LEARNING STRUCTURED PREDICTION MODELS FOR IMAGE LABELING

by

Xuming He

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Learning Structured Prediction Models for Image Labeling

Xuming He
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2008

Many fundamental tasks in computational vision can be formulated as predicting unknown properties of a scene from a static image. If the scene property is described by a set of discrete values in each image, then the corresponding vision task is an *image labeling* problem. A key issue in image labeling concerns how to exploit the context information in images, as local evidence is often insufficient to determine the label value. This thesis takes a statistical learning approach to the labeling problem, focusing on two main issues in incorporating context into the labeling process: 1) what are the efficient representations of contexts for labeling? and 2) how do we learn the context representations for a labeling task from data?

We discuss two learning situations based on different degrees of data availability. In the first case, enough fully-labeled data are available for learning. So we develop a discriminative labeling framework based on a Conditional Random Field (CRF), in which multiscale feature functions are proposed to capture the image/label contexts at several spatial scales. Those feature functions affect the labeling from local to global levels: some aspects of the contexts concern co-occurrence of objects in the image, while other aspects concern the geometric relationships between objects. To extend the range of object classes and image database size that the system can handle, we also propose a modular CRF model that integrates the bottom-up image cues and top-down categorical information. The second case has a less strict requirement on the training data, including not only a small number of fully-labeled data, but also a large number of coarsely-labeled ones. We present a hybrid unsupervised-supervised approach that combines a generative topic model with discriminative label classifiers. The topic

model is used to model the co-occurring image features for representing image context, and it is extended such that the topics are not only applied to image features but also to labels. We examine the performance of our models on several real-world image databases, and compare our systems to baseline methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One fundamental problem in computer vision is to analyze and understand the scene in a static intensity image. Although it seems to be an effortless task for humans, this remains perhaps the most intriguing and challenging problem in computer vision [18]. Generally, image understanding will be greatly facilitated if we can infer certain underlying properties of the image components, such as the curvature of a surface, or the type of objects in a region. For some situations, the image understanding task itself is to infer the object type of every image region. For example, Figure 1.1 shows a typical input image to an automatic driving system, and the system has to figure out which parts of an image are road surface and which parts are obstacles, in order to drive safely. Therefore, it is an interesting problem to design computational models to fulfill such visual inference tasks automatically.

We are interested in a major class of visual inference tasks, where the underlying properties come from a finite (possibly large) discrete set, such as the object categories in Figure 1.1. When the discrete set is viewed as a set of labels, the visual inference essentially assigns predefined labels to the components of raw intensity images. We refer to this type of vision assignment task as an *Image Labeling* problem. Image labeling spans a wide spectrum in the domain of computational vision. In low-level vision, edge or boundary detection can be formulated as labeling image pixels into 'edge' or 'non-edge' [22]. In mid-level vision, image segmentation can be viewed as assigning labels to every pixel of the image, but where the labels have no specific semantic meanings and are permutable [79]. In high-level vision, the object detection task becomes a special form of image labeling as we assign object categories to the image regions containing significant objects and treat other part of the image as background [39, 77].

In this thesis, we take an integrated approach to image labeling, including visual tasks from

Main class	Sub class	Data
Unlabeled		0x0000
Sky		0x0100
Vertical	Sidewalk	0x0201
	Guard rail	0x0202
	Car	0x0203
	Pedestrian	0x0204
	Vegetation	0x0205
	Mixed	0x0206
	Other	0x0207
Ground	Road surface	0x0308
	Road marking	0x0309
	Mixed	0x030a
Mixed		0x040b

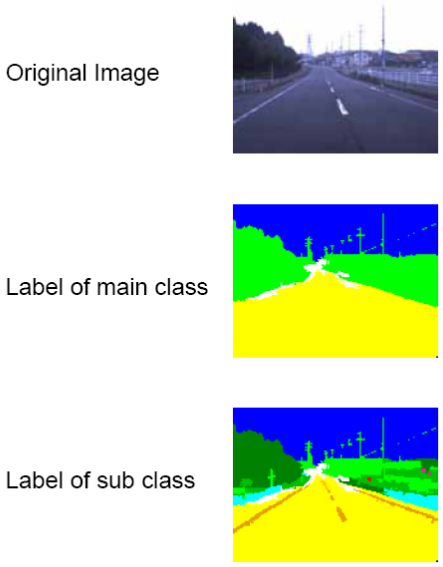

Figure 1.1: A typical example of an image labeling task, in which the label set corresponds to the object classes (from an automated driving data set). The labeling includes two levels: 'Main class' is at the coarse level, while 'Sub class' is at the detailed level. (Courtesy of Toyota, 2005)

the mid-level to the high-level. To be specific, our goal is to label every pixel of an image with certain general high-level information (e.g., object categories). By combining image segmentation with region categorization, we can explore the potential interaction between them and make them complement each other. From the viewpoint of segmentation, our approach overcomes the shortcomings in the standard bottom-up approach by incorporating top-down category-based information. From the viewpoint of region classification, we can not only predict the category of objects, but localize the objects and their boundary precisely. While the target of our formulation shares many similarities with the object recognition task, our perspective is mainly region-oriented and less concerned with individual objects.

Image labeling, in general, is still a challenging problem. Due to the loss of information in image formation and the complexity of the underlying three-dimensional world, the image features often cannot provide enough evidence to resolve the ambiguity in labeling a single static image. For instance, we can see in Figure 1.2, that the color and texture at a local level (a few pixels wide) can sometimes be enough to identify the pixel class–e.g., uniformly blue patches often correspond to the sky; however, typically this is complicated by the large overlap between classes (water can also be blue) and the noise in the image. On the other hand, the

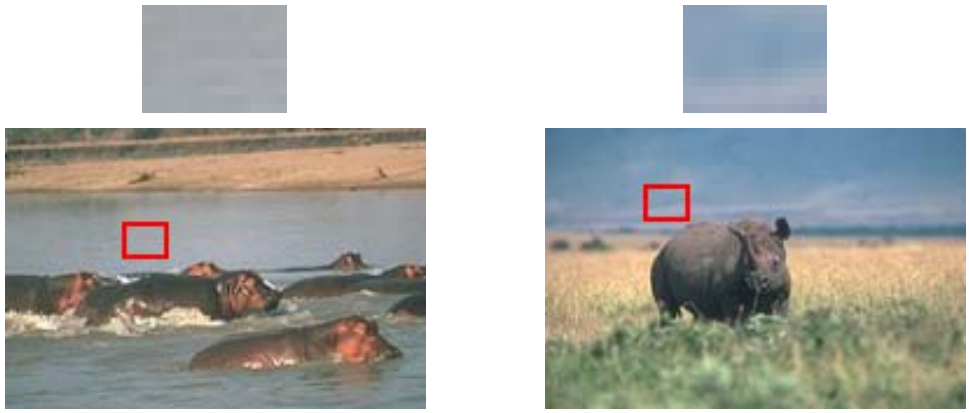

Figure 1.2: Top: two small image patches that are difficult to label based on local information. Bottom: images containing the patches. The surrounding scene makes it clear what the patches are (left: water; right: sky).

local image patches are clearly identifiable given our knowledge about the scene. Therefore, a key step in image labeling is to incorporate informative prior knowledge on the labels and images, so that the hidden properties of a scene (its labels) can be recovered from intensity images. In particular, the labels usually reflect the orderly structures of underlying scenes, such as the internal configuration or the external environment of objects. In order to make consistent label predictions, it is important to include such structural prior knowledge, which has been recognized by early researchers in scene interpretation [74, 23, 44].

To address the issue of integrating priors, we adopt a *probabilistic* approach to the image labeling problem. The family of probabilistic graphical models [32] are capable of representing and combining the structural priors in a consistent and yet flexible manner. With the probabilistic machinery, we can not only make use of different kinds of information in the same framework, but also model uncertainty in predicting image labels easily. More importantly, the models of image labeling can be learned systematically from an annotated image dataset, without requiring heuristic tuning or combining of model parts. This may save considerable effort in building a complex labeling system, and make the model adaptive to new environments.

In the following section, we will discuss the key questions raised in the probabilistic image labeling approach. In Section 1.2 and 1.3, we outline the main ideas of our work in response to the key issues. Section 1.5 summarizes the content of each chapter in this thesis, and Section 1.6 lists the major contributions from this thesis.

1.1 Main Issues in Probabilistic Image Labeling

To apply the probabilistic framework to the image labeling problem, we need to address the following three central issues:

1) In each image, the structural prior information of its scene required by consistent labeling essentially defines a *context* for labeling each image element. The context of an image element may include information from other parts belonging to the same object, or from the surrounding objects with different labels. For a single intensity image, an important class of context is the two-dimensional projection of 3D scenes. An example is given in Figure 1.2: two small image patches are ambiguous at a very local scale but clearly identifiable in their 2D context due to the surrounding objects. An important characteristic of the 2D context is that an image contains contextual information useful for labeling at several scales. Some aspects of the contextual information concern the *local* geometric relationships between objects—e.g., fish tend to be in water and airplanes in the sky; while other aspects concern the *global* location of objects in the image—e.g., the sky tends to be at the top of the image and the water at the bottom. To achieve better labeling performance, an image labeling approach should be able to incorporate these multiscale contexts into its model. This presents the first issue: how can we extract and represent the spatial structural priors, in order to combine relevant contextual information from different sources and levels?

2) There is a spectrum of methodologies of probabilistic modeling, ranging from *generative* to *discriminative* approaches. Generative approaches describe how image data are generated from hidden factors (including labels), and build the joint distribution of all observed and hidden variables. Using Bayes theorem, the labels then can be inferred from inputs. On the other hand, treating the labels as output and the images as input, discriminative models focus on modeling the input-output mapping and the prediction performance. Therefore, those two emphasize different aspects of statistical structure of the data. A typical example of two types of models are given in Figure 1.3. So the second issue is: which kind of modeling approach is most appropriate for the image labeling task? Further, image labeling usually involves complex probabilistic models, which makes the learning task typically hard. Thus, a related question is: how do we approximate the intractable computation in image models such that the models can be learned efficiently?

3) In vision, many different types of data are available, with different degrees of annotation. Images with detailed annotations (i.e., with every pixel labeled) are usually expensive to obtain. In practice, we may have access to a small set of fully annotated images, and many partially

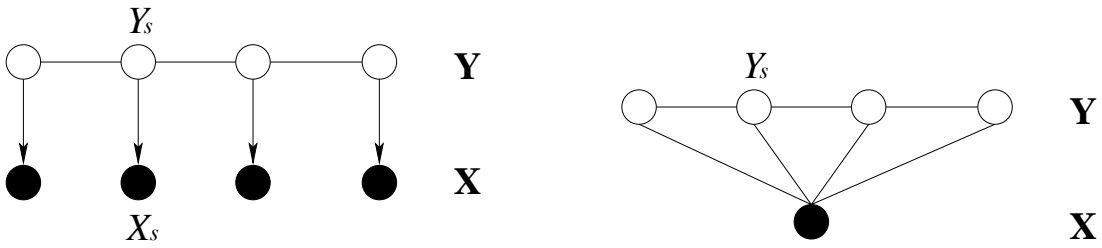

Figure 1.3: Generative model architecture (Left) vs. discriminative model architecture (Right) with linear chain structure for one-dimensional labeling tasks. The input is denoted as $\mathbf{X}$, and the output is denoted as $\mathbf{Y}$. s indexes the labeling sites.

annotated images. Figure 1.1 shows an example of two kinds of labeling with different levels of granularity: some regions contain detailed labels, while others are coarsely labeled, such as 'mixed' or 'other'. Therefore, to be able to scale up, we need to address the third issue: what model structures and learning methods are capable of utilizing coarsely or weakly labeled data to relieve the burden of data collection?

These three issues form the center of the investigations reported in this thesis. We will discuss these questions and provide our solutions to them by building a series of structured probabilistic models for image labeling. The remainder of this chapter presents the main ideas of our approaches, which will be developed under two circumstances characterized by different data availability: 1) Fully labeled image datasets are available for building the labeling model (see Section 1.2); 2) A small amount of fully-labeled images and a larger coarsely-labeled image set are available for constructing the model (see Section 1.3).

1.2 Structured Discriminative Models for Image Labeling

For some vision applications, such as automated navigation, fully labeled image datasets can be obtained from the corresponding research domains. In such a case, we propose to model the contexts for labeling by two-dimensional label patterns that also potentially depend on image information. By incorporating those label patterns, the labeling of an image site will not only depend on the local bottom-up image information, but also integrate the labeling information from its neighboring sites on the image plane. While using such context information in labeling is not new, e.g., [34], our approach emphasizes incorporating the neighboring label information from multiple scales. Those contexts from different scales impose different kinds of constraints that are useful for labeling. In particular, we use multiscale label patterns to model the contexts

from local to global scales: some aspects of the contexts concern local co-occurrence of objects in the image, while other aspects concern the global geometric relationship between objects.

Given the context representations, both generative and discriminative approaches can be applied in principle, as labels and images are fully observed. However, for real life images, generating input images from the categorical labels can be quite complex to model. Even if modeling the image is possible, many labeled images may be required to build the joint model of labels *and* images. We are mainly interested in estimating the distribution of labels given the observed image; even when this distribution is simple, the true underlying generative model may be quite complex. So devoting model resources and degrees-of-freedom to the generative image model can be unnecessary [21]. Therefore, we take a discriminative approach to image labeling. The goal is to build a model to predict the label configuration given an image, instead of generating the input image or its features. Focusing on the prediction task may simplify the model structure, and match the method used in building the model with target applications. As our approaches also include structured context modeling, we refer to them as *structured discriminative models*.

Based on different topologies of image elements on the image plane, we develop two different context representations based on label patterns. One is for the case that image elements lie on a regular lattice structure, where we use multiscale label templates to model the context. The other is for the case that image elements form an irregular structure on the image plane, in which we represent the contexts using global image/label statistics and a mixture of several sub-components modeling local label patterns. These two representations are described in further detail below.

1.2.1 Modeling Context with Multiscale Label Templates

When the image elements lie on a regular lattice structure, we propose to represent the context information in a form of label templates. In its simplest incarnation, a label template is a large and coarse region where each site (corresponding to an image element) has a certain label value, or possibly no label at all (meaning a "don't care" pixel). We use two different levels of label template to represent the contexts: one is the *global label templates*, which attempt to represent geometric relationships between objects, and have specific locations in the image. These relationships can be complex, such as an object lying below or inside other object, both of arbitrary shape. The other is the *regional label templates*, which are intended to represent local geometric relationships between objects, such as edges, corners or T-junctions. Their

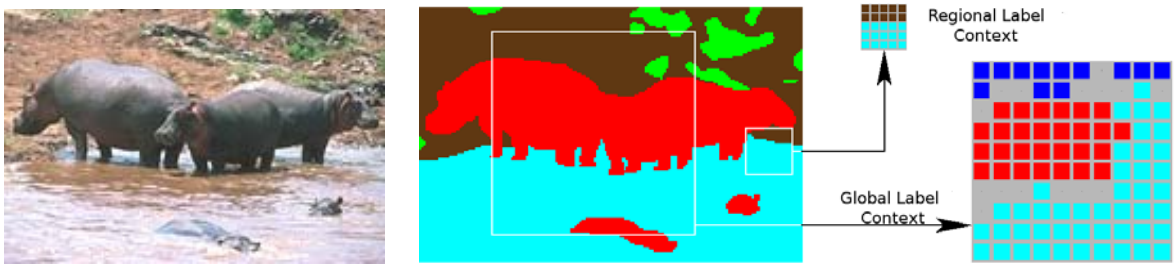

Figure 1.4: Left: Original input image. Right: Ground truth labeling and example label contexts: regional (each cell corresponds to one site), which matches a boundary with ground (brown) above water (cyan); and global (each cell corresponds to 10×10 sites), which matches a rhino or hippo (red) in the water (cyan) with sky (blue) above the horizon. "Don't care" cells are blank (gray color). For detailed label colors and abbreviations, see key in Figure 3.6.

extent is much smaller and they are location-independent–a moving template. Examples are shown in Figure 1.4. Here the smaller (regional) label template encodes a pattern of ground pixels above water pixels, while the bigger (global) label template encodes sky pixels at the top of the image, rhino/hippo pixels in the middle, and water pixels near the bottom. Note that the global features can operate at a coarser resolution, specifying a common value for a patch of sites in the label field.

The actual label templates (global and regional) we use are more general: at each pixel, rather than a single label we use a vector of parameters whose values indicate the relative presence of a label. Thus, at a given site a "don't care" label has uniform values for each possible label, while a sharply defined label has one large value. Our implementation of different sources of information is probabilistic, conditional on the image. That is, given a test image, each of them produces a conditional probability that the candidate labeling is correct. Thus we have at every pixel a number of probabilities, or opinions, that say which label value is correct. These probabilities are then combined to yield a single probability for the candidate labeling by taking their product. An image labeling that does not satisfy well many opinions will have a negligible probability. This does not mean that all opinions must be satisfied at every pixel, (which may be impossible); but that the labeling satisfying most will be the winner. Intuitively, given an image and a candidate labeling for the whole image, we match it against every valid label template. A good image labeling should match reasonably well one or more templates, while a bad one should not match well any of them.

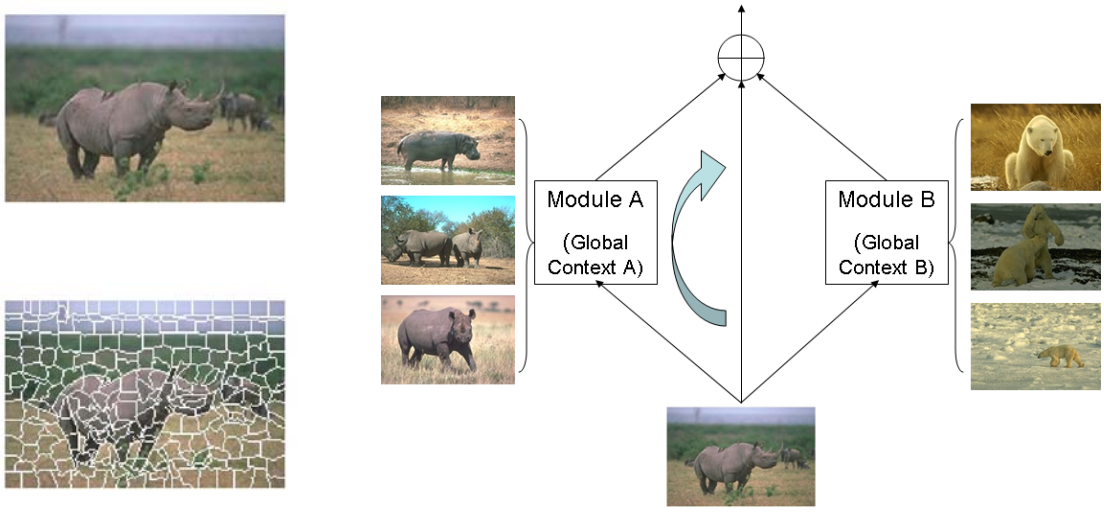

Figure 1.5: Left: An original image (top) with 120x180 pixels becomes a 300 super-pixel image (bottom), where each contiguous region with a delineated boundary is a super-pixel. Right: An example of divide-and-conquer strategy. Two global context groups are formed, one is for topic area and the other is for arctic area. The image statistics are used to select the relevant global context.

1.2.2 Modeling Context with Group Statistics

A labeling algorithm operating at the pixel level will typically be highly redundant, due to the similarity between neighboring pixels within each object category. A pixel level model will also be sensitive to, and limited by the resolution of an image. Therefore, a higher-level image representation than the pixel image is desirable for efficient labeling. We adopt a representation of image elements, called *super-pixels*, which groups a small patch of similar pixels to form a larger unit [59]. Figure 1.5 shows an instance of a super-pixel representation of an image. Note that even if the size of a super-pixel is small, we significantly reduce the number of units to be labeled, which allows a compact model to be constructed without much sensitivity to the resolution of the image.

