THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

présentée par **Armand Joulin**

pour obtenir le grade de
Docteur de l'École Normale Supérieure de Cachan

Domaine: **Mathématiques appliquées**

Sujet de la thèse:
Optimization convexe pour cosegmentation

—

Convex optimization for cosegmentation

Thèse présentée et soutenue à Cachan le 17 Decembre 2012 devant le jury composé de:

Francis BACH	Directeur de recherche, ENS/INRIA Paris	Directeur de thèse
Kristen GRAUMAN	Professeur, University of Texas	Rapporteur
Michael I. JORDAN	Professeur, University of California, Berkeley	Examinateur
Jean PONCE	Directeur de recherche, ENS Paris	Directeur de thèse
Cordelia SCHMID	Directeur de recherche, INRIA Grenoble	Examinateur
Dale SCHUURMANS	Professeur, University of Alberta	Rapporteur

Thèse préparée au sein des équipes SIERRA et WILLOW au département d'informatique de l'ENS (INRIA/ENS/CNRS UMR 8548)

RESUMÉ

Les hommes et la plupart des animaux ont une capacité naturelle à voir le monde et à le comprendre sans effort. La simplicité apparente avec laquelle un humain perçoit ce qui l'entoure suggère que le processus impliqué ne nécessite pas, dans une certaine mesure, un haut degré de réflexion. Cette observation suggère que notre perception visuelle du monde peut être simulée sur un ordinateur.

La vision par ordinateur est le domaine de la recherche consacré au problème de la création d'une forme de perception visuelle pour des ordinateurs. Les premiers travaux dans ce domaine remontent aux années cinquante, mais la puissance de calcul des ordinateurs de cette époque ne permettait pas de traiter et d'analyser les données visuelles nécessaires à l'elaboration d'une perception visuelle virtuelle. Ce n'est que récemment que la puissance de calcul et la capacité de stockage ont permis à ce domaine de vraiment émerger. Depuis maintenant deux décennies, la vision par ordinateur a permis de répondre à problèmes pratiques ou industrielles comme par exemple, la détection des visages, de personnes au comportement suspect dans une foule ou de défauts de fabrication dans des chaînes de production. En revanche, en ce qui concerne l'émergence d'une perception visuelle virtuelle non spécifique à une tâche donnée, peu de progrès ont été réalisés et la communauté est toujours confrontée à des problèmes fondamentaux. Un de ces problèmes est de segmenter une image ou une video en régions porteuses de sens, ou en d'autres termes, en objets ou actions.

La segmentation de scène est non seulement naturelle pour les humains, mais aussi essentielle pour comprendre pleinement son environnement. Malheureusement elle est aussi extrêmement difficile à reproduire sur un ordinateur. Une des raisons est qu'il n'existe pas de définition claire de ce qu'est une région "significative". En effet, en fonction de la scène ou de la situation, une région peut avoir des interprétations différentes. Par exemple, étant donnée une scène se passant dans la rue, on peut considérer que distinguer un piéton est important dans cette situation, par contre ses vêtements ne le semblent pas nécessairement. Si maintenant nous considérons une scène ayant lieu pendant un défilé de mode, un vêtement devient un élément important, donc une région significative. Dans cette thèse, nous nous concentrons sur ce problème de segmentation et nous l'abordons sous un angle particulier afin d'éviter cette difficulté fondamentale.

Nous allons considérer la segmentation comme un problème d'apprentissage faiblement supervisé, c'est-à-dire qu'au lieu de segmenter des images selon une certaine définition prédéfinie de régions "significatives", nous développons des méthodes permettant de segmenter simultanément un ensemble d'images en régions qui apparaissent régulièrement. En d'autres termes, nous définissons une région "significative"

d'un point de vue statistique: Ce sont les régions qui apparaissent régulièrement dans l'ensemble des images données. Pour cela nous concevons des modèles ayant une portée qui va au-delà de l'application à la vision. Notre approche prend ses racines dans l'apprentissage statistique, dont l'objectif est de concevoir des méthodes efficaces pour extraire et/ou apprendre des motifs récurrents dans des jeux de données. Ce domaine a récemment connu une forte popularité en raison de l'augmentation du nombre, de la taille des bases de données disponibles et la nécessité de traiter les données automatiquement.

Dans cette thèse, nous nous concentrons sur des méthodes conçues pour découvrir l'information "cachée" dans une base de données à partir d'annotations incomplètes ou inexistantes. Enfin, nos travaux prennent aussi racines dans le domaine de l'optimisation numérique afin d'élaborer des algorithmes efficaces et adaptés spécialement à nos problèmes. En particulier, nous utilisons et adaptons des outils récemment développés afin de relaxer des problèmes combinatoires complexes en des problèmes convexes pour lesquels il est garanti de trouver la solution optimale à l'aide de procedures developpees en optimisation convexe. Nous illustrons la qualité de nos formulations et algorithmes aussi sur des problèmes tirés de domaines autres que la vision par ordinateur. En particulier, nous montrons que nos travaux peuvent être utilisés dans la classification de texte et en biologie cellulaire.

ABSTRACT

People and most animals have a natural ability to see the world and understand it effortlessly. The apparent simplicity of this task for people suggests that this ability to understand our environment does not require, to some extent, high level thinking or profound reasoning about our surrounding. This observation suggests that this visual perception of the world should be reproducible on a mechanical device such as a computer.

Computer vision is the discipline dedicated to creating a form of visual perception on computers. The first work on computer vision dates from the 50's but the amount of power needed for treating and analyzing visual data was not available at that time. It is only recently that improvements in computer power and storage capacities, have permitted this field to really emerge.

On the one hand, constant progress in computer vision has allowed the development of dedicated solutions to practical or industrial problems. For example, detecting human faces, tracking people in crowded areas or detecting faults in production chains are some of the industrial applications where computer vision is now used. On the other hand, when it comes to creating a general visual perception for computers, it is probably fair to say that less progress has been made, and the community is still struggling with fundamental problems.

One of these problems is to reproduce our ability to group into meaningful regions, the visual input data recorded by an optical device. This procedure, called segmentation, separates a scene into meaningful entities (e.g., objects or actions). Segmentation seems not only natural but essential for people to fully understand a given scene, but it is still very challenging for a computer. One reason is the difficulty of clearly identify what "meaningful" should be, i.e., depending on the scene or the situation, a region may have different interpretations. Let us clarify this statement by a simple example: on the one hand, given a street scene, one may consider pedestrians as meaningful regions but not their clothes, on the other hand, given a fashion show, clothes may be considered as meaningful regions.

In this thesis, we will focus on the segmentation task and will try to avoid this fundamental difficulty by considering segmentation as a weakly supervised learning problem. Instead of segmenting images according to some predefined definition of "meaningful" regions, we develop methods to segment multiple images jointly into entities that repeatedly appear across the set of images. In other words, we define "meaningful" regions from a statistical point of view: they are regions that appears frequently in a dataset, and we design procedures to discover them. This leads us to design models whose a

scope goes beyond this application to vision. Our approach takes its roots in the field of machine learning, whose goal is to design efficient methods to retrieve and/or learn common patterns in data. The field of machine learning has also gained in popularity in the last decades due to the recent improvement in computer power, the ever growing size of databases and the ubiquitous necessity of automatic data processing.

In this thesis, we focus on methods tailored to retrieving hidden information from poorly annotated data, i.e., with incomplete or partial annotations. In particular, given a specific segmentation task defined by a set of images, we aim at segmenting the images and learn a related model as to segment unannotated images.

Finally, our research drives us to explore the field of numerical optimization so as to design algorithms especially tailored for our problems. In particular, many numerical problems considered in this thesis cannot be solved by off-the-shelf software because of the complexity of their formulation. We use and adapt recently developed tools to approximate problems by solvable ones. We illustrate the promise of our formulations and algorithms on other general applications in different fields beside computer vision. In particular, we show that our work may also be used in text classification and discovery of cell configurations.

We summarize the main contributions of this thesis below:

- The material of Chapter 2 is based on Joulin et al. (2010b): We propose a model for multiple image cosegmention with different instances of the same object class. We provide a convex semidefinite relaxation of this model and propose an efficient algorithm to solve it, based on convex optimization over manifolds. Experimentally, we show that our algorithm obtains good performances on classical cosegmentation problems and is also able to handle more complex segmentation problems.

  The limitation of the approach developed in this chapter is that it is not suited to multiclass problems and cannot be easily extended to other segmentation frameworks, such as interactive segmentation. These remarks encourages to develop a general framework for weakly supervised problems in the next chapter.

- The material of Chapter 3 is based on Joulin et al. (2010a): We propose a novel probabilistic interpretation of discriminative clustering with added benefits, such as fewer hyperparameters than previous approaches (Xu et al., 2005; Bach and Harchaoui, 2007). We provide a quadratic (non convex) local approximation of the log-likelihood of the parameters based on the EM auxiliary function. We design a low-rank optimization method for non-convex quadratic problems over a product of simplices. This method relies on a convex relaxation over completely

positive matrices. We perform experiments on text documents where we show that our inference technique outperforms existing supervised dimension reduction and clustering methods.

In this chapter, we show that our method works on tasks which are unrelated to segmentation. The next chapter naturally applies our framework to segmentation problems.

- The material of Chapter 4 is based on Joulin et al. (2012): we propose a simple and flexible energy-based formulation of true multi-class image cosegmentation that admits a probabilistic interpretation. We show that a convex quadratic approximation of our energy generalizes the cost function presented in the first chapter to the multi-class setting and affords a satisfactory initialization to the EM process. We develop an efficient algorithm that handles large numbers of input images and matches or improves the state of the art on two standard datasets. We also show that our framework can be easily extend to other segmentation problems such as interactive segmentation or weakly supervised segmentation.

  In this chapter, we use a loosened convex relaxation of our model for the initialization. Despite good performances in practice, it is not satisfactory from a theoretical point of view. In the next chapter, we propose a tight convex relaxation of the framework presented in Chapter 3.

- The material of Chapter 5 is based on Joulin and Bach (2012): We propose a full convex relaxation of the soft-max loss function with intercept, which can be applied to a large set of multiclass classification problems with any level of supervision. We also propose a novel convex cost function for weakly supervised and unsupervised problems and a dedicated and efficient optimization procedure.

We also investigate other directions that are not presented in this thesis but are related to problems studied here. In Hocking et al. (2011), we focus on hierarchical clustering, i.e., in producing a continuous set of labelling proposals which are ordered in a tree. We propose a convex formulation as well as an efficient procedure to obtain the whole set of labelling. In Duchenne et al. (2011), we focus on object classification using dense graph matching between pair of images. Dense graph matching is a way of finding correspondences between pair of images and has been used before to transfer segmentation labels from annotated images to unannotated images (Liu et al., 2009).

CONTENTS

Introduction

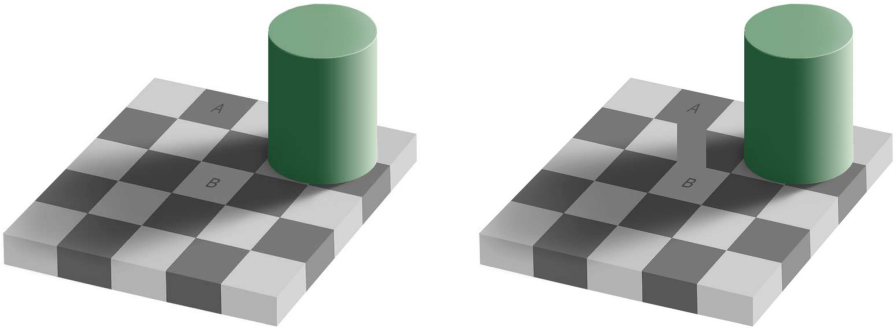
1.1 Vision

People and most animals have the natural ability to see the world and understand it with seemingly no effort. This *visual perception* as defined by Palmer (1999) is concerned with the acquisition of knowledge, which means that *vision* is a cognitive activity as opposed to a purely optical process such as photography. A camera or an eye acquire information about a scene, but do not know anything about it, whereas people and animals understand their environment. The knowledge achieved by visual perception is about *objects and situations in a given environment* and is obtained by *extracting information* from a given optical device (eye, camera, radio telescope...) which only captures the light emitted or reflected by objects (Palmer, 1999).

Computer vision, in some sense, is a field which aims to develop a visual perception of the world by an electronic device. Note that it does not necessarily imply that its goal is to duplicate the abilities of human vision, but rather to acquire, process and understand data taken from the real world through an electronic optical device in order to eventually produce an automatic decision by a computer.

In other words, computer vision tries to solve an *inverse problem*: Retrieve from an optical image of a scene the elements (e.g., objects or actions) which have originally created the scene (Palmer, 1999). More precisely, it disentangles data given by an optical device using some predefined models which may be based on geometry, physics or learning theory for example (Forsyth and Ponce, 2003).

For the rest of this thesis, we focus on optical images produced by devices such as digital photo and video cameras. We assume that the image produced by a digital camera is rectangular, spatially discrete with *pixels* as smallest unit. The signal recorded by an imaging sensor also consists of few values (the red, green and blue (RGB) intensities). We also assume that these devices do not give any explicit information about 3D as opposed to stereo cameras for example. In this particular setting, a typical optical input is thus simply a set of intensity values on a discrete grid.



(a) (b)

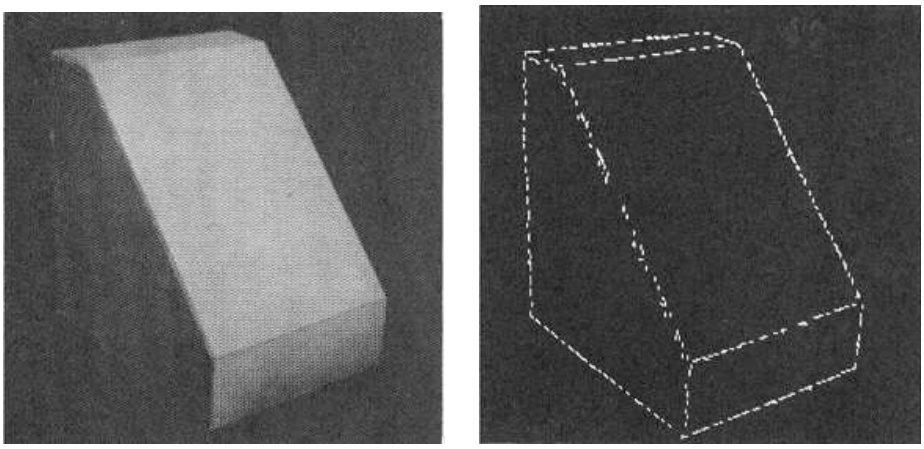
Figure 1.1: (a) Illusion created by Edward H. Adelson to illustrate the difficulty of illumination and color estimation[1]. (b) "Proof" of the illusion[1] (see text for details).

Transforming this rough input to some high-level representation of the world is challenging as a variety of problems arise on the way. We do not intend to give an exhaustive list of here but just give a few examples.

Illumination and color estimation. A color image is the result of the interaction between three different components: the sensors, the optical, geometric and physical properties of the scene, and the illumination sources. Even with calibrated sensors, separating the influence of the scene properties and the illumination is a difficult problem. A classical illusion introduced by Edward H. Adelson[1] to illustrate this problem is given in Figure 1.1. In this example, the pixels of squares "A" and "B" are clearly object with different illumination and color, "A" has a dark color and direct illumination whereas "B" has a light color and is in the shadow of the green object. However they have the same color intensity values in the image, as shown in panel (b) of Figure 1.1.

3-D scene understanding. Understanding a scene does not simply mean identifying the elements that have produced it. It also means capturing their orientation, size and shape, in other words, recovering the geometric scene layout. This fundamental problem in computer vision has been first studied by Roberts (1965) in the "blocks world" setting and further studied by Marr (1983) and many others. In Roberts' seminal work, the experiment was to understand a scene produced by simple, geometrical objects such as the ones shown in Figure 1.2. Even in this simple setting, clutter or occlusion would make the program fail. Nowadays, 3D scene understanding is still a challenging prob-

[1] http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html



(a) (b)

Figure 1.2: Illustration of Roberts' blocks world (Roberts, 1965): (a) Original image. (b) edge detection.

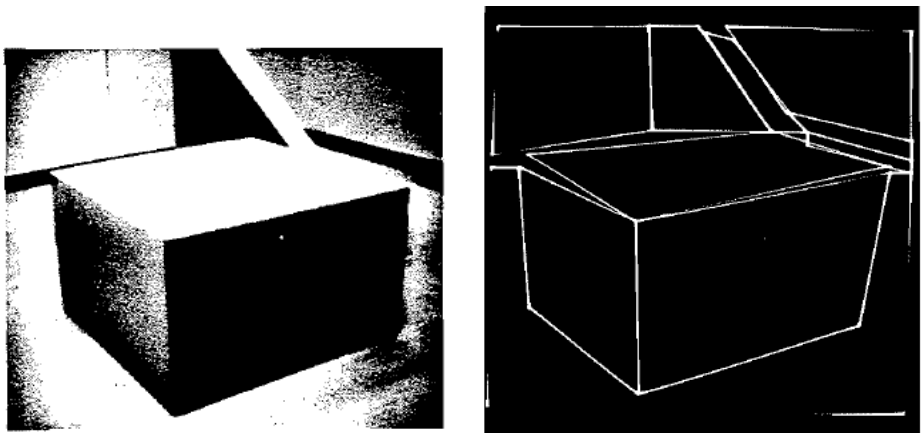
lem and most of the works are still restricted to special settings, for example street views (Hoiem et al., 2006; Gupta et al., 2010) or indoor scenes (Hedau et al., 2010).

Segmentation. Wertheimer (1923) pointed out the importance of perceptual grouping and organization in vision, i.e., some grouping of nearby pixels to form meaningful regions. In computer vision, the process of separating (in the simplest case) foreground from background is called *segmentation*. In the absence of 3D information, segmentation relies on pixel intensity values and a difficulty arises in the absence of edges between nearby objects or in the presence of strong edges in an object as shown by Figure 1.4. The first works on single image segmentation are from the early 70's (Brice and Fennema, 1970; Pavlidis, 1972). For example Brice and Fennema (1970) propose a segmentation algorithm for *blocks world* images, as shown by Figure 1.3.

In this thesis, we focus on an instance of this problem, dubbed "cosegmentation" where multiple images are to be segmented simultaneously and study it from a machine learning point of view. In the next section, we review different definitions of what is segmentation and their related approaches.

1.2 Segmentation

The objective of image segmentation is to divide a picture into P regions that are deemed meaningful according to some objective criterion, homogeneity in some feature space or separability in some other one for example.



(a) (b)

Figure 1.3: One of the first segmentation algorithms (Brice and Fennema, 1970): (a) Original image. (b) Regions retrieved from segmentation.

This definition relies on selecting an objective criterion and, depending on this criterion, the purpose of segmentation varies. For example, a natural criterion would be one that leads to segmenting the image into "objects" but defining what is an "object" is in itself not that easy. An "object" may refer to what is usually called a "thing" (a car, a cow, etc.) but might also be a texture (grass, rocks), or other "stuff" (a building, a forest) (Forsyth et al., 1996). An "object" may contain parts which are also "objects". The absence of a general objective criterion makes purely bottom-up segmentation an ill-posed problem. As illustrated by Figure 1.5, depending on what is perceived as an "object", segmentations obtained with different criteria may look reasonable.

The absence of an universal criterion has led to different definitions of segmentation in computer vision, and we review some of them below.

Bottom-up image segmentation. In this context, the objective criterion does not explicitly take into account the notion of meaningful object. The goal is thus usually to simply group nearby pixels according to local criteria (such as similar appearance) to produce image units more meaningful than pixels, often referred as *superpixels* or regions. In the 80's, Mumford and Shah (1985) propose a popular optimality criterion for segmenting an image into sub-regions. In the 90's, Shi and Malik (1997) propose another popular approach based on a graph-cut formulation. Figure 1.6 shows sample results obtained by these methods. Nowadays many other methods (Wright et al., 1997; Comaniciu and Meer, 2002; Felzenszwalb and Huttenlocher, 2004; Levinshtein et al., 2009) have been developed for that purpose. This type of segmentation is often referred to as *over-segmentation*, and has proven to be very useful to either simplify image repre-

(a) (b)

Figure 1.4: Illustration of segmentation difficulty. (a) Absence of edges between the woman face and the background. (b) Stronger edges inside the frog than between the frog and the background.

sentation (and thus reduce complexity for algorithms) or, to help "discover" meaningful objects (Russell et al., 2006; Wright et al., 1997; Cour and Shi, 2007; Malisiewicz and Efros, 2007; Lee and Grauman, 2010).

Top-down object segmentation. In this case, segmentation is not considered as a process that separates a whole image into meaningful regions, but as a process which extracts from an image a set of regions which may contain an object, possibly leaving unlabelled regions out. A good segmentation is thus a pool of regions which contains objects. Figure 1.8 illustrates this approach: Only regions which contains an object are segmented, living the “background” out. This approach relies heavily on the distinction between “stuff” and “things” briefly mentioned above. For clarity and completeness, let us give the original definition of this distinction as stated in Forsyth et al. (1996):

> The distinction between materials — “stuff” — and objects — “things” — is particularly important. A material is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape. An object has a specific size and shape.

Top-down object segmentation has been proposed to overcome the inherent limitations of bottom-up image approaches for segmenting “things”. The argument for this definition of segmentation is that only things have a “specific size and shape” and thus

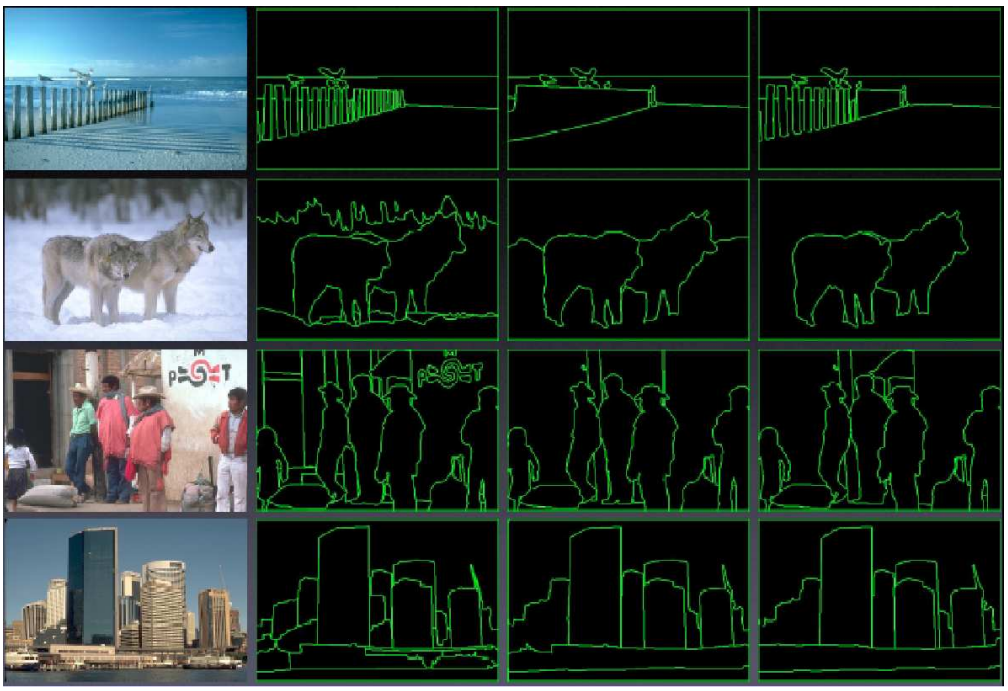
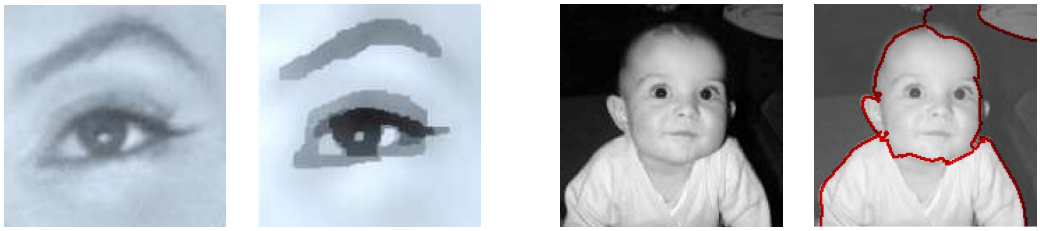


Figure 1.5: Examples showing why segmentation is ill-defined (from Olivier Faugeras' slides, "image segmentation, a historical and mathematical perspective") .



(a) (b) (c) (d)

Figure 1.6: Example of single image segmentation algorithms: (a-b) Mumford-Shah algorithm (Mumford and Shah, 1985). (c-d) Normalized cuts (Shi and Malik, 1997).

a clear boundary, as opposed to stuff. This approach thus considers a "objectness" criterion and produce a pool of segmentation based on this measure (Endres and Hoiem, 2010; Carreira and Sminchisescu, 2010; Alexe et al., 2010). Similarly, this idea has been apply to video to track moving objects (Lee et al., 2011). An interesting feature of this approach is that it aims to learn a single segmentation model independent of object

Figure 1.7: Distinction between things and stuff. See text for details.s

classes. For example, the segments shown Figure 1.8 are obtained by single model. In that sense, it is related to discovering saliency in an image, which is another active field in computer vision. On the other hand, the distinction between stuff and things is disputable which makes any "objectness" criterion imperfect. Figure 1.7 illustrates the difficulty of this distinction: In the first image, the tree in the middle is a thing if considered alone and stuff if considered with the surrounding forest. In the second image, the cars may be considered as stuff despite having "specific size and shape".

Figure 1.8: Example of top-down object segmentation (Carreira and Sminchisescu, 2010).

Interactive segmentation. In absence of an absolute criterion for segmentation, one may develop tools that allow a user to specify what he wants to segment. Such an *inter-*

active segmentation is popular in *computer graphics*, where the goal is to provide tools for users to perform some predefined tasks on optical data. Despite little difference in practice between computer vision and graphics, these two fields aim to solve fundamentally opposed problems. Computer vision is aimed at in creating a visual perception for electronic devices whereas computer graphics is aimed at creating tools to help people in specific visual tasks. As a consequence, interactive segmentation provides a setting for the user to give some information about what has to be segmented and the provide an answer based on pre-defined rules. To define these sets of rules, some implicit assumptions are made about the end goal of the segmentation: Most interactive segmentation methods assume that the user is interested in segmenting out a single region and this region is often assumed to be a "thing" which is distinctive from the rest of the image. Many frameworks have been proposed to allow the user to give some partial information about the regions he want to segment, but the most popular ones are drawing a bounding box around the object of interest (Blake et al., 2004) or drawing scribbles on the different regions of interest (Duchenne et al., 2008; Batra et al., 2010). Figure 1.9 shows examples of interactive segmentation. In practice, interactive segmentation works very well and has been implemented in commercial products.

(a) (b) (c) (d)

Figure 1.9: (a-b) Example of interactive segmentation with a bounding box (Blake et al., 2004). (c-d) Example of interactive segmentation with scribbles (Duchenne et al., 2008).

Multiple image segmentation. Another approach to segmentation consider multiple images associated with some given information and learn a segmentation model based on them. Despite relying on given information, this approach is fundamentally different from interactive segmentation as it aims to learn higher level representations based on the given set of images. These models can then be used to discover meaningful regions in new images. This approach is also different from top-down object segmentation in the sense that it aims at segmenting the whole image and not only producing possible "thing" candidates. This approach depends dramatically on the given information and usually, depending on the "quality" of the information, it can be roughly divided

in three subcategories: supervised segmentation, weakly supervised segmentation and cosegmentation.

Figure 1.10: Example of supervised segmentation (Krähenbühl and Koltun, 2011).

In supervised segmentation, each image is used to learn the model associated with a segmentation mask, i.e., some handmade segmentation of the image into meaningful regions. These regions are often labelled, i.e., each region is assigned to a particular visual category. A segmentation model for each of the labels is learned based on this information. An example is given in Figure 1.10. This approach gives good results in practice (Kohli et al., 2009; Gould et al., 2009; Liu et al., 2009; Tighe and Lazebnik, 2010; Krähenbühl and Koltun, 2011) but it requires ground truth for each category to learn. It also makes the assumption that any given region belongs to a pre-defined set of visual categories, and cannot handle categories that do not belong to this set.

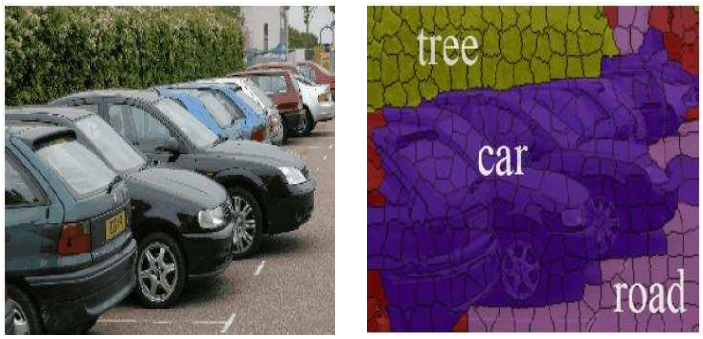


Figure 1.11: Example of weakly supervised segmentation (Vezhnevets et al., 2011).

Weakly-supervised segmentation aims to achieve the same goal as supervised segmentation but with a weaker form of information (Vezhnevets et al., 2011; Heess et al.,

2011). In this context, training images are associated with *tags*, i.e., the list of the objects in the image. Strong supervision with hand-labelled data is typically not available in this setting. The goal is thus to learn to simultaneously segment the images in regions representing the tags and learn a model. Figure 1.11 gives an example of a weakly supervised segmentation result: Given a set of tags (car, road, building), the algorithm is able to find the regions associated with these tags and learn a model that can then be used on images with no tags.

