kernels.

[1] We used the MKL implementation available at `http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/ikl-webpage/`.

5.3.2 Semi-supervised classification

Given these different classifiers, we now consider how we can apply them in a semi-supervised setting. We use $\mathscr{L}$ to denote the set of labelled training examples, and $\mathscr{U}$ to refer to the set of unlabelled training examples. As noted above, we assume that our training images have associated tags, but that our final task is to classify images that do not have such tags. We proceed by learning a first classifier on the labelled examples in $\mathscr{L}$, and then use it to predict the class labels for the unlabelled examples in $\mathscr{U}$.

In the case where the first classifier only uses the visual kernel, we do not expect to gain from the unlabelled examples as predicting their label is as hard as it would be for any test image. This is confirmed by our experimental results presented in Section 5.5. Our experimental results also show that the image tags make many of the classification tasks substantially easier. Therefore, we will use MKL to learn a joint visual-textual classifier from $\mathscr{L}$, and estimate the class labels for the images in $\mathscr{U}$. Assuming that the labels predicted using the MKL classifier f_c are correct, we train a visual-only SVM classifier f_v from all training examples in $\mathscr{L} \cup \mathscr{U}$.

In practice, however, the joint classifier is not perfect, and we consider two alternative approaches to leverage the predictions of the joint classifier on the unlabelled examples in $\mathscr{U}$. In the first alternative, we only add the examples that are confidently classified using the MKL classifier and fall outside the margin, *i.e.* those with $|f_c(x)| \geq 1$, instead of adding all examples in $\mathscr{U}$. This choice is motivated by the observation that these are precisely the examples that would not change the MKL classifier if they were included among the training data for it.

Our second alternative is motivated by the observation that the only information from the MKL classifier that we use when training the final visual classifier is the sign of the examples selected from $\mathscr{U}$. Therefore, the value of $f_v(x_i)$ can arbitrarily differ from $f_c(x_i)$ provided that it is consistent with the class labels of the labelled examples, and the estimated class label of the unlabelled ones. Instead, we will directly approximate the joint classification function f_c learnt using MKL. We do so by performing a least squares regression (LSR) on MKL scores $f_c(x)$ for all examples in $x \in \mathscr{L} \cup \mathscr{U}$, to find the function $f_v(x) = \sum_i \alpha_i k_v(x, x_i) + b$ based on the visual kernel. We choose to regularise LSR by projection on a lower-dimensional space using Kernel PCA (*c.f.* Shawe-Taylor and Cristianini [2004]). We perform singular value decomposition (SVD) to obtain a pseudo-inverse of $K_v = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$, the centred kernel matrix for k_v such that the columns have zero mean. We invert it by suppressing dimensions with singular value in $\mathbf{\Lambda}$ below $\epsilon = 10^{-10}$. Using $\mathbf{s}$ to denote the vector of centred classification scores obtained with f_c, we then obtain the α_i parameters in the vector $\boldsymbol{\alpha} = \mathbf{V}\overline{\mathbf{\Lambda}}\mathbf{U}^\top\mathbf{s}$, as described in Algorithm 2 below, and b is set to 0.

Algorithm 2: Procedure for learning a semi-supervised MKL+LSR visual classifier.

Input: Labelled data $\mathscr{L}$ and unlabelled data $\mathscr{U}$, visual kernel k_v, textual kernel k_t.
Output: Visual classifier $\boldsymbol{\alpha}$ using kernel k_v.

$f_c \leftarrow \text{MKL}(\mathscr{L}, \{k_v, k_t\})$ /* Learn MKL classifier */
foreach $\mathbf{x} \in \mathscr{L} \cup \mathscr{U}$ **do** /* Centre scores */
  $\mathbf{s}(\mathbf{x}) \leftarrow f_c(\mathbf{x}) - \left\langle f_c(\mathbf{x}') \right\rangle_{\mathbf{x}' \in \mathscr{L} \cup \mathscr{U}}$
end
foreach $\mathbf{x}, \mathbf{x}' \in \mathscr{L} \cup \mathscr{U}$ **do** /* Centre kernel columns */
  $\mathbf{K}_v(\mathbf{x}, \mathbf{x}') \leftarrow k_v(\mathbf{x}, \mathbf{x}') - \left\langle k_v(\mathbf{x}, \mathbf{x}'') \right\rangle_{\mathbf{x}'' \in \mathscr{L} \cup \mathscr{U}}$
end
$\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top = \mathbf{K}_v$ /* SVD of $\mathbf{K}_v$ */
for $i = 1$ **to** $|\mathscr{L} \cup \mathscr{U}|$ **do** /* Pseudo-invert $\mathbf{K}_v$ */
  $\overline{\Lambda}_{ii} \leftarrow \begin{cases} 0 & \text{if } \Lambda_{ii} < \epsilon \\ \Lambda_{ii}^{-1} & \text{otherwise} \end{cases}$
end
$\boldsymbol{\alpha} \leftarrow \mathbf{V}\overline{\boldsymbol{\Lambda}}\mathbf{U}^\top\mathbf{s}$ /* Least-squares regression of $\mathbf{s}$ */

5.4 Datasets and feature extraction

5.4.1 *PASCAL VOC 2007* and *MIR Flickr*

In our experiments we use the PASCAL VOC'07 (Everingham et al. [2007]) and the MIR Flickr (Huiskes and Lew [2008]) data sets. Both were collected from the Flickr website. Example images are given in Figure 5.1. For the PASCAL VOC'07 set we used the standard train/test split, and for the MIR Flickr set we randomly split the images into equally sized test and train sets.[2]

The PASCAL VOC'07 data set contains around 10000 images which were downloaded by querying for images of 20 different object categories in a short period of time. All the images were then annotated for each of the 20 categories. Using the image identifiers we downloaded the user tags for the 9587 images (*i.e.*, $\sim 95\%$ of the data set) that were still available on Flickr at time of download, and assumed complete absence of tags for the remaining ones.

The MIR Flickr data contains 25000 images collected by downloading images from Flickr over a period of 15 months. The collection contains images under the Creative Common license that scored highest according to Flickr's "interestingness" score. These images were annotated for 24 concepts, including object categories but also

[2]The test/train division for the MIR Flickr set and our visual and textual features described hereafter are publicly available at: `http://lear.inrialpes.fr/data/`.

***Figure 5.3**: For three (bird, car and night) of the fourteen concerned classes in MIR Flickr, we show in the top row a positive example image for the strict interpretation of the visual concept (bird∗, car∗ and night∗, respectively) and in the bottom row an example that is positive for the weaker sense of the concept and negative for the stricter sense. Note that a positive image in the strict sense is always positive in the weak sense.*

more general scene elements such as *sky, water* or *indoor*. For 14 of the 24 concepts a second, stricter, annotation was made: for each concept a subset of the positive images was selected where the concept is salient in the image. We refer to these more strictly annotated classes by using $*$ as a suffix, and provide some visual examples in Figure 5.3. In total we therefore have 38 categories for this data set.

5.4.2 Textual features

We use a binary vector $t_i \in \{0,1\}^W$ to encode the absence or presence of each of the W different tags in a fixed vocabulary in a linear kernel $k_t(t_i, t_j) = t_i^\top t_j$ which counts the number of tags shared between two images.

As shown in Figure 5.4, the tags associated with the training images in the *PASCAL VOC 2007* data set have very varying frequencies. More than ten thousands of them appear just once. We do not expect to benefit from these very rare tags. Therefore, for the *PASCAL VOC 2007* data set, we restricted the tag vocabulary to the ones that appear at least four times in the training and 4 times in the test sets. There are 804 such tags, such that the textual representations are binary vectors of size $W = 804$.

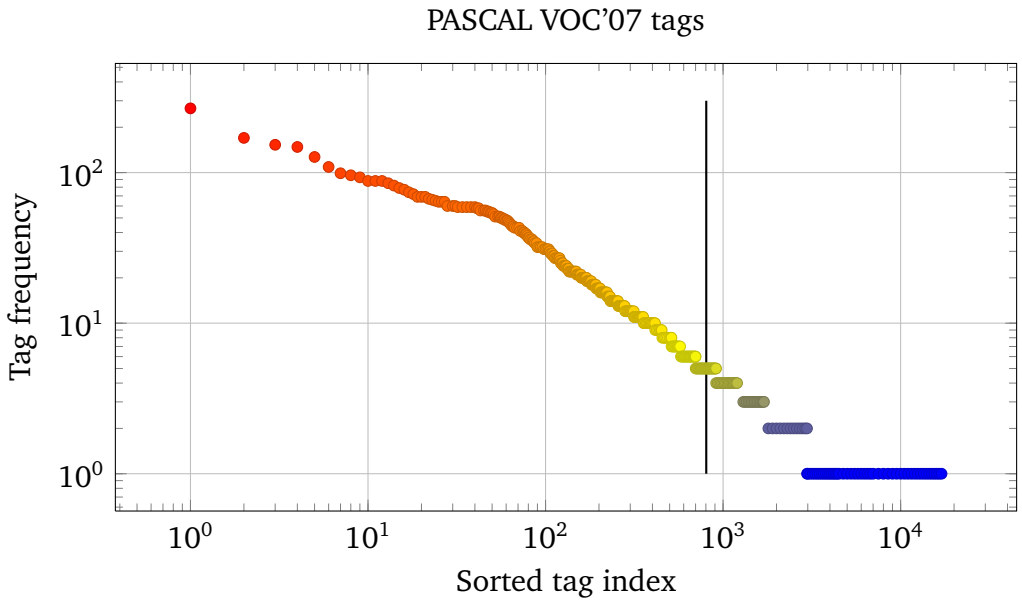


***Figure 5.4**: Tag frequency in the PASCAL VOC 2007 data set. Each of the ∼17000 dots corresponds to a keyword in the training data set. Their index are sorted by frequency, and the y-axis shows how often this tag appears in images.*

For the *MIR Flickr* data set, similar observations can be made. We kept the tags that appear at least 50 times (*i.e.* among at least 0.2% of the images), resulting in a vocabulary of 457 tags.

5.4.3 Visual features

For each image we extracted several different visual descriptors. We then averaged the distances between images based on these different descriptors, and use it to compute an RBF kernel. Thus, our visual kernel is defined as:

$$k_v(x_i, x_j) = \exp(-\lambda^{-1} d(x_i, x_j)), \tag{5.4}$$

where the scale factor λ is set to the average pairwise distance:

$$\lambda = N^{-2} \sum_{i,j=1}^{N} d(x_i, x_j), \tag{5.5}$$

$$d(x_i, x_j) = \sum_{m=1}^{M} \lambda_m^{-1} d_m(x_i, x_j), \tag{5.6}$$

$$\lambda_m = \max_{i,j} d_m(x_i, x_j). \tag{5.7}$$

Although orthogonal to the focus of this paper, we could also use MKL to learn the combination of separate visual kernels for each feature set.

As in Chapter 4 and in Guillaumin et al. [2009a], we use local SIFT features (Lowe [2004]), and local hue histograms (van de Weijer and Schmid [2006]), both were computed on a dense multi-scale grid and on regions found with a Harris interest-point detector. We quantise the local descriptors using k-means, and represent the image using a visual word histogram. We also compute global colour histograms over RGB, HSV, and LAB colour spaces.

Following Lazebnik et al. [2006], these histogram image representations were also computed over a 3×1 horizontal decomposition of the image, and concatenated to form a new representation that also encodes some of the spatial layout of the image. Furthermore we use the GIST descriptor (Oliva and Torralba [2001]), which roughly encodes the image layout. In total we thus combine $M = 15$ different image representations, using L1 distance for the colour histograms, L2 for GIST, and χ^2 for the visual word histograms.

We refer the reader back to Section 4.4.4 for additional details.

5.5 Experimental results

In our experiments we measure performance using the average precision (AP) criterion for each class, and also using the mean AP (mAP) over all classes.

5.5.1 Supervised classification

Our first set of experimental results, presented in Table 5.1, compares the classification performance using the visual representation and the tags, and their combination with MKL. For both data sets, we observe that it depends on the class whether the visual or the textual kernel is the strongest. On the particular choice of classes in the data sets, the visual classifier is typically stronger than the textual one, yielding a 10% higher mAP score.

Also on both data sets, the combined MKL classifier is significantly improving the results of visual object classification, the mAP score increases by more than 13% on the VOC classes and by more than 9% on the MIR classes. Only for four classes of PASCAL (*chair, diningtable, person* and *sofa*) and nine of MIR (*clouds, indoor, lake, male, night, sea, sky, sunset* and *tree*), the MKL does not significantly improve over the best of the two kernels. Interestingly, the mAP of 0.667 obtained by combining visual features and tags is also significantly above the 0.594 winning score of the VOC'07 which used a visual classifier alone.

PASCAL VOC'07	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable
Image	0.727	0.530	0.491	0.668	0.256	0.524	0.699	0.500	0.460	0.364	**0.433**
Tags	0.667	0.407	0.608	0.375	0.197	0.292	0.513	0.664	0.153	0.393	0.076
Image+Tags	**0.879**	**0.655**	**0.763**	**0.756**	**0.315**	**0.713**	**0.775**	**0.792**	**0.462**	**0.627**	0.414

	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	Mean
Image	0.439	0.747	0.595	0.834	0.390	0.395	0.399	0.743	0.428	0.531
Tags	0.570	0.676	0.539	0.635	0.248	0.457	0.191	0.712	0.278	0.433
Image+Tags	**0.746**	**0.846**	**0.762**	**0.846**	**0.480**	**0.677**	**0.443**	**0.861**	**0.527**	**0.667**

MIR Flickr	animals	baby	baby*	bird	bird*	car	car*	clouds	clouds*	dog	dog*	female	female*
Image	0.487	0.170	0.214	0.227	0.293	0.375	0.522	0.825	**0.755**	0.323	0.367	0.575	0.549
Tags	0.548	0.235	0.315	0.381	0.458	0.246	0.213	0.499	0.378	0.578	0.572	0.488	0.422
Image+Tags	**0.646**	**0.357**	**0.448**	**0.520**	**0.631**	**0.451**	**0.619**	**0.827**	0.753	**0.681**	**0.728**	**0.617**	**0.601**

	flower	flower*	food	indoor	lake	male	male*	night	night*	people	people*	plant life	portrait
Image	0.536	0.643	0.501	0.745	0.313	0.517	0.450	0.649	0.558	0.789	0.751	0.785	0.681
Tags	0.494	0.546	0.367	0.603	0.231	0.441	0.339	0.416	0.271	0.722	0.635	0.617	0.455
Image+Tags	**0.653**	**0.742**	**0.606**	**0.770**	**0.341**	**0.561**	**0.496**	**0.686**	**0.596**	**0.835**	**0.795**	**0.809**	**0.711**

	portrait*	river	river*	sea	sea*	sky	structures	sunset	transport	tree	tree*	water	Mean
Image	0.682	0.265	0.081	0.571	0.334	0.866	0.774	0.665	0.464	0.671	0.548	0.622	0.530
Tags	0.451	0.255	0.035	0.400	0.132	0.670	0.694	0.407	0.365	0.413	0.266	0.539	0.424
Image+Tags	**0.711**	**0.412**	**0.202**	**0.649**	**0.362**	**0.876**	**0.803**	**0.666**	**0.540**	**0.684**	**0.564**	**0.717**	**0.623**

***Table 5.1**: The AP scores for the supervised setting on both data sets, with the visual kernel alone (Image), a linear SVM on tags (Tags), and the combined kernel (Image+Tags) obtained by Multiple Kernel Learning. The best classification results for each class are marked in bold.*

These results are in line with those of Li et al. [2009a], where visual features and tags were combined for landmark classification. A difference is that we find the visual features to be stronger on average than the textual ones, where the situation was reversed in Li et al. [2009a]. This might be due to the fact that they used a weaker linear classifier on the visual features, or due to the different type of classification problems: landmarks might be more likely to be tagged than classes such as *diningtable*. Wang et al. [2009] also found textual features to improve the performance of visual classifiers, but only for relatively weak visual classifiers and not for strong non-linear classifiers.