However, the super-pixel representation introduces irregular structures on the image plane, making it difficult to build label templates for every image. Instead, we represent the contexts by position-independent image/label statistics, which are also implemented in two levels. At a global level, the context is described by a typical aggregate distribution of labels in the image, which specifies how frequently each class will appear given the input image. We can define

a number of global contexts by clustering the image labelings into groups such that the data in each group share a similar aggregate label distribution. While those constraints are weaker than the label templates in terms of describing spatial structure, they are invariant to changing of image positions. At a regional level, the context is represented by a local pairwise statistics between the labels of each image element and its neighbors. The statistics incorporate label co-occurrence priors as well as boundary information from the image.

We take a divide-and-conquer approach to integrate these two levels of contextual information with bottom-up image cues for labeling. Our model has the form of a mixture of several modules, each of which focuses on a single global context (See Figure 1.5). Within each module, the pairwise statistics are global-context specific, and model a simpler regional context. They specify a label distribution based on neighboring label values. Similarly, we have a module-specific classifier predicting label distributions based on local image cues. To label an image, each module combines the three levels of information in a multiplicative way to form a prediction, and the image information is used to select which module should output its prediction. The advantage of this approach lies in the modularity: because each module focuses on single global context, this model has a better capability to handle many context settings and categories.

1.2.3 Learning in Structured Discriminative Models

The context representations in our models have a parameterized form, and are learned from the labeled data. While those multiscale label patterns provide more flexibility in modeling the contexts, they also increase the complexity of the model structure. The learning algorithms based on efficient dynamic programming cannot be easily extended to those models. We design two efficient learning algorithms for building the structured discriminative models. First, we extend a powerful unsupervised learning algorithm to conditional models, leading to an effective approximate supervised learning method. The second algorithm decouples the learning of a full model into separate learning of its components, which is more manageable in labeling a large dataset.

1.3 Hybrid Models and Learning with Coarsely-Labeled Data

The structured discriminative models provide the current state-of-the-art method for labeling problems. However, we usually have to face the data availability issue in learning a structural

labeling model from data. The fully-labeled data set required by discriminative learning is difficult to obtain for real world problems. Especially in the vision domain, assigning detailed labels to every pixel manually is very time-consuming and sometimes intrinsically ambiguous. Even if it is possible for some cases, the fully-labeled data sets are typically small, which limits the applicability of discriminative labeling frameworks to large-scale problems.

On the other hand, it is easier to obtain some weak labels for images in several ways. First, when the label values have different levels of granularity in terms of their semantics, a coarser level of labeling is easier to achieve than more detailed levels. The regions with coarser labels have simpler boundaries and are easier to recognize by labelers (e.g., see Figure 1.1). Second, it is easy to partially label images and leave some image regions unlabeled, regions that are either not relevant to the target problem, or too vague to be recognized. A typical example is the LabelMe dataset [62], in which only key objects are labeled. Also, these datasets may not label the region precisely, and only provide bounding boxes for target regions. Finally, many images have captions that describe the content of images by a set of key-words. Those key-words can be viewed as weak labels in the sense that there are no correspondences between image regions and the labels. In general, the weaker those labelings are, the harder it is to make use of them for learning.

We focus on the case of weak labeling where images are labeled at a coarser level. Ideally, we want to develop a method that can utilize both types of datasets: not only a small set of detailed labels, also a larger set of coarsely-labeled inputs. As the labelings belong to different levels, it is difficult to model contexts with label patterns. Instead, we like to incorporate a data model of generating inputs to capture the *image contexts* for labeling, i.e., the context information in images. In order to model the contexts with potential high-order interactions, we consider a flexible generative approach that captures co-occurring patterns of image features, which are called "topics". Those topics can be viewed as the common image contexts for labeling. As the topics can be any feature configuration in the entire image, this method has the flexibility of modeling image contexts with different complexity. We also extend the model such that the topics are not just applied to input words, but also to labels. Given a topic, the model generates the input data, as well as a topic-dependent probabilistic classifier to predict labels for image regions. With the discriminative component, we can apply the topic model to the labeling tasks. Figure 1.6 shows a toy example of interesting topics for image labeling.

Our model is learned from a small fully-labeled image set and a larger coarsely-labeled set. The data with detailed labels provide precise information for learning the mapping from input to the outputs, whereas the data with coarse labels help the system build a better topic model for

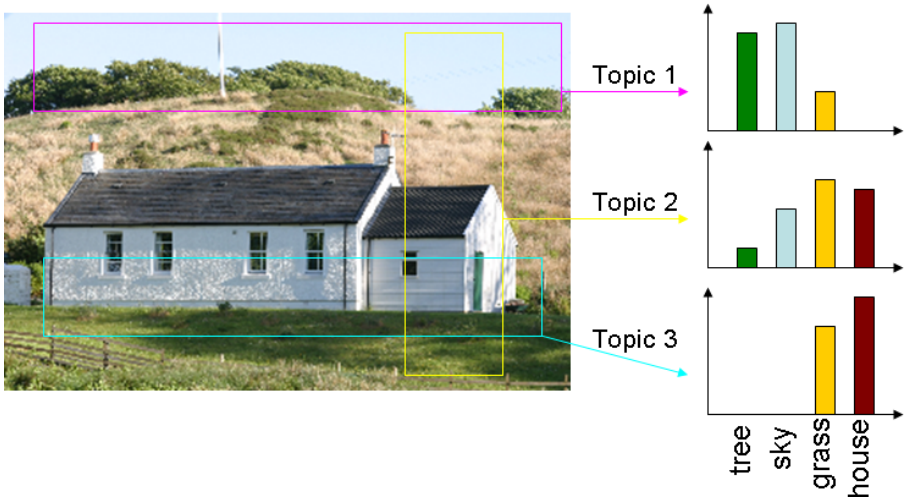

Figure 1.6: A toy example of topics labeled by colored bounding boxes. Three topics are shown in the image and the corresponding label distributions within those topic regions are also plotted in the right panel. Each topic focuses on a different type of context.

image features. The coarsely labeled set prevents the topics from overfitting a limited number of fully labeled images. In addition, those coarse labels give informative cues to learning the topics, which is valuable for the labeling task, but hard to achieve using purely unlabeled data.

1.4 Image Features and Invariance

While our focus has been on context modeling, bottom-up image information is also very important for correct labeling, as it provides direct evidence for object classes. In this thesis, we mostly use appearance properties of intensity images, including color, local texture and edge information, as the bottom-up cues. Those low-level image features are largely independent of object scale, position, and orientation, but are heavily affected by the illumination, shading, and the diversity of object appearance within each class. We also use image position as an informative feature for the classifiers predicting labels from local image cues. However, since we learn the classifiers from image data, the system will automatically decide whether each such feature is included as a useful descriptor of object classes.

Our context models are also moderately invariant to object position, scale and orientation due to the following reasons. First, we incorporate some multiscale and image position independent property into our context representations; Second, we ignore any object-specific

high-level information, such as global shape, so variation in shape does not matter for our models. Finally, while certain components of our models, such as the global label templates, depend on the image coordinates, our learning algorithm will decide the extent to which such components should be included in the model according to the training data. Note that an implicit assumption in our learning approach is that the test image will have the same statistical properties as the training data (i.e., sampling from the same distribution). If this condition does not hold, further adaptation of the systems with more data may be required, or more relevant prior knowledge has to be incorporated.

1.5 Roadmap

The next chapter discusses some of the research that is relevant to the problems addressed in this thesis. It first reviews different vision tasks that have been formulated as image labeling problems, and categorizes them according to the three-level framework of visual processing (low, mid, and high-level vision). The extensive works on different probabilistic models for image labeling are discussed, with emphasis on their flexibility in context modeling. Then labeling methods based on Bayesian decision theory are briefly summarized. At the end of the chapter, we discuss various learning techniques for estimating model parameters in detail.

In Chapter 3, we propose an approach to include contextual features for labeling images, in which each pixel is assigned to one of a finite set of labels. The features are incorporated into a probabilistic framework which combines the outputs of several components. Components differ in the information they encode. Some focus on the image-label mapping, while others focus solely on patterns within the label field. Components also differ in their scale, as some focus on fine-resolution patterns while others on coarser, more global structure. A supervised version of the contrastive divergence algorithm is applied to learn these features from labeled image data. We demonstrate performance on two real-world image databases and compare it to a classifier and a Markov random field.

Chapter 4 proposes an approach to utilizing category-based information in segmentation, through a formulation as an image labeling problem. Our approach exploits bottom-up image cues to create an over-segmented representation of an image. The segments are then merged by assigning labels that correspond to the object category. The model is trained on a database of images, and is designed to be modular: it learns a number of image contexts, which simplify training and extend the range of object classes and image database size that the system can handle. The learning method estimates model parameters by maximizing a lower bound on the

data likelihood. We examine performance on three real-world image databases, and compare our system to a standard classifier and other random field approaches, as well as a bottom-up segmentation method.

Chapter 5 presents a hybrid approach that utilizes both a small set of fully-labeled image data, and a larger set of coarsely labeled images. Our method integrates a generative topic model with discriminative label classifiers for image labeling. The topics model the appearance of image features, which can capture high-order image contexts. Given a topic, the classifiers are used to predict labels. Our learning method uses the fully-labeled data to estimate the mappings from input to detailed labels, and the coarsely labeled data to build a better topic model by regularization. We demonstrate its performance on a real-world image database and compare it to two discriminative approaches.

Chapter 6 summarizes the main results from this thesis and discusses future research directions. We discuss additional image information that could be incorporated into our models, and more structures for modeling context. We also suggest how our methods could be applied to other application domains.

1.6 Major Contributions

In this section, we summarize the main differences between our methods and other labeling methods, and list the most important results reported in the thesis.

- We take an integrated approach to image labeling by combining image segmentation with region categorization. It exploits the potential interaction between two modules: top-down categorical information is used to help the segmentation, and bottom-up image information is used to localize the object classes.

- We develop a probabilistic and prediction-based approach to the image labeling problem. The probabilistic framework allows us to learn the whole model systematically from annotated datasets, whereas the predictive modeling simplifies the model structure, and optimizes the target performance directly.

- We introduce a set of nonlinear feature functions, in the form of probabilistic components, for encoding different types of context. These include local, regional and global levels. A key attribute of those features is ***complementarity***: each component focuses on aspects modeled less well by others.

- We extend a powerful unsupervised learning algorithm (Contrastive Divergence) to its supervised version for estimating parameters.
- We develop a divide-and-conquer strategy that clusters the global structure of images and labels into context groups and constructs a random field for each group. The advantage of this approach is ***modularity***: with each module focusing on a single context group, it has a greater ability to handle a large number of context settings and categories.
- We build our models on larger image elements, called super-pixels, to scale up to bigger images. This pre-processing step simplifies the model structure significantly.
- We propose a hybrid of generative and discriminative models to leverage the labeling accuracy from coarsely labeled image data, substantially extending the applicability of the prediction-based approach.
- We extend the topic model such that the topics are not just applied to input words, but also to labels. Given a topic, the model generates the input data, as well as a topic-dependent probabilistic mapping from input data to the output labels.
- We introduce locality into the topic model, constraining topics to image features that occur together in some local spatial context. Unlike the traditional Markov models, our approach has the flexibility of modeling context with different complexity, including higher-order patterns. Also, the topics with locality constraints can potentially capture image context with different scope in the image plane, which is useful for labeling objects with weakly constrained configurations.

Chapter 2

Literature Review

Image labeling, as a two-dimensional extension of sequence labeling, provides a unifying framework for studying many different problems in computational vision. This single framework has led us to a better understanding of the modeling assumptions and the labeling methods used. Many approaches have been proposed for image labeling, which largely can be divided into non-probabilistic and probabilistic methods. The early work mostly consisted of non-probabilistic methods and brought out some important issues in image labeling. More recently, the probabilistic ones have gained in popularity due to their advantages in addressing those issues.

This chapter focuses on the probabilistic approaches for image labeling. We first review different vision tasks that have been formulated as image labeling problems, and categorize them according to the three-level framework of visual processing in the following section. Section 2.2 then discusses early and existing work on probabilistic models for image labeling, emphasizing their flexibility in context modeling. The probabilistic inference methods for labeling an image are briefly summarized in Section 2.3. In Section 2.4, we discuss various learning techniques for estimating model parameters in details.

2.1 Image Labeling in Computer Vision

While a number of tasks from low-level, mid-level and high-level vision have various and differing goals, they have been formulated as the image labeling problem since the early days of computer vision. The label values have different semantic meanings at different levels, concerning various aspects of a scene property. In general, the label can take values from any set. However, most labeling problems only involve a finite label set with discrete values.

Low-level vision

In low-level vision, the early work on labeling focuses on edge enhancement and boundary detection, in which image pixels are labeled as edge or no-edge pixels. As local evidence for an edge is noisy, neighboring information has been integrated for enhancing weak detections or smoothing out false ones. Several heuristics have been used for incorporating information from neighboring sites, such as using edge direction information to weight neighboring evidence [63], or applying collinearity and continuation constraints to neighboring edge labels [23]. While those problems focus on edges only, image de-noising treats both intensity and edge/no-edge as labels, and assigns the extended label set to an image, such that both region and boundary information from neighboring sites facilitate the labeling process, and their interactions are exploited [21].

More recent approaches treat some continuous scene properties as labels after they are discretized, and also rely on neighboring information on the pixel lattice for a consistent labeling. For instance, depth estimation is considered as labeling an image with a set of depth labels. As directly estimating depth from an intensity image is difficult, sparse depth information can be added as input to provide stronger neighbor evidence [78]. Similar to image de-noising, depth estimation can be combined with detecting depth discontinuities to form a joint labeling problem, with sparse range and intensity edge information as inputs [13]. Recovering shape and reflectance properties from intensity images can also be formulated as a labeling task [19]. Not only local image features, but also a context prior learned from a synthetic dataset are used for inferring those two kinds of properties.

Mid-level vision

Grouping edge segments into lines and curves has been investigated as binary labeling problem, dating back to the work of [93]. A wider scope of context information, such as edges in thin rectangular neighborhoods oriented in eight directions [93], is exploited in those groupings. In [58], object contour completion is also formulated as a binary labeling task, in which the authors assign contour or non-contour labels to all possible contour positions formed by triangularization of the image plane. Multiscale image cues, such as typical junction and continuation properties, are used to group detected edge segments and fill in missing edge parts.

A second task, image segmentation, is an important problem in mid-level vision. The aim of image segmentation is to partition the image plane to a set of non-overlapping regions with coherent image appearance. Assigning each region a distinctive label, we can view image

segmentation as a special labeling task. As those labels can have an arbitrary order, label assignment in general segmentation is fully permutable. Therefore, it is hard to use labels to represent prior knowledge about configuration of the partition. Traditional segmentation methods thus solely rely on the bottom-up image information [79], leading to segmentation inconsistent with surface boundaries in many cases. In an alternative approach, Ren and Malik [59] propose a classification model using a number of low- and mid-level cues to define features of proposed segments, and train a classifier to discriminate (or label) good segments (based on human segmented natural images) from random ones. By incorporating prior knowledge from the training data, this method overcomes certain limitations of the purely bottom-up segmentation approaches.

A subset of segmentation tasks have more meaningful labels coming from a pre-defined label set. For those problems, in addition to the bottom-up image information, we can utilize the prior information on the labels to help the segmentation. Two types of image segmentation tasks have been formulated as such image labeling problems. The first type is domain-specific image segmentation in which the label values of segments are limited and have their semantics. For example, the Sowerby database includes only images about the road scenes in suburban areas, and it has 92 types of labels corresponding to different objects in images. Also, geographical images with annotations have the similar property. For labeling or segmenting such images, local image contexts from over-segmented images are first exploited in a classification system [87, 88]. Recent work concerns contextual information from neighboring labels, in which a contextual prior about label configurations is combined with image information in a Bayesian framework (eg., [8]). The context prior information can come from a local neighborhood, or multiple neighborhoods with varying scales [17], or even dynamic multiscale neighborhoods that depend on image information [1]. The second type of segmentation approaches combine top-down object knowledge with bottom-up information, which has been implemented in several ways. One approach utilizes a deformable template to determine the boundary suggested by bottom-up cues [46], while another combines object templates that have pictorial structures with local smoothness constraint for figure-ground labeling [38]. Further, object knowledge can be represented as pairs of image fragments and their figure-ground labeling from a training set, and a test image is segmented by covering it with a set of fragments whose appearances match the data and whose labeling is locally compatible [7]. A globally compatible labeling approach with fragments is also used for object specific segmentation [43]. These methods have generally focused on the figure-ground task, attempting to precisely delineate the boundaries of a single object in an image. Therefore, they are highly class specific, working for a

particular object type. A recent method extends the patch-based object knowledge to work with a wider variety of objects [90].

High-level vision

In a general sense, object recognition and scene interpretation can be viewed as labeling interesting image regions with object classes or their parts, and treating other regions as background. Different types of prior knowledge about objects or scenes are explicitly built in the labeling process, depending on the varying difficulties of tasks. First, detecting the foreground object category, such as man-made structures in natural backgrounds, has been formulated as a binary image labeling problem [39]. The local image information is combined with label context from a local neighborhood to achieve consistent labeling. Simultaneous detection of multiple objects is also treated as image labeling with multiple classes [52, 77], which explore the interactions between objects from a common scene context. Shape information, along with other image features are further incorporated into a multiple-class labeling process to better capture contexts in images [66]. In order to handle partial occlusion in rigid objects like cars, object layout information is included as an additional type of labeling context [86], so that not only regions but also object instances can be correctly labeled. Those methods require a fully labeled dataset for building their models. In [11], image caption information is utilized to learn associations between image features and keywords. The information provided by captions is considerably weaker than other labeling datasets; one would expect this to lead to less precision in the test image labels. Finally, multiclass object detection is combined with image segmentation to form a joint labeling problem, in which the multiple level interactions between regions and objects can be exploited for consistent labeling [40].

Note that those approaches mentioned above are concerned with recognition accuracy, as well as the precise boundaries of objects in images. Other approaches focus on (interest-point) image features instead of image pixels, as those features are very informative for object recognition. In particular, feature-based shape matching has been formulated as a labeling problem a few decades ago [34, 14], in which feature points from the real-world image are aligned with prototype shape templates. Soft constraints on the configuration of feature points are imposed during the matching. Recent work assigns object part labels to interest-point features, such as SIFT [48], and recognize a subset of features in each image as an object class [57]. The labeling incorporates a prior on the configurations of objects in terms of their parts. Feature-based scene labeling with multiple objects inside has also been discussed in [71].

Their method captures the co-occurrence of object parts and objects in a scene and includes geometry information about relative position between them at multiple levels.

2.2 Probabilistic Approaches in Image Labeling

2.2.1 From Non-probabilistic to Probabilistic

In formulating vision tasks as image labeling problems, we see that a key issue is to incorporate various context information in the labeling process, as local evidence is often ambiguous in image interpretation. The importance of context has been recognized by researchers in early work on image labeling and recognition [34]; several important context representations were used in their research. In [74], the relational constraints for adjacent regions were incorporated into a knowledge-guided segmentation system. Hanson and Riseman [24] integrate the top-down information about the 3D world and object shapes with bottom-up information cues in their VISIONS system. A hierarchical representation of the 3D world knowledge and the scene structure in images are encoded in a rule-based expert system and used to facilitate the scene interpretation. In [44], scene labeling is also integrated with region segmentation with a three-level hierarchical system that incorporates high-level knowledge into low- and intermediate-level processing. The major limitation of these approaches is that they have many parameters in their system, which are tailored to a specific application by designers, and it is not easy to adapt the systems to new environments.

Recently, the advancement in probabilistic graphical models, such as Bayesian Networks [53] and Random Fields [32], provides a more flexible yet consistent framework for incorporating context information for image labeling. Treating the image components as random variables, the probabilistic approaches model the interactions between these components using parameterized probability families. The dependency of each label on its context is represented by the dependency between those random variables, which can be described succinctly by a graph structure. As different graph structures provide differing capacities with respect to modeling context, we will discuss those models by grouping them based on their model structure.

2.2.2 Context Modeling in Probabilistic Approaches

Models with Short-range Interactions

One basic way of labeling an image is to use a statistical classifier based on information at a local level only [37]. However, no matter how good the classifier technique is, how much training data we have or how large the image patch is, the classifier's performance is limited due to the ambiguity in the local image appearance. While certain post-processing procedures are used to smooth out the outputs of a local classifier, they are usually problem-specific and based on heuristics [10].