Finally, cosegmentation aims to simultaneously divide a set of images assumed to contain instances of P different object classes into regions corresponding to these classes. On the one hand, it can be seen as a special case of weakly supervised segmentation with tags present in all of the images. On the other hand, unlike weakly supervised segmentation, the model is not pre-trained on a predefined set of images but is learned on newly given images. It is thus applicable to more flexible settings and does not suffer from dataset bias (Torralba and Efros, 2011). Indeed, one may hope that cosegmentation methods could play a key role in the development of effective automated object discovery techniques and part-based approaches to object detection for example. For this reason, this field has been quite active recently: Early works on this

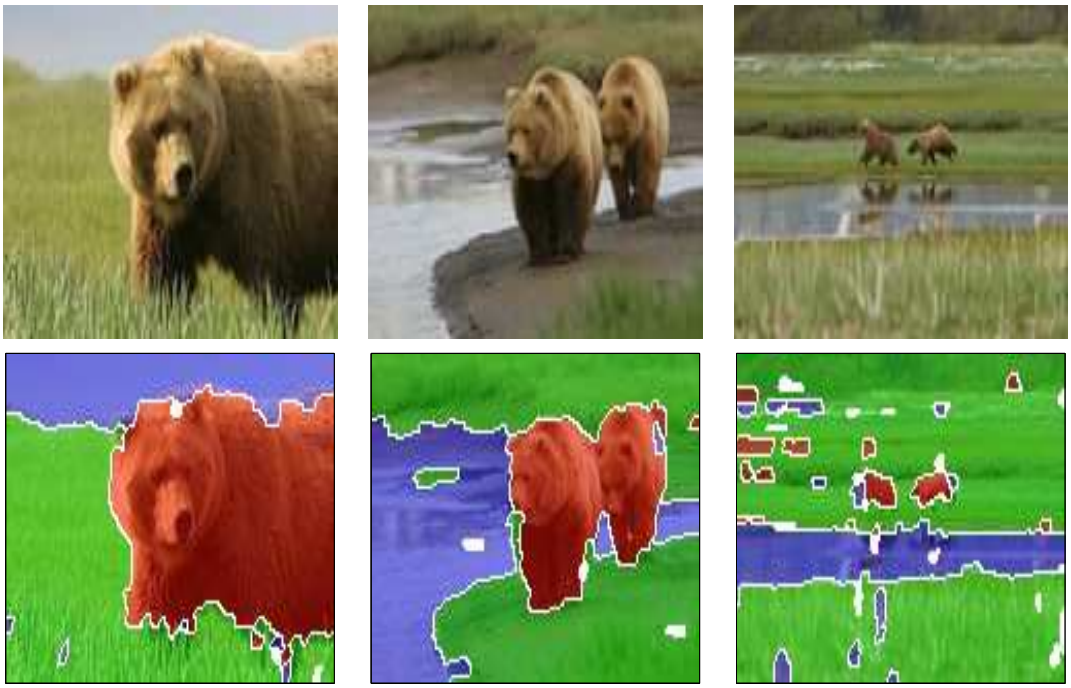


Figure 1.12: Example of cosegmentation (Joulin et al., 2012).

subject focus on a restricted setting involving only a pair of images containing the exact

same instance of an object (Rother et al., 2006; Hochbaum and Singh, 2009; Vicente et al., 2010). Several works (Joulin et al., 2010b; Lee and Grauman, 2010) have extended the original framework to multiple images and object categories. More recently, some works (Kim et al., 2011; Joulin et al., 2012) have proposed methods explicitly aimed at handling multiple object classes and images. Most of the work presented in this thesis is about cosegmentation with instances of the same class and not identical objects.

The distinction between these three forms of multiple image segmentation is not as strict as it appears, and some works have tried to combine different sources of information to learn a model (Rubinstein et al., 2012). In fact, even interactive segmentation also aims at segmenting an image giving a weak form of information. From a machine learning point of view, most segmentation approaches can be seen as solving different forms of weakly supervised problems. Some of the work presented in this thesis focuses on developing a general weakly supervised framework for image segmentation. In the next section, we define more precisely what this means.

1.3 Weakly supervised learning

Figure 1.13: Examples of labelling (see text for explanation).

In the previous section, we used the notions of supervision or weak supervision quite informaly. Let us now give more concrete definitions. In the context of machine learning, supervised learning is the task of inferring a function from labeled training data, i.e., a set of training examples. A training example consists of a pair of data, an *input object* (often referred as *feature* in this thesis) and an *output object*. In general, the output object can represent anything, it can be either discrete or continuous, a single value or a vector. However, in the scope of this thesis, we will focus on discrete, single-valued output objects, and will often refer to them as *labels* or *classes*. Supervised classifiers have proved to be very accurate tools for learning the relationship between input variables and certain labels. Usually, for these methods to work, the labeling of the training

data needs to be complete and precise. However, in many practical situations, this requirement is impossible to meet because of the challenges posed by the acquisition of detailed data annotations. This typically leads to partial or ambiguous labelings. This ambiguity in the labelling is in some sense related to the reason why segmentation is an ill-posed problem. Figure 1.13 illustrates some of the difficulties related to labelling a scene: A pixel may have different labels because of reflection (the mirror), superimposition of object (the wine in the glass), object definition (glass or container) or object parts (head or human).

Weakly supervised learning tries to solve this problem by considering that the observations are only associated with observable *partial* labels. Its implicit or explicit goal is thus to *jointly* estimate their *true (latent)* labels and learn a classifier based on these labels. In this thesis, we consider single-valued labels but, as suggested by the example shown Figure 1.13, multi valued latent labels are even more appropriated to the specific task of segmentation. Different weakly supervised methods have been proposed for different frameworks. We review some of the frameworks most related to segmentation:

Multiple instance learning. In the multiple instance learning (MIL) framework introduced by Dietterich and Lathrop (1997), *bags* of instances are labeled together instead of individually, and some instances belonging to the same bag may have different true labels. In the context of image segmentation, interactive segmentation with bounding boxes (Blake et al., 2004) or cosegmentation are special cases of multiple instance learning.

Semi-supervised learning. In the semi-supervised learning (SSL) framework (Chapelle et al., 2006), only a small number of points are labeled, and the goal is to use the unlabeled points to improve the performance of the classifier. In the context of image segmentation, interactive segmentation with scribbles is a form of semi-supervised learning where a small set of pixels (those given by the user) are labelled.

Unsupervised learning. Unsupervised learning, is an extreme case of weakly supervised learning where we do not possess any observable label information. Segmentation of a single image is a special case of unsupervised learning.

Other approach related to weak supervision. There are other approaches that are related to weak supervision. Some learn latent high dimensional representation for given tasks, such as neural networks, mixtures of experts (Jacobs et al., 1991) or discriminative restricted Boltzmann machines (Larochelle and Bengio, 2008). Others are related to supervised dimension reduction such as topic models (Blei et al., 2003; Blei and

Mcauliffe, 2008) or dictionary learning (Mairal et al., 2008).

In this thesis, in particular in Chapters 3 and 5, we develop a general framework for weakly supervised learning. Different modelling directions can be considered to design such a framework. In particular, one may either jointly model the input and output data, or model only the output given the input. These two directions are respectively called *generative* and *discriminative*. In the next section, we explain briefly the difference between them.

1.3.1 Discriminative versus generative models

Discriminative and generative models are fundamental concepts in machine learning.

Generative model. A generative model is a fully probabilistic model for randomly generating observable data. It supposes some distribution over the data. In the particular setting of learning, it specifies a joint probability distribution (or some energy function) over observation and label sequences. Classical examples are Gaussian mixture models, hidden Markov models and latent Dirichlet allocation (Blei et al., 2003).

Discriminative model. A discriminative model is a model specific to the supervised learning problem in the sense that it is defined only in presence of input and output data. It models only the conditional distribution (or some energy function) of the output data given the input. Classical models are logistic regression, neural networks (Fausett, 1994) and support vector machines (Cortes and Vapnik, 1995).

Decision boundary. In a supervised framework, the goal is to predict an output y based on some input x in $\mathcal{X}$. As stated before supervised learning aims to learn a function f of the input x such that $f(x)$ is a good predictor of the class y in the discrete space $\mathcal{Y}$. In other terms, this function assigns a label to a data point given the value of x and thus defines a "surface" that separate the classes in $\mathcal{X}$, i.e., depending on which side of the surface a point is, it has a different label. This surface is called the *decision boundary*.

An advantage of generative models over discriminative ones is that it is possible to sample new observations from them. Another advantage is that they are more flexible in expressing complex relation between input and output. For all these reasons, generative models are often considered more natural. On the other hand, for classification and regression tasks, discriminative models have shown better performance. The reason is that they are tailored to give the best possible output given the input, which makes them particularly suitable for decision problem such as classification. Another argument in

favor of discriminative models in the context of classification, is that these models focus on modeling only what is needed, i.e., the decision boundary, and thus, are more flexible than models that consider the whole space. In this thesis, we model segmentation as a classification problem and thus choose to use a discriminative model.

Complex weakly supervised models usually require learning the parameters of the model while infering some latent representation simultaneously. This usually leads to non-convex cost functions which are often optimized with a greedy method or a coordinate descent algorithm such as the expectation-maximization (EM) procedure. These methods are not guaranted to reach the best model parameters and the best labels simultaneously. This performance is often related to the quality of their initialization, i.e., some initial value of the parameters and the labels from which the algorithm will start. However, a special class of problems, called convex problems, does not possess this drawback, i.e., there exists an optimization procedure which is guaranteed to converge to its best configuration, regardless of the initialization. In this thesis, we explore a strategy called convex relaxation which aims at finding a convex problem related to an original non-convex one. In the next section, we give a brief introduction to convex relaxations as it is used in this thesis.

1.4 Convex relaxations

Given a non-convex problem, a convex relaxation is a convex problem closely related to the original one. In this section, we introduce the notion of convexity and then present a classical relaxation method in the context of a class of problems studied in this thesis.

1.4.1 Optimization

Optimization aims to find the best element of a given set according to some objective criterion. In this thesis, we restrict our attention to finding the minimum or the maximum value of a given function over a given set of points. In particular, given a set $\mathcal{X}$ and a cost function $f : \mathcal{X} \mapsto \mathbb{R}$, a minimization problem is defined as:

$$\min_{x \in \mathcal{X}} f(x). \tag{1.1}$$

Note that any maximization problem can be reformulated as a minimization problem:

$$\max_{x \in \mathcal{X}} f(x) = -\min_{x \in \mathcal{X}} -f(x).$$

In the rest of this thesis, we thus restrict our study to minimization problems without loss of generality.

Local and global minima. Given a function f, a point x is a local minimum over a topological set $\mathcal{X}$ if $f(x)$ is minimal over some neighborhood of x. A point x is a global minimum if its value $f(x)$ is minimal over the set $\mathcal{X}$.

The aim of minimization algorithm is to find a global minimum. A subclass of minimization problems, called convex problems, has the property that any local minimum is a global minimum, which makes them particularly interesting.

Convex problems. In the rest of this thesis, we suppose that $\mathcal{X}$ is in a vector space. We say that a set $\mathcal{X}$ is convex if Boyd and Vandenberghe (2003):

$$\forall (x, y) \in \mathcal{X}^2,\ \forall t \in [0, 1],\ tx + (1 - t)y \in \mathcal{X},$$

and that a real function $f : \mathcal{X} \mapsto \mathbb{R}$ is convex over the convex set $\mathcal{X}$ if:

$$\forall x, y \in \mathcal{X}^2,\ \forall t \in [0, 1],\ f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y).$$

Convex minimization studies the minimization of a convex function f over a convex set $\mathcal{X}$. A minimization problem is *convex* if both the space $\mathcal{X}$ and the function f are convex. Similarly, a problem is not convex if either the space $\mathcal{X}$ or the function f is not convex.

Non-convex optimization problems may possess local minima and thus, usually, any given optimization procedure has no guarantee to find the globally minimal value of the problem. The problems studied in the scope of this thesis are non-convex and one of the focuses of our work, is to relax them into convex problems. In the next section, we introduce the notion of relaxation.

1.4.2 Convex relaxations

A relaxation is an approximation of a difficult problem by a related problem that is simpler to solve. In particular, a convex relaxation replaces a non-convex problem by a convex one. Many convex relaxation schemes have been studies, such as linear programming for integer problems or Lagrangian relaxation and semidefinite positive relaxation for more complicated problems such as quadratically constrained quadratic programs (QCQP) (d'Aspremont and Boyd, 2003).

In this thesis we focus on *semidefinite positive* relaxations for a specific class of problem: non-convex quadratically constrained quadratic programs. The general form of a non-convex quadratically constrained quadratic program is:

$$\min_{x} \quad x^T P_0 x + q_0^T x + r_0 \tag{1.2}$$

$$\text{subject to} \quad x^T P_i x + q_i^T x + r_i \leq 0, \text{ for } i = 0, \ldots, m. \tag{1.3}$$

Nonconvex QCQPs are NP-hard (d'Aspremont and Boyd, 2003), and thus any algorithm designed to solve directly a QCQP has a complexity that is very likely to grow exponentially with the problem dimensions. Let us give some classical examples of QCQPs:

Minimum cardinality problems. The goal in this class of problem is to find a minimum cardinality solution over a set of linear inequalities:

$$\begin{aligned} \min_{x} \quad & \text{Card}(x) \\ \text{subject to} \quad & x \in \mathbb{R}^d \\ & b \preceq Ax. \end{aligned}$$

where $\text{Card}(x) = \{i \mid x_i \neq 0\}$.

Max-cut. Given a undirected graph G with N nodes, each pairs of nodes i and j are connected by an edge with a non-negative weight w_{ij}. If w_{ij} is equal to zero, the two nodes are not connected. The goal in max-cut problems is to partition the graph G in two sub-graphs G_{-1} and G_1 such that the sum of the weights along the *cut*, i.e., the set of couples of points (i,j) such that i is in G_{-1} and j is in G_i, is maximum (Goemans and Williamson, 1995). More precisely, the goal is to maximize the following quantity:

$$\sum_{i \in G_{-1},\ j \in G_1} w_{ij}.$$

Denoting by x_i in $\{-1, 1\}$, the assignment of the node i to either G_{-1} or G_1, the set of edges of the cut are defined by the set of pairs of nodes i, j such that $x_i x_j = -1$. Denoting by W the $N \times N$ symetric matrix with entries $W_{ij} = -w_{ij}$ if $i \neq j$ and $W_{ii} = \sum_j w_{ij}$, the max-cut problem is equivalent to:

$$\begin{aligned} \min_{x} \quad & x^T W x \\ \text{subject to} \quad & x^2 = 1, \end{aligned}$$

since $x \in \{-1, 1\}$ is equivalent to $x \in \mathbb{R}$ and $x^2 = 1$. Note that W is positive semidefinite. In this thesis, we are particularly interested in an extension of max-cut to partition of a graph in $k \geq 2$ sub-graphs called max-k-cut (Frieze and Jerrum, 1997). In that case, x_i is k dimensional vector such that $x_{ip} = 1$ if i is in the p-th sub-graph and 0 otherwise. The problem is then:

$$\begin{aligned} \min_{x} \quad & \text{tr}(x^T W x) \\ \text{subject to} \quad & x \geq 0, \\ & x^T x = 1_N, \\ & x^T 1_k = 1_N, \end{aligned}$$

where 1_k is the k dimensional vector with entries equal to 1. The constraints $x^T x = 1_N$ and $x^T 1_k = 1_N$ guarentees that for each n, only one x_{np} is equal to 1 and the others to 0.

Partitioning problems. The previous example partitions a graph given the weights of its edges. A more general problem is to consider any set of N points and partitioning them according to some $N \times N$ positive semidefinite matrix A representing some relations between the points. Among QCQPs, partitioning problems are those of the most importance for this thesis. Indeed, all along this thesis, we will design semidefinite matrix A for partitioning problems. A partitioning problem van be defined by:

$$\begin{aligned} \min \quad & \operatorname{tr}\left(x^T A x\right) \\ \text{subject to} \quad & x \geq 0, \\ & x^T x = 1_N, \\ & x^T 1_k = 1_N. \end{aligned}$$

In the next section, we introduce semidefinite relaxation in the particular context of QCQPs.

1.4.2.1 Semidefinite relaxation

As stated by Luo et al. (2010), even if a semidefinite relaxation can be found in the early work of Lovász (1979), it is Goemans and Williamson (1995) who have shown the interest of this strategy for QCQPs. In particular, they have shown that in the case of max-cut problems, such a strategy leads to solutions which are at least 0.8756 times the optimal value of the original NP-hard problem.

Semidefinite relaxation has been used in a wide range of applications, such as sensor network localization (Biswas and Ye, 2004), multiple channel access methods for communication technologies (Tan and Rasmussen, 2006), or aircraft conflict detection (Frazzoli et al., 1999). In the field of machine learning, semidefinite relaxation has been successfully applied to different problems: Lanckriet et al. (2004) have studied this type of relaxation to learn data driven kernel matrix. d'Aspremont et al. (2007) has proposed a convex formulation for the problem of dimensionality reduction. Srebro et al. (2005) has also used semidefinite programming for collaborative prediction. More related to the subjects studied in this thesis, Xu et al. (2005) have proposed a semidefinite convex relaxation for discriminative clustering based on finding maximum margin hyperplanes through data. In particular this work is related to the material presented in Chapter 2 of this thesis. Finally, Guo and Schuurmans (2008) have studied this type of relaxation in the context of models with latent variables (e.g., neural networks or mixture of experts). This work has been very influential for the material presented in Chapter 5 of this thesis.

Starting from a general non-convex QCQP as defined in Eq. (1.2), the first step of a semidefinite relaxation is to rewrite the problem, using the identity $\operatorname{tr}(x^T P x) =$

$\text{tr}(Pxx^T)$:

$$\begin{array}{rl} \min & \text{tr}(P_0X) + q_0^Tx + r_0 \\ \text{subject to} & \text{tr}(P_iX) + q_i^Tx + r_i \leq 0, \text{ for } i = 0, \ldots, m, \\ & X = xx^T. \end{array}$$

This problem is as difficult as the original one, but its formulation isoltes the source of non-convexity, i.e., the quadratic equality constraint $X = xx^T$. This equality can be replaced by the two constraints: $X \succeq xx^T$ and rank$(X)= 1$ (Luo et al., 2010). The semidefinite relaxation of a QCQP is thus simply to remove the rank constraint and only keep the inequality, $X \succeq xx^T$ (d'Aspremont and Boyd, 2003). Finally, using a Schur complement (Boyd and Vandenberghe, 2003, or see the appendix), the semidefinite relaxation of a general QCQP is the semidefinite program:

$$\begin{array}{rl} \min & \text{tr}(P_0X) + q_0^Tx + r_0 \\ \text{subject to} & \text{tr}(P_iX) + q_i^Tx + r_i \leq 0, \text{ for } i = 0, \ldots, m, \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{array}$$

General purpose toolboxes for solving semidefinite programs (SDP) have been developed in the past (Grant and Boyd, 2010). In the cases studied in this thesis, they are usually computationally inefficient, which has led us to develop optimization schemes tailored for our problems. In turn, this has led us to explore different aspects of convex optimization and use a variety of tools. For example, in the first chapter, we use tools from optimization on manifolds (Journée et al., 2010) and, in the last chapter, we investigate accelerated proximal methods (Beck and Teboulle, 2009).

1.5 Notations

In this thesis, we suppose that instances of a given dataset are grouped in *bags*. We suppose that we observe I bags. For i in $\{1, \ldots, I\}$, $\mathcal{N}_i$ is the set of instances in the i-th bag, and $N_i = |\mathcal{N}_i|$ is its cardinality. We denote by $N = \sum_i N_i$ the total number of instances. In the context of cosegmentation, a bag is an image and the set of instances $\mathcal{N}_i$ is a (coarse) grid of N_i pixels sample on the image.

In each bag i, an instance n in $\mathcal{N}_i$ is associated with a feature vector $x_n \in \mathbb{R}^d$ and a partially observable label z_n in a given set $\mathcal{L}$, in certain feature and label spaces.

In this thesis, we are interested in finding a *latent* label y_n in a given set $\mathcal{P}$. This latent label is supposed to give a better understanding of the data. In the particular context

of cosegmentation, this label gives the segmentation of the images.

We denote by P and L the cardinalities of $\mathcal{P}$ and $\mathcal{L}$. The variables z_n and y_n are associated with their canonical vectorial representation, i.e., $y_{np} = 1$ if the instance n has a latent label of p and 0 otherwise. We denote by y the $N \times P$ matrix with rows y_n.

We usually denote by K a positive definite kernel defined on our N d-dimensional vectors x_j, $j = 1, \ldots, N$ and by $\Phi : \mathcal{X} \mapsto \mathcal{F}$ the associated mapping into a high-dimensional Hilbert space $\mathcal{F}$, so that $K_{ml} = \Phi(x_m)^T \Phi(x_l)$ (Shawe-Taylor and Cristianini, 2004).

In the context of segmentation, the feature $x_n \in \mathcal{X}$ may be a SIFT vector or color histogram. This feature is used to discriminate among different object classes in different images.

We also associate with each pixel n its color $c_n \in \mathbb{R}^3$, its position $p_n \in \mathbb{R}^2$ within the corresponding image. These two features are used to encode the local spatial layout and appearance of each image.

2

Discriminative clustering for image co-segmentation

Abstract of this chapter: In this chapter, we combine existing tools for bottom-up image segmentation such as normalized cuts with kernel methods such as ridge regression commonly used in object recognition. These two sets of techniques are used within a discriminative clustering framework: the goal is to assign foreground/background labels jointly to all images, so that a supervised classifier trained with these labels leads to maximal separation of the two classes. In practice, we obtain a combinatorial optimization problem which is relaxed to a continuous convex optimization problem, that can itself be solved efficiently for up to a hundred of images.

The material of this chapter is based on the following work:

A. Joulin, F. Bach and J. Ponce. Discriminative Clustering for Image Co-segmentation. In *proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

2.1 Introduction

Co-segmentation is the problem of simultaneously dividing I images into regions (segments) corresponding to P different classes. When $I = 1$ and $P = 2$, this reduces to the classical bottom-up segmentation problem where an image is divided into *foreground* and *background* regions. Despite over 40 years of research, it is probably fair to say that there is still no reliable purely bottom-up single-image segmentation algorithm (Mumford and Shah, 1985; Felzenszwalb and Huttenlocher, 2004; Meyer, 2001; Shi and Malik, 1997). As explained in the introduction of this thesis, the situation is

different when a priori information is available, for example in a supervised or interactive setting where labelled samples are available for the foreground and background (or even additional, $P > 2$) classes (Boykov et al., 2001; Blake et al., 2004; Hochbaum and Singh, 2009). The idea of co-segmentation is that the availability of multiple images that contain instances of the same "object" classes makes up for the absence of detailed supervisory information.

Rother et al. (2006) first introduced this idea in the relatively simple setting where the same object lies in front of different backgrounds in a pair of images. At the same time, in the context of object recognition, where object instances may vary in pose, shape or color, co-segmentation should provide mid-level features which could improve recognition performance, (Russell et al., 2006; Winn and Jojic, 2005). In this chapter, our aim is to obtain a co-segmentation algorithm flexible enough to perform well in both instances, i.e., when foreground objects in several images are close to identical, and when they are not. The experiments presented in this chapter reflect this double objective. The framework we have chosen to use is based on *discriminative clustering*.

Discriminative clustering was first introduced by Xu et al. (2005) and relies explicitly on *supervised* classification techniques such as the support vector machine (SVM) to perform *unsupervised* clustering: it aims at assigning labels to the data so that if an SVM were run with these labels, the resulting classifier would separate the data with high margin. In order to solve the associated combinatorial optimization problem over labels, Xu et al. Xu et al. (2005) consider a convex relaxation in terms of a semidefinite program (SDP) (Boyd and Vandenberghe, 2003). Other discriminative clustering methods have been proposed based on different frameworks (De la Torre and Kanade, 2006; Bach and Harchaoui, 2007; Joulin et al., 2010a). In this chapter, we consider the least-squares classification framework of Bach and Harchaoui (2007), which also leads to a semidefinite program which can be solved by more efficient and flexible algorithms.

Discriminative clustering is well adapted to the co-segmentation problem for two reasons: first, we can re-use existing features for supervised classification or detection, in particular state-of-the-art architectures based on histograms of local features and kernel methods (Zhang et al., 2007a). Relying on supervised tools and previous research dedicated to fine-tuning these descriptors has proved to be advantageous in other weakly supervised tasks in computer vision (Duchenne et al., 2009; De la Torre and Kanade, 2006). Second, discriminative clustering easily allows the introduction of constraints into the partitions found by the clustering algorithm, in our case spatial and local color-consistency constraints.

In order to adapt discriminative clustering to the task of co-segmentation, we need to extend its original formulation (Bach and Harchaoui, 2007; Xu et al., 2005) in two directions: first, we include some local spatial consistency by incorporating a term based on a normalized Laplacian. This term is directly inspired by the spectral clustering framework of Shi and Malik (1997), as it has proven very well suited for the segmentation

problem. Second, we use recent techniques from the optimization literature (Journée et al., 2010) to find solutions of semidefinite programs over matrices representing more than tens of thousands of data points, which is necessary to co-segment up to hundred of images. There method takes full advantage of the special structure of QCQPs, in particular. of the manifold property of set of semidefinite positive matrices.

2.2 Problem formulation

While our approach is based on the multi-class discriminative framework of Bach and Harchaoui (2007) and is thus applicable to $P > 2$ classes, we focus for simplicity on the case $P = 2$ in this chapter (see Chapter 5, for a more sophisticated approach to the harder general multiclass case), and we aim at partitioning all the pixels from all images into only two classes, the *foreground* and the *background*. In this chapter, we slightly change our notation and denote the labels by the vector y in $\mathbb{R}^N$ such that:

$$y_j = \begin{cases} 1 & \text{if the } j^{th} \text{ pixel is in the } \textit{foreground}, \\ -1 & \text{otherwise.} \end{cases}$$

Our goal is to find $y \in \{-1, 1\}^N$, given only the I images and their associated features.

Co-segmenting a set of images to find a common object instance relies on maximizing the separability of two classes between different images and on maximizing spatial and appearance consistency within a particular image. The latter problem leads to methods designed for bottom-up *unsupervised* segmentation, e.g., spectral methods such as normalized cuts (Shi and Malik, 1997) without any sharing of information between different images, whereas the former problem leads to solving a top-down *discriminative* clustering problem which allows some shared information between images. The approach we propose combines both methods and solves the associated problems *simultaneously*.

2.2.1 Spatial consistency

In cosegmentation algorithms, visual and spatial consistency is usually enforced using binary terms based on total variation (Vicente et al., 2010) or the Laplacian of similarity matrices (Kim et al., 2011). While the former work well in interactive segmentation tasks (Boykov and Jolly, 2001), they do not admit the interpretation in terms of graphical spectral clustering of the latter (Shi and Malik, 1997). All along this thesis, we will develop graphical models for the purpose of segmentation and thus we choose to follow the approach of Shi and Malik (1997). We use a similarity matrix W^i to represent the local interactions between pixels of the same image i. This matrix is based on feature positions p_j and color vectors c_j, which is standard in spectral clustering (Shi and Malik, 1997), leading to high similarity for nearby pixels with similar color. We thus define the

similarity matrix W^i associated with image i as follows: for any pair (l, m) of pixels that belong to the i-th image, W^i_{lm} is zero if the two pixels are separated by more than two nodes in the image grid, and is given by:

$$W^i_{lm} = \exp(-\lambda_p \|p_m - p_l\|^2 - \lambda_c \|c_m - c_l\|^2) \tag{2.1}$$

otherwise. Empirically, values of $\lambda_p = 0.001$ and $\lambda_c = 0.05$ have given good results in our experiments. We can assemble the separate similarity matrices W^i, $i = 1, \ldots, I$, into a block-diagonal matrix $W \in \mathbb{R}^{N \times N}$, by putting the blocks $W^i \in \mathbb{R}^{N_i \times N_i}$ on the diagonal. We now consider the *Laplacian* matrix defined from the joint block-diagonal similarity matrix W. Denoting by D the diagonal matrix composed of the row sums of W, we define the normalized Laplacian matrix L as

$$L = I_N - D^{-1/2} W D^{-1/2},$$

where I_N is the N-dimensional identity matrix. Given the normalized Laplacian matrix, a spectral method like normalized cuts (Shi and Malik, 1997) outputs the second smallest eigenvector of L, which corresponds to:

$$\begin{aligned} \min \quad & y^\top L y, \\ \text{subject to} \quad & \|y\|^2 = N, \\ & y^\top D^{1/2} 1_N = 0, \end{aligned}$$

where 1_N denotes the N-dimensional vector of all ones. Following normalized cuts, we will thus include the term $y^\top L y$ into our objective function. Since L is block diagonal, minimizing this term alone leads to segmenting the images *independently* into two different groups, based solely on local features (color differences and position differences at nearby pixels).

2.2.2 Discriminative clustering

Our discriminative clustering framework is based on positive definite kernels (Shawe-Taylor and Cristianini, 2004). Since our d-dimensional features are all histograms, we consider a joint $N \times N$ positive semidefinite kernel matrix K (defined for all pairs of all pixels from all images) based on the χ^2-distance, with entries:

$$K_{lm} = \exp\left(- \lambda_h \sum_{f=1}^{d} \frac{(x_{lf} - x_{mf})^2}{x_{lf} + x_{mf}} \right), \tag{2.2}$$

where $\lambda_h > 0$. In the experiments, we use $\lambda_h = 0.1$. Note that we do not use the positions p_j to share information through images in order to be robust to object location.

Considering a positive definite kernel such as the one used in Eq. (2.2) is equivalent to mapping each of our N d-dimensional vectors x_j, $j = 1, \ldots, N$ into a high-dimensional Hilbert space $\mathcal{F}$ through a feature map Φ, so that $K_{ml} = \Phi(x_m)^T \Phi(x_l)$ (Shawe-Taylor and Cristianini, 2004). Kernel methods then aim at learning a classifier which is an affine function of $\Phi(x)$ through the minimization with respect to $f \in \mathcal{F}$ and $b \in \mathbb{R}$ of

$$\frac{1}{N} \sum_{j=1}^{N} \ell(y_j, f^T \Phi(x_j) + b) + \lambda_k \|f\|^2, \tag{2.3}$$

where $y_j \in \{-1, 1\}$ is the label associated with the j-th pixel and ℓ is a loss function. In this chapter, we consider the square loss $\ell(s, t) = (s - t)^2$ but other losses such as the hinge loss (for the SVM) could be considered (at the price of additional computational cost) (Xu et al., 2005).

Given the kernel matrix K (which is known and fixed) and the labels y (which are unknown), we denote by $g(y)$ the optimal solution of the supervised learning problem in Eq. (2.3) with the kernel matrix K and labels y. The optimal value $g(y)$ is a measure of the separability of the classes defined by $y \in \{-1, 1\}^N$.