5.5.2 Semi-supervised classification

In this section we present results for semi-supervised learning. We compare the following methods:

- SVM: visual classifier learnt on labelled examples,
- MKL+SVM(0): MKL classifier learnt on the labelled examples, followed by a visual SVM trained on all training examples using the MKL label prediction,
- MKL+SVM(1): same as MKL+SVM(0) but excluding the unlabelled examples in the margin of the MKL classifier to train the SVM,
- MKL+LSR: uses least-squares regression on the MKL scores for all examples to obtain the visual classifier,
- SVM+SVM(0): same as MKL+SVM(0) but using the visual SVM to predict the class of unlabelled examples.
- Co-training: iterative learning of textual and visual classifiers using the co-training paradigm.

The regularisation parameters of the SVM and MKL algorithms can be set using cross-validation, but for the sake of efficiency we adopted the constant value of $C = 10$ for all experiments after observing that this value was selected for many classes and settings in initial experiments. We do not expect major differences when performing cross-validation per class and experiment.

The co-training approach has a number of additional parameters to set: the number of iterations T in which examples are added, and the number of positive and negative examples to add in each iteration, which we denote as p and n respectively. Setting these parameters using cross-validation is relatively costly as each co-training iteration requires re-training of the visual and textual SVM classifiers. For two classes

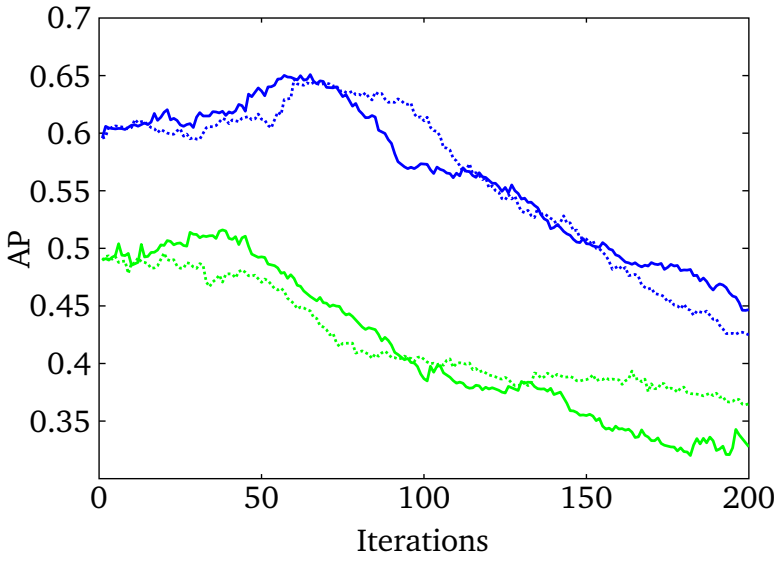


***Figure 5.5**: AP scores for the classes aeroplane (blue) and boat (green), using co-training with $p = 1, n = 3$ (solid) and $p = 1, n = 1$ (dashed), with varying number of co-training iterations.*

of the VOC'07 set we evaluated the performance over the first 200 iterations using $p = 1, n = 1$ and $p = 1, n = 3$, the latter reflecting the fact that for each class there are many more negative than positive examples.

From the results shown in Figure 5.5, we observe that using many iterations seems to have a detrimental effect on performance. This might be explained by the small number of positive examples in the unlabelled set. Given these results we used $p = 1, n = 3$ and compared $T = 30$ and $T = 50$ for the VOC'07 data set. Since only little difference was observed between the two options in terms of performance, we later opted for $T = 30$ for the MIR Flickr data set in order to reduce the computational load of the experiment.

We evaluated the performance for different amounts of labelled training images. In one set of experiments we randomly selected $k \in \{20, 50, 100\}$ positive and the same number of negative examples for each class. In another set of experiments we use a fraction $r \in \{10\%, 25\%, 50\%\}$ of the positive and negative examples from each class, *i.e.* with $r = 10\%$ and for a class with 2500 positive images and 10000 negative ones we randomly select 250 positive examples and 1.000 negative examples. Note that using 10% of the labelled images means that we use a total of 500 and 1250 labelled images for the VOC and MIR sets respectively, that is, many more than in the $k = 100$ setting.

In Table 5.2 we report the mAP scores for both data sets for the different learning algorithms with varying amounts of labelled data. For the sake of clarity, we report

PASCAL VOC'07	20	50	100	10%	25%	50%
SVM	0.268	0.294	0.370	0.345	0.427	0.468
MKL+SVM(0)	0.284	0.314	0.352	0.410	**0.458**	**0.482**
MKL+SVM(1)	0.278	0.322	0.371	0.367	0.440	0.478
SVM+SVM(0)	0.244	0.266	0.328	0.303	0.395	0.455
MKL+LSR	**0.336**	**0.366**	**0.406**	**0.413**	**0.458**	**0.482**
Co-training(30)	0.287	0.323	0.381	0.360	0.438	0.475
Co-training(50)	0.285	0.328	0.377	0.374	0.441	0.476

MIR Flickr	20	50	100	10%	25%	50%
SVM	0.276	0.333	0.370	0.412	0.462	0.501
MKL+SVM(0)	0.272	0.334	0.365	**0.441**	**0.479**	0.505
MKL+SVM(1)	0.283	0.340	0.373	0.424	0.471	0.504
SVM+SVM(0)	0.267	0.319	0.358	0.392	0.444	0.490
MKL+LSR	**0.316**	**0.367**	**0.395**	0.431	0.475	**0.510**
Co-training(30)	0.286	0.351	0.380	0.420	0.471	0.504

***Table 5.2**: Performance in mAP on the two data sets for different learning methods and various amounts of labelled training images.*

the individual AP of the 58 classes only when using 50 labelled training examples per class, see Table 5.3 and Table 5.4.

We observe that overall semi-supervised learning significantly improves the performance of the baseline visual-only SVM, in particular when few labelled training data is available. However, it does so only when using the textual features; the visual-only SVM+SVM(0) approach performs worse than the baseline on average and consistently for almost all classes and amount of labelled data. In cases with up to 100 positive and negative examples, MKL+SVM(0) seems to generalise better than MKL+SVM(1), and the MKL+LSR method clearly outperforms all other semi-supervised approaches, including co-training. As larger sets of labelled examples are available, all the methods except SVM+SVM(0) tend to perform similarly. From the per-class results in Table 5.4, we observe that the gain varies strongly across classes. For four out of the 38 MIR Flickr classes the baseline supervised classifier performs best: *male*$*$, *river*$*$, *tree* and *tree*$*$. However, this is largely compensated for by the improvements on the 34 other classes obtained by our MKL+LSR method.

5.5.3 Learning classes from Flickr tags

In our third set of experiments we consider learning classifiers without using any manually labelled examples. For this purpose we use the 18 classes of the MIR Flickr set for which the class name also belongs to the tag dictionary. For the training im-

PASCAL VOC'07	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable
SVM	0.387	0.218	0.217	0.462	0.150	0.213	0.439	0.271	0.265	0.112	0.258
MKL+SVM(0)	0.549	0.163	0.271	0.409	**0.169**	0.253	0.453	0.311	**0.310**	0.127	0.261
MKL+SVM(1)	0.479	0.218	0.248	0.466	0.145	0.233	0.467	0.296	0.297	**0.186**	0.247
MKL+LSR	**0.592**	**0.324**	**0.376**	**0.519**	0.154	0.278	**0.501**	**0.366**	0.300	0.117	0.255
SVM+SVM(0)	0.326	0.185	0.201	0.398	0.142	0.205	0.444	0.233	0.299	0.108	0.233
Co-training (30)	0.475	0.199	0.299	0.400	0.158	**0.326**	0.497	0.306	0.209	0.148	**0.299**

	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	Mean
SVM	0.318	0.347	0.321	0.651	0.199	0.182	0.175	0.451	0.239	0.294
MKL+SVM(0)	0.280	0.452	0.251	0.685	0.181	0.213	0.183	0.550	0.219	0.314
MKL+SVM(1)	0.306	0.464	0.326	0.652	0.209	**0.239**	**0.193**	0.522	0.238	0.322
MKL+LSR	**0.331**	**0.637**	**0.383**	**0.703**	**0.212**	0.218	0.191	**0.617**	0.236	**0.366**
SVM+SVM(0)	0.310	0.215	0.249	0.647	0.143	0.218	0.164	0.403	0.197	0.266
Co-training (30)	0.289	0.517	0.362	0.662	0.148	0.233	0.170	0.517	**0.249**	0.323

***Table 5.3**: AP scores for the 20 classes of the PASCAL VOC 2007 data set using 50 positive and 50 negative labelled examples for each class.*

MIR Flickr	animals	baby	baby*	bird	bird*	car	car*	clouds	clouds*	dog	dog*	female	female*
SVM	0.299	0.043	0.162	0.057	0.094	0.204	0.246	0.569	0.481	0.155	0.181	0.431	0.319
MKL+SVM(0)	0.278	0.055	0.151	**0.141**	0.065	0.210	0.228	0.573	0.503	0.124	0.170	0.436	0.321
MKL+SVM(1)	0.300	0.037	0.159	0.085	0.077	0.220	0.242	0.597	0.508	0.160	0.176	0.425	0.324
MKL+LSR	0.310	**0.075**	**0.161**	0.124	**0.163**	**0.229**	**0.305**	**0.612**	**0.537**	0.182	**0.212**	0.440	0.313
SVM+SVM(0)	0.266	0.038	0.146	0.054	0.073	0.196	0.224	0.560	0.446	0.151	0.169	0.432	0.312
Co-training (30)	**0.345**	0.035	0.136	0.076	0.097	0.199	0.287	0.597	0.471	**0.187**	0.194	**0.443**	**0.357**

	flower	flower*	food	indoor	lake	male	male*	night	night*	people	people*	plant life	portrait
SVM	0.264	0.359	0.295	0.518	0.139	0.358	**0.296**	0.471	0.289	0.588	0.529	0.602	0.443
MKL+SVM(0)	0.278	0.360	0.267	0.522	0.137	0.319	0.295	0.439	0.259	0.612	**0.553**	0.600	0.441
MKL+SVM(1)	0.353	0.387	0.297	0.516	0.132	0.312	0.281	**0.482**	0.285	0.615	0.545	0.617	**0.477**
MKL+LSR	**0.373**	**0.424**	**0.333**	0.514	0.159	0.366	0.255	0.471	**0.368**	0.629	**0.554**	0.613	0.474
SVM+SVM(0)	0.197	0.343	0.289	0.519	0.122	0.358	0.267	0.460	0.222	0.586	0.528	0.602	0.441
Co-training (30)	0.359	0.419	0.282	**0.559**	**0.172**	**0.380**	0.249	0.466	0.289	**0.634**	0.544	**0.634**	0.465

	portrait*	river	river*	sea	sea*	sky	structures	sunset	transport	tree	tree*	water	Mean
SVM	0.404	0.154	**0.054**	0.361	0.166	0.661	0.614	0.470	0.285	**0.461**	**0.254**	0.378	0.333
MKL+SVM(0)	0.413	0.150	0.047	0.410	0.209	0.670	0.649	0.503	0.278	0.458	0.155	0.413	0.334
MKL+SVM(1)	0.421	0.149	0.041	0.423	0.158	0.673	0.643	0.515	0.279	0.439	0.178	0.406	0.340
MKL+LSR	0.429	**0.234**	0.047	**0.437**	**0.255**	0.693	**0.655**	**0.543**	**0.321**	0.453	0.231	**0.452**	**0.367**
SVM+SVM(0)	0.391	0.135	0.043	0.357	0.147	0.655	0.615	0.448	0.275	0.454	0.230	0.374	0.319
Co-training (30)	**0.432**	0.181	0.050	0.417	0.213	**0.705**	0.636	0.493	0.273	0.443	0.209	0.426	0.351

***Table 5.4**: AP scores for the 38 classes of the MIR Flickr data set using 50 positive and 50 negative labelled examples for each class.*

ages we exclude the class name from the textual representation to avoid learning a degenerate classifier that uses the tag to perfectly predict itself.

As before, performance is measured using AP based on the manual ground truth class labels on the test set. Our baseline approach takes all images tagged with the class name as positives, and all other images as negatives.

The tags have a noisy relation to the class labels since the tags are not always relevant to the image content, and most images have only a few tags and lack many relevant ones. The positive examples from tag annotation are relatively clean (82.0% precision averaged over all 18 classes), but a large portion of the true positive images is not tagged (17.8% recall on average).

As in the semi-supervised setting, we first learn a joint visual-textual MKL classifier, albeit from all 12500 images in this case, and then use it to learn a visual only classifier. In this setting we use our semi-supervised approach to remove examples that are likely to be incorrectly tagged, rather than to add unlabelled examples. Given that the positive examples have a relatively low label noise, and that we have many more negative examples than positives, we will remove only the negative examples with the highest scores according to the MKL classifier. We experimented with removing between 2000 and 10000 negative examples from the total 12500 training examples.

In Table 5.5 we show the performance of the baseline visual-only SVM and of the MKL+LSR approach for various numbers of negative examples that were removed. Not surprisingly, when learning from the user tags, AP scores are lower than those obtained using manual annotations for training, *c.f.* results for "Image" in Table 5.1. However, also in this more difficult scenario, our semi-supervised approach improves on average over the performance of the baseline that directly learns a visual classifier from the noisy labels.

As before, the results vary strongly among the classes: for 5 classes the baseline is better (up to 5.6% on *baby*), while for 13 classes our MKL+LSR approach improves results (up to 9.8% on *night*). On average, the improvement is 2.2%. On the same subset of 18 classes, the supervised approach has a mAP of 53.0% compared to 40.7% for MKL+LSR, demonstrating the significant gain obtained by adding supervised information.