Systematic approaches that incorporate the context information in probabilistic labeling start from Markov Random Fields (MRF) [5, 21]. In MRFs, neighboring label variables are connected to each other so that their values are not independent. By combining local pairwise interactions between variables, MRFs impose a global constraint on the label prediction, leading to more consistent labeling of an image. A typical structure of MRFs is shown in Figure 2.1.

In spite of their successes in many labeling applications, MRFs suffer from two key limitations with respect to the labeling problem. The first drawback concerns their locality. Generally, due to the complexity of inference and parameter estimation, only local relationships between neighboring nodes are incorporated into the model. This allows the model to locally smooth the assigned labels, based on very local regularities. However, labeling an image region may depend on not only the local information, but also the contexts from a wider scope. For instance, the sky information at the top part of an image may affect the labeling of its bottom part as indoor or outdoor setting. The local connectivity of MRFs makes them highly inefficient at capturing such long-range interactions. Furthermore, a local labeling will likely depend on structure at different levels of granularity in the image.

The second main drawback of MRFs lies in their generative nature. Generative models describe the way that the observed data are generated, and build the joint distribution of all observed and hidden variables. While this is a powerful modeling methodology, many labeled images are required to estimate the parameters of the model of labels *and* images. In image labeling, we are interested in estimating the posterior over labels given the observed image; even when this posterior is simple, the true underlying generative model may be quite complex. Because we are only interested in the distribution of labels given images, devoting model resources and degrees-of-freedom to the generative image model is unnecessary.

A very different *non-generative* approach is to directly model the conditional probability of

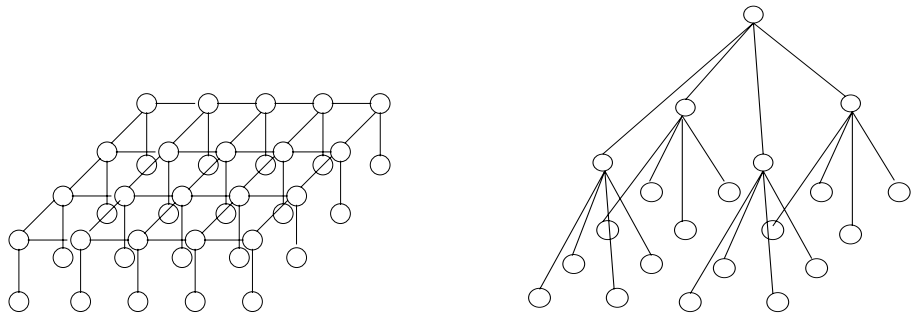

Figure 2.1: Left: The graphical model of 2 dimensional random field on a lattice; Right: The Tree-structured Random Field.

labels given images: fewer labeled images will be required, and the resources will be directly relevant to the task of inferring labels. This is the key idea underlying the conditional random field (CRF) [42]. Originally proposed for segmenting and labeling 1-D text sequences, CRFs directly model the posterior distribution of labels as a Gibbs field. More specifically, denote the input by $\mathbf{X}$, and the labeling by $\mathbf{L}$. Assume the structure of the output label $\mathbf{L}$ can be represented by a graph, and denoted by $G = (V, E)$. Each node in the graph represents the label variable for the corresponding input elements. Let c index the *cliques* in G, then a CRF defines a conditional probability distribution with a form of log-linear model:

$$P(\mathbf{L}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \exp\{-\sum_{c} \sum_{k} \alpha_k f_{kc}(\mathbf{l}_c, \mathbf{x}_c)\} \tag{2.1}$$

where $\{f_{kc}(\cdot, \mathbf{x}_c)\}$ are called feature functions, and $Z(\mathbf{X})$ is the normalizing factor, called the partition function. The graph G and the corresponding feature functions are used to represent the interactions between labels. This conditional probability model can depend on arbitrary non-independent characteristics of the observation, unlike a generative image model which is forced to account for dependencies in the image, and therefore requires strict independence assumptions to make inference tractable.

To apply the CRF to 2-D image labeling, the linear chain graph in the original model [42] is extended to more complex graphs with tree-like or loopy structures. The loopy graphs usually introduce additional complexity in modeling and label prediction. While 1-D CRFs have an efficient inference/learning procedure based on dynamic programming, this property often does not hold for loopy graphical models. For instance, Kumar and Hebert introduce the Discriminative Random Field (DRF) [39] which is defined on a graph with a two-dimensional lattice structure. As each node is connected to its four nearest neighbors on a grid, approximate inference and learning procedures have to be employed. One main difference from the earlier

MRF approaches is the DRF includes image information in the learned pairwise compatibilities between labels of different sites. However, it shares the same locality issue as the MRFs.

Models with Long-range Interactions

Many efforts have been made to overcome the locality issue in the probabilistic models with short-range connections between label variables. A straightforward way to capturing less local context is to directly build long-range connections into model structure. For instance, MRFs with large neighborhood sizes in principle can model any type of interactions between labels. However, those high-order MRFs have more complicated graph structure, and the number of model parameters increases exponentially with the order. Therefore, inference and learning for them become very difficult. Either simple approximation or expensive computation has to be used, which is unsatisfactory for most applications [45].

The Maximum Entropy Model [92, 55] is derived from the maximum entropy principle given constraints on the expectation of some predefined feature functions. The feature functions can have arbitrary span, so it has the potential to capture long-term interactions. One weakness of MEM is that we need to know a predefined feature library, which may be hard to specify in labeling tasks.

A better approach to capturing large-scale context is to build hierarchical models. By introducing extra hidden variables, we could model the long-range interactions with relatively simple graphical models. The simplest type of model has a tree-like structure, in which the label variables are leaf nodes of the tree [8, 41]. The hidden variables in the tree models usually have the same semantics as the label variables, and the models are specified by the potential functions on the edges (See Figure 2.1). In practice, it is usually convenient to view the model as a directed model in which the top layer is the root and every layer is the parent of the layer below. Here we refer to this form of model as a Tree-Structured Random Field (TSRF). A typical TSRF model is the quad-tree model, which has four offspring for each parent node and has been widely used in domain specific image segmentation, such as satellite image processing [8] and road scene analysis [17].

The TSRFs have two advantages over the traditional MRFs. First, they provide a hierarchical model with explicit long-range interaction, capturing large-scale dependency more efficiently; secondly, we can do exact inference and learning efficiently in TSRFs. Unfortunately, TSRFs with fixed structure have un-stationary statistical property, which may cause blocky artifacts in label predictions. To handle this problem, an improved TSRF with a dynamic struc-

ture, called dynamic tree, has been proposed [1, 69]. In the dynamic tree model, the structure of the tree is also decided by a set of hidden variables that are inferred from input data. However, after introducing the additional hidden variables for structure, the underlying model does not have a tree structure any more. Thus, there is no efficient exact inference/learning algorithm for this model, and some approximation has to be made. In addition, most TSRFs involve generative models of labels and images, and hence they suffer from this second main drawback of MRFs.

An alternative way to build in long-range interaction without sacrificing the stationary property is using multiple two-level trees to form random fields. The undirected model with such structure is called a Restricted Boltzmann Machine (RBM) [27, 20, 83]. This model has a bipartite structure, in that no connection exists between hidden variables or visible variables. Therefore, the hidden variables are conditionally independent given the observation, and vice versa. While the model is intractable for exact inference/learning, its structure leads to an efficient sampling procedure for approximate approaches. Hinton et al. further extend it to Causally linked MRFs [28], in which laterally-connected MRFs are causally linked into a multilayer model. The lateral connected MRFs can be viewed as a complementary prior as the dependency from the causal links in the posterior is reduced by them. In this case, the variational learning becomes more effective because the factorial approximation used is closer to the real posterior. However, the procedure of training a network is still quite slow so that it is hard to scale up to the applications with large images.

Some hierarchial generative models focus on subsets of image regions that include interesting objects. They are mostly based on a form of latent topic model, such as Latent Dirichlet Allocation (LDA) [6]. Topic models were originally used in text modeling. Treating each document as a collection of orderless words (the so-called "bag-of-words" assumption), topic models capture the co-occurrence of words. Recently, topic models have been used for modeling objects and scenes [67, 71, 47], in which object parts or objects are viewed as "words". Sudderth et al. also extended the topic model by including a locality constraint, which is based on geometric information about the relative position between object parts. This model works under constrained viewing conditions, but is difficult to extend to arbitrary viewpoints, deformable shapes and varying appearances. Also, unlike random field models, they are used to model interest-point features of images instead of non-overlapping image regions.

The discriminative models have also been extended in several ways to capture long-range interactions. Feng et al. combine a tree-structured prior over label variables and a neural network classifier for label prediction [17], in which the modeling of image data is avoided.

In [40], a two-layer conditional random field is built for capturing both region-region and object-object interactions. It essentially integrates multiple object recognition with image segmentation, requiring an intensive sampling scheme to find good parsing of images. Our work instead aims to incorporate several levels of long-range dependency with complex patterns [25, 26].

2.3 Inference in Image Labeling

Labeling a new image requires inferring or predicting the label values given the image. In a probabilistic framework, the possible label configurations are fully described by the posterior distribution of the label variables given the input. In practice, we usually want to summarize the distribution by certain point estimators, such as its mean or mode, in order to obtain a single labeling of the input. This is a standard Bayesian decision problem. Each estimator corresponds to a loss function that penalizes the discrepancy between the estimated configuration and the "ideal" random one. Two estimators and their loss functions are widely used in the literature:

- *MAP estimate*: Maximum A Posteriori Estimate (MAP) of labeling $\mathbf{L}$ given image $\mathbf{X}$ is the mode of the posterior distribution, i.e.,

$$\mathbf{L}^* = \arg\max_{\mathbf{L}} P(\mathbf{L}|\mathbf{X}), \tag{2.2}$$

  where the loss function is the 0-1 function: $L(\mathbf{L}, \hat{\mathbf{L}}) = \delta(\mathbf{L}, \hat{\mathbf{L}})$.

- *MPM estimate*: Marginal Posterior Mode estimate is the mode of the marginal posterior distribution,

$$l_i^* = \arg\max_{l_i} P(l_i|\mathbf{X}) \quad \forall i, \tag{2.3}$$

  where the loss function is the Hamming distance: $L(\mathbf{L}, \hat{\mathbf{L}}) = |\{i : l_i \neq \hat{l}_i\}|$.

Two basic operations are involved in the computation of estimators: marginalization and maximization. For certain probabilistic models with tree-like structure, we can carry out the operations efficiently, so that the exact computation of the estimators is feasible. Approximate algorithms have to be used in other cases with more complicated graph structures since exact inference is an NP-hard problem. We will discuss three types of inference approaches as follows.

2.3.1 Deterministic Energy Minimization

For probabilistic models in image labeling, computing the MAP is essentially a combinatorial optimization problem. The negative log posterior distribution has a form of energy function; thus MAP estimation can be formulated as energy minimization in which the domain is discrete. Two approximate approaches in combinatorial optimization have been applied to minimization-based labeling: one method uses heuristic-based local search, and the other is relaxation-based.

Heuristic local search starts from an initial estimate, and searches for the local minima of the energy function. Thus, the quality of the solution found by local search is usually decided by the size of neighborhood. The neighborhood in the state space is defined with respect to certain transformations of the state configuration. The neighbors are those configurations lying within a single transformation. In the traditional Iterative Conditional Mode (ICM) approach for MRFs [45], the transformation is defined by changing a single node's value in the random field. The neighborhood induced by that transformation usually is too small to provide a local minimum of high quality.

Boykov et al. proposed an effective local search method with a large neighborhood [9]. The algorithm greedily searches for the local minima based on the current estimate until no improvement can be made. This algorithm includes two different transformations, called $\alpha-$expansion and $\alpha-\beta-$swap, generating a much larger neighborhood in the state configuration space. Each step in the search finds the local optimal transformation that gives the largest decrease of the energy. The key step of the algorithm is that it formulates the local search as a graph cut problem by appropriately adding two auxiliary terminal nodes, which can be efficiently solved in spite of the combinatorial nature of the neighborhood. It can be shown that it obtains good performance in practice and is a 2-approximation algorithm (i.e., the approximate minimal energy is not greater than 2 times of the global minimal energy) under some mild assumptions. The drawback of this deterministic search approach is that it cannot provide a confidence measure for the solution it finds, even through the model has a probabilistic interpretation.

Relaxation-based methods have been studied in the context of labeling for a few decades [61, 34]. Early work used probability to represent relaxed labeling, and aimed to minimize heuristic-based cost functions to achieve a consistent label configuration. The probability here is mainly viewed as a relaxation tool, instead of a systematic modeling method. Recent work formulates the energy minimization problem as an integer program. By relaxing the integer constraint, the

problem can be converted to a Linear Program (LP) that can be solved. The integer solution is recovered from the fractional solution of LP using randomized rounding [35]. Although this approach has a provable theoretical bound on its expected performance, it is not clear if the approximate algorithm is practically useful due to questions concerning the tightness of the bound.

2.3.2 Exact Probabilistic Inference

The general framework for inference in graphical model is the Junction Tree algorithm [32]. The general Junction Tree algorithm mainly consists of constructing a junction tree from the model and propagating probabilities across the tree. The efficiency of the Junction Tree algorithm depends on the underlying graph structure. It is well known that the algorithm is computationally practical only for graphs with small tree-width (size of clique).

The most popular exact inference algorithm is a variant of the Junction Tree algorithm, called Belief Propagation (BP) [53]. In BP, we propagate a set of messages carrying the interaction information through a tree model until they achieve consistency. The marginals or modes of the model distribution can be computed from those messages. Two key components in the BP algorithm are the way that messages are exchanged between neighboring nodes and the global protocol for the message propagation [41, 89].

2.3.3 Approximate Probabilistic Inference

For those graphical models with dense connections, the Junction Tree algorithm usually induces an intractable tree with large tree-width. For random field models, the partition functions in their distributions usually are also intractable in computation. Thus we have to resort to approximate approaches. There are two primary approaches in the literature: variational approximation and sampling-based approximation.

The general variational method introduces an approximating family of label probability distributions, which have a simpler form than the original distribution such that the inference is tractable. For instance, the approximating distribution may be defined on a subgraph of the full model. During inference, we choose a specific distribution from that approximate family to match the original distribution as close as possible. The marginals or mode of the approximate distribution are used as substitutes for the original ones.

The simplest approximate inference, called mean-field approximation, uses an approximate family with a fully factorized form [89, 1]. In general, the mean field approximation

is quite crude when the nodes in a random field fluctuate a lot around their mean values. A better approximation, leading to the so-called Loopy BP algorithm, uses a more complicated approximating family that includes pairwise marginals [89]. The resultant update equations have the same form as the BP algorithm for tree-like model. More complicated approximate families have been investigated to achieve lower approximation error, such as a tree-based re-parametrization [80, 36].

Another general approach to handle an intractable posterior distribution is to use sampling methods to estimate the posterior approximately. For graphical models in image labeling, Markov Chain Monte Carlo (MCMC) sampling, including Gibbs sampling and Metropolis-Hastings sampling, are widely used in practice. The general MCMC methods sample a distribution by constructing a Markov chain having the target distribution as its equilibrium distribution. The state of the chain after a large number of steps is used as a sample from the target distribution. In particular, Gibbs sampling [21] assumes that we can sample each label variable given its neighbors' configuration. It repeatedly scans all the variables, and each step in a cycle consists of taking a sample from the posterior distribution of a variable given the values of all others. Due to the local Markov property of graphical models, Gibbs sampling can be implemented very efficiently in many cases. Metropolis-Hastings (MH) algorithm [49] provides a more general approach to sample difficult distributions. It uses a proposal distribution to sample a candidate labeling given current configuration iteratively, and only changes the current labeling with certain acceptance probability at each iteration. After samples are collected, the distribution or its statistics can be derived from those samples.

Theoretically, the estimates provided by sampling methods converge to its real value when the number of samples approaches infinity. In practice, those methods are usually computationally expensive as many samples are needed to obtain a good estimate. Several ways have been suggested to improve the sampling efficiency, especially for graphical models with special structure. For example, block Gibbs sampling has been applied to random field models with a bipartite form [29].

2.4 Learning Probabilistic Labeling Models

One main advantage of probabilistic approaches to image labeling is that the model parameters can be learned from data automatically and systematically. This is the key difference between the model building approaches in the early labeling literature and the more recent probabilistic ones. A learning method is defined by its learning criterion (or cost function) and optimization

method. We will discuss the learning methods used in two modeling approaches: generative models and discriminative models, and organize them according to their learning criteria.

2.4.1 Learning in Generative Models

For generative models, Maximum Likelihood (ML) and its variants [16] are the main methods for estimating the model parameters. Aiming at reconstruction of the observed data distribution, this criterion essentially minimizes the KL divergence between the empirical training data distribution and the model distribution. In image labeling, a generative model describes the joint distribution of image $\mathbf{X}$ and label $\mathbf{L}$, i.e., $P(\mathbf{L}, \mathbf{X})$. It can be factorized into a prior model $P(\mathbf{L})$ and an observation model $P(\mathbf{X}|\mathbf{L})$. In addition, to make learning and inference tractable, a common assumption is that the image elements are independent given the label, that is,

$$P(\mathbf{L}, \mathbf{X}) = \prod_i P(\mathbf{x}_i|\mathbf{L})P(\mathbf{L}) \tag{2.4}$$

where i indexes the image elements. We consider two cases in the following: 1) the labels and images are fully observed; 2) the labels or other latent variables are partially observed.

Fully-observed Case

In the fully-observed case, the learning criterion of ML is the log of the joint distribution, which can be decomposed into a prior term and an observation term. As the observation model is factorized, it is usually easy to learn from data. The prior model, however, can be difficult to estimate because many graphical models used in image labeling, such as MRFs, have an intractable partition function. While in principle it can be approximated by MCMC sampling techniques [21, 27], the approximation is computationally intensive, and not very practical for image labeling. Recently, a new class of approximate learning algorithms, called Contrastive Divergence (CD), is proposed based on minimizing a contrastive divergence cost function [29, 84]. Essentially, the CD algorithm approximates the intractable expectation used in optimizing likelihood by a sample average from a truncated Markov chain. The Markov chain in CD starts from observed configurations in data and runs a few steps to obtained a sample, instead of sampling from the equilibrium distribution. Intuitively, this method minimizes the tendency that the model distorts the training data distribution. This can result in huge computational savings as the gradients must be updated repeatedly. However, the CD algorithm may lead to a biased estimate in certain circumstances [12]. An alternative simple and popular way to

tackle the partition function is to replace the joint log likelihood by a sum of local site-wise log conditional likelihood, which is called pseudo-likelihood criterion [5].

In practice, log-linear models (e.g., MRF and CRF) form the most common model family for labeling problems. The energy function of those models is a linear combination of a set of feature functions [55]. While many candidate feature functions are available, an important learning issue is to choose a small relevant subset to avoid overfitting. A popular way to handle this feature selection problem is to use a class of incremental learning algorithms based on greedy search [55, 92]. The common feature of those algorithms is that they search for a feature that locally increases the data likelihood most, and then combine it into the model by weighting this feature appropriately. In [55], the local feature search is carried out in both new weight and feature function space, and all the weights are adjusted when the new feature is combined. In [85], the optimal feature is chosen given an infinitely small step in its weight space, and only the current weight is adjusted when a new feature is merged, similar to a boosting procedure. A weakness of those approaches is that they may get stuck in local minima easily and cannot reverse any wrong selection made at a previous stage.

Partially-observed Case

When the model has hidden variables or a subset of the labeling is missing, direct maximization of data likelihood is difficult. In this situation, Expectation-Maximization (EM) is widely used to maximize the likelihood iteratively (e.g., see [41]). At each iteration, the algorithm finds the optimal lower bound of the likelihood function, which has a simpler form, and optimizes the lower bound instead. The optimal lower bound involves computing the posterior distribution of the hidden variables. In intractable cases, a commonly used approximate approach is the variational EM algorithm [32], in which the true posterior is replaced by some approximate but tractable distribution. Variational EM essentially maximizes an alterative lower bound of the data likelihood, and the most popular variational approximation is the mean field approximation (e.g., see [1]).

The generative approach provides a consistent probabilistic framework for modeling image and label. If the model is a good approximation to a real generating process, it will achieve good performance on labeling tasks. Besides, it can easily handle partially labeled data or even unlabeled data by the EM algorithm. However, the purely generative modeling approach can be difficult for image labeling in practice. First, the independence assumption in its observation model is convenient for learning and inference, but may be over-simplified for real images.