Following Bach and Harchaoui (2007), for the square loss, we show that $g(y)$ can be obtained in closed form. The global minimum of Eq. (2.3) in f is:

$$f^* = \frac{1}{N} \Pi_N (I_N - \Phi(\Phi^T \Pi_N \Phi + N\lambda_k I)^{-1} \Phi^T) \Pi_N,$$

where $\Pi_N = I_N - 1_N 1_N^T$ and Φ is the matrix containing $\Phi(x_n)$ for all n. Similarly the global minimum in b is:

$$b^* = \frac{1}{N} 1_N^T (y - \Phi f^*).$$

Replacing these closed form solutions in Eq. (2.3) leads to following problem:

$$g(y) = y^\top B y,$$

where the $N \times N$ matrix B is defined as:

$$B = \frac{1}{N} \Pi_N (I_N - \Phi(\Phi^T \Pi_N \Phi + N\lambda_k I)^{-1} \Phi^T)' P i_N.$$

Finally, using Schur complement (Boyd and Vandenberghe, 2003), $g(y)$ is obtained in closed form, and it depends on K instead of Φ:

$$g(y) = y^\top A y, \tag{2.4}$$

where:

$$A = \lambda_k (I_N - \frac{1}{N} 1_N 1_N^T)(N\lambda_k I_N + K)^{-1} (I_N - \frac{1}{N} 1_N 1_N^T).$$

Degrees of freedom. Another advantage of using the square loss is that it gives a natural interpretation of the regularization parameter λ_k in terms of the implicit number of parameters of the learning procedure (Hastie et al., 2001). Indeed, the *degree of freedom* defined as $df = N(1 - \mathrm{tr}A)$, provides a simple way to set the regularization parameter λ_k (Hastie et al., 2001). In the experiments, we use $df = 100$ and deduce from it the value of λ_k.

Incomplete Cholesky decomposition. Evaluating A is prohibitive since inverting an $N \times N$ square matrix has an $O(N^3)$ complexity. Following Bach and Harchaoui (2007), we use an incomplete Cholesky decomposition for the kernel matrix K to reduce this complexity to $O(N)$: For a fixed rank $r < N$, we obtain an $N \times r$ dimensional matrix G such as $K \approx GG^T$. Using the matrix inversion lemma, this allows us to invert an $r \times r$ system instead of an $N \times N$ one. The overall complexity is therefore $O(Nr^2)$. In our simulations, we use $r = \min(N, 400)$.

Cluster size constraints. Putting all pixels into a single class leads to perfect separation (this can be seen by noticing that the matrix A is positive semidefinite and satisfies $1_N^T A 1_N = 0$). Following (Bach and Harchaoui, 2007; Xu et al., 2005), we add constraints on the number of elements in each class to avoid this trivial solution. In our situation where the N observations (pixels) belong to I different images, we constrain the number of elements of each class in *each image* to be upper bounded by λ_1 and lower bounded by λ_0. If $\delta_i \in \mathbb{R}^N$ is the indicator vector of the i-th image, with $(\delta_i)_j = 1$ if the j-th pixel is in the i-th image and 0 otherwise, then the constraints are equivalent to the component-wise inequalities:

$$\lambda_0 N_i \delta_i \leqslant \tfrac{1}{2}(yy^\top + 1_N 1_N^T)\delta_i \leqslant \lambda_1 N_i \delta_i.$$

Empirically, we have observed that different choices of λ_0 and λ_1 do not change the results much as long as λ_0 is small enough. Therefore we have fixed $\lambda_0 = 5\%$ and $\lambda_1 = 95\%$.

Problem formulation. Finally, combining a spatial consistency term associated with the Laplacian matrix L with a discriminative cost associated with the matrix A, and adding the cluster size constraints, we obtain the following problem:

$$\begin{aligned} \min \quad & y^T\big(A + \tfrac{\mu}{N} L\big) y && (2.5) \\ \text{subject to} \quad & y \in \{-1, 1\}^N, \\ & \forall i,\ \lambda_0 N_i \delta_i \leqslant \tfrac{1}{2}(yy^\top + 1_N 1_N^T)\delta_i, && (2.6) \\ & \forall i,\ \tfrac{1}{2}(yy^\top + 1_N 1_N^T)\delta_i \leqslant \lambda_1 N_i \delta_i. && (2.7) \end{aligned}$$

In the next section, we show how this optimization problem may be relaxed to a convex one that can be solved efficiently.

2.3 Optimization

In this section, we present an efficient convex relaxation of the hard combinatorial optimization problem defined in Eq. (2.5). It follows mostly the ideas of the semidefinite convex relaxation for QCQPs presented in the introduction.

2.3.1 Convex relaxation

The problem define by Eq. (2.5) is a special case of quadratic constrained quadratic programming. We thus follow (Goemans and Williamson, 1995), and notice that the objective function may be rewritten as:

$$y^T\big(A+\frac{\mu}{N}L\big)y = \mathrm{tr}\big(yy^T\big(A+\tfrac{\mu}{N}L\big)\big).$$

Thus, we may reparameterize Eq. (2.5) with $Y = yy^\top$. The matrix Y is usually referred to as the *equivalence matrix* ($Y_{ij} = 1$ if pixels i and j are in the same cluster and -1 otherwise). The constraint $y \in \{-1,1\}^N$ is then equivalent to Y being symmetric, positive semidefinite, with diagonal equal to one, and unit rank. Thus, if we denote by $\mathcal{E}$ the *elliptope*, i.e., the convex set defined by:

$$\mathcal{E} = \{Y \in \mathbb{R}^{N\times N}\ ,\ Y = Y^T\ ,\ \mathrm{diag}(Y) = 1_N\ ,\ Y \succeq 0\},$$

Eq. (2.5) is equivalent to:

$$\begin{aligned} &\min_{Y\in\mathcal{E}} \mathrm{tr}\big(Y\big(A+\tfrac{\mu}{N}L\big)\big), \\ \text{subject to}\quad &\forall i,\ \lambda_0 N_i\delta_i \leqslant \tfrac{1}{2}(Y + 1_N 1_N^T)\delta_i \leqslant \lambda_1 N_i\delta_i \\ &\mathrm{rank}(Y) = 1. \end{aligned} \tag{2.8}$$

As noted in the introduction, the rank constraint ensures that the solution of Eq. (2.8) is an integer matrix but makes the continuous problem Eq. (2.8) non-convex. We thus remove this constraint, to obtained a relaxed convex optimization problem over positive definite matrices, i.e, a semidefinite program (SDP) (d'Aspremont and Boyd, 2003; Boyd and Vandenberghe, 2003).

2.3.2 Efficient low-rank optimization

Without using the structure of this problem, general purpose toolboxes would solve it in $O(n^7)$ (Boyd and Vandenberghe, 2003; Grant and Boyd, 2010), which is clearly not acceptable in our situation. Bach and Harchaoui (2007) consider a partial dualization technique that solves the relaxed problem through a sequence of singular value decompositions and scales up to thousands of data points. To gain another order of magnitude, we adopt the framework for optimization through low-rank matrices proposed in (Journée et al., 2010).

From constraints to penalties. Unfortunately, the procedure developed in Journée et al. (2010) cannot deal with inequality constraints. Therefore we use an *augmented Lagrangian method* to transform these into penalties (Bertsekas, 1995). Such a method consists in replacing a given constraint by another one that ensure that the original constrain is satisfied. More precisely, for each constraint of the form $h(Y) \leqslant 0$, we add a twice differentiable convex penalty term to the objective function, i.e:

$$C(h(Y)) = \max\{0, h(Y)^3\}.$$

Denoting by $h_i(Y)$, the i-th linear inequality in Eq. (2.8), our minimization problem is thus reduced to:

$$\begin{aligned} \min \quad & f(Y) = \text{tr}(AY) + \nu \sum_i C(h_i(Y)), \\ \text{subject to} \quad & Y \in \mathcal{E}. \end{aligned}$$

To ensure that the constraints are respected after convergence, we follow Bertsekas (1995) and increase ν by a constant factor at every iteration of our iterative scheme. More precisely, at each iteration k, we minimize the function $f_k(Y) = \text{tr}(AY) + \nu_k \sum_i C(h_i(Y))$, such that $\mu_k = \alpha\mu_{k-1}$ with $\alpha = 1.01$. We increase μ_k slowly as to keep the convergence guarentees. In the rest of this chapter, with a slight abuse of notation, we denote by f the function f_k for the k-th iteration.

Low-rank solutions. We are now faced with the optimization of a convex function $f(Y)$ on the elliptope $\mathcal{E}$, potentially with rank constraints. The unconstrained minimization of convex functions on the elliptope is convex and empirically often leads to low-rank solutions (Journée et al., 2010). In this chapter, we propose to take advantage of this observation. Instead of considering the entire elliptope, we restrict our search space to matrices in the elliptope with low rank. More precisely, we denote by r be the unobserved true rank of the solution and by $\hat{r}$ its estimation. We consider the function $g_{\hat{r}} : y \mapsto f(yy^\top)$ defined for matrices $y \in \mathbb{R}^{N\times\hat{r}}$ such that $yy^\top$ is in the elliptope, i.e., such that $\text{diag}(yy^\top) = 1_N$. Even if $g_{\hat{r}}$ is not convex, this function has the interesting property that for any $\hat{r} > r$, all its local minima correspond to a global minimum of f over the elliptope (Journée et al., 2010). In the case were the rank r of the optimal solution is known, a simple local descent procedure would be guaranteed to minimize $g_{\hat{r}}$ for $\hat{r} = r + 1$. When r is not known, Journée et al. (2010) have designed an adaptive procedure, that first considers $\hat{r} = 2$, finds a local minimum of $g_{\hat{r}}$, and checks whether it corresponds to a global optimum of f using second order derivatives of f. If not, then $\hat{r}$ is increased by one and the same operation is performed until the actual rank r is reached. Thus, when $\hat{r} = r + 1$, we must get an optimum of the convex problem, which has been obtained by a sequence of local minimizations of low-rank non-convex problems. Note that we obtain a global minimum of $f(Y)$ regardless of the chosen initialization of the low-rank descent algorithm.

Trust-region method on a manifold. Crucial to the rank-adaptive method presented earlier is the guarantee of obtaining local minima of the low-rank problems Note that unfortunately, simple gradient descent schemes on $y \in \mathbb{R}^{N \times \hat{r}}$ would only give stationary points, i.e., points such that $\nabla g_{\hat{r}}(y) = 0$. Instead, following Absil et al. (2008), we first notice that the cost $g_{\hat{r}}$ is invariant by right-multiplication of y by an $\hat{r} \times \hat{r}$ orthogonal matrix. Therefore, we perform our minimization on the quotient space:

$$\bar{\mathcal{E}}_{\hat{r}} = \mathcal{E}_{\hat{r}} / \mathcal{O}_{\hat{r}},$$

where

$$\mathcal{E}_{\hat{r}} = \{Y \in \mathcal{E}, \text{rank}(Y) = \hat{r}\}$$

and

$$\mathcal{O}_{\hat{r}} = \{P \in \mathbb{R}^{\hat{r} \times \hat{r}} | PP^T = I_{\hat{r}}\}.$$

Journée et al. (2010) show that, for $\hat{r}$ greater than 2, $\bar{\mathcal{E}}_{\hat{r}}$ is a Riemannian manifold. In order to find a local minimum on this quotient space, we can thus use a second-order trust-region method for such manifold[1], with guaranteed convergence to local minima rather than stationary points (Absil et al., 2008). Note the following interesting phenomenon: our overall goal is to minimize $g_{\hat{r}}$ for $\hat{r} = 1$, which is a combinatorial problem, but replacing this original rank constraint by other rank constraints setting $\hat{r}$ greater than 2, we get a Riemannian manifold, and for $\hat{r}$ large enough, all local minima are provably global minima. Thus, in this case, increasing dimension helps the optimization. This non intuitive property of this algorithm is directly related to semidefinite relaxations where low dimension problems are relaxed to high dimension problems, as explained in the introduction. We show in Section 2.3.3 how to project back the solution to rank-one matrices.

Preclustering. Since our cost function f uses the full $N \times N$ matrix $A + (\mu/N)L$, the memory cost of our algorithm may be prohibitive. This has prompted us to use superpixels obtained from an oversegmentation of our images. As shown in the introduction, there are many algorithms for bottom-up segmentation and for this chapter, we use the watershed implementation of (Meyer, 2001). We show a segmentation example in Figure 2.1. Using s superpixels is equivalent to constraining the matrix Y to be block-constant and thus reduces the size of the SDP to a problem of size $s \times s$. In our experiments, for a single image, s can be between 50 to 200. For 30 images, we use in general $s = 3000$.

Running time. We perform our experiments on a 2.4 gHz processor with 6 gB of RAM. Our code is in MATLAB. The optimization method has an overall complexity of

[1] we use the code from www.montefiore.ulg.ac.be/~journee/ in our experiments.

$O(s^2)$ in the number of superpixels. Typically, depending on the number of superpixels in an image, it takes a few seconds to segment a pair of images. For 30 images, it takes less than 10 minutes.

2.3.3 Rounding

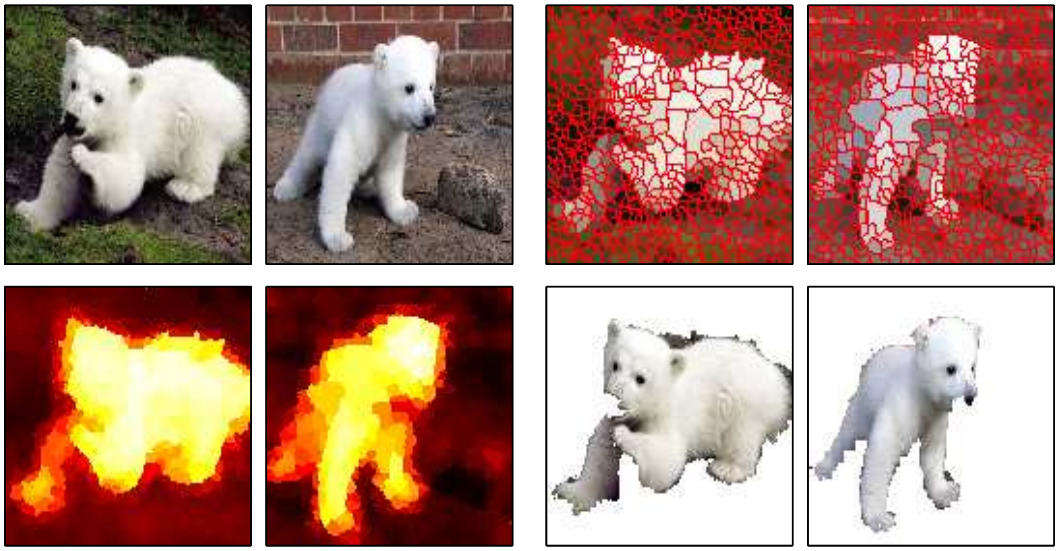


Figure 2.1: Illustrating the co-segmentation process on two bear images. From the first to the last row and from left to right: input images, over-segmentations, scores obtained by our algorithm and co-segmentations. We use $\mu = 1$.

We have presented in Section 2.3.2 an efficient method for solving the optimization problem of Eq. (2.8) without the rank constraint. In order to retrieve $y \in \{-1, 1\}$ from a matrix Y in $\mathcal{E}$ with rank larger than one, several alternatives have been considered in the literature, using randomization or eigenvalue decomposition for example (Goemans and Williamson, 1995; Shi and Malik, 1997). In this chapter, we follow the latter approach, and compute the eigenvector $e \in \mathbb{R}^N$ associated with the largest eigenvalue of Y, which is equivalent to projecting Y on the set of unit-rank positive definite matrices (Boyd and Vandenberghe, 2003). We refer to $e \in \mathbb{R}^N$ as the segmentation score of our algorithm. We then consider $y \in \mathbb{R}^N$ as the component-wise sign of e, i.e., 1 for positive values, and -1 otherwise. Our final clustering is obtained by thresholding the score at 0 (see example in Figure 2.1). Note that an adaptive threshold selection could be considered as well. Empirically, we have noticed that adapting the threshold does not give better results that fixing it to 0.

Post-processing. In this chapter, we subsample the grid to make the algorithm faster: we clean the coarse resulting segmentation by applying a fast bottom-up segmentation algorithm based on graph cuts on the original grid, seeded by the score e (Boykov et al., 2001; Kolmogorov and Zabih, 2004). We use the same parameters for this algorithm in all our experiments, except the dog (Figure 2.2), for which we adjusted them to obtain more restrictive segmentation. We could also use our algorithm as an initialization for other co-segmentation methods (Rother et al., 2006).

2.4 Experiments

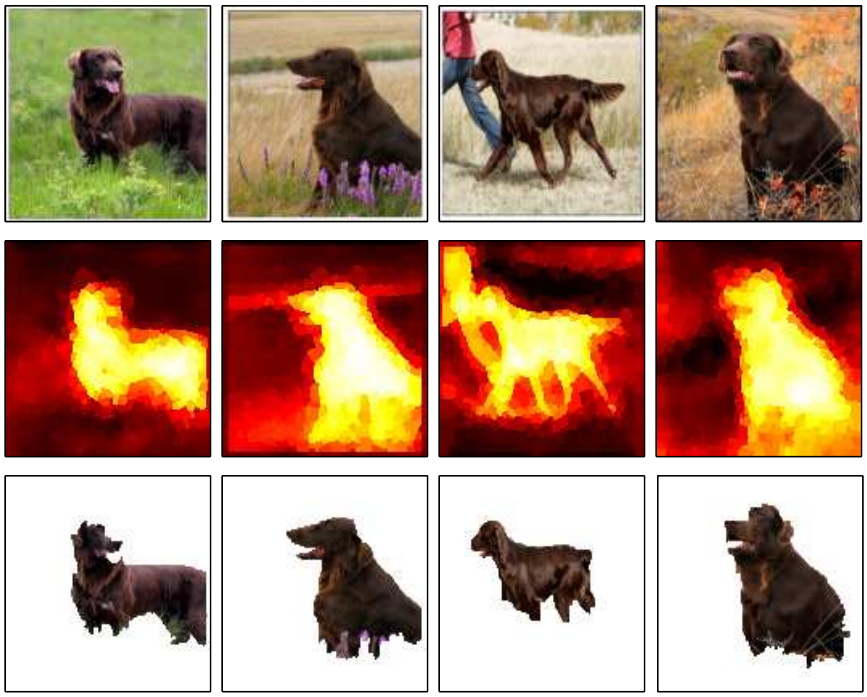


Figure 2.2: Dog images: (top) input images, (middle) scores obtained by our algorithm and (bottom) co-segmentations. $\mu = 1$.

We present our results on different datasets. In Section 2.4.1, we first consider images with foreground objects which are identical or very similar in appearance and with few images to co-segment, a setting that was already used in (Rother et al., 2006) and extended in (Hochbaum and Singh, 2009). Then, in Section 2.4.2, we consider images where foreground objects exhibit higher appearance variations, with more images to co-segment (up to 30).

We present both qualitative and quantitative results. In the latter case, co-segmentation performance is measured by its accuracy, which is the proportion of correctly classified pixels (foreground *and* background). To evaluate the accuracy of our algorithm on a dataset, we evaluate this quantity for each image separately. Note that in our unsupervised approach we have one indeterminacy, i.e., we do not know if positive labels correspond to foreground or to background. We thus select by hand the best candidate (one single choice for *all* images of the same class), but simple heuristics could be used to alleviate this manual choice.

Tradeoff between bottom-up segmentation and discriminative clustering. The parameter μ, which weighs the spatial and color consistency and discriminative cost function, is the only free parameter; in our simulations, we have considered two settings: $\mu = 1$, corresponding to foreground objects with fairly uniform colors, and $\mu = 0.001$, corresponding to objects with sharp color variations.

2.4.1 Experiments with low-variability datasets

Figure 2.3: (Left) stone images and (right) girl images: (top) input images, (middle) scores obtained by our algorithm, (bottom) co-segmentations. $\mu = 0.001$.

We first present results obtained by our algorithm on a set of images from (Hochbaum and Singh, 2009; Rother et al., 2006). Following the experimental set-up in these papers, our feature vector is composed of color histograms and Gabor features. For synthetic examples with identical foreground objects (girl, stone, boy), we use 25 buckets per color channel, while for natural images (bear, dog) we use 16 buckets as to be more robust to change of light. Since we only consider a few images (2 in all cases, except 4 for the dogs), we do not need to subsample the images, i.e. we did not use superpixels. Segmentation results are shown in Figures 2.1 to 2.4 (note that these are best seen on screen).

Qualitatively and quantitatively, our co-segmentation framework gives similar results to (Hochbaum and Singh, 2009) and (Rother et al., 2006), except on the boy (Figure 2.4), where our algorithm fails to find the head. This is due to the strong edge between the coat and the head and the similarity in color with the wood in the second image. Setting $\lambda_c = 0$ in the Laplacian matrix would improve the results, but this would add an additional parameter to tune.

Figure 2.4: A failure case: (left) input images, (middle) scores obtained by our algorithm, (right) co-segmentation. We use $\mu = 0.001$.

Quantitative results are given in Table 2.1. We compare our algorithm with Hochbaum and Singh (2009) and Rother et al. (2006). In general, their results are also better than ours, but, their algorithm exploits some a priori knowledge of background and foreground colors. Our algorithm starts from scratch, without any such prior information.

	Girl	Stone	Boy	Bear	Dog
our method	**0.8** %	**0.9** %	6.5 %	5.5%	6.4 %
Hochbaum and Singh (2009)	-	1.2%	**1.8** %	**3.9** %	**3.5** %
Rother et al. (2006)	-	1.9%	2.2%	-	-

Table 2.1: Segmentation errors on pairs of images.

2.4.2 Experiments with high-variability datasets

In this section, we consider co-segmentation problems which are much harder and cannot be readily solved by previous approaches. They demonstrate the robustness of our framework as well as its limitations.

Oxford flowers. We first consider a class of flowers from the Oxford database[1], with 30 images, subsampled grids (with a ratio of 4), and oversegmentation into an average of 100 superpixels. Results are shown in Figure 2.5 and illustrate that our co-segmentation algorithm is able to co-segment almost perfectly larger sets of natural images.

Weizman horses and MSRC database. We co-segment images from the Weizmann horses database[2] and the MSRC database[3], for which ground truth segmentations are available. Our aim is to show that our method is robust to foreground objects with higher appearance variations. Our feature vectors are 16×16 SIFT descriptors taken every 4 pixels. We choose SIFT instead of color histograms because SIFT is usually more robust to such variability. We use an over-segmentation with an average of 400 super-pixels to speed up the algorithm. Sample segmentation results are shown in Figures 2.6 – 2.8, with quantitative results in Table 2.2.

We consider three different baselines: for the first one ("single-image"), we simply use our algorithm on each images *independently* as it can be used for bottom-up single image segmentation. Once each of these images are segmented into two segments, we choose the assignments of the two segments to foreground/background labels so that the final segmentation accuracy is maximized. In other words, we use the test set to find the best assignment, which can only make this baseline artificially better.

The second baseline ("MNcut") is another bottom-up image segmentation with a multiscale normalized cut framework (Cour et al., 2005). The third baseline ("uniform") simply classifies all the pixels of all the images into the same segment (foreground or background), and keep the solution with maximal accuracy. Qualitatively, our

[1] www.robots.ox.ac.uk/~vgg/data/flowers/17/
[2] www.msri.org/people/members/eranb/
[3] www.research.microsoft.com/en-us/projects/objectclassrecognition/

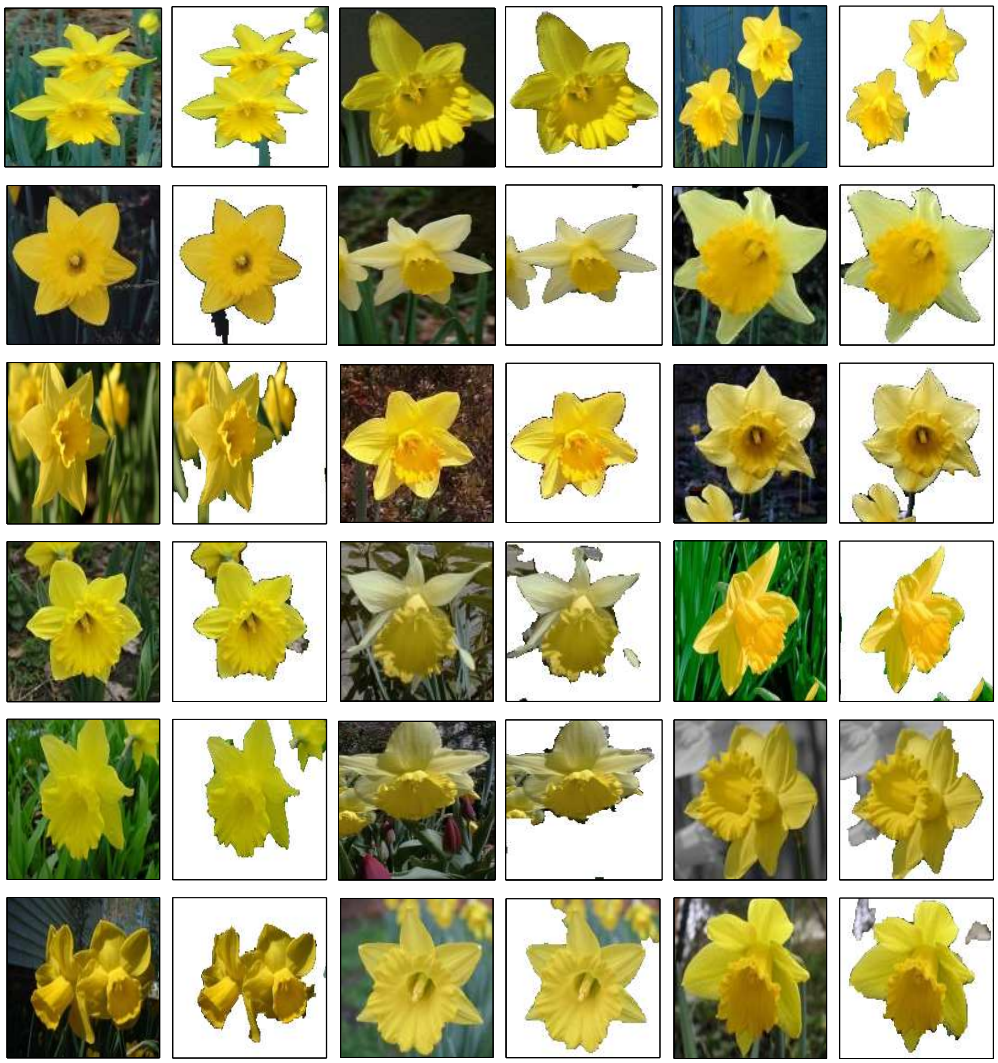


Figure 2.5: Flower images: the original image is given with its segmentation.

method works well on the cows, faces, horses and car views but does not do as well on cats and planes and worse on bikes. For cats, this can be explained by the fact that these animals possess a natural camouflage that makes it hard to distinguish them in their own environment. Also, the cats in the MSRC database have a wide range of positions and textures. The low score on planes may be explained by the fact that the background does not change much between images, so in fact our method may consider that the airport is the object of interest, while the planes are changing across images. The score on bikes is low because our algorithm fails to segment the regions inside the wheels, leading to

class	images	our method	single-image	Cour et al. (2005)	uniform
Cars (front)	6	87.65% ±0.1	**89.6 % ±0.1**	51.4 % ±1.8	64.0 % ±0.1
Cars (back)	6	**85.1 % ±0.2**	83.7 % ±0.5	54.1%±0.8	71.3 % ±0.2
Face	30	**84.3% ±0.7**	72.4% ±1.3	67.7% ±1.2	60.4% ±0.7
Cow	30	**81.6 % ±1.4**	78.5 % ±1.8	60.1% ±2.6	66.3 % ±1.7
Horse	30	**80.1 % ±0.7**	77.5 % ±1.9	50.1% ±0.9	68.6 % ±1.9
Cat	24	**74.4 % ±2.8**	71.3 % ±1.3	59.8% ±2.0	59.2 % ±2.0
Plane	30	73.8 % ±0.9	62.5 % ±1.9	51.9% ±0.5	**75.9 % ±2.0**
Bike	30	**63.3 % ±0.5**	61.1 % ±0.4	60.7% ±2.6	59.0% ±0.6

Table 2.2: Segmentation accuracies on the Weizman horses and MSRC databases.

low scores even though, qualitatively, the results are still reasonable.

Quantitatively, as shown in Table 2.2, our method outperforms the baselines except for the bikes and frontal views of cars. To be fair, it should be noted, however, that a visual inspection of the single-and multi-image versions of our algorithm give qualitatively similar results on several datasets. One possible explanation is that the various backgrounds are not that different from one another. Thus, much of the needed information can be retrieved from a single image, with the discriminative clustering still improving the results. Note also that our discriminative framework on a single image outperforms "MNcut".

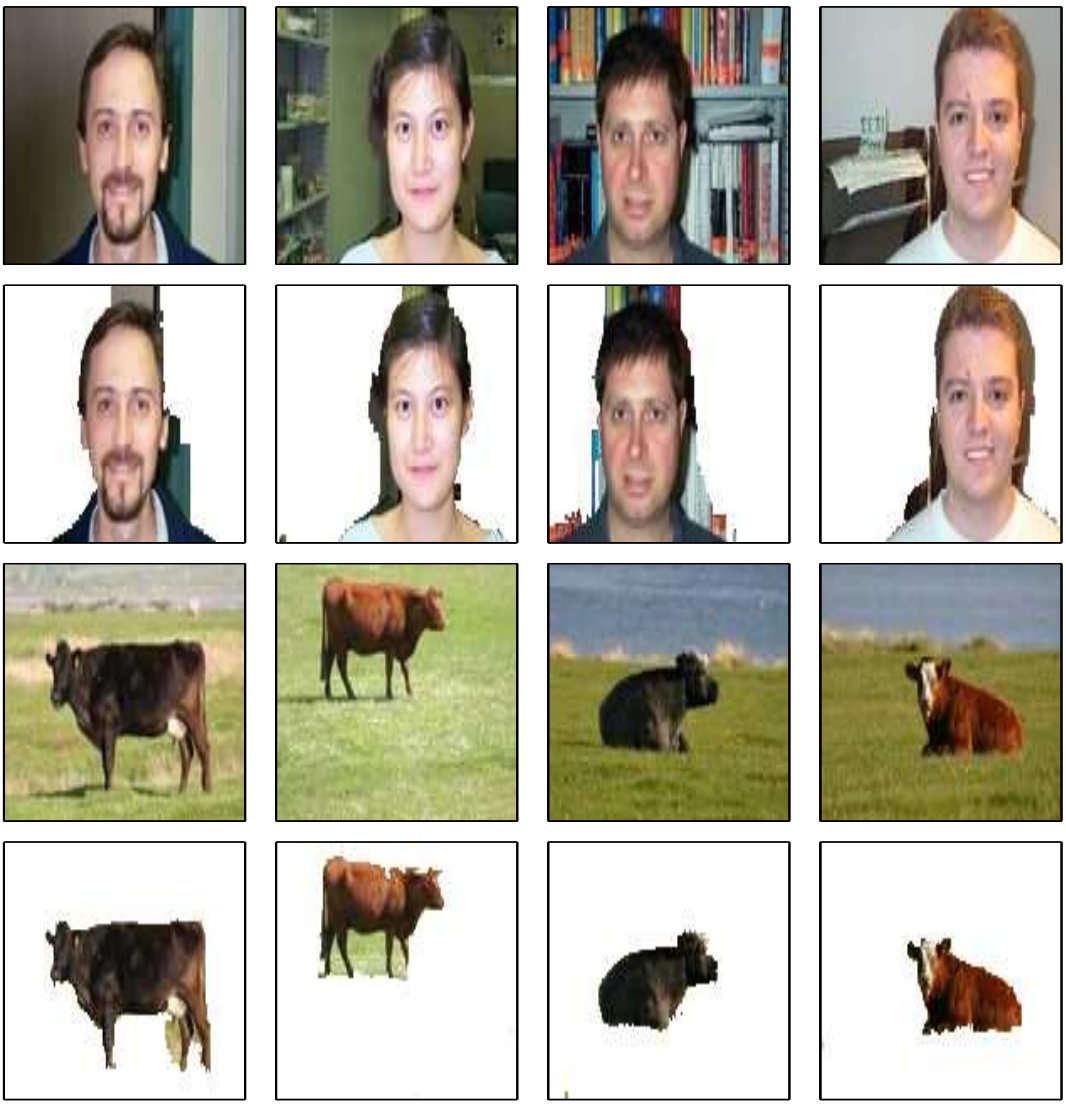


Figure 2.6: Images and segmentation results for our method on MSRC databases.