5.6 Conclusion and Discussion

In this chapter, we have considered how learning image classifiers can benefit from unlabelled examples in the case where the training images have associated tags. We presented a novel semi-supervised approach that operates in two stages. First, we learn a strong classifier from the labelled examples that uses both visual features

	Removed	animals	baby	bird	car	clouds	dog	flower	food	lake	night
SVM	0	0.304	**0.133**	**0.180**	0.288	0.621	0.249	0.438	0.402	**0.256**	0.465
MKL+LSR	0	0.279	0.082	0.167	0.298	0.628	0.237	0.437	0.405	0.237	0.485
MKL+LSR	2000	0.279	0.073	0.173	0.304	0.662	0.255	0.464	**0.429**	0.207	0.525
MKL+LSR	4000	0.285	0.078	0.128	**0.307**	0.679	**0.258**	**0.468**	0.427	0.254	0.544
MKL+LSR	8000	0.299	0.077	0.129	0.305	0.695	0.256	0.462	0.419	0.216	0.563
MKL+LSR	10000	**0.313**	0.076	0.114	0.293	**0.698**	0.250	0.454	0.414	0.208	**0.565**

	Removed	people	portrait	river	sea	sky	sunset	tree	water		Mean
SVM	0	0.556	0.440	**0.216**	0.353	0.656	0.600	0.368	0.403		0.385
MKL+LSR	0	0.578	0.450	0.214	0.336	0.650	0.593	0.370	0.402		0.380
MKL+LSR	2000	0.582	0.480	0.164	0.362	0.665	0.615	0.372	0.430		0.391
MKL+LSR	4000	0.589	0.503	0.182	0.380	0.676	0.613	0.388	0.445		0.400
MKL+LSR	8000	0.606	**0.517**	0.181	0.418	0.695	**0.614**	0.413	**0.463**		**0.407**
MKL+LSR	10000	**0.616**	**0.517**	0.178	**0.432**	**0.708**	0.604	**0.428**	0.461		**0.407**

***Table 5.5**: AP scores for 18 of the MIR Flickr classes when learning from image tags using a visual-only SVM approach and our MKL+LSR approach that also uses the image tags. For the latter, we removed varying amounts of negative examples to obtain the visual-only classifier.*

and tags as inputs. The first classifier is then evaluated on both the labelled and unlabelled training examples. In the second stage we learn a visual-only classifier by fitting a function on the scores of the strong classifier, or re-training a classifier.

Our experiments compared several variants of this semi-supervised approach with a co-training approach and SVM baselines. From the results we conclude the following: (i) The tags provide a useful feature that improves classification performance for most classes when combined with visual features. (ii) Classifiers learnt from limited amounts of labelled training data can be improved by using unlabelled training images, but only when additional information in the form of tags is available. (iii) Our semi-supervised method that uses regression to learn the second visual-only classifier outperforms the other approaches we considered. (iv) When learning from noisy image tags rather than manual labelling we can improve the performance by using our multimodal semi-supervised approach to remove noisy negative examples.

In parallel, we also considered learning the textual-visual classifier and the visual-only classifier *jointly*, rather than *sequentially* as presented in this paper. However, it appeared unclear how to make the combined classifier benefit from the visual classifier. Integrating discriminative classifiers (such as MKL, SVM or logistic regression) and image auto-annotation in the "coaching" framework is a natural extension of our work to be considered.

In future work, we also want to explore more powerful text representations than the current linear kernel over binary tag absence/presence vectors. In addition, we will consider automatically adding unlabelled training data from Flickr, which can provide us with millions of tagged images. Using these additional images in combination with the existing labelled data we hope to improve state-of-the-art performance on these benchmarks without additional manual labelling.

6

Conclusion

In this dissertation, we have explored models for face recognition and more general visual object classification using weak supervision that comes from metadata. For the task of face recognition, we have used the captions that accompany news images to learn face recognition models automatically, while we have exploited user tags as found on Flickr to address image auto-annotation and keyword-based image retrieval and also to improve object recognition. In Section 6.1, we summarise our contributions and conclusions for each task, and in Section 6.2, we propose directions for future research.

6.1 Contributions

- For the task of face recognition in uncontrolled settings, we have introduced in Guillaumin et al. [2009b] a supervised metric learning algorithm, Logistic discriminant-based Metric Learning (LDML), which we use for face verification. Additionally, we have extended nearest neighbour approaches to classify pairs of examples whose respective classes are not present in the training set. This method, marginalised k-nearest neighbour classification (MkNN), combined with metric learning methods to define the neighbourhood of a face image, helps improve the accuracy of verification, but at a relatively high computational cost. Using the two methods described above, we have obtained state-of-the-art results on the challenging *Labeled Faces in the Wild* data set of face images in uncontrolled setting, with application to unconstrained clustering of faces. LDML has been extended to perform dimensionality reduction (*c.f.* Guillaumin et al. [2010a]), showing that face descriptors with as few as 20 dimensions can advantageously replace full Mahalanobis metrics with the same level of performance.

- For the problem of face naming, *i.e.* associating names extracted from captions to faces detected in the corresponding images of news events, we have

presented in Guillaumin et al. [2008] a graph-based approach for constrained clustering. To approximate the solution of the resulting NP-hard problem, we resort to a form of local optimisation where we express local constraints as a bipartite graph matching problem which can be solved exactly and efficiently. To evaluate our methods, we have introduced the *Labeled Yahoo! News* data set, a manually annotated subset of the *Yahoo! News* data set, which consists of 20000 news images with captions. We have compared our approach using a collection of different edge weights to constrained clustering with mixtures of Gaussians. When only the most confident faces are named, our graph-based approach significantly outperforms the generative one. At the opposite, when trying to name the maximum number of faces, the generative approach obtains a better accuracy. In both approaches, the accuracy of the resulting associations is very good, and significantly improved when using learnt metrics. For instance, with LDML projections, precision levels around 90% can be obtained with both methods, but for different numbers of named faces.

- We have also proposed in Guillaumin et al. [2010c] to exploit news images with captions directly to learn metrics for face recognition. To this end, we have introduced a Multiple Instance Learning (MIL) formulation of metric learning, MildML, that optimises a metric without explicitly assigning a name to each face. We compare MildML to the baseline approach of learning an LDML metric for the automatically labeled faces obtained by the face naming algorithm mentioned above. Experimentally, we have shown that our MIL formulation of metric learning improves over the LDML baseline. Learnt metrics from such automatic supervision perform well, although not as good as fully supervised ones for our tasks of face verification and naming.

- For the problem of image auto-annotation, we have proposed in Guillaumin et al. [2009a] a weighted nearest neighbour model, TagProp, to annotate an image using the tags of its most similar images. TagProp differs from other nearest neighbour approaches in that it automatically sets the neighbourhood size to use and is able to optimise the similarity measure to obtain the neighbours that best predict the image tags. Word-specific modulations are proposed to improve the recall of rare words in nearest neighbour methods, which can otherwise be low. We have considered several parametrisations of our model, using rank-based and distance-based weights, and extended them for handling a collection of similarity measures, for instance by learning their optimal combination. For both image auto-annotation and keyword-based retrieval, we have obtained state-of-the-art performance on the challenging *Corel 5000*, *ESP Game* and *IAPR TC-12* data sets. The integration of metric learning directly in our model is one of the keys to explain the high performance of TagProp.

- Finally, we have addressed in Chapter 5 and in Guillaumin et al. [2010b] the more general problem of image categorisation. Using two data set of images with associated tags, namely *PASCAL VOC 2007* and *MIR Flickr*, we have studied the training of visual models from the weak supervision provided by image tags. We have also introduced a *multimodal semi-supervised learning* framework and proposed an approach to successfully exploit this setting. The idea is to regress, using visual features alone, stronger classification functions that can be learnt from multimodal data. We have experimentally shown our method to outperform non-linear SVMs trained on labeled data and also co-training, a state-of-the-art approach that can also profit from a textual classifier to improve the visual one in a semi-supervised setting.

For news images with captions and for images with user tags, we have successfully shown that the additional information conveyed by textual data is exploitable for improving the learning of visual models. This is particularly interesting since it can allow access to very large archives without any explicit manual labelling, in a way that is semantically more meaningful than using visual information alone.

6.2 Perspectives for future research

Building on our work, we propose several directions for future research, that go from low level feature extraction to long-term research goals.

- As discussed in Chapter 2, our face description pipeline does not model explicitly the variations in face pose. Since major pose changes are still the main cause of failure for our algorithm, it is natural to consider models that include explicit handling of pose for recognising faces. There are two ways to address this problem. First, it is possible to improve the alignment procedure. In this way, the descriptors are naturally more robust to pose changes. Second, it can be done by improving the detection of facial features. These detections have to be adapted to handle the absence of visibility of the features, a typical consequence of large pose variations. At the same time, this would address the problem of occlusions, which are also a major cause of failure in recognition.
- As we showed with TagProp, the integration of metric learning directly in the model improves over metric learning performed using a different objective function. For our proposed marginalised nearest neighbour classification, we have used metrics learnt for standard nearest neighbour classification. Nevertheless, directly optimising the metric for MkNN classification is possible. The straightforward implementation would lead to a algorithm that grows with $O(N^4)$ if

N is the number of training points, but more efficient implementations are possible. To improve the computational time of both training and test steps, approximate nearest neighbour techniques can also be considered.

- More generally, there are options of metric learning that we have not fully experimented. We can perform non-linear metric learning by using the kernelized version of LDML, or explicitly embed the data in a higher dimensional space. This a promising solution to handle different poses in the face recognition process. Similarly, the metric learning capabilities of TagProp have been, in our experiments, restricted to learning a positive combination of several distances. We showed how TagProp can be extended to more general Mahalanobis metrics and, just as LDML, to non-linear kernelized extensions. Notably, the low-rank version of TagProp would effectively learn a data representation such that the weighted nearest neighbour predictions would depend on the simple Euclidean distance. This would take advantage of the efficient computation and indexing techniques that exist for the L2 metric.

- Concerning our work on general image auto-annotation, TagProp performs feature selection at train time, in such a way that the extraction of useless features can be skipped at test time to speed up the process. This feature selection, and more generally the feature weighting that is learnt, can be used directly to define very good kernels for classification (*c.f.* Douze et al. [2009]). TagProp is therefore a very promising machine learning tool to explore for other learning scenarios and applications. We are planning to extend it to multi-class classification, image region classification and other applications. For instance, a max-margin objective comparable to Grangier and Bengio [2008] could be used to optimise the parameters of TagProp for ranking and retrieval. Multi-class classification can be obtained from TagProp by replacing the sigmoidal modulations with a soft-max output, or by replacing the multiple binary Bernoullis in the model with a multinomial.

- For both caption analysis and textual description of tagged images, we have used simple features that simply account for the presence/absence of textual entities. As already discussed in Section 3.4.1, the analysis of captions can be significantly refined. For tag prediction, stemming and explicit modelling of correlations between tags can be performed to improve annotation and retrieval performance. For instance, a tree-like model for semantic relations between words would allow very efficient inference and computations. Evidence for such semantic relations can also be gathered from external data sets such as WordNet to help improve this inference. There is a related promising research direction of learning semantic embeddings of images using text as supervision.

- Most metric learning algorithms scale quadratically in the number of training data points. TagProp is linear, but only after the neighbourhoods are computed,

which requires computing all pairwise distances. This quadratic scaling prevents large scale applications. It is interesting to apply approximate techniques developed for large scale image retrieval (*c.f.* Jégou et al. [2010]). These methods allow accurate nearest neighbour search in the order of a second in large collections of up to 10^7 images. Hopefully, the decrease of performance that can be expected from moving to approximate neighbours is compensated by the massive amount of data that can be used.

- Similarly, with larger data sets, weakly supervised methods are expected to perform all the better. It is of prime interest to determine the quantity of weakly labelled data that can match the performance of a given labelled training set. Using larger amounts of automatically labelled data could, in the end, outperform models that are currently trained on smaller but manually labelled data. For instance, using external weakly labelled data, there is a hope to outperform visual classifiers are learnt on the training set of *PASCAL VOC 2007*. The recent results of Perronnin et al. [2010], in which the authors learn classifiers on variable amounts of images gathered from Flickr groups, suggest that this is possible.

- Finally, many of the applications that we have discussed for photographs and associated text are also possible for videos that come with subtitles and scripts. This is especially interesting because manual annotations for videos are even more time-consuming to obtain than for images. Automatically detecting names in scripts or subtitles provide us with weak labels that can be associated with video sequences where several persons appear in tracks with temporal variations. Compared to our setting with photos, these tracks yield a large quantity of different appearances for a single person such that weak supervision and multiple instance learning frameworks have shown to cope very well (Yang et al. [2005], Sivic et al. [2009]). Even without any metadata supervision, simultaneous tracks can be used to sample many positive and negative pairs of face images to use as training for face recognition systems. We can therefore expect that more difficult tasks can now be considered using weak supervision in videos, such as spatio-temporal action localisation, human-object and human-human interaction modelling. Similarly, in images, segmentation (*c.f.* Vasconcelos et al. [2006], Verbeek and Triggs [2007]) and pose estimation are difficult tasks for which precise manual annotation is very expensive, making them primary targets for weakly supervised learning techniques.

A

Labelling cost

Definition

In this appendix, we describe our performance measure for the clustering task considered in Section 2.5.5, which we denote as the *labelling cost*. Previous work on measuring clustering quality and comparing clusterings include for instance Fowlkes and Mallows [1983], Meilă and Heckerman [1998], Meilă [2007]. Informally, we want our measure to reflect the labelling effort needed for a user to label the faces, *e.g.* for a personal photo album. We assume the user has two buttons: one to assign a single label to all data instances in a cluster, and one to assign a label to a single data instance. This setting is realistic, and is already used in practice in tools such as Picasaweb.[1]

With these two buttons, the most efficient way to label all instances in a cluster is to first label the cluster with the label of the most frequent class, and then to correct the errors. For a cluster c of n^c instances, the cost is:

$$\mathcal{L}(c) = 1 + n^c - \max_i n_i^c, \tag{A.1}$$

where n_i^c denotes the number of instances of class i in the cluster.

The cost to label all instances is then the sum of the costs to label the instances in each cluster. Let N be the total number of data points and C denote a clustering as a set of clusters, and $|C|$ is the number of clusters in C. The cost to label all instances is then the sum of the costs to label the instances in each cluster:

$$\mathcal{L}(C) = \sum_{c \in C} \mathcal{L}(c) \tag{A.2}$$

$$= |C| + N - \sum_{c \in C} \max_i n_i^c. \tag{A.3}$$

[1] URL: `http://picasaweb.google.com/`

Therefore, our labelling cost is related to the Meilă-Heckerman criterion $\mathcal{H}$ (see Meilă and Heckerman [1998]):

$$\mathcal{H}(C) = 1 - \frac{1}{N}\sum_c \max_i n_i^c. \tag{A.4}$$

Notably, like our criterion, it does not take into account the quality of clustering of the instances in minority in the clusters but only the most frequent class in each cluster. But our labelling cost takes into account the number of clusters: trivial clusterings (either one cluster, or as many clusters as data points) yield very high costs, while lower costs can be attained when the number of clusters is close to the number of classes.