Also, an observation model generating images from labels is not easy to build, due to the complex appearance distribution of each label class. Even if the image modeling is feasible, many training examples may be required to learn the joint distribution of both image and label. Furthermore, aiming to reconstruct data, this approach does not optimize the system with respect to the goal of labeling problems, i.e., the prediction performance.

2.4.2 Learning in Discriminative Models

In discriminative models, a general assumption is that a labeled data set is available for learning. Given the training data, the goal of learning is to minimize the prediction error, or the difference between the model output and the observed prediction. We will discuss different ways to measure the difference, including the likelihood criterion and the margin-based criterion.

Maximum Likelihood-based Learning

Many discriminative models define a conditional distribution of output label given the input. Applying the Maximum Likelihood principle to the conditional distribution, we obtain a discriminative learning criterion, called Conditional Maximum Likelihood (CML) [17]. Learning based on CML shares similar issues with ML learning in the generative setting, which can be solved in the same way. For instance, the EM algorithm can be applied to the conditional models with auxiliary hidden variables [57]. The Pseudo-likelihood method is also used to handle the partition function [39]. A variant of the log-likelihood learning criterion, called per-label log-likelihood, is used when we are interested in minimizing the number of mislabeled output variables. EM may be used to optimize this cost function [33].

Given the conditional likelihood objective, many optimization techniques have been proposed to address various issues in learning discriminative models. In [42], Improved Iterative Scaling is used to construct and maximize a lower bound of the data likelihood iteratively. Although it has a closed-form update equation, the speed of its convergence is usually slow [64]. Gradient-based methods, such as the conjugate gradient method, provide a faster convergence rate for maximizing the log likelihood [82]. When the computation of the gradient is intractable, a variant of Contrastive Divergence algorithm can be applied [25].

In the case of log-linear models, incremental learning has also been investigated. Torralba et al. suggest to interleave the loopy BP algorithm with a boosting procedure. At each step, the boosting procedure not only adds a new feature function, but also approximately implements

message passing in densely connected graphs [77]. In [3], a multiclass boosting procedure is directly generalized to the conditional random field, in which an upper bound of the labeling error rate is minimized. The main issue of this method is that large dynamic range of the feature sum is hard to handle.

The Maximum Likelihood learning of discriminative models may suffer from both the overfitting and difficult model selection problems. One possible extension is the Bayesian approach, in which learning is used to estimate the posterior distribution of the model parameters. In [56], the authors suggest a Gaussian prior on the parameters of the CRF, and use Gaussian based Expected Propagation (EP) to estimate the posterior of the parameters. Although the Bayesian approach makes use of the estimated uncertainty, it is also much more expensive in computation than point estimate based approaches.

Maximal Margin-based Learning

The labeling problem can be viewed as a multiclassification problem, and the discriminative model is essentially a classifier with structural output. Using the energy function of the model as a discriminative function, several authors take a margin-based approach to minimize the bound of generalization errors [2, 73]. Different margin definitions corresponding to different loss functions have been discussed. For example, a margin related to the per label loss is used in [73].

The advantage of the margin-based approach is that the learning can be formulated as a quadratic programming problem. Also, we can introduce the kernel trick to create a set of more powerful feature functions. However, this approach results in exponentially many constraints in the optimization. In [2], an incremental approach is proposed to exploit the sparseness of the Lagrange multipliers in the solution. It can be shown that it is a coordinate ascent method converging toward the optimum if the problem is feasible. In [73], the authors treat the Lagrange multiplier as an unnormalized distribution defined on the graph of the random field, and utilize the sparse structure of the graph to reduce the constraints into an equivalent set with polynomial size. But approximations must be made if the random field has a loopy graph structure.

2.5 Summary

Image labeling has wide application in computational vision. Many image and object properties can be viewed as certain types of labels, and our task is to infer those label values from images. In image labeling, a key step is to exploit the interactions between labels within each image, as local evidence can be insufficient to determine the label value. In this chapter, we mainly reviewed the probabilistic approaches to image labeling problems. The probabilistic methods provide a consistent framework for image labeling, in which the labeling process is formulated as probabilistic inference and models are built from data automatically based on statistical learning approaches. Most importantly, the context information of labeling can be modeled based on probabilistic graphical models.

However, many probabilistic models in image labeling use over-simplified assumptions when modeling interactions between labels and image features. One main issue is that previous approaches focus on limited types of interactions, and do not provide flexible and efficient representations for modeling informative context at multiple scales. Second, many models are based on purely generative approaches, which require certain unrealistic simplifications and many resources to build the models. The third important issue concerns the amount of labeled data needed for learning the probabilistic models, especially for the discriminative approaches. In reality, collecting a large number of labeled image data is expensive and troublesome. Therefore, it is important to find a better modeling approach that can utilize other types of data, such as weakly labeled images. This thesis will address those three issues by incorporating more flexible modeling structures into a probabilistic framework.

Chapter 3

Multiscale Conditional Random Fields for Spatial Context

3.1 Introduction

This chapter aims to generalize the original CRF approach to the image labeling problem, which is considerably more complicated due to the 2-D nature of images versus the 1-D nature of text. A chief focus of our approach concerns contextual information about the labeling coming from different scales (from local to global). This presents two problems: First, how can we extract and represent the contextual information at each level? Second, how should we combine the possibly conflicting information from the different levels?

We represent the multiscale contexts as probabilistic label templates with different scopes. At the local level, a mapping from a local image patch to the label describes the local image context for each pixel; at a regional level, a set of probabilistic label templates with medium scope imposes mid-range context constraints for labeling within every small region; at the global level, another set of probabilistic label templates provides global constraints for the labeling of the whole image. Some aspects of the context representations concern the co-occurrence of objects in the image, while other aspects concern the geometric relationship between objects.

The label templates are instantiated as parameterized feature functions in the CRF framework. Those feature functions encode higher order label patterns, which typically are quite difficult to represent as they tend to use a large number of parameters. To avoid unduly increasing model complexity, our method involves capturing a subset of label patterns that are most frequent in the data. This development is analogous to that provided by latent models

for continuous data, such as factor analysis. We implement it by adding latent variables to a CRF, and each latent variable encodes a label pattern. Here the latent variables essentially allow the model to maintain a distributed representation, and to capture the most significant label contexts.

The model, which we refer to as the Multiscale Conditional Random Field (mCRF), is capable of learning the feature functions in the random field that operate at different scales of the image. In other words, we can automatically learn the context representations from the data. We adopt a discriminative learning approach, where such information is learned from a training set of labeled images, and combined in a probabilistic manner.

The remainder of this chapter is organized as follows. We present the mathematical definition of the Multiscale Conditional Random Field in the following section. Section 3.3 describes the probabilistic inference procedure for image labeling with our model. A basic learning algorithm is discussed in Section 3.4, and its incremental version is detailed in Section 3.5. Section 3.6 tests our model in real life image datasets, and compares it to other models. Some further comparison and discussion is presented in Section 3.7, and Section 3.8 summarizes this chapter.

3.2 Model Definition

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ be the observed data from an input image where S is a set of image sites to be labeled. We use the term sites to refer to elements of the label field, while pixels refer to elements of the image. The local observation $\mathbf{x}_i$ at site i is the response of a set of filters applied to the image at that site, or any other image feature related to the site. The site has an associated label l_i from a finite label set $\mathcal{L}$.

Standard CRFs [42] employ two forms of feature functions, which would be defined in a 2D image as follows: state feature functions, $f(l_i, \mathbf{X}, i)$, of the label at a site i and the observed image; and transition feature functions $f(l_i, l_j, \mathbf{X}, i)$, of the image and labels at site i and a neighboring site j in the image. We extend this to *label features*, which encode particular patterns within a subset of label variables. The label features are a form of potential function, encoding a particular constraint between the image and the labels within a region of the image.

3.2.1 Label Features

Let $\mathbf{l}_q = \{l_1, l_2, ..., l_q\}$ be a subset of label variables from a set of sites. A joint label feature defined on $\mathbf{l}_q$ has the following form:

$$f(\mathbf{l}_q, \mathbf{X}, h; \mathbf{w}) = h[\mathbf{w} \cdot \mathbf{l}_q + d(\mathbf{X}|\eta)], \tag{3.1}$$

where h is a binary hidden variable associated with the label feature. Taking values in $\{0, 1\}$, it acts as a switch for that feature. Each label variable l_i is represented by an indicator vector where the jth element of the vector is one if and only if l_i is the jth possible value in $\mathcal{L}$. The weight $\mathbf{w}$ encodes a particular label pattern to the label sites within a region. The bias term $d(\mathbf{X}|\eta)$ is a function of input X with parameter η. Notice that this label feature has the form of energy function in a restricted Boltzmann machine (RBM) [20] (see Fig. 3.1). Therefore, the hidden variables are assumed to be conditionally independent given the corresponding label variables, and vice versa, which greatly simplified inference and learning in such model.

The weight can be represented as a matrix, or *template*, with one element for every possible label (row) of each of the q components (columns). If the columns of a weight matrix are softmaxed (exponentiated and normalized), such a feature can be thought of as inducing a distribution over each label variable. A high entropy distribution indicates that the feature does not care what the label is since the probabilities for each value are roughly equal. As the entropy decreases, the feature becomes more specific in what it predicts. Considering all q labels together, the feature can be seen as carving out a region in the space of configurations that it matches; a collection of such features performs dimensionality reduction with respect to a fully-enumerated q^{th}-order label configuration. Mathematically, a set of label features introduces a conditional distribution over the labels $\mathbf{l}_q$ and hidden variables $\{h_i\}$ through

$$P_q(\mathbf{l}_q, \{h_i\}|\mathbf{X}) \propto \exp\left(\sum_i h_i[\mathbf{w_i} \cdot \mathbf{l}_q + d_i(\mathbf{X}|\eta_i)]\right). \tag{3.2}$$

After marginalizing the hidden variables out, we can define a conditional distribution that captures a label context with the region $\mathbf{l}_q$:

$$P_q(\mathbf{l}_q|\mathbf{X}) \propto \prod_i \left(1 + \exp[\mathbf{w_i} \cdot \mathbf{l}_q + d_i(\mathbf{X}|\eta_i)]\right). \tag{3.3}$$

Therefore, a 'marginal' label feature defined on the labels only has the following nonlinear form:

$$g(\mathbf{l}_q, \mathbf{X}; \mathbf{w}) = \log\left(1 + \exp[\mathbf{w} \cdot \mathbf{l}_q + d(\mathbf{X}|\eta)]\right). \tag{3.4}$$

We will use these types of label features as building blocks for our model.

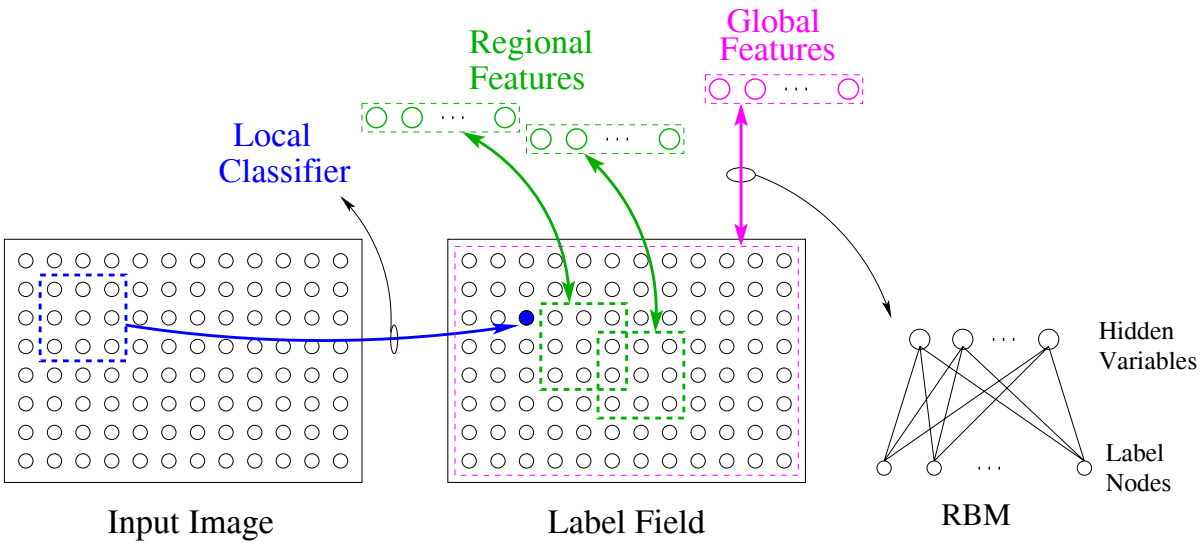

Figure 3.1: Graphical model representation. The local classifier maps image regions to label variables, while the hidden variables corresponding to regional and global features form an undirected model with the label variables. Note that features and labels are fully interconnected, with no intra-layer connections (restricted Boltzmann machine).

3.2.2 Multiscale Conditional Random Field

Our multiscale conditional random field defines a conditional distribution over the label field $\mathbf{L} = \{l_i\}_{i \in S}$ given input image $\mathbf{X}$ by multiplicatively combining component conditional distributions that capture statistical structure at different spatial scales s:

$$P(\mathbf{L}|\mathbf{X}) = \frac{1}{Z} \prod_s P_s(\mathbf{L}|\mathbf{X}) \tag{3.5}$$

where $Z = \sum_L \prod_s P_s(\mathbf{L}|\mathbf{X})$ is a normalization factor (summed over all labelings). An mCRF is therefore a conditional form of the product-of-experts model [29]: for a given site, there are multiple predictors of its label conditioned on the image. As in a standard CRF, each component conditional distribution in our model is introduced by a type of feature functions that operate at a particular scale in the label field. The product form of this combination has two chief effects on the system. First, label features need not specify the label of every site within the region. If a feature has uniform values for each possible label, it will play no role in the combination. We call this a "don't care" prediction. This enables a feature to focus its prediction on particular sites in the region. Second, the label of a site may be sharper than any of the component distributions. If two multinomials favor a particular value, then their

product will be more sharply peaked on that value. Hence unconfident predictions that agree can produce a confident labeling.

In this thesis, we instantiate the mCRF framework with three separate components, operating at three different scales s: a local classifier, regional features, and global features, as shown in Fig. 3.1.

1. Local Classifier

One powerful way of predicting the label of a pixel using information at a local level only is to use a statistical classifier, such as a neural network. Independently at each site i, the classifier produces a distribution over label variable l_i given filter outputs $\mathbf{x}_i$ within an image patch centered on pixel i:

$$P_{\mathcal{C}}(\mathbf{L}|\mathbf{X}, \boldsymbol{\lambda}) = \prod_i P_{\mathcal{C}}(l_i|\mathbf{x}_i, \boldsymbol{\lambda}) \tag{3.6}$$

where $\boldsymbol{\lambda}$ are the classifier parameters. The size of the input patches can vary: while a larger image patch provides more information, it is limited by the complexity of the classifier and the number of training data. In practice, we usually use image patches with a few pixel width as the inputs. We refer to this classifier as a local classifier, and use a multilayer perceptron as its implementation. Any probabilistic classifier can be used here. Note that different label classes may share very similar intensity patterns within an image patch due to similar appearance of objects, lighting conditions, or other image noises. Therefore, the classifier's performance is limited by this class overlap, no matter how large the patches are [37].

2. Regional Label Features

This second component, using the label features, is intended to represent local geometric relationships between objects, such as edges, corners or T-junctions. Note that these are more than edge detectors: they specify the actual objects involved, thus impossible combinations such as a ground-above-sky border can be avoided. Each feature in this component is defined on a small region of the label field, so these are called regional label features. We achieve a degree of translation invariance in the regional features by dividing the label field for the whole image into overlapping regions of the same size, on which these features are defined. The feature for a given region has its own hidden variables but shares the weights with other regions.

Specifically, let r index the regions, a index the different regional features within each region, and $j = \{1, \ldots, J\}$ index the label nodes (sites) within region r. The parameter $w_{a,j}$ connecting hidden regional variable $f_{r,a}$ and label node $l_{r,j}$ specifies preferences for the possible

label value of $l_{r,j}$. So $w_{a,j}$ is represented as a vector with $|\mathcal{L}|$ elements. We also represent the label variable $l_{r,j}$ as a vector with $|\mathcal{L}|$ elements, in which the vth element is 1 and the other is 0 when $l_{r,j} = v$. Thus, the probabilistic model describing regional label features has the following joint distribution:

$$P_{\mathcal{R}}(\mathbf{L}, \mathbf{f}) \propto \exp\left\{\sum_{r,a} f_{r,a} \mathbf{w}_a^T \mathbf{l}_r\right\} \tag{3.7}$$

where $\mathbf{f} = \{f_{r,a}\}$ represents all the binary hidden regional variables, $\mathbf{w}_a = [w_{a,1}, \ldots, w_{a,J}, \alpha_a]$, $\mathbf{l}_r = [l_{r,1}, \ldots, l_{r,J}, 1]$, and α_a is a bias term. Here the sites i are indexed by (r, j), because site i corresponds to a different node j in region r based on the position of that region in the image.

Intuitively, the most probable configuration of each feature is either the label pattern $\mathbf{l}_r$ in region r matching $\mathbf{w}_a$ and $f_{r,a} = 1$, or the label pattern $\mathbf{l}_r$ does not match $\mathbf{w}_a$ and $f_{r,a} = 0$. Given the hidden regional variables, the label variables are conditionally independent and the distribution of each label node can be written as

$$P_{\mathcal{R}}(l_i = v|\mathbf{f}) = \frac{\exp[\sum_{a,(r,j)=i} f_{r,a} w_{a,j,v}]}{\sum_{v'} \exp[\sum_{a,(r,j)=i} f_{r,a} w_{a,j,v'}]} \tag{3.8}$$

where the site is indexed by i and the summation ranges over all features defined on regions that contain i. Thus, the features specify a multinomial distribution over the label of each site. Finally, the regional component of our model is formed by marginalizing out the hidden variables in this sub-model:

$$P_{\mathcal{R}}(\mathbf{L}) \propto \prod_{r,a} [1 + \exp(\mathbf{w}_a^T \mathbf{l}_r)]. \tag{3.9}$$

3. Global Label Features

Each *global feature* is also a label feature but its domain is the label field for the whole image (though in principle we could use smaller fields anchored at specific locations, as in Fig. 3.2). In addition, these global features have a coarser resolution than the regional label features. Let b index the global label patterns encoded in the parameters $\{\mathbf{u}_b\}$ and $\mathbf{g} = \{g_b\}$ be the binary hidden global variables. In order to encourage these variables to represent coarse aspects of the label field, we divide the label field into non-overlapping patches $p_m, m \in \{1, ..., M\}$, and for each hidden global variable g_b, its connections with the label nodes within patch p_m are assigned a single parameter vector u_{b,p_m}. These tied parameters effectively specify the same distribution for each label node within the patch (and reduce the number of free parameters). Like the regional component, the global label feature model has a joint distribution

$$P_{\mathcal{G}}(\mathbf{L}, \mathbf{g}) \propto \exp\left\{\sum_b g_b \mathbf{u}_b^T \mathbf{L}\right\}. \tag{3.10}$$

The global features also specify a multinomial distribution over each label node by their parameters. Note that a global feature, as well as a regional feature, can specify that it effectively "doesn't care" about the label of a given node or patch of nodes p, if its parameters $u_{bp}(v), v = 1, \ldots, |\mathcal{L}|$ are equal across label values v. This enables a feature to be sparse, and focus on labels in particular regions, allowing other features to determine the other labels. The joint model is marginalized to obtain the global feature component:

$$P_{\mathcal{G}}(\mathbf{L}) \propto \prod_b [1 + \exp(\mathbf{u}_b^T \mathbf{L})]. \tag{3.11}$$

4. Combining the Components

The multiplicatively combined probability distribution over the label field has a simple closed form (see Eqn. 3.5):

$$P(\mathbf{L}|\mathbf{X};\boldsymbol{\theta}) = \frac{1}{Z} \prod_i P_{\mathcal{C}}^{\gamma}(l_i|\mathbf{x}_i, \boldsymbol{\lambda}) \times \prod_{r,a} [1 + \exp(\mathbf{w}_a^T \mathbf{l}_r)] \times \prod_b [1 + \exp(\mathbf{u}_b^T \mathbf{L})] \tag{3.12}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\lambda}, \{\mathbf{w}_a\}, \{\mathbf{u}_b\}, \gamma\}$ is the set of parameters in the model. We include a tradeoff parameter γ because the classifier can be learned before the other components, and the model needs to modulate the effect of over-confident incorrect classifier outputs. While including the hidden variables, we can write the joint distribution as

$$P(\mathbf{L}, \mathbf{f}, \mathbf{g}|\mathbf{X};\boldsymbol{\theta}) = \frac{1}{Z'} \prod_i P_{\mathcal{C}}^{\gamma}(l_i|\mathbf{x}_i, \boldsymbol{\lambda}) \times \exp\left(\sum_{r,a} f_{r,a} \mathbf{w}_a^T \mathbf{l}_r + \sum_b g_b \mathbf{u}_b^T \mathbf{L} \right). \tag{3.13}$$

A pictorial example of our implementation is given in Fig. 3.2.

