2009

Learning from text and images: generative and discriminative models for partially labeled data

Oksana Yakhnenko
Iowa State University

Learning from text and images: generative and discriminative models for partially labeled data

by

Oksana Yakhnenko

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:
Vasant Honavar, Major Professor
Drena Dobbs
Yan-Bin Jia
Jack Lutz
Dimitris Margaritis

Iowa State University

Ames, Iowa

2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express deep gratitude to my colleagues, mentors, friends and family without whose support this dissertation would not have been possible.

First, I would like to thank my major professor, Vasant Honavar, with whom I have started working with as an undergraduate. For the past six years, Vasant has been an inspirational adviser, teacher, mentor and collaborator. His continuous support and insightful discussions helped me find interesting research problems, without losing sight of the big picture.

I would like to thank my PhD committee members: Jack Lutz for discussions about theory of computing and especially for his advice and sharing many of his personal experiences about a journey from being a student, to PhD student, to a researcher and becoming a life-long mentor. Dimitris Margaritis for insightful discussions and feedback many research projects. Drena Dobbs for introducing me to the world of molecular biology. Yan-Bin Jia for discussions on computational geometry and robotics.

I would like to thank Barbara Rosario who I have had the pleasure of working with during my internship at Intel Research for introducing me to the world of Natural Language Processing and for becoming my role-model, mentor and friend. I would like to thank my colleagues and mentors during the two internships at Siemens Medical Solutions: Lucian Lita for his continuous advice, support and friendship. Balaji Krishnapuram, for teaching me skills of thinking from the first principles when it comes to problem-solving and finding new research projects, for inviting me to organize the KDD cup and the workshop at NIPS 2008. Romer Rosales, Stefan Niculescu, Vikas Raykar and Shipeng Yu for many helpful discussions.

I would like to thank current and former students in my lab for helpful discussions during seminars, lunches and coffee breaks: Adrian Silvescu, Cornelia Caragea, Fadi Towfic and Kewei

Tu.

None of my work would have been possible without the unconditional love and support of my family: my parents, Katya and Volodya and my brother Denys. I love you very very much.

I would like to thank all my friends and Taryn Packheiser, Paula Herrera, Emilie Slaby, Sheng Ly, Michelle Ruse, Jose-Pablo Soto Arias, Clotilde Aceto, Ehsan Kayal, Zakria Hussain, Jean-Pierre Tautel. Finally, I would like to thank Rishi Ganti for sharing my passions for science, applied gradient descent in the extreme weather conditions, dance and art, and making me believe that anything is possible despite distance and time.

During my graduate studies I was supported in part by research assistantships funded by National Science Foundation (IIS 0711356) and the Iowa State University Center for Computational Intelligence, Learning and Discovery and teaching assistantships in Computer Science Department.

ABSTRACT

Image annotation is a challenging task of assigning keywords to an image given the content of an image. It has a variety of applications in multi-media data-mining and computer vision. Traditional machine learning approaches to image annotation require large amounts of labeled data. This requirement is often unrealistic, as obtaining labeled data is, in general, expensive and time consuming. However, large amounts of weakly labeled data and tagged images is readily available, in particular in the web and social network communities. In this thesis we address the problem of image annotation using weak supervision. In particular, we formulate the problem of image annotation as multiple instance multiple label learning problem and propose generative and discriminative models to tackle this learning problem. Multiple instance multiple label learning is a generalization of supervised learning in which the training examples are bags of instances and each bag is labeled with a set of labels. We explore two learning frameworks: generative and discriminative, and propose models within each framework to address the problem of assigning text keywords to images.

The first approach, the generative model attempts to describe the process according to which the data was generated, and then learn its parameters from the data. This model is a non-parametric generalization of the known mixture model used in the past. We extend this model to a Hierarchical Dirichlet Process which allows for countably infinite mixture components. Our experimental evaluation shows that the performance of this model does not depend on the number of mixture components, unlike the standard mixture model which suffers from over-fitting for a large number of mixture components.

The second approach is a discriminative model, which unlike generative model answers the following question: given the input bag of instances what is the most likely assignment of labels to the bag. We address this problem by learning as many classifiers as there are possible labels and force the classifiers to share weights using trace-norm regularization. We show that the performance of this model is comparable to the state-of-the-art multiple instance multiple label classifiers and that unlike some state-of-the-art models, it is scalable and practical for datasets with a large number of training instances and possible labels.

Finally we generalize the discriminative model to a semi-supervised setting to allow the model take advantage of labeled and unlabeled data. We do so by assuming that the data lies in a low-dimensional manifold and introducing a penalty that enforces the classifiers assign similar labels to indirectly similar instances (i.e. instances that are near-by in the manifold space). The manifold is learned by constructing a similarity neighborhood graph over bags, and then graph-Laplacian is used to compute the penalty term.

CHAPTER 1. Learning from Images and Text: an Overview

1.1 Introduction and Motivation

Recent years have witnessed rapid advances in our ability to acquire and store massive amounts of data across different modalities (such as text, speech, images, etc.). The availability of such data presents us with challenges of information organization, retrieval of images based on user text, identification of regions of interest in the image based on user text and so on. In computer vision, it is not only interesting whether a particular object is present or absent in the scene, but rather what the objects are, where they are located and what the relationships between the objects are. Thus the efforts are gearing towards learning models for scene understanding. Traditional supervised learning solutions to these problems require large amounts of labeled data, and manually labeling data becomes prohibitively time-consuming and expensive. While there are large sets of labeled images available (such as LabelMe (Russell et al., 2005)) most of the images labeled in these datasets are urban, indoor or natural scenes and they may not always be used as training images for all tasks. For example, labeled objects in urban and natural scenes scenes may be useful to identify people, animals, trees or flowers, but they will be useless when the goal is to identify musical instruments, therefore a additional labeled datasets will be needed to train classifiers for such tasks.

In this thesis present several solutions to automatic image annotation and content-based image retrieval that do not involve the use of fully-labeled data. We do so by using weakly labeled data and unlabeled data.

We begin with giving a brief introduction to the problem of computer vision and scene understanding, and motivate the problem of scene understanding as learning to predict textual description of the images and image annotation. We then describe related work in image annotation, identify key challenges in image annotation, and propose our solutions to the challenges. We then describe the roadmap of the thesis.

1.2 Scene Understanding in Computer Vision

The general goal of *computer vision* (Forsyth and Ponce, 2002; Szeliski, 2009) is to develop mathematical and computational techniques that enable computers perform tasks easily performed by humans - perceive two-dimensional and three-dimensional world and be able to, for example, identify the color and shape of shoes and shirts and other clothing items from a cluttered background by looking inside a closet, or from a crowd of people identify familiar faces and name the familiar people, and perhaps recognize their postures and emotions. Computer vision continues to present many unsolved challenges. One of the goals of computer vision is to develop computational models that in some way follow mechanism of visual perception of humans an animals. However understanding how human vision works is mainly unsolved and up to day contains many puzzle pieces (Marr, 1982; Palmer, 1999). One of the main challenges of computer vision as identified by Szeliski (2009) is that it is an *inverse problem*: instead of modeling the visual world as in computer graphics, computer vision attempts to solve the problem of identifying the properties (such as shape, brightness, segmentation) from images or videos.

Scene understanding is a subfield of computer vision aimed at building models to explain the content of images and scenes (Winston, 1975; Hanson and Riseman, 1978). Some of the basic questions addressed by scene understanding can be viewed as classification problems. These questions include, but are not limited to: What kind of scene is it (urban, sea, jungle...)? What objects are present in the scene? Where are the different objects located relative to each other? etc. In a study of human scene understanding conducted by Fei-Fei et al. (2007), the human subjects were asked to describe (in English) what they saw in photographs. The descriptions of the study subjects included the identities and locations of different objects and the relations between the objects in the scene. Some of the recent work in scene understanding has focused on systems that can simultaneously segment the image into regions and identify the object class that the region belongs to (Torralba, 2008; Li et al., 2009). A more challenging task is to build models that, in addition to performing segmentation, identify the objects and relationships between the objects and environment in a scene.

These three problems (image segmentation, object recognition and relationship identification) solving which is a necessary step for complete scene understanding, can be addressed in three separate steps: perform image segmentation using a segmentation algorithm (such as Normalized Cuts (Shi and Malik, 2000)) and recognize objects using a classifier (Heisele, 2003; Ponce et al., 2007). The task of identifying relations between objects is a fairly new problem with the additional difficulty that the relationship may be spatial (for example, whether the person is above or below the table), or semantic (for example, whether the person is *holding* a pen), however some attempts have been made to utilize text and prepositions to learn such relationships (Gupta and Davis, 2008). A big challenge is that solving the problem of object recognition requires a large amount of labeled data specific to the problem. Acquiring labeled data can be expensive and time-consuming and the resulting labels may be ambiguous.

While labeled data can be expensive to obtain, there are large amounts of tagged data, weakly-labeled data and unlabeled data. Therefore there arises the question: Is it possible to describe the scene without the need of fully-labeled data? A field that addresses the question of assigning keywords to the image given the image content (and the one we focus on in this thesis) is the field of image annotation.

1.2.1 Automatic Image Annotation

The wide availability of tagged data faces us with the problem of *automatic image annotation*: given an image (or a collection of image segments) as input, the task is to predict a set of keywords from a prespecified vocabulary that describe the contents of the image. The individual keywords may or may not describe the individual objects that appear in the image. For instance, a scene that contains a collection of objects such as *cars*, *pedestrians*, *skyscrapers*, etc., may be annotated with the keyword *city* although none of the individual objects in the image can be labeled with that keyword. We will give an overview of general methods for image annotation as we describe the history of work in image annotation later on. The general work in image annotation falls into two categories:

1. *Generative models* (Barnard et al., 2003; Duygulu et al., 2002; Blei and Jordan, 2003; Feng et al., 2004) these models attempt to describe the process according to which the image and the corresponding text was generated

2. *Discriminative Multiple Label classification models* that learn to discriminate most likely keywords from the image content (Loeff and Farhadi, 2008; Makadia et al., 2008).

1.2.2 Applications

One of the major application of image annotation is semantic image retrieval for web search engines. The user may be interested in typing in a text query and retrieve images that satisfy the semantic context of the text query. One way of retrieving the relevant images is to first annotate the images with the semantic (text) concepts, and then present the user with the images that have the text that is most similar to the user entry. Web search engines are the immediate domains that call for image annotation, however recently other fields such as medical imaging and computer-aided diagnostics began using automatic image annotation (Yao et al., 2006; Caputo et al., 2009).

1.2.3 Image Processing and Image Representation

Images are naturally encoded as matrices of color pixels. Direct use of pixel color intensities may cause several problems. The images usually have different number of pixels and information may be lost under rescaling. Even slight variation in colors can cause similar images have drastically different feature values. Simple pixel color intensity based features are neither translation nor rotation invariant. Hence, much work in scene understanding has focused on sophisticated methods for feature extraction and image representation (Forsyth and Ponce, 2002; Nixon and Aguado, 2008; Szeliski, 2009) which remain active areas of research in computer vision. However the primary focus of this thesis is not on the design of optimal feature extraction methods. Our primary focus is on the design of models that make the best use of the available features in image understanding.

We take advantage of several feature extraction methods that have been shown to yield good results in the past.

1. The first method relies on local features. We use local features - i.e., features extracted from small patches all over the images (Lowe, 2004; Fei-Fei and Perona, 2005). First, the patches are extracted from the training data, quantized into a codebook using a clustering method (such as k-means), and then each image is represented as a bag of indexes where each extracted patch belongs in the codebook. Such representation has been shown to be invariant to scale and occlusions. This representation is used in our generative model for image annotation(Chapter 3).

2. The second method relies on first segmenting an image (with assumption that each segment corresponds to an object) and then extracting features from each segment. The extracted features represent major visual properties, such as color, shape, size, etc. Then the image is represented as a collection of the segments. Such representation has been shown to be effective for image annotation (Duygulu et al., 2002; Lavrenko et al., 2003). We also take advantage of texton features (Shotton et al., 2006) to encode the texture of the images.

1.3 Previous Work

Detailed discussion of the large body of work on scene understanding, although relevant, will distract us from the main focus of this thesis. We limit our discussion to the work that is most closely related to the main focus of this thesis, namely, image annotation. Because we approach image annotation using tools of machine learning, we start with a brief survey of work on image annotation, followed by a discussion of work on Multiple Instance learning, Multiple Label learning, and Multiple Instance Multiple Label learning that form the basis of the approaches developed in this thesis.

Related Work in Multiple Instance Learning Multiple Instance learning (initially introduced by Dietterich et al. (1997)) is a generalization of a standard supervised learning in which a training instance is not a single instance (i.e. a vector, a sequence, etc), but rather a collection or a bag of instances. The bag is labeled positive if at least one instance in the bag is positive, however it is not known which one. A bag is labeled negative if it contains no positive instances (therefore the negative instances are labeled "for free").

There exists a large body of work in the Multiple Instance learning (see surveys such as (Ray and Craven, 2005)), and here we summarize some of the key algorithms. Dietterich et al. (1997) used axis-parallel rectangles approach so solve the Multiple Instance learning problem: the axis parallel rectangle in the feature space should contain the positive instances and no negative instances.

Diverse Density (Zhang and Goldman, 2001) attempts to find a point in the feature space that is close to at least one point in the positive bags and far away from the points in the negative space. Diverse Density models the probabilities that the instances are positive using a Gaussian-distribution-like model, and then combines the probabilities to model that bags are positive or negative using a Noisy-Or model (Maron and Lozano-Pérez, 1997; Viola et al., 2005). The model parameters are optimized using gradient descent method. In a similar spirit, the Expectation Maximization Diverse Density approach (Maron and Lozano-Pérez, 1997) models instance labels as "hidden" variables, and in iterative fashion uses the model to assign values to the hidden variables, then uses these values to re-estimate model parameters until convergence. Xu and Frank (2004) proposed a boosting approach to Multiple Instance learning and a logistic regression variant for Multiple Instance learning. The boosting algorithm generalizes a single instance AdaBoost (Freund and Shapire, 1996) by minimizing the expected loss of weak classifiers over bags as opposed to single instances as proposed in the original AdaBoost. The proposed logistic regression algorithm models the probability of an instance using a logistic regression model, and then combined the probabilities of the instances in the bag by averaging them.

Recently, Raykar et al. (2008) used a logistic model to model the probability that the instance is positive and then combined the probabilities using a Noisy-Or model. They also proposed to use a prior to allow for feature selection, and this approach showed to yield the state of the art results on several the Multiple Instance learning benchmark datasets.

Related Work in Multiple Label Learning Multiple Label classification is another generalization of traditional supervised learning. It assumes a single instance, however with each instance multiple output variables may be associated. The term "multiple label learning" has also been used by Jin and Ghahramani (2002) in a setting where multiple output variables are associated with the instance, however only one of them is the correct label. From here on "multiple label" learning will be used in reference to the first scenario.

It was first studied in the context of text categorization (McCallum, 1999) and the solution proposed was to consider a power-set over the possible label assignment and then to learn a single label classifier where the label can take as many values as the size of the power set. This approach however, can only deal with a small number of possible labels. Alternatively, Boutell et al. (2004); Zhang and Zhou (2006), suggested learning as many binary classifiers as there are labels, and then use these classifiers to assign labels for each instance. A survey of these two approaches and their applications are described in (Tsoumakas and Katakis, 2007).

A more recent work has focused on solving Multiple Label problem by accounting correlations among labels. Ghamrawi and McCallum (2005) used an undirected model to model the probability of a set of labels assigned to an instance. Such approach, however, results in the need to compute a summation over all possible label assignment and thus exact solution is intractable. More recently, Amit et al. (2007) proposed to train as many classifiers as there are possible labels (one for each label), and then correlated the classifiers using a special regularization penalty (Trace Norm) which forces the classifiers find the low-rank solution, thus enforcing the classifiers to share weights.

Related Work in Multiple Instance Multiple Label Learning Multiple Instance Multiple Label learning is a recent framework proposed by Zhang and Zhou (2006). This framework combines Multiple Instance learning and Multiple Label learning, and the learning is done over bags and sets of labels. Zhang and Zhou (2006) proposed two solutions to this problem: 1) converting Multiple Instance learning into Single Instance learning and then applying Single Instance Multiple Label learning (this approach resulted in Multiple Instance Multiple Label SVM) and 2) generalizing a Multiple Instance learning algorithm to Multiple Label algorithm (this resulted in Multiple Instance Multiple Label boost). Zhang and Zhou (2008) proposed a learning algorithm which attempts to solve both problems simultaneously by formulating a margin over bags and Multiple Labels. A major disadvantage of this approach is its scalability to datasets with a large number of bags and labels. The soft margin formulation relies on the presence of slack variables so that each instance in each class has a slack variable. Therefore, there are at least *number of total instances* × *number of total classes.* This problem will not be feasible to solve for large datasets (due to memory and time restrictions).

Zha et al. (2008) proposed a discriminative model based on a collective Multiple Label model (Ghamrawi and McCallum, 2005) which was shown to yield state-of-the-art results on several image classification tasks that are naturally posed as Multiple Instance Multiple Label learning problems They proposed an undirected graphical model, and in order for it to model a valid probability of labels given the bags, it requires the summation over all possible label assignments for each bag, and therefore, exact computation is exponential in the number of labels. Zha et al. (2008) proposed an approximation for this computation using Gibbs sampling, however 1) the exact solution is intractable and 2) Gibbs sampling can be slow for a large number of labels. Due to this, this algorithm is also not suitable for large datasets with a large number of possible labels.

Very recently Vijayanarasimhan and Grauman (2009) proposed a Multiple Instance Multiple Label algorithm which relies on a Multiple Instance kernel (Gärtner et al., 2002) and one-vs-one SVM training. As with previous formulations, this algorithm is only feasible to work with datasets with a small number of possible labels.

Related Work in Image Annotation Image annotation was first introduced by Picard and Minka (1995) where a visual system was used to propagate users' tags to other visually similar images in the database to account for ambiguity of users' textual description of the visual scenes. Barnard and Forsyth (2001) introduced the problem of learning semantics of images and the associated text: given the images and the associated caption, the goal was to construct a model that captured the correlation among visual and textual features of the image. They introduced a probabilistic latent semantic model that was trained using expectation maximization. Later, Duygulu et al. (2002) introduced a novel formulation of the image annotation problem as a variant of the machine translation problem. A scene to be labeled is encoded in a source language as a collection of regions in the image (obtained after segmenting the image). Sentences in the target language are encoded as a collection of of keywords in the caption. Barnard et al. (2003) have examined several solutions for the image annotation and image-object label correspondence problems. They developed several models for the joint distribution of image regions and words, including those that explicitly learn the correspondence between image regions and words. They studied a multi-modal and correspondence extensions to hierarchical mixture models (Hofmann and Puzicha, 1999), and probabilistic latent semantic indexing (pLSI) (Hofmann, 1999) for text, a translation model adapted from statistical machine translation (Brown et al., 1993), and a multi-modal extension to mixture of Latent Dirichlet Allocation (MoM-LDA) (Blei and Jordan, 2003) that generalizes LDA (Blei et al., 2003) to the setting where the data combines multiple modalities (e.g., image, text).

This translation approach was later extended to generative models that capture associations between image regions and keywords e.g., the Correspondence Latent Dirichlet Allocation (CorLDA) model (Blei and Jordan, 2003). Lavrenko et al. (2003) proposed a Continuous Relevance Model (CRM), that modeled the joint probability distribution of continuous image features and words. The model then could be used to predict the probability of generating a caption word given the image regions.

CorLDA model can be viewed as a probabilistic interpretation to canonical correlation analysis. Hardoon et al. (2006) have explored a kernelized version of canonical correlation analysis for image retrieval and annotation. Specifically, they show how a semantic representation of images and their associated text can be learned and how the resulting representation in a common semantic space can be used to compare data from the text and image modalities. However, the primary focus of this work was not on solving the image object-label correspondence problem.

Feng et al. (2004) proposed a Multiple Bernoulli Relevance Model (MBMR) that models the joint probability of absence and presence of the keywords in the caption and the image regions.

Recently, Makadia et al. (2008) have introduced a new baseline (Joint Equal Contribution Model, or JEC) for image annotation using a variant of image-content-based keyword retrieval problem. Given a target image to be annotated, JEC finds k images that are most similar to the target image in the training data. The target image is then annotated by a process of *greedy label transfer* from the k nearest neighbors to the target image. This simple method was shown to outperform several previous state-of-the-art approaches, including CorLDA (Blei and Jordan, 2003), as well as MBRM (Feng et al., 2004) and SML (Carneiro et al., 2007), on several datasets including the widely used benchmark Corel-5k dataset (Duygulu et al., 2002). Like several of the previous methods, the nearest neighbor approach also requires the training data to be stored in order to be able to identify the nearest neighbors of a target image to be annotated. Loeff and Farhadi (2008) introduced an algorithm for image annotation that uses a Single Instance classifier for each keyword, and forces the classifiers to share weights using Trace Norm regularization. This algorithm achieved results comparable to JEC.

1.4 Challenges in Image Annotation and Proposed Solutions

Given the body of work and the history in image annotation, we now identify some of the main challenges in image annotation:

1. *Whether to model the distribution of images and text or to solve classification problem*

   Problem The classification task can be solved in two distinct yet related frameworks: generative and discriminative. The goal of generative models is to model the underlying process of how the data is generated. In contrast, the goal of discriminative models is to solve classification problem directly. A natural question that arises has to do with which of the two frameworks is more suitable for image annotation. In particular, the related work in image annotation falls into 2 categories: generative models (Duygulu et al., 2002; Barnard et al., 2003; Lavrenko et al., 2003; Carneiro et al., 2007), and discriminative models (Loeff and Farhadi, 2008). The positives and negatives of each framework are described Chapter 2.

Solution *Generative and Discriminative models for image annotation:* We first revisit the generative model for image annotation and address the problem of selecting the number of mixture components by making use of a stochastic process that generates the number of mixture components. Then, we propose a discriminative model for image annotation. The parameters of the discriminative model are learned to maximize the probability of predicting the caption correctly given the bag of image segments and their descriptors. This model is also a novel and effective learning algorithm for general Multiple Instance Multiple Label learning.

2. *Selecting the number of latent components for the Generative Latent Mixture Model*

Problem Several generative models proposed for image annotation (Duygulu et al., 2002; Barnard et al., 2003; Lavrenko et al., 2003), rely on the assumption that the image features and text are correlated through latent hidden semantic components. These generative models in spirit are similar to latent mixture models: the hidden mixture components represent the unobserved semantic and the observed text and image features are dependent through the mixture components. The performance of these mixture models highly depend on the number of the mixture components - if the number is too small, the model may have a poor fit and if the number is too large, the model may fit very well on the training data, but not generalize well on the unseen data. Typically, a different number of mixture components is tried, and the number that yields the best performances as measured on the validation set is used. A more principled approach that does not rely on trial and error in selecting an optimal number of mixture components is desirable in practice.

Solution We propose a non-parametric approach to the generative models used in the past. In particular, we extend the Mixture-of-Multinomials Latent Dirichelt Allocation model (Duygulu et al., 2002; Barnard et al., 2003) to a Hierarchical Dirichlet Process. Hierarchical Dirichlet Process assumes that for each image-text pair there are countably infinitely many mixture components and these mixture components come from Multinomial distrubitions that share a prior. We present an efficient solution to estimate the model parameters using variational inference (generalization of Expectation Maximization). Our experimental evaluation shows that the performance of this model does not depend on the number of mixture components unlike Mixture-of-Multinomials Latent Dirichlet Allocation.

3. *Coping with lack of word-object correspondence information in training data.*

Problem In tagged images, it is often the case that the objects present in the images may or may not correspond to a keyword in a tag, given rise to three possible scenarios:

(a) Objects in the image correspond to the tags (for example in an urban scene, where a car is present the keyword “car” is also present in the tag)

(b) The keywords which correspond to some objects may not appear in the tag (for example, an image that contains clouds and grass may not always be annotated with the keywords “grass” and “sky”)

(c) There may be keywords in the annotation which do not correspond to any of the objects in the image (for example, an urban scene containing cars, buildings and pedestrians may be tagged as “city”, and while none of the individual objects has the label “city” all objects collectively contribute to the tag

Although the Multiple Instance Multiple Label (MIML) learning framework (Zhang and Zhou, 2006) addresses these problems, the most current state-of-the-art MIML algorithms (Zhang and Zhou, 2008; Zha et al., 2008; Vijayanarasimhan and Grauman, 2009) are impractical in settings with for a large vocabulary of possible labels.