We now formally derive the minimal and maximal labelling costs, as a function of the number of clusters. These bounds are interesting since the possible values of $\mathcal{L}(C)$ do not span the entire range of $[0,N]$, and the effective range notably depends on the number of clusters in C. The bounds, which consists in the labelling costs of the best possible and worst possible clusterings, will provide us with a clue about the relative quality of the clustering with those extremes.

To obtain the bounds, let us first order the class labels by their frequency in the data set: label 1 is the most frequent, label 2 the second-most frequent, and so on, up to label I who is the least frequent, where I is the number of classes. Let n_i be the number of images for label i, *i.e.* $n_1 \geq n_2 \cdots \geq n_I$. For mathematical convenience, we also define $n_i = 0$ for $i > I$.

Lower bound

Given a clustering size $|C|$, the lower bound, *i.e.* the best possible clustering for our measure, is found by maximising $\sum_c \max_i n_i^c$. Straightforwardly, we have:

$$\forall C, \quad \sum_{c\in C} \max_i n_i^c \leq \sum_{i=1}^{|C|} n_i, \tag{A.5}$$

which constitutes an attainable bound. Therefore, the minimum as a function of the clustering size is given by:

$$\underline{\mathcal{L}}(|C|) = |C| + N - \sum_{i=1}^{|C|} n_i. \tag{A.6}$$

To see this, consider $|C| \leq I$ clusters, and separate the $|C|$ most frequent class labels in individual clusters. Then place the remaining images, if any, in any cluster. Then

$\sum_{c\in C}\max_i n_i^c = \sum_{i=1}^{|C|} n_i$ which proves the result. Note that when $|C| = I$, $\underline{\mathcal{L}}(|C|) = \underline{\mathcal{L}}(I) = I + N - N = I$. The cost is exactly the number of clusters, and the number of different class labels: all clusters are pure. Let us call C^* this perfect clustering. Importantly, C^* attains the strict *global minimum* $|C^*| = I$ of the labelling cost, which is a desirable property.

When $|C| > I$, the (straightforward from Equation A.3) bound $|C|$ is reached by splitting pure clusters of C^* to obtain $|C|$ clusters. We use here the fact that $n_i = 0$ for $i > I$ to obtain the equality $|C| = |C| + N - \sum_{i=1}^{|C|} n_i$. Notice also that these lower bounds can be seen as a hierarchical clustering of a perfect similarity matrix.

Upper bound

For the maximum $\overline{\mathcal{L}}(|C|)$, we want to minimise $\sum_c \max_i n_i^c$. When $|C| \geq n_1$, we can actually have $\forall c, \max_i n_i^c = 1$, by simply ensuring that at most one image of any class label is in a cluster. This is possible because we have enough clusters, even for the most frequent label. This value is obviously a minimum since we do not allow for empty clusters. Thus

$$\forall C \text{ s.t. } |C| \geq n_1, \quad \overline{\mathcal{L}}(|C|) = N. \tag{A.7}$$

When $|C| < n_1$, note that we can at least achieve a labelling cost of $|C| + N - n_1$ by having label 1 as the most frequent label in each cluster (there might be other labels equally frequent in those clusters). We now show that a cost larger than that cannot be achieved. For this we look at clusterings with clusters where 1 is strictly not the most frequent label. Let c be such a cluster, and let $j > 1$ be the most frequent label in that cluster. Then there exists a cluster c' with $n_j^{c'} < n_1^{c'}$, because otherwise we contradict the fact that label 1 is the most frequent label overall. Now, we can increase the labelling cost by moving an element with label j from cluster c to cluster c', to see this note that all terms in the sum $\sum_c \max_i n_i^c$ stay constant except for c which is reduced by 1.

This finalises our proof that for a clustering of size $|C|$ the maximum label cost is given by

$$\overline{\mathcal{L}}(|C|) = \min\left(N,\ |C| + N - n_1\right) = N - \left[n_1 - |C|\right]_+. \tag{A.8}$$

In Figure 2.22, we show these theoretical bounds together with the labelling costs of our clustering methods. This helps compare the methods and also appreciate their quality as compared to the best and worst possible clusterings.

B

Rapport de thèse

B.1 Introduction

Récemment, de grandes archives numériques et multimédia sont apparues. Ceci est le résultat de larges efforts de numérisation provenant de trois sources principales. La première source est la numérisation d'archives pour permettre la redistribution de contenu initialement analogique. Parmi les institutions qui ont recours à ce procédé, nous pouvons citer les chaînes de télévision, les principales entreprises de production cinématographique et les bibliothèques et archives publiques ou privées, qui souhaitent rendre accessible leurs données archivées à la consultation en ligne. Deuxièmement, les données numériques sont désormais produites directement par ces services. Par exemple, les média de communication ou les réalisateurs de films utilisent désormais des appareil photographiques et des caméra vidéo numériques qui capturent leurs travaux directement sous forme de signal numériques, évitant ainsi la perte de qualité résultant de la conversion de l'analogique vers le numérique. Ces travaux peuvent ainsi être directement publiés en ligne ou en utilisant des supports numériques tels que les DVD ou les disques Blue-ray. Enfin, avec l'essor des produits numériques de grande consommation et des sites web de partage de documents, le contenu numérique provenant des utilisateurs finaux a connu une croissance extrêmement rapide ces dernières années. Des milliards de documents numériques sont ainsi déjà disponibles sur des sites tels que Facebook, Dailymotion, Youtube, Picasa et Flickr.[1] Dans la figure Figure B.1, nous illustrons cette croissance en montrant le nombre d'images soumises sur Flickr entre avril 2006 et décembre 2009.[2] À compter de février 2010, le nombre total d'images sur Flickr excède 4 milliard.

En conséquence de cette croissance exponentielle, est apparu le besoin de développer des méthodes pour permettre d'accéder à ces archives d'une manière à la fois intui-

[1] Les URL respectives sont : `htt://www.facebook.com/`, `http://www.dailymotion.com`, `http://www.youtube.com`, `http://www.picasa.com/` et `http://www.flickr.com/`.

[2] Ces chiffres ont été obtenus à l'adresse `http://wiki.creativecommons.org/Metrics/License_statistics`.

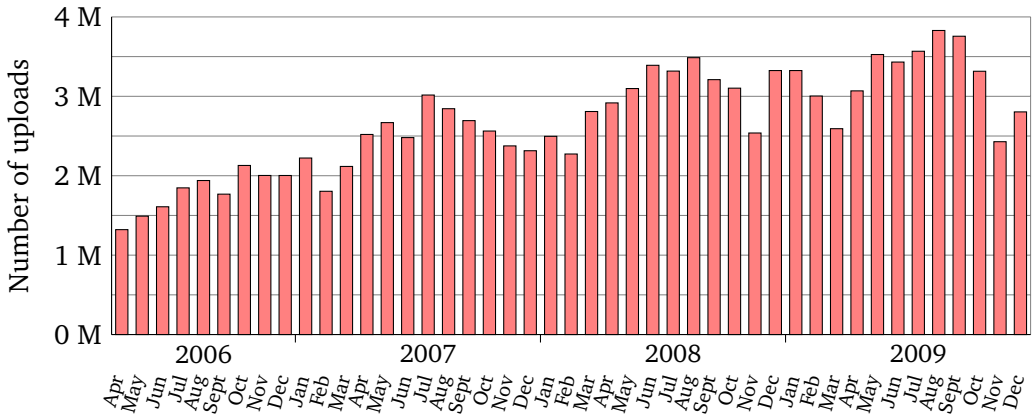


FIG. *B.1*: *Graphique montrant le nombre d'images sous licence Creative Common (CC) publiées sur Flickr entre avril 2006 et décembre 2009. La croissance régulière fluctue avec des pics annuels les mois d'été. Le nombre total d'images en CC sur Flickr dépasse désormais 135 millions et 4 milliards toutes licences confondues.*

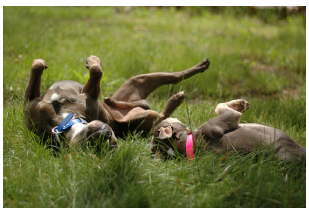
tive pour l'utilisateur et correcte sémantiquement. En effet, étant donnée la vitesse à laquelle de nouvelles données sont publiées, le coût que représente l'indexation manuelle est devenu prohibitif. Il y a actuellement un large effort de recherche (*c.f.* Jégou et al. [2008], Torralba et al. [2008], Fergus et al. [2009], Perronnin et al. [2010]) pour développer des méthodes automatiques pour indexer et chercher de larges bases de données d'images.

Dans le but d'indexer automatiquement les images d'archive avec l'objectif d'offrir aux utilisateurs un accès simple et efficace, il est nécessaire d'extraire automatiquement de l'information sémantique qui est pertinente pour les utilisateurs. Cela suppose de construire des systèmes qui sont capables de combler le fossé entre les caractéristiques bas-niveau et la sémantique visuelle (Smeulders et al. [2000]), c'est-à-dire le fossé entre les valeurs brutes des pixels et l'interprétation d'une scène visuelle qu'un humain est capable d'effectuer.

Pour illustrer ce fait, considérons un problème important pour la vision par ordinateur, à savoir la classification d'image. Le but de la classification est le suivant. Étant données quelques images, qui ne sont rien de plus qu'un tableau bidimensionnel de valeurs de pixel, le système doit être capable de décider si elles sont pertinentes pour un concept visuel donné, qui peut aller de la détection d'une instance d'objet à la reconnaissance de catégorie d'objet, en passant par la reconnaissance de formes générales. Nous illustrons la variété des concepts sémantique qui doivent être pris en charge dans la figure Figure B.2. Les compétitions PASCAL VOC (Everingham et al. [2007]) et ImageCLEF (Nowak and Dunker [2009]) sont de bons exemples du vaste intérêt de la communauté en ce problème.

Nuages, Plantes, Ciel, Arbre

Fleurs, Plantes



Animaux, Chien, Plantes

Voiture, Femelle, Mâle, Gens, Plantes, Structures, Transport

Bébé, Intérieur, Mâle, Gens

Nuages, Ciel, Structures, Coucher de soleil, Transport, Eau

FIG. *B.2*: *Illustration du problème de classification d'image rencontré en vision par ordinateur, issue de la base MIR Flickr : pour chacun des 24 concepts sémantiques visuels de la base, des systèmes sont construits pour décider automatiquement si les images sont pertinentes ou non. Les concepts manuellement annotés et utilisés pour l'évaluation des systèmes sont donnés sous les images.*

En parallèle, il est remarquable que l'énorme quantité de données visuelles actuellement disponible est de plus en plus fréquemment accompagnée d'informations supplémentaires. Par exemple, cette information supplémentaire peut consister de texte entourant l'image dans une page web, comme par exemple des données techniques sur Wikipedia : la Figure B.4 montre qu'il est techniquement possible d'extraire des informations de classification hiérarchique à partir de ce genre de pages. Nous trouvons aussi des labels d'utilisateurs sur les sites de partage de photos et de vidéos comme Youtube et Flickr. Ces labels, illustrés dans la Figure B.3, sont en général donnés par les utilisateurs pour l'indexation, ou pour fournir des informations complémentaires aux visiteurs (comme le modèle d'appareil photo utilisé, etc...). Enfin, des légendes accompagnent régulièrement les photos de presse trouvées sur les sites d'agrégation comme Google News ou Yahoo ! News. Souvent, ces légendes décrivent le contenu visuel des images, en mentionnant aussi l'évènement à l'origine de la publication de la photo, comme illustré dans la Figure B.5

La croissance de ces données *multimodales* est particulièrement observable sur le web, mais elle ne se limite pas à cette source : on les trouve aussi dans les vidéos accompagnées de sous-titres, scripts et pistes audio. Les *méta-données* textuelles, typiquement,

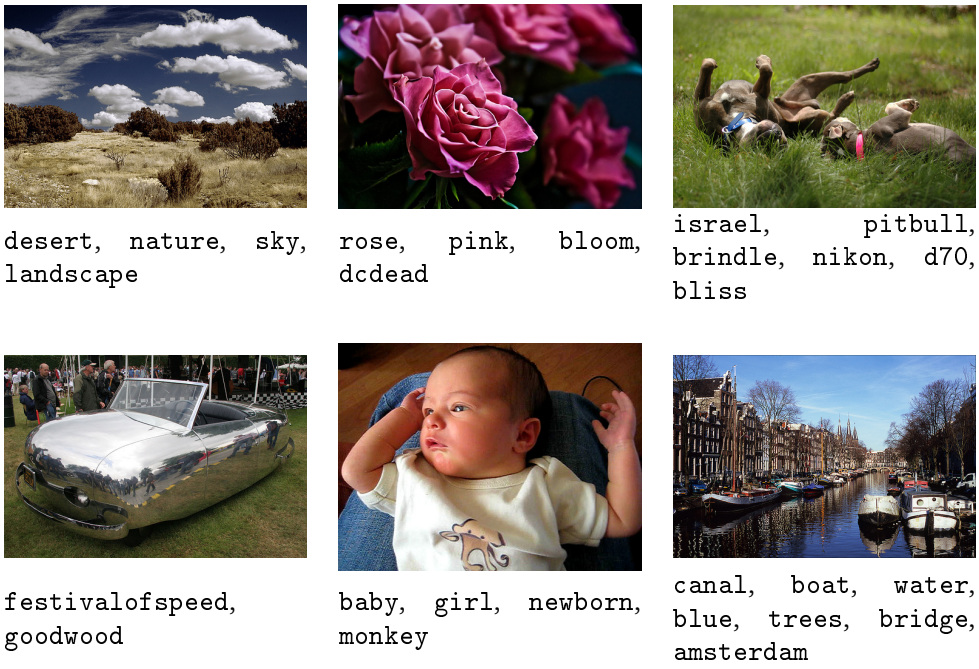


FIG. *B.3*: *Exemples de documents multimodaux formés d'images et de labels : Les images de la Figure B.2 avec quelques uns de leurs labels trouvés sur Flickr.*

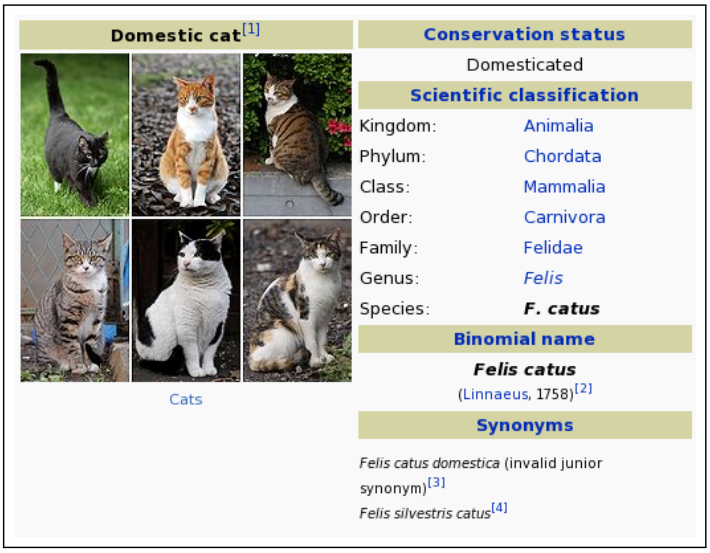


Figure B.4: *Exemples de documents multimodaux formés d'images et de textes: Un extrait d'une page Wikipedia sur les chats, montrant des images de chats et les informations de classification taxonomiques associées.*

In this March 23, 2010 file photo, House Speaker Nancy Pelosi of Calif. clutches a pen used by President Barack Obama to sign the health care bill, in the East Room of the White House in Washington. [...]
(AP Photo/J. Scott Applewhite, File)

FIG. *B.5*: *Exemples de documents multimodaux formés d'images et de légendes : Une photographie de presse et sa légende, publiée par l'agence Associated Press.*

décrivent de manière imparfaite et bruitée le contenu visuel. Cela motive notre intérêt pour l'utilisation de ces sources d'information pour aider la classification d'image et la reconnaissance d'objet en général, et la reconnaissance de visages en particuliers. Ci-dessous, nous décrivons en plus amples détails les problèmes que nous avons étudiés.