Co-segmentation vs. independent segmentations. One may therefore wonder if co-segmentation offers a real gain, but there are at least two reasons for using it. First, there is a quantitative gain on almost all datasets and, secondly, co-segmentation from multiple images not only finds the foreground and background regions but it *automatically* classifies them, whereas this must be done manually if the images are segmented independently. Figure 2.9 shows the different segmentations obtained with "MNcut", single-image segmentation and co-segmentation. The first row shows an example where, on

Figure 2.7: Images and segmentation results for our method on Weizman horses and MSRC databases.

a single image, our algorithm outperforms "MNcut", but where the difference between single-and multi-image segmention is less clear. In fact, for several images, both our versions give the same results. The second row shows a case where on a single image "MNcut" and our algorithm behave similarly but adding information from other images enhances the results, i.e., co-segmentation has noticeably improved performance. Another way to improve the background/foreground segmentation performance is to con-

Figure 2.8: Images and segmentation results for our method on Weizman horses and MSRC databases.

sider a multiclass framework instead of a single class for the background. This direction is considered in Chapter 5.

Influence of μ. We show an example of the influence of μ on the segmentation, Figure 2.10. As we can if the value is small, our algorithm tends to find multiple small regions whereas when the value is big, it segments the image in big regions. Empiri-

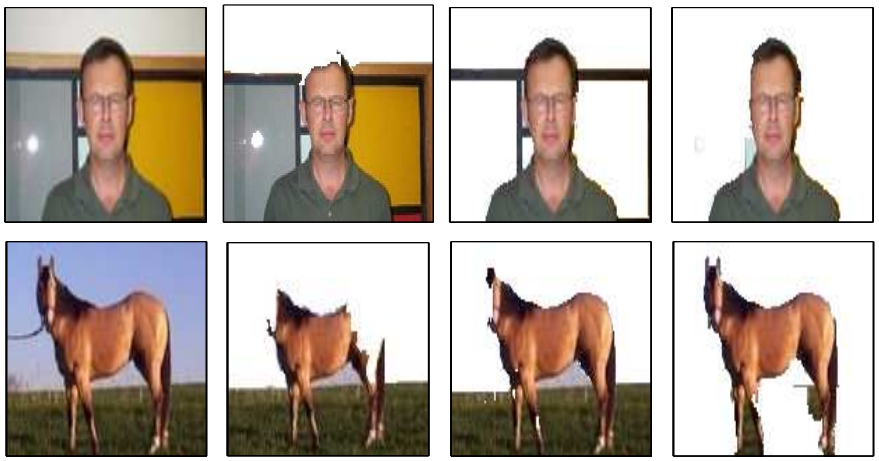
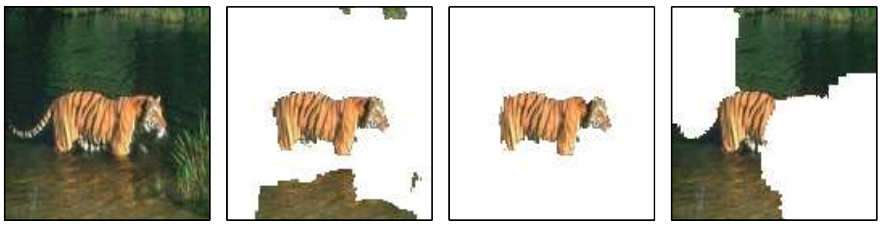


Figure 2.9: Comparing multi-image with single-image segmentations; from left to right: original image, multiscale normalized cut, our algorithm on a single image, our algorithm on 30 images.



original $\mu = 0.001$ $\mu = 1$ $\mu = 1000$

Figure 2.10: Influence of the value of μ on the segmentation.

cally, we observe that the best results are obtain for μ between 1 and 0.01.

3

Optimization for Discriminative Latent Class Models

Abstract of this chapter: Dimensionality reduction is commonly used in the setting of multi-label supervised classification to control the learning capacity and to provide a meaningful representation of the data. In this chapter, we introduce a simple forward probabilistic model which is a multinomial extension of reduced rank regression, and show that this model provides a probabilistic interpretation of discriminative clustering methods with added benefits in terms of number of hyperparameters and optimization. While the expectation-maximization (EM) algorithm is commonly used to learn these probabilistic models, it usually leads to local optima because it relies on a non-convex cost function. To avoid this problem, we introduce a local approximation of this cost function, which in turn leads to a quadratic non-convex optimization problem over a product of simplices. In order to optimize quadratic functions, we propose an efficient algorithm based on convex relaxations and low-rank representations of the data, capable of handling large-scale problems.

The material of this chapter is based on the following work:

A. Joulin, F. Bach and J. Ponce. Optimization for Discriminative Latent Class Models. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

3.1 Introduction

Latent data representations are wide-spread tools in supervised, weakly supervised and unsupervised learning. They are used for dimensionality reduction for two main reasons: on the one hand, they provide numerically efficient representations of the data; on the other hand, they may lead to better predictive performance than directly using the original data. In supervised learning, latent models are often used in a generative way,

e.g., through mixture models on either the input variables only or jointly on the input and output variables. As explained in the introduction, the predictive performance is often better for discriminative models than for generative ones. Sometimes dimensionality reduction methods may not lead to any increase in predictive performance compare to directly using the input data. This has led to numerous works on supervised dimension reduction (Blei et al., 2003; Blei and Mcauliffe, 2008), where the final discriminative goal of prediction is taken explicitly into account during the learning process.

In this context, various probabilistic models have been proposed, such as mixtures of experts (Jacobs et al., 1991) or discriminative restricted Boltzmann machines (Larochelle and Bengio, 2008), where a layer of low dimensional hidden variables is used between the inputs and the outputs of the supervised learning model. Parameters are usually estimated by block-coordinate gradient descent procedures such as expectation-maximization (EM) or back/forward propagation. These methods are computationally efficient but usually converge to local optima which can be arbitrarily far from the global optimum. In this chapter, we are interested in learning the parameters of a probabilistic model and thus consider the EM procedure whose cost function may have many local optima in high dimensions. We propose a quadratic approximation of the EM cost function which is optimized to obtain robust initializations for the EM procedure. In this chapter, we consider a simple *discriminative latent class* (DLC) model where inputs and outputs are independent given the latent representation.

3.2 Probabilistic discriminative latent class models

In this chapter, each instance or observation n is associated with an observable label $z_n \in \{1, \dots, L\}$ and a latent label $y_n \in \{1, \dots, P\}$. We suppose that this latent label is predictive of the observed label z_n. We model directly the conditional probability of y_n given the input data x_n and the probability of the label z_n given y_n, while making the assumption that z_n and x_n are independent given y_n (leading to the directed graphical model $x_n \rightarrow y_n \rightarrow z_n$). More precisely, we assume that given x_n, y_n follow a multinomial logit model while given y_n, z_n may take any value independent of n:

$$p(y_n = p \mid x_n) = \frac{e^{w_p^T x_n + b_p}}{\sum_{j=1}^{P} e^{w_j^T x_n + b_j}} \tag{3.1}$$

$$p(z_n = l \mid y_n = p) = \alpha_{pl}, \tag{3.2}$$

with $w_p \in \mathbb{R}^d$, $b_p \in \mathbb{R}$ and $\sum_{l=1}^{L} \alpha_{pl} = 1$. We use the notation $w = (w_1, \dots, w_P)$, $b = (b_1, \dots, b_P)$ and $\alpha = (\alpha_{pl})_{1 \leq p \leq P, 1 \leq l \leq L}$. Note that the model defined by (3.1) can be kernelized by replacing implicitly or explicitly x by the image $\Phi(x)$ of a non linear mapping.

Related models. The simple two-layer probabilistic model defined in Eq. (3.1), can be interpreted and compared to other models in various ways. First, it is an instance of a mixture of experts Jacobs et al. (1991) where each expert has a constant prediction. It has thus weaker predictive power than general mixture of experts; however, it allows a more efficient optimization as shown in Section 3.4. It would be interesting to extend the optimization techniques derived in their paper to the case of experts with non-constant predictions. This is what is done in Quadrianto et al. (2009) where a convex relaxation of EM for a similar mixture of experts is considered. However, Quadrianto et al. (2009) considers the maximization with respect to hidden variables rather than their marginalization, which is essential in our setting to have a well-defined probabilistic model. Note also that in (Quadrianto et al., 2009), the authors derive a convex relaxation of the softmax regression problems, while we derive a quadratic approximation. It is worth trying to combine the two approaches in future work.

Another related model is a two-layer neural network with only one neuron which can take more than two states. Extending our model to multiple neurons is not straightforward since our robust initialization would lead to the same values for all the neurons. This is a common issue with neural networks as there is nothing enforcing neurons to be different. Another difference with a two-layer neural network with softmax functions for the last layer is the fact that our last layer considers linear parameterization in the mean parameters rather than in the natural parameters of the multinomial variable. These two parameterizations are equivalent but ours allows us to provide a convexification of two-layer neural networks in Section 3.4.

Among probabilistic models, a discriminative restricted Boltzmann machine (RBM) (Larochelle and Bengio, 2008) models $p(y|z)$ as a softmax function of linear functions of z. Our model assumes instead that $p(y|z)$ is linear in z. Again, this distinction between mean parameters and natural parameters allows us to derive a quadratic approximation of our cost function. It would of course be of interest to extend our optimization technique to the discriminative RBM.

Finally, one may see our model as a multinomial extension of reduced-rank regression (Hastie et al., 2001), which is commonly used with Gaussian distributions and reduces to singular value decomposition in the maximum likelihood framework.

3.3 Inference

We consider the negative conditional log-likelihood of z_n given x_n as a function of the parameters $\theta = (\alpha, w, b)$:

$$\ell(\theta) \quad = \quad -\frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L} z_{nl} \log p(z_{nl} = 1|x_n) + \frac{\lambda}{2P}\|w\|_F^2,$$

where z_{nl} is equal to 1 if $z_n = l$ and 0 otherwise, and λ is a regularization parameter. We add the regularization on w to avoid overfitting, i.e., to avoid to some extent to learn the bias introduced by the training data (Hastie et al., 2001).

3.3.1 Expectation-maximization

A popular tool for solving maximum likelihood problems is the EM algorithm (Hastie et al., 2001). A traditional way of viewing EM is to add auxiliary variables and minimize the following upperbound of the negative log-likelihood ℓ, obtained by using the Jensen inequality:

$$F(\xi, \theta) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} z_{nl} \Big[\sum_{p=1}^{P} \xi_{np} \log \frac{z_n^T \alpha_p}{\xi_{np}} \frac{e^{w_p^T x_n + b_p}}{\sum_{p=1}^{P} e^{w_p^T x_n + b_p}} \Big] + \frac{\lambda}{2P} \|w\|_F^2,$$

where $\alpha_p = (\alpha_{p1}, \ldots, \alpha_{pL})^T \in \mathbb{R}^L$ and $\xi = (\xi_1, \ldots, \xi_P)^T \in \mathbb{R}^{N \times P}$ with $\xi_n = (\xi_{n1}, \ldots, \xi_{nP})^T \in \mathbb{R}^P$. Traditionally, the auxiliary function $F(\xi, \theta)$ is described as above, i.e., as a tight upperbound of the log-likelihood ℓ obtained by the Jensen inequality. Another interpretation of this quantity is to consider it in terms of Fenchel conjugate, as we will do in Chapter 5. The two interpretations are equivalent and the important feature of this auxiliary function is that its minimum in the auxiliary variables is equal to the original log-likelihood.

The EM algorithm can be viewed as a two-step block-coordinate descent procedure (Hunter and Lange, 2004), where the first step (E-step) consists in finding the optimal auxiliary variables ξ, given the parameters of the model θ. In our case, the result of this step is obtained closed form :

$$\xi_{np} \propto z_n^T \alpha_p e^{w_p^T x_n + b_p},$$

with $\xi_n^T 1_P = 1$.

The second step (M-step) consists of finding the best set of parameters θ, given the auxiliary variables ξ. Optimizing the parameters α_p leads to the closed form updates:

$$\alpha_p \propto \sum_{n=1}^{N} \xi_{np} z_n,$$

with $\alpha_p^T 1_L = 1$. Optimizing jointly on w and b leads to a softmax regression problem.

Since $F(\xi, \theta)$ is not jointly convex in ξ and θ, this procedure stops when it reaches a local minimum, and its performance strongly depends on its initialization. We propose in the following section a robust initialization for EM given our latent model, based on an approximation of the auxiliary cost function obtained with the M-step.

3.3.2 Initialization of EM

Minimizing F w.r.t. ξ leads to the original log-likelihood $\ell(\theta)$ depending on θ alone. Minimizing F w.r.t. θ gives a function of ξ alone. In this section, we focus on constructing a quadratic approximation of this function, which will be minimized to obtain an initialization for EM.

We consider second-order Taylor expansions around the value of ξ corresponding to the uniformly distributed latent variables y_n, independent of the observations x_n, i.e., $\xi_0 = \frac{1}{P} 1_N 1_P^T$. This choice is motivated by the lack of a priori information on the latent classes. For clarity we divide our cost function into three terms and explain the calculation of the expansion of these terms independently.

Cost function as a sum of three terms. Using the relation between θ and ξ given by the M-step, we propose to divide our cost function into a term depending on α, another depending on (w, b) and a third one independent of θ. Taking the part of our cost function that depends on α, and replacing α by its expression, we get the function J_α:

$$J_\alpha(\xi) = \sum_{p=1}^{P} \sum_{l=1}^{L} \left(\sum_{n \in A_m} \frac{\xi_{np}}{N} \right) \log \left(\sum_{n \in A_m} \frac{\xi_{np}}{N} \right) - \sum_{p=1}^{P} \left(\sum_{n=1}^{N} \frac{\xi_{np}}{N} \right) \log \left(\sum_{n=1}^{N} \frac{\xi_{np}}{N} \right),$$

where A_m is the set of n such as $z_{nm} = 1$. Similarly with (w, b), we get the function J_{wb}:

$$J_{wb}(\xi) \quad = \quad \max_{\substack{w \in \mathbb{R}^{N \times P}, \\ b \in \mathbb{R}^P}} \frac{1}{N} \sum_{n=1}^{N} \xi_n (w^\top x_n + b) - \frac{1}{N} \sum_{n=1}^{N} \varphi(w^\top x_n + b) - \frac{\lambda}{2P} \|w\|_F^2,$$

where $\varphi(u) = \log(\sum_{p=1}^{P} \exp(u_p))$ is the log-sum exp function, and ξ_n is the n-th row of ξ. Finally there is a third term independent of θ in $F(\xi, \theta)$:

$$J_C(\xi) \quad = \quad -\frac{1}{N} \sum_{n=1}^{N} \sum_{p=1}^{P} \xi_{np} \log \xi_{np}.$$

We call $F(\xi)$ the sum of $J_C(\xi)$, $J_{wb}(\xi)$ and $J_\alpha(\xi)$.

Second-order Taylor expansion of $J_{wb}(\xi)$. Assuming uniformly distributed variables y_n and independence between y_n and x_n implies that $w_p^T x_n + b_p = 0$. Therefore, using the second-order expansion of the log-sum-exp function $\varphi(u) = \log(\sum_{p=1}^{P} \exp(u_p))$ around 0 leads to an approximation of the terms depending on (w, b), J_{wb} (up to an additive constant):

$$J_{wb}(\xi) = \frac{P}{2N}\mathrm{tr}(\xi\xi^T) - \frac{1}{2P}\min_{w,b}\Big[\frac{1}{N}\|(P\xi - Xw - b)\Pi_P\|_F^2 + \lambda\|w\|_F^2 + O(\|Xw+b\|_F^3)\Big],$$

where $\Pi_P = I - \frac{1}{P}1_P 1_P^T$ is the usual centering projection matrix, and $X = (x_1, \ldots, x_N)^T$. The third-order term $O(\|Xw+b\|_F^3)$ can be replaced by third-order terms in $\|\xi - \xi_0\|$, which makes the minimization with respect to w and b correspond to a multi-label classification problem with a square loss (Bach and Harchaoui, 2007; Hastie et al., 2001; Shawe-Taylor and Cristianini, 2004). Its solution may be obtained in closed form and leads to the second-order expansion of J_{wb}:

$$J_{wb}(\xi) \quad = \quad C_0 + \frac{P}{2N}\mathrm{tr}\Big[\xi\xi^T\big(I - A(X,\lambda)\big)\Big] + O(\|\xi - \xi_0\|^3),$$

where C_0 is a constant independent of ξ, and:

$$A(X,\lambda) = \Pi_N\Big(I - X(N\lambda I + X^T\Pi_N)^{-1}X^T\Big)\Pi_N.$$

Second-order Taylor expansion of $J_C(\xi)$. A simple calculation shows that $J_C(\xi)$ is given by (up to an additive constant)

$$J_C(\xi) = -\frac{P}{2N}\mathrm{tr}(\xi\xi^T) + O(\|\xi - \xi_0\|_F^3).$$

Second-order Taylor expansion of $J_\alpha(\xi)$. Denoting by $Z \in \mathbb{R}^{N\times L}$, the matrix with entries z_{nl} we obtain the expression (up to an additive constant and third order residuals $O(\|\xi - \xi_0\|_F^3)$):

$$J_\alpha(\xi) = \sum_{l=1}^{L}\frac{|A_l|}{N}\log(|A_l|) + \frac{P}{2N}\Big(\mathrm{tr}(\xi^T Z(Z^TZ)^{-1}Z^T\xi) - \frac{1}{N}\mathrm{tr}(\xi 1_n 1_n^T\xi)\Big),$$

since:

$$Z(Z^TZ)^{-1}Z^T = \sum_{l=1}^{L}\frac{1}{|A_l|}1_{A_l}1_{A_l}^T.$$

Quadratic approximation. Omitting the terms that are independent of ξ or of an order in ξ higher than two, the second-order approximation J_{app} of the function obtained for the M-step is:

$$J_{\text{app}}(\xi) \quad = \quad \frac{P}{2}\mathrm{tr}\Big[\xi\xi^T\Big(B(Z) - A(X,\lambda)\Big)\Big],$$

where:

$$B(Z) = \frac{1}{N}\Big(Z(Z^TZ)^{-1}Z^T - \frac{1}{N}1_N 1_N^T\Big).$$

Link with ridge regression. The first term, $\text{tr}(\xi\xi^T B(Z))$, is a concave function in ξ, whose maximum is obtained for $\xi\xi^T = I$. In terms of class assignment, the configuration equivalent to $\xi\xi^T = I$ is when each variable is in a different class.

The second term, $A(X, \lambda)$, is the matrix obtained in ridge regression (Bach and Harchaoui, 2007; Hastie et al., 2001; Shawe-Taylor and Cristianini, 2004). Since $A(x, \lambda)$ is a positive semi-definite matrix such that $A(X, \lambda)1_N = 0$, the maximum of the second term is obtained for $\xi\xi^T = 1_N 1_N^T$, which occurs when all the variables are in the same class. $J_{\text{app}}(\xi)$ is thus a combination of a term trying to put every point in the same class and a term trying to spread them equally.

Non linear predictions. Using the matrix inversion lemma, $A(X, \lambda)$ can be expressed in terms of the Gram matrix $K = XX^T$, which allows us to use any positive definite kernel in our framework (Shawe-Taylor and Cristianini, 2004), and tackle problems that are not linearly separable. Moreover, the square loss gives a natural interpretation of the regularization parameter λ in terms of the implicit number of parameters of the learning procedure (Hastie et al., 2001). Indeed, the *degree of freedom* defined as $df = N(1 - \text{tr}A)$ provides a simple method for setting the value of λ (Hastie et al., 2001).

Initialization of EM. We optimize $J_{\text{app}}(\xi)$ to get a robust initialization of the EM algorithm. Since the entries of each vector ξ_n sum to 1, we optimize J_{app} over a set of N simplices in P dimensions, $\mathcal{S} = \{v \in \mathbb{R}^P \mid v \geq 0,\ v^T 1_P = 1\}$. However, since this function is not convex, minimizing it directly leads to local minima. We propose, in Sec. 3.4, a general reformulation of any non-convex quadratic program (QP) over a set of N simplices and propose an efficient algorithm to optimize it.

3.3.3 Discriminative clustering

The goal of clustering is to find a low-dimensional representation of unlabeled observations, by assigning them to P different classes, Xu et al. (2005) proposes a discriminative clustering framework based on the SVM and (Bach and Harchaoui, 2007) simplifies it by replacing the hinge loss function by the square-loss, leading to ridge regression. By taking $L = N$ and the labels $Z = I$, we obtain a formulation similar to Bach and Harchaoui (2007) where we are looking for a latent representation that can recover the identity matrix. However, unlike (Xu et al., 2005; Bach and Harchaoui, 2007), our discriminative clustering framework is based on a probabilistic model, which may allow other extensions. This is a consequence of the logistic regression cost function. Moreover, our formulation naturally avoids putting all variables in the same cluster, whereas (Xu et al., 2005; Bach and Harchaoui, 2007) need to introduce constraints on the size of each cluster. Also, our model leads to a soft assignment of the variables, allowing flexibility in the shape of the clusters, whereas (Xu et al., 2005; Bach and Harchaoui, 2007)

is based on hard assignment. Finally, we obtain a natural rounding by applying the EM algorithm after the optimization whereas Bach and Harchaoui (2007) uses a coarse k-means rounding. Comparisons between these algorithms can be found Sec. 3.5.

3.4 Optimization of quadratic functions over simplices

To initialize the EM algorithm, we must minimize the *non-convex* quadratic cost function defined by Eq. (3.3) over a product of N simplices. More precisely, we are interested in problems of the general form:

$$\begin{aligned} \min_{V} \quad & f(V) = \tfrac{1}{2}\text{tr}\,(VV^TB) \\ \text{subject to} \quad & V = (V_1, \ldots, V_N)^T \in \mathbb{R}^{N\times P}, \\ & \forall n,\ V_n \in \mathcal{S}. \end{aligned} \tag{3.3}$$

where B can be any $N \times N$ symmetric matrix. Denoting $v = \text{vec}(V)$ the NP vector obtained by stacking all the columns of V into one vector and defining $Q = (B^T \otimes I_P)^T$, where $\otimes$ is the Kronecker product (Golub and Van Loan, 1996), problem (3.4) is equivalent to:

$$\begin{aligned} \min_{v} \quad & \tfrac{1}{2}\, v^TQv \\ \text{subject to} \quad & v \in \mathbb{R}^{NP}, \\ & v \geq 0, \\ & (I_N \otimes 1_P^T)v = 1_N. \end{aligned} \tag{3.4}$$

Note that this formulation is general, and that Q could be any $NP \times NP$ matrix.

Traditional convex relaxation methods (Anstreicher and Burer, 2005) would rewrite the objective function as $v^TQv = \text{tr}(Qvv^T) = \text{tr}(QT)$ where $T = vv^T$ is a rank-one matrix which satisfies the set of constraints:

$$- \quad T \in \mathcal{DN}_P = \{T \in \mathbb{R}^{NP\times NP} \mid T \geq 0,\ T \succcurlyeq 0\} \tag{3.5}$$

$$- \quad \forall\, n, m \in \{1, \ldots, N\},\ 1_P^T T_{nm} 1_P = 1, \tag{3.6}$$

$$- \quad \forall\, n, i, j \in \{1, \ldots, N\},\ T_{ni}1_P = T_{nj}1_P. \tag{3.7}$$

We note $\mathcal{F}$ the set of matrix T verifying (3.6-3.7). With the unit-rank constraint, optimizing over v is exactly equivalent to optimizing over T. The problem is relaxed into a convex problem by removing the rank constraint. As in Chapter 2, this leads to an optimization problem over positive matrices, i.e., a semidefinite programming problem (SDP) (Boyd and Vandenberghe, 2003).

Relaxation. Optimizing over T instead of v is computationally inefficient since the running time complexity of general purpose SDP toolboxes is in this case $O\big((PN)^7\big)$. On the other hand, for problems without pointwise positivity, (Burer, 2010; Journée et al., 2010) have considered low-rank representations of matrices T, of the form $T = VV^T$ where V has more than one column. In particular, Journée et al. (2010) show that the non convex optimization with respect to V leads to the global optimum of the relaxed convex problem in T.

In order to apply the same technique here, we need to deal with the pointwise non-negativity. This can be done by considering the set of *completely positive matrices*, i.e.,

$$\mathcal{CP}_P = \{T \in \mathbb{R}^{NP\times NP} | \exists p \in \mathbb{N}^*,\ \exists V \in \mathbb{R}^{NP\times R},\ V \geq 0,\ T = VV^T\}.$$

This set is *strictly* included in the set of matrices T which are both pointwise nonnegative and positive semi-definite (matrices often referred to as doubly nonnegative matrices). For $R \geq 5$, it turns out that the intersection of $\mathcal{CP}_P$ and $\mathcal{F}$ is the convex hull of the matrices vv^T such that v is an element of the product of simplices (Burer, 2010). This implies that the convex optimization problem of minimizing tr (QT) over $\mathcal{CP}_P \cap \mathcal{F}$ is equivalent to our original problem.

However, even if the set $\mathcal{CP}_P \cap \mathcal{F}$ is convex, optimizing over it is computationally inefficient (Berman and Shaked-Monderer, 2003). We thus follow Journée et al. (2010) and consider the problem in term of the low-rank pointwise nonnegative matrix $V \in \mathbb{R}^{NP\times R}$ instead of in term of matrices $T = VV^T$.

Note that following arguments from (Burer, 2010), if R is large enough, there are no local minima. However, because of the positivity constraint, one cannot find in polynomial time a local minimum of a differentiable function. Nevertheless, any gradient descent algorithm will converge to a stationary point. In Section 3.5, we compare results with $R > 1$ than with $R = 1$, which corresponds to a gradient descent directly on the simplex.

Problem reformulation. In order to derive a local descent algorithm, we reformulate the constraints (3.6-3.7) in terms of V. Denoting by V_r the r-th column of V, V_r^n the K-vector such as $V_r = (V_r^1, \ldots, V_r^N)^T$ and $V^n = (V_1^n, \ldots, V_R^n)$, condition (3.7) is equivalent to $\|V_r^m\|_1 = \|V_r^n\|_1$ for all n and m. Substituting this in (3.6) yields that for all n, $\|V^n\|_{2-1} = 1$, where $\|V^n\|_{2-1}^2 = \sum_{r=1}^R (1^T V_r^n)^2$ is the squared ℓ_{2-1} norm. We drop this condition by using a rescaled cost function, which is equivalent. Finally, defining by $\mathcal{D}$, the set of constraints:

$$\mathcal{D} = \{W \in \mathbb{R}^{NP} \mid W \geq 0,\ \forall n, m,\ \|W^n\|_1 = \|W^m\|_1\},$$

leads to a new formulation:

$$\begin{aligned} \min_{V} \quad & \tfrac{1}{2}\text{tr}(VD^{-1}V^TQ) \\ \text{subject to} \quad & V \in \mathbb{R}^{NP\times R}, \\ & \forall r,\ V_r \in \mathcal{D}, \\ & D = \text{diag}((I_N \otimes 1_P)^T VV^T (I_N \otimes 1_P)), \end{aligned} \tag{3.8}$$

where $\text{diag}(A)$ is the matrix with the diagonal of A and 0 elsewhere. Since the set of constraints defined in (3.9) is convex, we can use a projected gradient method (Bertsekas, 1995). In the next section, we propose a method to project any NP vector on the set $\mathcal{D}$.

Projection on $\mathcal{D}$. Given N P-vectors Z^n stacked in an NP vector $Y = [Y^1; \ldots; Y^N]$, we consider the projection of Y on $\mathcal{D}$ which is the solution of:

$$\begin{aligned} \min_{U} \quad & \tfrac{1}{2}\|U - Z\|_2^2 \\ \text{subject to} \quad & U \in \mathcal{D}. \end{aligned} \tag{3.9}$$

For a given positive real number a, the projection of Y on the set of all $U \in \mathcal{D}$ such that for all n, $\|U^n\|_1 = a$, is equivalent to N independent projections on the ℓ_1 ball with radius a. Thus projecting Y on $\mathcal{D}$ is equivalent to finding the solution of:

$$\begin{aligned} \min \quad & L(a) = \left[\sum_{n=1}^{N} \max_{\lambda_n \in \mathbb{R}} \min_{U^n \geq 0} \tfrac{1}{2}\|U^n - Y^n\|_2^2 + \lambda_n(1_P^T U^n - a)\right], \\ \text{subject to} \quad & a \geq 0, \end{aligned}$$

where the scalars $(\lambda_n)_{n\leq N}$ are Lagrange multipliers. The problem of projecting each Y^n on the ℓ_1-ball of radius a is well studied (Brucker, 1984; Maculan et al., 1989), with known forms for the optimal Lagrange multipliers $(\lambda_n(a))_{n\leq N}$ and the corresponding projection for a given a. It is straightforward to prove that the function $L(a)$ is convex, piecewise-quadratic and differentiable, which yields the first-order optimality condition $\sum_{n=1}^{N} \lambda_n(a) = 0$ for a. Several algorithms can be used to find the optimal value of a. For example, one can perform a descent gradient on a with updates:

$$a_{t+1} = a_t + \alpha_t \sum_{n=1}^{N} \lambda_n(a_t).$$

We prefer to use a binary search by looking at the sign of $\sum_{n=1}^{N} \lambda_n(a)$ on the interval $[0, \lambda_{max}]$, where λ_{max} is found iteratively. This method has been found to be empirically faster than gradient descent.