Solution We tackle the problem of lack of correlation between image objects and keywords by introducing a discriminative Multiple Instance multiple label model. In this model, the contributions of the instances (corresponding to image segments) to each keyword in the caption are modeled using a Noisy-OR model and the correlation among keywords are modeled by using a regularization that forces the classifiers for each label in the vocabulary to share their parameters. Our experiments show that this algorithm yields predictors whose classification performance is comparable to the state-of-the-art algorithms. Unlike previous state-of-the-art approaches, it scales well to settings in which the number of possible labels in the vocabulary used to annotate the images is very large.

4. *Scarcity of labeled data*

Problem As mentioned previously, training a model using a fully supervised learning approach requires a fully labeled training corpus. However, in practice, only a small fraction of the available data is labeled. Hence, there is a need for learning algorithms that can take advantage of weakly labeled data and tagged data, as well as unlabeled data.

Solution We propose to address this problem by taking advantage of weakly-labeled and unlabeled images. The generative and discriminative models take advantage of the image and the image caption only (without the need of individual regions to be labeled). In addition, we propose a novel solution to incorporate the use of unlabeled data (images for which no text description is known) in constructing predictors of caption words for the multiple instance multiple label approach.

1.5 Thesis Overview

In this thesis we address the problem of learning from images and text and, in particular, the problem of image annotation from a machine learning point of view. Given an image and its associated caption, the goal is to construct a model which can predict a set of keywords given a novel image. If an image is represented as a feature vector, this problem can be viewed as *Multiple Label* prediction. Frequently, the keywords in the image caption represent the objects that appear in the image. Given an image segmentation, the image segments may correspond to some of the objects keywords for which appear in the image caption, however it is not known which segment corresponds to which object. In this case, the learning problem can be formulated as *Multiple Instance Multiple Label learning problem.*

We begin by revisiting a generative model for image annotation proposed in Barnard et al. (2003) and we generalize this model to a Hierarchical Dirichlet Process to circumvent the problem of model selection. We then propose a discriminative model, aimed to solve the problem of image annotation directly. We then extend the discriminative model to allow the use of unlabeled data by incorporating geometry of the data into the model.

We begin with motivating the choice of the models. The Machine Translation model (MT (Duygulu et al., 2002)), Continuous Relevance Model (CRM (Lavrenko et al., 2003)), and Multiple Bernoulli Relevance Model (MBRM (Feng et al., 2004)) models can be understood within a unifying framework as follows: In each case, one assumes that the image regions x and keywords y are independent given some hidden variable z: $p(x, y|z) = p(x|z)p(y|z)$. In the case of MT and CorLDA the hidden variable denotes some *semantic component* that correlate the image region features and textual description of the image. In the case of CRM and MBRM, the hidden variable denotes an *image*. In the latter case, annotating a novel image involves computing the expectation over the training data (thus the time to annotate the test image increases with the size of the training dataset used). A big challenge is the selection of the dimensionality of the hidden variable: if it is too small, the model fits the data poorly; if it is too large, the model fits the training data however it may not generalize well on the test data. Typically, different number of mixture components are tried, and tested on a validation set. Then the number of mixture components that yields the best performance is chosen. The second approach is to assume that instead of a fixed distribution for the mixture components, they are generated according to a stochastic process and there can be infinitely many mixture components. Thus we address the problem of selection of mixture components by taking the latter approach, and by assuming a Hierarchical Dirichlet Process. We describe this model in detail in Chapter 3.

Since the goal of in image annotation the is the accurate reconstruction of the caption, it may be a better alternative to avoid constructing the generative model, and instead attempt to solve the problem of caption reconstruction directly. One such attempt was by Carneiro et al. (2007) where they introduced an approach that models the joint probability of regions and words directly: $p(x, y) = p(x|y)p(y)$. This involves estimating the probability of image features conditioned on the words that appear in the corresponding image captions. We propose a model that describes the the probability of caption given the image by modeling $p(y|x)$ directly. This model is described in Chapter 4 and it addressed using the Multiple Instance Multiple Label learning. Each image is viewed as a bag of instances, where each instance contains the descriptors (features) for each image segment. The model minimizes the trade-off between the Log Loss over bags of vectors and regularization that enforces the classifiers for each label to share weights, thus modeling correlation among the labels. Our empirical evaluation on Microsoft Visual classes dataset shows that this model yields the new state-of-the-art results. In addition, we present evaluation of this algorithm on several large datasets (including the benchmark Corel-5k) and we show that unlike other state-of-the-art algorithms, our model is scalable for datasets with the large number of instances and the large number of possible keywords.

Finally we tackle the problem of using the available unlabeled data by allowing the learning algorithm to incorporate labeled and unlabeled data. We propose a framework to generalize the discriminative Multiple Instance Multiple Label model. This framework incorporates graph-based regularization on instance and label level. In addition to trading-off between the loss and penalty, it forces the classifiers to have similar values for the points that are near-by, and for labels that are correlated. In this framework unlabeled data comes in naturally thus allowing semi-supervised learning.

This thesis is organized as follows:

- In Chapter 2, we establish the notation and introduce preliminary concepts.

- In Chapter 3, we describe the generative latent variable correlation model and address the problem of model selection using a Hierarchical Dirichler Process.
- In Chapter 4, we present the discriminative model for multiple instance Multiple Label learning.
- In Chapter 5, we generalize the model described in Chapter 4 to a semi-supervised setting.
- Finally, in Chapter 6, we summarize the main ideas of the thesis and conclude with discussion and directions for future work.

1.6 Thesis Contributions

The contributions of this thesis fall in the areas of computer vision and machine learning. The major contributions are:

1. Non-parametric Bayesian model

   - We develop a new multi-modal generative model based on a Hierarchical Dirichlet Process (Chapter 3). This model generalizes previous work in image annotation, and in particular it generalizes Mixture of Multinomial Latent Dirichlet Allocation model (Barnard et al., 2003). Unlike previous work, however, the number of mixture components is not fixed, and the model adapts the number of mixture components to the data, thus circumventing the problem of model selection. Our experimental evaluation shows that unlike previous work, the performance of the model is invariant to the truncation level unlike MoM-LDA, the performance of which depends on the number of mixture components.

2. Multiple Instance Multiple Label learning

   - We develop a novel *discriminative* Multiple Instance Multiple Label model (in Chapter 4). This model trains as many discriminative Multiple Instance classifiers as there are labels and forces the classifiers to share weights, thus enforcing correlations among labels. The discriminative models use a Noisy-Or model to model the probability that the bag is positive with respect to one label, and uses Trace Norm regularization to enforce classifiers share weights. Unlike other state-of-the-art classifiers, it is scalable to the large number of possible labels as it does not require to compute the normalization factor (over all possible label assignments) as in MILML model (Zha et al., 2008) and it does not rely on one-vs-one training of the classifiers (Vijayanarasimhan and Grauman, 2009), however it achieves similar or better performance than these state-of-the-art classifiers.

3. Semi-supervised Multiple Instance Multiple Label learning

   - We develop a framework for semi-supervised Multiple Instance Multiple Label learning that allows the classifier to incorporate unlabeled data. To the best of our knowledge, this is the first formulation of semi-supervised learning in Multiple Instance Multiple Label learning and its application to computer vision. The semi-supervised framework generalizes the discriminative Multiple Instance Multiple Label model (described in Chapter 4) by introducing additional regularization terms. These regularization terms assume that the images lie in a low-dimensional manifold to account for images that may not be visually similar, however are considered to be "indirectly similar" if they share a path in a graph constructed from the nearest-neighbor images (Belkin et al., 2005). The manifold (graph) is recovered by using labeled and unlabeled images. The regularization term enforces the classifiers to assign similar labels to images that share a path on the manifold. Two penalties are used: one on the image level, another on the label level.

CHAPTER 2. Preliminaries

In this Chapter we establish the notation and formalize the learning problems for standard supervised learning, Multiple Label learning, Multiple Instance learning and Multiple Instance Multiple Label learning. We then proceed with introducing two distinct frameworks that can be used for solving the learning problems: generative and discriminative. We describe parameter estimation for each framework and in particular we provide background for Bayesian parameter estimation (which will be used in the model described in Chapter 3).

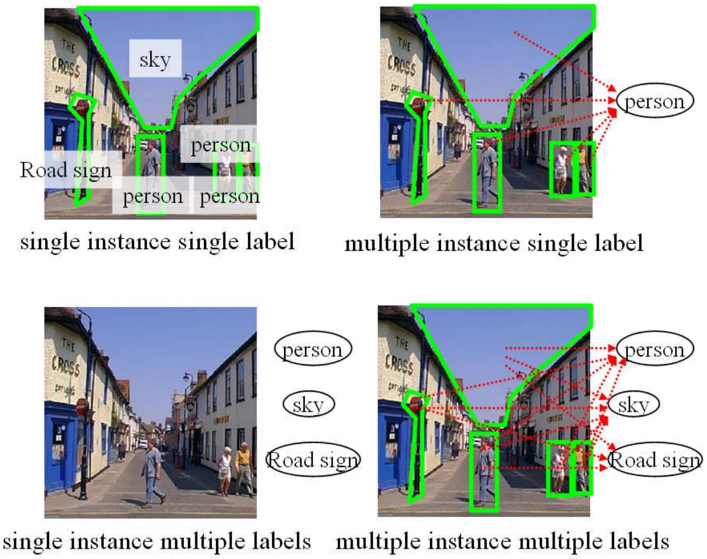
2.1 Learning Scenarios: Single Instance Learning and its Generalizations

We first describe a standard Single Instance Single Label supervised learning problem, then describe its generalization to Multiple Label learning and Multiple Instance learning. We then formalize the core learning problem addressed in this thesis: Multiple Instance Multiple Label learning problem, as well as describe some limitations with the current state-of-the-art algorithms in this area.

2.1.1 Single Instance Classification

We begin with establishing notation. Let $x_1, ...x_n$ be a set of observed instances (i.e. x_i is a vector in R^d, a sequence over a finite alphabet, or features extracted from an image, etc.) generated according to some distribution $P_{\mathcal{D}}$ and let $\mathcal{X}$ be the instance space. Let $\mathcal{L} = \{l_1...l_M\}$ be the label space and let $y_i \in \mathcal{L}$ be a label assigned to each instance x_i. The *standard supervised learning* tasks can then be stated as follows: given a labeled dataset $D = \{\langle x_i, y_i\rangle\}_{i=1}^{N}$ construct a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$ that can accurately predict a label $h(x_t)$ for an unseen instance x_t

A simple example of this is given in Figure 2.1: given image regions and features extracted from the regions (therefore each instance is a vector) and the labels associated with each region, learn to predict labels for unseen regions.



single instance single label
multiple instance single label
single instance multiple labels
multiple instance multiple labels

Figure 2.1 Various Learning scenarios

This formulation, however has some severe limitations in practice: There may be cases in which there are multiple keywords associated with the images (Figure 2.1 bottom right) or given the image regions the exact correspondence between regions and keywords is unknown (Figure 2.1 left). Therefore, the simple supervised learning formulation is extended to Multiple Label learning or Multiple Instance learning, or both.

2.1.2 Multiple Label Learning

The task of image annotation is much more complex than a simple classification set-up introduced above. To cope with the multiple labels that are associated with an image, we consider the multiple label learning scenario. The Multiple Label learning is formalized as follows: each instance x_i has a set of labels $y_i = \left\{ y_i^1, ...y_i^{M_i} \right\}$ (M_i is the number of labels for instance i) assigned to it such that each $y_i^j \in \mathcal{L}$. The goal is to learn a classifier that maps from the instance space to the set of labels $h : \mathcal{X} \rightarrow 2^{\mathcal{L}}$.

As an example, let M be the cardinality of the set formed by unique keywords $\mathcal{L} = \{l_1 ..., l_M\}$ that can be found in the image captions. An image can be represented as a vector of features $x_i \in R^d$ and the keywords associated with the image is a subset of labels from $\mathcal{L}$, and these keywords can be represented as an M-dimensional vector $y_i \in R^M$ so that each position j denotes whether the keyword l_j appeared in the caption ($y_i^j = 1$ if yes, and $y_i^j = -1$ otherwise).

Therefore the task of image annotation can be viewed as *Multiple Label learning* problem. An example is shown in Figure 2.1: given an image (features extracted from the image) and a set of keywords associated with the image, learn a model that assigns a set of keywords to an unseen image.

A vast majority (McCallum, 1999; Boutell et al., 2004; Sotiris et al., 2005; Rousu et al., 2005) of the earlier solutions to Multiple Label prediction problems considered two scenarios:

1. Transform Multiple Label classification problem as M independent classification problems and learn M binary classifiers such that classifier h_j is trained on the dataset $\mathcal{D}_j^{'} = \left\{x_i, y_i^{'}\right\}$ where $y_i^{'} = 1$ if $y_i^j = 1$ and negative otherwise.

2. Transform Multiple Label classification problem into Single Label classification problem by considering a power-set of labels y_i. This transformation results in the exponential number of possible classes ($\mathcal{O}\left(2^M\right)$ in the worst case) and is infeasible if the number of possible labels is large.

A more recent work in Multiple Label prediction problem involves exploiting correlation among labels. Ghamrawi and McCallum (2005) proposed a probabilistic undirected model which models the pair-wise correlation among labels present in one instance. This model is learned by maximizing the probability of predicting the correct labels given the instance. Unfortunately, the computation of this probability requires summation over all possible assignments over labels, and thus it is not feasible if the large number of labels is present. Amit et al. (2007) suggest training M classifiers (one per label) and use a special type of regularization to enforce a low-rank solution and force the classifiers to share weights and capture correlations among labels.

2.1.3 Multiple Instance Learning

Image analysis and classification tasks typically make use of features extracted from the image. Some of the current state-of-the-art approaches to image annotation Makadia et al. (2008); Loeff and Farhadi (2008); Amit et al. (2007) use global features extracted from the images. However, such global features are not especially well-suited in the case of images that consist of multiple objects each of which corresponds to an image segments. In such settings, it is beneficial to first segment the image to be annotated into regions (using an image segmentation algorithm (e.g., the normalized cut Shi and Malik (2000)), extract local features from each resulting segment, and attempt to learn the relationships between the features of image segments and keywords that appear in the annotations associated with images in the training set. Thus, representing the image as a collection of segments may be a more suitable representation for the task of image annotation: given the segmentation of the image and the features extracted from the segments, the image x_i is a collection segments. Let ith image contain K_i segments, then image x_i is given as $x_i = \{x_{i1}, ...x_{iK_i}\}$. Such image representation can be tackled by *Multiple Instance learning* problem. This problem was initially introduced by Dietterich et al. (1997) in the context of a drug activity prediction application. The multiple instance classification problem is a generalization of the standard Single Instance binary classification. Formally, the Multiple Instance classification problem can be stated as follows: An input x_i to be classified is a *bag* of instances $x_i = \{x_{i1}...x_{iK_i}\}$ (where K_i is the number of instances in the bag x_i). A bag is labeled as positive ($y_i = 1$) if it contains at least one positive instance and it is labeled as negative ($y_i = -1$) otherwise. Individual instances in the bags are not labeled (however if the bag is labeled as negative we know that each instance in it has a negative label, therefore negative instances are provided “for free”). A training data set $\mathcal{D}$ is a collection of examples of the form (x_i, y_i). Given a dataset $\mathcal{D}$ of labeled bags of instances, construct a predictive model $h(x)$ that can, given a bag of instances x, assign to it a label $y \in (-1, 1)$.

2.1.4 Multiple Instance Multiple Label Learning

Multiple Instance Multiple Label classification problem (Zhang and Zhou, 2006) is a natural generalization of the Multiple Instance classification problem. Each bag x_i of instances is labeled with a subset of labels drawn from a set of possible labels $\mathcal{L} = \{l_1 ... l_M\}$. Thus, the label y_i of x_i can be represented by an M-dimensional vector $y_i^1, y_i^2 .... y_i^M$ where $\forall j \in \{1, ..., M\}$, we have $y_i^j = 1$ if the set of labels assigned to the bag x_i contains the label l_j and is -1 otherwise.

2.2 Generative and Discriminative Models

Let x and y be some random variables that describe input x and output y. *Generative models* specify a process that generates the data by modeling a join probability distribution $p(x, y)$. The model parameters θ are usually learned by maximizing the probability of data given the model parameters $p(x, y|\theta)$ (maximum likelihood estimation) or by maximizing $p(\theta|x, y)$ the posterior distribution of the parameters after observing the data (maximum a posteriori estimation).

Discriminative models, on the other hand, attempt to solve the problem of predicting the output y given the input x directly by maximizing the probability $p(y|x)$. For the majority of known generative models there is a known discriminative counterpart (such as Naive Bayes and its discriminative counterpart Logistic Regression, or Hidden Markov Model (Rabiner, 1989) and its discriminative counterpart Conditional Random Field (Lafferty et al., 2001)).

In general, given any generative model for $p(x, y)$ a discriminative model can be learned by finding the parameters to maximize the probability $p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\sum_{y \in \mathcal{Y}} p(x,y)}$ (typically using numerical optimization procedure), however some discriminative models solve the classification problem directly without using the generative model as an intermediate step (i.e Logistic Regression or Support Vector Machine).

The question of whether to use generative or discriminative models has been a subject to many debates. In particular, Vapnik (1995) wrote that "one should solve the problem directly and never solve a more general problem as an intermediate step". This statement favors the use of discriminative models when the goal is a model with accurate prediction. Generally, if the model structure encodes the task, and if there is a good prior knowledge about the task, generative model and modeling the probability $p(x, y)$ seems like a natural thing in learning. However, the model structure is often not known and the model oversimplifies the task. Ng and Jordan (2001) considered Logistic Regression and Naive Bayes (a discriminative/generative model pair) and provided theoretical and empirical analysis that showed that if there is a sufficient amount of labeled data, discriminative model (Logistic Regression) has a better performance than generative model (Naive Bayes), however generative model has better performance over the discriminative when there is not enough labeled data. A similar result was observed and concluded by Klein and Manning (2002) when generative and discriminative models were applied for text classification and sentence parsing.

2.3 Probabilistic Graphical Models

We now briefly review the "language" used in machine learning literature to describe probabilistic models and how they model relationships between random variables. See Pearl (1988); Koller and Friedman (2009) for details. In this thesis we use directed models. It is a standard practice to use a circle to denote a random variable and plate notation to denote replication of the events. The circle is shaded if this random variable is observed in data, and is white if it is not observed but is assumed by the model. Let $x_1 ... x_n$ be some random variables. The estimation of probability distribution $p(x_1 ... x_n)$ is exponential in the number of variables unless some independence assumptions for the variables are made by the model. Graphical models help describing these independence assumptions Arrows are used to denote the independence assumptions in the model and so a variable x_i is independent from the variables in the graph if the value of parents $\pi(x_i)$ of x_i are known. Then given a graph G which describes the independence assumptions among the random variables, the probability distribution $p(x_1 ... x_n)$ can be written as $p(x_1 ... x_n) = \prod_{i=1}^{n} p(x_i | \pi(x_i))$ and the join probability distribution is exponential in the number of parents (efficiently computable if the number of the parents is small). An example is given in Figure 2.2 for a simple mixture model for documents $X_1 ... X_N$. The probability distribution of all words in the documents is written as:

$$p(z_1, ...z_N, X_1, ...X_n) = \prod_{i=1}^{N} p(z_i)p(X_i|z_i) = \prod_{i=1}^{N} p(z_i) \left(\prod_{j=1}^{M_i} p(x_{ij}|z_i) \right)$$

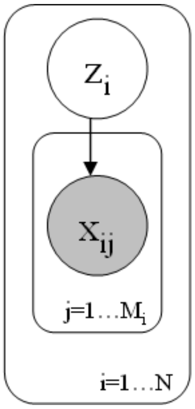


Figure 2.2 Graphical Model Representation of Mixture Model

2.4 Maximum Likelihood and Maximum A-Posteriori Parameter Estimation (and Optimization)

If the data is fully observable, one can use Maximum Likelihood parameter estimation (Myung, 2003). Maximum Likelihood estimation assumes that the data is drawn from some distribution (such as Multinomial for variables that take discrete values or Gaussian for continuous), and the model parameters are chosen as to maximize the likelihood of the data, and so $\theta^* = \arg\max_{\theta} p(D|\theta)$.

One may also play some prior belief on the distribution of the model parameters $p(\theta)$. Then instead of maximizing likelihood of the data, one may maximize the likelihood of parameters given the data $\theta^* = \arg\max_{\theta} p(\theta|D) = \arg\max_{\theta} \frac{p(D|\theta)p(\theta)}{p(D)} = \arg\max_{\theta} p(D|\theta)p(\theta)$. This is known as Maximum A-Posteriori principle (DeGroot, 1970) .

Given the functional form for $p(D|\theta)$ or $p(D, \theta)$ one can find optimal model parameters by solving the optimization problem: $\theta^* = \arg\max_{\theta} p(D|\theta)$ (or $\theta^* = \arg\max_{\theta} p(D, \theta)$). This problem is solved by taking a gradient of a function with respect to model parameters $\frac{\partial p}{\partial \theta}$ and setting is to 0 and then solving the equation $\frac{\partial p}{\partial \theta} = 0$ for θ. If the closed form solution does not exist, the problem can be solved using numeric optimization (Nocedal and Wright, 2000). The simplest case is gradient ascent. It is an iterative procedure that begins with some initialized parameters θ_0, at step k given parameters θ_k the parameters are updated using $\theta_{k+1} = \theta_k + \eta d$ where $\eta > 0$ is the step size, d is a direction and θ_{k+1} are used in the next iteration. This process is repeated until convergence (either $\|d\|_2$ becomes close to 0, or the value of the function does not change or changes very little). If the step-size η is fixed and direction is the gradient of the function $\frac{\partial p}{\partial \theta_k}$ this procedure is equivalent to gradient ascent, however it is usually slow (many iterations are required for convergence). It can be sped up by finding the appropriate step size and by finding a better direction than $\frac{\partial p}{\partial \theta_k}$. Newton method (Nocedal and Wright, 2000), in particular, uses the inverse of the Hessian of gradient to compute the direction $d = [H_p(\theta_k)]^{-1} \frac{\partial p}{\partial \theta_k}$, however because it may be infeasible to compute the Hessian and find its inverse other approximations are typically used (such as Limited Memory BFGS (Liu and Nocedal, 1987) which approximates the direction based on a number of previous directions).

2.5 Bayesian Parameter Estimation

Approximate inference Often a probabilistic model assumes some variables that cannot be observed in data (latent variables). For example, a mixture model assumes that the data is generated according to K generative models (according to some distributions), and the index of the model which generates the data is treated as a latent variable. We give a review of general frameworks for parameter estimation and inference for models with latent variables (Bishop, 2006). In particular we concentrate on Expectation Maximization, and Variational Inference, for more general cases.

First, let θ denote model parameters and z denote hidden variables. It is straightforward to show that

$$p(x|\theta) = \mathcal{L}(q, \theta) + KL\left(q||p\right)$$

where

$$\mathcal{L}(q,\theta) = \int q(z) \ln \frac{p(x,z|\theta)}{q(z)} dz$$
$$KL\left(q||p\right) = -\int q(z) \ln \frac{p(z|x,\theta)}{q(z)} dz$$

and q is any probability distribution function. Since KL divergence is non-negative, it can be seen that $p(x|\theta) \geq \mathcal{L}$ and so $\mathcal{L}$ is the lower-bound on the distribution p.

Expectation maximization Expectation maximization (Dempster et al., 1977) is a procedure which finds the distribution q over the hidden variables z as well as finds the parameters θ that minimizes the lower bound on the distribution p. Expectation maximization is an iterative two-step procedure: in the expectation step (E-Step) the bound $\mathcal{L}$ is maximized with respect to $q(z)$. This happens when $q(z) = p(z|x,\theta^{old})$. In the maximization step (M-Step) $q(z)$ is held fixed, and $\mathcal{L}$ is maximized with respect to parameters θ. The lower-bound $\mathcal{L}$ for convenience is re-written using $p(z|x)$ instead of q and so

$$\mathcal{L}(q,\theta) = \int p(z|x,\theta^{old}) \ln p(x,z|\theta) dz - \int p(z|x,\theta^{old}) \ln p(z|x,\theta^{old}) dx$$

in particular the second term is a constant with respect to θ and $\int p(z|x,\theta^{old}) \ln p(x,z|\theta) = E_{p(z|x,\theta^{old})} p(x,z|\theta)$ is the expectation of $p(x,z,\theta)$ under distribution over hidden variables $p(z)$. In summary, the EM algorithm is an iterative algorithm that involves two steps:

$$\text{E-step: Compute } p(z|x,\theta)$$
$$\text{M-step: } \theta^{new} = \arg\max_{\theta} E_{p(z|x,\theta^{old})} p(x,z|\theta)$$

One drawback is that EM requires either that $p(z|x,\theta)$ is explicitly known, or that the expectation $E_{p(z|x,\theta^{old})} p(x,z|\theta)$ can be computed. This requirement can be bypassed by variational inference where $q(z)$ can be assumed to be some tractable distribution.