B.2 Objectifs

Dans cette thèse, nous nous concentrons sur l'analyse visuelle de photographies numériques, avec un intérêt particulier pour les humains et leurs identités. Reconnaître les humains et leurs actions possède évidemment beaucoup d'applications en surveillance. Mais cela peut aussi aider à l'organisation de collections de photos. Par exemple, de telles techniques peuvent être utilisées pour regrouper les images en fonction de l'identité des gens représentés. Il y a aussi un intérêt grandissant pour des systèmes de recherche d'image capables de retrouver les images de personnes semblables ou ressemblantes, ou d'une personne spécifique (*c.f.* Sivic et al. [2005a], Ozkan and Duygulu [2010]). L'émergence récente de telles fonctionnalités dans les logiciels Picasa et iPhoto soulignent le large intérêt pour de tels outils. Dans le second chapitre, nous construisons des systèmes de vérification de visages qui permettent d'atteindre de tels objectifs en étant capable de décider, pour deux images de visages, si elles représentent la même personne.

Pour obtenir de grandes quantités de données de visages, nous exploitons les photographies de presse, qui sont disponibles en très grand nombre. Comme déjà évoqué, ces images sont accompagnées légendes descriptives. En particulier, il est fréquent que les légendes mentionnent nommément les individus présentés visuellement, comme dans la Figure B.5. Dans le troisième chapitre, nous investiguons l'exploitation de ces légendes pour associer automatiquement les noms corrects aux visages trouvés dans la base. Nous adaptons aussi le système de vérification de visages, présenté ci-dessus, à ce type de données. De cette manière, l'organisation de photos vis-à-vis de l'identité des personnes représentées peut être obtenue sans aucune intervention humaine supplémentaire.

Cependant, l'organisation automatique de collections de photos pour l'indexation efficace et la recherche documentaire et, plus généralement, l'analyse d'image sont des problèmes qui vont au-delà du traitement particulier des visages. La recherche d'objets (*c.f.* Hirata and Kato [1993], Sivic and Zisserman [2003], Torralba et al. [2008]) ou de documents multilingues, multimodaux et complexes (*c.f.* Peters et al. [2008]) sont, par exemple, des problèmes bien plus complexes. Une des applications prometteuses de la vision par ordinateur pour la recherche documentaire est la possibilité de chercher des images dans une base en utilisant une requête formulée par une chaîne de caractères (comme c'est le cas de la recherche d'image sur Google Image Search). Les requêtes textuelles apportent des difficultés supplémentaires : les mots utilisés sont parfois ambigus, les utilisateurs peuvent être imprécis dans leur formulation de la requête, et bien entendu un lien doit être construit entre mots et images. Il est possible de voir ce problème comme de la traduction automatique entre texte et image (*c.f.* Duygulu et al. [2002]).

Établir de manière automatique quels mots-clés sont pertinents pour une image est un problème ouvert difficile pour la communauté, mais cela permettrait d'aider les utilisateurs à annoter et labelliser leurs images, tout en améliorant la pertinence de la recherche documentaire par mots-clés. Dans le chapitre 4, nous étudions l'utilisation de large bases de données d'images labellis es telles qu'illustrées dans la Figure B.3 pour l'annotation automatique d'image et la recherche d'image par mots-clés. En s'appuyant sur une similarité visuelle, il est possible de prédire la pertinence de labels pour une image cible, par exemple en regardant les labels des images de la base les plus similaires à la cible.

Enfin, nous montrons que les méta-données associées aux images sont utiles pour la catégorisation générique des images, en mesurant également la différence empirique de performance entre les différentes formes de supervision : l'approche complètement supervisée où des annotations manuelles sont requises, l'apprentissage semi-supervisé pour lequel l'ensemble d'apprentissage n'est que partiellement annoté, et enfin l'approche faiblement supervisée, lorsque les annotations sont imparfaites. Cette étude est l'objet du chapitre 5.

B.3 Contexte

Depuis son origine, la vision par ordinateur a eu pour but l'analyse, en particulier sémantique, des images. Pour comprendre une image dans sa globalité, il est intéressant, comme déjà mentionné, d'être capable de décider si un objet spécifique y apparaît. Cependant, cela n'est pas suffisant. Il est également nécessaire de localiser les objets, et expliquer les relations entre eux.

En regardant l'image dans la Figure B.6, nous pouvons prendre conscience qu'un haut niveau de connaissance est requis pour comprendre une image dans toute sa signification. Par exemple, les humains sont capables de récolter de nombreuses informations de cette image, dont certaines à propos de choses qui ne sont pas visibles sur la photo. D'abord, il y a trois humains sur une scène, qui jouent tous de la guitare. À partir de la position relative des guitares, nous pouvons dire que les trois hommes sont droitiers. Celui qui apparaît le plus avancé est certainement le leader du groupe. Nous sommes capables de localiser les guitares bien que leurs apparences et couleurs soient très différentes et que nous n'en avions probablement jamais vues d'exactement identiques, et nous voyons trois microphones dont on peut dire qu'un seul est utilisé. Cela se détermine par la proximité de celui-ci avec une bouche ouverte, et malgré l'absence de contact physique entre objets. Enfin, il n'est pas difficile de voir une batterie au fond malgré sa dissimulation presque complète, et d'imaginer un parterre de spectateurs en dehors de la photo.

Pour qu'un ordinateur obtienne une telle intelligence, il est d'abord nécessaire de pouvoir représenter ces connaissances. Par exemple, les modèles à constellation (Fischler and Elschlager [1973]) forment un cadre pionnier et influent pour représenter des objets complexes. Ces modèles combinent des relations spatiales vagues entre les parties des objets, c'est-à-dire en autorisant une certaine flexibilité dans leur disposition, avec des modèles d'apparence qui tentent de localiser les parties, comme illustré Figure B.7 (gauche). Par exemple, un visage est composé de deux yeux, deux oreilles, une bouche et un nez, avec un arrangement particulier. Similairement, un corps humain a une tête, un torse, deux bras, et deux jambes, etc, mais les variations de pose peuvent être plus importantes (*c.f.* Figure B.7, droite). Bien que fructueux, ces modèles s'avéraient souvent complexes et computationellement intensifs à exploiter.

Pour contourner temporairement le problème lié à la représentation de l'image, la communauté s'est d'abord attaquée à des problèmes plus simples, comme la reconnaissance de caractères manuscrits ou de visages. D'importants travaux en reconnaissance de visage sont apparus à la fin des années 1980 et au début des années 1990 (*c.f.* Turk and Pentland [1991]), en utilisant un environnement contrôlé : les objets sont bien centrés dans l'image, facilement identifiables par rapport au fond, et sans grande variabilité d'apparence, de pose, d'illumination, etc. En conséquence, les images elles-mêmes, ou des représentations schématiques, ont pu être utilisées

FIG. *B.6:* *Une compréhension globale de cette image requiert une connaissance avancée des poses, des actions et des comportements sociaux humains (par exemple, que les humains forment des groupes pour jouer de la musique sur scène), des apparences et usages des objets (les guitares peuvent avoir de multiples formes et couleurs, et sont manipulées d'une manière spécifique par les humains, tandis que les microphones interagissent, sémantiquement parlant, avec les bouches, et ce, sans contact physique, etc...). La modélisation et l'acquisition de ces connaissance est un des buts de la vision par ordinateur.*

FIG. *B.7:* *Illustration d'un modèle à constellation pour les caractéristiques faciales et l'estimation de pose humaine. À gauche, le schéma (dû à Fischler and Elschlager [1973]) illustre l'idée des modèles à constellation avec des contraintes élastiques entre les parties du modèle. À droite, en utilisant la méthode à parties de Eichner and Ferrari [2009], une relativement bonne estimation de la pose du haut du corps est obtenue malgré la pose inhabituelle observée.*

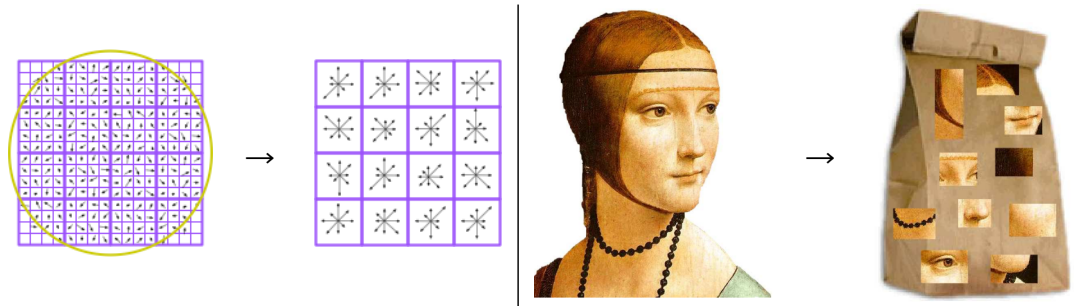


FIG. *B.8*: *Illustration d'un descripteur local d'apparence, SIFT (Lowe [1999]), à gauche, qui consiste en une grille d'histogrammes de gradients orientés. L'approche par sac-de-caractéristiques (image à droite, due à L. Fei-Fei), consiste à collecter les descripteurs locaux dans un ensemble non ordonné.*

pour représenter les images. Cependant, cette approche ne généralisait pas bien aux images possédant des fonds surchargés et aux objets partiellement dissimulés.

Le concept de description locale a donc été proposé pour décrire seulement des régions des objets. Ces descripteurs sont donc robustes au masquage des autres régions des objets et au surchargement du fond. Ils peuvent représenter soit la forme locale (par exemple, voir Belongie et al. [2002]), ou l'apparence locale (Schmid and Mohr [1997]) des objets. L'extraction de caractéristiques visuelles a progressivement été améliorée et celles-ci sont désormais fréquemment constituées d'histogrammes de gradients orientés, comme SIFT (Lowe [1999], voir Figure B.8, gauche), ou des réponses d'ondelettes de Haar (*c.f.* SURF, Bay et al. [2006]). Pour représenter une image ou une région d'une image, il fut proposé par Csurka et al. [2004] de collecter ces descripteurs locaux dans des ensembles non ordonnés appelés «sacs-de-caractéristiques», comme illustré dans la Figure B.8 (droite). Avec l'amélioration des caractéristiques visuelles locales, les modèles à constellation ont récemment suscité un net regain d'intérêt. Par exemple, Felzenszwalb et al. [2010] exploite de tels modèles pour localiser les objets dans des images en obtenant des résultats à l'état de l'art.

Avec l'élaboration de ces représentations complexes d'image et la mise à disposition de plus grandes bases d'images, il est apparu de plus en plus nécessaire d'analyser les données en utilisant des modèles mathématiques. Puisque ces modèles se sont avérés impossible à régler manuellement, les systèmes de vision faisant appel à l'*apprentissage machine* ont été mis au point pour généraliser la connaissance obtenue à partir d'exemples fournis par un humain. Ces méthodes sont qualifiées de «supervisées», car elles requiert la supervision d'un humain avant que tout apprentissage puisse avoir lieu.

Avec la proximité de l'approche par sac-de-caractéristiques par rapport à l'approche «sac-de-mot» issue du domaine de l'analyse de texte, un large effort a été consacré à l'adaptation des techniques utilisées dans ce domaine pour résoudre des problèmes similaires pour la vision. Par exemple, l'approche par sacs-de-mots s'est avérée très fructueuse pour la catégorisation de texte (par exemple, Joachims [1998]). Ainsi, un succès similaire a été obtenu pour la reconnaissance d'objet (Csurka et al. [2004]) en utilisant les mêmes outils d'apprentissage machine comme les machines à vecteurs support (SVM, Vapnik [1998]).

Le succès des méthodes supervisées a permis à la communauté de vision par ordinateur à résoudre des problèmes de plus en plus difficiles et de plus en plus haut niveau sémantique : de la classification de texture, la détection d'instances d'objet, la classification de catégories d'objet à la reconnaissance d'actions humaines dans les vidéos. Par exemple, les détecteurs de visage (*c.f.* Viola and Jones [2004]) et d'objets possédant une forte structure comme des piétons (Dalal and Triggs [2005]) sont désormais disponibles librement[3] et ont atteint les marchés de masse et les utilisateurs finaux.

Typiquement, la performance de ces systèmes augmente avec la quantité d'annotations manuelles fournies. Cependant, ce travail d'annotation devient prohibitif si de nombreux concepts visuels doivent être appris sur de grandes bases d'apprentissage. Il en résulte un intérêt croissant pour les méthodes qui dépendent moins de cette annotation manuelle. À l'extrême, les systèmes *non supervisés* (*c.f.* Hinton and Sejnowski [1999], Ghahramani [2004]) ne s'appuient sur aucune annotation manuelle pour analyser le contenu visuel, mais utilisent la structure ou la distribution des données pour identifier des groupes de données visuellement similaires (ou des modes dans la distribution). Par exemple, la détection de topic dans des documents textuels (Hofmann [1999]) et la découverte de catégories d'objet (*c.f.* Sivic et al. [2005b], Quelhas et al. [2007]) ont été considérées, en utilisant l'analyse sémantique latente probabiliste (PLSA, Hofmann [2001]), une méthode non supervisée. Nous pouvons noter que la famille de techniques d'apprentissage machine qui ont été utilisées pour les tâches liées aux visages ne diffère pas sensiblement de celle utilisée pour des problèmes plus génériques, tant dans le domaine visuel que textuel, voir par exemple Jain et al. [2007] qui utilise PLSA pour la détection de topics liés à des personnes.

Des approches intermédiaires existent entre la présence et l'absence totales de supervision. L'apprentissage «semi-supervisé» (*c.f.* Chapelle et al. [2006], Li et al. [2007], Fergus et al. [2009]) traite le cas où l'ensemble des données d'apprentissage n'est que partiellement annoté, tandis que l'apprentissage «faiblement supervisé» regroupe les méthodes qui utilisent des bases d'apprentissage qui sont annotées de manière imparfaite, soit par des humains, soit par des machines, par exemple en utilisant les fonctionnalités de recherche d'image de Google Image (Fergus et al. [2005]).