3.5 Implementation and results

We compare our algorithm and other classical methods to optimize the problem (3.9). We show that the performances are equivalent, but our algorithm can scale up to larger datasets. We also consider supervised and unsupervised multilabel classification. In both cases, we show that our algorithm outperforms existing methods.

Implementation. For supervised and unsupervised multilabel classification, we first optimize the second-order approximation J_{app}, using its reformulation (3.9). We use a projected gradient descent method with Armijo's rule along the projection arc for backtracking (Bertsekas, 1995). It is stopped after a maximum number of iterations (500) or when relative updates become too small (10^{-8}). When the algorithm stops, the matrix U has rank greater than 1 and we use the heuristic $u^* = \sum_{r=1}^{R} U_r \in \mathcal{S}$ as our final solution ("avg round"). We compare this rounding with another heuristic obtained by taking $u^* = \text{argmin}_{U_r} f(U_r)$ ("min round"). The value u^* is then used to initialize the EM algorithm described in Sec. 3.2.

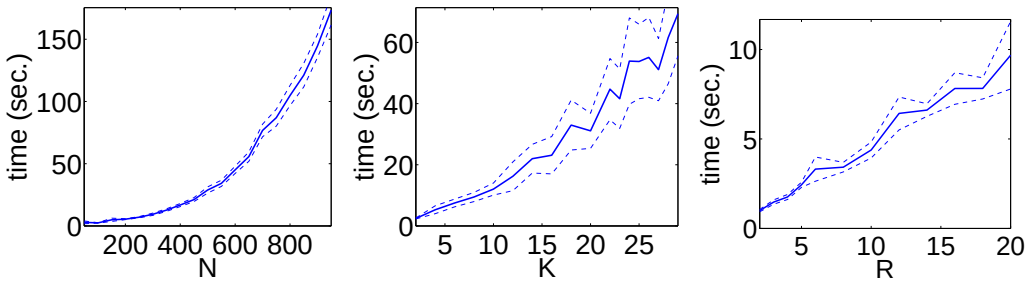


Figure 3.1: Running time as a function of N, P (K in this figure), and R.

Overall complexity and Running time. We use projected gradient descent. The bottleneck of our algorithm is the projection with a complexity of $O(RN^2P\log(P))$. Empirically, we have verified this complexity on a toy example. Results are shown in Figure 3.1. We show the running time of our algorithm on 50 random matrices Q obtained with a uniform distribution over $[0, 1]$ for increasing values of N, P, and R.

3.5.1 Optimization over simplices

We compare our optimization of the *non-convex* quadratic problem (3.9) in U, to the *convex* SDP in $T = UU^T$ on the set of constraints defined by $T \in \mathcal{DN}_P$, (3.6) and (3.7). To optimize the linear program, we use generic algorithms, CVX (Grant and Boyd,

2010) and PPXA (Combettes, 2004). CVX uses interior points methods whereas PPXA uses proximal methods (Combettes, 2004). Both algorithms are computationally inefficient and do not scale well in either the number of points or the number of constraints. Thus we set $N = 10$ and $P = 2$. We compare the performances of these algorithms *after* rounding. For the SDP, we take $\xi^* = T1_{NP}$ and for our algorithm we report performances obtained for both rounding discuss above ("avg round" and "min round"). On these small examples, our algorithm associated with "min round" reaches similar performances than the SDP ($f(\xi^*) = -1.9 \pm 0.2$), whereas, associated with "avg round", its performance drops ($f(\xi^*) = -0.92 \pm 0.85$).

3.5.2 Study of rounding procedures.

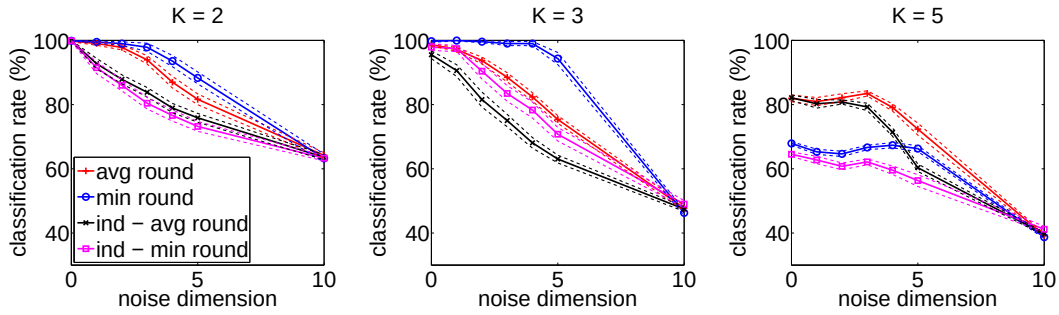


Figure 3.2: Comparison between our algorithm and R independent optimizations. Also comparison between two rounding: by summing and by taking the best column. Average results for $P = 2, 3, 5$ (K in this figure) (Best seen in color).

We compare the performances of the two different roundings, "min round" and "avg round" on discriminative clustering problems. After rounding, we apply the EM algorithm and look at the classification score. We also compare our algorithm for a given R, to two baselines where we solve independently problem (3.4), R times and then apply the same roundings ("ind - min round" and "ind - avg round"). Results are shown figure 3.2. Our setting is several two-dimensional clusters of points where we add dimensions of noise (see "Applications to discriminative clustering"). We consider three different problems, $N = 100$ and $P = 2$, $P = 3$ and $P = 5$. We take 50 different sets of points for each problem and for each of these configurations, we take 10 random initializations for our algorithm. We look at the average performances as the number of noise dimensions increases. Our method outperforms the baseline whatever rounding we use. Figure 3.2 shows that, on problems with a small number of latent classes ($P < 5$), we obtain better performances by taking the column associated with the lowest value of the cost function ("min round"), than summing all the columns ("avg round"). On the

other hand, when dealing with a larger number of classes ($P \geq 5$), the performance of "min round' drops significantly while "avg round" maintains good results. The reason is that summing the columns of U gives a solution close to $\frac{1}{P}1_N 1_P^T$ in expectation, thus in the region where our quadratic approximation is valid. Moreover, the best column of U is usually a local minimum of the quadratic approximation, which we have found to be close to similar local minima of our original problem, therefore, preventing the EM algorithm from converging to another solution. In all subsequent experiments, we choose "avg round".

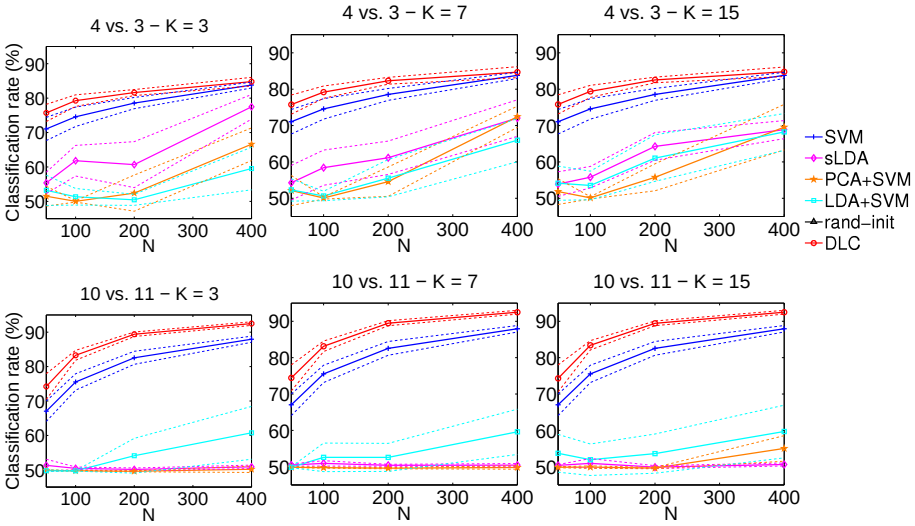


Figure 3.3: Classification rate for several binary classification tasks (from top to bottom) and for different values of P (K in this figure), from left to right (Best seen in color).

3.5.3 Application to classification

We evaluate the optimization performance of our algorithm (denoted DLC) on text classification tasks. For our experiments, we use the *20 Newsgroups* dataset[1], which contains postings to Usenet newsgroups. The postings are organized by content into 20 categories. We use the five binary classification tasks considered in (Lacoste-Julien, 2009, Chapter 4, page 91). To set the regularization parameter λ, we use the degree of freedom df (see Sec. 3.3.2). Each document has 13,312 entries, and we take $df = 1000$. We use 50 random initializations for our algorithm. We compare our method with classifiers such as the linear SVM and the supervised Latent Dirichlet Allocation (sLDA) classifier of Blei and Mcauliffe (2008). We also compare our results to those obtained by an SVM

[1] http://people.csail.mit.edu/jrennie/

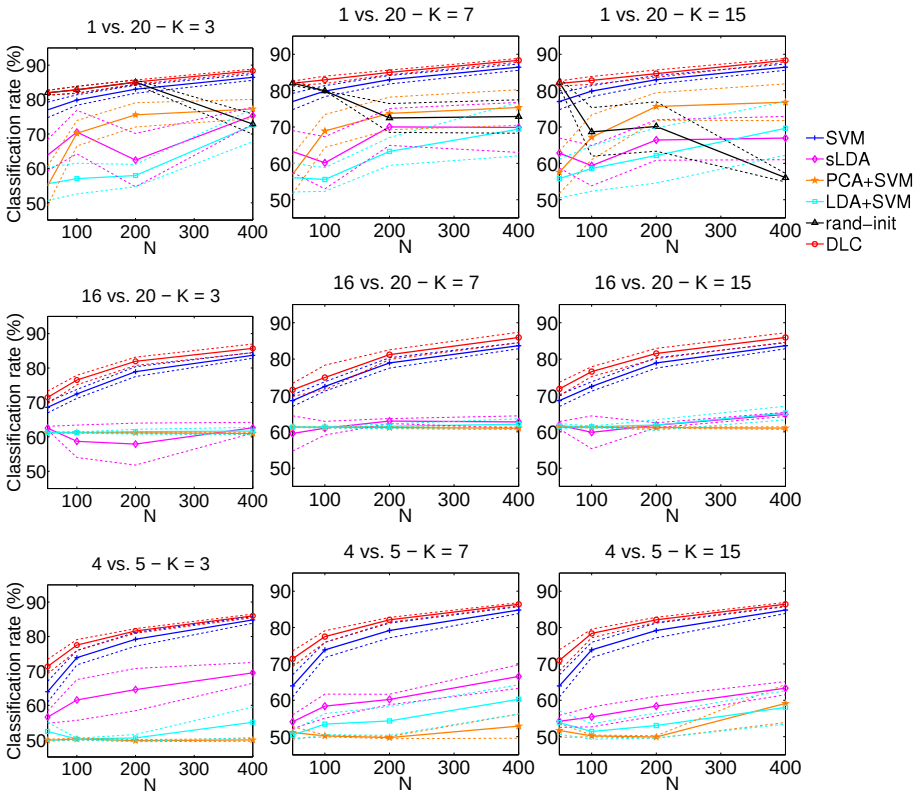


Figure 3.4: Classification rate for several binary classification tasks (from top to bottom) and for different values of P (K in this figure), from left to right (Best seen in color).

using the features obtained with dimension-reducing methods such as LDA (Blei et al., 2003) and PCA. For these models, we select parameters with 5-fold cross-validation. We also compare our method to the EM without our initialization ("rand-init") but also with 50 random initializations, a local descent method which is essentially equivalent to back-propagation in a two-layer neural network, which in this case strongly suffers from local minima problems. An interesting result on computational time is that EM without our initialization needs more steps to obtain a local minimum. It is therefore *slower* than with our initialization in this particular set of experiments. We show some results in Figure 3.3 and Figure 3.4 for different values of P and with an increasing number N of training samples. In the case of topic models, P represents the number of topics. Our method significantly outperforms all the other classifiers. The comparison with "rand-init" shows the importance of our convex initialization. We also note that our performance increases slowly with P. Indeed, the number of classes needed to correctly separate two classes of text is small. The algorithm tends to automatically select P.

Empirically, we notice that starting with $P = 15$ classes, our average final number of active classes is around 3. This explains the relatively small gain in performance as P increases.

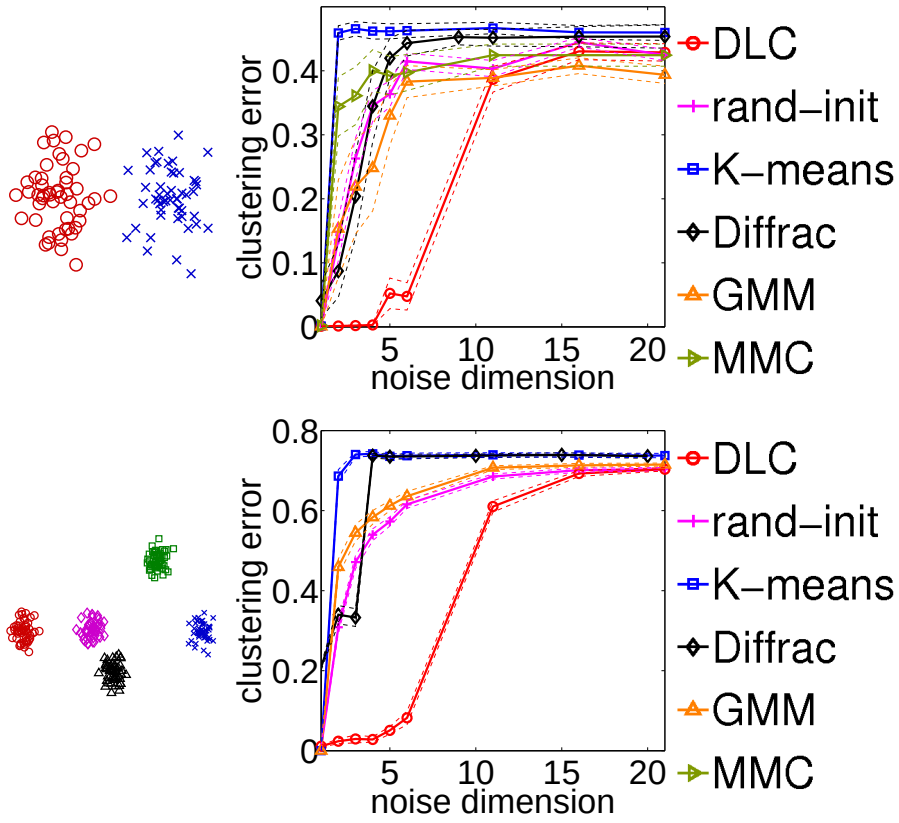


Figure 3.5: Clustering error when increasing the number of noise dimensions. We have take 50 different problems and 10 random initializations for each of them. $P = 2$, $N = 100$ and $R = 5$ (on the left) and $P = 5$, $N = 250$ and $R = 10$ (on the right).

3.5.4 Application to discriminative clustering

Figure 3.5 shows the optimization performance of the EM algorithm with 10 random starting points with ("DLC") and without ("rand-init") our initialization method. We compare their performances to K-means, a Gaussian mixture model (GMM), Diffrac (Bach and Harchaoui, 2007) and max-margin clustering (MMC) (Zhang et al., 2007b). Following the experimental setting of Bach and Harchaoui (2007), we take linearly separable bumps in a two-dimensional space and add dimensions containing random independent Gaussian noise (e.g. "noise dimension") to the data. We evaluate the ratio

of misclassified observations over the total number of observations. For the first experiment, we fix $P = 2$, $N = 100$, and $R = 5$, and for the second $P = 5$, $N = 250$, and $R = 10$. The additional independent noise dimensions are normally distributed. We use linear kernels for all the methods. We set the regularization parameters λ to 10^{-2} for all experiments but we have seen that results do not change much as long as λ is not too small ($> 10^{-8}$).
Note that we do not show results for the MMC algorithm when $P = 5$ since this algorithm is specially designed for problems with $P = 2$. It would be interesting to compare to the extension for multi-class problems proposed by Zhang et al. (2007b). On both examples, we are significantly better than Diffrac, k-means and MMC.

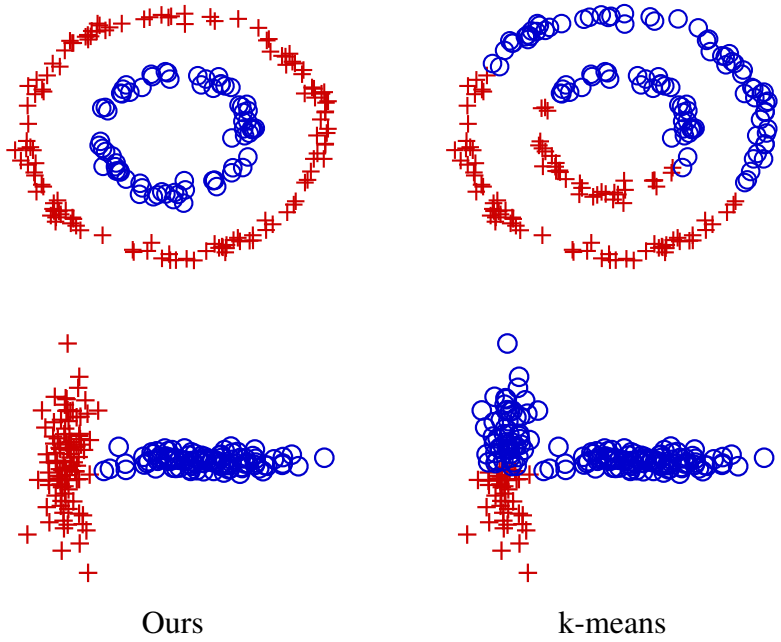


Figure 3.6: Comparison between our method (left) and k-means (right) on different problems. First, circles with RBF kernels. Secondly linearly separable bumps. $P = 2$, $N = 200$ and $R = 5$ in both cases (best seen in color).

Finally we show in Figure 3.6 additional examples to illustrate that our method compares favorably with k-means in different situations. The first row shows an example where linear separation is impossible and a non linear transformation such as kernels, must be used. THis example illustrates that our method works well with non linear classifier, in that particular case, an rbf kernel. The second row shows an example where a linear classifier is used to seperate clusters with different shapes. This example shows

that with linear kernels our method is more flexible that k-means since as it is discriminative, it works regardlessly of the cluster shapes.

4

Multi-Class Cosegmentation

Abstract of this chapter: The availability of multiple images assumed to contain instances of the same object classes provides a weak form of supervision that can be exploited by discriminative approaches to segmentation. Unfortunately, most algorithms are limited to a very small number of images and/or object classes (typically two of each). This chapter proposes a novel energy-minimization approach to cosegmentation that can handle multiple classes and a significantly larger number of images. The proposed cost function combines spectral- and discriminative-clustering terms. The discriminative-clustering terms are similar to the one presented in Chapter 3. Our framework thus admits a probabilistic interpretation. It is optimized using an efficient EM method, initialized using a convex quadratic approximation of the energy. Comparative experiments show that the proposed approach matches or improves the state of the art on several standard datasets.

The material of this chapter is based on the following work:

A. Joulin, F. Bach and J. Ponce. Multi-Class Cosegmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

4.1 Introduction

As mention in the introduction of this thesis, the objective of image segmentation is to divide a picture into $P \geq 2$ regions that are deemed meaningful according to some objective criterion, homogeneity in some feature space or separability in some other one for example. Segmentation in the absence of any supervisory information remains a daunting challenge. On the other hand, when supervisory information is available, in the form of labelled training data (full images or, in interactive settings, smaller groups

of pixels), accurate segmentations can be achieved (Blake et al., 2004). We give a brief definiton of cosegmentation and we refer to Chapter 2 for a more detailed explanation: the aim of *cosegmentation* methods is to simultaneously divide a set of images assumed to contain instances of P different object classes into regions corresponding to these classes.

In the field of cosegmentation, Kim et al. (2011) have proposed the first method explicitly aimed at handling multiple object classes and images. They maximize the overall temperature of image sites associated with a heat diffusion process and the position of sources corresponding to the different object classes. They use a greedy procedure guaranteed to achieve a local minimum within a fixed factor of the global optimum thanks to submodularity properties of the diffusion process (Kim et al., 2011). We present in this chapter an effective energy-based alternative that combines a spectral-clustering term (Shi and Malik, 1997) with a discriminative one (Joulin et al., 2010b), and can be optimized using an efficient expectation-minimization (EM) algorithm. Our energy function is not convex and, like Kim et al. (2011), we can only hope to find a local minimum. Fortunately, a satisfactory initialization can be obtained by constructing a convex quadratic relaxation closely related to the cost function proposed in the two-class case in Chapter 2 (Joulin et al., 2010b).

The proposed approach has been implemented and tested on several datasets including video sequences. It easily handles multiple object classes and input images, and compares favorably to Kim et al. (2011) and a simple multi-class extension of the method presented in Chapter 2 in a comparative evaluation on two standard benchmarks. Furthermore, unlike the methods proposed by Kim et al. (2011) and our previous method, ours admits a probabilistic interpretation, with the potential to be easily combined with other components of an end-to-end recognition system.

4.2 Proposed model

Cosegmentation can be thought of as a multi-label pixel classification task. As in Chapter 2, cosegmentation is modeled in this chapter as the minimization over the pixel labels of an energy function that combines local appearance and spatial consistency terms (as in spectral clustering) with class-level discriminative ones (as in discriminative clustering, Joulin et al. (2010b); Xu et al. (2005)) and a cluster size balancing term.

Image representation. In this chapter we suppose that P, the number of object classes, can be greater than 2. As is common in the cosegmentation setting, P is assumed in the following to be fixed and known a priori. As previously, we denote by y_n the label of

the pixel n. Given the set $\mathcal{I}$ of images, our goal is thus to find y without any other prior information.

As noted above, the idea of cosegmentation is to divide each image into P visually and spatially consistent regions while maximizing class separability across images. As in Chapter 2, we deal with the first problem by using unsupervised spectral-clustering methods such as *normalized cuts* (Shi and Malik, 1997) with little or no sharing of information between different images. The second one leads to multi-class discriminative clustering methods with information shared among images. As in Chapter 2, we combine the two approaches. However, generalizing the two-class (foreground/background) model to the multi-class setting leads to a completely different approach to discriminative clustering. Our overall energy function is the sum of spectral- and discriminative-clustering terms, plus a regularizer enforcing class-size balance (naturally introduced by the model presented in the previous chapter). We now detail these three terms.

4.2.1 Spectral clustering

We enforce visual and spatial consistency as in Chapter 2. We follow Section 2.2.1 and recall formulas here just for clarity.

The similarity matrix W^i is based on feature positions p_n and color vectors c_n, which leads to high similarity for nearby pixels with similar colors. Concretely, for any pair (n, m) of pixels in i, W^i_{nm} is given by:

$$W^i_{nm} = \exp(-\lambda_p \|p_n - p_m\|_2^2 - \lambda_c \|c_n - c_m\|^2)$$

if $\|p_n - p_m\|_1 \leq 2$ and 0 otherwise. We denote by W the $N \times N$ block-diagonal matrix obtained by putting the blocks $W^i \in \mathbb{R}^{N_i \times N_i}$ on its diagonal, and by $L = I_N - D^{-1/2} W D^{-1/2}$ the *Laplacian* matrix. We thus include the following quadratic term into our objective function:

$$E_B(y) \quad = \quad \frac{\mu}{N} \sum_{i \in \mathcal{I}} \sum_{n,m \in \mathcal{N}_i} \sum_{k=1}^{K} y_{np} y_{mp} L_{nm}, \tag{4.1}$$

where μ is a free parameter. This term encourages an *independent* segmentation of the images into different groups, based solely on local features.

4.2.2 Discriminative clustering

The goal of discriminative clustering is to find the pixel labels y that minimize the value of a regularized discriminative cost function (Xu et al., 2005). More precisely, given some labels y and some feature map Φ, a multi-class discriminative classifier finds the

optimal parameters $A \in \mathbb{R}^{P\times d}$ and $b \in \mathbb{R}^K$ that minimize

$$E_U(y, A, b) = \frac{1}{N}\sum_{n=1}^{N} \ell(y_n, A\phi(x_n) + b) + \frac{\lambda}{2P}\|A\|_F^2, \tag{4.2}$$

where $\ell : \mathbb{R}^P \times \mathbb{R}^P \mapsto \mathbb{R}$ is a loss function, y_n is the n-th column of y^T, and $\|A\|_F$ is the Frobenius norm of A. A discriminative-clustering method minimizes the response of the classifier over the set $\mathcal{Y}$ of labels, i.e, it solves:

$$\begin{aligned} \min \quad & E_U(y, A, b), \\ \text{subject to} \quad & y \in \{0, 1\}^{N\times P}, \\ & y1_P = 1_N, \\ \text{and} \quad & A \in \mathbb{R}^{P\times d}, \\ & b \in \mathbb{R}^P. \end{aligned}$$

Different choices for the loss function ℓ lead to different algorithms. In the two-class case, we have used in Chapter 2 the square loss, which has the advantage of leading to a convex problem that can be solved efficiently, but is not adapted to the multi-class setting (this is related to the *masking problem*, see Section 4.2 in Hastie et al. (2001)). In this chapter we use instead the soft-max loss function defined as:

$$\ell(y_n, A, b) = -\sum_{k=1}^{K} y_{np} \log\left(\frac{\exp(a_p^T\phi(x_n) + b_p)}{\sum_{l=1}^{L}\exp(a_l^T\phi(x_n) + b_l)}\right),$$

where a_p^T is the p-th row of A, and b_p the p-th entry of b. This loss is well adapted to the multi-class setting, and it encourages a soft assignement of the pixels to the different classes Hastie et al. (2001).

Mapping approximation. As in Chapter 2, we use a χ^2-kernel and an incomplete Cholesky decomposition to approximate the mapping. Once again, we recall the formula for clarity.

Since our features are histograms, we use the χ^2-kernel defined by

$$K_{nm} = \exp\left(-\lambda_h \sum_{d=1}^{D} \frac{(x_{nd} - x_{md})^2}{x_{nd} + x_{md}}\right),$$

where $\lambda_h > 0$ (in the experiments, we use $\lambda_h = 0.1$).

In the case where K is known but Φ is not, a common trick is to construct an incomplete Cholesky decomposition (Shawe-Taylor and Cristianini, 2004) of K—that is, calculate a matrix $\psi \in \mathbb{R}^{N\times d}$ such that $\psi\psi^T \approx K$, then directly use Eq. (4.2), where $\Phi(x_n)$ has been replaced by ψ_n, where ψ_n^T is the n-th row of ψ. With a slight abuse of notation, we still use $\Phi(x_n) = \psi_n$ to denote the approximated mapping in the rest of this presentation.

4.2.3 Cluster size balancing

A classical problem with spectral- and discriminative-clustering methods is that assigning the same labels to all the pixels leads to perfect separation. A common solution is to add constraints on the number of elements per class (Xu et al., 2005; Bach and Harchaoui, 2007). Despite good results, this solution introduces extra parameters and is hard to interpret. Another solution is to encourage the proportion of points per class and per image to be close to uniform. An appropriate penalty term for achieving this is the entropy of the proportions of points per image and per class:

$$H(y) = -\sum_{i\in\mathcal{I}}\sum_{p=1}^{P}\left(\frac{1}{N}\sum_{n\in\mathcal{N}_i} y_{np}\right)\log\left(\frac{1}{N}\sum_{n\in\mathcal{N}_i} y_{np}\right). \tag{4.3}$$

As shown later, there is a natural interpretation that allows us to set the parameter in front of this term to 1.

Weakly supervised segmentation. Cosegmentation can be seen as a "very weakly" supervised form of segmentation, where one knows that P object classes occur in the images, but not which ones of the P do occur in a given image. Indeed, our entropy term encourages (but does not force) every class to occur in every image. Our framework is easily extended to *weakly supervised segmentation*, where tags are attached to each image i, specifying the set P_i of object classes appearing in it: This simply requires replacing the sum over indices p varying from 1 to P in Eq. (4.3) by a sum over indices p in P_i. For any pixel n in image i, this naturally encourages y_{np} to be zero for any p nor in P_i.

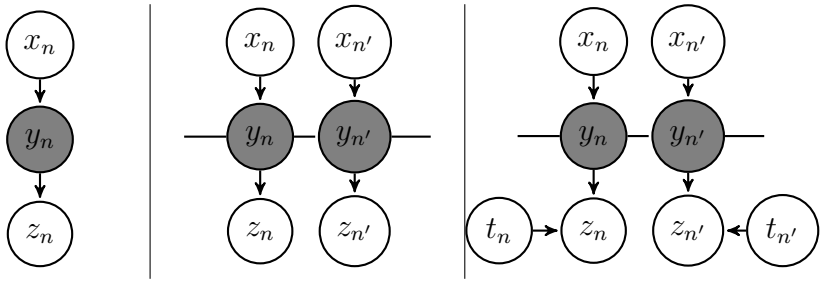


Figure 4.1: Left: Original model. Middle: Spatially consitent hidden classes. Right: Adding the image indicator t_n for each node n. For clarity, we represent our planar graph in two dimensions as a one-dimensional line in this diagram.

4.2.4 Probabilistic interpretation

Combining the three terms defined by Eqs. (4.1)–(4.3) we finally obtain the following optimization problem:

$$\begin{aligned} \min \quad & E_B(y) - H(y) + \left[\min_{\substack{A \in \mathbb{R}^{d \times P}, \\ b \in \mathbb{R}^P}} E_U(y, A, b) \right] \\ \text{subject to} \quad & y \in \{0,1\}^{N \times P}, \\ & y 1_P = 1_N. \end{aligned} \tag{4.4}$$

Let us show that the labels y can be seen as latent variables in a directed graphical model similar to the one exposed in Chapter 3 (Wainwright and Jordan, 2008). First, we introduce a variable z_n in $\{1, \dots, L\}$ giving for each pixel n some *observable information*, e.g., some information about its true label or its relation with other pixels. The relations between this observable partial label, the true label and the feature are the same as in Chapter 3 and shown on the left panel of Figure 4.1. Second, since we work on images, a natural extension is to connect the latent variables according to the graph defined by an image, as on the middle panel of Figure 4.1. Finally, for each pixel n, we introduce a variable t_n in $\{0,1\}^{|\mathcal{I}|}$ indicating to which image n belongs The resulting directed graphical model as shown Figure 4.1 defines the label y as a latent variable of the observable information z given x, as shown on the right panel of Figure 4.1. Given some pixel n, this model induces an "explain away" phenomenon: the label y_n and the variable t_n compete to explain the observable information z_n. This model can be seen as an extension of topic models (Blei et al., 2003; Cao and Fei-Fei, 2007) where the labels y represent *topics* which explain the *document* z given the *words* x, independently of the *group of documents* t from which z has been taken. More precisely, we suppose a bilinear model:

$$P(z_{nl} = 1 \mid t_{ni} = 1,\ y_{np} = 1) \quad = \quad y_{np} G_m^{ip} t_{ni},$$

where $\sum_{m=1}^{N} G_m^{ip} = 1$, and we show below that the problem defined by Eq. (4.4) is equivalent to the mean-field variational approximation of the following (regularized) negative conditional log-likelihood of $Y = (y_1, \dots, y_N)$ given $X = (x_1, \dots, x_N)$ and $T = (t_1, \dots, t_n)$ for our model:

$$\begin{aligned} \min \quad & -\frac{1}{N} \sum_{n=1}^{N} \log \left(p(y_n \mid x_n,\ t_n) \right) + \frac{\lambda}{2P} \|A\|_2^2, \\ \text{subject to} \quad & A \in \mathbb{R}^{d \times P},\ b \in \mathbb{R}^P, \\ & G \in \mathbb{R}^{N \times P |\mathcal{I}|}, \\ & G^T 1_N = 1, \\ & G \geq 0. \end{aligned}$$

The introduction of the variable z makes our model suitable for a a semi-supervised setting where z would encode "must-link" and "must-not-link" constraints between pixels. This may prove particularly useful when superpixels are used, since it is equivalent to adding "must-link" constraints between pixels belonging to the same superpixel (in this case, L is the total number of superpixels).