Variational Inference Details on variational inference can be found in Blei and Jordan (2004); Bishop (2006). Here we summarize the main results that will be used in the thesis.

Let $Z = \{\theta, z\}$ denote model parameters and hidden variables. The lower-bound on the distribution p is still given as $\mathcal{L}(x) = -\int q(Z) \ln \frac{p(x,Z)}{q(Z)} dZ$. Let $q(Z)$ be a factorized distribution, such that $q(Z) = \prod_{i=1}^{n} q_i$ where we use $q_i = q_i(Z_i)$ is distribution of factor Z_i. Then the lower bound can be re-written as:

$$
\begin{aligned}
\mathcal{L}(q) &= \int \prod_i q_i \left(\ln p(x,Z) - \sum_i \ln q_i \right) dZ \\
&= \int q_j \left(\int \ln p(x,Z) \prod_{i \neq j} q_i dZ_i \right) dZ_j - \int q_j \ln q_j dZ_j + \text{const} \\
&= \int q_j \ln \tilde{p}(x, Z_j) dZ_j - \int q_j \ln q_j dZ_j + \text{const}
\end{aligned}
$$

where $\tilde{p}$ is a new distribution defined as the expectation over all factors that are not j: $\tilde{p}(x, Z_j) = E_{i \neq j} \ln p(x,Z) = \int \ln(x,Z) \prod_{i \neq j} q_i dZ_i$. Now we can keep all factors $q_{i \neq j}$ fixed and maximize $\mathcal{L}$ with respect to i. It is easy to see that $\int q_j \ln \tilde{p}(x, Z_j) dZ_j - \int q_j \ln q_j dZ_j + \text{const}$ is a negative of Kulback-Liebler divergence between q_j and $\tilde{p}(x, Z_j)$ and it is minimized when $q_j = \tilde{p}(x, Z_j)$. Therefore, the optimal solution is given by

$$
q_j^*(Z_j) = \exp\left(E_{i \neq j} \ln p(x, Z)\right)
$$

Then the functional $\mathcal{L}$ is maximized by iteratively maximizing each factor at a time using the solution $q_j^*(Z_j) = \exp\left(E_{i \neq j} \ln p(x, Z)\right)$ until convergence.

2.6 Non-parametric Bayesian methods and processes

As mentioned before, the goal of Bayesian inference is to find parameters θ that best describe some data D. Maximum a posteriori principle suggests using Bayes rule and finding the parameters that are best described by the data: $p(\theta|D) = \frac{p(\theta,D)}{P(D)} = \frac{p(D|\theta)p(\theta)}{P(D)}$. Here $p(\theta)$ is the prior, and it is some distribution on the model parameters, assumed in advance based on the prior belief of the task or data. However, it a difficult task to find a good prior distribution. This gave a rise to non-parametric Bayesian methods. The name "non-parametric" is somewhat misleading, as it does not imply that the model has "no parameters". Rather, it assumes the possibility of infinitely-many parameters, and the parameters are not drawn from a known distribution $p(\theta)$, but rather are generated by a *stochastic process* $G(\theta)$.

In this thesis we use Dirichlet Process (Antoniak, 1974; Blei and Jordan, 2004), which is a generalization of a Dirichlet distribution. Dirichlet Process is a distribution on probability measures. It has two parameters: a base distribution G_0 which can be thought of as the mean of the Dirichlet Process, and a strength parameter α which can be thought of as an inverse variance of the Dirichlet Process. It is denoted as $G \sim DP(\alpha, G_0)$. DP has a property such that for all natural numbers k given k-partitions $\{P_1...P_k\}$:

$$(G(P_1), ...G(P_k)) \sim \text{Dirichlet } \ (\alpha G_o(P_1), ..., \alpha G_0(P_k))$$

There are several ways of constructing a stochastic process that has a property of Dirichlet Process such as Chinese Restaurant Process (Aldous, 1983; Ishwaran and James, 2003), or Stick-breaking construction (Sethuraman, 1994). In this thesis we concentrate on the latter and the application of the Dirichlet Process to a probabilistic model for text and images will be described in details in Chapter 3.

CHAPTER 3. Generative Correlation Model: Multi-modal Hierarchical Dirichlet Process

In this Chapter we introduce a generative model for learning from text and images. We begin by re-visiting a known model, Mixture-of-Multinomials Latent Dirichlet Allocation (MoM-LDA) (Barnard et al., 2003; Blei and Jordan, 2003) which uses latent variables to learn hidden semantic concepts of image features and words. One limitation with this model is that the number of the latent variables needs to be specified a priori. Therefore, we extend this model to a Mixture-of-Multinomials Hierarchical Dirichlet Process (MoM-HDP) to allow countably infinite number of mixture components. Our experimental results on two large image datasets show that unlike MoM-LDA the performance of MoM-HDP does not depend on the number of mixture components.

3.1 Introduction

Learning the relationships between image regions and words is a challenging problem in computer vision. Much of the available training data do not provide explicit labels for individual objects in an image. As more and more data becomes available, human annotation and labeling becomes prohibitively time consuming and expensive. This is especially true in the case of data that is derived from more than one modality (e.g., text and images; sound and images). More importantly, straightforward reductions of multi-modal data mining problems to standard supervised classification problems often fail to fully exploit the natural correlations that might exist among the basic entities within each modality and across modalities.

Given the expense of obtaining training datasets of images where each object in the image is labeled by a human annotator, there is the need for methods that can, given a dataset of images and their associated captions, learn to label individual objects in an image. Against this background, this paper focuses on the following problem: Given a dataset of images and their associated captions, can we build a model that not only predicts a collection of labels for an entire image (the image *annotation* problem), but specifically labels the individual objects (or regions of interest) in the image (the image object-label *correspondence* problem)? Consequently, there is a growing interest in developing principled solutions to the image annotation problem and the image object-label correspondence problems (Barnard et al., 2003; Blei and Jordan, 2003; Blei et al., 2003; Li and Fei-Fei, 2007) (see Chapter 1 for detailed review).

We describe an approach to solving the image annotation and image correspondence problems using a *Mixture of Multinomials hierarchical Dirichlet Process* (MoM-HDP) model which is a natural generalization of the Mixture of Multinomials latent Dirichlet Allocation model (MoM-LDA) (Barnard et al., 2003; Blei and Jordan, 2003). Latent Dirichlet Allocation (LDA) is a generative probabilistic model for independent collections of data where each collection is modeled by a randomly generated mixture over latent factors. In topic modeling for text documents LDA assumes the following generative process: each document has its own distribution of topics, and given a specific topic, the words are generated. MoM-LDA is a generalization of LDA where the documents contain multiple types (modalities) of entities such as words, image regions (also called blobs). MoM-LDA describes the following generative process for the data: each document (consisting of both words and pictures) has a distribution for a fixed number of mixture components (topics), and given a specific mixture component the words and the image features are generated. However, selecting the number of mixture components to be used in a MoM-LDA model is difficult. In practice, several different MoM-LDA models corresponding to different choices of the number of mixture components are trained and evaluated using cross-validation and the best performing model is chosen.

The proposed MoM-HDP model is based on the Hierarchical Dirichlet Process (Teh et al., 2006), a *stochastic process* that can be thought of as the analog of a mixture model, but with a *countably infinite* number of mixture components assumed in a mixture model. MoM-HDP thus allows us to circumvent the need for a priori (and hence potentially arbitrary) choice of the number of mixture components or the computational expense of training multiple MoM-LDA models before choosing one based on the results of cross-validation.

We compare the performance of the proposed MoM-HDP model with that of MoM-LDA model on the image annotation and image-label correspondence task on a dataset with variety of labels and objects using two datasets which provide the ground truth needed in order to evaluate the performance of the two approaches: Visual Object Classes (VOC) 2007 challenge data which has 20 possible labels and a subset of LabelMe, which has over 1700 possible labels. We also compare MoM-HDP and MoM-LDA with some simple alternatives: Naive Bayes and Logistic Regression classifiers based on the formulation of the image annotation and image-label correspondence problems as one-against-all classification problems. Our results show that the generalization performance of MoM-HDP is superior to that of MoM-HDP as well as the Naive Bayes and Logistic Regression classifiers. The results of our experiments show that the generalization performance of the MoM-LDA model is sensitive to the choice of the number of components that are assumed to exist in the mixture. In contrast, the performance of the MoM-HDP model is relatively insensitive to the specific choice of the cutoff used to truncate the Dirichlet Process.

Thus, the main contributions of this Chapter are:

- Development of MoM-HDP, a HDP counterpart of MoM-LDA model for solving image annotation and image object-label correspondence problems under fairly general assumptions that circumvents the need for a priori (and hence potentially arbitrary) choice of the number of mixture components or the computational expense of training and evaluating multiple MoM-LDA models before choosing one based on the results of cross-validation.
- Experimental results that demonstrate that modeling the problem directly using MoM-LDA and MoM-HDP produces a better performance than one-against-all-learning scenario and that MoM-HDP outperforms MoM-LDA on image annotation and image-object label correspondence problems.

This Chapter is organized as follows: We describe the MoM-LDA model in Section 3.2. In Section 3.3 we describe non-parametric models that we adapt in our work, in particular Dirichlet

Process, and its generalization as a Hierarchical Dirichlet Process. Then we generalize MoM-LDA model to a Hierarchical Dirichlet Process and describe in detail the algorithm to estimate the model parameters using variational inference in Section . We describe the dataset, experimental setup, evaluation procedure, and the results of our comparison of MoM-LDA and MoM-HDP models in Section 3.5. We conclude the Chapter with related work in Section 3.6 and a summary and a brief discussion of some directions for further research in Section 3.7.

3.2 Mixture of Multinomials Latent Dirichlet Allocation

We first describe a Mixture of Multinomials Latent Dirichlet Allocation model (MoM-LDA) introduced by Blei and Jordan (2003), and then generalize this model using a Hierarchical Dirichlet Process (MoM-HDP). Informally, the following generative process is assumed for images and captions. The image topic (e.g. horseback riding) generates a distribution for intermediate level components (e.g. horse, person, grass, fence, sky, sun, building) and the intermediate level components generate specific words and image regions observed in the training data (e.g. the words “horse” and “person”, and the image regions which correspond to horse’s eyes, ears, person’s face, arms and legs, etc). MoM-LDA assumes a pre-defined number of clusters which group the related entities in the modalities, and it groups the related visual words and the related words in the same clusters. In addition, the probability distribution of the clusters is different for each image-caption pair, which is achieved by introducing a Dirichlet prior for the distribution of clusters.

Let x_i be the feature vector for image i and y_i be the feature vector for the caption associated with the image i. Formally, the images-caption pair (x_i, y_i) are generated by the following generative process:

1. For each image i, pick a distribution of latent topics $\pi_i \sim \text{Dirichlet}(\alpha)$.

   (a) For each caption word y_i^j

      i. pick a latent factor $t_{ij} \sim \text{Multinomial}(\pi_i)$

      ii. pick the word $y_i^j \sim F(t_{ij})$.

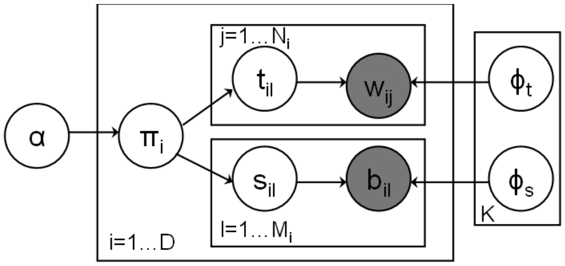


Figure 3.1 Mixture of Multinomials Latent Dirichlet Allocation

(b) For each image feature x_i^j

i. pick a latent factor $s_{ij} \sim \text{Multinomial}(\pi_i)$

ii. pick the feature $x_i^j \sim F(s_{ij})$.

The graphical model for this process is shown in Figure 3.1. Here $F(x)$ can be any appropriate distribution, such as Multinomial for words and discrete features, or Gaussian for continuous features. In our model and in our experiments, we use discrete-valued image features (visual words). Hence, we focus our discussion on the MoM-HDP model based on the Multinomial distribution. However, the model described in this paper can be easily extended to other distributions.

3.3 Mixture of Multinomials Hierarchical Dirichlet Process model

A limitation of mixture models is the need to specify a number of components (namely K). The choice of number of the mixture components can have a major influence on how well the model fits the data, and its ability to generalize beyond the training data. Hence, we consider a model based on a hierarchical Dirichlet Process (HDP) (Teh et al., 2006), with *countably infinite* number of mixture components. Further information and details on the HDP and their applications in probabilistic graphical models can be found in Teh et al. (2006) or Blei and Jordan (2004) and here we provide a background on Dirichlet Processes needed for this Chapter.

3.3.1 Dirichlet Process and Dirichlet Process Mixture Model

Dirichlet Process is a distribution on probability measures. It has two parameters: a base distribution G_0 which can be thought of as the mean of the Dirichlet Process, and a strength parameter α which can be thought of like an inverse variance of the Dirichlet Process, and is denoted as $G \sim DP(\alpha, G_0)$. DP has a property such that for all natural numbers k given k-partitions $\{P_1 ... P_k\}$:

$$(G(P_1), ...G(P_k)) \sim \text{Dirichlet } \ (\alpha G_o(P_1), ..., \alpha G_0(P_k))$$

3.3.1.1 Stick-Breaking Construction of Dirichlet Process

One way of constructing a process that has this property is via stick-breaking construction (Sethuraman, 1994). Define $G(\theta) = \sum_{i=1}^{\infty} \beta_i \delta_{\theta_k}(\theta)$ where δ is the Dirach delta function and $\{\theta_k\}_{k=1}^{\infty}$ are drawn from of G_0. Define $\beta_k = u_k \prod_{i=1}^{k-1}(1 - u_i)$ where u_k is distributed according to Beta distribution with the base parameter α, $u_k = Beta(1, \alpha)$. Such construction is denoted by $\beta \sim GEM(\alpha)$. Sethuraman (1994) proved that this constructive process has the property of Dirichlet Process.

Intuitively, β's are constructed as follows: we start off with a stick of unit length. At Step 1 we break of a piece of a stick according to the proportion u_1, and we have $\beta_1 = u_1$ and $1 - u_1$ is left. At Step 2 we break off another piece of the remaining stick according the proportion u_2 and so $\beta_2 = u_2(1 - u_1)$ and $(1 - u_1) - u_2(1 - u_2)$ is what's left of a stick. Then in general, at step k the broken off piece equals to $\beta_k = u_k \prod_{i=1}^{k-1}(1 - u_i)$ and what's left of the stick is $\prod_{i=1}^{k}(1 - u_i)$.

There are other ways of constructing a measure which has the properties of a DP such as Chinese Restaurant Process (Ishwaran and James, 2003).

3.3.1.2 Dirichlet Process Mixture Model

One application of the Dirichlet Process is the Dirichlet Process Mixture Model (Antoniak, 1974). Such mixture model assumes that instead of a fixed number of mixture components there are countably infinitely many mixture components, and the prior probability for the distribution of mixture components is defined according to the Dirichlet Process.

Dirichlet Process mixture model is the following generative process:

Let $z = \{z_1, z_2 ...\}$ be the mixture components, and let $X = \{x_1 ... x_N\}$ be a sample from the DP mixture. Then we can assume the following generative process for the data:

1. Draw mixture priors $\beta \sim DP(\alpha, G_0)$
2. For each mixture component $z = \{z_1, z_2 ...\}$ draw parameters $\phi_z \sim G_0$ which specify the distribution for the observations X
3. For each instance $i = 1...N$
   (a) Draw parameters $\pi_i \sim \beta$ which specify the distribution of the mixture components
   (b) Draw a mixture component $z_i \sim \text{Multinomial}(\pi_i)$
   (c) From the mixture component z_i draw $x_i \sim \phi_{z_i}$.

If we assume K mixture components the distribution of which has Dirichlet distribution, we get a standard Mixture Model.

3.3.2 Hierarchical Dirichlet Process

A Hierarchical Dirichlet Process (Teh et al., 2006) assumes a number of Dirichlet Processed $G_1 ... G_J$ such that the base distribution G_0 for each of the G_j is shared and it is also a Dirichlet Process: $G_j \sim DP(\alpha_0, G_0), j = 1, ...J$ and $G_0 \sim DP(\alpha, H)$.

This hierarchical construction assumes that the data generated according to this process shares the base distribution. When this construction is applied to a mixture model (Hierarchical Process Mixture Model), this process allows the data to share cluster identities among groups of data. Each image or each document can be viewed as a “group”, and by sharing the cluster identities, the latent topics are shared among each document or each image. A similar assumption used in Latent Dirichlet Allocation Model, however unlike Latent Dirichlet Allocation, the number of mixture components is countably infinite, and thus HDP mixture model provides a non-parametric generalization of Latent Dirichlet Allocation. The HDP mixture model is described using the following generative process:

1. Draw mixture priors $\beta \sim DP(\alpha_0, G_0)$
2. For each mixture component $z = \{z_1, z_2 ...\}$ draw parameters $\phi_z \sim G_0$ which specify the distribution for the observations X
3. For each instance $i = 1...N$
    (a) Draw parameters $\pi_i \sim DP(\alpha, \beta)$ which specify the distribution of the mixture components
    (b) Draw a mixture component $z_i \sim \text{Multinomial}(\pi_i)$
    (c) From the mixture component z_i draw $x_i \sim \phi_{z_i}$.

3.4 Mixture of Multinomials Hierarchical Dirichlet Process (MoM-HDP)

We now apply the Hierarchical Dirichlet Process to the Mixture of Multinomials generative model. Like in the case of MoM-LDA, we assume that each observable modality is clustered by the mixture components, so that each word y is generated by a cluster t, each image feature x is generated by a cluster s. The clusters for image-caption pair x_i, y_i have Multinomial distribution parametrized by π_i ($p(s_i) = p(t_i) = \pi_i$) drawn from $DP(\alpha^\pi, \beta)$ were $\beta \sim GEM(\alpha)$ is constructed using a stick-breaking distribution. Furthermore, the parameters for observations given their clusters $\phi_t^y = p(y|t)$ and $\phi_s^x = p(x|s)$ are generated from some base distribution G_0 (such as a Dirichlet distribution).

We show MoM-HDP using plate notation in Figure 3.2. We also note that if the prior β is assumed to be drawn from finite Dirichlet instead of a stick-breaking distribution, this model becomes a Dirichlet-smoothed version of the MoM-LDA (Blei and Jordan, 2003).

We summarize the generative processes modeled by MoM-HDP and MoM-LDA below.

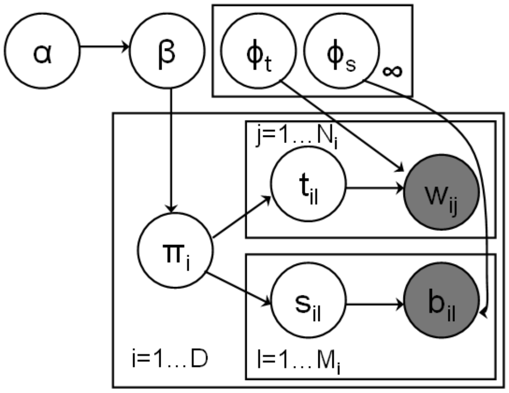


Figure 3.2 Mixture of Multinomials Hierarchical Dirichlet Process

MoM-HDP	MoM-LDA
draw $\beta \sim \text{GEM}(\alpha)$	chose priors $(\alpha_1 ... \alpha_K)$
for each $z = 1, 2...$	for each $z = 1, ..., K$
draw $\phi_z^y \sim \text{Dirichlet}(\alpha_y)$	draw $\phi_z^y \sim G_0$
draw $\phi_z^x \sim \text{Dirichlet}(\alpha_x)$	draw $\phi_z^x \sim G_0$
for each image $i = 1, ..., D$	for each image $i = 1, ..., D$
draw $\pi_i \sim \mathbf{DP}(\alpha^\pi, \beta)$	draw $\pi_i \sim \mathbf{Dirichlet}(\alpha_1 ... \alpha_K)$
for each word $j = 1 ... N_i$	for each word $j = 1 ... N_i$
draw $t_{ij} \sim \text{Multinomial}(\pi_i)$	draw $t_{ij} \sim \text{Multinomial}(\pi_i)$
draw $y_i^j \sim \text{Multinomial}(\phi_{t_{ij}}^y)$	draw $y_i^j \sim \text{Multinomial}(\phi_{t_{ij}}^y)$
for each visual word $j = 1, ..., M_i$	for each visual word $j = 1, ..., M_i$
draw $s_{ij} \sim \text{Multinomial}(\pi_i)$	draw $s_{ij} \sim \text{Multinomial}(\pi_i)$
draw $x_i^j \sim \text{Multinomial}(\phi_{s_{ij}}^x)$	draw $x_i^j \sim \text{Multinomial}(\phi_{s_{ij}}^x)$

To make the parameter estimation feasible, we assume a truncated Dirichlet Process (Ishwaran and James, 2001), and truncate β at K, so that $\beta_z = 0$ for all $z > K$. In this case, $\pi_i \sim \text{DP}(\alpha^\pi, \beta)$ simply becomes $\pi_i \sim \text{Dirichlet}(\alpha^\pi, \beta_1 ... \beta_K)$, since any K partitions generated by the DP are distributed according to Dirichlet Distribution (by the definition of DP). Ishwaran and James (2001) proved that the truncated DP approximates true DP and the approximation improves as the cut-off K increases.

Next we describe the parameter estimation procedure for the hierarchical Dirichlet Process model using variational inference.

3.4.1 Parameter Estimation via Variational Inference

We use variational inference to find the model parameters. Define the factorized distribution $\mathcal{Q}$ as:

$$\begin{aligned}\mathcal{Q} =& q(\beta, \pi, s, t, \phi_z^y, \phi_z^x) \\ =& q(\beta)q(\pi)\prod_{i=1}^{M} q(s)\prod_{i=1}^{N} q(t)\prod_{z=1}^{K}(q(\phi_z^x)q(\phi_z^y))\end{aligned}$$

where $q(\beta) \sim \text{GEM}(\alpha)$ is drawn from the stick-breaking distribution, $q(\pi) \sim \text{DP}(\alpha_\pi, \beta)$ is drawn from the truncated Dirichlet Process (thus results in Dirichlet Distribution), $q(\phi_z)$'s are drawn from the Dirichlet distributions, and $q(s)$, $q(t)$ are Multinomial.

The parameter estimation using variational inference for the Hierarchical Dirichlet Process can be viewed as a three-step process: the expectation step involves optimizing hidden Multinomial factors $q(s)$ and $q(t)$ (equivalent E-step in the EM). The maximization step involves parameter estimation to optimize $q(\phi)$ and $q(\pi)$ (equivalent to the M-step in the EM). The last step is optimizing the top-level distribution $q(\beta)$ (this step has no counterpart in the standard EM).