[3] OpenCV, la bibliothèque ouverte de vision par ordinateur : `http://willowgarage/`.

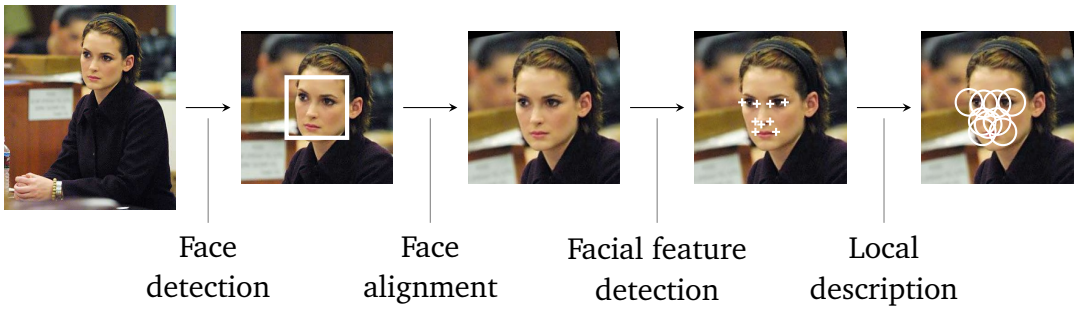


FIG. *B.9*: *Illustration du processus d'extraction de caractéristiques visuelles pour les visages dû à Everingham et al. [2006]. Après détection des visages dans les images, les visages sont visuellement alignés et traités par un détecteur de points-clés spécialisé aux visages. Ensuite, des descripteurs locaux sont extraits aux points localisés et concaténés pour former une description vectorielle des visages.*

Les méta-données que nous considérons dans cette thèse décrivent de manière faible et bruitée le contenu des images, ce qui en font des candidats particulièrement adaptés à l'apprentissage faiblement supervisé. Bien qu'imparfaites, ces annotations sont très intéressantes pour une raison majeure : elles peuvent être obtenues à très faible coût, et donc de large quantités de données peuvent être collectées. Cette immense quantité peut potentiellement compenser le désavantage à utiliser une supervision faible pour l'apprentissage. Il y a actuellement un vaste effort de recherche pour exploiter cette idée. Ces travaux s'attaquent à un large panel de problèmes de vision comme l'annotation d'image (Blei and Jordan [2003], Barnard et al. [2003]), l'agglomération de données (Bekkerman and Jeon [2007]) ou la classification de monuments touristiques (Li et al. [2009b]).

Nos travaux sont également encouragés par une tentative récente par Berg et al. [2004a] (voir aussi Pham et al. [2010]) de nommer automatiquement les visages détectés dans une base d'images de presse. Pour cela, les légendes associées aux images fournissent un ensemble de noms propres qui peuvent être associés aux visages. Citons également les travaux d'Everingham et al. [2006] qui visent à identifier automatiquement les personnages d'une vidéo. Le traitement typique des images pour la description des visages dans ces travaux est illustré dans la Figure B.9. D'autres tâches liées aux humains ont également été considérées, comme la reconnaissance du langage des signes dans les vidéos (Buehler et al. [2009]). Potentiellement, des données automatiquement annotées, qu'il s'agisse d'images ou de vidéos, peuvent être utilisées pour l'apprentissage de systèmes de reconnaissance faciale, mais ces annotations n'ont pas toujours le degré de qualité espéré, phénomène qui a pu être observé lors de tentatives de construction de bases d'images (*c.f.* Huang et al. [2007b]).

Dans ce contexte, nous pensons qu'il est possible de nous baser sur ces avancées récentes, qui concernent autant l'extraction de caractéristiques visuelles que l'apprentissage machine, pour atteindre nos objectifs d'amélioration de la reconnaissance faciale en environnement non contrôlé et d'utilisation de scénarii faiblement supervisés pour l'annotation d'image et la reconnaissance d'objets.

B.4 Contributions

Dans cette thèse, nos contributions sont les suivantes :

- Pour le problème de la vérification de visages en environnement non contrôlé, nous avons tout d'abord introduit un algorithme supervisé d'apprentissage de distance basé sur la régression logistique et que nous avons nommé LDML. L'idée sous-jacente est d'optimiser une métrique pour que les représentations de la même personne soient plus proches entre elles que deux représentations de personnes différentes. Nous proposons aussi de marginaliser la classification par plus proches voisins (kNN) pour permettre la classification de paires de données dont la classe est potentiellement inconnue. Cette méthode, la classification marginale par plus proches voisins (MkNN), combinée à l'apprentissage de distance pour définir les voisinages, améliore la performance de vérification. En utilisant les deux méthodes décrites ci-dessus, nous avons obtenu des résultats à l'état de l'art sur une base de donnée difficile d'images de visages en environnement non contrôlé. Nous avons également montré l'intérêt de nos méthodes pour les problèmes de reconnaissance à partir d'un seul exemple et d'agglomération de données. Ces travaux ont été publiés dans Guillaumin et al. [2009b], avec des améliorations en ce qui concerne la régularisation par contrainte de rang dans Guillaumin et al. [2010a], et sont décrits plus en détails dans le Chapitre 2. Cela permet d'envisager des applications de reconnaissance automatique de visages sur Internet, par exemple sur le réseau social Facebook.

- Pour le problème d'association entre noms extraits des légendes et les visages détectés dans les images, nous présentons une approche basée sur les graphes pour l'agglomération contrainte et la recherche de visages. Pour obtenir une solution approchée à ce problème NP-dur, nous proposons une forme d'optimisation locale où les contraintes locales sont exprimées comme un problème de couplage maximum dans un graphe bipartite, ce qui peut se résoudre exactement et efficacement. Nous évaluons notre approche sur une base d'environ 30000 images de presse qui ont été manuellement annotées par nos soins, et rendues disponibles en ligne. Ces travaux ont été publiés dans Guillaumin et al. [2008] en utilisant une métrique Euclidienne, puis améliorés dans Guillaumin

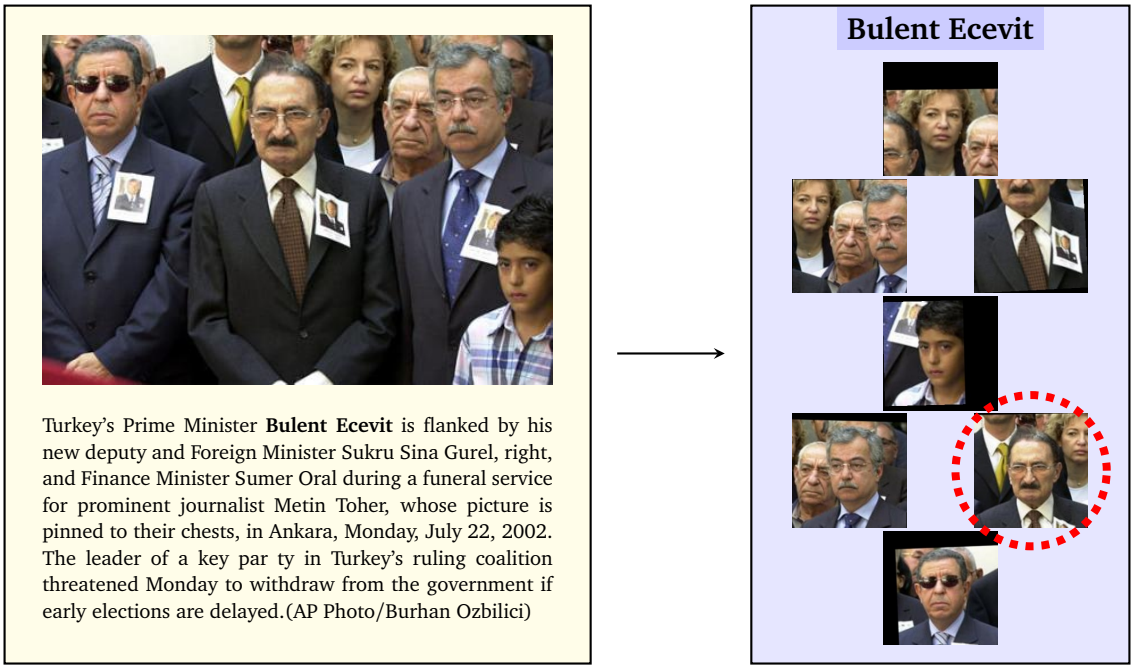


FIG. *B.10*: *Illustration du point de vue de l'apprentissage à instances multiples pour les dépêches de notre base telle qu'utilisée par notre modèle MildML d'apprentissage de distance. L'image est considérée comme un ensemble non ordonné d'images de visages, avec l'hypothèse qu'un moins un de ces visages est effectivement un exemple pertinent pour l'annotation du sac. Celle-ci est obtenue de manière automatique par application de techniques de traitement du langage naturel sur la légende. Dans cet exemple, le visage correct pour l'annotation est entourée en rouge.*

et al. [2010a] en utilisant des métriques apprises. Nous les présentons dans le Chapitre 3.

- Nous proposons également d'exploiter ces images de presse directement pour pour la reconnaissance faciale. À cette fin, nous proposons une formulation de l'apprentissage de distance (MildML) basée sur l'apprentissage à instances multiples (MIL), illustrée dans la Figure B.10, qui tente d'optimiser une distance sans explicitement associer les visages à des noms. Nous montrons que l'apprentissage de distances performantes est possible malgré l'absence d'intervention humaine, et que la formulation MIL est plus efficace que les métriques apprises à partir de visages annotés automatiquement. Comme attendu, les métriques MIL sont toutefois inférieures aux métriques supervisées. Ces travaux ont été publiés dans Guillaumin et al. [2010c] et présentés dans le Chapitre 3.

- Pour le problème de l'annotation automatique d'image, nous proposons d'utiliser les labels des images les plus similaires. Nous introduisons TagProp, pour

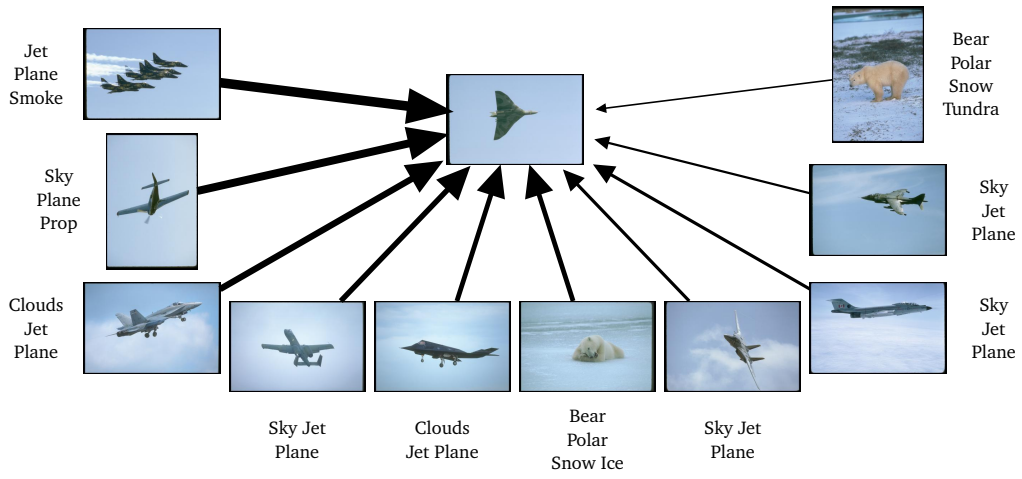


FIG. *B.11*: *Prédiction de labels par plus proches voisins pondérés comme effectué dans TagProp. La probabilité de pertinence du label pour une image peut être vue dans une somme pondérée de présence de ce label dans le voisinage de l'image. L'épaisseur des flèches représente la pondération du voisin, laquelle pondération dépend du score de similarité visuelle avec l'image cible.*

«Propagation de Tags», un nouveau modèle par plus proches voisins de prédiction de labels, comme illustré en Figure B.11. TagProp se démarque d'autres approches locales car il règle automatiquement la taille du voisinage à utiliser et optimise la combinaison linéaire de plusieurs représentations visuelles pour obtenir les voisins qui permettent la meilleure prédiction. Pour les deux tâches que sont l'annotation automatique d'image et la requête par mots-clés, nous obtenons des performances à l'état de l'art sur plusieurs bases d'images difficiles. Ces travaux ont été publiés dans Guillaumin et al. [2009a] et nous avons obtenu d'excellents résultats à la compétition ImageCLEF 2009 (*c.f.* Douze et al. [2009]). Le Chapitre 4 est dédié à ces travaux.

- Enfin, nous nous attaquons au problème plus général de la reconnaissance de catégories d'objets et la classification visuelle. En utilisant des images qui sont accompagnées de labels, nous étudions l'apprentissage de modèles de reconnaissance visuelle à partir de la supervision faible fournie par les labels. Nous considérons également un scénario semi-supervisé où nous supposons la disponibilité des labels au moment de l'apprentissage, mais avec seulement une partie des images annotée manuellement. Au moment du test, seules les images sont disponibles, c'est-à-dire que la classification doit se faire en l'absence de labels. Une représentation schématique de ce cadre de travail d'*apprentissage multimodal semi-supervisé* est fournie dans la Figure B.12. Pour s'attaquer à un tel scénario, l'idée est d'utiliser des classifieurs forts qui peuvent être appris sur les données multimodales d'apprentissage. Ces classifieurs peuvent être utilisés

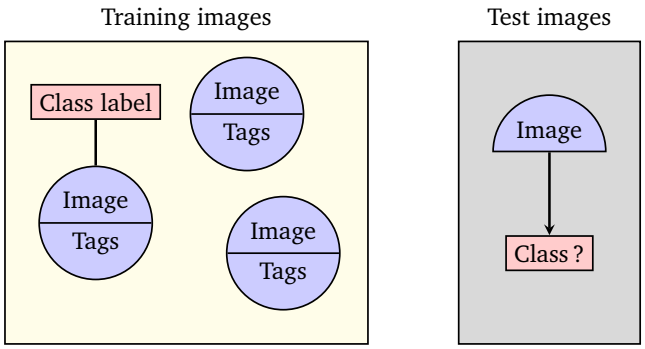


FIG. *B.12*: *Illustration du scénario d'apprentissage multimodal semi-supervisé que nous étudions au Chapitre 5. Les images d'apprentissage sont accompagnées de labels, dont seul un sous-ensemble est annoté manuellement. Le but est ensuite de prédire les catégories pertinentes pour des images de test qui sont dépourvues de labels.*

pour catégoriser les données d'apprentissage non annotées, et ainsi augmenter de manière plus fiable l'ensemble d'apprentissage pour un classifieur uniquement visuel. Plus précisément, nous utilisons un noyau visuel non linéaire pour régresser la fonction de prédiction obtenue par un classifieur à noyaux multiples (MKL, Lanckriet et al. [2004]) qui combine des noyaux visuels et textuels. Nous évaluons la performance de nos approches avec des quantités variables de supervision. Ces travaux, présentés dans la Chapitre 5, sont publiés dans Guillaumin et al. [2010b].