Proof of the equivalence. The entropy $H(y)$ defined previously can be rewritten as:

$$H(y) = \max_{\substack{G\in\mathbb{R}^{N\times P|\mathcal{I}|} \\ G^T 1_N = 1,\ G\geq 0}} \frac{1}{N} \sum_{i\in\mathcal{I}} \sum_{p=1}^{P} \sum_{n\in\mathcal{N}_i} \Big[t_{ni} y_{np} \log(G_n^{pi}) - y_{np} \log(y_{np}) \Big].$$

Also since $\min f(x) = -\max -f(x)$, our problem can be reformulated as the maximum of a, b and G of the following function:

$$\begin{aligned} J(a,b,G,y) &= \frac{1}{N} \Bigg[- E_U(y,a,b) - E_B(y) \\ &\quad + \sum_{p=1}^{P} \sum_{i\in\mathcal{I}} \sum_{n\in\mathcal{N}_i} y_{np} \log(G_n^{pi}) - \sum_{p=1}^{P} \sum_{n=1}^{N} y_{np} \log(y_{np}) \Bigg]. \end{aligned}$$

Denoting by $A(x_n,a,b) = \log(\sum_{p=1}^{P} \exp(a_p \Phi(x_n) + b_p))$ the log-partition, the function $J(a,b,G,y)$ can be rewritten as:

$$\begin{aligned} J(a,b,G,y) &= \frac{1}{N} \sum_{i\in\mathcal{I}} \sum_{n\in\mathcal{N}_i} \Bigg[\sum_{p=1}^{P} y_{np} \log(G_n^{pi} \exp(a_p \Phi(x_n) + b_p - A(x_n,a,b))) \\ &\quad - y_{np} \log(y_{np}) \Bigg] - E_B(y). \end{aligned}$$

Let us first suppose that there is no binary term in y, e.g., $E_B(y) = 0$. In this case, $J(a,b,G,y)$ is simply the auxiliary function associated with the EM procedure thus we have:

$$\ell(a,b,G) = \min_{\substack{y\in\{0,1\}^{N\times P}, \\ y 1_P = 1_N,\ y\geq 0}} J(a,b,G,y)$$

where:

$$\begin{aligned}\ell(a,b,G) &= \frac{1}{N}\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}_i}\log\left(\sum_{p=1}^{P}G_n^{pi}\exp(a_p\Phi(x_n)+b_p-A(x_n,a,b))\right)\\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{i\in\mathcal{I}}\log(\sum_{k=1}^{K}p(y_n\mid t_n=i,\ z_n=p)p(z_n=p\mid x_n))\\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{i\in\mathcal{I}}\log(p(y_n\mid t_n=i,\ x_n)),\end{aligned}$$

which is the conditional log-likelihood of Y given X.

The binary term in y, $E_B(y)$, is exactly the meanfield approximation of a classical binary term of a Markov Random Field:

$$E_{MRF}(y)=\sum_{n,m}\sum_{p=1}^{P}\sum_{l=1}^{L}y_{nm}(p,l)L_{nm}(p,l)$$

where $y_{nm}(p,l)$ are the marginals over the edge (n,m). Indeed the meanfield approximation replace the marginals over the edges by the product of the marginals over the nodes (thus making the assumption that the nodes are independent).

Also, in the presence of a binary term, the log-partition $\prod_n A(a,b,G)$ is the low bound the following log-partition:

$$A_{MRF}=\log\left(\sum_{y\in\mathcal{Y}}\exp\left(\sum_{n=1}^{N}y_n(a_p\Phi(x_n)+b_p)+\sum_{n,m}y_{nm}L_{nm}\right)\right),$$

where $\mathcal{Y}$ represents the set of all possible marginals over the graph. Thus our cost function is exactly the auxiliary function obtained with both a meanfield approximation and the previous approximation of our log-partition of the following log-likelihood:

$$\ell(a,b,G)=\frac{1}{N}\sum_{n=1}^{N}\sum_{i\in\mathcal{I}}\log(p(y_n\mid t_n=i,\ x_n)).$$

4.3 Optimization

We now present a non-convex relaxation of our combinatorial problem, which leads to an optimization scheme based on an expectation-maximization (EM) procedure, that can itself be initialized by efficiently solving a convex optimization problem closely related to the one given in Chapter 2.

4.3.1 EM algorithm

We use a continuous relaxation of our combinatorial problem, replacing the set of possible y values by the convex set $\mathcal{Y} = \{y \in [0,1]^{N\times P} \mid y1_P = 1_N\}$. In this setting, y_{np} can be interpreted as the probability for the n-th point to be in the p-th class. Our cost function is a difference of convex functions, which can be optimized by either *difference-of-convex* (DC) programming (Yuille and Rangarajan, 2003) or a block-coordinate descent procedure. We choose the latter, and since our energy is closely related to a probabilistic model, dub it an EM procedure with a slight abuse of notation.

M-step. For some given value of y, minimizing $E_U(y, A, b)$ in terms of (A, b) is a (convex) softmax regression problem which can be solved efficiently by a quasi-Newton method such as L-BFGS (Liu and Nocedal, 1989).

E-step. For given A and b, the cost function of Eq. (4.4) is convex in $y \in \mathcal{Y}$, and can thus be minimized with a simple projected gradient descent method on $\mathcal{Y}$. This first-order optimization method is slower than the second-order one used in the M-step, and it is the bottleneck of our algorithm, leading us to use superpixels for improved efficiency.

Superpixels. We oversegment every image i into $\mathcal{S}_i$ superpixels. For a given image i, this is equivalent to forcing every pixel n in $\mathcal{N}_i$ in a superpixel s to have the same label $y_n = y_s$. Denoting by $|s|$ the number of pixels contained in a superpixel s, each term of our cost function depending directly on y is reduced to:

$$\begin{cases} E_U(y) &= \frac{1}{N}\sum_{s\in\mathcal{S}} y_s(A\Phi_s + |s|b), \\ E_B(y) &= \frac{\mu}{2}\sum_{i\in\mathcal{I}}\sum_{s,t\in\mathcal{S}_i^2} y_{sp}y_{tp}\Lambda_{st}, \end{cases}$$

where $E_U(y)$ is the part of $E_U(y, A, b)$ depending on y, $\Phi(s) = \sum_{n\in s}\phi(x_n)$, and $\Lambda_{st} = \sum_{n\in s}\sum_{m\in t} L_{nt}$. The entropy has the form:

$$H(y) = -\sum_{i\in\mathcal{I}}\sum_{p=1}^{P}\left(\frac{1}{N}\sum_{s\in\mathcal{S}_i}|s|y_{sp}\right)\log\left(\frac{1}{N}\sum_{s\in\mathcal{S}_i}|s|y_{sp}\right).$$

Since the problem defined by Eq. (4.4) is not jointly convex in (A, b) and y, a reasonable initial guess is required. In the next section, we propose a convex approximation of our cost function that can be used to compute such a guess. Moreover, we show that this approximation is closely related to the the cost function proposed in Chapter 2. This allows us to use a modified version of the algorithm proposed in Chapter 2 to initialize ours.

4.3.2 Quadratic relaxation

The quadratic relaxation proposed in this section is closely related to the one developed in Chapter 3: cosegmentation is characterized by the lack of prior information on the classes present in the images. A reasonable initial guess for our model parameters is thus to assume a uniform distribution y_n^0 of the classes over each pixel n, and to predict a pixel's class using a linear model whose parameters are independent of the corresponding feature value, which is easily shown to be equivalent to

$$\ell(y_n^0, 0) = \sum_{k=1}^{K} \frac{1}{P} \log(P).$$

Any approximation of our cost function that might be used to initialize our algorithm should be related in some way to this configuration. As in Chapter 3, we approximates our cost function by its second-order Taylor expansion around y^0:

$$J(y) = \frac{P}{2}\left[\text{tr}(yy^T C) + \frac{2\mu}{NP}\text{tr}(yy^T L) - \frac{1}{N}\text{tr}(yy^T \Pi_I)\right], \tag{4.5}$$

where $\Pi_I = I_N - \Lambda$, and Λ is the $N \times N$ block diagonal matrix where there is a block equal to $\frac{1}{N_i} 1_{N_i} 1_{N_i}^T$ for each image i. Note that the projection matrix Π_I centers the data for each image independently. Finally, the matrix C in Eq. (4.5) is equal to:

$$C = \frac{1}{N}\Pi_N (I - \Phi(N\lambda I_P + \Phi^T \Pi_N \Phi)^{-1} \Phi^T) \Pi_N,$$

where the projection matrix $\Pi_N = I - \frac{1}{N} 1_N 1_N^T$ centers the data *across all images*. Note that C is closely related to the solution of the ridge regression (or Tikhonov regularization) of y over Φ.

The first two terms in Eq. (4.5) add up to the cost function used in Chapter 2 (up to a multiplicative constant). As in Chapter 3, the last term is a non-convex quadratic penalization encouraging a uniform distribution over classes on each image. We replace it (during initialization only) by linear constraints that force the pixels in any class k to represent at most 90% of the pixels in each image i, and at least 10% of the pixels in all other images:

$$\begin{aligned} \sum_{n \in \mathcal{N}_i} y_{np} &\leq 0.9 N_i \\ \sum_{j \in \mathcal{I} \setminus i} \sum_{n \in \mathcal{N}_j} y_{np} &\geq 0.1 (N - N_i). \end{aligned}$$

These constraints generalize those used in Chapter 2 to the multi-class case, and using them has the added benefit of allowing us to use a slightly modified version of their

publicly available software.[1] However, the output of this code is the $N \times N$ matrix $Y = yy^T$ and not y, thus a rounding step is necessery to initialize the new algorithm. The standard approach to this kind of problem is to use either *k-means* or a Gaussian mixture model (GMM) over the eigenvectors associated with the P highest eigenvalues (Ng et al., 2001) for this purpose.

Practical issues. Initializing our algorithm with the convex approximation proposed in this section usually leads to good results, but sometimes fails completely, due to the masking problem mentioned earlier. Therefore, we also start our EM procedure with five random initializations. We compare the final values of our cost function obtained from these initializations, and pick the solution associated with the lowest value as our result. An effective rounding procedure is also a key to good performance. Thus, we perform both the k-means and GMM rounding procedures, run one M-step for each of the corresponding initializations, and run the rest of the algorithm with the one yielding the lowest value of the cost function.

4.4 Implementation and results

4.4.1 Experimental set-up

We use the watershed algorithm (Wright et al., 1997) to find superpixels. The rest of our algorithm is coded in MATLAB. Since a good initialization is crucial, we use a modified version of the method presented in Chapter 2 to initialize our method as explained in Section 4.3.2. The complexity of our algorithm is $O(NP)$, and its running time typically varies from 30mn to one hour for 30 images, depending on the number of superpixels (this could be improved using a C implementation and exploiting the fact that parts of our algorithm are easily parellelized).

We present qualitative multi-class cosegmentation results on various datasets in the rest of this section. We also present quantitative comparisons with Kim et al. (2011)[2], Mukherjee et al. (2011) and the method presented in Chapter 2 on two standard benchmarks, MSRC-v2[3] and iCoseg (Batra et al., 2010).[4] We use the publicly available versions of (Kim et al., 2011) and set their free parameters so as to maximize their performance for the sake of fairness. Likewise, we set the free parameter μ of our algorithm by trying $\mu = 10^k$ for $k \in \{0, \dots, 4\}$, and keeping the value leading to the best performance (taking $\mu = 0.1$ works well in all our experiments in practice).

[1] http://www.di.ens.fr/ joulin/
[2] http://www.cs.cmu.edu/∼gunhee/r_seg_submod.html
[3] http://research.microsoft.com/en-us/projects/ObjectClassRecognition/
[4] http://chenlab.ece.cornell.edu/projects/touch-coseg/

The images in iCoseg only have two labels, and MSRC is not well suited to a multi-class evaluation because of its "clutter" class that does not correspond to a well-defined visual category. We have thus used the main "object" category for each MSRC image as foreground, and the rest of the pixels as background, and limited our quantitative evaluation to the binary case. Segmentation performance is measured by the *intersection-over-union* score that is standard in PASCAL challenges and defined as $\max_p \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \frac{GT_i \cap R_i^p}{GT_i \cup R_i^p}$, where GT_i is the ground truth and R_i^p the region associated with the p-th class in the image i.

In average, there are 30 images per class in MSRC and between 5 to 50 images per class on iCoseg.

images	class	Ours	Kim et al. (2011)	Chapter 2	Mukherjee et al. (2011)
30	Bike	**43.3**	29.9	42.3	42.8
30	Bird	**47.7**	29.9	33.2	-
30	Car	**59.7**	37.1	59.0	52.5
24	Cat	**31.9**	24.4	30.1	5.6
30	Chair	**39.6**	28.7	37.6	39.4
30	Cow	**52.7**	33.5	45.0	26.1
26	Dog	**41.8**	33.0	41.3	-
30	Face	**70.0**	33.2	66.2	40.8
30	Flower	**51.9**	40.2	50.9	-
30	House	51.0	32.2	50.5	**66.4**
30	Plane	21.6	25.1	21.7	**33.4**
30	Sheep	**66.3**	60.8	60.4	45.7
30	Sign	**58.9**	43.2	55.2	-
30	Tree	**67.0**	61.2	60.0	55.9
	Average	**50.2**	36.6	46.7	40.9

Table 4.1: Binary classification results on MSRC. Best results in bold.

4.4.2 MSRC two-class experiments

Qualitative results obtained on the MSRC-v2 database with two classes are shown in Figure 4.2 and Figure 4.3. Table 4.1 gives a quantitative comparison with (Kim et al., 2011; Mukherjee et al., 2011) and the method presented in Chapter 2.[5] Note that the algorithm proposed in Mukherjee et al. (2011) fails to converge on 5 out of 14 classes. Our algorithm achieves the best performance for 12 out of 14 object classes. We use

[5]There is no error bar since we test on the maximum number of images per class.

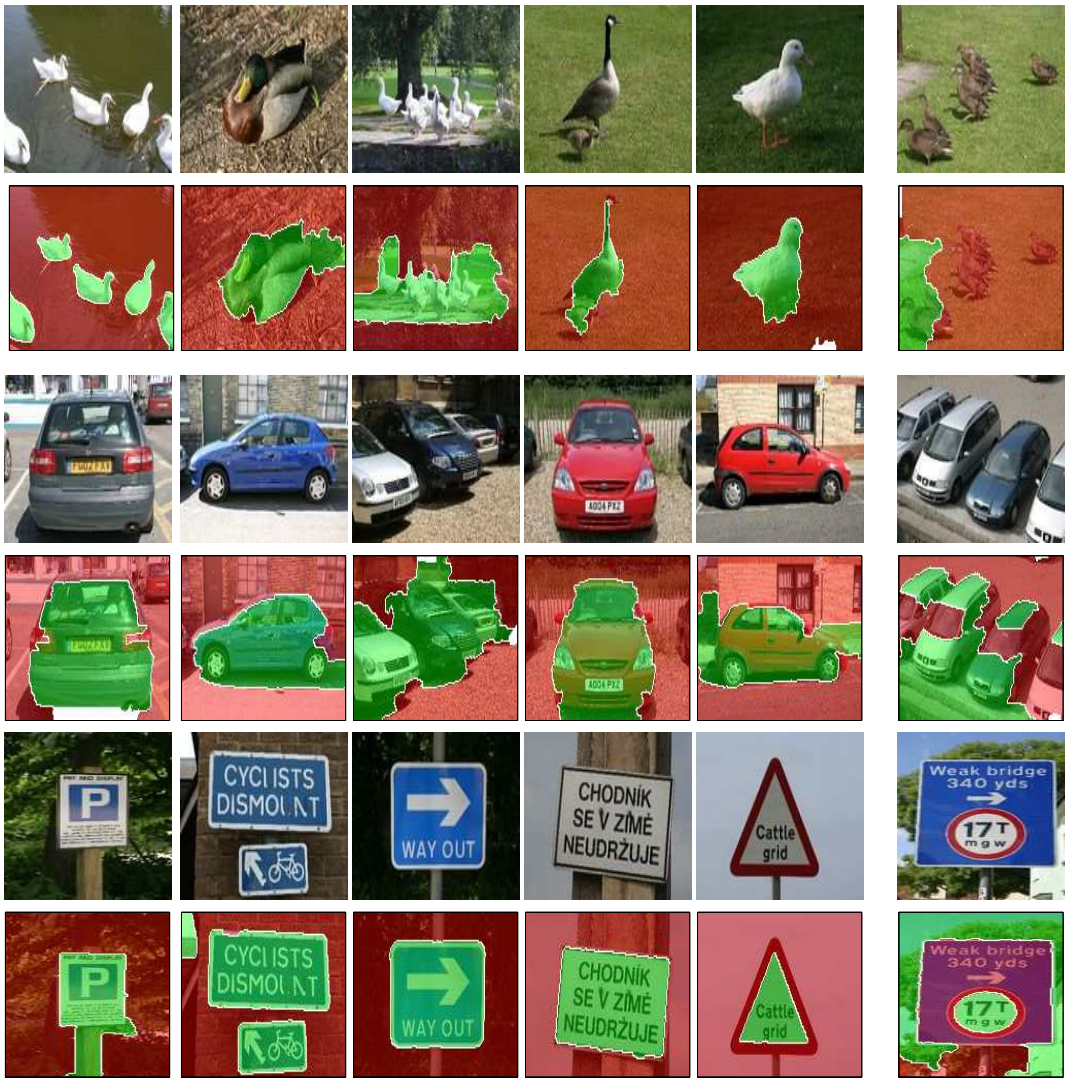


Figure 4.2: Results on binary classification. There are one set of images by row, the last example is a failure case. The images are taken from MSRC and the features are SIFT.

SIFT for discriminative clustering here because of the high appearance variability of MSRC.

This experiment calls for some additional comments: First, it is interesting to note that our method works best for faces, despite the high background variability compared to sheep or cow for example. Second, for classes with very high variability (e.g., cat, dog, or chair), the three cosegmentation algorithms perform rather poorly, as expected. Third, it appears that the low performance on the bike class is caused by too-coarse

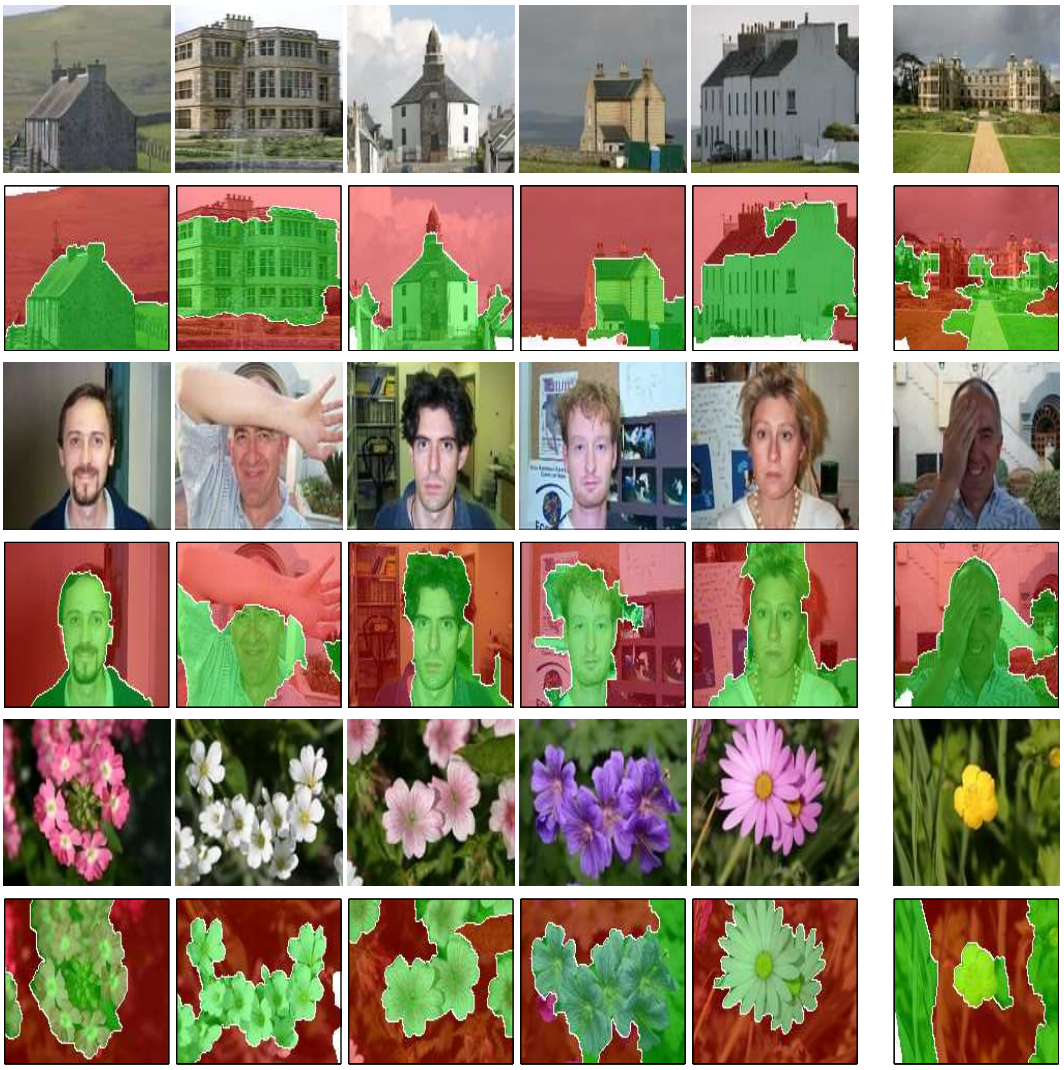


Figure 4.3: Results on binary classification. There are three set of images by row. The images are taken from MSRC and the features are SIFT.

superpixels. Finally, the poor performance of our algorithm on the plane category is mostly due to the fact that the background is (essentially) always the same, and is composed of two kinds of “stuff", i.e., grass and sky, as shown in Figure 4.4. Therefore, with only two classes, our algorithm simply separates sky+plane from grass, which motivates the need for true multi-class cosegmentation as demonstrated in the next section.

Figure 4.4: This figure shows how increasing the number of classes leads to a better segmentation. Columns 2 to 3 respectively show results for $K = 2$ and $K = 3$ (best seen in color).

dataset	images	class	P	Ours	Kim et al. (2011)	Chapter 2
iCoseg	25	Baseball player	5	**62.2**	51.1	24.9
	5	Brown bear	3	**75.6**	40.4	28.8
	15	Elephant	4	**65.5**	43.5	23.8
	11	Ferrari	4	**65.2**	60.5	48.8
	33	Football player	5	**51.1**	38.3	20.8
	7	Kite Panda	2	57.8	**66.2**	58.0
	17	Monk	2	**77.6**	71.3	76.9
	11	Panda	3	**55.9**	39.4	43.5
	11	Skating	2	**64.0**	51.1	47.2
	18	Stonehedge	3	**86.3**	64.6	62.3
MSRC	30	Plane	3	**45.8**	25.2	25.1
	30	Face	3	**70.5**	33.2	66.2
		Average		**64.8**	48.7	43.9

Table 4.2: Results on iCoseg and MSRC using more than two segments. Here, P indicates the number of segments used for our algorithm.

4.4.3 Multi-class experiments

We present in this section our experiments with multiple object categories using the recently released iCoseg database, along with two MSRC classes. iCoseg provides a setting closely related to video segmentation in the sense that, for a given class, the images are similar to key frames in a video, with similar lighting and background. As in

dataset	images	class	P	Ours	Modified Joulin et al. (2010b)
iCoseg	25	Baseball player	5	**62.2**	53.5
	5	Brown bear	3	75.6	**78.5**
	15	Elephant	4	**65.5**	51.2
	11	Ferrari	4	**65.2**	63.2
	33	Football player	5	**51.1**	38.8
	7	Kite Panda	2	57.8	**58.0**
	17	Monk	2	**77.6**	76.9
	11	Panda	3	**55.9**	49.1
	11	Skating	2	**64.0**	47.2
	18	Stonehedge	3	**86.3**	85.4
MSRC	30	Plane	3	**45.8**	39.2
	30	Face	3	**70.5**	56.4
		Average		**64.8**	58.1

Table 4.3: Results compared to the modified version of our previous algorithm on iCoseg and MSRC using more than two segments. Here, P indicates the number of segments used for our algorithm.

the case of the plane in Figure 4.4 (first two columns), this makes binary segmentation very difficult (sometimes meaningless) since multiple object classes may be merged into a single one. As shown by Figure 4.4 (right), adding more classes helps.

The number of meaningful "objects" present in the images varies from one problem to the next, and K must be set by hand. In practice, we have tried values between 2 and 5, and Figure 4.5 shows that this gives reasonably good results in general. Quantitative results are given in Table 4.2 and 4.3. Since, as argued earlier, MSRC and iCoseg are not well adapted to benchmarking true multi-class cosegmentation, we report the maximum of the intersection-over-union scores obtained for the P classes against the "object" region in the ground-truth data.

As before, we use SIFT features for the two MSRC classes used in this experiment. Due to little change in illumination, we use instead color histograms for iCoseg, which are in general more appropriate than SIFT ones in this setting.[6] We compare our algorithm with both our multiclass implementation of the method present in Chapter 2 and the original implementation (with $P = 2$) using the same features as ours. We also compare our method to Kim et al. (2011) with P between 2 and 5, and keep the P value with the best performance.

[6]SIFT features lead to better performance in some of the cases (for example, the performance rises to 85.2% for the brown bear class and to 75.9% for pandas), but for a fair comparison we keep the same features for the entire dataset.

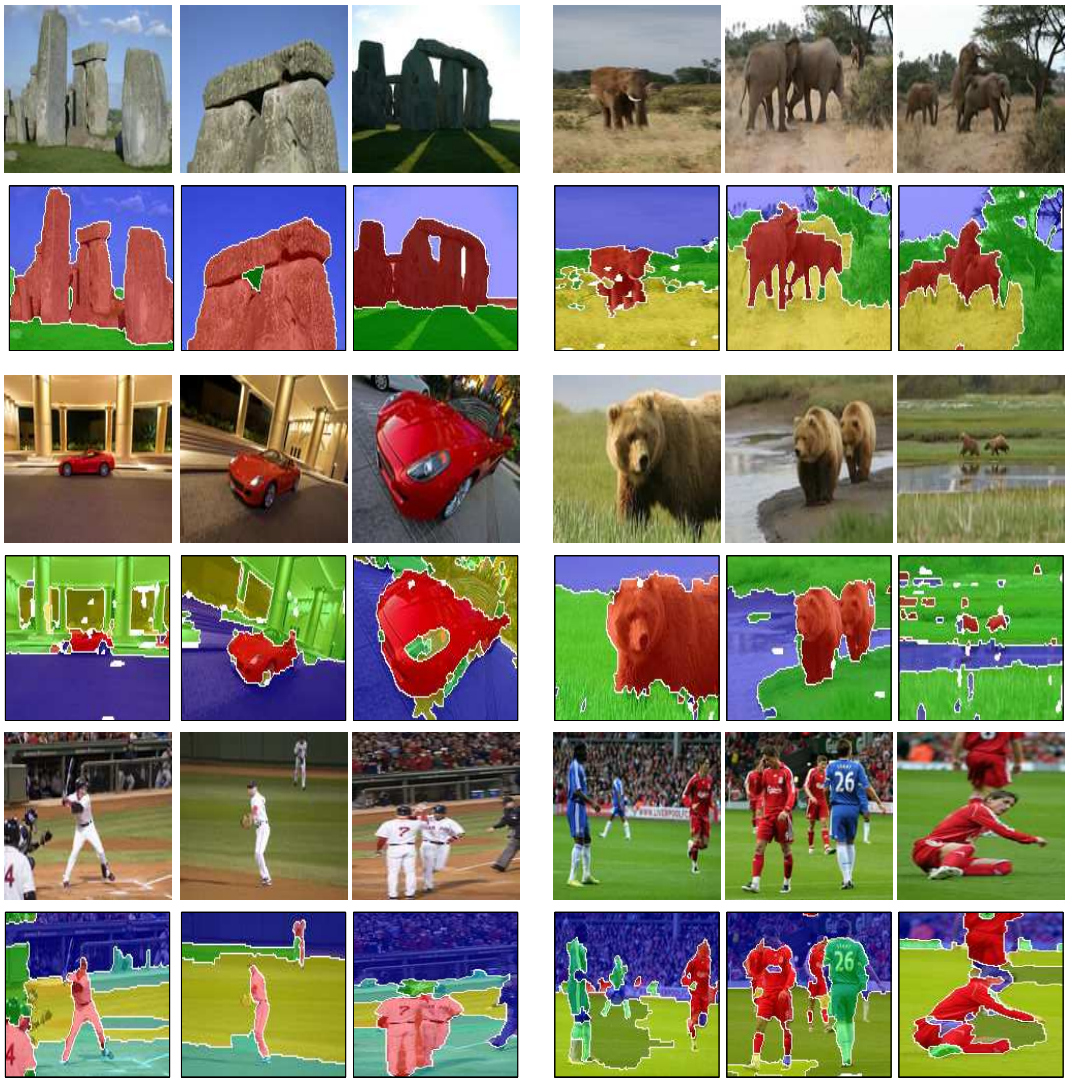


Figure 4.5: Results for the cosegmentation with multiple classes. There are two experiments by row with respectively. The images are taken from iCoseg and the features are color histograms.