3.4.1.1 Updating Dirichlet Distribution Factors $q(\pi)$, $q(\phi_z^w)$, $q(\phi_z^b)$ (M-step)

Since we have truncated β at a finite K, the Dirichlet Process reduces to a finite Dirichlet distribution. Using mean-field $q(\pi) \propto \mathbb{E}_q \log(p(\pi|t, s)) \propto \mathbb{E}_q \log(p(t, s, \pi))$. The optimal $q(\pi)$ parametrized by γ is given by standard update for a Dirichlet distribution. Computing the expectation we get the following expression:

$$\begin{aligned}
q(\pi) =& \exp \mathbb{E}_q \log[\prod_{z \in Z} \pi_z^{\alpha_\pi \beta} \prod_{z \in Z} \pi_z^{\sum\limits_{i=1}^{M} \mathbb{1}_{(s_i,z)}} \prod_{z \in Z} \pi_z^{\sum\limits_{i=1}^{N} \mathbb{1}_{(ti,z)}}] \\
=& \exp \mathbb{E}_q (\alpha_\pi \beta + \sum_{i=1}^{M} \mathbb{1}_{(s_i,z)} + \sum_{i=1}^{N} \mathbb{1}_{(t_i,z)}) \sum_{z \in Z} \log \pi_z \\
=& \exp \sum_{z \in Z} \log \pi_z^{\mathbb{E}_q \alpha_\pi \beta + \mathbb{E}_q \sum_{i=1}^{M} \mathbb{1}_{(s_i,z)} + \mathbb{E}_q \sum_{i=1}^{N} \mathbb{1}_{(ti,z)}} \\
=& \prod_{z \in Z} \pi_z^{\alpha_\pi \beta + C_s(z) + C_t(z)} \\
=& \text{Dirichlet}\left(\alpha_\pi \beta + C_s(\cdot) + C_t(\cdot)\right)
\end{aligned}$$

Therefore the solution to factor π of the form $\gamma = \alpha_\pi \beta + C_t(\cdot) + C_s(\cdot)$ as the update for the Dirichlet parameters, where $C_t(\cdot) = C_t(t_1...t_k)$ is a vector of expected counts of the values that the factor t can take. Similarly $C_s(\cdot) = C_s(s_1...s_k)$ is the vector of expected counts that the factor s can take. These expected counts are computed using $q(s)$ and $q(t)$ that we describe below (E-step).

The updates for the $q(\phi)$ are obtained similarly, and are $q(\phi_z^y | \lambda_z^y) = \text{Dirichlet}(\alpha_y + C^y(z, \cdot))$ where $\lambda_z^y = \alpha_y + C^y(z, \cdot)$ and $q(\phi_z^x | \lambda_z^x) = \text{Dirichlet}(\alpha_x + C^x(z, \cdot))$ where $\lambda_z^x = \alpha_x + C^x(z, \cdot)$. Here $C^y(z, \cdot) = C(z, y_1...y_M)$ is the vector of expected counts of words of the image in cluster z and $C^x(z, \cdot) = C(z, x_1...x_B)$ is the vector of expected counts for visual words in cluster z that describe the image.

3.4.1.2 Updating Multinomial Distribution Factors $q(t)$, $q(s)$ (E-step)

In order to introduce dependency of the data, we first define $q(t_j | y_i) \propto q(t_j, y_i)$ and $q(t_j)$ can be recovered by marginalizing over the words w. Using mean-field approximation,

$$\begin{aligned}
q(t_j | w_i) =& \exp\left(\mathbb{E}_q \log(p(t_j | y_i))\right) \\
\propto& \exp\left(\mathbb{E}_q \log(p(t_j, w_y))\right) \\
\propto& \exp\left(\mathbb{E}_q \log \pi(j)\right) \exp\left(\mathbb{E}_q \log \phi_{t_j}^y (y_i)\right)
\end{aligned}$$

Define Multinomial weights as $W(t_j) = \exp\left(\mathbb{E}_q \log \pi(j)\right)$ and $W_{t_j}(y_i) = \exp\left(\mathbb{E}_q \log \phi^y_{t_j}(y_i)\right)$. The weights W can be computed efficiently, namely $W_{t_j}(w_i) = \frac{\exp(\Psi(\lambda_t(y_i))}{\exp\Psi(\sum_i \lambda_t(y_i)}$ and $W_t(t_j) = \frac{\exp(\Psi(\gamma_j)}{\exp\Psi(\sum_i \gamma_i)}$ where $\Psi(x) = \frac{\partial}{\partial x}\log\Gamma(x)$ is the Digamma function (which can be computed using Taylor-series approximation). The Dirichlet priors λ and γ are used after updating the Dirichlet distribution factors (which was described in the previous step). The expectation of a Dirichlet distribution is described in Appendix.

3.4.1.3 Updating Top-Level Component $q(\beta)$

Finally we summarize the updates for the stick-breaking parameters β. Again, using mean-field it is easy to show that $q(\beta) \propto \mathbb{E}_q p(\beta|\alpha) + \mathbb{E}_q p(\pi|\beta)$, and so $q(\beta) = \mathbb{E}_q\text{logGEM}(\beta;\alpha) + \mathbb{E}_q\text{logDP}(\alpha_\pi, \beta)$, however since we truncated β at K, it becomes $q(\beta) = \mathbb{E}_q \log\text{GEM}(\beta;\alpha) + \mathbb{E}_q \log\text{Dirichlet}(\alpha_\pi, \beta)$. There are no closed-form solutions for β, however it is possible to maximize $q(\beta)$ using gradient ascent and update the components of β with $\eta\frac{\partial q(\beta)}{\partial \beta_k}$ iteratively (where η is the learning rate). The update equations are derived similarly to those in Liang et al. (2007). In order to satisfy the constraint $\sum_{i=1}^{K}\beta_i = 1$ we use Quadratic Penalty method (Nocedal and Wright, 2000). The details of the optimization are described in Appendix.

3.4.2 Making Predictions

Given the model, we can now use it to predict region annotations. To predict the label for a region described by $x = x_1...x_T$, we can use the word which has the highest probability given all the visual words in the region: $p(y|x)$. This probability can be computed as follows:

$$\begin{aligned} p(y|x) &= \sum_{m=1}^{T}\sum_{z_m} p(y|z_m)\int p(z_m|\pi_s)p(\pi_s|x_m)d\pi_s \\ &\approx \sum_{m=1}^{T}\sum_{z_m} p(y|z_m)q(z_m|x_m) \end{aligned}$$

Note that the integral can be computed efficiently using variational inference for the test region.

The label with the highest probability given the region is then assigned to the region. That is, $y_{pred} = \arg\max_{y_i\in L} p(y_i|x)$.

3.5 Experiments and Results

We now describe the datasets used in our evaluation, and the experimental set-up.

3.5.1 Data

In order to evaluate the performance of the model on image object label correspondence, we need to assume that the image to be labeled is segmented into regions or objects and need to have labels for each region or object in each test image. The images can be segmented using one of the magnitude of available segmentation algorithms (such as normalized cuts (Shi and Malik, 2000) or superpixels (Ren and Malik, 2003)). Note that we do not use object-level labels in training the model. A major goal of this work is to explore the feasibility of using models trained on a dataset of images and their associated annotations to perform both image annotation as well as labeling of individual objects in each images. We proceed with describing the image data and feature extraction from the images.

3.5.1.1 PASCAL Visual Objects Classes

We compare both MoM-LDA and MoM-HDP on the image annotation and image-label correspondence tasks on Visual Object Classes 2007 challenge data (Everingham et al., 2007).

The VOC 2007 database contains 2501 training images in 20 categories and 4952 images in the test set. We re-sized the images for the maximum height or width (whichever is greater) of 256 pixels. We use grid sampling to extract patches of 13x13 pixels from each image. We then use SIFT representation of each patch Lowe (2004) to extract 128 features for all images in the training set. These features are invariant to rotation and object occlusions. The 150,000 descriptors (extracted randomly from the training images) were clustered into 1500 clusters using k-means clustering to create a codebook of “visual words”. Each *image* was then represented as a bag of visual words, and a bag of caption words (labels). The codebook created from the training images was used to represent the test objects.

We assume that the test images are segmented and extract the SIFT features from the regions, and use the codebook created at training to represent the *test objects*. If the images

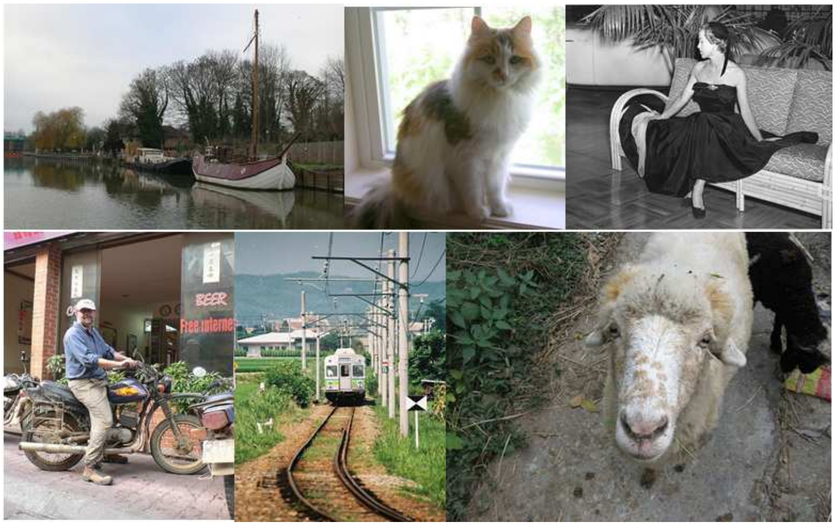


Figure 3.3 Sample from the VOC 2007 training and test images.

were not segmented, we could have used segmentation algorithms (such as normalized cuts (Shi and Malik, 2000) or superpixels (Ren and Malik, 2003)) to segment each image into regions before processing them further. However, the results of such segmentation may or may not coincide with the segmentation that forms the basis of object-level labels used as reference to evaluate the performance of the model on the image object-label correspondence task. Hence we assume here that segmented images are provided during the test phase. There are 14,976 objects in the test set.

We show some representative training and test images in Figure 3.3 to demonstrate the variety of the images and complexity of the task.

3.5.1.2 LabelME

To evaluate the performance of the models on the large scale data set with many possibilities for captions we use LabelMe (Russell et al., 2005) database. LabelMe is a web-based image database and an annotation tool which allows users to annotate images and objects in the images in the database. The annotators select the regions which correspond to the objects in the image, and label these regions with the keywords. The database contains a great variety of image categories and themes, and it continues to grow over-time as more and more people contribute the new images and annotate the existing ones.

For our experiments we selected a subset using 9 keywords to query for images and used the union of these images as the data (“building”, “car”, “tree”, “cat”, “dog”, “person”, “plant”, “water” and “sky” were the keywords). We then selected images which have between 4 and 19 objects. From the resulting subset we used 80% of the images as the training data set (resulting in 7373 images), and the rest as the test set (1513 images). The test set contains ∼14000 regions, and so on average each image has 10 captions. All images were rescaled for the maximum height and width of 256 pixels.

The captions were changed to lower-case and stemmed, resulting in ≈ 1700 distinct caption words in the vocabulary. As before, we extract SIFT features in order to create a codebook of 1500 visual words from the training data from 15,000 image patches randomly sampled from the training images, train the model on the image and caption information only, and test the model on the regions.

3.5.2 Experiments and Results

3.5.2.1 Multiple Label Learning as One-Against-All Classification

To establish a baseline, we first consider a transformation of the multiple label problem to one-against-all learning scenario, similar to the set-up in Zhang and Zhou (2006). Given the dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{D}$, vocabulary of caption words L of size M, we train M binary classifiers. Each classifier h_{y_j} is trained on a new dataset were all the target words were kept and considered as one class, and all the words that are not the target words where considered the second class: $\mathcal{D}^{'} = \{x_i, y_i^{'}\}_{i=1}^{D}$ where x_i is that as in $\mathcal{D}$ and $y_j^{'} = 1$ if y_i of $\mathcal{D}$ contains word $y_i \in L$ and 0 otherwise. Given a test object x_{test} each of the classifiers $h_{y_j}(x_{test})$ assigns a score r_{w_i} and the word with the highest score is used as a prediction: $y_{pred} = \arg\max_{w_i}[r_{w_1}, ..., r_{w_T}]$. We considered Naive Bayes and Logistic Regressions as the classifiers.

3.5.2.2 Initialization for Parameter Estimation

Variational inference is susceptible to local minima. Since one of the local minima corresponds to the setting where all factors are equally likely, we initialize the model by randomly assigning several image/caption pairs to a factor. We set the hyperparameters $\alpha = \{\alpha, \alpha_\pi, \alpha_b, \alpha_w\}$ to 1. Given the large size of the training dataset, we believe that the choice of hyperparameters for priors is not especially critical.

3.5.2.3 Image Annotation and Region Labeling

In order to assess the performance of the models on the image annotation task, we used accuracy of annotation as the performance measure. Let C be the predicted set of words in a caption. Let R be the actual caption (the actual set of words that appear in the caption for a particular image). To avoid the complication of having to deal with multiple objects with the same name, we binarize C and R. To measure how close C is to R we count how many elements are in common in C and R; In other words, we are interested in the cardinality of the intersection $|C \cap R|$. We can now define accuracy as $Acc = P(R|C) = \frac{|C \cap R|}{|C|}$.

Since we have the ground truth or object-level labels for the regions, we can also evaluate the performance of the model on the object recognition task on the per-label basis using standard performance measures such as precision (the fraction of the actual objects with a given label out of all the objects classified as such), recall (the fraction of the objects that were assigned a particular label out of all the existing objects with that label), and accuracy (the fraction of correctly labeled objects in the entire set of test images).

VOC2007 In the VOC 2007 dataset, the number of labels is 20, and so predicting a label at random results in 5% accuracy.

Table 3.1 shows the comparison of one-against-all learning scenario and the combined LDA model.

Notice that Logistic Regression has the worst performance. We believe that this could be due to over-fitting on the training data. Since one Logistic Regression is trained to maximize

	per region	per caption
NB OneVsAll	30.56	38.03
LR OneVsAll	20.19	20.79
MoM-LDA	31.67	40.82
MoM-HDP	**34.5**	**41.92**
Chance prediction	5	5

Table 3.1 Comparison of accuracies (in %) of various algorithms for per-region and per-caption annotation task for VOC 2007 dataset

accuracy for each keyword, since the distribution of the target word and its compliment is very unbalanced, it is possible that Logistic Regression overfits on the compliment of the keyword, thus assigning low scores to the words.

Statistical Significance Test It is a standard practice to use a k-fold cross-validation (Dietterich, 1998). However, the VOC 2007 challenge dataset that we used consists of a pre-specified training set and a test set; with test set being much larger and more "difficult" that the training set (Everingham et al., 2007). Moreover, because the words that appear in a given caption as well as the objects that appear in an image are unlikely to be independent which presents challenges in devising reliable tests for comparing different models - see Section 3.7.

In order to test significance of the results on region labeling we use a simple statistical test for difference in two error proportions Snedecor and Cochran (1989). Let the null hypothesis be that two algorithms f_1 and f_2 have the same error on the same test dataset T of size N. Let $e_1 = \frac{N_{f_1}}{N}$ be the fraction of the test examples that f_1 predicted incorrectly and let $e_2 = \frac{N_{f_2}}{N}$ be the fraction of the test examples that f_2 predicted incorrectly. Then the quantity $e_1 - e_2$ can be viewed as a random variable with 0 mean and standard deviation $s_e = \sqrt{\frac{2p(1-p)}{N}}$ where $p = \frac{e_1+e_2}{2}$ is the average of two errors. From this, we use the statistic $z = \frac{e_1-e_2}{s_e}$ and if $|z| > Z_{0.975} = 1.96$ then the null hypothesis is rejected. We compute the z-value between the MoM-HDP and the other algorithms considered, and the improvement on the test set is statistically significant. The z-values for the difference between errors of various algorithms are: z(MoM-HDP,NB)=7.1, z(MoM-HDP,LR)=27.6, z(MoM-HDP, MoM-LDA)=5.02.

Since we use region labeling to construct the full caption, we believe that the significance tests on the region labeling are enough for the caption reconstruction. We also note that to the best of our knowledge there is no well-defined statistical significance test for a learning algorithm which predicts multiple labels to a test instance, and that it is of interest to develop such test.

We next take a closer look the performance of the combined mixture models MoM-LDA and MoM-HDP on the region annotation and overall image annotation as a function of the number of the mixture components K, in Figure 3.4. The best precision of MoM-LDA in terms labels assigned to objects in the image and in terms of the caption assigned to the image was obtained at $K = 5$. The performance of MoM-HDP is less sensitive to the choice of K used to truncate the HDP model. We also observe that the performance of Mom-LDA degraded when the number of mixture components exceeded the optimum value ($K = 5$ whereas the performance of MoM-HDP was more robust with respect to K.

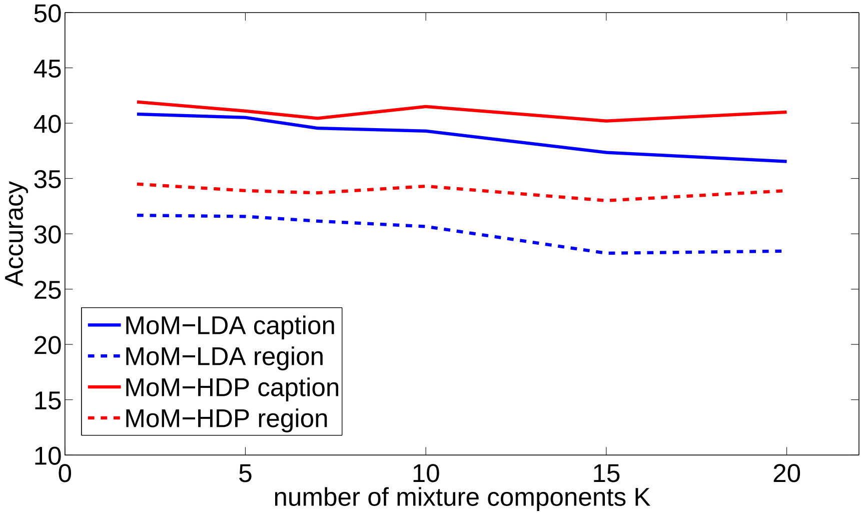


Figure 3.4 Performance of MoM-LDA (represented by the red line) and performance of MoM-HDP (represented by the blue line) vs the number of mixture components. The accuracy on the region labeling is shown using solid line, and the overall accuracy on the captions constructed from the region labels is shown using dashed line.

While an accuracy of 41% may be viewed as poor in the standard supervised learning setting, it is worth noting that the more general multi-modal learning setting considered in

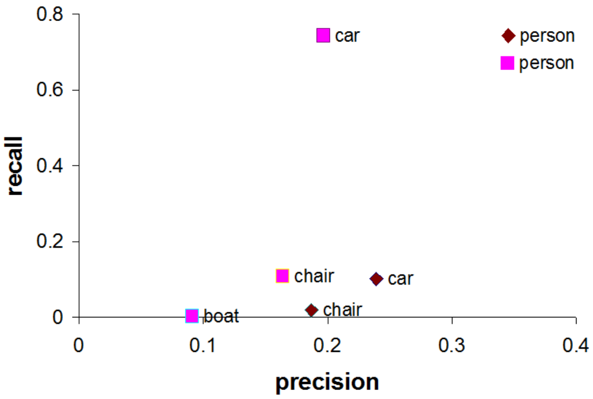


Figure 3.5 Region annotation result: per-label precision recall on all predicted region labels for VOC 2007. Square: HDP precision/recall, diamond: LDA precision/recall.

this paper is far more challenging. In particular, similar results were reported in Barnard et al. (2003) (using precision/recall), however they allowed several labels to be assigned for a given region by using a threshold for each keyword. Threshold was also used in Hardoon et al. (2006) to evaluate performance for image annotation.

We now take a closer look at the performance of the models on the per-region task, and examine in detail the performance measured by precision/recall on the per-label basis. The results are presented in Figure 3.5 for MoM-HDP and LDA for a cut-off $K = 5$.

Only several labels have relatively high precision/recall measures (Barnard et al., 2003) reported similar trends, however the precision/recall was calculated for whether a word was present or absent in the caption, not for the labels directly). While both models have similar performance on the precision measure, the recall is much higher for HDP model. In addition, HDP was able to assign a relatively high precision/recall for the label "boat", while LDA did not predict any boats correctly.

Note that the label "person" has a very high recall and low prevision, which indicates that it was often predicted as a possible label. We discovered that in the training data about half of the captions included the word "person". Consider an image which has many objects of which only a few have corresponding labels in the caption. In such a scenario, the visual words associated with the image (which could be very diverse) are likely to get assigned to the clusters associated with the few labels that appear in the image caption, thereby biasing the predictions towards those labels. We conjecture that the sparsity of captions relative to the number of objects in the image biases the model towards the labels that are overrepresented in captions. One possible approach to correcting this bias is to use partially supervised training data and to add region/caption pairs as additional training examples. Another possible source of improvement is better quality captions, i.e., captions that are descriptive of all objects in the image.

LabelMe Lastly, we show the performance of the model on region labeling on the large data set derived from LabelMe. Here we set K to 20, and train MoM-LDA and MoM-HDP and we report the accuracy on the per-region basis and on the entire caption set in Table 3.2. A random classifier would have 1 in 1700 chance to predict a caption correctly (0.06% accuracy).

LabelME	image annot.	region annot.
MoM-LDA	15.56	10.5
MoM-HDP	34.84	**28.45**
NB OneVsAll	**38.2**	24.21

Table 3.2 Performance (accuracy in %) of MoM-LDA and MoM-HDP on image annotation and region recognition for LabelME

While Naive Bayes one-against-all training scenario produces a better result on the caption prediction than the MoM-HDP model, MoM-HDP model has a much better result on the region labeling, and the improvement is statistically significant (the z-values for the difference between errors of various algorithms are: z(MoM-LDA,MoM-HDP)=37.92, z(MoM-HDP, NB)=8.05).

3.6 Related Work

We briefly summarize work on the image annotation and image object-label correspondence or closely related problems. Learning from multi-modal data, and in particular, learning to annotate images, has been cast as a multiple label multiple instance learning problem (Zhang and Zhou, 2006). In this formulation, each image is represented by a bag of objects (instances), and the corresponding image caption is represented by a bag of words (set of labels). Zhang and Zhou (2006) proposed to use a multiple-instance learning for each label using one-against all multiple-instance learners. However, this work did not address the problem of labeling each individual object within an image.

Hardoon et al. (2006) have explored a kernelized version of Canonical Correlation Analysis for image retrieval and annotation. Specifically they show how a semantic representation of images and their associated text can be learned and how the resulting representation in a common semantic space can be used to compare data from the text and image modalities. However, the primary focus of this work was not on solving the image object-label correspondence problem.

Barnard et al. (2003) have examined several solutions for the image annotation and image-object label correspondence problems. They developed several models for the joint distribution of image regions and words, including those that explicitly learn the correspondence between image regions and words. They studied a multi-modal and correspondence extensions to hierarchical mixture models (Hofmann and Puzicha, 1999), and probabilistic latent semantic indexing (pLSI) (Hofmann, 1999) for text, a translation model adapted from statistical machine translation (Brown et al., 1993), and a multi-modal extension to mixture of Latent Dirichlet Allocation (MoM-LDA) (Blei and Jordan, 2003; Li and Fei-Fei, 2007) which generalizes LDA (Blei et al., 2003) to the setting where the data combines multiple modalities (e.g., image, text).

Selecting the number of mixture components to be used in a MoM-LDA model is a difficult problem. In practice a model is trained for several numbers of mixture components, evaluated on a held-out set, and the best performing model is chosen. The need to train and evaluate several models makes this approach computationally expensive, especially in the case of large datasets consisting of large numbers of image and text features. In contrast, the proposed MoM-HDP model allows to circumvent the need for a priori choice of the number of mixture components. It addressed the computational expense associated with the model selection for MoM-LDA since in practice multiple MoM-LDA models need to be trained before choosing one based on the results of cross-validation.

In contrast to previous work (Barnard et al., 2003) which relied on representation of image segments using *global* features such as shape, color, texture, etc. we have chosen to use *local* features (Bosch et al., 2006). A consequence of reliance on global properties of image segments is that the images must be segmented prior to training the model. In contrast, representation of image segments (blobs) using local image features makes it possible to train the model on images without segmenting them prior to training. Furthermore, recent work in the image processing community has shown that local representation of the image can substantially improve the performance of the resulting models (Bosch et al., 2006). In Barnard et al. (2003), the experiments were performed using the Corel dataset which only provides the captions for the image, and this dataset is also no longer publicly available. In the absence of labels for individual objects or image segments, their study provided a limited assessment on the image object-label correspondence task on a small number of hand-annotated objects. In contrast, in this Chapter, we used two datasets which provide the ground truth needed evaluating the performance of alternative solutions image annotation and image object-label correspondence tasks: Visual Object Classes (VOC) 2007 challenge data which has 20 possible labels and a subset of LabelMe, which has over 1700 possible labels.