Equation 3.12 shows that the model forms redundant representations of the label field. A key attribute of our model, as in boosting and other expert combination approaches, is complementarity: each component should learn to focus on aspects modeled less well by others. This characteristic comes from the multiplicative nature of the model. Unlike the additive models, in which the data likelihood can be increased by adding components with uniform-like distribution, the product model will treat those components as "don't care", and their contribution will be ignored after normalization. Therefore, as long as other weak components are not confidently wrong, any component in our model will be specialized at the regions in data space with low data likelihood. This can also be seen from the learning rules (See Section 3.4). Also, the labeling of an image must maximally satisfy all relevant predictions (the classifier's and the features') at every site. In particular, we expect the global and regional features to help disambiguate (or even override) the local classifier's judgment.

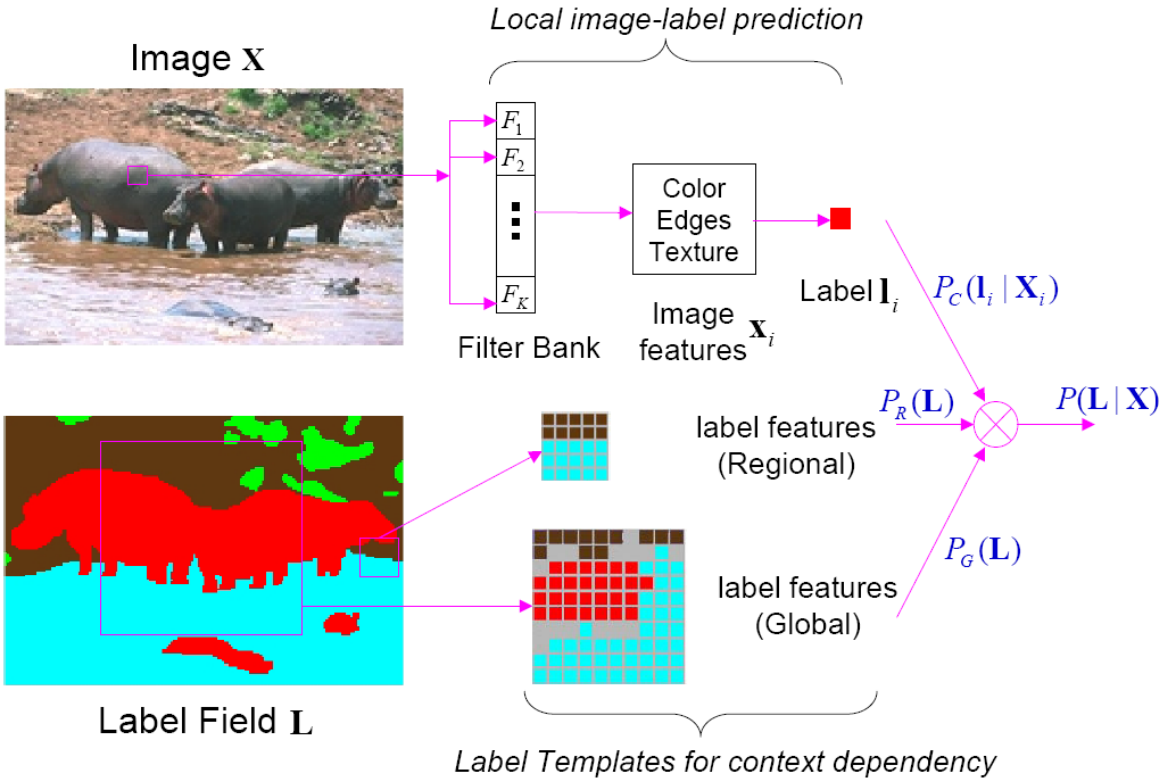

Figure 3.2: Pictorial example of the implementation of our model. A local classifier output, a regional label template, and a global label template are shown by their most probable configurations. For color coding of labels, see Figure 3.6.

3.3 Inference for Image Labeling

To label a new image $\mathbf{X}$, we need to infer the optimal label configuration $\mathbf{L}$ given $\mathbf{X}$. There are two main criteria for inferring label configurations from the posterior distribution [8]: maximum a posteriori (MAP) and maximum posterior marginals (MPM). Exact MAP is difficult to compute due to the high dimensionality and discrete domain of $\mathbf{L}$. Also, it can be too conservative in searching approximate solutions because it only considers the most probable case and disregards the difference between other solutions. The MPM criterion, which minimizes the expected number of the mislabeled sites by taking the modes of posterior marginals:

$$l_i^* = \arg\max_{l_i} P(l_i|\mathbf{X}), \qquad \forall i \in S$$

usually produces a better solution. In this paper, we adopt MPM as the criterion of image labeling.

Evaluating $P(l_i|\mathbf{X})$ in our model is intractable due to its loopy structure, so we must resort to approximate inference methods. We use Gibbs sampling due to its simplicity and fast convergence. Each step in a sampling cycle consists of taking a sample from the posterior distribution of a variable given the values of all others [50]. The bipartite nature of the joint model (see Figure 3.1) leads to an efficient implementation of Gibbs sampling. The label variables can be sampled in parallel because they are conditionally independent given the latent variables. Similarly, the latent variables are conditionally independent given the label variables and can be sampled in parallel. Besides, each hidden variable is connected to many label variables rather than just local neighbors, and vice versa, so changing the value of one variable can potentially affect a large neighborhood of a variable. Therefore, the Gibbs sampler can explore the state space more efficiently than in a simple standard lattice graph.

The posterior probability of a label variable given the latent variables $\mathbf{f}$ and $\mathbf{g}$ is

$$P(l_i = v|\mathbf{f}, \mathbf{g}, \mathbf{X}) \propto \exp\left(\sum_{(r,j)=i} w_{a,j,v} f_{r,a} + \sum_{b, i \in p_m} u_{b,p_m,v} g_b\right) P_{\mathcal{C}}^{\gamma}(l_i = v|\mathbf{x}_i). \tag{3.14}$$

where the summation $\sum_{(r,j)=i}$ is over regions r that reference y_i at index j. The posterior probability of a latent variable given labels depends on the possible type of the latent node. For the latent variables in regional features and global features, we have the following posterior distributions, respectively:

$$P(f_{r,a} = 1|\mathbf{L}) \propto \exp\left(\mathbf{w}_a^T \mathbf{l}_r\right) = \sigma(\mathbf{w}_a^T \mathbf{l}_r), \tag{3.15}$$

$$P(g_b = 1|\mathbf{L}) \propto \exp\left(\mathbf{u}_b^T \mathbf{L}\right) = \sigma(\mathbf{u}_b^T \mathbf{L}). \tag{3.16}$$

where $\sigma(x) = 1/(1+e^{-x})$. Other methods of inference, such as variational techniques or faster sampling techniques like Swendsen-Wang [72], could also be used to compute marginals. Note that we can take advantage of our architecture to start sampling the chain in a reasonable initial point, given by the label distribution output by the classifier.

3.4 Parameter Estimation

For estimating the parameters $\boldsymbol{\theta}$, we assume a set of labeled images $D = \{(\mathbf{L}^t, \mathbf{X}^t), t = 1, ..., N\}$ is available. We train the conditional model discriminatively based on the Conditional Maximum Likelihood (CML) criterion, which maximizes the log conditional likelihood:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_t \log P(\mathbf{L}^t|\mathbf{X}^t; \boldsymbol{\theta}). \tag{3.17}$$

A gradient-based algorithm can be applied to maximize the conditional log likelihood. Calling g_s the unnormalized $P_s(\mathbf{L}|\mathbf{X})$, we obtain the following learning rule:

$$\Delta\boldsymbol{\theta}_s \propto \left\langle \frac{\partial \log g_s}{\partial \boldsymbol{\theta}_s} \right\rangle_{P_0(\mathbf{L}|\mathbf{X})} - \left\langle \frac{\partial \log g_s}{\partial \boldsymbol{\theta}_s} \right\rangle_{P_\theta(\mathbf{L}|\mathbf{X})} \tag{3.18}$$

where $\boldsymbol{\theta}_s$ are the parameters in component P_s, $P_0(\mathbf{L}|\mathbf{X})$ is the data distribution defined by D, and $P_{\boldsymbol{\theta}}(\mathbf{L}|\mathbf{X})$ is the model distribution. However, we need to calculate expectations under the model distribution, which is difficult due to the normalization factor Z. One possible approach is to approximate the expectations by Markov chain Monte Carlo (MCMC) sampling, but this requires extensive computation and the estimated gradients tend to be very noisy.

In this paper, we apply the contrastive divergence (CD) algorithm [29]. CD is an approximate learning method that overcomes the difficulty of computing expectations under the model distribution. The key benefit of applying CD to learning parameters in a random field is that rather than requiring convergence to equilibrium, such as in MCMC, one only needs to take a few steps in the Markov chain to approximate the gradients, which can be a huge savings, particularly during learning when the gradients must be updated repeatedly. In addition, because our model is a form of additive random field, a block Gibbs sampling chain can be implemented efficiently, simply computing the conditional probabilities of the feature sets $\mathbf{f}$ and $\mathbf{g}$ given $\mathbf{L}$ and vice versa. The original CD algorithm optimizes the parameters of a model by approximately maximizing data likelihood; we extend it here to the objective of maximizing conditional likelihood, and dub it as the *Supervised Contrastive Divergence* algorithm.

To be specific, denote the log conditional likelihood as $\mathcal{L}$, and assume a single data pair $(\mathbf{L}^t, \mathbf{X}^t)$ for simplicity. The gradient of the likelihood with respect to the parameters of our model can be written as follows:

$$\frac{\partial \mathcal{L}}{\partial w_{a,j,v}} = \sum_{r,j \in r \& (r,j)=i} P(f_{r,a} = 1|\mathbf{L}^t) \langle l_{i,v} \rangle_{P_0(l_i|\mathbf{X}^t)} - \langle P(f_{r,a} = 1|\mathbf{L}) l_{i,v} \rangle_{P(\mathbf{L}|\mathbf{X}^t)} \tag{3.19}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_a} = \sum_r P(f_{r,a} = 1|\mathbf{L}^t) - \langle P(f_{r,a} = 1|\mathbf{L}) \rangle_{P(\mathbf{L}|X^t)} \tag{3.20}$$

$$\frac{\partial \mathcal{L}}{\partial u_{b,p,v}} = \sum_{i \in p} P(g_b = 1|\mathbf{L}^t) \langle l_{i,v} \rangle_{P_0(l_i|\mathbf{X}^t)} - \langle P(g_b = 1|\mathbf{L}) l_{i,v} \rangle_{P(\mathbf{L}|\mathbf{X}^t)} \tag{3.21}$$

$$\frac{\partial \mathcal{L}}{\partial \beta_b} = P(g_b = 1|\mathbf{L}^t) - \langle P(g_b = 1|\mathbf{L}) \rangle_{P(\mathbf{L}|\mathbf{X}^t)} \tag{3.22}$$

Here $P_0(l_i|\mathbf{X}^t)$ is a delta function centered at the observation l_i^t. So the expectation in the first term is just assigning the observed values of labels to the variables. The conditional probabilities of any hidden variable given L can be directly computed from the distribution defined in Equation 3.16. We can see from those equations that if the model predictions cannot match the observed labels well, the gradients for the parameters of active features will be large, driving those features to capture the data that are not well modeled by other components.

In the supervised CD algorithm, we replace the model distribution in the second expectation by a reconstructed conditional label distribution. First, we collect label samples after a few steps of the Gibbs sampling that starts from the true label observation $\mathbf{L}^t$. Then the reconstructed distribution of each label variable can be computed by normalizing the histogram of label samples. Usually we take only one label sample for each data point, and the expectation in the second term will be just assigning the sampled value of labels to the variables. To compute the total gradients, we average the gradients above over the whole training data set. Note that the data likelihood of the random field is hard to compute during training, so we monitor the average log of marginal probabilities of individual labels to decide when to stop the training process. Specifically, we stop the training when the average change of the monitored statistics in a few steps is less than some pre-specified threshold.

3.5 Incremental Feature Learning

The complexity of mCRF models depends on the size of feature patterns and the number of features in the model. While the feature size can be viewed as a 'knob' providing flexibility to the model designer, it is desirable to determine automatically how many features should

be included during learning. In general, without much domain knowledge, methods such as (cross-)validation are used to aid this decision, but they are computationally expensive.

We notice that the label distribution (Equation 3.12) can be written in the following form:

$$P(\mathbf{L}|\mathbf{X};\boldsymbol{\theta}) = \frac{1}{Z}P_{\mathcal{C}}^{\gamma}(\mathbf{L}|\mathbf{X},\boldsymbol{\lambda})\exp\left(\sum_{r,a}\log[1+\exp(\mathbf{w}_a^T\mathbf{l}_r)] + \sum_{b}\log[1+\exp(\mathbf{u}_b^T\mathbf{L})]\right) \tag{3.23}$$

Let n index all the label features, and g_n be the n^{th} 'marginal' label feature as in Equation 3.4, the exponential part of the distribution is an additive function of g_n:

$$F(\mathbf{L}|\mathbf{X}) = \sum_n g_n(\mathbf{L}|\mathbf{X},\theta_n), \tag{3.24}$$

where θ_n denotes the parameter inside the function g_n. Therefore, we choose to start from the local classifier and induce the label features in a forward step-wise fashion. To be specific, the log likelihood can be viewed as a functional $\mathcal{L}$ of the additive function $F(\mathbf{L}|\mathbf{X})$. We adopt a functional gradient ascent method, adding a new g into F at each step to maximize the log likelihood. Assuming $k-1$ label features have been induced, at the kth step, we add a new $\hat{g}$ such that

$$\hat{g} = \max_g \mathcal{L}(F_{k-1}+g). \tag{3.25}$$

We impose two constraints on g: first, g has the same functional form as the family of g_n, which is a nonlinear function with respect to the parameters in our case; second, the parameter θ in g has a bounded norm, i.e., $||\theta|| \leq C$, so that adding each g will change F only slightly. Under these assumptions, the cost function in Equation 3.25 can be approximated by its first order expansion:

$$\mathcal{L}(F_{k-1}+g) \approx \mathcal{L}(F_{k-1}) + < \nabla\mathcal{L}(F_{k-1}), g > \tag{3.26}$$

and the optimal $\hat{g}$ at step k can be written as $g(\mathbf{L}|\mathbf{X},\hat{\theta}_k)$, where

$$\hat{\theta}_k = \arg\max_{\theta_k} < \nabla\mathcal{L}(F_{k-1}), g(\mathbf{L}|\mathbf{X},\theta_k) > \tag{3.27}$$

For clarity, we assume one training data pair $\{\mathbf{X}^t,\mathbf{L}^t\}$, then $\mathcal{L} = \log p(\mathbf{L}^t|\mathbf{X}^t)$. The functional gradient can be written as

$$< \nabla\mathcal{L}(F), g(\mathbf{L}^t|\mathbf{X}^t,\theta_k) > = g(\mathbf{L}^t|\mathbf{X}^t,\theta_k) - < g(\mathbf{L}|\mathbf{X}^t,\theta_k) >_{p_{F_{k-1}}(\mathbf{L}|\mathbf{X}^t)} \tag{3.28}$$

where $p_{F_{k-1}}(\mathbf{L}|\mathbf{X}^t)$ is the model probability with F_{k-1} as its exponential part. The functional gradient is a nonlinear function of the parameter θ_k only, so that we can use gradient-based

method to search $\hat{\theta}_k$ in Equation 3.27. The detailed gradient for each parameter has the same form as in Equation 3.19, 3.20,3.21, and 3.22.

Notice that each induction step in the standard functional gradient method, as in [55, 85], requires first searching the direction in feature space that maximizes data likelihood, followed by a line search to determine the stepsize in that direction. Usually, both steps in the induction involve expensive Markov Chain Monte Carlo (MCMC) sampling of the random field [92, 85]. In [85], the first step is approximated by the Contrastive Divergence (CD) algorithm, and a re-weighting scheme in the second, which requires careful monitoring of the effective sample size and an approximation of the feature functions.

In our functional gradient approach, we also use the CD algorithm to compute the new cost in Equation 3.27 approximately, but we use a fixed stepsize. Therefore, our approach can be viewed as a simpler and faster unweighted version of the induction procedure of [85], in which each induced expert is optimized directly given a bound on the norm of its weights, and always a one-unit stepsize. As no line search is involved, the approximation of feature functions in facilitating the MCMC sampling and the re-weighting scheme are avoided. The modified procedure is equivalent to boosting with a fixed step size in functional gradient ascent. Adopting a fixed step size avoids the line search, but it does not allow the contribution of each feature function to be weighted. However, one would expect the need to re-sample after each round of induction, and the use of small search steps might mitigate any advantage of weighting the features.

This feature induction approach can be viewed as learning a second form of structure in CRFs. Whereas the parametrized features allow the system to determine an appropriate basis in which to represent regularities in the label/observation patterns, induction allows the system to find an appropriate number of bases for the given dataset.

3.6 Experimental Evaluations

3.6.1 Data Sets

We applied our mCRF to two natural image datasets. The first dataset is a 100-image subset of the Corel image database, consisting of African and Arctic wildlife natural scenes. We labeled them manually into 7 classes: 'rhino/hippo', 'polar bear', 'vegetation', 'sky', 'water', 'snow' and 'ground'. The training set includes 60 randomly selected images and the remaining 40 for testing; each image is 180×120 pixels.

The second dataset, the Sowerby Image Database of British Aerospace, is a set of color images of outdoor scenes and their associated labels. The images contain many typical objects near roads in rural and suburban areas. After preprocessing the images as in [17], we obtain 104 images with 8 labels: 'sky', 'vegetation', 'road marking', 'road surface', 'building', 'street objects', 'cars' and 'unlabeled'. We down-sample the original images (768×512) to a lower resolution 96×64 to keep the model complexity manageable, and assign a label to each new 'pixel' using the majority label value of the corresponding patch in the original ground-truth. During testing, we do not consider the unlabeled sites and the model's output for them. We randomly select 60 images as training data and use the remaining 44 for testing.

We extract a set of image statistics $\mathbf{x}_i$ at each image site i, including color, edge and texture information. In these experiments, each site corresponds to a single image pixel. For the color information, we transform the RGB values into CIE Lab* color space, which is perceptually uniform. More specifically, we assume that images are encoded in the sRGB format, and the white point is CIE standard illuminant D50. Suppose the sRGB component values Rsrgb, Gsrgb, Bsrgb are in the range 0 to 1. We first convert those into CIE XYZ space as follows,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} g(R_{srgb}) \\ g(G_{srgb}) \\ g(B_{srgb}) \end{bmatrix} \tag{3.29}$$

where

$$g(x) = \begin{cases} (\frac{x+0.055}{1+0.055})^{2.4}, & x > 0.04045 \\ \frac{x}{12.92}, & \text{otherwise} \end{cases}$$

Then they are further converted into CIE Lab* space based on the following nonlinear transform:

$$L = 116 f(Y/Y_n) - 16 \tag{3.30}$$

$$a = 500[f(X/X_n) - f(Y/Y_n)] \tag{3.31}$$

$$b = 200[f(Y/Y_n) - f(Z/Z_n)] \tag{3.32}$$

where

$$f(x) = \begin{cases} x^{1/3}, & x > 0.008856 \\ 7.787x + 16/116, & \text{otherwise} \end{cases}$$

The edge and texture features are based on the gray-level image information, and extracted by a set of filterbanks including the difference-of-Gaussian filter at 3 different scales ($\sigma =$

$\{0.5, 1, 2\}$), and quadrature pairs of oriented even- and odd-symmetric filters at 4 orientations $(0, \pi/4, \pi/2, 3\pi/4)$ and the same 3 scales. Thus each pixel is represented by 30 image statistics.