We obtain the best performance for 10 of the 12 classes, including the MSRC plane category for which our two-class algortihm only obtained 21.6% in our previous experiment. Note that we do not claim that using multiple classes solves the binary cosegmentation problem. Indeed, we do not know which one of the P classes corresponds to the "foreground" object. On the other hand, our experiments suggest that this object is indeed rather well represented by one of the classes in most of our test cases, which

may be sufficient to act as a filter in an object discovery setting for example (Russell et al., 2006).

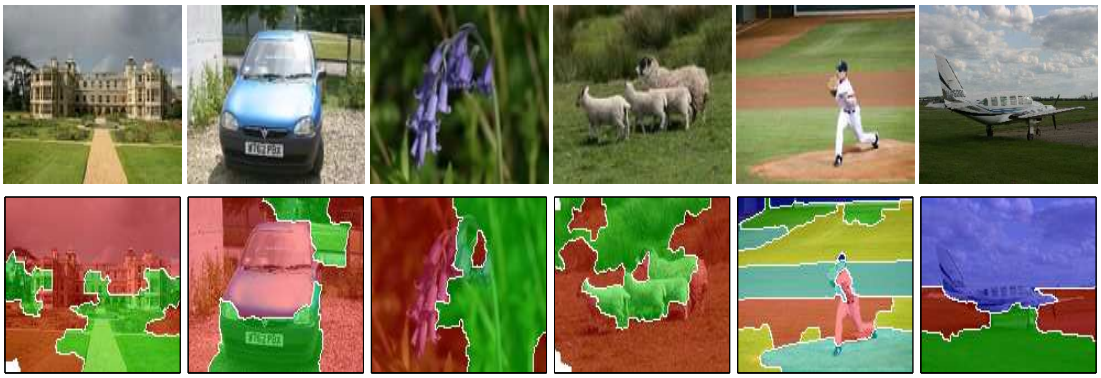


Figure 4.6: Some failure cases.

Of course, our method, like any other, makes mistakes, sometimes giving completely wrong segmentations. Figure 4.6 shows a few examples.

4.4.4 Extensions

Let us close this section with a few proof-of-concept experiments involving simple extensions of our framework.

Figure 4.7: Weakly supervised segmentation results with known tags and SIFT features.

Weakly supervised segmentation. We start with the case where each image is tagged with the object classes it contains. As explained in Section 4.2, this can be handled by a simple modification of our entropy-based regularizer. Figure 4.7 shows qualitative results obtained using 60 sheep and plane images in the MSRC database, labelled with tags from the set {sheep, plane, grass, sky}. The performance is essentially the same as when the two sets of images are segmented separately, but the grass is now identified uniquely in the 60 images.

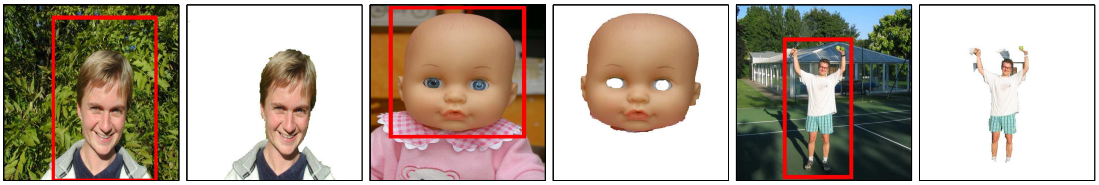


Figure 4.8: Interactive segmentation results with color histogram features.

Interactive segmentation. The weakly supervised version of our method is itself easily generalized to an interactive setting, as in GrabCut (Blake et al., 2004), where the user defines a bounding box around the object of interest. For us, this simply amounts to picking a foreground or background label for each pixel inside the box, and a background label for all the pixels outside. Figure 4.8 shows a few qualitative examples obtained using this method. Again, these are just for proof of concept, and we do not claim to match the state of the art obtained by specialized methods developed since the introduction of (Blake et al., 2004).

Video segmentation. Our experiments with iCoseg suggest that our method is particularly well suited to keyframes from the same video shot, since these are likely to feature the same objects under similar illumination. This is confirmed with our experiments with two short video clips taken from the Hollywood-2 and Grundmann datasets (Grundmann et al., 2010; Marszalek et al., 2009). We pick five key frames from each video and cosegment them using color features without any temporal information such as frame ordering or optical flow. As shown by Figure 4.9, reasonable segmentations are obtained. In particular, the main characters in each video are identified as separate segments.

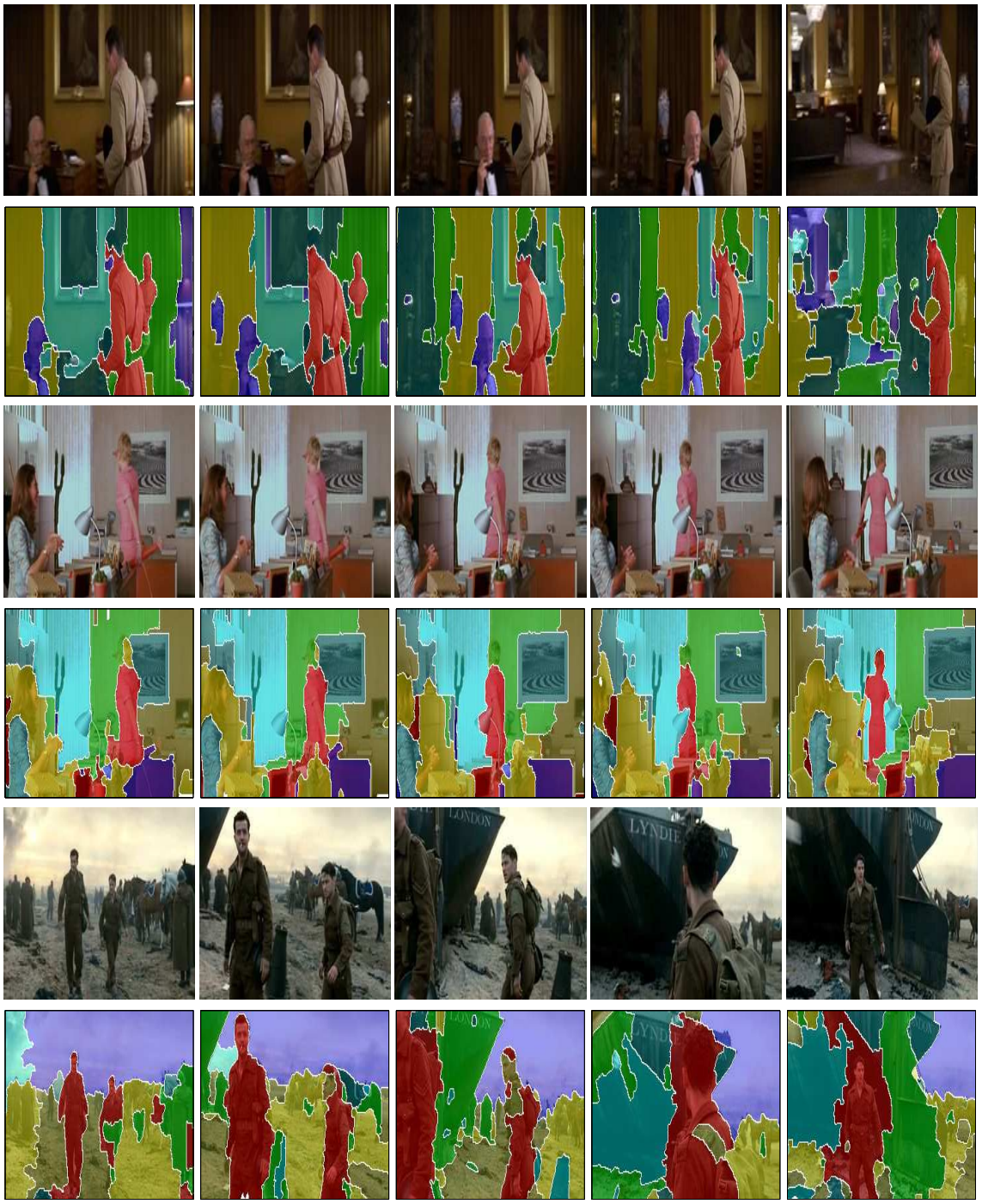


Figure 4.9: Results on two videos. The first row represent the input images, the second one is the segmentation obtained with our algorithm.

5

A convex relaxation for weakly supervised classifiers

Abstract of this chapter: This chapter introduces a general multi-class approach to weakly supervised classification. Inferring the labels and learning the parameters of the model is usually done jointly through a block-coordinate descent algorithm such as expectation-maximization (EM), which may lead to local minima. To avoid this problem, we propose a cost function based on a convex relaxation of the soft-max loss. We then propose an algorithm specifically designed to efficiently solve the corresponding semidefinite program (SDP). Empirically, our method compares favorably to standard ones on different datasets for multiple instance learning and semi-supervised learning, as well as on clustering tasks.

The material of this chapter is based on the following work:

A. Joulin and F. Bach. A convex relaxation for weakly supervised classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

5.1 Introduction

Discriminative supervised classifiers have proved to be very accurate data-driven tools for learning the relationship between input variables and certain labels. Usually, for these methods to work, the labeling of the training data needs to be complete and precise. However, in many practical situations, this requirement is impossible to meet because of the challenges posed by the acquisition of detailed data annotations. This typically leads to partial or ambiguous labelings.

Different weakly supervised methods have been proposed to tackle this issue. In the semi-supervised framework (Chapelle et al., 2006), only a small number of points are

labeled, and the goal is to use the unlabeled points to improve the performance of the classifier. In the multiple-instance learning (MIL) framework introduced by Dietterich and Lathrop (1997), *bags* of instances are labeled together instead of individually, and some instances belonging to the same bag may have different true labels. Finally, in the ambiguous labeling setting (Jin and Ghahramani, 2003; Hullermeier and Beringer, 2006), each point is associated with multiple potential labels.

More generally, in all these frameworks, the points are associated with observable partial labels and the implicit or explicit goal is to *jointly* estimate their true latent labels and learn a classifier based on these labels. This usually leads to a non-convex cost function which is often optimized with a greedy method or a coordinate descent algorithm such as the expectation-maximization (EM) procedure. These methods usually converge to a local minimum, and their initialization remains an open practical problem.

In this chapter, we propose a simple and general framework which can be used for any of the aforementioned problems. We explicitly learn the true latent label and the classifier parameters. We also propose a convex relaxation of our cost function and an efficient algorithm to minimize it. More precisely, we use a discriminative classifier with a soft-max loss, and our convex relaxation extends the work of Guo and Schuurmans (2008).

We develop our framework for the general weakly supervised case. We propose results on both toy examples as proof of concept of our claims, and on standard MIL and semi-supervised learning (SSL) datasets.

5.1.1 Related work

Multiple instance learning. This framework has received much attention because of its wide range of applications. First used for drug activity prediction, it has also been used in the vision community for different problems such as scene classification (Maron and Ratan, 1998), object detection (Viola et al., 2006), object tracking in video (Babenko et al., 2009), and image database retrieval (Yang, 2000). Many MIL methods have been developed in the past decade. Some are based on boosting (Auer and Ortner, 2004), others on nearest neighbors (Wang and Zucker, 2000), on neural networks (Zhang and Zhou, 2006), on decision trees (Blockeel et al., 2005), or the construction of an appropriate kernel (Wang et al., 2008; Gärtner et al., 2002; Kwok and Cheung, 2007). Much of the work in the MIL community has focused on the use of discriminative classifiers, the most popular one being the support vector machine (SVM) (Andrews et al., 2003; Chen and Wang, 2004; Gehler and Chapelle, 2007). In this chapter, we concentrate on the logisitic loss which makes little difference with the hinge loss with the additional advantage of being twice differentiable. Note that this loss has already been used in the context of multiple instance learning (Xu and Frank, 2004; Ray and Craven, 2005), but with different optimization schemes.

Semi-supervised learning. Many semi-supervised learning methods have also been proposed in the past decade (Chapelle et al., 2006; Zhu, 2006). For example, some are based on margin maximization with an SVM framework (Joachims, 1999; Bennett and Demiriz, 1998; Xu and Schuurmans, 2005), and others use the unlabeled data for regularization (Belkin et al., 2004) or co-training of weak classifiers (Blum and Mitchell, 1998).

The link between SSL and MIL has been widely studied in the community. For example, in the context of image segmentation with text annotation, Barnard et al. (2003) propose a general weakly supervised model based on a multi-modal extension to a mixture of latent Dirichlet allocation. An important issue with this family of generative models is that learning the parameters is often untractable. Another example is Zhou and Xu (2007) who use the relation between MIL and SSL to develop a method for MIL.

Unsupervised learning. Many methods have been proposed for unsupervised learning, but, in this chapter, we focus on method based on discriminative clustering (Xu et al., 2005; De la Torre and Kanade, 2006; Bach and Harchaoui, 2007; Joulin et al., 2010a). Discriminative clustering provides a principled way to reuse existing supervised learning machinery while *explicitly* estimating the latent labels. For example, following the SVM approach of Xu et al. (2005), algorithms using linear discriminant analysis (De la Torre and Kanade, 2006) or ridge regression (Bach and Harchaoui, 2007) have been proposed.

The idea of using a convex cost function in the weakly supervision context has been already studied in different contexts such as, for example, ambiguous labeling (Cour et al., 2009) or discriminative clustering (Xu et al., 2005; Bach and Harchaoui, 2007). In this chapter, we are interested in the convex relaxation of a general multiclass loss function, i.e., the soft-max loss. Guo and Schuurmans (2008) propose a related relaxation but do not consider the intercept in the linear classifier. We extend their work to the case of linear classifiers with an intercept and show in the experiment section, why this difference is crucial when it comes to classification. Note that by using kernels, we can use non-linear classifiers as well. Also, our dedicated optimization scheme is more scalable than the one developed in Guo and Schuurmans (2008) and could be applied to their problem as well.

5.2 Proposed model

5.2.1 Notations

We suppose that we observe I bags of instances. In each bag i, an instance n in $\mathcal{N}_i$ is associated with a feature $x_n \in \mathcal{X}$ and a label z_n in $\mathcal{L}$, in certain feature and label space.

In this chapter, we suppose that this label is common to all the instances of a same bag and explain only partially the instances contained in the bag. We are thus interested in finding a *latent* label $y_n \in \mathcal{P}$ which would give a better understanding of the data.

In this chapter, we assume that the latent label y_n of an instance n can only take its values in a subset $\mathcal{P}_{z_n}$ of $\mathcal{P}$ which depends on the label z_n of the bag.

Instance reweighting. In many problems, a set of instances can be bigger than the other, this is the case for example in a *one-vs-all* classifer where the number of positive instances is often very small compared to the number of negative examples. A side-contribution of this work is to consider explicitly a reweighting of the data to avoid undesired side effects: Each point is associated with a weight $\pi_n \geq 0$ which denotes its importance compared to others. Some examples are the uniform case, i.e., $\pi_n = \frac{1}{N}$ or when bags have to be reweighted, i.e., $\pi_n = \frac{1}{IN_i}$ for n in the bag i. We denote by π the vector with entries equal to π_n. Note that $\pi \geq 0$ and $\sum_n \pi_n = 1$.

This setting is very general, so let us now show how it applies to several concrete settings.

Semi-supervised learning. Given a set of true labels $\mathcal{P}$ and N_l points with known label, there are $N_l + 1$ bags, i.e., one for each labeled point and one for all the unlabeled instances. The set $\mathcal{L}$ is equal to $\mathcal{P}$ plus a label for the unlabeled bag (i.e., $L = P + 1$). The true label of an instance in a positive bag is fixed whereas in the unlabeled bag it can take any value in $\mathcal{P}$.

Unsupervised learning. This is an extreme case of the semi-supervised framework with only the unlabeled bag.

Multiple instance learning. There are two possible labels for a bag ($L = 2$), i.e., *positive* ($z_n = 1$) or *negative* ($z_n = 0$). The true label y_n of an instance n in a negative bag is necessarily negative ($y_n = 0$) and in a positive bag it can be either positive or negative ($\mathcal{P}_1 = \{0, 1\}$).

Ambiguous labelling. Each bag is associated with a set of possible true labels $\mathcal{P}_l$. The set of partial labels is thus the combination of all possible subsets of $\mathcal{P}$, i.e., each label $l \in \mathcal{L}$ represents a subset of $\mathcal{P}$ ($L = 2^P$).

5.2.2 Problem formulation

The goal of a discriminative weakly supervised classifier is to find the latent labels y that minimize the value of a regularized discriminative loss function. More precisely, given some latent label y and some feature map $\Phi : \mathcal{X} \mapsto \mathbb{R}^d$ (note that Φ could be explicitly

defined or implicitly given through a positive-definite kernel), we train a multi-class discriminative classifier to find the parameters $w \in \mathbb{R}^{P \times d}$ and $b \in \mathbb{R}^P$ that minimize:

$$L(y, w, b) = \sum_{n=1}^{N} \pi_n \ell(y_n, w^T \phi(x_n) + b),$$

where $\ell : \mathbb{R}^P \times \mathbb{R}^P \mapsto \mathbb{R}$ is a loss function. In this paper, we are interested in the multi-class setting where a natural choice for ℓ is the soft-max loss function Hastie et al. (2001). Note that for a given instance n, the set of possible true labels depends on the the label z of its bag, our loss function $\ell(y_n, w^T \phi(x_n) + b)$ then takes the following form:

$$-\sum_{l \in \mathcal{L}} z_{nl} \sum_{p \in \mathcal{P}_l} y_{np} \log \left(\frac{\exp(w_p^T \phi(x_n) + b_p)}{\sum_{k \in \mathcal{P}_l} \exp(w_k^T \phi(x_n) + b_k)} \right),$$

where w_p^T is the p–th row of w^T and b_p the p–th entry of b.

Cluster-size balancing term. In many unsupervised or weakly supervised problems, a common issue is that assigning the same label to all the instances leads to perfect separation. In the MIL community, this is equivalent to considering all the bags as negative and a common solution is to add a non-convex constraint which enforces at least one point per positive bag to be positive. Another solution used in the discriminative clustering community is to add constraints on the number of elements per class and per bag Xu et al. (2005); Bach and Harchaoui (2007). Despite good results, this solution introduces extra parameters and may be hard to extend to other frameworks such as MIL, where a positive bag may not have any negative instances. Another common technique is to encourage the proportion of points per class and per bag to be close to uniform. An appropriate penalty term for achieving this is the entropy (i.e., $h(v) = -\sum_k v_k \log(v_k)$) of the proportions of points per bag and per latent label, leading to:

$$H(y) = \sum_p \sum_{i \in I} h\left(\sum_{n \in \mathcal{N}_i} \pi_n y_{np} \right).$$

Penalizing by this entropy turns out to be equivalent to maximizing the log-likelihood of a graphical model where the features x_n explain the labels z_n through the latent labels y_n as in Chapter 3. An important consequence is that the natural weight of this penalty in the cost function is 1, so we do not add any extra parameters.

To avoid over-fitting, we penalize the norm of w, leading to the following cost function:

$$f(y, w, b) = L(y, w, b) - H(y) + \frac{\lambda}{2P} \|w\|_F^2,$$

where $\lambda > 0$ is the regularization parameter and the problem thus takes the following form:

$$\min_{\forall n \leq N,\ y_n \in \mathcal{S}_{P_{z_n}}} \ \min_{w \in \mathbb{R}^{d \times P},\ b \in \mathbb{R}^P} f(y, w, b), \tag{5.1}$$

where $\mathcal{S}_P = \{t \in \mathbb{R}^P \mid t \geq 0,\ t^T 1_P = 1\}$ is the *simplex* in $\mathbb{R}^P$. To avoid cumbersome double subscripts, we suppose that any instance n in a bag with a label z_n (which is common to the entire bag), has a latent label y_n in $\mathcal{P}$ instead of $\mathcal{P}_{z_n}$.

In the next section we show how to obtain a convex relaxation of this problem.

5.3 Convex relaxation

An interesting feature of the soft-max cost function is its link to the entropy through the Fenchel conjugate (Boyd and Vandenberghe, 2003), i.e., given a P-dimensional vector t, the log-partition can be written as

$$\log\Big(\sum_{p=1}^{P} \exp(t_p)\Big) = \max_{v \in \mathcal{S}_P} \sum_{p=1}^{P} v_p t_p + h(v).$$

This is the cornerstone of expectation-maximization procedures and more generally, of variational methods (Yedidia et al., 2003; Wainwright and Jordan, 2008). Substituting in the loss function, the weakly supervised problem defined in Eq. (5.1) can be reformulated as:

$$\min_{y \in \mathcal{S}_P^N} \max_{q \in \mathcal{S}_P^N} \sum_{i \in I} \sum_{n \in \mathcal{N}_i} \pi_n h(q_n) - H(y) + g(y, q), \tag{5.2}$$

where q is an $N \times P$ matrix with n-th row q_n^T, and $g(y, q)$ is equal to:

$$\min_{\substack{w \in \mathbb{R}^{P \times d} \\ b \in \mathbb{R}^P}} \sum_{i \in I} \sum_{n \in \mathcal{N}_i} \pi_n (q_n - y_n)^T (w^T \phi(x_n) + b) + \frac{\lambda}{2P} \|w\|_F^2.$$

Minimizing this function w.r.t. the intercept b leads to an *intercept constraint* on the dual variables, i.e, $(q - y)^T \pi = 0$. The minimization w.r.t. w leads to a closed-form expression for g:

$$g(y, q) \ = \ -\frac{P}{2\lambda} \mathrm{tr}\big((q - y)(q - y)^T K\big),$$

where K is the positive definite kernel matrix associated with the reweighted mapping ϕ, i.e., with entries equal to $K_{nm} = \pi_n \phi(x_n)^T \phi(x_m) \pi_m$. The cost function is

not convex in general in z since it is the maximum over a set indexed by q of concave functions in y. A common way of dealing with this issue is to relax the problem into a semidefinite program (SDP) in yy^T. Unfortunately, our cost function does not directly depend on yy^T, but a reparametrization in terms of q inspired by (Guo and Schuurmans, 2008) allows us to get around this technical difficulty.

Reparametrization in q. We reparametrize the problem by introducing an $N \times N$ matrix Ω such that $q = \Omega y$ (Guo and Schuurmans, 2008). The *intercept constraint* and the *normalization constraint* on q (i.e., $q1_K = 1_N$) become constraints over Ω, i.e., respectively $\Omega^T\pi = \pi$ and $\Omega 1_N = 1_N$. Translating the addition of an intercept to a linear classifier into a simple constraint on the columns of Ω provides a significant improvement over Guo and Schuurmans (2008), as shown in Section 5.5.1. This reparametrization has the side-effect of introducing a non-convex term in the cost function since the entropies over q_n in Eq. (5.2) is replaced by an entropy over the n–th row of Ωz which is not jointly concave/convex in Ω and y.

Tight upper-bound on the entropy. We consider the following equality:

$$\begin{aligned}
&\max_{\substack{\Omega,\\ \Omega y = q}} \sum_n \pi_n \sum_{p=1}^{P} \sum_{m \in \mathcal{B}_k} \Omega_{nm} \log\left(\frac{\Omega_{nm}}{\pi_m}\right) \\
&= \sum_{p=1}^{P} \sum_n \pi_n q_{nk} \log(q_{nk}) - \sum_{p=1}^{P} \sum_n \pi_n q_{nk} \log\left(\sum_{m \in \mathcal{B}_k} \pi_m\right)
\end{aligned}$$

which is attained for $\Omega_{nm} = \sum_{p=1}^{P} \delta(m \in \mathcal{B}_k) \frac{\pi_m}{\sum_{p \in \mathcal{B}_k} \pi_p} q_{nk}$. From this equality, we bound the entropy in q by a difference of entropy in Ω and y, up to an additive constant C_0:

$$\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_i} \pi_n h(q_n) \leq -\sum_n \pi_n h(\Omega_n) + H(z) + C_0. \tag{5.3}$$

This upper-bound is tight in the sense that given a discrete value of y (i.e., before the relaxation), the maximum of the left part among discrete values of q is equal to the maximum of the right part among corresponding discrete values of Ω. Note also that the term in y appearing in Eq. (5.3) cancels out with the entropy term in Eq. (5.2). This relaxation leads to the minimization of the following function of y:

$$\max_{\Omega \in \mathcal{O}} -\frac{P}{2\lambda} \mathrm{tr}\big(yy^T (I - \Omega)^T K (I - \Omega)\big) - \sum_n \pi_n h(\Omega_n),$$

where $\mathcal{O} = \{\Omega \mid \Omega 1_N = 1_N,\ \Omega^T\pi = \pi,\ \Omega \geq 0\}$. This problem depends on y solely through the matrix yy^T, and can thus be relaxed into an SDP in yy^T.

Reparametrization in y. With the change of variable $Y = yy^T$, we have the maximum of a set of *linear* functions of Y, which is convex. However, the set $\mathcal{Y}$ of possible values for Y is non-convex since it is defined by:

$$\begin{cases} \text{diag}(Y) = 1_N,\ Y \geq 0,\ Y \succeq 0, \\ \text{rank}(Y) = k - 1. \end{cases} \tag{5.4}$$

Let us review these constraints:

- In practice, the piecewise-positivity constraint is not necessary and removing it leads to a matrix Y with entries in $[-1, 1]$ since Y is positive semi-definite with ones on the diagonal.
- The rank constraint is the main source of non-convexity, and will be removed, thus leading to a convex relaxation.
- The rest of the constraints defines the *elliptope*:

$$\mathcal{E}_N \ = \ \{Y \in \mathbb{R}^{N \times N} \mid \text{diag}(Y) = 1_N,\ Y \succeq 0\}.$$

Note that an additional linear constraint may be needed depending on the considered weakly supervised problem. We give below some examples:

- In the case of MIL, this constraint takes the form of $Y_- = 1_{N_-} 1_{N_-}^T$, where N_- is the number of negative examples, and Y_- is the restriction of Y to the negative bags.
- "Must-not-link" constraints on the instances can be handled: If two bags i and j have labels z_i and z_j such that the set of possible latent labels are dissimilar (i.e., $\mathcal{P}_{l_i} \cap \mathcal{P}_{l_j} = \emptyset$), we can constrain the submatrix Y_{ij} to be equal to 0. These constraints are of particular interest in the case of SSL, where labeled bags with different labels should not be assigned to the same latent label.

In the rest of this paper, we consider the specific cases of SSL, MIL and discriminative clustering:

- In SSL, we can reduce the dimensionality of Y: Since all the values of y with a same known label are equal, it is equivalent to replace them by a single element in Y. Denoting by N_u is the number of unlabeled points, P the number of labels and $N_R = N_u + P$, this is equivalent to considering a matrix $R^T Y R$ instead of Y, where R is a $N \times N_R$ matrix whose restriction to the unlabeled bags is the identity and all other entries are zero except for $R_{n(N_u+l)}$ which is equal to 1 if the instance n has a known label l.

- In MIL, the same reduction can be done with $P = 1$ and N_u denoting the total number of positive instances.
- Discriminative clustering is similar to SSL with $P = 0$.

By taking into account all of these modifications and by dropping the rank constraint, we replace the non-convex set $\mathcal{Y}$ by the elliptope $\mathcal{E}_{N_R}$, leading to the minimization of $g(Y)$ over $\mathcal{E}_{N_R}$, where $g(Y)$ is equal to:

$$\max_{\Omega \in \mathcal{O}} -\frac{P}{2\lambda}\text{tr}\big(YR(I-\Omega)^T K(I-\Omega)R^T\big) - \sum_n \pi_n h(\Omega_n). \tag{5.5}$$

In the next section we propose an efficient algorithm to solve this convex optimization problem.

5.4 Optimization

Since our optimization involves a maximization in our *inner loop*, it cannot be solved directly by a general-purpose toolbox. We propose an algorithm dedicated to our case. In the rest of this chapter we refer to the maximization as the inner loop and the overall minimization of our cost function as the *outer loop*.

5.4.1 Inner loop

Evaluating the cost function defined in Eq. (5.5) involves the maximization of the sum of the entropy of Ω and a function T defined as:

$$T(\Omega) \;=\; -\frac{1}{2\lambda}\text{tr}\big((I-\Omega)R^T Y R(I-\Omega)^T K\big).$$

We use a proximal method with a reweighted Kullback-Leibler (KL) divergence which naturally enforces the point-wise positivity contraint in $\mathcal{W}$, and leads to an efficient Bregman projection with a KL divergence (an *I-projection* to be more precise) on the rest of the constraints defining $\mathcal{W}$. The choice of a KL divergence is natural in our case since Ω is point-wise positive with rows suming to 1. More precisely, given a point Ω^0, the proximal update is given by maximizing the following function:

$$l_D(\Omega) = \text{tr}\big(\Omega^T \nabla T(\Omega^0)\big) - \sum_n \pi_n h(\Omega_n) - L D_\pi(\Omega \,\|\, \Omega^0), \tag{5.6}$$

where L is the Lipschitz constant of ∇T and D_π is a reweighted KL divergence defined as:

$$D_\pi(\Omega \parallel \Omega^0) = \sum_i \sum_{n \in \mathcal{N}_i} \pi_n \sum_{m=1}^{N} \Omega_{nm} \log\left(\frac{\Omega_{nm}}{\Omega^0_{nm}}\right).$$

The *I-projection* can be done efficiently with an *iterative proportional fitting procedure* (IPFP), which is guaranteed to converge to the global minimum with linear convergence rate Fienberg (1970). The classical IPFP has a complexity of $O(N^3)$ but can be reduce to $O(N^2)$ by using a *factor estimation*.

More precisely, at each iteration t, we update α, β and Ω in the following way:

$$\begin{aligned} \alpha_n^t &\leftarrow \frac{\pi_n}{\sum_{m=1}^N \Omega_{nm}^{t-1}\pi_m\beta_m^{t-1}} \\ \beta_n^t &\leftarrow \frac{\pi_n}{\sum_{n=1}^N \Omega_{nm}^{t-1}\alpha_n^t} \\ \Omega_{nm}^t &\leftarrow \alpha_n^t\Omega_{nm}^{t-1}\beta_n^t. \end{aligned}$$

Note that we also try the Euclidean projection over $\mathcal{W}$ which requires the use of Dijkstra algorithm. Experimentally, we found that Euclidean projection was significantly slower than I-projection in our case.

Accelerated proximal method. Note that to obtain a faster convergence of the inner loop, we may take advantage of a low-rank decomposition of K and R^TYR and we use an accelerated proximal scheme on the logarithm of Ω (Beck and Teboulle, 2009). To control the distance from the optimum Ω^*, we can use a provably correct duality gap which can be computed efficiently.