3.7 Summary

In this Chapter we considered an interesting problem with many applications in image retrieval and multi-media data-mining: Given a dataset of images and their associated captions, can we build a model that not only predicts a caption i.e., a collection of labels for an entire image (the image annotation task), but specifically labels the individual objects (or regions of interest) in the image with a collection of labels (the image object-label correspondence task)? We have described a solution to this problem based on a *Mixture-of-Multinomials Hierarchical Dirichlet Process* (MoM-HDP). MoM-HDP generalizes the hierarchical Dirichlet Process (HDP) model (that can be thought of as the analog of a mixture model, but with an infinite number of mixture components assumed in a mixture model) to deal with multi-modal data (e.g., images, text). MoM-HDP thus allows us to circumvent the need, in the case of alternatives such as the multi-modal latent Dirichlet Allocation (MoM-LDA), for a priori and hence potentially arbitrary choice of the number of mixture components or the computational expense of choosing the best performing model from among multiple models corresponding to different choices of the number of mixture components. During training, the model has access to an un-segmented image and its caption, but not the labels for each object in the image. The trained model is used to predict the label for each region of interest in a segmented image. We use variational inference to efficiently estimate model parameters. Our experiments using two large-scale datasets show that the generalization performance of MoM-HDP is superior to that of MoM-HDP as well as the Naive Bayes and Logistic Regression classifiers (in one-against-all learning scenario).

Although our experiments with the MoM-HDP model have been limited to data consisting of images and text, the underlying probabilistic model and the algorithm for training the model readily generalize to data that include multiple modalities (e.g., text, image, speech, etc.). MoM-HDP model can be extended along several interesting directions: The current model is based on a simple bag of features (words, visual features or visual words) representation of the data from each modality. It would be interesting to consider more sophisticated models of interaction among features within and across modalities.

CHAPTER 4. Discriminative Multiple Instance Multiple Label Model with Trace Norm Regularization

In this Chapter we introduce a discriminative model that can be used for image annotation. Unlike generative model, this approach solves the classification problem directly. In particular, we formulate the problem of image annotation as Multiple Instance Multiple Label learning problem by considering each image as a bag of image segments (Multiple Instance assumption) as the set of keywords as a set of labels assigned to each image (Multiple Label assumption). The goal of the model is to maximize the probability of each correct label given the bag using a Noisy-Or model, and to model correlation among labels by using Trace Norm regularization to enforce classifiers for each labels to share weights. We show that this model, unlike some of the previous state-of-the-art models is scalable to a setting where a large number of labels and a large number of images are present in the training data. We also show that the performance of this model is comparable to some of the recent Multiple Instance Multiple Label state-of-the-art models.

4.1 Introduction and Motivation

We introduce a solution to a problem of image annotation. We formulate the image annotation problem as Multiple Instance Multiple Label (MIML) classification task, and present a novel solution to this learning problem.

We begin with describing the related work and how it motivates that the discriminative model is a suitable choice to address the problem of image annotation. The previous work in image annotation relied on probabilistic model to jointly model the probability of the set of image features (or a set of vectors that describe image segments) and a set of keywords that describe the image. The translation model proposed by Duygulu et al. (2002) relies on learning the hidden variables that represent semantic concepts that correlate image features and word features. The correlation model proposed by Blei and Jordan (2003) also models correlations among image features and word features through hidden variables. The work in Duygulu et al. (2002) and Blei and Jordan (2003) was extended by in Chapter 3 to a Hierarchical Dirichlet Process to circumvent the problem of selection of the number of mixture components. The Relevance Model proposed by Lavrenko et al. (2003) models the correlation between the image features and the words locally, within the image. This work was extended on modeling correlation of absence and presence of words in the image by Feng et al. (2004). These models have one underlying principle in common: they model the dependency between a set of image features x and a set of words y using a set of some hidden variable z and then computing the expectation over these hidden variables. In other words, the assumption is that x and y are independent given a hidden variable z, or that $p(x, y|z) = p(x|z)p(y|z)$. Carneiro et al. (2007) observed this, and proposed modeling the probability of image features and keywords directly assuming that the image features are independent given keywords: $p(x, y) = p(x|y)p(y)$.

In contrast we take a discriminative approach: we suggest solving the problem of predicting the keywords from image features and learning the model parameters to optimize $p(y|x)$ directly. Given a collection of feature vectors extracted from the image segments the goal is to predict a set of keywords associated with the image. Therefore this problem is naturally cast as Multiple Instance Multiple Label learning problem. We address this problem by learning a Multiple Instance classifier for each keyword, and by forcing the classifiers to share weights to account for the correlation among the keywords in the images. The classifiers are based on logistic Noisy-OR discriminative models, and the correlations among the keywords is obtained by using Trace Norm regularization as proposed by Amit et al. (2007) for multi-class learning setting.

This Chapter is organized as follows: we describe Single Instance Multiple Label learning in Section 4.2. We present an approach that solves the Multiple Instance learning by converting it into Single Instance learning in Section 4.3. We then introduce the Discriminative Multiple

Instance Multiple Label learning model in Section 4.4. We describe some of the recent state-of-the-art algorithms in Multiple Instance Multiple Label learning in Section 4.5. We then proceed with describing experimental set-up and comparison of our algorithm with other known algorithms in Section 4.6. We conclude this Chapter with discussion in Section 4.7.

4.2 Overview of Multiple Label Learning

We now proceed to introduce our proposed solution to the Multiple Instance Multiple Label learning problem. Our basic approach is to adapt a well-studied framework for learning classifiers that balance the classification error (*loss*) against the complexity of the classifier (using a *penalty* term). Specifically, our solution to the MIML learning problem generalizes the Trace Norm regularization framework introduced by Amit et al. (2007) to a Multiple Instance setting. In particular, we use the observation that the Logistic Loss for single instance learning is the probability of predicting the label correctly, and we generalize the loss to the Multiple Instance setting by using a Noisy-Or model to compute the probability of a label of the bag given the bag. Trace Norm regularization is aimed at classification problems where the classes share some common characteristics, and hence it is useful to exploit the shared characteristics to learn predictors for the classes.

We first briefly describe the Trace Norm regularization framework for Single Instance multi-class classification problem before proceeding to introduce our proposed solution to the Multiple Instance multiple label problem.

4.2.1 Trace Norm Regularization for Single Instance Multi-Class Learning

Multi-class learning problem (also related to Multi-task learning (Caruana, 1997)) is a problem of learning a classifier to recognize multiple and sometimes related categories. It entails learning a mapping $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{L}$ from instances in an instance space $\mathcal{X}$ to labels in $\mathcal{L} = \{l_1...l_M\}$. Crammer and Singer (2001) suggest learning a collection of classifiers, each parameterized by a weight vector w_j for each class l_j by minimizing a trade-off between an average *empirical loss* and a regularizer that controls the complexity of the classifier (some penalty term) and in general terms they propose the general framework

$$h^* = \min_h \left(loss(h) + CPenalty\right)$$

where C is a constant that controls the amount of trade-off. The common penalty term is a regularizer of the form: $||W||_{\mathcal{F}} = \sum_j |w_j||^2$ i.e., the Frobenius norm of the matrix W whose columns are the vectors w_j.

Within this formulation a variety of loss functions can be used (Rosset et al., 2004). For example, if the Hinge Loss is used, this formulation results in an alternative view of Support Vector Machine. If a Logistic Loss is used, the resulting model is known as a regularized Logistic Regression. If a squared-error loss is used, this formulation will result in Regularized Least Squares.

For multi-class setting, Amit et al. (2007) suggested that Trace Norm is a more suitable alternative than the Frobenius norm. They suggested training M classifiers $h_1 ... h_M$ (one for each of the M classes, or equivalently, classifiers $h_1 ... h_M$ predicting the corresponding elements of the vector of binary labels $y_i^1, y_i^2 .... y_i^M$) by trading off the empirical loss of classifiers parameterized by the weight vectors w_j, against the Trace Norm of the weight matrix of all classifiers:

$$\{h_1 ... h_M\}^* = \min_{h_1 ... h_M} \sum_{i=1}^{N} \sum_{j=1}^{M} loss\left(y_i^j, h_j(x_i)\right) + C \left\|W\right\|_\Sigma$$

The $W = [w_1 ... w_M]$ is the matrix containing columns of weights w_j that parameterize each of classifiers h_j. The penalty factor C controls the amount of trade-off between the regularization and the empirical loss. The Trace Norm $\|W\|_\Sigma$ is defined as $\min_{W=FG} \frac{1}{2}\left(\|F\|_{\mathcal{F}}^2 + \|G\|_{\mathcal{F}}^2\right)$ where $\|\cdot\|_F$ is the Frobenius norm. Trace Norm can be thought of as jointly factorizing the matrix W of weights that define the classifiers into the matrices F and G, where F maps the inputs to some feature space space and G performs classification in that space. While Trace Norm is defined by factorizing $W = FG$, such factorization is not needed to compute the Trace Norm, and it can be computed as the sum of singular values of W.

An intuition behind why Trace Norm captures correlations among labels is the following. Let W be factorized as $W = FG$. Then if the decision function h is given as a linear function $h(X) = W^T X$, the function h given the factorization of W equals to $h(X) = G^T \left(F^T X\right)$. Therefore the decision function can be viewed as a two-step process: first, the data is mapped to some lower-dimensional semantic space $Z = \left(F^T X\right)$, and then the classifier is learned in this lower dimensional space using weights G^T via $h = G^T Z$. If there are correlated labels, the instances with correlated labels will be mapped near-by in the semantic space, and this correlation will be captured by the classifier G^T.

A variety of loss functions have been considered. For example, Amit et al. (2007) used a generalized Logistic Loss (to approximate the Hinge Loss). Loeff and Farhadi (2008) used a numerical approximation of the hinge-loss.

4.3 Multiple Label Learning for Single Instances

4.3.1 Hinge Loss for Single Instance Learning

We begin with describing the general framework for Single Instance Multiple Label learning. In the case of Single Instances, Hinge Loss can be used to approximate the Support Vector Machine. Hinge Loss is defined for binary output as

$$hinge(z) = \max\left(0, 1 - z\right)$$

where z is defined as $z_i = y_i^j \left(w_j^T x_i\right)$.

The main complication with using the Hinge Loss as defined, is that max function is not differentiable, thus the Hinge Loss is not differentiable. Therefore Hinge Loss is frequently approximated with some differentiable function. Here we use a Shifted Generalized Logistic approximation of Hinge Loss (as proposed by Zhang and Oles. (2001)):

$$g(z, \gamma) = \frac{1}{\gamma} \log\left(1 + \exp\left(\gamma\left(1 - z\right)\right)\right)$$

which approximates Hinge Loss as $\gamma \rightarrow \infty$. Figure 4.1 shows Logistic Loss, Hinge Loss and Shifted Generalized logistic approximation of Hinge Loss. Other approximations have been

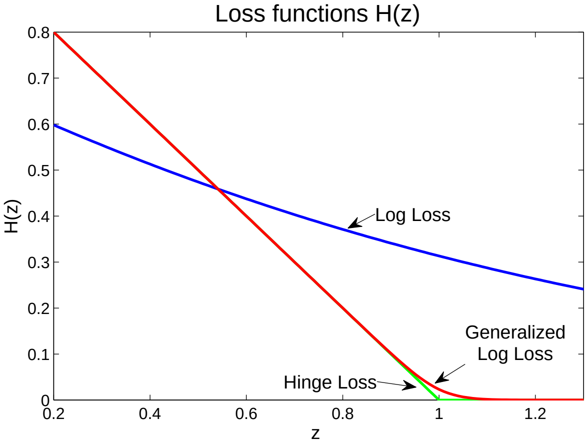


Figure 4.1 Hinge Loss (green), its approximation with Shifted Generalized Logistic Loss with $\gamma = 30$ (red) and Log-loss (blue)

used, such as a numerical approximation of max function (Loeff and Farhadi, 2008).

Since the loss is decomposed as the sum of the losses for each label for each instance, loss term becomes:

$$loss = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{\gamma} \log\left(1 + \exp\left(\gamma\left(1 - z_i\right)\right)\right)$$

The gradient of the loss is given by

$$\frac{\partial loss}{\partial w_j} = \sum_{i=1}^{N} - \frac{\exp\left(\gamma\left(1 - z_i\right)\right)}{1 + \exp\left(\gamma\left(1 - z_i\right)\right)} y_i^j x_i$$

Then the classifiers are learned by trading off loss and penalty on the training dataset D and the model parameters can be learned by solving the following unconstrained minimization problem

$$\{h_1 ... h_M\}^* = \min_{h_1 ... h_M} \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{\gamma} \log\left(1 + \exp\left(\gamma\left(1 - z_i\right)\right)\right) + C \left\| W \right\|_\Sigma$$

by either gradient descent or numeric optimization (such as LBFGS (Liu and Nocedal, 1987)). This problem is convex in W.

4.3.2 Trace Norm Regularization

The penalty term $\|W\|_{\Sigma}$ is computed as the sum of absolute values of the singular values of the matrix: $\|W\|_{\Sigma} = \sum |\gamma_i|$ where γ is a vector of singular values of W and $|\cdot|$ is the absolute value. However, the absolute value function is not differentiable at 0 it is approximated with a smooth function at 0 which can be differentiated. $\|W\|_{\Sigma} = \sum a_{\tau}\left(\gamma_i\right)$ where

$$a_{\tau}(x) = \begin{cases} |x| & |x| > \tau \\ \frac{x^2}{2\tau} + \frac{\tau}{2} & |x| \leq \tau \end{cases}$$

and τ is a small positive number (we used $\tau = 10^{-9}$)

The gradient of the regularizer is given by:Amit et al. (2007):

$$\frac{\partial}{\partial W} J_{reg} = CUa_{\tau}^{'}\left(D\right)V^{T}$$

where $W = UDV^{T}$ is the singular value decomposition of W and $a_{\tau}^{'}(D)$ is a derivative of a_{τ} applied to each element of diagonal of D. The function $a_{\tau}(x)$ is twice-differentiable everywhere and its first derivative is

$$a_{\tau}^{'}(x) = \begin{cases} sign(x) & |x| > \tau \\ \frac{x}{\tau} & |x| \leq \tau \end{cases}$$

4.4 Multiple Instance Learning

Multiple Instance learning poses an additional complication: the loss function needs to be suitable for the case where each data point is a collection of instances. We use two solutions to the Multiple Instance problem: 1) by transforming Multiple Instance learning into Single Instance learning and use the Single Instance Multiple Label framework directly and 2) by deriving the loss function and distance function that accounts for Multiple Instance learning directly.

4.4.1 Multiple Instance to Single Instance Reduction

The first solution proposes the mapping from Multiple Instances into single instances, and then using the single instance Multiple Label learning framework directly. We use the mapping proposed in Zhang and Zhou (2006). First, the training bags are clustered using k-medoids algorithm using Hausdorff distance to measure the distance between the bags. Then after the k medoids are identified, each bag is mapped into a k-dimensional space, so that each feature for a bag i is a distance from this bag to each bag in the resulting k medoids. Formally, let $X = \{x_1 ... x_n\}$ be the training bags, so that each of the points x_i is a bag of K_i vectors $x_i = \{x_{i1} ... x_{iK_i}\}$. The bags are clustered into S medoids using k-medoid clustering, and the distance between two bags is defined as maximum Hausdorff distance (to ensure that the metric is symmetric): $d(x_i, x_j) = \max(d_H(x_i, x_j), d_H(x_j, x_i))$ where $d_H(x_i, x_j) = \sup_{x_{im} \in x_i} \inf_{x_{jl} \in x_j} d(x_{im}, x_{jl})$. Let $c_1 ... c_k$ be the output of the k-medoid algorithm. Then a bag x_i is mapped into R^k as $x_i \rightarrow [d(x_i, c_1), ..., d(x_i, c_S)]$.

Then the following algorithm is used to learn the model parameters W^*

1. Map Multiple Instance dataset D into Single Instance dataset $D^{'}$ and the store k-medoids
2. Learn W^* using framework described in Section 4.3

To classify novel instances, the test bags are first transformed to Single Instances using the k-medoids learned on the training set using the Hausdorff distance.

4.4.2 Multiple Instance Multiple Label Learning Problem

Now we turn to the problem of solving the Multiple Instance Multiple Label problem directly. In order to do so, we are faced with the issue of defining the loss function for the Multiple Instance Multiple Label learning problem. If we were given the label for each of the instance in a bag of instances to be classified, we can use the same hinge loss as in Amit et al. (2007) or Loeff and Farhadi (2008) and compute the loss for the bag of instances by summing up the loss over all of the instances in a bag. How can we compute the loss $l(y_i^j, h_j(x_i))$ for bags of instances when the labels for individual instances in a bag are unknown? We can answer this question by adapting an approach introduced by Raykar et al. (2008) in the simpler setting of the binary Multiple Instance learning problem. Consider the Log Loss: $l(y_i^j, h_j(x_i)) = -\log \frac{1}{1+\exp(-y_i^j h_j(x_i))}$. In the case of logistic regression, log-loss is simply negative log of the probability of y_i^j given the observation x_i. Hence,

$$l(y_i^j, h_j(x_i)) = -\log p(y_i^j|x_i) = -\left(\delta\left(y_i^j, 1\right)\log p(y_i^j = 1|x_i) + \delta\left(y_i^j, -1\right)\log p(y_i^j = -1|x_i)\right)$$

where $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. Let y_{ik}^j denote the jth bit of the vector of class labels for the kth *instance* x_{ik} in the ith bag x_i. Let A^T be the transpose of a matrix A. If we had a classifier defined by w_j with respect to membership in class l_j, we can use sigmoid function to model the probability that the kth *instance* x_{ik} in the ith bag x_i is positive (with respect to membership in class label l_j): $p(y_{ik}^j = 1|x_{ik}) = \sigma(w_j^T x_{ik}) = \frac{1}{1+\exp(-w_j^T x_{ij})}$. Then the probability that the instance is negative with respect to membership in the jth class is given by $1 - p(y_{ik}^j = 1|x_{ik})$. Because a bag is labeled negative only if all the instances in it are negative, we can use a Noisy-Or model to combine the probabilities that the individual instances in the bag are negative:

$$p(y_i^j = -1|x_i, w_j) = \prod_{k=1}^{K_i}\left(1 - p(y_i^j|x_{ik}, w_j)\right) = \prod_{k=1}^{K_i}\left(1 - \sigma(w_j^T x_{ik})\right)$$

The probability that the bag is positive is then given by

$$p(y_i^j = 1|x_i, w_j) = 1 - p(y_i^j = -1|x_i, w_j)$$

4.4.3 Solving the Optimization Problem

Our goal is to find parameters that minimize $J = J_{loss} + J_{reg}$, where $J_{loss} = \sum_{i=1}^N \sum_{j=1}^M l(y_i^j, h_j(x_i))$ and $J_{reg} = C\,\|W\|_\Sigma$. We present the analytical gradients to the loss function over bags and the penalty term.

The gradient of the loss function with respect to the jth column of the weight matrix W is

$$\frac{\partial J_{loss}}{\partial w_j} = -\sum_{i=1}^N\left[\left(\delta\left(y_i^j, 1\right)\frac{\left(1 - p(y_i^j = 1|x_i)\right)}{p(y_i^j = 1|x_i)} - \delta\left(y_i^j, -1\right)\right)\sum_{k=1}^{K_i}\sigma(w_j^T x_{ik})x_{ik}\right]$$

The correctness of the analytical gradient and its implementation was verified using numerical approximation ($\frac{\partial f(x)}{\partial x} \approx \frac{f(x+\epsilon)-f(x)}{\epsilon}$ for $\epsilon \to 0$). In order to speed up the convergence of the gradient-based minimization, we used the Limited BFGS method (Liu and Nocedal, 1987) that avoids computing and storing the Hessian matrix explicitly.

4.5 Limitations of Other State-of-the-Art Multiple Instance Multiple Label Algorithms

We now describe several other Multiple Instance Multiple Label datasets that have been reported as state-of-the-art algorithms on datasets with a small number of labels, and we describe their limitation for tasks that have many labels and many images (therefore these algorithms are impractical for such tasks).

- **Multiple Instance Multiple Label Boost (MIMLBoost)** (Zhang and Zhou, 2006)

  This work draws on two existing algorithms: Multiple Instance boosting (Xu and Frank, 2004) and Multiple Label boosting (Freund and Shapire, 1996). The Multiple Instance boosting generalizes AdaBoost by computing the expected loss over the bag of instances instead of a Single Instance for each label. Multiple Label boosting (AdaBoost.MH) is an algorithm designed to jointly deal with Multiple Labels (or multiple classes). It does so by transforming Multiple Label (or multi-class) learning into binary classification learning problem in the following way: let x_i be an instance and $y_i = \{y_{i1}, ..., y_{iM_i}\}$ be a set of labels assigned to x_i such that each $y_{ij} \in \mathcal{L} = \{l_1 ... L_M\}$. AdaBoost.MH replicates x_i as many times as there are possible labels using the index of the label as an additional feature, and labels each transformed instance positive if the label at that index is assigned to x_i, Formally, the tuple (x_i, y_i) is transformed into M tuples $([x_i, k], \mathcal{Y}(y_i, l_k))$ so that $\mathcal{Y}(y_i, l_k) = 1$ if $l_k \in y_i$ and -1 otherwise. Therefore, if there are N training examples, AdaBoost.MH transforms this dataset into another dataset with MN training examples. Such transformation requires prohibitive memory usage for datasets with a large number of labels with a large number of bags.

- **Joint Multiple Instance Multiple Label Learning** (Zha et al., 2008)

  This algorithm proposes a discriminative model similar in spirit to an undirected Multiple

Label model (Ghamrawi and McCallum, 2005). In short, it models the probability of predicting the correct set of labels by $p(y_i|x_i) = \frac{E(x_i,y_i)}{Z}$. Here $E(x_i, y_i)$ is some energy function computed over instances in the bag and labels and Z is the normalization term to ensure that $p(y_i|x_i)$ is a valid probability distribution. In general, for case of Multiple Labels to compute the normalization term Z one needs to consider a summation of energy functions computed over all possible assignments over the label set. Such assignment is exponential in the number of labels. Zha et al. (2008) used Gibbs Sampling to estimate this probability, however this can be very slow for a large number of labels. In addition, the training scenario is non-trivial to implement and no implementations have been made available by Zha et al. (2008).

- **Kernel Multiple Instance Multiple Label Learning** (Vijayanarasimhan and Grauman, 2009)

  This learning algorithm uses one-against-one Support Vector Machine and a Multiple Instance Multiple Label kernel similar to multiple instance kernel proposed by Gärtner et al. (2002). The major limitation is the one-against-one SVM training, which results into needing to train M^2 classifiers.

- **Maximum Margin Multiple Instance Multiple Label Learning** (Zhang and Zhou, 2008)

  This algorithm is similar in spirit to the Support Vector Machine Vapnik (1995). It defines margin for the bag over the instances in the bag for the labels that are assigned to the bag. The soft-margin version of this learning algorithm defines slack variables for each instance for each class, and therefore there are MN variables in the problem. In addition, the proposed algorithm uses Quadratic Programming to solve the resulting problem, and for datasets with many instances and many labels this program may not even fit in physical memory (as is the case with Corel-5K dataset or IAPR TC-12 dataset).

4.6 Experiments and Results

4.6.1 Data

Microsoft Object Class Recognition (v2) The version 2 of Microsoft Object Class Recognition dataset (MSRC) consists of 591 images. The dataset also provides pixel level ground truth and each pixel is labeled with one out of 23 possible classes or 'void' (class 'void') was not used. Around 80% of the images are associated with more than one label, and there are on average three labels per image. This dataset has been used in the past to evaluate Multiple Instance Multiple Label classifiers in computer vision (Zha et al., 2008; Vijayanarasimhan and Grauman, 2009). While the dataset provides ground truth at pixel level, this information is not used in training, and only the image and the image-level label information is used. As in Vijayanarasimhan and Grauman (2009) we segment the images using normalized cuts (Shi and Malik, 2000) and from each region we extract texton features (Shotton et al., 2006) and color histograms, and there are around 900 features total. Each segment is then treated as an instance, and each image is treated as a bag of segments.

IAPR TC-12 One recent benchmark in image annotation and retrieval is the IAPR TC-12 dataset. It consists of 20,000 images each annotated with keywords from 274 categories. Each image has been manually segmented and annotated according to a predefined vocabulary of labels. From each segment the following visual features were extracted: area, boundary/area, width and height of the region, average and standard deviation in x and y, convexity, average, standard deviation and skewness in both color spaces RGB and CIE-Lab. Each segment is treated as an instance, and each image is treated as a bag.