3.6.2 Baseline Models

We compare our approach with the following models:

Pixel-wise Classifier

The first baseline model is a non-linear classifier that predicts the label of each pixel independently. We use a 3-layer multilayer perceptron (MLP) with sigmoid hidden units, and $|\mathcal{L}|$ outputs with softmax activation function (so we can interpret the output as the posterior distribution over labels). For each image site, the input of the MLP is the image statistics within a local 3×3 pixel window centered at that site. Larger window sizes (e.g. 5×5) produced only small improvements in the classification rate but need much longer training. The MLP is trained to minimize the cross-entropy for multiple classes with a scaled conjugate gradient algorithm.

Generative Markov Random Field

The generative Markov Random Field we used has the following form:

$$P(\mathbf{L}, \mathbf{X}) = \prod_i P(\mathbf{x}_i|l_i)P(\mathbf{L}), \tag{3.33}$$

$$P(\mathbf{L}) \propto \exp\left\{-\sum_{i\in S}\sum_{j\in\mathcal{N}_i}\sum_{u,v}\mu_{u,v}\delta(l_i-u)\delta(l_j-v)\right\}, \tag{3.34}$$

where $\mathbf{x}_i$ is the image statistics vector at image site i. The class-conditional density $P(\mathbf{x}_i|l_i)$ is modeled by a Gaussian mixture with diagonal covariance matrices. We learn the Gaussian mixture with the EM algorithm and choose the number of mixture components using a validation set. The label field $P(\mathbf{L})$ is modeled by a homogeneous random field defined on the conventional 4-neighbor lattice. Its parameter $\mu_{u,v}$ measures the compatibility between neighboring nodes (l_i, l_j) when they take the value (u, v). We trained the random field model $P(\mathbf{L})$ using the pseudo-likelihood algorithm [45]. To infer the optimal labeling given a new image, we use the same MPM criterion used in mCRF, where the marginal distribution is calculated by the loopy belief propagation algorithm [19].

Pairwise Conditional Random Field

We build a simple CRF that uses the pixel-wise MLP classifier as its state feature function, and has pairwise state transition function between each site and its 4 nearest neighbors on the lattice of pixels. The state transition function is a set of input-independent and homogeneous feature functions as in [42]. The model is trained using the conditional pseudo-likelihood method, and the labeling approach is the same as in the generative Markov Random Field.

3.6.3 Learning the Full Model

We train the system sequentially: first we train the local classifier; then we fix the classifier and train the label features. Although potentially suboptimal with respect to a joint training of all parameters, the sequential approach is more efficient. In the CD algorithm, we always run a Markov chain for 3 steps from the correct label configuration. For the Corel dataset, the local classifier is an MLP with 50 hidden nodes, and the regional features are defined on 8×8 regions with overlap 4 in each direction, while the global features are defined on the whole label field with patch size 18×12. There are 30 regional features and 15 global features. For the Sowerby data, the local classifier has 50 hidden units and the regional features are defined on 6×4 regions overlapped by 2 horizontally and 3 vertically. The global features are defined on 8×8 patches of label sites. There are 10 global features and 20 regional features. For both mCRFs, we set the classifier weighting parameter ($\gamma = 0.9$) and the model structure—number of regional and global features, and region sizes—using a small validation set.

We evaluate the performance of our model by comparing with the generative MRF and the local classifier over the Sowerby and Corel datasets. The correct classification rates on the test sets of both datasets are shown in Table 3.1. We can see that the performance of the MLP classifier is comparable to the MRF, while our model provides a significant improvement. The result shows the advantage of discriminative over generative modeling and the weakness of local interactions captured by the MRF model. The confusion matrix for the testing results on our mCRF model is shown in Tables 3.2–3.3, where the values show the percentage of labels in the whole testing data. The tables show that the errors made by our model are consistent across the classes. For the Sowerby data, the model can sometimes generate too smooth a labeling (see below) at the cost of classes with few data (rdm, str, car), but the overall performance is comparable to the best result in published classification results on this dataset: 90.7% in [69].

Figure 3.3 shows a subset of the parameters learned, i.e., the conditional probability tables in the regional and global features. For legibility, only the most probable labels are shown for

Table 3.1: Classification rates for the models.

Database	Classifier	MRF	mCRF
Corel	66.9%	66.2%	80.0%
Sowerby	82.4%	81.8%	89.5%

Table 3.2: Confusion matrix in percentage for Corel data. Entry (row i, column j) means true label i was estimated as j. The label values are written in their abbreviations: rhino-hippo(r-h), polar bear(br), water(w), snow(sn), vegetation(vg), ground(grd) and sky(sk).

	r-h	br	w	sn	vg	grd	sk
r-h	9.27	0.14	0.53	0.01	1.01	1.00	0
br	0.08	8.06	0.01	0.52	0.12	0.63	0
w	0.33	0	12.87	0	0.42	0.76	0.05
sn	0	0.82	0	12.83	0.23	0.09	0.04
vg	0.95	0.55	0.09	3.18	15.06	2.99	0.06
grd	1.13	1.18	1.11	0.26	1.56	21.19	0
sk	0	0	0	0	0.19	0.01	0.66

each site and each feature pattern is displayed as a matrix of blocks. The color of each block represents the label value with the highest probability (cf. the key in Fig. 3.6) and the block size is proportional to the probability values. Figure 3.3 shows 5 regional features from the Sowerby data and 5 global features from the Corel data. We can see that the regional features capture within-label regularities as well as cross-label boundary regularities. For example, the first regional feature is mostly devoted to 'ground', while the fourth one represents the boundary between 'vegetation' and 'sky'. The global features capture coarser patterns in the entire label field and reflect the global context in the data. For instance, the second global feature shows the rhino or hippo is usually surrounded by vegetation and water, and the sky is above them, while the fourth one shows the bear is often surrounded by snow.

We also show the outputs of the local classifier, MRF and our model on some test images in Figure 3.6. The classifier works reasonably well but can be easily fooled since no contextual information is included. The MRF produces quite smooth label configurations but it may smooth in a wrong way because it captures only local context, which can be misleading. Our mCRF

Table 3.3: Confusion matrix in percentage for Sowerby data. The label values are written in their abbreviations: sky(sk), vegetation(vg), road map(rdm), road surface(rds), building(bd), street objects(str) and car(car).

	sk ■	vg ■	rdm	rds ■	bd ■	str ■	car ■
■	12.01	0.53	0.00	0.01	0.03	0.00	0.01
■	0.83	33.39	0.01	1.41	2.71	0.03	0.09
	0.00	0.00	0.08	0.10	0.00	0	0
■	0.01	0.94	0.02	40.33	0.10	0.01	0.05
■	0.06	2.60	0.02	0.30	3.05	0.01	0.05
■	0.02	0.25	0	0.03	0.12	0.02	0.01
■	0.02	0.27	0.00	0.09	0.24	0.00	0.14

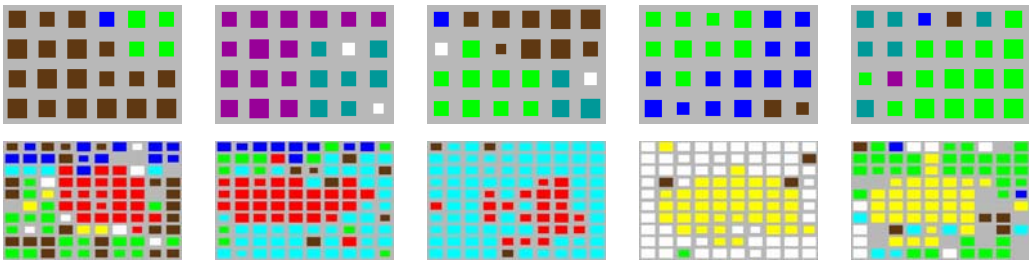

Figure 3.3: Examples of learned regional label features from the Sowerby dataset (above, 6×4 sites) and global label features on the Corel dataset (below, 10×10 blocks each of 18×12 sites). For the color key of labels, see Figure 3.6 for details.

model generates more reasonable labelings in which the contextual information provided by regional and global features corrects most of the wrong predictions from the local classifier—even when these occupy large, scattered portions in the image. We can take the probability of the winning label class for each site as a confidence measure, and form a confidence map of the labeling (see Fig. 3.6, rightmost column). This confidence measures the quality of the prediction in a consistent way: note how it tends to be low around boundaries and where the model cannot reverse the classifier's wrong labeling due to confusion by highlights or shadows. The model performance in this case could be improved by letting the label features have access to image statistics as well.

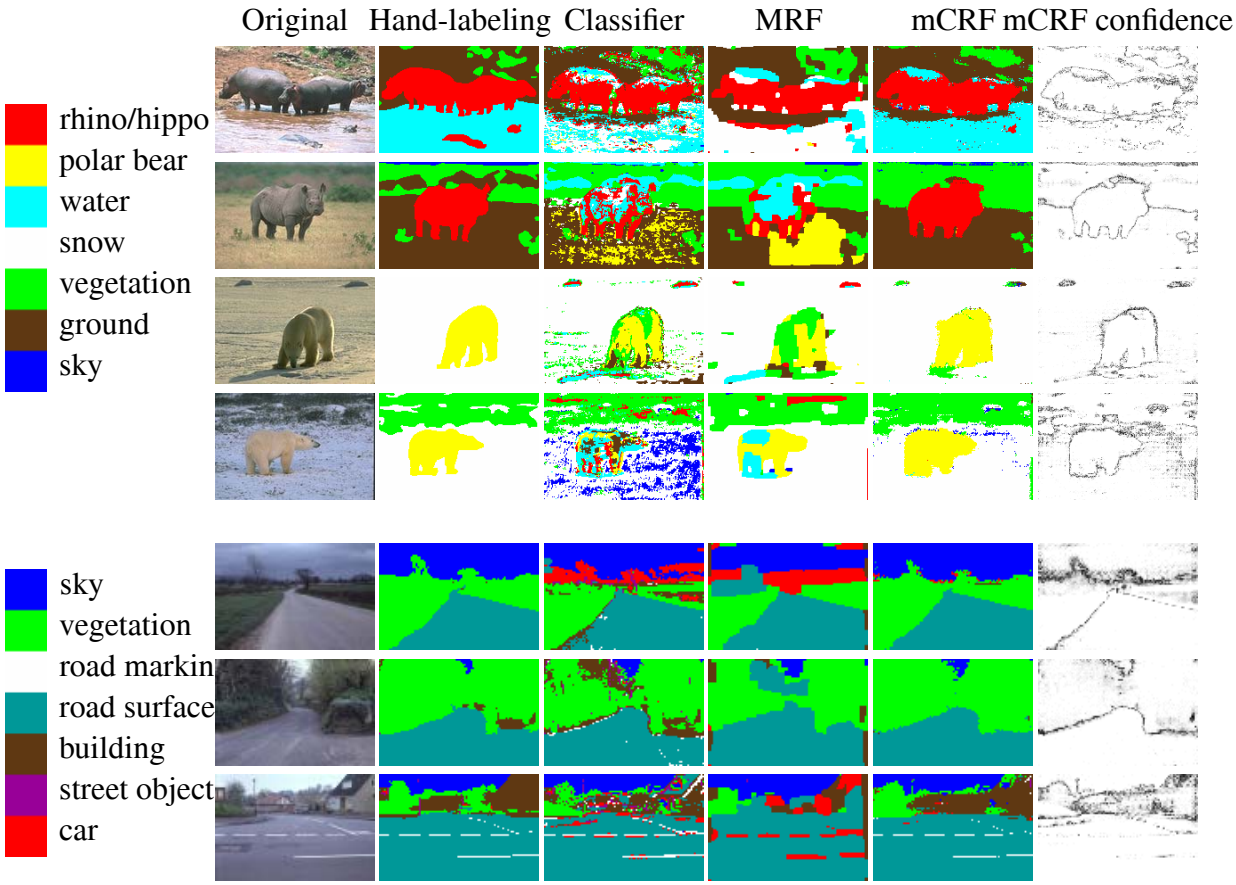

Figure 3.4: Some labeling results for the Corel (4 top rows) and Sowerby (3 bottom rows) datasets, using the classifier, MRF and mCRF models. The color keys for the labels are on the left. The mCRF confidence is low/high in the dark/bright areas.

3.6.4 Incremental Feature Learning

We investigated and evaluated the incremental feature learning on the Corel dataset. We use label features with different sizes. The regional features are defined on 6×6 regions and overlap by 3 horizontally and 3 vertically. The overlap is chosen to achieve a trade-off between coverage of the label field and the model complexity. The global features are defined on the whole label field, which is divided into non-overlapped 6×6 patches of size 20×30 pixel units. All the connections of each large feature within each patch share the same parameters. During training, we alternately induced the regional and global features for 16 epochs for a total of 32 features.

The effectiveness of the feature induction is shown in Figure 3.5, in which we evaluated the accuracy rate of the model with the first k features on the test data for different k values. The

performance increases as features are induced, and it asymptotes after several iterations. The whole training process required 13 hours with our system setup, which is slightly less than half the time required to train a full model with the same size, in which the features are trained in parallel as opposed to the sequential feature induction scheme employed here.

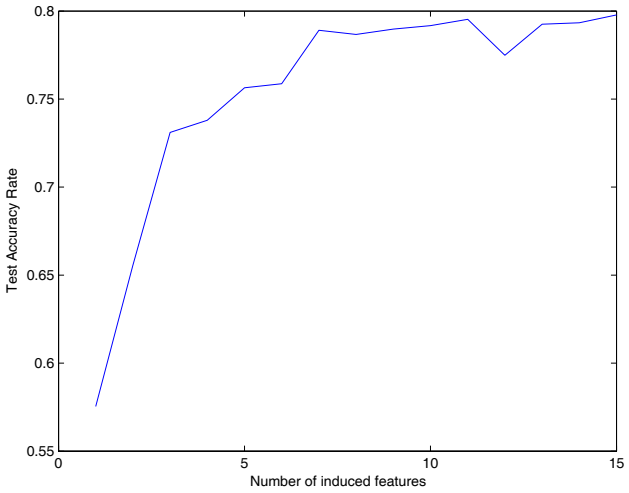

Figure 3.5: Test performance with different numbers of features.

We also trained two additional models with different feature sizes. One had 4×4 features with an overlap of 2 in each direction; the other used 12×12 features overlapping by 6 in each direction. Table 3.4 summarizes the models' testing accuracy averaged over 5 runs, which shows that the mCRF performance is not very sensitive to different feature configurations.

Table 3.4: Image labeling accuracy rates for the models with different feature sizes.

	4×4	6×6	12×12
Accuracy	78.3 ± 0.5	79.8 ± 0.5	79.0 ± 0.6

We evaluate the performance of our model by comparing with the pairwise CRF and the pixel-wise MLP classifier. The correct classification rates on the test sets are shown in Table 3.5. The results from the incremental learning are also compared to the full model learning approach in Section 3.6.3, and the TextonBoost model in [66]. In the full model learning, the number of features is tuned manually, and all the features are learned simultaneously. In the incremental learning, we only specify the maximum size of label pattern, leaving the incremental learning procedure to decide on the complexity of the feature set, based on the dataset.

Table 3.5: Image labeling accuracy rates for the different models. All results are in percent.

	MLP	MLP+MRF	mCRF_inc	mCRF_ful	TextonBoost
Accuracy	66.9	70.6	79.8 ± 0.5	80.0	74.6

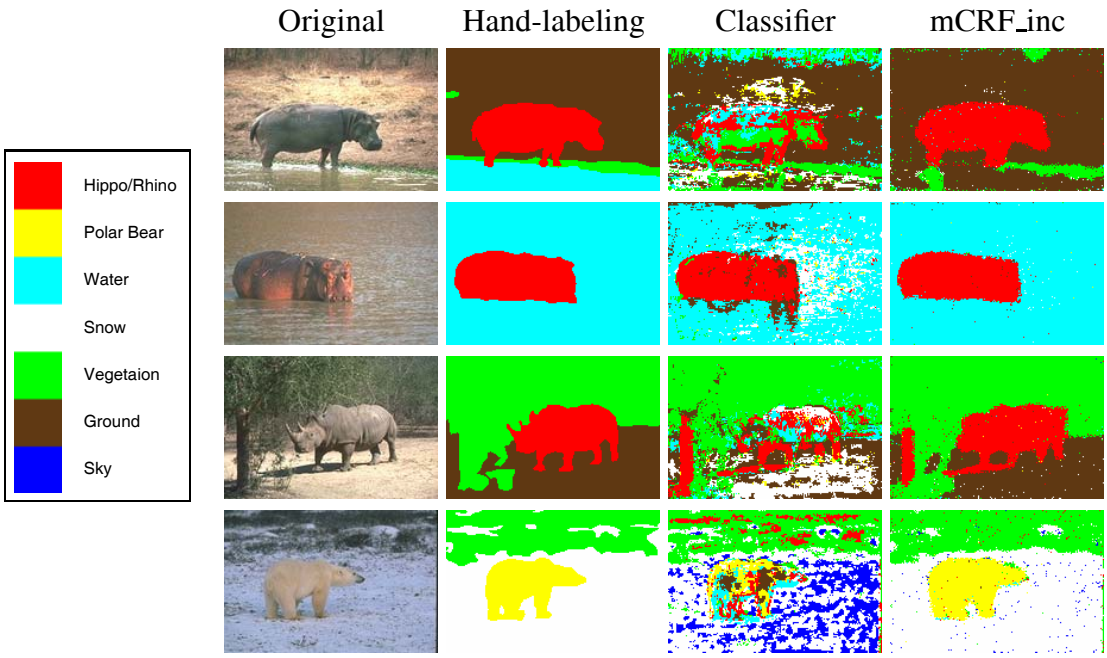

Figure 3.6: Some labeling results for the Corel dataset using the pixel-wise classifier and the mCRF model with induced features.

The TextonBoost model used a different set of bottom-up shape cues, and included pairwise interactions only.

The outputs of selected models are shown in several images in Figure 3.6. We can see that the induced mCRF model has better performance than the MLP classifier and the pairwise CRF model. Furthermore, it provides almost identical performance to the full model learning. Note that our induced model has a simpler structure with only 16 features in total. Compared to the batch training in that model, the feature induction procedure reduces the computational requirements since we only search the parameter space of a single feature function at each stage.

3.7 Discussion

The method proposed here is similar to earlier approaches to the problem of object detection, or the more general task of image labeling, in that it combines local classifiers with probabilistic

models of label relationships. Insight into these various models can be gained by comparing the solutions to the basic problem posed in the introduction: how can information at different scales be represented, learned, and combined?

A primary difference between these earlier models and our model is the form of the representation over labels. One method of capturing label relationships is through a more conceptual graphical model, such as an abstraction hierarchy consisting of scenes, objects, and features [75]. The distribution over labels can also be obtained based on *pairwise* relationships between labels at different sites [39].

An alternative to a pairwise label model is a tree-based model [17, 69]. Tree-based models have the potential to represent label relationships at different scales, corresponding to conditional probability tables at different levels in the tree. Static tree-based models are limited in their flexibility due to the fixed nature of the tree, which tends to lead to blocky labeling. The dynamic tree model [69] elegantly overcomes this approach by constructing the tree on-line for a given image; however, inference is quite complicated in this model, necessitating complicated variational techniques. Thus the CPTs learned in this model were restricted to very simple label relationships.

In our model, a wide variety of patterns of labels, at different scales, are represented by the features, and the features all interact at the label layer. The mCRF model is flatter than the trees, and the features redundantly specify label predictions. The model is therefore searching for a single labeling for a given image that maximally satisfies the constraints imposed by the active learned features. In the tree-based models, alternative hypotheses are represented as different trees, and inference considers distributions over trees. Our method instead combines the probabilistic predictions of different features at various scales using a product model, which naturally takes into account the confidence of each feature's prediction.