B.5 Perspectives

En nous appuyant sur nos travaux, nous proposons ci-dessous plusieurs pistes de recherche future, qui vont de l'amélioration de l'extraction de caractéristiques visuelles à des objectifs de recherche à long terme.

- Comme discuté dans le Chapitre 2, notre traitement d'image pour la description de visages ne prend pas en compte explicitement les variations de pose des visages. Puisque les changements de pose importants sont la cause principale d'échec de notre algorithme, il est naturel de considérer dans le futur des modèles qui incluent explicitement l'estimation de la pose pour reconnaître les visages. Il y a deux manières principales de traiter ce problème. D'abord, il est possible d'améliorer le processus d'alignement des visages. De cette manière, les descripteurs sont naturellement plus robustes aux changements de pose. Deuxièmement, cela peut être obtenu en prenant en charge l'absence d'observation des caractéristiques visuelles, ce qui est un problème classique lorsque

d'important changement de pose interviennent. Par la même occasion, ce type de méthode résoudrait le problème causé par les dissimulations partielles des visages, qui sont également une cause majeure d'échecs.

- Comme nous l'avons montré avec TagProp, l'intégration directe de l'apprentissage de distance dans les modèles est préférable à l'apprentissage de distance effectuée en utilisant une fonction objectif différente. Pour notre approche de classification par plus proches voisins marginalisée, nous avons utilisé des distances apprises pour la classification classique par plus proches voisins. Néanmoins, optimiser directement la distance pour la classification MkNN est possible. Une implémentation triviale mène à un algorithme dont la complexité croît en $O(N^4)$, N étant le nombre de données d'apprentissage, mais une dérivation plus efficace de l'algorithme est également possible. Pour améliorer davantage la complexité et les temps d'apprentissage et d'évaluation, les méthodes d'approximation de recherche de plus proches voisins peuvent être considérées.

- Plus généralement, il y a des pistes pour l'apprentissage de distance qui restent à explorer. Par exemple, il est aisé d'utiliser l'astuce du noyau pour généraliser l'apprentissage de distance au cas non linéaire, ou de projeter explicitement les données en plus haute dimension. Ces approches sont également prometteuses pour la prise en charge de différentes poses pour la reconnaissance de visages. De la même façon, nos expérimentations avec TagProp se sont limitées à optimiser une combinaison linéaire de distances de bases, or TagProp est capable, de manière plus générale, d'apprendre des distances directement, de Mahalanobis par exemple, et l'astuce du noyau peut aussi être utilisée avec TagProp pour utiliser des produits scalaires arbitraires. On peut d'ailleurs remarquer qu'une formulation de TagProp avec contraintes de rang permettrait de projeter les données dans un espace de faible dimension dans lequel la distance Euclidienne peut être utilisée pour la pondération des plus proches voisins. Cela résulte en une amélioration notable de la complexité de l'algorithme, et permet l'exploitation de techniques d'indexation adaptées à la métrique L2, qui forment en eux-mêmes un vaste sujet de recherche.

- Concernant notre travail sur l'annotation d'image, nous avons remarqué que TagProp opère une sélection des caractéristiques pendant l'apprentissage. Ainsi, les caractéristiques inusitées peuvent être oubliées au moment du test, ce qui accélère le processus de prédiction des labels. Cette sélection de caractéristiques, et plus généralement l'apprentissage de distance, permet donc de mettre au point des noyaux extrêmement performants pour la classification (*c.f.* Douze et al. [2009], Mensink et al. [2010]). TagProp s'avère donc une technique très prometteuse pour d'autres scenarii d'apprentissage et d'autres applications. Nous prévoyons de l'étendre par exemple à la classification multi-classe et la classification de région d'image. Également, un objectif à maximum de marge

tel qu'utilisé par Grangier and Bengio [2008] peut être utilisé pour optimiser les paramètres de TagProp explicitement pour les tâches de recherche d'image par mots-clés et de ré-arrangement. La classification multi-classe s'obtient facilement en remplaçant le modèle de Bernoulli multiple par une sortie multinomiale.

- Dans nos travaux, l'analyse des légendes et la description textuelle des images labellisées est constituées de caractéristiques simples qui dénotent la présence et l'absence de certaines entités textuelles. Nous l'avons déjà évoqué dans la section 3.4.1, l'analyse des légendes peut être significativement raffinée. Pour la prédiction de labels, la lemmatisation ou la modélisation explicite des corrélations entre mots peuvent être utilisées pour améliorer les performances d'annotation et de prédiction. Par exemple, un modèle à base d'arbre pour représenter les relations sémantiques entre mots autoriserait des calculs rapides pour l'inférence des paramètres. Des informations concernant ces relations sémantiques peuvent aussi être récupérées de données externes, comme WordNet, ce qui est susceptible d'améliorer l'inférence du modèle. L'apprentissage, en utilisant la supervision textuelle faible, de représentations sémantiques pour les images est une direction de recherche en lien avec cette idée, et représente également une piste très prometteuse de recherche.

- La plupart des algorithmes d'apprentissage de distance sont quadratiques en le nombre de données d'apprentissage. TagProp est linéaire, mais seulement après calcul des voisinages, ce qui implique de calculer toutes les distances deux-à-deux entre points de données. Cette croissance quadratique du temps de calcul empêche les applications à grande échelle. Il est intéressant de considérer l'utilisation de méthodes d'approximation déjà développées pour traiter de très grandes bases de données (*c.f.* Jégou et al. [2010]). Ces méthodes permettent de trouver de manière précise les plus proches voisins d'une image parmi environ 10^7 images dans un temps de l'ordre d'une seconde. Il est envisageable que la perte de performance occasionnée par l'utilisation de ces méthodes d'approximation puissent être compensées par l'immense quantité de données qui peut alors être utilisée.

- En utilisant des bases d'images de grande taille, les méthodes faiblement supervisées sont susceptibles d'être d'autant plus performantes. Il serait très intéressant de déterminer, à performance donnée, la quantité de données faiblement supervisées nécessaire à l'obtention de performances comparables à la supervision forte, ou de procéder à une comparaison à « coût d'annotation » fixé. L'utilisation de plus grandes quantités de données pourrait s'avérer préférable aux annotations manuelles sur de plus petites bases d'apprentissage. Par exemple, en utilisant des données additionnelles faiblement supervisées, nous pouvons espérer mettre au point des techniques plus performantes que les classifieurs vi-

suels classiques, par exemple sur la base *PASCAL VOC 2007*. Les résultats récent obtenus par Perronnin et al. [2010], qui entraîne des classifieurs en faisant varier la quantité d'images collectées de groupes Flickr, alimente les spéculations en ce sens.

- Enfin, de nombreuses applications considérées ou évoquées dans le cas des photographies associées à du texte sont également envisageable pour des vidéos accompagnées de sous-titres et scripts. Ceci est particulièrement intéressant car les annotations manuelles pour la vidéo sont encore plus coûteuses en temps que pour les images. La détection automatique de noms dans les scripts ou les sous-titres fournissent des annotations faibles qui sont associées à des séquences vidéos de plusieurs secondes où plusieurs personnes ont de multiples apparences au cours du temps. Si l'on compare à notre situation avec des images fixes, le suivi des personnes permet donc d'obtenir de grandes quantités d'informations supplémentaires sur les différentes apparences possibles d'une même personne. Des travaux récents montrent que l'apprentissage faiblement supervisé et l'apprentissage par instances multiples sont très bien adaptées à ce type de données (Yang et al. [2005], Sivic et al. [2009]). Même en l'absence totale de supervision, le suivi des personnes permet d'obtenir quantité de paires d'images que l'on peut considérer comme étant de la même personne (c'est le cas de paires d'images échantillonnées sur la même piste de suivi) ou n'étant pas de la même personne (pour des personnes suivies qui apparaissent simultanément dans au moins une image). Ces paires positives et négatives peuvent ensuite être utilisées pour l'apprentissage de systèmes de reconnaissance faciale. Nous pouvons donc nous attendre à ce que des problèmes de plus en plus difficiles soient traités par des approches faiblement supervisées dans les vidéos, comme la localisation spatio-temporelle d'actions, ou la modélisation des relations humain-objet et humain-humain. De la même manière, pour les images fixes, la segmentation (*c.f.* Vasconcelos et al. [2006], Verbeek and Triggs [2007]) et l'estimation de pose sont des tâches difficiles pour lesquelles l'annotation manuelle est très coûteuse, ce qui en fait des cibles de choix pour le développement futur de techniques faiblement supervisées.

Publications

This thesis has led to a submitted journal publication and several publications in major conferences. We summarize them below.

Journal articles

- M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid.

  Face recognition from caption-based supervision

  Submitted to International Journal of Computer Vision (IJCV), June 2010.

International conferences

- M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid.

  Automatic face naming with caption-based supervision.

  Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008.

- M. Guillaumin, J. Verbeek, C. Schmid.

  Is that you? Metric learning approaches for face identification.

  Proceedings of the IEEE International Conference on Computer Vision (ICCV), September 2009.

- M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid.

  TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation.

  Proceedings of the IEEE International Conference on Computer Vision (ICCV), September 2009. (Oral)

- M. Guillaumin, J. Verbeek, C. Schmid.

  Multimodal semi-supervised learning for image classification.

  Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2010. (Oral)

- M. Guillaumin, J. Verbeek, C. Schmid.

  Multiple instance metric learning from automatically labeled bags of faces.

  Proceedings of the European Conference on Computer Vision (ECCV), September 2010.

National conferences

- M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid.

  Apprentissage de distance pour l'annotation d'images par plus proches voisins.

  Reconnaissance des Formes et Intelligence Artificielle, January 2010.

Other publications

- M. Douze, M. Guillaumin, T. Mensink, C. Schmid, and J. Verbeek.

  INRIA-LEARs participation to ImageCLEF 2009.

  Working Notes for the CLEF 2009 Workshop, September 2009.

- J. Verbeek, M. Guillaumin, T. Mensink, C. Schmid.

  Image Annotation with TagProp on the MIRFLICKR set.

  Proceedings of the ACM International Conference on Multimedia Information Retrieval, March 2010. (Invited paper)

Bibliography

T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Proceedings of the European Conference on Computer Vision*, pages 469–481. Springer, 2004.

S. Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *Proceedings of the Conference on Neural Information Processing Systems*, 1998.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahanalobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, pages 404–417. Springer, 2006.

R. Bekkerman and J. Jeon. Multi-modal clustering for multimedia collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. URL `http://www.cs.umass.edu/~ronb/image_clustering.html`.

P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Proceedings of the European Conference on Computer Vision*, pages 43–58. Springer, 1996.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence*, pages 509–522, 2002.

T. Berg and D. Forsyth. Animals on the web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1463–1470, 2006.

T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 848–854, 2004a. URL http://csdl.computer.org/comp/proceedings/cvpr/2004/2158/02/215820848abs.htm.

T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces. Technical report, UC Berkeley, 2007.

T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth. Whos in the picture. In *NIPS*, 2004b. URL http://books.nips.cc/papers/files/nips17/NIPS2004_0603.pdf.

D. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

D. Bertsekas, M. Homer, D. Logan, and S. Patek. *Nonlinear programming*. Athena scientific, 1995.

M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceddings of the International Conference on Machine Learning*, page 11. ACM, 2004.

C. Bishop. *Pattern recognition and machine learning*. Spinger-Verlag, 2006.

D. Blei and M. Jordan. Modeling annotated data. In *Proceedings of the 26th annual intetational ACM SIGIR conference*, 2003.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998. URL citeseer.ist.psu.edu/blum98combining.html.

L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

M. Bressan, G. Csurka, Y. Hoppenot, and J.-M. Renders. Travel blog assistant system. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2008.

P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Z. Cao, Q. Yin, J. Sun, and X. Tang. Face recognition with learning-based descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

E. Chang, K. Goh, G. Sychay, and G. Wu. CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):26–38, 2003.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill, 2001.

M. Cox, S. Lucey, S. Sridharan, and J. Cohn. Least-squares congealing for large numbers of images. In *International Conference on Computer Vision (ICCV)*, August 2009.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:585, 2006.

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 1, page 22. Citeseer, 2004.

C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proceedings Internet imaging (SPIE)*, volume 5304, 2004.

N. Dalal and W. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. URL `http://lear.inrialpes.fr/pubs/2005/DT05`.

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In Z. Ghahramani, editor, *Proceddings of the International Conference on Machine Learning*, volume 227 of *ACM International Conference Proceeding Series*, pages 209–216. ACM, 2007. ISBN 978-1-59593-793-3. URL `http://doi.acm.org/10.1145/1273496.1273523`.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

K. Deschacht and M. Moens. Efficient hierarchical entity classification using conditional random fields. In *Proceedings of Workshop on Ontology Learning and Population*, 2006.

T. Dietterich, R. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

M. Douze, M. Guillaumin, T. Mensink, C. Schmid, and J. Verbeek. INRIA-LEARs participation to ImageCLEF 2009. In F. Borri, A. Nardi, and C. Peters, editors, *Working Notes for the CLEF 2009 Workshop*, sep 2009. URL `http://lear.inrialpes.fr/pubs/2009/DGMSV09`.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge Univ Pr, 1998.

P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision*, 2002.

M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, 2009.

P. Ekman and W. Friesen. *Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978.

M. Everingham, J. Sivic, and A. Zisserman. 'Hello! My name is... Buffy' - automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*, 2006.

M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). http://www.pascal-network.org/challenges/voc/voc2007/workshop/index.html, 2007.

P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision*, 61(1):55–79, 2005.

P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

A. Ferencz, E. Learned-Miller, and J. Malik. Learning to locate informative features for visual identification. *International Journal on Computer Vision*, 77:3–24, 2008.

R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *Proceedings of the European Conference on Computer Vision*, 2004.

R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 10, pages 1816–1823, 2005.

R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *Proceedings of the Conference on Neural Information Processing Systems*, 2009.

M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 100(22):67–92, 1973.

R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, pages 7: 179–188, 1936.

E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007. URL `http://dx.doi.org/10.1109/ICCV.2007.4408839`.

K. Fukunaga. *Introduction to statistical pattern recognition*. Boston Academic Press, 1990. URL `http://platon.serbi.ula.ve/librum/librum_ula/ver.php?ndoc=207234`.

K. Funes Mora. Robust face descriptors in uncontrolled settings. Master's thesis, Erasmus Mundus in Vision and Robotics, jun 2010.

A. Gaidon, M. Marszałek, and C. Schmid. Mining visual actions from movies. In *British Machine Vision Conference*, page 128, sep 2009. URL `http://lear.inrialpes.fr/pubs/2009/GMS09`.

A. Gallagher and T. Chen. Understanding images of groups of people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Y. Gao, J. Fan, X. Xue, and R. Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 901–910, New York, NY, USA, 2006. ACM. ISBN 1-59593-447-2. doi: http://doi.acm.org/10.1145/1180639.1180840.

E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, page 602. ACM, 2005.