Corel 5K This dataset, first introduced by Duygulu et al. (2002) which is a widely used benchmark for image annotation. The data set contains 100 images from each of the 50 CDs, where each CD corresponds to a category ('nature', 'safari', 'coast' etc). The data set is split into training and testing sets, such that 10 images from each CD are included in the test set. The images were first segmented using normalized cuts (Shi and Malik, 2000), and for each segment 33 features were computed. These features represent (roughly) four major visual features such as color, shape, size, and texture. These features include:

- Position: Coordinates of the center of mass of the segments
- Size: Area of the segment
- Shape: Convexity and Moment of Inertia of rotation around the center of mass
- Color: Averages and standard deviations of RGB and Lab color spaces of the segments
- Texture: Mean oriented energy computed in 30 degree increments

4.6.2 Experiments

For each dataset we experiment the following classifiers:

- **DMIML$_\Sigma$**: The discriminative Multiple Instance multiple label model with Trace Norm regularization
- **DMIML**: The discriminative Multiple Instance Multiple Label model that uses no regularization ($C = 0$).
- **MI-MatFact**: is Matrix Factorization model (Amit et al., 2007; Loeff and Farhadi, 2008) , but trained using bags of instances transformed into Single Instances using the procedure Section 4.3.
- **MIML-SVM** (Zhang and Zhou, 2006): The bags of instances are transformed into a single meta instance, and an SVM (with linear kernel) is trained for each label (see Zhang and Zhou (2006) for details about the transformation from Multiple Instances into a Single Instance).

4.6.3 Results

MSRC_v2

We perform 5-fold cross-validation on this dataset. In each fold we compute the average area for all classes, and estimate the mean and standard deviation of average areas to report results consistent with previous literature (Zha et al., 2008; Vijayanarasimhan and Grauman, 2009). Following Zha et al. (2008) we remove the label 'horse' and 'mountain' because they have very few instances in this dataset.

We use area (averaged over classes) under the ROC curve (Hanley and McNeil, 1982) to evaluate the performance of our model. In particular, ROC curve describes the trade-off between true-positive and false-positive and the area under ROC curve described that a randomly chosen positive sample will be ranked higher by a classifier than a randomly chosen negative sample. A random classifier has an AUC 0.5, while a classifier that ranks the samples perfectly has an AUC of 1.

The Trace Norm parameter C for DMIML$_\Sigma$ was tuned on the validation set (subset of training data) as to maximize the average AUC.

Method	average AUC
MIMLSVM	0.776 ± 0.02
MIMLBoost (Zha et al., 2008)	0.766
MIMIL (Zha et al., 2008)	0.902
MIL-Kernel (Vijayanarasimhan and Grauman, 2009)	0.896
MI-MatFact	0.8076 ± 0.02
DMIML	0.829 ± 0.031
DMIML$_\Sigma$	**0.906 ± 0.013**

Table 4.1 Performance (Average AUC ± standard deviation) for MSRC V2 dataset

The results suggest that the proposed Multiple Instance Multiple Label approach yields a significantly higher performance than the state-of-the-art Joint Multiple Instance Multiple Label classifier (Zha et al., 2008) and a Multiple Instance Multiple Label kernel (Vijayanarasimhan and Grauman, 2009). Unlike previous approaches, it is scalable to a large number of labels.

IAPR TC-12

We split the dataset into 60% (12,000 images) training and 40% testing (8,000 images).

Method	average AUC
MIMLSVM	0.711
MI-MatFact	0.761
DMIML	0.779
DMIML$_\Sigma$	**0.797**

Table 4.2 Performance (Average AUC) of various MIML approaches on large-scale IAPR-TC dataset

The performance of the MIML classifiers is summarized in Table 4.2.

The results for the large-scale dataset show a similar pattern in performance as the smaller-scale MSRC-v2 dataset: both models benefit from Trace Norm regularization, however not to that much of an extent as for a smaller dataset. The learning algorithm with the highest average AUC is the proposed Trace Norm regularized DMIML followed by DMIML with no regularization. The learning algorithm benefits from using all instances during learning unlike MIMLSVM or MI-MatFact models which lose information during the transformation from Multiple Instance learning to Single Instance learning.

Corel-5k

Finally, we evaluate the algorithms on the Corel dataset. The training set was split into training (4000 images) and validation (500 images) and the parameters were tuned on the validation set. After the parameters were tuned, the model was then retrained with that parameter setting on the full dataset (4500 images) and evaluated on the test set.

To ensure fair comparison, we use average AUC for all algorithms even though it is a common practice to use precision and recall for this dataset (Duygulu et al., 2002; Lavrenko et al., 2003; Blei and Jordan, 2003; Feng et al., 2004; Makadia et al., 2008; Loeff and Farhadi, 2008). We do not do this due to two reasons:

1. The first one is the is a lack of consistency in evaluation protocol of recent advances in image annotation. Given the image annotation literature, there is a wide discrepancy among how the images are annotated. Most works (Duygulu et al., 2002; Lavrenko et al., 2003; Makadia et al., 2008) rank the keywords using the learned classifiers, and then assign keywords that achieve top 5 scores to each test image. However a recent

work that achieves state-of-the-art results (Loeff and Farhadi, 2008) uses a threshold and assigns the keywords if the classifier's score for a given image was above that threshold.

2. The second one is inconsistency in the choice of features. Duygulu et al. (2002); Blei and Jordan (2003); Lavrenko et al. (2003) use features computed from segments. However, Lavrenko et al. (2003) uses features computed from images after partitioning them into rectangles and Makadia et al. (2008) and Loeff and Farhadi (2008) use global features (and report recent state-of-the art results). Given these inconsistencies it is difficult to determine whether the improvement in precision/recall comes from the new features set, or from the number of keywords assigned, or from the learning algorithm itself.

Therefore, we keep the feature set fixed for all the experiments as our goal is to compare the modeling power of the algorithms. We note, however, that we compare our algorithm with the state of the art Matrix Factorization model (Loeff and Farhadi, 2008) as trained on the transformed Multiple Instance to Single Instance problem.

Method	average AUC
MIMLSVM	0.691
MI-MatFact	0.713
DMIML	0.758
DMIML$_\Sigma$	**0.789**

Table 4.3 Performance (Average AUC) on Corel test set

The results in Table 4.3 show what's been consistent with results on other datasets: using Multiple Instance learning directly improves average AUC.

4.7 Conclusion

4.7.1 Discussion

It is interesting to consider the relationship between the proposed Multiple Instance Multiple Label classification problem and some of the other models that have been proposed in the literature. MBRM (Feng et al., 2004) models the joint likelihood of the image regions and labels present and absent in the annotation using a generative model (See Figure 4.2 left). SML (Carneiro et al., 2007) models the joint likelihood of the image regions and words present in the annotation directly (See Figure 4.2 middle). The loss function used in DMIML$_\Sigma$ models the probability of predicting the labels present and absent in the caption correctly given the image regions. Hence, DMIML$_\Sigma$ model (See Figure4.2 right) can be thought of a discriminative counterpart of MBRM and SML. Several previous studies have observed that when the goal is classification, it is often better to optimize a measure that is directly related to classification accuracy instead of relying on a generative model to perform classification (Vapnik, 1995). The results of our experiments with DMIML$_\Sigma$ lend support to this conclusion in the more general setting of Multiple Instance Multiple Label classification e.g., image annotation .

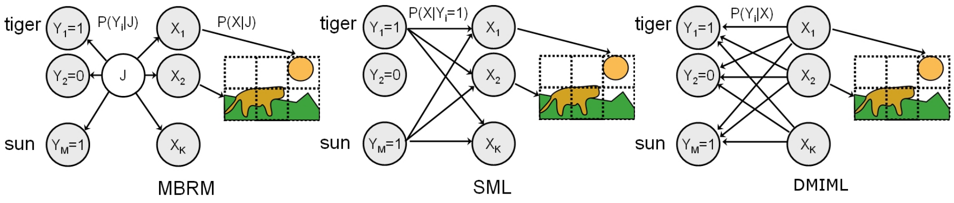


Figure 4.2 Graphical model representation of MRMB, SML and Discriminative Multiple Instance Multiple Label model for image annotation

4.7.2 Summary of Contributions

We explored an approach to image annotation that casts the image annotation problem as a Multiple Instance Multiple Label (MIML) learning problem. The proposed algorithm trains a discriminative model for each label, and uses Trace Norm regularization to find a low-rank solution and to force classifiers to share weights, thus it captures the correlation among labels. The learning algorithm is motivated by loss-penalty formulation, and the loss function, inspired by the Noisy-Or model for Multiple Instance learning is designed specifically for Multiple Instances. The penalty is the Trace Norm of the matrix formed by classifier weights.

We compared the performance of resulting algorithm, DMIML$_\Sigma$ with several existing approaches to Multiple Instance Multiple Label on several datasets: small but challenging Microsoft Visual Classes dataset, and two large image datasets including the widely-used Corel-5K benchmark. In particular, we considered 2 state-of-the-art algorithms in Multiple Instance Multiple Label learning (Zha et al., 2008; Vijayanarasimhan and Grauman, 2009) and a state-of-the-art algorithm in Multiple Label learning (Amit et al., 2007) (and in image annotation (Loeff and Farhadi, 2008)) applied to Multiple Instance learning by transforming Multiple Instance learning to Single Instance learning. We show that on small datasets our learning algorithm has performance comparable to the state-of-the art Multiple Instance Multiple Label algorithms and that it outperforms other known algorithms for MIML on large datasets. In addition, the proposed learning algorithm is scalable to a setting in which a large number of images and the large number of labels are present, unlike many other state-of-the-art Multiple Instance Multiple Label algorithms.

4.7.3 Future Work

We now describe several questions which are of interest to investigate in the future. We are also interested in using this model for *object recognition* and *region annotation*. We would also like to apply this model for annotation of other image datasets, and other tasks which can be posed and Multiple Instance Multiple Label learning problems. It is also interesting to further investigate the effect of Trace Norm regularization and the choice of the regularization parameter. Some theoretical analysis of the consistency of Trace Norm regularization has been done in the past for square loss functions (Bach, 2008). It will be of interest to analyze the consistency of such regularization for Multiple Instance learning.

CHAPTER 5. Graph-Based Semi-Supervised Multiple Instance Multiple Label Learning

In this Chapter we present a solution to semi-supervised Multiple Instance Multiple Label learning. Multiple Instance learning provides a learning framework for weakly-labeled data, and semi-supervised learning is used to strengthen the classifier by utilizing labeled and unlabeled data. Therefore Multiple Instance Multiple Label learning together with semi-supervised learning may serve as a powerful combination to further reduce the cost of aquisition of labeled data. We consider manifold learning framework - a framework that uses labeled and unlabeled data in order to learn a low-dimensional manifold (similarity nearest-neighbor graph). The manifold penalty forces the classifier to assign similar labeles to indirectly similar bags of instances (i.e. bags of instances that may be deemed far away as measured by a certain distance metric, however they may be near-by in the manifold space). We propose two solutions to this problem. The first solution transforms Multiple Instance into Single Instance learning and then it adapts manifold framework for semi-supervised learning in the context of Multiple Label learning. The second solution uses Multiple Instance learning directly. This solution considers the learning framework with trades off between loss and penalty for the classifiers, and we design a loss function suitable for Multiple Instance learning, a penalty function that correlates the classifiers, and a distance metric for bags of instances to learn the manifold for bags directly. Our experimental results show that the simple solution to semi-supervised Multiple Instance learning does not work (the solution that transforms Multiple Instance learning into Single Instance learning), however when using Multiple Instance learning directly, there is improvement in performance of classifiers in the presence of unlabeled data.

5.1 Introduction and Motivation

Labeled data is expensive to obtain for many computer vision tasks, however unlabeled and weakly-labeled data is plentiful. While tagged data is widely available, there is often an issue of label ambiguity and quality, and in general tagged data with high quality of tags is expensive and time-consuming to obtain. Unlabeled data, on the other hand, is plentiful. The goal of semi-supervised learning is to increase the performance of the algorithm by allowing the learning algorithm simultaneously use labeled and unlabeled examples. A variety of semi-supervised learning algorithms have been proposed in the past (see a survey by Zhu (2006)) for simple binary classification tasks. Only recently semi-supervised mutliple label learning began receving attention in computer vision Chen et al. (2008); Zha et al. (2009); Loeff et al. (2009). On the other hand, Multiple Instance Multiple Label (MIML) learning is often used for weakly-labeled data Zha et al. (2008); Vijayanarasimhan and Grauman (2009), and therefore the combination of MIML learning with semi-supervised learning serves as a poweful motivation to even further reduce the cost of aquisition of labeled data. In this work we propose a formulation of MIML learning problem that allows incorporating unlabeled data by allowing the algorithm to take advantage of geometry of the data via manifold regularization.

We begin with a brief overview of related work and approaches to semi-supervised learning, followed by their extension to Multiple Label learning and recent applications in computer vision. The main principle of semi-supervised learning algorithms are that if two points x_1 and x_2 are close in some space, the learning algorithm should produce similar outputs $y_1 = f(x_1)$ and $y_2 = f(x_2)$. Some of the methods included incorporating unlabeled data into generative models using Expectation Maximization (Nigam et al., 2000). *Co-training* (Blum and Mitchell, 1998) assumes that there exist several independent views of data (or independent feature sets), and the classifiers trained on each view are used to label the unlabeled data. The instances that have the most confident predictions (or those with the highest agreement among the classifiers) are labeled by agreement and included into the training set. This procedure is then repeated for a number of iterations. In the similar spirit, *self-training* (Clark et al., 2003; Mihalcea, 2004) uses predictions of the classifier to label the unlabeled corpus, and then includes the instances with the most confident predictions to the training set. There has been a growing body of work in graph-based methods and manifold learning (Sindhwani et al., 2005) where the labeled and unlabeled data are used to recover a low-dimensional *manifold* in which the data lies. Then the classifier is regularized to enforce similar predictions for the instances that lie near-by in the manifold space.

In light of recent advances in semi-supervised learning, and due to the fact that images generally are very high dimensional we use manifold assumption: the high-dimensional data lies on a low-dimensional manifold. If points in high-dimensional Euclidean space are near-by, they may be far away in the manifold space. We therefore use a manifold learning by incorporating manifold regularization (Sindhwani et al., 2005), a framework in which unlabeled data comes into play naturally. Manifold regularization for semi-supervised learning has been used in the past in the context of Support Vector Machines (for binary classification) and Least-Squares Regression (for univariate output functions).

Only recently semi-supervised learning has received attention in computer vision and in image annotation in particular. Li and Sun (2006) proposed to use conditional random field, and then apply co-training to incorporate unlabeled data. However their model assumes that the images are partitioned into small rectangular patches, and that each patch is labeled. This assumption is often unrealistic and such labeling is often not available. (Zha et al., 2009; Chen et al., 2008) proposed graph-based approach for Multiple Label learning for image annotation. They adapted squared loss for Multiple Labels and manifold regularization to incorporate unlabeled data. They considered two manifolds, one of the instance level, and the second one on the label level (the works differ in how the label correlation matrix is estimated). Both works used Sylvester equation approach to solving the optimization learning to estimate the parameters of the model.

In this work we formulate image annotation problem as Multiple Instance Multiple Label problem, and propose a learning framework for this problem which 1) allows the instances be bags of vectors, 2) assumes that the bags can have more than one labels and explore correlation between labels and 3) allows for use of the unlabeled data.

To the best of our knowledge, there has been no work on addressing semi-supervised for joint Multiple Instance Multiple Label classification problem. There has been some work in semi-supervised Multiple Label learning (Zha et al., 2009; Li and Sun, 2006; Chen et al., 2008; Liu et al., 2006; Loeff et al., 2009) and on semi-supervised Multiple Instance learning (Rahmani and Goldman, 2006) however no known work has addressed both problems within one framework.

This Chapter is organized as follows: we introduce the idea behind manifold (graph-based) learning in Section 5.3. In Section 5.4 we present a framework for manifold learning for Single Instances with Multiple Labels. In Section 5.5 we present two approaches for manifold-based semi-supervised Multiple Instance Multiple Label learning: the first approach reduces Multiple Instance learning to Single Instance learning while the second approach uses Multiple Instance learning directly by generalizing the framework proposed in Chapter 4. In Section 5.6 we describe experiments and comparison of classifiers that use manifold regularization versus classifiers that do not, and we describe experiments with using unlabeled data. We conclude in Section 5.7.

We begin with describing a framework for single-instance multiple label learning problem that incorporates the manifold assumption.

5.2 Preliminaries

Let $X_L = \{x_1, ...x_l\}$ be a set of points in $\mathcal{R}^d$ and let these points be labeled with $y_l = \{y_1...y_l\}$ so that $y_j \in \{0, 1\}$. Let $X_U = \{x_{l+1}, ...x_{l+u}\}$ be a set of unlabeled points in $\mathcal{R}^d$.

Let $G = \{\mathcal{V}, \mathcal{E}\}$ be a graph constructed from nodes in $\mathcal{V} = X_L \cup X_U$ and E be the edges that have weights assigned where $\Lambda_{ij}(e)$ indicates the similarity of two nodes. Define the weighted adjacency matrix as Λ so that $\Lambda(i, j) = \Lambda_{ij}(e), e = (i, j) \in \mathcal{E}$ and 0 otherwise. The graphΛ can be constructed using k nearest neighbors for some similarity metric (such as a Gaussian kernel), or distance metric (such as Euclidean Distance).

5.3 Graph-Based Semi-Supervised Learning with Multiple Labels

The general formulation of learning allows for a trade-off between misclassification (loss) and penalty for the complexity of the classifier. (Belkin et al., 2005, 2006) proposed the penalty term to consist of two components: extrinsic (complexity of the classifier) and intrinsic (complexity of the task).

$$J = \min \left(loss + \alpha \underbrace{P_{ext}}_{extrinsic} + \beta \underbrace{P_{int}}_{intrinsic} \right)$$

where α and β control the amount of trade-off between the loss and penalties.

Multi-class learning problem entails learning a mapping $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{L}$ from instances in an instance space $\mathcal{X}$ to labels in $\mathcal{L} = \{l_1 ... l_M\}$. Let M classifiers $h_1 ... h_M$ (one for each of the M classes, or equivalently, classifiers $h_1 ... h_M$ predicting the corresponding elements of the vector of binary labels $y_i^1, y_i^2 .... y_i^M$) be parameterized by weights W, so that w_j are weights for classifier h_j. The loss can be decomposed over instances and labels, and so

$$loss = \sum_{i=1}^{N} \sum_{j=1}^{M} l(h_j(x_i), y_i^j)$$

The extrinsic penalty (complexity of the classifier) can be the standard Frobenius norm of the classifier weights W, however in this work, we use the Trace Norm regularization as described in the previous section.

5.3.1 Manifold Regularization

The intrinsic (data complexity) penalty is based on the manifold assumption. It is assumed that data lies on a low-dimensional manifold (i.e. graph in the instance space) and that the geometry of the data effects the decision function (see Figure 5.2 for example) While the manifold itself is unknown, it can be approximated given the points of some dataset X. The manifold can be approximated by constructing a k-nearest-neighbor graph Λ over the points in X. This assumption allows for the following interesting property: if two points x_i and x_j are far away as measured in Euclidean space, they may be close together on the manifold space if there exists a path between x_i and x_j in Λ (see Figure 5.1 for example on NIST handwritten digits).

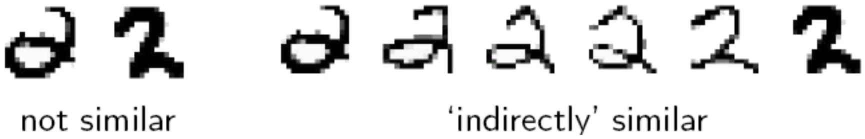


Figure 5.1 Example from NIST handwritten digits recognition. Two handwritten 2's that are dissimilar as measured by Euclidean distance (left). Same handwritten 2's but given a path of other handwritten 2's

Belkin et al. (2006) suggested that since the data assumed to lie in a sub-manifold $\mathcal{M}$, it is natural to use regularization P_{int} or the form $P_{int} = \int_{\mathcal{M}} \langle \nabla f_{\mathcal{M}}, \nabla f_{\mathcal{M}} \rangle = \int_{\mathcal{M}} f \Delta f$ where Δf is the Laplace-Beltrami operator on $\mathcal{M}$. Since the data lies in the discrete world, the Laplace-Beltrami operator is equivalent to graph Laplacian Chung (1997) and is equal to $P_{int} = f^T \Delta f$. The graph Laplacian is defined as $\Delta = D - \Lambda$ where Λ is the weighted adjacency matrix and D is the diagonal matrix so that $D_{ii} = \sum_j^l \Lambda_{ij}$.

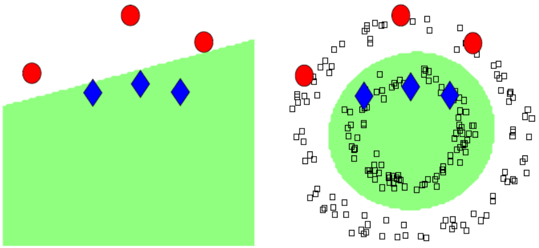


Figure 5.2 Linear classifier for several labeled points (left). What the classifier should be like given the geometry of the data as constructed by unlabeled data

Alternatively the regularization can be seen as the harmonic energy function: $P_{int} = \sum_{i \sim j} \Lambda_{ij} \left(f(x_i) - f(x_j)\right)^2 = f^T \Delta f$ where Δ is as defined earlier. Instead of regular graph Laplacian, normalized Laplacian can also be used and it is defined as $\Delta = D^{-\frac{1}{2}} (D - \Lambda) D^{-\frac{1}{2}}$.

Notice that the labels are not needed to neither construct the manifold (and the k-nearest neighbor graph) nor to compute the penalty term. Therefore, in this framework the unlabeled data comes in naturally. Given a labeled dataset X_L and unlabeled data X_U the adjacency matrix is constructed using $X_L \cup X_U$ as the vertices of the graph and the distance between the points as the weight. The penalty term is then also computed over predictions of the classifier on the labeled and unlabeled data $X_L \cup X_U$.

5.3.2 Manifold Regularization for Multiple Labels

We now extend the graph-based regularization to a setting with multiple labels. In the case of Multiple Labels the predictions for $X_L \cup X_U$ is a matrix $\mathbf{f} = [\mathbf{f_1}, ..., \mathbf{f_l}, \mathbf{f_{l+1}}, ..., \mathbf{f_{l+u}}]$ since each of the predictions $\mathbf{f}_i$ is a vector of the form $\mathbf{f} = [f_1^1 ... f_1^j ... f_1^M]$.

In case of Multiple Labels we consider two manifolds: the first manifold is where the image set lies; the second manifold is where the data labels lie. Therefore, in case of Multiple Labels the penalty P_{int} consists of two terms: $P_{int} = P_{intX} + P_{intL}$ (Chen et al., 2008; Zha et al., 2009) and we consider each one separately.

Let $\mathbf{f}^j$ be a column of $\mathbf{f}$ and so it is predictions of the classifier h_j on the set $X_L \cup X_U$. We consider the sum over the individual penalties of the classifiers as the joint penalty, and so $P_{intX} = \sum_{i=1}^{M} \left(\mathbf{f}^j\right)^T \Delta \mathbf{f}^j = trace(\mathbf{f}^T \Delta \mathbf{f})$ where $trace(A) = \sum_i A_{ii}$ is the trace of the matrix.

The second term enforces consistency with the label assignment and the correlation among the labels as described by Chen et al. (2008); Zha et al. (2009). Let C be $M \times M$ matrix that captures the correlations among labels, as a Laplacian over the label space. Let $C^{'}$ be the matrix that captures similarity among labels. Then its Laplacian C is computed as $C = D_c - C^{'}$ where D_c is a diagonal matrix with $D_{cii} = \sum_{j=1}^{l+u} C^{'}_{ij}$.

Several solutions have been proposed in order to compute the label similarity matrix $C^{'}$ (Chen et al., 2008; Zha et al., 2009). In this work we adapt the construction of the label similarity matrix as described by (Chen et al., 2008). Let $C^{'}_{ij} = \exp(-\lambda(1 - \cos(c_i, c_j)))$ where c_i is a binary vector with $0/1$ entries, so that $c_i^k = 1$ if the label k is contained in caption i and 0 otherwise. The term $\cos(c_i, c_j) = \frac{\langle c_i, c_j \rangle}{\|c_i\| \|c_j\|}$ is the cosine similarity between label l_i and l_j (as assigned to the entire dataset), and λ is a user-defined parameter. Given the matrix $C^{'}$, one then constructs a k-nearest-neighbor graph and then the Laplacian is computed.