Another important aspect of these models concerns the learning of the features. In standard random field methods, learning involves repeatedly evaluating the utility of every possible feature in a pre-enumerated catalog, and greedily selecting a single feature to add to the model library [55]. In vision applications, particularly with features operating at different levels of resolution, it is difficult to pre-enumerate a feature catalog. Instead, we construct a set of parametrized features, and use an optimization approach to learn the parameters. As a result, the features are tailored to the statistics of the image dataset, as they specify labels in some pixels and a number of don't-care pixels.

Our model is an instantiation of a larger framework, where individual sub-models specialize on tasks and have access to particular information. Further work can consider, for example,

label features over a range of scales (rather than just local and global), or label features that have also access to some image statistics. Generative models cannot include image information as well as label patterns into learned features. We expect that this will enable the features to localize boundaries between objects in a more precise manner. Also, ideally the system we described would be applied to a higher level of input image representation, to apply to labeled image features rather than individual pixels. However, this requires a consistent and reliable method for extracting such representations from images.

Finally, automatic image labeling has several direct applications, including video surveying or object detection and tracking. A primary application is content-based image retrieval. Many current content-based query methods rely on global image properties, which do not handle searches for specific objects in a variety of scenes [10]. As the quality of image data increases, it becomes more important to have a mechanism for classifying images as fully as possible prior to insertion into a database. After learning our model on a small, representative data set, the entire database can be labeled automatically. Then, user queries such as "find images with hippos in water" can be processed very quickly. Indexes for the classes associated with each image could be generated for each image, which would allow rapid retrieval; alternatively, more specific regions of images can be retrieved based on the pixel labels.

3.8 Conclusion

This chapter presents a novel probabilistic model for labeling images into a predefined set of class labels. The model is a product combination of individual models, each providing labeling information from different aspects of the image: a classifier that looks at local image statistics; regional label features that look at local label patterns; and global label features that look at large, coarse label patterns. Both the classifier and the label features are learned from a training set of labeled images. This strategy results in consensual labelings that have to agree with the image statistics but at the same time respect geometric relationships between objects at a local and global scale. The main reasons for our model's success are its direct representation of large-scale interactions and its devoting resources to modeling the label space but not the image space. A chief novelty of the work is that we generalize the standard form of feature functions used in CRFs to use hidden variables, each encoding a learned pattern within a subset of label variables.

Chapter 4

Mixture of Conditional Random Fields for Context Integration

4.1 Introduction

Predicting a label for each pixel integrates segmentation and region classification within a single framework. However, the probabilistic model for such a task is highly redundant, as neighboring pixels are strongly correlated and usually share the same label. This chapter describes an image labeling approach that exploits the redundancy by combining the advantages of the state-of-art segmentation algorithms and the probabilistic labeling framework. Instead of making pixel-wise prediction, our approach labels a higher level image representation, which consists of coherent image patches, called super-pixels.

We adopt a random field approach to this labeling problem, learning discriminatively the statistics of the correspondence between image features and labels, as well as the interactions between labels. We further decompose the problem by assigning images to contexts, and again use learning to define the contexts, and to find features that characterize the contexts. The resulting system produces a detailed segmentation of a test image into coherent regions, with a semantic label associated with each region in the image. The key contribution of this work is a modular, adaptive segmentation method that holds the potential for scaling up to large image databases and large numbers of object categories.

From the viewpoint of segmentation, our method integrates bottom-up cues with top-down information about object categories. Shortcomings in the standard bottom-up approach to image segmentation, together with evidence from studies of human vision [54], suggest that prior knowledge about objects facilitates segmentation. However, incorporating top-down informa-

tion faces several challenges: (1) the appearance of objects in a class varies greatly in natural images; (2) shape also varies considerably, and is often corrupted by occlusion; (3) if the number of classes is large, local features may be insufficient to discriminate the class. The images in Figure 4.1 illustrate some of these difficulties.

Figure 4.1: Lighting and background effects create highly variable appearances of objects. The animal shapes also vary considerably, due to viewpoint changes, articulation, and occlusion, as shown in the hippo images. Discriminating classes based on local cues is often hard, as can be seen by comparing local patches of the two images.

We propose to incorporate more category-level rather than class-specific knowledge; the emphasis is on grouping image pixels into various categories across the whole image rather than a precise specification of a single figure-ground boundary. In our approach, bottom-up cues are used to produce an over-segmentation that is assumed to be consistent with object boundaries but breaks large objects into small patches. The problem then becomes how to group those patches into larger regions. The top-down category-based information is used to help merge those segments into object components. Formulated as an image labeling problem, it aims to assign labels to the patches so that the patches belonging to the same object category have the same labels. The labels are assigned jointly to an image, taking into account interactions between patches.

The rest of this chapter is organized as follows. In Section 4.2, we describe the new integrated approach, focusing on the architecture of our random field model. Section 4.3 presents the inference algorithm for labeling a new image. The learning procedure is detailed in Section 4.4. We compare our model with other approaches in Section 4.5. Section 4.6 summarizes this chapter.

4.2 Model Architecture

4.2.1 Super-pixel representation of images

A label algorithm operating at the pixel level will typically be highly redundant, and limited by the resolution of an image. Instead, we build our model based on a higher level image representation than the pixel image, in which a small patch of similar pixels are grouped together to form a larger unit, a *super-pixel* [87, 59]. Segmentation methods based on the bottom-up image cues can be utilized to generate such an image representation by over-segmenting the image into small but coherent regions. When the regions are small enough, their boundaries are usually consistent with the boundaries between object categories, and the potential error induced by such a decomposition will be relatively small.

In this work, we use a variant of the Normalized Cut segmentation algorithm [65], with a specific parameter setting to generate an over-segmentation of an image. The segmentation algorithm clusters the image pixels based on their color, texture and proximity cues. By varying the number of clusters and the relative weights of those cues, we can segment the image into super-pixels of a roughly consistent size, and build our approach on this super-pixel representation.

The super-pixelization of an image can be viewed as a part of the bottom-up process in our system, while the labeling model discussed in the next section uses both top-down information and image cues to merge those super-pixels into segments with semantic meanings. Figure 4.2 shows an instance of a super-pixel representation of image with varying number of super-pixels. Note that even if the size of a super-pixel is small, we significantly reduce the number of units to be labeled, which allows a compact model to be constructed without much sensitivity to the resolution of the image.

This new representation also provides a better description of input images for labeling, as we can extract image features from the super-pixels (sets of pixels). Those small image patches include more information than individual pixels. The resulting *image descriptor* of each super-pixel summarizes the statistics of the contained region with respect to image features such as texture, edges, and color. More specifically, we first apply the filterbank used in Chapter 3 to each pixel location in a super-pixel; then the histograms of their outputs are used as the descriptor of that super-pixel (See Section 4.5 for details).

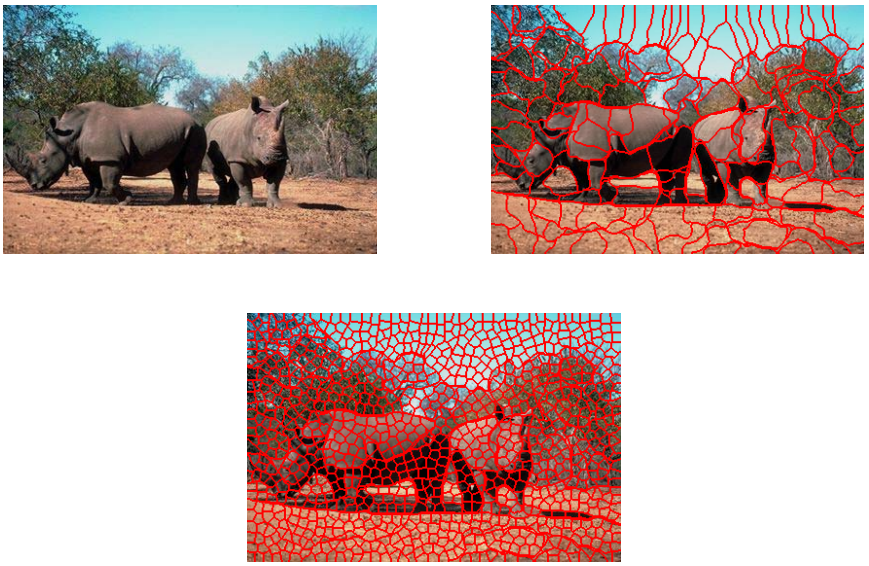

Figure 4.2: Two examples of super-pixelized images. An original image with 380x250 pixels becomes a 200 and a 1000 super-pixel image, where each contiguous region with a delineated boundary is a super-pixel.

4.2.2 A Mixture of Conditional Random Fields

Our probabilistic model assigns labels to the super-pixels for a given input image by combining top-down category-based information with image cues. First, we introduce some notation. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ be the input image, where S is a set of sites associated with the super-pixels and $\mathbf{x}_i$ is the image descriptor from the ith super-pixel. Each super-pixel $\mathbf{x}_i$ will be assigned a label $\mathbf{l}_i$ from a finite label set $\mathcal{L}$. The set of label variables $\{\mathbf{l}_i\}_{i \in S}$ for image $\mathbf{X}$ forms a structural output $\mathbf{L}$.

We further decompose the labeling problem by assigning each image to a particular *context*; several recent approaches have demonstrated that the statistics of an image can be used to categorize the scene context (e.g., [76]). Suppose the images in a database can be grouped into several contexts. We denote the context set for the images in a database as $\mathcal{C}$, and c as the context variable for input image $\mathbf{X}$. Our model defines a conditional distribution over the output $\mathbf{L}$ given input $\mathbf{X}$:

$$P(\mathbf{L}|\mathbf{X}) = \sum_{c \in \mathcal{C}} P_M(\mathbf{L}|\mathbf{X}, c) P_G(c|\mathbf{X}) \tag{4.1}$$

where $P_M(\mathbf{L}|\mathbf{X}, c)$ is a conditional random field (CRF) for the context c, which is defined in

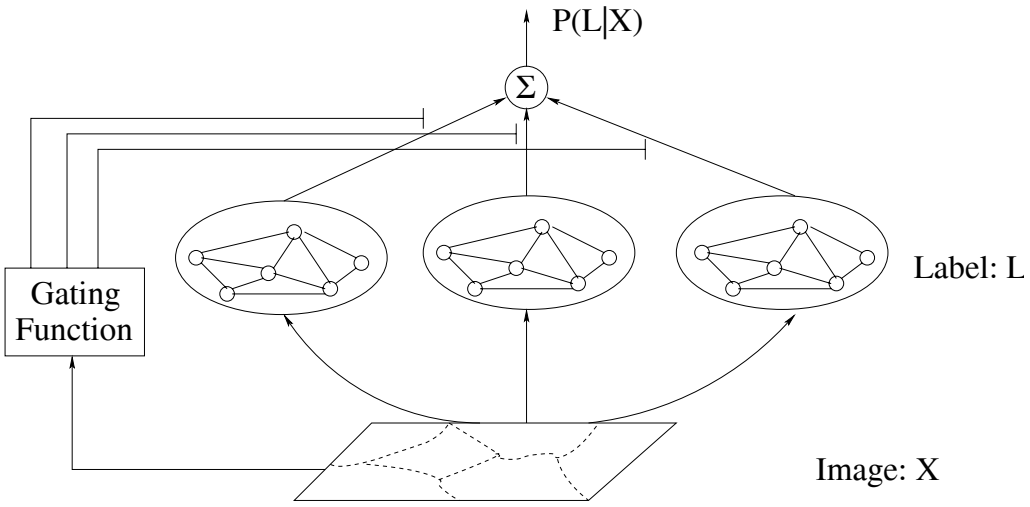

Figure 4.3: Graphical model representation of the MoCRF. The super-pixel descriptors are input to context-specific processing, with the gating function modulating the relevance of each context to a given image.

the Section 4.2.3 below. The probability $P_G(c|\mathbf{X})$ is a gating function which yields the probability distribution of context given the information from image $\mathbf{X}$. We refer to the model in Equation 4.1 as a Mixture of Conditional Random Fields (MoCRF). With CRFs as its mixture components, this model can be viewed as an extension of a mixture of experts model [30] by predicting a structural output from data. Figure 4.3 provides an overview of the main components of the model. Below we describe the component CRF models in detail, followed by the gating function.

4.2.3 Context-dependent conditional random field

Given a context, the model captures the interactions between the labels of an image using a conditional random field of the labels $P_M(\mathbf{L}|\mathbf{X}, c)$. The random field is defined with respect to a graph G in which the label sites of neighboring super-pixels on the image plane are connected. We denote the neighbors of site i as $N(i)$.

The context-dependent CRF has three types of feature functions in its distribution, encoding the top-down contextual constraint of the labeling at three levels:

$$P_M(\mathbf{L}|\mathbf{X}, c) = \frac{1}{Z_c} \exp\{\sum_i f_a(\mathbf{l}_i, \mathbf{x}_i, c) + \sum_i \sum_{j \in N(i)} f_b(\mathbf{l}_i, \mathbf{l}_j, c) + f_c(\mathbf{L}, c)\}, \tag{4.2}$$

where $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$ is a feature function describing the compatibility of the local image descrip-

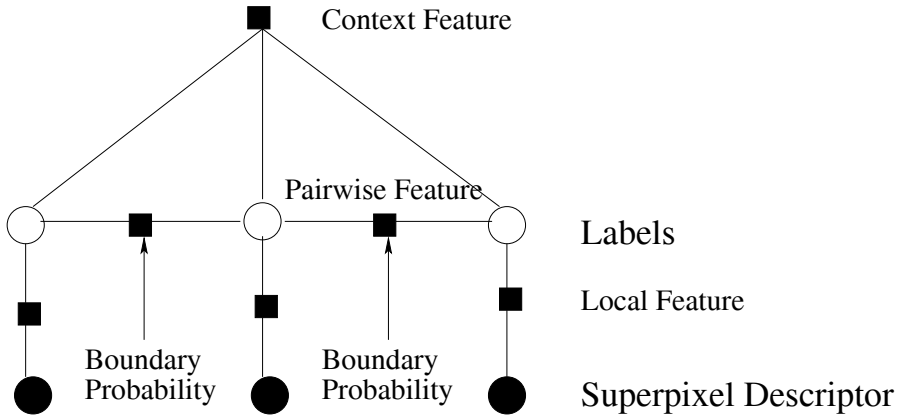

Figure 4.4: Graphical model representation of the context-dependent CRF. The context-specific processing combines local information based on super-pixel descriptor and specific label compatibility; pairwise interactions between labels of neighboring sites, modulated by the boundary probability; and global bias provided by the context-specific average label distribution.

tor $\mathbf{x}_i$ at super-pixel i to a particular label variable $\mathbf{l}_i$; $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$ accounts for pairwise interactions between labels of neighboring sites; and $f_c(\mathbf{L}, c)$ is a feature function for the global statistics of the label field $\mathbf{L}$ under context c. In our model, we implement those feature functions as follows. Figure 4.4 shows the graphical representation of the context-dependent CRF.

(a). Local features $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$

We utilize a classifier that independently predicts the label of every super-pixel to build the local feature function. The classifier provides a label distribution $\Phi_I(\mathbf{l}_i|\mathbf{x}_i, c)$ given input $\mathbf{x}_i$ and context c. The local feature $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$ has the following form:

$$f_a(\mathbf{l}_i, \mathbf{x}_i, c, \gamma^c) = \alpha^c \sum_{k \in \mathcal{L}} \delta(\mathbf{l}_i = k) \log \Phi_I(\mathbf{l}_i = k|\mathbf{x}_i, c, \gamma^c), \tag{4.3}$$

where $\delta(x) = 1$ if x is true and 0 otherwise, α^c is a coefficient for modulating the entropy of the classifier output for context c, and γ^c represents the classifier parameters. The feature function describes the preference of different label configurations given the input. In this paper, we use a multilayer perceptron (MLP) as the classifier which takes color, edge magnitude and texture information from the ith super-pixel's descriptor as the input. Note that these feature functions may be able to find local image features that uniquely characterize a particular class, such as the combination of color, texture, and edges in a rhino's horn.

(b). Pairwise features $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$

The pairwise feature functions exploit the local interactions between labels of neighboring super-pixels. Two super-pixels are neighbors if they share a segment of boundary on the image plane. We use a pairwise feature with a linear form in this model:

$$f_b(\mathbf{l}_i, \mathbf{l}_j, c) = \sum_{k \in \mathcal{L}} \sum_{k' \in \mathcal{L}} \delta(\mathbf{l}_i = k)\delta(\mathbf{l}_j = k') \log \Psi_{ij}^c(k, k'), \tag{4.4}$$

where Ψ_{ij}^c is a $|\mathcal{L}| \times |\mathcal{L}|$ compatibility matrix between label $\mathbf{l}_i$ and $\mathbf{l}_j$. The compatibility matrix incorporates both the statistics of neighboring label configurations and image descriptor information; it is defined as follows:

$$\Psi_{ij}^c(k, k') = \begin{cases} (1 - P_{ij}^b) \exp(\theta_{k,k'}^c) & k = k' \\ P_{ij}^b \exp(\theta_{k,k'}^c) & k \neq k' \end{cases} \tag{4.5}$$

where $\theta_{k,k'}^c$ is a scalar parameter for the compatibility of label values k, k' in context c, and P_{ij}^b is the boundary probability between super-pixel i and j.

This formulation incorporates boundary information provided by a separate boundary classifier [51]: P_{ij}^b is the average edge probability along the shared boundary between neighboring super-pixels, which modulates the label pair compatibility, implementing the intuitive notion that the compatibility of labels of neighboring sites depends on the presence of a boundary between them. For example, one would expect that the likelihood of neighboring labels taking on the same value would decrease if there is a boundary between them, while the compatibility of taking on different values would decrease if no boundary exists. Therefore, $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$ can be viewed as a data-dependent feature function specifying the regional context of labels.

(c). Global features $f_c(\mathbf{L}, c)$

The global feature function provide a coarse level constraint for the label configuration of the random field. In our model, the global features constrain the overall image label distribution to conform to a typical, average label distribution that characterizes the relative proportion of the various labels in a specific context. Assuming this average label distribution is $\mu^c = (\mu_1^c, ..., \mu_{|\mathcal{L}|}^c)$ for a given context c, we define a global feature that maximizes the match between the actual label distribution and the distribution μ^c:

$$f_c(\mathbf{L}, c) = \beta^c \sum_i \sum_{k \in \mathcal{L}} \delta(\mathbf{l}_i = k) \log \mu_k^c, \tag{4.6}$$

where β^c is the weighting coefficient. This feature function is equivalent to the negative Kullback-Leibler divergence between the image label distribution and the target distribution for the given context. Note that this feature provides a global bias to the single node potential in the conditional random field.

4.2.4 Gating function $P_G(c|\mathbf{X})$

The gating function is specified by a context classifier which generates a distribution of context c given an input image. The inputs to the classifier are the aggregate statistics of the image descriptors, including color, edge density and texture information. We use a multilayer perceptron as the context classifier in this model.

4.2.5 Model summary

To summarize, our model has the following form:

$$P(\mathbf{L}|\mathbf{X}) = \sum_c \frac{P_G(c|\mathbf{X})}{Z_c} \exp\{\sum_{i,j} \mathbf{l}_i^T \log \Psi_{ij}^c \mathbf{l}_j + \alpha^c \sum_i \mathbf{l}_i^T \log \Phi_I + \beta^c \sum_i \mathbf{l}_i^T \log \mu^c\} \quad (4.7)$$

where the label variable $\mathbf{l}_i$ is represented as a vector with $|\mathcal{L}|$ elements, in which the kth element is 1 and the other elements are 0 when $\mathbf{l}_i = k$. Note that the final label distribution can readily be used to define a segmentation of the image into coherent regions, where a segment corresponds to each contiguous group of pixels that are assigned the same label.

4.3 Labeling Inference

Given a new image $\mathbf{X}$, we predict its labeling based on the Maximum Posterior Marginals (MPM) criterion:

$$\mathbf{l}_i^* = \arg\max_{\mathbf{l}_i \in \mathcal{L}} \sum_{c \in \mathcal{C}} P_M(\mathbf{l}_i|\mathbf{X}, c) P_G(c|\mathbf{X}), \quad (4.8)$$

where the marginal label distributions of each super-pixel, $P_M(\mathbf{l}_i|\mathbf{X}, c)$, are computed by applying loopy belief propagation to every context-dependent CRF.