Duality gap. To have an accurate estimation of our cost function in Y, our inner loop must produce a candidate Ω which is guaranteed to be close to the global optimum Ω^*. This can be ensured by a good stopping criteria. A good candidate is a duality gap but, since speed is an issue in our case, we have to find a dual variable and its associated function which can be efficiently evaluated at any point. Our candidate for the dual function is:

$$f_D(B) = \frac{1}{2\lambda}\text{tr}(YK) - \frac{1}{2\lambda}\text{tr}(B^TB) + \min_{\Omega\in\mathcal{W}}\left[\frac{1}{\lambda}\text{tr}(\Omega^T(YZ - kBy^T)) - h(\Omega)\right],$$

and, given a primal variable Ω, its associated dual variable is $B = \Phi^T\Omega u$ and the duality gap is thus:

$$D(\Omega) = f_i(\Omega) - \pi^T[\Omega\log(\Omega)]1_N - f_D(\Phi^T\Omega u).$$

5.4.2 Outer loop

The outer loop minimizes $g(Y)$ as defined in Eq. (5.5) over the elliptope $\mathcal{E}_{N_R}$. Many approaches have been proposed to solve this type of problems (Goemans and Williamson, 1995; Burer and Monteiro, 2003; Journée et al., 2010) but, to the best of our knowledge,

they all assume that the function and its gradient can be computed efficiently and put the emphasis on the projection. This is not the case in our problem, and we thus propose a method adapted to our particular setting.

First, to simplify the projection on the $\mathcal{E}_{N_R}$, we replace our cost function $g(Y)$ by its diagonally rescaled version $g_R(Y) = g(\text{diag}(Y)^{-1/2}Y\text{diag}(Y)^{-1/2})$. Note that even if this function is in general non-convex, it coincides with $g(Y)$ on $\mathcal{E}_{N_R}$, making its restriction to this set convex. This modification allows us to rescale the diagonal of any update Y to a diagonal equal to 1_N without modifying the value of our cost function.

Our minimization of g_R over the elliptope is also based on a proximal method with a Bregman divergence to guarantee updates that stay in the feasible set. A natural choice for the Bregman divergence is the KL divergence based on the von Neumann entropy, i.e, the entropy of the eigenvalues of a matrix (see more details in the appendix):

$$D_{vn}(Z \parallel Y_P) \quad = \quad \text{tr}(Z\log(Z)) - \text{tr}(Z\log(Y_P)).$$

This divergence guarantees that each update has non-negative eigenvalues. Given a point Y_0, its update can then be obtained in closed-form as the diagonally rescaled version of $V\text{Diag}(\exp(\text{diag}(\frac{1}{t}E)))V^T$, where V and E are the eigenvectors and the eigenvalues of $-\nabla g_R(Y_0)+t\log(Y_0)$ and t is a positive step size computed using a line-search with backtracking.

Duality gap. As in the inner loop, we use a computationally tractable provable duality gap, i.e., $-N_R\lambda_{min}$, where λ_{min} is the lowest eigenvalue of $\nabla g_R(Y)$.

This duality gap is obtained by relaxing the diagonal constraint by a trace one:

$$D(\mu,\Omega) = \min_{\substack{U\succeq 0,\\ \text{tr}(U)=N.}} -\frac{1}{2\lambda}\text{tr}\big(UP^T(I-\Omega)^TK(I-\Omega)P\big) + \pi^T[\Omega\log(\Omega)]1_N + \mu^T(1_n - U)).$$

Its associated dual variable is $\mu = \text{diag}(Y\nabla g(Y))$. Also it is easy to see that denoting by λ_{min} the lowest eigenvalue of $\nabla g(Y) - \text{diag}(\text{diag}(Y\nabla g(Y)))$, we have:

$$D(\mu,\Omega) = \mu^T P 1_N + N\lambda_{min} + \pi^T[\Omega\log(\Omega)]1_N.$$

After some calculation, we show that $D(\mu,\Omega) = g_R(Y) + N\lambda_{min}$. The duality gap is thus simply:

$$DG(Y,\Omega) = -N\lambda_{min}.$$

5.4.3 Rounding

Many rounding schemes can be applied with similar performances. Following (Bach and Harchaoui, 2007) and Chapter 3, we use k-means clustering on the eigenvectors

step complexity	Inner loop update $O(N^2)$	Inner loop duality gap $O(N^2)$
step complexity	Outer loop proximal $O(N^3)$	Outer loop duality gap $O(N)$

Figure 5.1: Complexity of the different steps in our algorithm.

associated with the k highest eigenvalues (Ng et al., 2001) to obtain a rounded solution y^*. This y^* is then used to initialize an EM procedure to solve the problem defined in Eq. (5.1) and obtain the parameters (w, b) of the classifier, leading to finer details not caught by the convex relaxation. In the general case, the E-step, i.e., the optimization over y is done by a projected gradient descent and the M-step, i.e., the optimization over (w, b), we use L-BFGS. A specificity of the MIL framework is that strictly no point from a negative bag should be classified as positive, which leads to adding to Eq. (5.1), the following linear constraints on the parameters of the classifier:

$$\forall i \in I_-,\ n \in \mathcal{N}_i,\ w_0^T \phi(x_n) + b_0 \geq w_1^T \phi(x_n) + b_1. \tag{5.7}$$

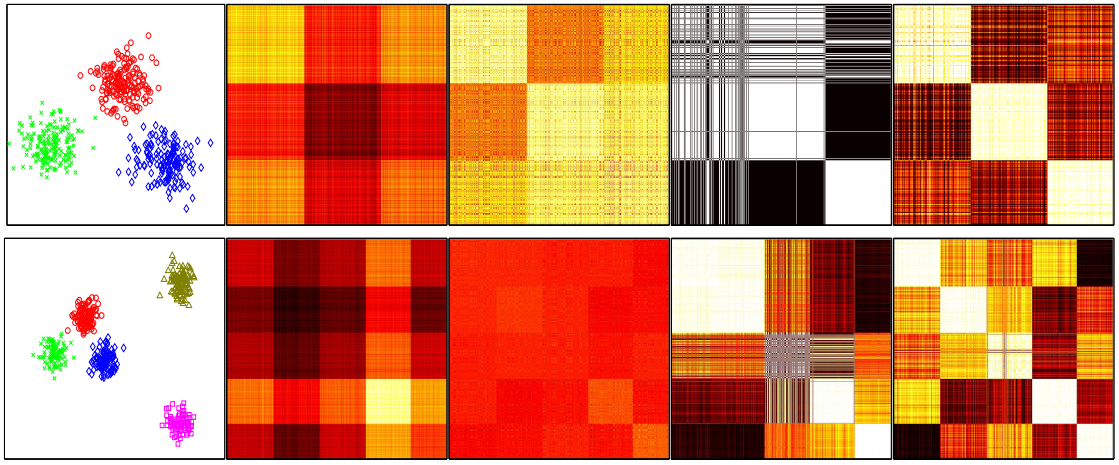
We add these hard constraints in the M-step (optimization over w and b) of the EM procedure. The projection over this set of linear constraints is performed efficiently with an homotopy algorithm in the dual (Mairal et al., 2010).

5.5 Results

Implementation. Our algorithm is implemented in MATLAB and takes from 1 to 5 minutes for 500 points. Note that we can efficiently compute the solutions for different values of λ using warm restarts. Our overall complexity is $O(N^3)$ but we can scale up to several thousands of points. The complexity of the different steps in our algorithm is given in Figure 5.1. On larger datasets, we can use our relaxation on subsets of instances or on pre-clustering the instances (with k-means) and use it to initialize the EM on the complete dataset.

5.5.1 Discriminative clustering

In this section, we compare our method to two different discriminative clustering methods for the multiclass case: the SDP relaxation of the soft-max problem with no intercept (Guo and Schuurmans, 2008) and the discriminative clustering framework introduced by Bach and Harchaoui (2007). The latter comparison is relevant since they propose a convex cost function based on the square loss with intercept.

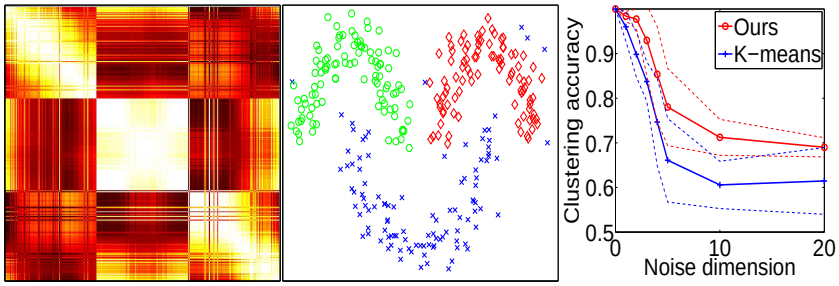


(a) (b) (c) (d) (e)

Figure 5.2: (a) The clustering problem, (b) the given kernel matrix $K = xx^T$, (c) the matrix Z obtained with Bach and Harchaoui (2007), (d) the matrix Z obtained with no intercept and (e) our method (best seen in color).

We consider in Figure 5.2, as a proof of concept, two toy examples where the goal is to find 3 and 5 clusters with *linear* kernels and $N = 500$. Even if the clusters are linearly separable, the set of values of w and b which leads to a perfect separation is very small (Figure 5.2, panel (a)), making the problem challenging. With a RBF kernel, these clustering problems are too simple and thus the influence of the intercept on the performances would have been impossible to perceive. For fair comparison, we test different regularization parameters and show the one leading to the best performances. We show the matrix Y obtained for the three methods as well as the matrix $K = xx^T$ in Figure 5.2. We see that our method clearly obtains a better estimation of the class assignment compared to the others, showing the importance of both the soft-max loss and the intercept.

In panels (a) and (b) of Figure 5.3, we also show that our method works with non-linear kernels in a multiclass setting. Finally, in the panel (c) of Figure 5.3, we show a comparison with k-means as we increase the number of dimensions containing only noise, following the setup of Bach and Harchaoui (2007). Our setting is the 3-cluster problem shown in Figure 5.2 with an RBF kernel and $N = 300$. We see that our algorithm is more robust than k-means.



(a) (b) (c)

Figure 5.3: (a) The matrix obtained with our method and (b) its corresponding clusters. (c) Comparison with k-means on noise robustness ($P = 3$, $N = 300$).

Algorithm	Musk1	Tiger	Elephant
Citation k-NN (Wang and Zucker, 2000)	**91.3**	78.0	80.5
EM-DD (Zhang and Goldman, 2001)	**84.8**	72.1	78.3
mi-SVM (Andrews et al., 2003)	**87.4**	78.9	82.0
MI-SVM (Andrews et al., 2003)	77.9	**84.0**	81.4
PPMM Kernel (Wang et al., 2008)	**95.6**	80.2	82.4
Random init / Uniform	71.1	69.0	74.5
Random init / Weight	76.6	71.0	74.5
No intercept / Uniform	75.0 ± 19.5	67.8 ± 10.4	77.3 ± 9.2
No intercept / Weight	77.8 ± 15.7	71.0 ± 10.8	78.9 ± 9.8
Ours / Uniform	$\mathbf{84.4 \pm 14.0}$	73.0 ± 8.2	$\mathbf{86.7 \pm 3.5}$
Ours / Weight	$\mathbf{87.7 \pm 13.3}$	78.0 ± 5.4	$\mathbf{83.9 \pm 4.2}$

Figure 5.4: Accuracy of our approach and of standard methods for MIL. We evaluate our method with and without the intercept and with two types of weights. In bold, the significantly best performances.

Algorithm	Fox	Trec1
Citation k-NN (Wang and Zucker, 2000)	**60.0**	87.0
EM-DD (Zhang and Goldman, 2001)	56.1	85.8
mi-SVM (Andrews et al., 2003)	**58.2**	**93.6**
MI-SVM (Andrews et al., 2003)	**59.4**	**93.9**
PPMM Kernel (Wang et al., 2008)	**60.3**	**93.3**
Random init / Uniform	**61.0**	81.3
Random init / Weight	**59.0**	84.4
No intercept / Uniform	51.3 ± 6.4	87.5 ± 5.2
No intercept / Weight	52.1 ± 5.0	87.3 ± 5.6
Ours / Uniform	$\mathbf{57.5 \pm 5.9}$	$\mathbf{93.0 \pm 4.7}$
Ours / Weight	$\mathbf{62.5 \pm 6.4}$	$\mathbf{89.0 \pm 6.2}$

Figure 5.5: Accuracy of our approach and of standard methods for MIL. We evaluate our method with and without the intercept and with two types of weights. In bold, the significantly best performances.

5.5.2 Multiple instance learning

In Figure 5.4 and 5.5, we show some comparisons with other MIL methods on standard datasets (Dietterich and Lathrop, 1997; Andrews et al., 2003) for a variety of tasks: a drug activity prediction (*musk*), image classification (*fox, tiger* and *elephant*), and text classification (*trec1*). In the *musk* dataset, the goal is to predict drug activity. Each bag represents a molecule each instance a spatial configuration for the molecule. In the *fox, tiger* and *elephant* datasets, the goal is image classification where the difficulty arises from the fact that each object is in front of a background. The goal is thus to find region related to the object of interest by using negative bags which are images composed only of background. Finally in the *trec* dataset, the goal is text classification.

For comparison, we use the setting described by Andrews et al. (2003), where we create 10 random splits of the data, train on 90% of them and test on the remaining 10%. We test our algorithm with and without the intercept and with uniform or bag-specific (i.e., $\frac{1}{IN_i}$ for instances in the bag i) weights, and compare it to some classical MIL algorithms. Note that we have only tried a linear kernel, and we select the regularization parameter using a 2-fold cross-validation for each split. Our algorithm obtains comparable performances with methods dedicated to the MIL problem.

	Dataset	Linear	Nonlinear	Entropy-Reg.	**Ours (Linear)**	**Ours (Nonlinear)**
l=10	Digit1	79.41	82.23	75.56	**84.57 ± 0.67**	75.45 ± 2.88
	BCI	49.96	**50.85**	**52.29**	**52.22 ± 1.13**	**50.21 ± 1.09**
	g241c	79.05	75.29	52.64	**87.15 ± 0.21**	**87.29 ± 0.42**
	g241d	**53.65**	**49.92**	**54.19**	**54.44 ± 9.09**	**53.15 ± 10.09**
	USPS	69.34	74.80	**79.75**	57.08 ± 13.34	**79.48 ± 0.50**
l=100	Digit1	81.95	**93.85**	**92.72**	91.24 ± 1.66	**93.31 ± 0.97**
	BCI	57.33	66.75	71.11	**78.12 ± 2.26**	64.04 ± 0.87
	g241c	81.82	81.54	79.03	**86.02 ± 0.72**	85.13 ± 0.71
	g241d	**76.24**	**77.58**	74.64	**77.11 ± 1.65**	73.03 ± 3.02
	USPS	78.88	**90.23**	87.79	71.62 ± 2.62	73.04 ± 0.19

Figure 5.6: Comparison in accuracy on SSL databases with methods proposed in Chapelle et al. (2006). In bold, the significantly best performances.

5.5.3 Semi-supervised learning

For the SSL setting, we choose the standard SSL datasets and we compare with methods proposed in Chapelle et al. (2006). The benchmarks (Linear and Nonlinear) are based on a SVM formulation and the benchmark (Entropy-Reg.) uses an entropy regularization. We use our method with either a linear or a RBF kernel. To fix our parameters, we follow the experimental setup of Chapelle et al. (2006). Each set contains 1500 points and either $l = 10$ or 100 of them are labeled. We show the results in Figure 5.6. As expected, since the benchmarks and our formulation are very related, the performances are mostly similar when $l = 100$. However, when $l = 10$, our method is more robust and its performances get significantly higher showing that a convex relaxation is less sensible to noise and poorly labeled data.

Conclusion

During this PhD thesis, we have investigated the image segmentation problem in the presence of partial observable information. This particular setting may seem at first glance restrictive but in fact, cover a large scale of previously studied frameworks such as cosegmentation, supervised segmentation, interactive segmentation, weakly labelled segmentation or video segmentation. One of the key reasons that push us to follow this direction during this thesis, is the fact that, bottom-up segmentation is an ill-posed problem and "acceptable" segmentation can only be obtained in a task-driven framework such as the ones described before. We thus aim at providing the more general and flexible framework for segmentation.

In Chapter 2, we first use a simple model for binary classification only tailored for the restrictive setting of foreground/background segmentation of multiple images. Our model worked well for the special problem of cosegmentation but was not considered for multiclass problems and other type of segmentation problems.

In Chapter 3, we introduce a general discriminative graphical model to tackle these issues. Our model bears some similarities with existing discriminative frameworks such as neural networks or mixture of experts. We also showed that this model was to some extend related to the simple model used in the previous chapter. We show in this chapter that this model was well suited for a large range of problems, from unsupervised to supervised learning settings.

In Chapter 4, we apply the model developed in Chapter 3 to segmentation problems. For that matter, we made some modifications as to take into account the specificities of image segmentation. Even if we focused more on cosegmentation, we showed that this model could be apply to a larger set of segmentation problems with no modifications.

Finally in the last chapter, we investigate more deeply the relation found in the second chapter between our model and the simple model of the first chapter. The motivation was that the optimization problem related to our model was not convex whereas the model used in the first chapter was related to a convex formulation. We manage to obtain a tight convex relaxation for the problem related to our model and we developed a dedicated optimization scheme for it. An interesting feature of this relaxation is that its core results can be apply to other discriminative model such as neural networks or mix-

ture of experts. We also show that using our convex relaxation on real world problems lead to better performances.

In this thesis, we try to answer a very general question about computer vision and on the way, explored different fields of research. Our work have partially answered our preliminary problem but has open also a series of directions that will be interesting to follow after this thesis:

First, it would be interesting to use the convex relaxation developed in Chapter 5 to segmentation problems. This would allow us to measure on a concrete difficult problem the improvement made by using tighter convex relaxation than the one used in Chapter 4.

Another direction regarding segmentation, is to explore more deeply other form of segmentation. More precisely, we could compare to methods dedicated to supervised and weakly supervised segmentation. Also, we could investigate modifications of our model to take into account video streams or to encourage top-down object segmentation. More precisely, we could investigate how to modify our graphical model to add other sources of information such as optical flow or saliency maps.

Regarding discriminative graphical models, another interesting direction is to apply our convex relaxation to other models and see how it might improve their performances. Of particular interest, neural network models have been initialized with different heuristics to avoid convergence to a simple single activated neuron. The either rely on randomization or on encoders which may not be completely satisfactory solutions. Developing a dedicated tight convex relaxation may lead to better initializations.

Finally, as explained in the introduction, we considered mostly semidefinite relaxations which often lead to use computationally expensive algorithms. In particular, most of the algorithms used have to “look” at all the data simultaneously. It would be interesting to explore algorithm which avoids this limitation and thus could be use in parallel as to speed up the algorithm while keeping guarantees of convergence.

6

Appendix A

6.1 The Schur Complement and the Woodbury matrix identity

6.1.1 The Schur Complement

the Schur complement is quantity used in matrix theory and linear algebra. Given a $nm \times nm$ block matrix M whose decomposition in blocks is:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

where A, B, C and D are $n \times n$, $n \times m$, $m \times n$ and $m \times m$ matrices.

Suppose that D is invertible then the *Schur complement* is:

$$A - BD^{-1}C.$$

Let us give an important property using the Schur complement:

- Given that $B = C^T$, then:

  $$M \succeq 0 \Leftrightarrow A \succeq 0, S = C - B^T A^{-1} B \succeq 0$$

  and equivalently:

  $$M \succeq 0 \Leftrightarrow C \succeq 0, A - BC^{-1}B^T \succeq 0.$$

  The second one is of particular interest for the use of the Schur complement made in the introduction of this thesis.

6.1.2 The Woodbury matrix identity

The Woodbury matrix identity is a direct application of the Schur complement to matrix decomposition. Keeping the same notations as above, the equality is:

$$(A + CDB)^{-1} = A^{-1} - A^{-1}C\left(D^{-1} + BA^{-1}C\right)^{-1} BA^{-1}.$$

This inequality is of particular interest for the first chapter of this thesis.

6.2 Differentiability of the maximum

Given a compact space V, suppose the function $\phi : \mathbb{R}^N \times V \mapsto \mathbb{R}$, is jointly continuous, C^1 in the first coordinate and strictly convex in the second coordinate. We note:

$$f(z) = \max_{v \in V} \phi(z, v)$$

and:

$$U(z) = \{u \mid \phi(z, u) = f(z)\}.$$

The Danskin theorem Given the hyposthesis stated above, for any given z, $U(z)$ is reduce to a singleton $\{u^*(z)\}$ and the function f is differentiable at any z with derivative:

$$\nabla f(z) = \left. \frac{\delta g(x, u^*(z))}{\delta x} \right|_{x=z}.$$

We know show that our function defined in Chapter 5 correspond to the setting described above: First it is clear that the function g can be indifferently written in terms of Z or vec(Z). Then it is clear that the associated function ϕ is jointly continuous and C^1 in the first coordinate (as a linear function of the first coordinate). Also ϕ is convex in the second coordinate and since the entropy over Ω is strictly convex over the set $\mathcal{O}$, ϕ is strictly convex in Ω.

Therefore we can apply Danskin's theorem to our problem and state that g is differentiable in Z.

6.3 Logarithm of matrices

The logarithm of a N square matrix A is the N square matrix U such that:

$$A = e^U,$$

where $e^U = \sum_{k=1}^{\infty} \frac{1}{k!} U^k$. If A is a symmetric matrix, there is a orthonormal matrix P and a diagonal matrix D such that:

$$A = PDP^T,$$

then we have:

$$U = P \log(D) P^T,$$

where $\log(D)$ is a diagonal matrix with entries equal to $\log(d_{nn})$ The eigenvalues and eigenvectors of $\mathbf{Z}$ are respectively equal to $\sum_n z_{nk}$ and the z_k for any k. Thus we have:

$$\log(\mathbf{Z}) = Q\Delta Q^T,$$

where Δ is the K diagonal matrix with entries equal to $\log(\sum_n z_{nk})$ and Q is a orthonormal matrix with entries equal to $\frac{z_{nk}}{(\sum_n z_{nk})^{1/2}}$. With a simple calculation we obtain the von Neumann entropy:

$$\mathbf{Z}\log(\mathbf{Z}) \;=\; \sum_{k=1}^{K}\left(\sum_{n=1}^{N} z_{nk}\right)\log\left(\sum_{n=1}^{N} z_{nk}\right).$$

Bibliography

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton U. P., 2008.

B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80. IEEE, 2010.

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

K. Anstreicher and S. Burer. D.c. versus copositive bounds for standard qp. *Journal of Global Optimization*, 33(2):299–312, October 2005.

P. Auer and R. Ortner. A boosting approach to multiple instance learning. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2004.

B. Babenko, M-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

F. Bach and Z. Harchaoui. Diffrac : a discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2010.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of the Conference on Computational Learning Theory (COLT)*, 2004.

K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, 1998.

A. Berman and N. Shaked-Monderer. *Completely Positive Matrices*. World Scientific Publishing Company, 2003.

D. Bertsekas. *Nonlinear programming*. Athena Sci., 1995.

P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

A. Blake, C. Rother, and V. Kolmogorov. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

David M. Blei and Jon D. Mcauliffe. Supervised topic models. In *In preparation*. MIT Press, 2008.

H. Blockeel, D. Page, and A. Srinivasan. Multi-instance tree learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Conference on Computational Learning Theory (COLT)*, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge U. P., 2003.

Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, 2001.

Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2001.

Claude R. Brice and Claude L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3–4):205–226, 1970.

P. Brucker. An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3:163–166, 1984.

S. Burer. Optimizing a polyhedral-semidefinite relaxation of completely positive programs. *Mathematical Programming Computation*, 2(1):1–19, March 2010.

S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95: 329–357, 2003. ISSN 0025-5610.

L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2007.

J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3241–3248. IEEE, 2010.

O. Chapelle, B. Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT press, 2006.

Yixin Chen and James Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, December 2004.

D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, may 2002.

P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53:475–504, 2004.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, june 2007.

T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Timothee Cour, Ben Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

A. d'Aspremont and S. Boyd. Relaxations and randomized methods for nonconvex qcqps. *EE392o Class Notes, Stanford University*, 2003.

A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G.R.G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

F. De la Torre and T. Kanade. Discriminative cluster analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2006.

T. G. Dietterich and R. H. Lathrop. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

O. Duchenne, J.Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2009.

O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2011.

I. Endres and D. Hoiem. Category independent object proposals. *Computer Vision–ECCV 2010*, pages 575–588, 2010.

L. Fausett, editor. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994. ISBN 0-13-334186-0.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.

S.E. Fienberg. An iterative procedure for estimation in contingency tables. *The Annals of Mathematical Statistics*, pages 907–917, 1970.

D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2003. ISBN 9780131911932.

D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. *Object Representation in Computer Vision II*, pages 335–360, 1996.

E. Frazzoli, Z.H. Mao, JH Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. 1999.

A. Frieze and M. Jerrum. Improved approximation algorithms for max k-cut and max bisection. *Algorithmica*, 18:61–77, 1997.

T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi–instance kernels. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2002.

P. V. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

M. X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, October 1996.

S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the International Conference in Computer Vision (ICCV)*, pages 1–8. IEEE, 2009.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21, August 2010.

M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Y. Guo and D. Schuurmans. Convex relaxations of latent variable training. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 482–496, 2010.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *Proceedings of the European Conference in Computer Vision (ECCV)*, pages 224–237. Springer, 2010.

N. Heess, N. Le Roux, and J. Winn. Weakly supervised learning of foreground-background segmentation using masked rbms. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 9–16, 2011.

D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2009.

T.D. Hocking, A. Joulin, F. Bach, and J.P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2137–2144. IEEE, 2006.

E. Hullermeier and J. Beringer. Learning from ambiguously labeled examples. In *Intell. Data Analysis*, 2006.

D. R Hunter and K. Lange. A tutorial on mm algorithms. *The American Statistician*, 58 (1):30–37, February 2004.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

R. Jin and Z. Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1999.

A. Joulin and F. Bach. A convex relaxation for weakly supervised classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

A. Joulin, F. Bach, and J. Ponce. Efficient optimization for discriminative latent class models. In *Advances in Neural Information Processing Systems (NIPS)*, 2010a.

A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010b.

A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

G. Kim, E.P. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation vis submodular optimization on anisotropic diffusion. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2011.

Pushmeet Kohli, L'ubor Ladický, and Philip Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82:302–324, 2009.

V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

P. Krähenbühl and V. Koltun. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. NIPS, 2011.

J. T. Kwok and P. Cheung. Marginalized multi-instance kernels. In *In Proceedings of International Joint Conference on Artificial Intelligence*, 2007.

Simon Lacoste-Julien. *Discriminative Machine Learning with Structure*. PhD thesis, University of California, Berkeley, 2009.

G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, December 2004.

H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. 2008.

Y. J. Lee and K. Grauman. Collect-cut: Segmentation with top-down cues discovered in multi-object images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3185–3192, june 2010.

Y.J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2011.

A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297, dec. 2009.

C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1972–1979. IEEE, 2009.

D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

L. Lovász. On the shannon capacity of a graph. *Information Theory, IEEE Transactions on*, 25(1):1–7, 1979.

Z. Luo, W. Ma, A.M.C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *Signal Processing Magazine, IEEE*, 27(3):20–34, 2010.

N. Maculan, J. R. de Paula, and G. Galdino. A linear-time median-finding algorithm for projecting a vector on the simplex of rn. *Operations Research Letters*, 8(4):219–222, 1989.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

T. Malisiewicz and A.A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference (BMVC)*, 2007.

O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1998.

D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt & Company, 1983.

M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

F. Meyer. Hierarchies of partitions and morphological segmentation. In *Scale-Space and Morphology in Computer Vision*. Springer-Verlag, 2001.

L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. URL http://www.biostat.wisc.edu/~vsingh/mcoseg.pdf.

D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 1985.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

S. E. Palmer. *Vision Science: Photons to Phenomenology*. MIT press, 1999.

Theodosios Pavlidis. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing*, 1(4):360–372, 1972.

N. Quadrianto, T. Caetano, J. Lim, and D. Schuurmans. Convex relaxation of mixture regression with efficient algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

S. Ray and M. Craven. Supervised Versus Multiple Instance Learning: An Empirical Comparison. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

L. Roberts. *Machine perception of 3-d solids*. PhD thesis, Massachusetts Institued of Technology, 1965.

C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

Michael Rubinstein, Ce Liu, and William T. Freeman. Annotation propagation in large image databases via dense image correspondence. 2012.

B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

J Shawe-Taylor and N Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

J. Shi and J. Malik. Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 1997.

N. Srebro, J.D.M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 1329–1336. Cambridge, MA, 2005.

P. H. Tan and L. K. Rasmussen. The application of semidefinite programming for detection in cdma. *IEEE Journal on Selected Areas in Communications*, 19(8):1442–1449, September 2006.

J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. *Computer Vision–ECCV 2010*, pages 352–365, 2010.

A. Torralba and A.A. Efros. Unbiased look at dataset bias. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528. IEEE, 2011.

A. Vezhnevets, V. Ferrari, and J. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *Proceedings of the International Conference in Computer Vision (ICCV)*, pages 3249–3256. IEEE, 2011.

S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: models and optimization. In *Proceedings of the European Conference in Computer Vision (ECCV)*, 2010.

P. Viola, John C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

M. J. Wainwright and M. I. Jordan. *Graphical models, exponential families, and variational inference*. Foundations and Trends in Machine Learning, 2008.

Hua-Yan Wang, Qiang Yang, and Hongbin Zha. Adaptive p-posterior mixture-model kernels for multiple instance learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

J. Wang and J. Zucker. Solving multiple-instance problem: A lazy learning approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.

M. Wertheimer. Laws of organization in perceptual forms. *Psycologische Forschung*, 4:301–350, 1923.

J. Winn and N. Jojic. Locus: learning object classes with unsupervised segmentation. In *Proceedings of the International Conference in Computer Vision (ICCV)*, 2005.

A. S. Wright, A. S. Wright, and S. T. Acton. Watershed pyramids for edge detection. In *In IEEE Int. Conf. on Image Processing*, pages 578–581, 1997.

L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, 2005.

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *Proc. of the PacificAsia Conf. on Knowledge Discovery and Data Mining*, 2004.

C. Yang. Image database retrieval with multiple-instance learning techniques. In *Proceedings of the International Conference on Data Engineering*, 2000.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. pages 239–269, 2003.

A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007a.

Kai Zhang, Ivor W. Tsang, and James T. Kwok. Maximum margin clustering made practical. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007b.

M. Zhang and Z. Zhou. Adapting rbf neural networks to multi-instance learning. *Neural Processing Letters*, 23(1):1–26, 2006.

Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

Z. Zhou and J. Xu. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.

Xiaojin Zhu. Semi-supervised learning literature survey, 2006.