5.3.3 Learning with Multiple Labels with Trace Norm and Manifold Regularization

The goal of the learning algorithm is to find functions $h_1 ... h_M$ that minimize the trade-off between the loss and the penalties:

$$\{h_1^*, ..., h_M^*\} = \min_{\{h_1,...,h_M\}} (loss + \alpha P_{ext} + \beta P_{intD} + \gamma P_{intL})$$

where the loss term is computed over the labeled points in X_L

$$loss = \sum_{i=1}^{N} \sum_{j=1}^{M} l(h_j(x_i), y_i^j)$$

the classifier penalty term is the Trace Norm regularization of the classifiers

$$P_{ext} = \|W\|_\Sigma$$

the instance manifold penalty assumes that the observations X_L and X_U lie on a sub-manifold and forces consistency of predictions on the graph that approximates the manifold

$$P_{intD} = trace(\mathbf{f}^T \Delta \mathbf{f})$$

and finally the label manifold penalty assumes that the labellings are consistent with the manifold of the correlation matrix between the labels

$$P_{intL} = trace(\mathbf{f} C \mathbf{f}^T)$$

5.4 Graph-Based Single Instance Learning

We now introduce the learning algorithm that uses Hinge Loss trace norm regularization and manifold regularization. Consider linear classifiers $h_j(x) = w_j^T x$, the Hinge Loss as described in Chapter 4 and the Trace Norm regularization. Then the prediction matrix $\mathbf{f}$ is given by $\mathbf{f}(x) = W^T x$ and the weight matrix W is estimated by minimizing the following function:

As defined previously, let $z_i = w_j^T x_i$ and let $Z = W^T X$

$$
\begin{aligned}
W^* = & \min_W \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{1}{\gamma} \log\left(1 + \exp\left(\gamma\left(1 - z_i\right)\right)\right) \\
& + \alpha \left\| W \right\|_{\Sigma} \\
& + \beta Z^T \Delta Z \\
& + \gamma Z C Z^T
\end{aligned}
$$

Notice that if $\beta = \gamma = 0$ the problem is identical to the learning algorithm presented in Chapter 4. As before, we use LBFGS (Nocedal and Wright, 2000) to find the minimum. The gradient of the loss term and the Trace Norm regularization term are presented in Chapter 4. All that is left is to compute the gradient of the manifold regularization term:

$$
\begin{aligned}
\frac{\partial}{\partial W} Z^T \Delta Z &= \left(\left[\Delta + \Delta'\right] Z\right)^T \frac{\partial Z}{\partial W} \\
&= X \left(2 \Delta Z\right)^T
\end{aligned}
$$

The gradient of the label-level manifold regularization term is computed similarly:

$$
\frac{\partial}{\partial W} Z \Delta Z^T = X^T \left(2 C Z\right)^T
$$

If the last term is ignored by setting γ equal to 0, then this algorithm is similar to the one proposed by Loeff et al. (2009) (the difference is the choice of approximation of Hinge Loss).

5.5 Graph-Based Multiple Instance learning

Multiple Instance learning poses two additional complications: the loss function and the distance metric/similarity metric need to be suitable for the case where each data point is a collection of instances. We use two solutions to the Multiple Instance problem: 1) by transforming Multiple Instance learning into single-instance learning and use the single-instance Multiple Label framework directly and 2) by deriving the loss function and distance function that accounts for multiple instance learning directly.

5.5.1 Distance Metric for Multiple Instances

We begin with defining the distance between two bags. Following Zhang and Zhou (2006), we use Hausdorff distance.

Hausdorff distance defined as the largest of the smallest distances between all possible elements in the set X and Y: $d(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y)$ where $d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$ is Euclidean distance between x and y. Since Hausdorff distance as defined is not symmetric, it is made symmetric using

$$d_H(X, Y) = \max \{d(X, Y), d(Y, X)\}$$

We use this distance to construct the k-NN graph and the resulting Laplacian.

In addition, several alternatives can be considered: Multiple Instance kernel as described in Gärtner et al. (2002) or a set kernel as described in Kondor and Jebara (2003).

5.5.2 Transforming Multiple Instance Learning to Single Instance Learning

We adapt the same reduction from Multiple Instance learning to single instance learning as proposed in Chapter 4: use the k-medoid clustering on the training bags, then use the distance from the training bags to the output of the k-medoid clustering as features for the transformed single instances. Given this transformation, learn the parameters W by optimizing the objective function described in 5.4

5.5.3 Solving Multiple Instance Problem Directly

Finally, we propose a method for solving the semi-supervised Multiple Instance Multiple Label learning problem directly. We use the log loss function proposed in Chapter 4 and we use the Hausdorff distance to compute the k-NN graph needed to approximate the image manifold. The value of the function $f^j(x)$ is computed as $p(y^j = 1|x)$ and it is $f(x) = 1 - \prod_{k=1}^{K_i} \left(1 - \sigma(w_j^T x_{ik})\right)$.

Let $f_j(X)$ be prediction of classifier j on all the training bags and let $\mathbf{f}$ be the matrix of predictions for all bags and let $\mathbf{f}_i$ be all predictions for bag x_i. The model parameters W are learned by minimizing the following objective function:

$$
\begin{aligned}
W^* = \min_W \sum_{i=1}^{N} - & \left[\left(\delta\left(y_i^j, 1\right) \frac{\left(1 - p(y_i^j = 1|x_i)\right)}{p(y_i^j = 1|x_i)} - \delta\left(y_i^j, -1\right) \right) \sum_{k=1}^{K_i} \sigma(w_j^T x_{ik}) x_{ik} \right] \\
& + \alpha \left\| W \right\|_\Sigma \\
& + \beta \sum_{j=1}^{M} f_j(X)^T \Delta f_j(X) \\
& + \gamma \sum_{i=1}^{N} \mathbf{f}_i(X) C \mathbf{f}_i^T
\end{aligned}
$$

The gradients of the loss function and the Trace Norm regularization terms are presented in Chapter 4. We present the gradient for the manifold regularization terms.

$$\frac{\partial}{\partial w_j} f_j(X)^T \Delta f_j(X) = \left(2\Delta f_j(X)\right) \frac{\partial}{\partial w_j} f_j(X)$$

and

$$\frac{\partial}{\partial w_j} \mathbf{f}_i C \mathbf{f}_i^T = \left(2C\mathbf{f}_i^T\right) \frac{\partial}{\partial w_j} \mathbf{f}_i(X)$$

The gradient of the function f_j evaluated for one training bag x_i is

$$\frac{\partial}{\partial w_j} f_j(x_i) = \left(1 - p(y_i^j = 1|x_i)\right) \sum_{k=1}^{K_i} \sigma(-w_j^T x_{ik}) x_{ik}$$

and the matrix $\frac{\partial}{\partial w_j} \mathbf{f}_i(X)$ is $M \times d$ matrix in which the row M_j equals to $\frac{\partial}{\partial w_j} f_j(x_i)$ and other rows are zeros.

5.6 Experiments and Results

We now describe experimental set-up and describe the results of our model.

MSRC dataset We use the Microsoft visual classes dataset as described in the previous chapter. First, we examine the effect of regularization without using any unlabeled data. The results (average AUC) as estimated using 5-fold cross-validation is shown in Table 5.1. In each fold entire data was used as labeled. It is remarkable that manifold regularization does not help and does not improve the AUC for the transformed Multiple Instance into single instance algorithm. This result is at odds with the the conclusion by Loeff et al. (2009) where they observed improved performance when using unlabeled data in the manifold framework. This leads to believe that the transformation from Multiple Instance to Single instance loses enough information that the manifold assumption no longer holds. In particular, this transformation potentially causes the learned graph (after using Euclidean distance to learn the k-nearest-neighbor graph) to be very different from the graph in the original Multiple Instance space.

We see however that manifold learning improves the AUC for Multiple Instance Multiple Label algorithm when the Multiple Instance assumption is used and when the k-nearest-neighbor graph used for graph-Laplacian is learned by measuring distances between the bags (using the Hausdorff distance).

Method	AUC
DMIML$_\Sigma$ $\beta = 0, \gamma = 0$	0.906
DMIML$_\Sigma\beta$ tuned, $\gamma = 0$	0.915±0.01
DMIML$_\Sigma\beta, \gamma$ tuned	**0.919±0.01**
MI-MatFact $\beta = 0, \gamma = 0$	0.8076 ± 0.02
MI-MatFact β tuned, $\gamma = 0$	0.8076 ± 0.02
MI-MatFact β, γ tuned	0.8076 ± 0.02

Table 5.1 Effect of Manifold regularization for MSRC dataset

We also notice that tuning of the third parameter, γ does not help neither the transformed Multiple Instance to Single Instance learning nor the Multiple Instance Learning (unlike results reported in previous work by Zha et al. (2009); Chen et al. (2008)). This can be explained by the fact that none of these approaches penalized the classifier complexity (only loss function and manifold penalties were used, but not the classifier penalty). In the previous work, the only way to capture label correlation was through the label manifold. In our work, the label correlation is captured by Trace Norm regularization, and we can conclude that the effect of Trace Norm regularization is much stronger than the effect of regularization using the manifold on the label level.

We now explore the effect of using unlabeled data for learning and manifold regularization. In each fold we randomly split the data into labeled and unlabeled sets, keep the labels in the labeled set and remove the label information in the unlabeled set. The results are shows in Table 5.2.

% labeled data	10%	20%	50%	70%
DMIML$_\Sigma$ $\beta = \gamma = 0$	0.71±0.02	0.75±0.012	0.79±0.01	0.85±0.012
DMIML$_\Sigma$$\beta$ and γ tuned	0.73±0.02	0.77±0.015	0.80±0.01	0.86±0.009
MI-MatFact $\beta = \gamma = 0$	0.569±0.02	0.62±0.016	0.727±0.01	0.738±0.034
MI-MatFact β, γ tuned	0.551±0.017	0.62±0.02	0.728±0.01	0.738±0.034

Table 5.2 Using unlabeled data with manifold regularization for MSRC dataset

5.7 Conclusions

In this Chapter we described an approach to semi-supervised multiple instance Multiple Label learning. To the best of our knowledge, this is the first formulation of semi-supervised Multiple Instance multiple label learning and its application to a computer vision problem. We proposed a framework which uses graph-based (manifold) penalty to enforce the classifiers assign similar labels to examples that are indirectly similar if they lie nearby in the graph (manifold) space.

We described two approaches. The first approach converts multiple instance learning into a single instance learning, and then applies semi-supervised learning for single instance Multiple Label learning. This framework trains as many correlated classifiers as there are possible labels, and then uses regularization to enforce classifiers assign similar labels to instances that are indirectly similar. The indirect similarity is determined by constructing a manifold (k-nearest-neighbor graph) and then computing the graph Laplacian. In addition, the second regularization assigns similar predictions to correlated labels. This framework is similar to that proposed by Loeff et al. (2009) with additional regularization term (developed at the same time independently). Remarkably, unlike results reported in Loeff et al. (2009) our experimental evaluation shows that for transformed Multiple Instance learning to single instance learning this framework provides little to no benefit. We believe that this is due to the fact that the transformation to single instance leads to the features be very similar, and thus the “wrong” manifold is constructed so that the examples in the manifold space that are nearby are not necessarily as similar as desired.

The second approach uses Multiple Instances directly. This approach generalizes the correlated Multiple Instance classifier as described in Chapter 4, and then generalizes it using manifold regularization. We developed a distance metric using Hausdorff distance needed to construct the similarity graph for bags. We show that unlike the first approach, there is significant improvement in average AUC when using manifold regularization and no unlabeled data. We also show that there is improvement when using unlabeled data and a small percentage of labeled data.

CHAPTER 6. Conclusion and Discussion

6.1 Summary

Image annotation is a challenging and important problem in computer vision and multimedia information retrieval. It is a problem of assigning a subset of keywords from a fixed vocabulary given the image content. The problem of image annotation presents many challenging and interesting problems from a machine learning point of view. It can be tackled using a classification taks, mutliple lable predition task, or Multiple Instance Multiple Label prediction task.

We have addressed the problem of image annotation in a setting where fully labeled data is scarce, and we presented algorithms that learn from weakly-labeled data (tagged images) and in a semi-supervised learning manner, a combination of weakly-labeled data and unlabeled data. In particular we used generative and discriminative models to tackle the problem of image annotation.

We answered several important questions:

1. *How to select the number of latent components for the Generative Latent Mixture Model for image annotation.*

   We generalized MoM-LDA, a generative model used in the past for image annotation, to a non-parametric model. MoM-LDA is a model that learns a joint mixture model of visual features and text keywords and it poses a problem of selection of the number of mixture components. We have generalized MoM-LDA model to a Hierarchical Dirichlet Process. Dirichlet process allows countably infinite mixture components, thus it allows the model to adapt to the data during learning. We have showed that unlike MoM-LDA, whose performance depends on the number of mixture components, the performance of this model on two datasets is invariant to the number of mixture components.

2. *How to deal with lack of word-object correspondence.*

   We have also addressed the problem of image annotation using Multiple Instance Multiple Label learning. While there may not be direct correspondense between image regions (that are assumed to be objects) and the keywords that the images are annotated with, there is always strong correlation among labels and image keywords. We proposed a novel Multiple Instance Multiple Label learning algorithm that learns as many Multiple Instance classifiers as there are labels in the dataset, and uses trace-norm regularization to force classifiers share weights, and thus captures correlation among labels. We showed that this algorithm has comparable performance to the state-of-the-art algorithms for Multiple Instance Multiple Label algorithms as well as it produces results comparable to state-of-the-art image annotation algorithms. This algorithm is scalable to a setting when the number of keywords is large, unlike previous state-of-the-art MIML algorithms such as Joint Multiple Instance Multiple Label algorithm (Zha et al., 2008) and Multiple Instance Kernel SVM model (Vijayanarasimhan and Grauman, 2009).

3. *How to build good predictive models with limited labeled data.*

   Finally, we have extended the discriminative model to a semi-supervised learning setting to allow the learning algorithm take advantage for labeled and unlabeled data. To the best of our knowledge, this is the first solution of semi-supervised Multiple Instance Multiple Label learning task and its application in computer vision. We have proposed two graph-based frameworks for semi-supervised learning: the first framework first transforms Multiple Instance learning into single instance learning, and the second framework addresses Multiple Instance learning directly. The semi-supervised Multiple Instance Multiple Label framework generalizes the Multiple Instance Multiple Label discriminative model by incorporating additional penalties and enforcing the classifiers to assign similar scores to the instances which lie near-by in the manifold space, as well as assign similar scores to labels with high correlation. Surprisingly, using unlabeled data for transformed Multiple Instance learning to single instance learning gives no benefit, however the benefit of manifold learning and unlabeled data can be observed when using Multiple Instance learning directly.

6.2 Contributions

Specific contributions of this thesis are:

- **Design and implementation of Multi-Modal Hierarchical Dirichlet Process Model** (Yakhnenko and Honavar, 2009c)

  We have developed a learning algorithm for learning joint correlated mixtures of images and text which allows the number of mixtures components to go to infinity thus allowing the model to adapt to the data. This model uses a Hierarchical Dirichlet Process constructed using truncated Stick-Breaking Distribution, and the model parameters are efficiently estimated using variational inference. The model is a generalization of previous probabilistic mixture models used for image annotation (such as MoM-LDA (Barnard et al., 2003)). We showed on two datasets that the performance of this model is invariant to the truncation level, unlike MoM-LDA whose performance is sensitive to the number of mixture components.

- **Design and implementation of scalable model for Multiple Instance Multiple Label learning** (Yakhnenko and Honavar, 2009a)

  We developed a discriminative learning algorithm for Multiple Instance Multiple Label learning. This algorithm models the probability of predicting the correct labels given the bag of instances directly by using a Multiple Instance classifier for each label, and then models correlation among labels by using trace-norm regularization by allowing classifiers share weights. Unlike other Multiple Instance algorithms it does not rely on single instance learning. Unlike other state-of-the-art Multiple Instance Multiple Label classifiers it neither requires to evaluate the sum over all possible label assignments, nor it requires solving a complex quadratic program in a large number of variables. Therefore this is a suitable algorithm for datasets with large numbers of labels. Our experimental evaluation on a benhcmark dataset shows that this algorithm has larger area under the ROC curve as compared to the state-of-the-art Multiple Instance Multiple Label classifiers (Zha et al., 2008; Vijayanarasimhan and Grauman, 2009). We also presented experimental evaluation of this algorithm on two large datasets for which the current state-of-the-art algorithms may be infeasible to train.

- **Formulation of semi-supervised learning for Multiple Instance Multiple Label learning** (Yakhnenko and Honavar, 2009b)

  We introduces a framework which allows semi-supervised learning for Multiple Instance Multiple Labels learning problems and generalized the discriminative model to a setting in which weakly labeled (Multiple Instance assumption) and unlabeled data are available. We introduced graph-based (manifold) regularization which allows to use labeled and unlabeled data and enforce the classifiers to assign similar labels to instances that are near-by in the graph space. Our experimental evaluation showed that this additional regularization improves the prediction power (as measured by average area under the ROC curve) of the classifier for Multiple Instance Multiple Label learning, especially in the presence of unlabeled data, and it further improves the performance of the Multiple Instance Multiple Label algorithm. Using this regularization the performance of our discriminative algorithm is significanlty better than the existing state-of-the-art algorithms (Zha et al., 2008; Vijayanarasimhan and Grauman, 2009). This regularization, however, does not work for transformed Multiple Instance learning to single instance learning, which could be due to loss of information during transformation.

6.3 Future work

Some directions for the future work include:

- **Exploration of other methods for semi-supervised learning and the effect of**

larger amounts of unlabeled data

In this thesis we explored graph-based learning which is one of the many approaches to semi-supervised learning. Many of the other advances in single instance semi-supervised learning can be considered for Multiple Instance semi-supervised learning. In particular, our on-going research is geared towards investigation of *self-training* (Clark et al., 2003; Mihalcea, 2004) for Multiple Instance Multiple Label learning. It will be of interest to consider co-training (Blum and Mitchell, 1998) as well as compare graph-based semi-supervised learning, self-training and co-training and to understand strengths and weaknesses of each of these frameworks.

It would also be of interest to explore the effect of using even more unlabeled data. In this thesis, we explored the effect of using unlabeled data by hiding labels, however in many real-world applications one would like to use as much labeled data as available and generate and incorporate unlabeled data from available sources (such as the web, LabelMe, imageNet, Flikr, etc).

- **Application of the models to other domains**

  In this thesis we considered the problem of image annotation and developed algorithms that were applied in the field of computer vision. However there are other domains that have problems which can be solved using Multiple Instance Multiple Label learning which also have plenty unlabeled data available. It would be interesting to apply the proposed methods in the field of natural language processing and document classification/tagging, web mining, web page classification and tagging and computational biology to name a few.

- **Application and extension of the models to video sequences**

  Videos are naturally encoded as sequences of frames (images) and therefore these models can be applied for video mining (Rosenfeld et al., 2003), prediction of people and objects in the videos (Sivic et al., 2009), recognition of sign language , etc. Video sequences, however, violate the assumption that the data is independently identically distributed (i.i.d.) and therefore these models may not capture implied correlation in the samples. Therefore, it would be of interest to extend the proposed models to a setting that relaxes i.i.d. assumption for the data.

- **Development of richer models for scene understanding with weak supervision**

  So far, we have considered models that predict keywords and annotate images by "naming" (i.e. assign a subset of keywords from a set of possible keywords). While the models described in this thesis account for correlations among keywords, they do not account for structures that may be defined between the keywords. A simple example of a structure that may be interesting to consider is a hierarchical ontology over the keywords (such as that found in WordNet). In particular, an object "person" may be assigned a label "child", "woman", "man" or other, however it may not be necessary to make a prediction on such a low level of granularity. Another structure that exists over the keywords may be semantic or spatial relations between the keywords and objects in the image (such as "person"-walking on-"street" or "sky"-above-"water"). Exploiting and being able to predict such structures in addition to the keywords is of interest to advance the field of full scene understanding.

- **Investigation of alternative methods to keyword assignment**

  One common way of annotating the image is to rank the classifier scores and assign top K scoring labels to the image. However in actuality there typically more or less than K objects that are present in the scene. Therefore alternative ways of annotating images should be investigated. Loeff and Farhadi (2008) assign the keywords by learning classifiers thresholds and assigning labels if the classifier score is above the threshold. The threshold can be further optimized by choosing an appropriate operating point for each label on ROC curve. It is also possible to learn a classifier that will predict the number of keywords for each image. It will be of interest to compare these different strategies of assigning keywords and understand advantages and limitations of each.

- **Exploitation of cheap sources of labeled data**

High-quality labeled data may be expensive and time consuming to obtain in some domains (such as computer-aided diagnostics, computational biology, etc). However, some tasks may require data that can be labeled by people who have general common sense and not necessarily expertise in the domain. Most tasks in computer vision and scene understanding fall in the latter group. Mechanical Turk allows general computer users complete such labeling tasks with high quality and low cost to those who need to have the data labeled. Recently, cost-sensitive active learning began receiving attention in computer vision. Only recently *active learning* began receiving attention in computer vision and Multiple Instance Multiple Label learning (Vijayanarasimhan and Grauman, 2009), however much remains to be done in this area (theoretical analysis, better learning algorithm, performance efficiency and scalability).

APPENDIX A. Exponential families of Distributions

In this appendix we describe exponential family distributions referred to in this chapter. The exponential family defines a unified form that every distribution can be written as. Let θ be paramters of the distribution and let x be a random variable. The exponential family defined the probability of x as:

$$P(x|\theta) = f(x) \exp\left[\theta^T \mathbf{u}(x) - g(\theta)\right]$$

where $\phi(\theta)$ is the natural parameter vector, $\mathbf{u}(x)$ is the sufficient statistic vector and $g(\theta)$ is the log of the normalization factor. The exponential family allows us to easily compute the expectation and variance of the sufficient statistic as the first and second derivative of the log-normalization factor with respect to its natural parameter and therefore:

$$E\left[\mathbf{u}(x)\right] = \frac{\partial g(\theta)}{\partial \theta}$$

$$Var\left[\mathbf{u}(x)\right] = \frac{\partial^2 g(\theta)}{\partial \theta^2}$$

A.1 Multinomial (Discrete) Distribution

A multinomial distribution is a probability distribution over a variable that can take countably finite values. Let x take K values $x_1, x_2, ..., x_K$, then the distribution of x is defined as

$$\begin{aligned} P(x = x_j) &= Multinomial(x|p_1...p_K) \\ &= \sum_{i=1}^{K} p_i \delta(i,x) \\ &= p_j \end{aligned}$$

where $\delta(i,x)$ is the indicator function and $\delta(i,x) = 1$ if $x = i$ and 0 otherwise.

The exponential family for multinomial distribution is written as

$$Multinomial(x|p_1...p_K) = \exp\left(\begin{bmatrix} \log p_1 \\ \vdots \\ \log p_K \end{bmatrix}^T \begin{bmatrix} \delta(x,1) \\ \vdots \\ \delta(x,K) \end{bmatrix} - \log\left(\sum_{i=1}^{K} p_i\right)\right)$$

The log normalization function is $g(\theta) = \log\left(\sum_{i=1}^{K} p_i\right) = 0$ (since $\sum_{i=1}^{K} p_i = 1$) and the parameter vector $\theta = [\log p_1 ... \log p_K]$, therefore the expectation of the natural statistic vector is $\frac{\partial g(\theta)}{\partial \log p_i}$ (first write $\frac{\partial g(\theta)}{\partial p_i} = \frac{\partial g(\theta)}{\partial \log p_i} \frac{\partial \log p_i}{\partial p_i}$ which is equivalent to $1 = \frac{\partial g(\theta)}{\partial \log p_i} \frac{1}{p_i}$ and then solve the resulting equation for $\frac{\partial g(\theta)}{\partial \log p_i}$) and it is:

$$E\left\langle \begin{bmatrix} \delta(x,1) \\ \vdots \\ \delta(x,K) \end{bmatrix} \right\rangle = \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix}$$

A.2 Dirichlet Distribution

The Dirichlet Distribution is the conjugate prior for the parameters of the Multinomial distribution $p_1 ... p_K$:

$$\begin{aligned} P(p_1...p_k) &= Dir(p_1...p_K|\alpha) \\ &= Z(\alpha) \prod_{i=1}^{K} p_i^{\alpha_i - 1}, p \geq 0, \alpha > 0, \sum_{i=1}^{K} p_i = 1 \end{aligned}$$

such that α_i is the ith element of the parameter vector α and the normalization constant $Z(\alpha)$ is defined as

$$Z(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha)}$$

Using the exponential family, this distribution is defined as:

$$Dir(p_1...p_K|\alpha) = \exp\left(\begin{bmatrix} \alpha_1 - 1 \\ \vdots \\ \alpha_K - 1 \end{bmatrix}^T \begin{bmatrix} \log p_1 \\ \vdots \\ \log p_k \end{bmatrix} + \log\Gamma(\sum_{i=1}^{K} \alpha) - \sum_{i=1}^{K} \log\Gamma(\alpha_i)\right)$$

The log of normalization factor $g(\alpha) = \log\Gamma(\sum_{i=1}^{K} \alpha) - \sum_{i=1}^{K} \log\Gamma(\alpha_i)$ and the expectation of sufficient statistic is computed as the derivative of g and it is defined as:

$$E\left\langle \begin{bmatrix} \log p_1 \\ \vdots \\ \log p_k \end{bmatrix} \right\rangle = \begin{bmatrix} \Psi(\alpha_1) - \Psi(\sum_{i=1}^{K} \alpha) \\ \vdots \\ \Psi(\alpha_K) - \Psi(\sum_{i=1}^{K} \alpha) \end{bmatrix}$$

where the Digamma function $\Psi = \frac{\partial \log\Gamma(x)}{\partial x}$ is the derivative of log of Gamma, and can be computed numerically.