Z. Ghahramani. Unsupervised learning. In O. Bousquet, U. von Luxburg, and G. Raetsch, editors, *Advanced Lectures in Machine Learning.*, number 3176 in Lecture Notes in Computer Science, pages 72–112. Berlin: Springer-Verlag, 2004.

A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS*, 2005. URL `http://books.nips.cc/papers/files/nips18/NIPS2005_0388.pdf`.

D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1371–1384, 2008.

D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *Proceedings of the European Conference on Machine Learning*, pages 162–173, 2006.

M. Grubinger. *Analysis and Evaluation of Visual Information Systems Performance*. PhD thesis, Victoria University, Melbourne, Australia, 2007.

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, jun 2008. doi: 10.1109/CVPR.2008.4587603. URL `http://lear.inrialpes.fr/pubs/2008/GMVS08/`.

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009a. URL `http://lear.inrialpes.fr/pubs/2009/GMVS09`.

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *Proceedings of the IEEE International Conference on Computer Vision*, sep 2009b. URL `http://lear.inrialpes.fr/pubs/2009/GVS09`.

M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face recognition from caption-based supervision. *International Journal on Computer Vision*, 2010a. Submitted.

M. Guillaumin, J. Verbeek, and C. Schmid. Multi-modal semi-supervised learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, jun 2010b.

M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the European Conference on Computer Vision*, sep 2010c.

J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. Unpublished, 1971. URL `http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf`.

J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM SIGGRAPH*, 2007.

X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using Laplacianfaces. *Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

G. Hinton and T. Sejnowski, editors. *Unsupervised learning: foundations of neural computation*. The MIT Press, 1999.

K. Hirata and T. Kato. Rough sketch-based image information retrieval. *NEC research & development*, 34(2):263–273, 1993.

T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999. URL `http://citeseer.ist.psu.edu/hofmann99probabilistic.html`.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

A. Holub, P. Moreels, and P. Perona. Unsupervised clustering for Google searches of celebrity images. In *IEEE Conference on Face and Gesture Recognition*, 2008.

G. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007a.

G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: a database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007b.

G. Huang, M. Jones, and E. Learned-Miller. LFW results using a combined Nowak plus MERL recognizer. In *Workshop on Faces Real-Life Images at ECCV*, 2008.

M. Huiskes and M. Lew. The MIR Flickr retrieval evaluation. In *ACM MIR*, 2008.

N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

V. Jain, A. Ferencz, and E. Learned-Miller. Discriminative training of hyper-feature models for object identification. In *Proceedings of the British Machine Vision Conference*, 2006.

V. Jain, E. Learned-Miller, and A. McCallum. People-LDA: Anchoring topics to people using face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision*, pages 304–317, 2008. URL `http://lear.inrialpes.fr/pubs/2008/JDS08`.

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. URL `http://lear.inrialpes.fr/pubs/2010/JDSP10`.

J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

R. Jin, S. Wang, and Z.-H. Zhou. Learning a distance metric from multi-instance multi-label data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.

T. Joachims. A support vector method for multivariate performance measures. In *Proceddings of the International Conference on Machine Learning*, page 384. ACM, 2005.

H. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

N. Kumar, P. N. Belhumeur, and S. K. Nayar. FaceTracer: A search engine for large collections of images with faces. In *Proceedings of the European Conference on Computer Vision*, pages 340–353, Oct 2008.

N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceddings of the International Conference on Machine Learning*, volume 18, pages 282–289, 2001.

G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5: 27–72, 2004.

I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proceedings of the Conference on Neural Information Processing Systems*, 2003.

S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 649–655, 2003.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

F.-F. Li, R. Fergus, and P. Perona. One-shot learning of object categories. *Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

J. Li and J. Wang. Real-time computerized annotation of pictures. *Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic object picture collection via incremental model learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, November 2009a.

Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009b.

C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009a.

J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, 42(2):218–228, 2009b.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. URL `http://computer.org/proceedings/iccv/0164/vol%202/01641150abs.htm`.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 7, pages 674–679. Morgan Kaufmann, San Mateo, CA, USA, 1981.

J. Luo, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Proceedings of the Conference on Neural Information Processing Systems*, pages 1168–1176, 2009.

A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *Proceedings of the European Conference on Computer Vision*, 2008. URL `http://www.cis.upenn.edu/~makadia/annotation/`.

T. Malisiewicz and A. Efros. Beyond categories: the visual memex model for reasoning about object relationships. In *Proceedings of the Conference on Neural Information Processing Systems*, 2009.

T. Mei, Y. Wang, X. Hua, S. Gong, and S. Li. Coherent image annotation by learning semantic distance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

M. Meilă. Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. URL `http://www.stat.washington.edu/mmp/Papers/compare-jmva-revised.ps`.

M. Meilă and D. Heckerman. An experimental comparison of several clustering and initialization methods. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, pages 386–395, 1998.

T. Mensink and J. Verbeek. Improving people search using query expansions: How friends help to find people. In *Proceedings of the European Conference on Computer Vision*, volume II of *LNCS*, pages 86–99. Springer, oct 2008. URL `http://lear.inrialpes.fr/pubs/2008/MV08`.

T. Mensink, G. Csurka, F. Perronnin, J. Sánchez, and J. Verbeek. LEAR and XRCE's participation to visual concept detection task - ImageCLEF 2010. In *Working Notes for the CLEF 2010 Workshop*, sep 2010. URL `http://lear.inrialpes.fr/pubs/2010/MCPSV10`.

D. Metzler and R. Manmatha. An inference network approach to image retrieval. In *ACM International Conference on Image and Video Retrieval*, 2004.

F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *ACM Multimedia*, pages 275–278, 2003.

F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: Constraining the latent space. In *ACM Multimedia*, 2004.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer, 1998.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, jun 2007. URL `http://lear.inrialpes.fr/pubs/2007/NJ07`.

S. Nowak and P. Dunker. Overview of the clef 2009 large scale-visual concept detection and annotation task. *CLEF working notes*, 2009.

T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal on Computer Vision*, 42(3):145–175, 2001.

D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1477–1482, 2006.

D. Ozkan and P. Duygulu. Interesting faces: A graph-based approach for finding people in news. *Pattern Recognition*, 43(5):1717–1735, May 2010.

J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *ACM SIGKDD*, 2004.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, page 2 (6): 559–572, 1901.

F. Perronnin, J. Sanchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. Oard, A. Peñas, V. Petras, and D. Santos. *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007*, volume 5152 of *Lecture Notes In Computer Science*. Springer Verlag, 2008.

P. Pham, M.-F. Moens, and T. Tuytelaars. Linking names and faces: Seeing the problem in different ways. In *Proceedings of ECCV Workshop on Faces in Real-Life Images*, 2008.

P. Pham, M. Moens, and T. Tuytelaars. Cross-media alignment of names and faces. *IEEE Transactions on Multimedia*, 12(1):13–27, 2010.

N. Pinto, J. J. DiCarlo, and D. D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

A. Qamar, E. Gaussier, J. Chevallet, and J. Lim. Similarity learning for nearest neighbor classification. In *Proceedings of the IEEE International Conference on Data Mining*, pages 983–988, 2008.

A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars. A thousand words in a scene. *Transactions on Pattern Analysis and Machine Intelligence*, 29: 1575–1589, 2007. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.1155.

L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 –286, feb 1989. ISSN 0018-9219. doi: 10.1109/5.18626.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

S. Satoh and T. Kanade. Name-It: association of face and name in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 368–373, 1997.

S. Satoh, Y. Nakamura, and T.Kanade. Name-it: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, 1999.

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997. URL `http://lear.inrialpes.fr/pubs/1997/SM97`.

F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge Univ Pr, 2004.

L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America A*, 4(3):519–524, 1987.

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *ACM International Conference on Image and Video Retrieval*, 2005a. URL `http://www.robots.ox.ac.uk/~vgg`.

J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the IEEE International Conference on Computer Vision*, 2005b.

J. Sivic, M. Everingham, and A. Zisserman. "Who are you?": Learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

R. K. Srihari. PICTION: A system that uses captions to label human faces in newspaper photographs. In A. Press, editor, *Proceedings of the AAAI-91*, pages 80–85, 1991.

Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proceedings of the British Machine Vision Conference*, Sept. 2009. URL `http://www.openu.ac.il/home/hassner/projects/multishot`.

Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

R. Tibshirani and G. Hinton. Coaching variables for regression and classification. *Statistics and Computing*, 8(1):25–33, 1998.

A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

M. Urschler, M. Storer, H. Bischof, J. Birchbauer, and S. Center. Robust facial component detection for face alignment applications. In *Proc. 33rd Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM)*, pages 61–72, 2009.

J. van de Weijer and C. Schmid. Coloring local feature extraction. In *Proceedings of the European Conference on Computer Vision*, 2006.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.

M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

M. Vasconcelos, N. Vasconcelos, and G. Carneiro. Weakly supervised top-down image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1001–1006, 2006.

J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

S. Vijayanarasimhan and K. Grauman. Multi-Level Active Prediction of Useful Image Annotations for Recognition. In *Proceedings of the Conference on Neural Information Processing Systems*, volume 21, 2008.

P. Viola and M. Jones. Robust real-time object detection. *International Journal on Computer Vision*, 57(2):137–154, 2004.

L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM SIGCHI*, 2004.

K. Wagstaff and S. Rogers. Constrained k-means clustering with background knowledge. In *Proceddings of the International Conference on Machine Learning*, pages 577–584, 2001.

F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008.

G. Wang, D. Hoiem, and D. Forsyth. Building text features for object image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of the Conference on Neural Information Processing Systems*, 2005. URL `http://books.nips.cc/papers/files/nips18/NIPS2005_0265.pdf`.

L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop on Faces Real-Life Images at ECCV*, October 2008. URL `http://www.openu.ac.il/home/hassner/projects/Patchlbp`.

L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Proceedings of the Asia Conference on Computer Vision*, 2009.

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Proceedings of the Conference on Neural Information Processing Systems*, pages 505–512. MIT Press, 2002. ISBN 0-262-02550-7. URL `http://books.nips.cc/papers/files/nips15/AA03.pdf`.

O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. In *Workshop on Multimedia Data Mining ACM SIGKDD*, 2008. doi: http://doi.acm.org/10.1145/1509212.1509213.

J. Yang, R. Yan, and A. Hauptmann. Multiple instance learning for labeling faces in broadcasting news video. In *ACM Multimedia*, 2005.

A. Yavlinsky, E. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *ACM International Conference on Image and Video Retrieval*, 2005. URL `www.edschofield.com/publications/yavlinsky05automated.pdf`.

H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2126–2136, 2006.

M. Zhao, J. Yagnik, H. Adam, and D. Bau. Large scale learning and recognition of faces in web videos. In *IEEE Conference on Face and Gesture Recognition*, 2008.

Z. Zhou and M. Zhang. Multi-instance multi-label learning with application to scene classification. In *Proceedings of the Conference on Neural Information Processing Systems*, 2007.

Exploiting Multimodal Data for Image Understanding

This dissertation delves into the use of textual metadata for image understanding. We seek to exploit this additional textual information as weak supervision to improve the learning of recognition models. There is a recent and growing interest for methods that exploit such data because they can potentially alleviate the need for manual annotation, which is a costly and time-consuming process.

We focus on two types of visual data with associated textual information. First, we exploit news images that come with descriptive captions to address several face related tasks, including *face verification*, which is the task of deciding whether two images depict the same individual, and *face naming*, the problem of associating faces in a data set to their correct names. Second, we consider data consisting of images with user tags. We explore models for automatically predicting tags for new images, *i.e. image auto-annotation*, which can also used for keyword-based image search. We also study a *multimodal semi-supervised learning* scenario for image categorisation. In this setting, the tags are assumed to be present in both labelled and unlabelled training data, while they are absent from the test data.

Our work builds on the observation that most of these tasks can be solved if perfectly adequate similarity measures are used. We therefore introduce novel approaches that involve metric learning, nearest neighbour models and graph-based methods to learn, from the visual and textual data, task-specific similarities. For faces, our similarities focus on the identities of the individuals while, for images, they address more general semantic visual concepts. Experimentally, our approaches achieve state-of-the-art results on several standard and challenging data sets. On both types of data, we clearly show that learning using additional textual information improves the performance of visual recognition systems.

Keywords

Face recognition • Face verification • Image auto-annotation • Keyword-based image retrieval • Object recognition • Metric learning • Nearest neighbour models • Constrained clustering • Multiple instance metric learning • Multimodal semi-supervised learning • Weakly supervised learning.

Données multimodales pour l'analyse d'image

La présente thèse s'intéresse à l'utilisation de méta-données textuelles pour l'analyse d'image. Nous cherchons à utiliser ces informations additionelles comme supervision faible pour l'apprentissage de modèles de reconnaissance visuelle. Nous avons observé un récent et grandissant intérêt pour les méthodes capables d'exploiter ce type de données car celles-ci peuvent potentiellement supprimer le besoin d'annotations manuelles, qui forment un processus coûteux en temps et en ressources.

Nous concentrons nos efforts sur deux types de données visuelles associées à des informations textuelles. Tout d'abord, nous utilisons des images de dépêches qui sont accompagnées de légendes descriptives pour s'attaquer à plusieurs problèmes liés à la reconnaissance de visages. Parmi ces problèmes, la *vérification de visages* est la tâche consistant à décider si deux images représentent la même personne, et le *nommage de visages* cherche à associer les visages d'une base de données à leur noms corrects. Ensuite, nous explorons des modèles pour prédire automatiquement les labels pertinents pour des images, un problème connu sous le nom d'*annotation automatique d'image*. Ces modèles peuvent aussi être utilisés pour effectuer des recherches d'images à partir de mots-clés. Nous étudions enfin un scénario d'*apprentissage multimodal semi-supervisé* pour la catégorisation d'image. Dans ce cadre de travail, les labels sont supposés présents pour les données d'apprentissage, qu'elles soient manuellement annotées ou non, et absentes des données de test.

Nos travaux se basent sur l'observation que la plupart de ces problèmes peuvent être résolus si des mesures de similarité parfaitement adaptées sont utilisées. Nous proposons donc de nouvelles approches qui combinent apprentissage de distance, modèles par plus proches voisins et méthodes par graphes pour apprendre, à partir de données visuelles et textuelles, des similarités visuelles spécifiques à chaque problème. Dans le cas des visages, nos similarités se concentrent sur l'identité des individus tandis que, pour les images, elles concernent des concepts sémantiques plus généraux. Expérimentalement, nos approches obtiennent des performances à l'état de l'art sur plusieurs bases de données complexes. Pour les deux types de données considérés, nous montrons clairement que l'apprentissage bénéficie de l'information textuelle supplémentaire résultant en l'amélioration de la performance des systèmes de reconnaissance visuelle.

Mots-clés

Reconnaissance de visage • Vérification de visages • Annotation automatique d'image • Recherche d'image par mots-clés • Reconnaissance d'objet • Apprentissage de distance • Modèles par plus proches voisins • Agglomération de données sous contrainte • Apprentissage de métrique par instances multiples • Apprentissage multimodal semi-supervisé • Apprentissage faiblement supervisé.