APPENDIX B. Updates for Stick-Breaking Distribution

B.1 Updates for Stick-Breaking Distribution Parameters β

The goal is to optimize the term

$$\begin{aligned}\text{maximize: } & L(\beta) = \mathbb{E}_q \log \mathrm{GEM}(\beta; \alpha) + \mathbb{E}_q \log \mathrm{Dirichlet}(\pi, \alpha_\pi \beta) \\ \text{subject to: } & \sum_{i=1}^{K} \beta_i = 1\end{aligned}$$

we use Quadratic Penalty method to optimize this function and satisfy the constraints. We present the gradient of the objective function that will be used in the Quadratic Penalty method as discussed later on.

First, we present the gradient of $\mathbb{E}_q \log \mathrm{Dirichlet}(\pi, \alpha_\pi \beta)$. It is straighforward derivative of exponential familty of Dirichlet distribution with respect to β.

$$\begin{aligned}\frac{\partial \mathbb{E}_q \log \mathrm{Dirichlet}(\pi, \alpha_\pi \beta)}{\partial \beta_k} &= \frac{\partial}{\partial \beta_k} \left(\sum_{i=1}^{K} \sum_{j=1}^{N} \alpha_\pi \beta_i \mathbb{E}_q \log \pi_j(i) + \log \Gamma(\sum_{i=1}^{K} \alpha_\pi \beta_i) - \sum_{i=1}^{K} \log \Gamma(\alpha_\pi \beta_i) \right) \\ &= \alpha_\pi \sum_{j=1}^{N} \mathbb{E}_q \log \pi_j(k) + \frac{\partial}{\partial \beta_k} \left(\underbrace{\log \Gamma(\sum_{i=1}^{K} \alpha_\pi \beta_i)}_{= \log \Gamma(\alpha_\pi) \text{ since } \sum_{i=1}^{K} \beta_i = 1} - \sum_{i=1}^{K} \log \Gamma(\alpha_\pi \beta_i) \right) \\ &= \alpha_\pi \sum_{j=1}^{N} \mathbb{E}_q \log \pi_j(k) - \Psi(\alpha_\pi \beta_k)\end{aligned}$$

To construct the gradient for the $\mathbb{E}_q \log \mathrm{GEM}(\beta; \alpha)$ term some additional work needed (see Liang et al. (2009) for more detail). The GEM is defined in terms of β, however β are defined in terms of proportions u from Beta distribution. Therefore change of variables is needed.

First, let $\beta_K = 1 - \sum_{i=1}^{K-1} \beta_i$. Then define a map from stick-breaking proportions u to stick-breaking weights β: $\phi : [0,1]^{K-1} \rightarrow \mathcal{T}$ so that $\beta_z = u_z \prod_{i=1}^{z-1}(1-u_i)$. The inverse map is given by $\phi^{-1}(\beta) = \left(\frac{\beta_1}{T_1}, \frac{\beta_2}{T_2}, ..., \frac{\beta_{K-1}}{T_{K-1}}\right)$ where $T_z = 1 - \sum_{j=1}^{z-1} \beta_j$. The density of GEM is naturally defined in terms of u, however we need to define it in terms of β.

Using the general fact that given $f(x_1...x_n)$ and a map $y_i = g_i(x_1, ...x_n), i = 1...n$ we can define $f(y_1, ...y_n) = f\left(g_1^{-1}(y_1, ...y_n), ..., g_n^{-1}(y_1, ...y_n)\right) |J(y_1, ...y_n)|$ where $|\cdot|$ is the determinant and J is Jacobian of g^{-1}.

Then

$$\begin{aligned}\log GEM(\beta) &= \log\left(\left|J(\phi^{-1})\right| \prod_{z=1}^{K-1} Beta(\frac{\beta_z}{T_z}, 1, \alpha)\right)\\ &= \log \prod_{z=1}^{K-1} Beta(\frac{\beta_z}{T_z}, 1, \alpha) + \log\left|J(\phi^{-1})\right|\end{aligned}$$

Therefore, we need to compute Jacobian of ϕ^{-1} and its determinant. Liang et al. (2009) showed that the Jacobian is a lower-triangular matrix with $\frac{1}{T_j}$ on the diagonal, and therefore its determinant is the product of diagonal entries, and so

$$\left|J(\phi^{-1})\right| = \prod_{i=1}^{K-1} \frac{1}{T_i}$$

Therefore,

$$\begin{aligned}\log GEM(\beta) =& \log \prod_{i=1}^{K-1} Beta(\frac{\beta_i}{T_i}, 1, \alpha) + \log \prod_{i=1}^{K-1} \frac{1}{T_i}\\ =& \sum_{i=1}^{K-1} \log \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)\Gamma(1)} \left(1 - \frac{\beta_{zi}}{T_i}\right)^{\alpha-1} - \sum_{i=1}^{K-1} \log T_i\\ =& (\alpha-1)\log T_K - \sum_{i=1}^{K-1} \log T_i\end{aligned}$$

Then the gradient is given by

$$\frac{\partial}{\partial \beta_k} \log GEM(\beta) = \begin{matrix} 0, k = K \\ \sum_{z=k+1}^{K-1} \frac{1}{T_z} + \frac{\alpha-1}{T_K}, k \neq K \end{matrix}$$

The beta prior is only applied to the first $K-1$ proportions, since the Kth one is always fixed to 1.

B.2 Quadratic Penalty

Quadratic penalty method (Nocedal and Wright (2000)) is a method of solving a constrained optimization problem where the contraints could be any function (linear or non-linear). The optimization problem is of the form:

$$\text{maximize: } f(x)$$
$$\text{subject to: } c_i(x) = 0, i = 1...N$$

Quadratic penalty method transforms constrained problem into unconstrained by suggesting to solve an alternative problem:

$$\text{maximize } Q(x) = f(x) + \frac{\mu}{2}\sum_{i=1}^{N} c_i^2(x)$$

as $\mu \to \infty$. At solution $\nabla Q = \nabla f + \mu \sum c_i \nabla c_i = 0$ and $c_i = 0$ so the constraints are satisfied. Since the optimization problem may be unstable for large values of μ, the problem is solved iteratively. For each step k, fix μ_k and solve the problem Q_k for x_k^*. At step $k+1$, intialize $x_{k+1} = x_k *$ and set $\mu_{k+1} = 10^t \mu_k$ ($t = 1$ or 2). Such procedure results in a more stable solution.

Since the contraint function is linear in β other optimization methods can be considered, such as projected gradient, or reduced gradient.

APPENDIX C. Gradient of MIML loss function

Gradient of probability of a negative bag:

$$\begin{aligned}
\frac{\partial}{\partial w_j} \log p(y_i^j = -1|x_i, w_j) =& \sum_{k=1}^{K_i} \frac{\partial}{\partial w_j} \log \left(1 - \sigma(w_j^T x_{ik})\right) \\
=& \sum_{k=1}^{K_i} - \frac{\frac{\partial}{\partial w_j} \sigma(w_j^T x_{ik})}{1 - \sigma(w_j^T x_{ik})} \\
=& \sum_{k=1}^{K_i} - \frac{\left(1 - \sigma\left(w_j^T x_{ik}\right)\right) \sigma\left(w_j^T x_{ik}\right) x_{ij}}{1 - \sigma(w_j^T x_{ik})} \\
=& \sum_{k=1}^{K_i} -\sigma\left(w_j^T x_{ik}\right) x_{ij}
\end{aligned}$$

Gradient of probability of a positive bag:

$$\begin{aligned}
\frac{\partial}{\partial w_j} \log p(y_i^j = 1|x_i, w_j) =& \frac{\partial}{\partial w_j} \log \left(1 - \prod_{k=1}^{K_i} \left(1 - \sigma(w_j^T x_{ik})\right)\right) \\
=& \frac{-\frac{\partial}{\partial w_j} \prod_{k=1}^{K_i} \left(1 - \sigma(w_j^T x_{ik})\right)}{1 - \prod_{k=1}^{K_i} \sigma(w_j^T x_{ik})} \\
=& \frac{-\frac{\partial}{\partial w_j} \prod_{k=1}^{K_i} \left(1 - \sigma(w_j^T x_{ik})\right)}{p(y_i^j = 1|x_i, w_j)}
\end{aligned}$$

in order to take the gradient $\frac{\partial}{\partial w_j} \prod_{k=1}^{K_i} \left(1 - \sigma(w_j^T x_{ik})\right)$ use the following trick: $\prod_{i=1}^{n} f_i(x) =$

$\exp\left(\log\left(\prod_{i=1}^{n} f_i(x)\right)\right)$. Therefore, we use chain rule:

$$\begin{aligned}\frac{\partial}{\partial x}\prod_{i=1}^{n} f_i(x) &= \frac{\partial}{\partial x}\exp\left(\log\left(\prod_{i=1}^{n} f_i(x)\right)\right)\\ &= \prod_{i=1}^{n} f_i(x)\frac{\partial}{\partial x}\log\left(\prod_{i=1}^{n} f_i(x)\right)\\ &= \prod_{i=1}^{n} f_i(x)\left(\sum_{i=1}^{n}\frac{\partial}{\partial x}\log f_i(x)\right)\end{aligned}$$

therefore

$$\begin{aligned}\frac{\partial}{\partial w_j}\prod_{k=1}^{K_i}\left(1-\sigma(w_j^T x_{ik})\right) &= \prod_{k=1}^{K_i}\left(1-\sigma(w_j^T x_{ik})\right)\frac{\partial}{\partial w_j}\log\left(1-\prod_{k=1}^{K_i}\left(1-\sigma(w_j^T x_{ik})\right)\right)\\ &= -\frac{p(y_i^j=-1|x_i,w_j)}{p(y_i^j=1|x_i,w_j)}\sum_{k=1}^{K_i}\frac{\partial}{\partial w_j}\log\left(1-\sigma(w_j^T x_{ik})\right)\\ &= \frac{1-p(y_i^j=1|x_i,w_j)}{p(y_i^j=1|x_i,w_j)}\sum_{k=1}^{K_i}\sigma\left(w_j^T x_{ik}\right)x_{ij}\end{aligned}$$

APPENDIX D. Computational Tricks and Dealing with Large Exponents

We often need compute things like $\log\left(\exp(a)+\exp(b)\right)$ for large exponents a and b. If the value of the exponent is very large, such operation will cause an overflow and result in ∞ however the log of the result is computable. We show trick on how to make this quantity computable.

$$\begin{aligned}
\log(e^a + e^b) =& \log\left(\frac{e^a + e^b}{e^{\max(a,b)}} e^{\max(a,b)}\right) \\
=& \log\left(e^{a-\max(a,b)} + e^{b-\max(a,b)}\right) \\
&+ \log\left(e^{\max(a,b)}\right) \\
=& \log\left(e^{a-\max(a,b)} + e^{b-\max(a,b)}\right) + \max(a,b)
\end{aligned}$$

Now the exponents become numbers less than or equal to 0, and they can be computed.

Another application of the trick is computation of $\frac{e^a}{e^b+e^c}$. If the exponents are very large, we run into the problem of taking a log of infinity or dividing infinity by infinity. However we can apply similar trick to make the computation feasible.

$$\begin{aligned}
\frac{e^a}{e^b + e^c} =& \frac{e^{\max(a,b,c)} e^a}{e^{\max(a,b,c)} \left(e^b + e^c\right)} \\
=& \frac{e^{a-\max(a,b,c)}}{e^{b-\max(a,b,c)} + e^{c-\max(a,b,c)}}
\end{aligned}$$

The special case is the Hinge loss and its derivative. We need to compute $\log(\exp(a)+1)$ and $\frac{\exp(a)}{\exp(b)+1}$. Note that $1 = e^0$.

BIBLIOGRAPHY

Aldous, D. (1983). Exchangeability and related topics. *l'École d'été de probabilités de Saint-Flour*, XIII:1198.

Amit, Y., Fink, M., Srebro, N., and Ullman, S. (2007). Uncovering shared structures in multiclass classification. In *Proceedings of International Conference in Machine Learning*, pages 17–24.

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174.

Bach, F. (2008). Consistency of trace norm regularization. *The Journal of Machine Learning Research*, 9:1019–1048.

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

Barnard, K. and Forsyth, D. (2001). Learning the semantics of words and pictures. In *Proceedings of International Conference on Computer Vision*, pages 408–415.

Belkin, M., Niyogi, P., and Sindhwani, V. (2005). On manifold regularization. In *Proceedings of Artificial Intelligence and Statistics*, pages 17–24.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. and Jordan, M. I. (2004). Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144.

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of ACM SIGIR conference on Research and development in information retrieval*, pages 127–134, New York, NY, USA. ACM.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of Computational Learning Theory*, pages 92–100.

Bosch, A., Zisserman, A., and Munoz, X. (2006). Scene classification via pLSA. In *Proceedings of European Conference on Computer Vision*, pages 517–530.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757 – 1771.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Caputo, B., Tommas, T., Müller, H., Deserno, T. M., and Kalpathy-Cramer, J. (2009). ImageCLEF. http://www.imageclef.org/2009/medanno.

Carneiro, G., Chan, A. B., Moreno, P. J., and Vasconcelos, N. (2007). Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:394–410.

Caruana, R. (1997). Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28:41–75.

Chen, G., Song, Y., Wang, F., and Zhang, C. (2008). Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of SIAM International Conference on Data Mining*, pages 410–419.

Chung, F. (1997). *Spectral Graph Theory.* Number 92 in CBMS Regional Conference Series in Mathematics, American Mathematical Society.

Clark, S., Curran, J. R., and Osborne., M. (2003). Bootstrapping pos taggers using unlabelled data. In *Proceedings of Conference on Computational Language Learning*, pages 49–55.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

DeGroot, M. H. (1970). *Optimal Statistical Decisions (Probability & Statistics).* McGraw-Hill, New York.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

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

Duygulu, P., Barnard, K., de Freitas, N., and Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of European Conference on Computer Vision*, pages 349–354.

Everingham, M., Van-Gool, L., Williams, C., Winn, J., and Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challgenges/VOC/voc2007/workshop/index.html.

Fei-Fei, L., Iyer, A., Koch, C., and Perona, P. (2007). What do we perceive in a glance of a real-world scene? *Journal of Vision*, 7:1–29.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Proceedings of Computer Vision and Pattern Recognition*, pages 524–531.

Feng, S., Manmatha, R., and Lavrenko, V. (2004). Multiple bernoulli relevance models for image and video annotation. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1002–1009.

Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall.

Freund, Y. and Shapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of International Conference in Machine Learning*, pages 148–156.

Gärtner, T., Flach, P., Kowalczyk, A., and Smola, A. (2002). Multi-instance kernels. In *Proceedings of International Conference in Machine Learning*, pages 179–186.

Ghamrawi, N. and McCallum, A. (2005). Collective multi-label classification. In *Proceedings of 14th ACM international conference on Information and knowledge management*, pages 195–200.

Gupta, A. and Davis, L. S. (2008). Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proceedings of European Conference on Computer Vision*, pages 16–29.

Hanley, J. and McNeil, B. J. (1982). The meaning and use of area under a receiver operating characterisitc (ROC) curve. *Radiology*, 143:29–36.

Hanson, A. R. and Riseman, E. M., editors (1978). *Computer Vision Systems*. Academic Press, New York.

Hardoon, D., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). A correlation approach for automatic image annotation. In Li, X., Zaiane, O., and Li, Z., editors, *Proceedings of Second International Conference on Advanced Data Mining and Applications, ADMA 2006*, volume 4093, pages 681–692. Springer.

Heisele, B. (2003). Visual object recognition with supervised learning. *IEEE Intelligent Systems*, 18:38–42.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of International Joint Conferences in Artificial Intelligence*, pages 688–693.

Ishwaran, H. and James, L. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96:161 – 174.

Ishwaran, H. and James, L. (2003). Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13:1211 – 1235.

Jin, R. and Ghahramani, Z. (2002). Learning with multiple labels. In *Advances in Neural Information Processing Systems 15*, pages 897–904.

Klein, D. and Manning, C. (2002). Conditional structure versus conditional estimation in NLP models. In *Proceedings of Empirical Methonds in Natural Language Processing*, pages 9–16.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* MIT Press.

Kondor, R. and Jebara, T. (2003). A kernel between sets of vectors. In *Proceedings of International Conference in Machine Learning*, 361-368.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference in Machine Learning*, pages 282–289.

Lavrenko, V., Manmatha, R., and Jeon, J. (2003). A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems 16*, pages 251–259.

Li, L.-J. and Fei-Fei, L. (2007). What, where and who? classifying event by scene and object recognition. In *Proceedings of International Conference on Computer Vision*, pages 1–8.

Li, L.-J., Socher, R., and Fei-Fei, L. (2009). Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 2036–2043.

Li, W. and Sun, M. (2006). Semi-supervised learning for image annotation based on conditional random fields. In *Proceedings of Image and Video Retrieval*, pages 463–472.

Liang, P., Jordan, M. I., and Klein., D. (2009). *The Oxford Handbook of Applied Bayesian Analysis*, chapter Probabilistic grammars and hierarchical Dirichlet processes.

Liang, P., Petrov, S., Jordan, M., and Klein, D. (2007). The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 688–697.

Liu, D. C. and Nocedal, J. (1987). On the limited memory method for large scale optimization. *Mathematical Programming*, 45:503–528.

Liu, Y., Jin, R., and Yang, L. (2006). Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of Conference on Artificial Intelligence*, pages 421–426.

Loeff, N. and Farhadi, A. (2008). Scene discovery by matrix factorization. In *Proceedings of European Conference on Computer Vision*, pages 451–464.

Loeff, N., Farhadi, A., Endres, I., and Forsyth, D. A. (2009). Unlabeled data improves word prediction. In *Proceedings of International Conference on Computer Vision*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Makadia, A., Pavlovic, V., and Kumar, S. (2008). A new baseline in image annotation. In *Proceedings of European Conference on Computer Vision*, pages 316–329.

Maron, O. and Lozano-Pérez, T. (1997). A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10*, pages 570–576.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Henry Holt and Co., Inc. New York, NY, USA.

McCallum, A. (1999). Multi-label text classification with a mixture model trained by EM. In *Proceedings of Conference on Artificial Intelligence Workshop on Text Learning.*

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of Computational Natural Language Learning*, pages 33–40.

Myung, J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47:90 – 100.

Ng, A. and Jordan, M. I. (2001). On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, pages 841–848.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134.

Nixon, M. and Aguado, A. S. (2008). *Feature extraction and Image Processing.* Academic Press, 2nd edition.

Nocedal, J. and Wright, S. (2000). *Numerical Optimization.* Springer, New York.

Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology.* The MIT Press, Cambridge, Massachusetts.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman, San Francisco.

Picard, R. W. and Minka, T. P. (1995). Vision texture for annotation. *Multimedia Systems*, 3:3–14.

Ponce, J., Hebert, M., Schmid, C., and Zisserman, A., editors (2007). *Toward Category-Level Object Recognition.* Springer, New York.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286.

Rahmani, R. and Goldman, S. A. (2006). MISSL: Multiple-instance semi-supervised learning. In *Proceedings of International Conference in Machine Learning*, pages 705–712.

Ray, S. and Craven, M. (2005). Supervised versus multiple instance learning: an empirical comparison. In *Proceedings of International Conference in Machine Learning*, pages 697–704.

Raykar, V. C., Krishnapuram, B., Bi, J., Dundar, M., and Rao., R. B. (2008). Bayesian multiple instance learning: automatic feature selection and inductive transfer. In *Proceedings of International Conference in Machine Learning*, pages 808–815.

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings of International Conference on Computer Vision*, pages 10–17.

Rosenfeld, A., Doermann, D., and DeMenthon, D., editors (2003). *Video Mining.* Springer, New York.

Rosset, S., Zhu, J., and Hastie, T. (2004). Margin maximizing loss functions. In *Advances in Neural Information Processing Systems 16*, page 1237.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor., J. (2005). Learning hierarchical multi-category text classification models. In *Proceedings of International Conference in Machine Learning*, pages 744–751.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2005). LabelMe: a database and web-based tool for image annotation. Technical report, MIT AI Lab Memo AIM-2005-025.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Shotton, J., Winn, J., Rother, C., and Criminisi., A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of European Conference on Computer Vision*, pages 1–15.

Sindhwani, V., Niyogi, P., Belkin, M., and Keerthi, S. (2005). Linear manifold regularization for large scale semi-supervised learning. In *Proceedings of International Conference in Machine Learning Workop on Learning with Partiall Classified Training Data*.

Sivic, J., Everingham, M., and Zisserman, A. (2009). "Who are you?" - Learning person specific classifiers from video. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 1145–1152.

Snedecor, G. W. and Cochran, W. G. (1989). *Statistical Methods*. Iowa State University Press.

Sotiris, D., Grigorios, T., A., M. P., and Ioannis, V. (2005). Protein classification with multiple algorithms. In *Advances in informatics: (10th Panhellenic Conference on Informatics, PCI 2005)*, pages 448–456.

Szeliski, R. (2009). *Computer Vision: Algorithms and Applications*. In preparation.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Torralba, A. (2008). Object recognition and scene understanding. http://people.csail.mit.edu/torralba/courses/6.870/6.870.recognition.htm.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag, New York.

Vijayanarasimhan, S. and Grauman, K. (2009). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 2262–2269.

Viola, P., Platt, J., and Zhang, C. (2005). Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems 18*, pages 1417–1424.

Winston, P. H., editor (1975). *The Psychology of Computer Vision*. McGraw-Hill, New York.

Xu, X. and Frank, E. (2004). Logistic regression and boosting for labeled bags of instances. *Lecture Notes in Artificial Intelligence*, 3056:272–281.

Yakhnenko, O. and Honavar, V. (2009a). Discriminative multiple instance multiple label model with trace-norm regularization for image annotation. In *Preparation*.

Yakhnenko, O. and Honavar, V. (2009b). Graph-based semi-supervised multiple instance multiple label learning. In *Preparation*.

Yakhnenko, O. and Honavar, V. (2009c). Multi-modal hierarchical Dirichlet process model for predicting image annotation and image-object label correspondence. In *Proceedings of SIAM International Data Mining Conference*, pages 281–294.

Yao, J., Antani, S., Long, R., Thoma, G., and Zhang, Z. (2006). Automatic medical image annotation and retrieval using SECC. In *Proceedings of 19th IEEE Symposium on Computer-Based Medical Systems*, pages 820–825.

Zha, Z.-J., Hua, X.-S., Mei, T., Wang, J., Qi, G.-J., and Wang, Z. (2008). Joint multi-label multi-instance learning for image classification. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Zha, Z.-J., Mei, T., Wang, J., Wang, Z., and Hua, X.-S. (2009). Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20:97–103.

Zhang, M.-L. and Zhou, Z.-H. (2006). Multi-instance multi-label learning with application to scene classification. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 1609–1616. MIT Press.

Zhang, M.-L. and Zhou, Z.-H. (2008). M3MIML: A maximum margin method for multi-instance multi-label learning. In *Proceedings of International Conference on Data Mining*, pages 688–697.

Zhang, Q. and Goldman, S. (2001). EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems 14*, pages 1073–1080.

Zhang, T. and Oles., F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31.

Zhu, X. (2006). Semi-supervised learning literature survey. Technical Report TR-1530, Department of Computer Scienece, University of Madison-Wisconsin.