4.4 Parameter Estimation

4.4.1 Learning criterion

Given a set of labeled image data $\mathcal{X} = \{(\mathbf{L}^n, \mathbf{X}^n)\}$, we estimate the model parameters based on the Conditional Maximum Likelihood criterion, that is,

$$\hat{\Theta} = \arg\max_{\Theta} \sum_n \log P(\mathbf{L}^n|\mathbf{X}^n), \tag{4.9}$$

where Θ denotes all the parameters in the model. Treating the context variable c as missing data, we could apply the EM algorithm to the learning problem. However, due to the partition functions in the mixture components, the posterior distribution $q(c|\mathbf{L}^n, \mathbf{X}^n)$ is intractable. Instead, we define a new cost function which is a lower-bound of the conditional data likelihood:

$$Q = \sum_n \sum_c P_G(c|\mathbf{X}^n) \log P_M(\mathbf{L}^n|\mathbf{X}^n, c). \tag{4.10}$$

Note that $Q \leq \sum_n \log[\sum_c P_G(c|\mathbf{X}^n) P_M(\mathbf{L}^n|\mathbf{X}^n, c)] = \sum_n \log P(\mathbf{L}^n|\mathbf{X}^n)$.

4.4.2 A modular training approach

Given the cost function in Eqn. 4.10, we can compute its gradient and estimate all the parameters using a gradient ascent method. However, training all parameters together becomes difficult in practice when we have a large label set, and large image database. In this work, we propose a modular approach to estimate the parameters, such that many components are learned separately and are then merged into the full system in a consistent way. This learning procedure may not produce an optimal system ultimately, but the approach leads to a more efficient learning process, capable of scaling up to large datasets.

The learning procedure is carried out as follows:

1. We cluster the training data, where each training image is represented by its aggregate label distribution, and define each cluster as a context. The clustering divides the training data into subsets, such that each image corresponds to a specific context.

2. Given this division of training data, we can train the gating function that predicts which context an image is in given its image features. The image features are based on the global image statistics, and do not require the super-pixelization.

3. Within each subset, we estimate the parameters $\{\gamma^c\}$ of each context-dependent image classifier to independently predict the label distribution given the super-pixel descriptors as input.

4. Finally, we combine these components and jointly learn the remaining parameters in the model (the coefficients $\{\alpha^c, \beta^c\}$ and the compatibility parameters θ^c) by maximizing the cost function in Eqn. 4.10.

More specifically, in step 1, the clustering method is based on a mixture of unigram model for the labels: $P_u(\mathbf{L}) = \sum_c \prod_i P_u(\mathbf{l}_i|c)P_u(c)$, which we learn using the EM algorithm on the training data set. The conditional probability $P_u(\mathbf{l}_i|c)$ acts as the cluster center, or the prototype label distribution in context c, and is thus used as μ^c in the global feature function. In step 2, given the mixture of unigram model, we can compute the cluster responsibility of every image. Those responsibilities are used as training targets for the gating function $P_G(c|\mathbf{X})$. Step 3 can occur in parallel with step 2, as by sampling the responsibilities, we can form the context-dependent subsets from the training data, and learn the parameters γ^c of the local feature functions on the appropriate subsets.

Finally, in step 4, after parameters of the local and global feature functions as well as the gating function have been learned, we merge them into the model and optimize the remaining parameters with respect to the cost function. Note that the context-dependent CRFs are log-linear models with parameters $\{\theta^c, \alpha^c, \beta^c\}$, which can be estimated by gradient ascent:

$$\Delta\theta^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_{i,j\in N(i)} (\mathbf{l}_i^n \mathbf{l}_j^{nT} - \left\langle \mathbf{l}_i \mathbf{l}_j^T \right\rangle_{P_M(\mathbf{l}_i,\mathbf{l}_j|\mathbf{X}^n,c)}) \tag{4.11}$$

$$\Delta\alpha^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_i (\mathbf{l}_i^{nT} - \left\langle \mathbf{l}_i^T \right\rangle_{P_M(\mathbf{l}_i|\mathbf{X}^n,c)}) \log \Phi_I(\mathbf{l}_i|\mathbf{x}_i^n, c) \tag{4.12}$$

$$\Delta\beta^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_i (\mathbf{l}_i^{nT} - \left\langle \mathbf{l}_i^T \right\rangle_{P_M(\mathbf{l}_i|\mathbf{X}^n,c)}) \log \mu^c. \tag{4.13}$$

To avoid overfitting, we add a Gaussian prior on the parameters, which is equivalent to weight decay during learning. As the CRFs are defined on loopy graphs with intractable partition functions, the marginal distributions of the label variables in the gradient updates cannot be computed exactly. In this work, we approximate them by applying the loopy belief propagation algorithm. An alternative approach is to apply contrastive divergence [29] to each component CRF. The empirical results show that both of these approaches obtain similar and satisfactory performance in our model; below we report results using loopy belief propagation.

4.5 Experimental Evaluation

4.5.1 Data sets

We applied our model to three different real data sets. In order to compare our method with an alternative approach, we utilized the two datasets used in our mCRF work, and used the same training and testing split as in that work. The first dataset is the Sowerby database, including a set of color images of outdoor scenes and their associated labels. The data set has a total of 104 images with 7 labels: 'sky', 'vegetation', 'road marking', 'road surface', 'building', 'street objects' and 'cars'. 60 of these images are used for training and the remaining 44 for testing. The second dataset is a 100-image subset of the Corel image database, consisting of African and Arctic wildlife natural scenes. It also has 7 classes: 'rhino/hippo', 'polar bear', 'vegetation', 'sky', 'water', 'snow' and 'ground'; and has a train/test split of 60/40.

To explore the scaling potential of our approach, we defined a third dataset by expanding this Corel dataset to include 305 manually labelled images with 11 classes: 'rhino/hippo', 'tiger', 'horse','polar bear', 'wolf/leopard', 'vegetation', 'sky', 'water', 'snow', 'ground' and 'fence'. The training set includes 229 randomly selected images and the remaining 76 are used for testing. We call this extended Corel data set CorelB, and refer to the smaller one as CorelA in the following sections.

Again, for comparison purposes, we use the same set of basic image features as in Chapter 3, including color, edge and texture information. For the color information, we transform the RGB values into CIE Lab* color space, which is perceptually uniform. The edge and texture are extracted by a set of filter-banks including a difference-of-Gaussian filter at 3 different scales, and quadrature pairs of oriented even- and odd-symmetric filters at 4 orientations (0; $\pi/4$; $\pi/2$; $3\pi/4$) and 3 scales. We also include the vertical and horizontal position of each pixel. Thus each pixel is represented by a 32 dimensional image feature vector. For super-pixels, we compute the normalized histograms of those image features extracted from the pixels in each super-pixel. We choose 20 bins for each dimension of the pixel-wise feature vector, and concatenate the histograms into a descriptor vector of each super-pixel.

4.5.2 Model specification

We use the normalized cut segmentation algorithm to build the super-pixel representation of the images, in which the segmentation algorithm is tuned to generate more than 300 segments for each image. The number of segments is set based on the average label error rate induced

by assigning each segment its majority label value. Segments smaller than a minimum size (6 pixels) are merged into the neighboring super-pixels. This yields approximately 300 super-pixels per image on average, and the induced average label errors are below 5% for the datasets we used in experiments. The boundary information is extracted using the algorithm in [51]. To avoid underflow due to zero compatibilities, we convert the raw output of boundary probability into interval $[0.1, 0.9]$ by an affine transform.

The number of contexts can be chosen based on the trade-off between the performance of the gating function and the complexity of each context group. In our experiments, we specified the number of contexts empirically based on the complexity of data set. For Sowerby and CorelA data sets, we use 2 contexts in clustering, and for CorelB, we use 4 contexts. The model selection issue is not explored here, and is left to future work.

The gating function is a MLP with 25 hidden units. It takes the normalized histograms of the image features in each image as input. We use 20 bins for each image feature. To avoid overfitting, the MLP is trained with Gaussian priors on weights. The local classifiers are also MLPs with 30 hidden units, using the histograms of the image features in each super-pixel as input. They are trained with cross-validation.

We compare our approach with a simple pixel-wise classifier, a simple CRF model and the mCRF model from Chapter 3. These comparisons provide insight into the utility of the pairwise compatibilities (CRF vs. classifier), the contexts (MoCRF vs. CRF) and the simplified structure (MoCRF vs. mCRF). The pixel-wise classifier is a MLP with one hidden layer, taking image features from a 3×3 window centered at each pixel and predicting the pixel's label. The CRF uses context-independent local feature and pairwise feature functions. The feature functions have the same form as our model. The distribution of label configuration $\mathbf{L}$ defined by the CRF has the following form:

$$P_{CRF}(\mathbf{L}|\mathbf{X}) = \frac{1}{Z} \exp\{\sum_{i,j} \mathbf{l}_i^T \log \Psi_{ij} \mathbf{l}_j + \alpha \sum_i \mathbf{l}_i^T \log \Phi_I(\mathbf{l}_i|\mathbf{x}_i)\} \tag{4.14}$$

where Φ_I is a local classifier trained separately on all the data and Ψ_{ij} is the compatibility function including boundary information. We trained the CRF model using the pseudo-likelihood algorithm, and tested its performance using the same MPM criterion where the marginal distribution is calculated by the loopy belief propagation algorithm.

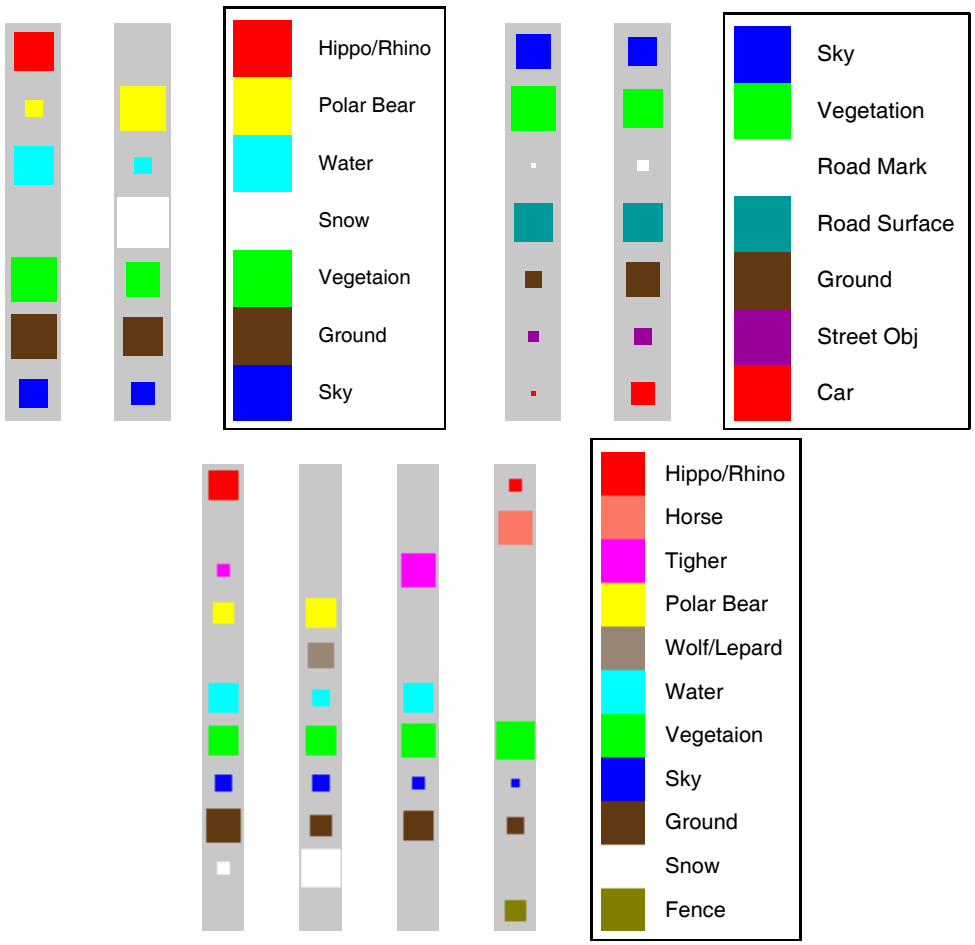

Figure 4.5: The learned prototype label distribution for each of the three datasets: CorelA, Sowerby, and CorelB, is shown, with its associated key. The size of each square is proportional to the probability of the associated class. See text for discussion.

4.5.3 Results

We clustered the training images in each dataset as described above, yielding 2 clusters for the CorelA and Sowerby datasets, and 4 clusters for CorelB. In Fig. 4.5, we visualize the typical label distributions of the contexts from all three datasets. Note that these distributions usually have semantic meaning which is easy to interpret. For instance, the contexts in CorelA dataset represent the tropical and arctic environments, while the Sowerby dataset contexts are rural and suburban areas. CorelB dataset has 'tropic','field','jungle' and 'arctic' as its contexts.

Given the context settings, we trained a context classifier as the gating function for each dataset. To evaluate those context classifiers, we use the largest cluster responsibility as the target context, and compute the accuracy of the classifier output. Based on that metric, the

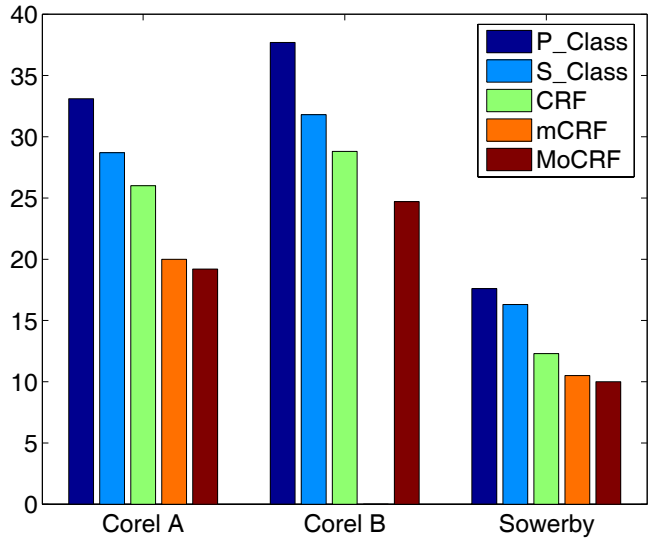

Figure 4.6: Classification error rates for the models: P_Class is the pixel level classifier, S_Class is the super-pixel level classifier, CRF is the simple CRF model, mCRF is the multiscale Conditional Random Field in Chapter3 and MoCRF is the Mixture of Conditional Random Fields.

context classifiers we trained achieve 82%, 92% and 85% accuracy on Sowerby, CorelA and CorelB, respectively.

The performance of MoCRF is first evaluated according to the label error metric on the pixel level, i.e., the percentage of incorrectly labelled pixels. We compared the performance of MoCRF to a simple pixel-wise classifier (P_Class), the super-pixel classifier in MoCRF considered alone (S_Class), and the CRF model over three datasets. We also include the performance of mCRF on the Sowerby and CorelA datasets from Chapter 3. The classification error rates on the test sets of three datasets are shown in Figure 4.6. Note that, for Corel B dataset, the performance of mCRF is missing due to difficulty in learning it on that large dataset.

We can see that the super-pixel based classifiers alone provide a significant improvement over the pixel-wise classifiers. Built on the the same bottom-up cues, our model also has better performance over the super-pixel classifier and the conventional CRF model. Furthermore, it provides a slightly better performance than the mCRF model. Note that our MoCRF model has a much simpler structure than the mCRF model: for the Sowerby and CorelA datasets, MoCRF has approximately 300 label variables, (equal to the number of super-pixels), no hidden variables, and approximately 120 parameters for training excluding the classifiers; while mCRF has about 2×10^4 label variables, 10^3 hidden variables and 10^3 free parameters. Learning is

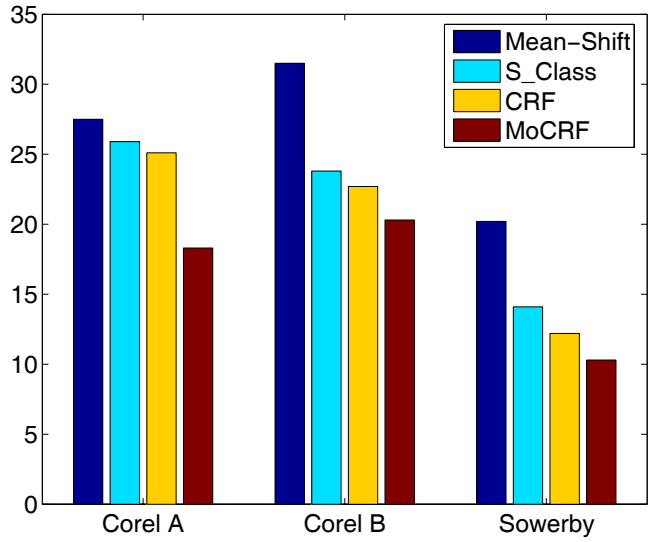

Figure 4.7: Segmentation error rates measured by the percentage of pixel pairs that are incorrectly segmented. S_Class is the super-pixel level classifier, CRF is the simple CRF model, MoCRF is the Mixture of Conditional Random Fields.

therefore quite slow in mCRF, and the model has poor scaling properties. Thus, although we only match this earlier model in terms of classification accuracy, our model can be applied to the problems with a considerably larger set of labels and larger image sizes.

We compare the performance of the pixel-wise classifier, our model, and Mean-Shift segmentation [15] in Figure 4.7. We tune the parameters of Mean-Shift such that it generates the best results according to the manual labeling for a small set of randomly chosen images. The performance is measured according to a second metric used for evaluation, a segmentation metric which computes the percentage of pixel pairs that are correctly segmented. In other words, a pair of pixels are correctly segmented if two with the same labels shares the same segment, and two with different labels are in different segments. To reduce the computational burden, we randomly sampled 10% pixels from each image to estimate the accuracy. Again, we can see that our model obtains better results by adding top-down category information, and multi-level contextual constraints.

We also show the outputs of these methods on some test images in Figure 4.8 and Figure 4.9. The figure shows the approaches based solely on low-level cues can be fooled, such that some single objects in the images are split. MoCRF works much better on those images by integrating the super-pixel representation and mixture of CRF framework. Note that the super-

pixelization will cause some errors which cannot be corrected by the top-down information. Also, the model cannot use global spatial configuration to correct errors since no geometric information is included in the global feature functions.

4.6 Conclusion

In this chapter, we have presented a discriminative framework that integrates bottom-up and top-down cues for image segmentation. We adopt a labeling approach to provide some purchase on the segmentation problem. A chief contribution of our model with respect to segmentation is the resulting extension of top-down cues to include a considerably wider range of object classes than earlier methods. The proposed framework is modular, in that images in a database are classified as to their context, and separate processes are learned for the different contexts. This modularity presents some promise of the system extending to large databases of images. While the top-down cues can be learned in a context-specific manner, the system integrates these with bottom-up cues, which are utilized in several ways: to define super-pixels in an image; to determine probabilities of local boundaries between super-pixels, which are used to constrain and guide labeling; and to enable context classification.

The results of applying our method to three different image datasets suggest that this integrated approach may extend to a variety of image types and databases. The labeling system consistently out-performs alternative approaches, such as a standard classifier and a standard CRF. Its performance matches that of the mCRF model in Chapter 3, which operates at the pixel level and entails a considerably more involved training procedure, one which is unlikely to scale to larger images and image databases. Relative to a standard segmentation method, the segmentations produced by our method are more accurate, even when the standard method is optimized for a given test image. A relatively weak component in our model appears to be the gating function, as the images whose contexts are incorrectly classified contain a disproportionate number of label errors. A possible direction is to use other methods of summarizing the statistics of an image (e.g., [75]) in order to facilitate more accurate context classification. A second limitation of our model concerns the omission of shape information. While the local feature functions can potentially detect local shape cues, the model is unable to learn and utilize spatial relations between parts of an object in the labeling procedure. Finally, a third limitation of our model concerns its reliance on detailed training data. However, a growing effort to label images (e.g., [62]) should lead to a rapid growth in the volume of available labeled images, and the issue of relaxing this requirement will be addressed in the next chapter.

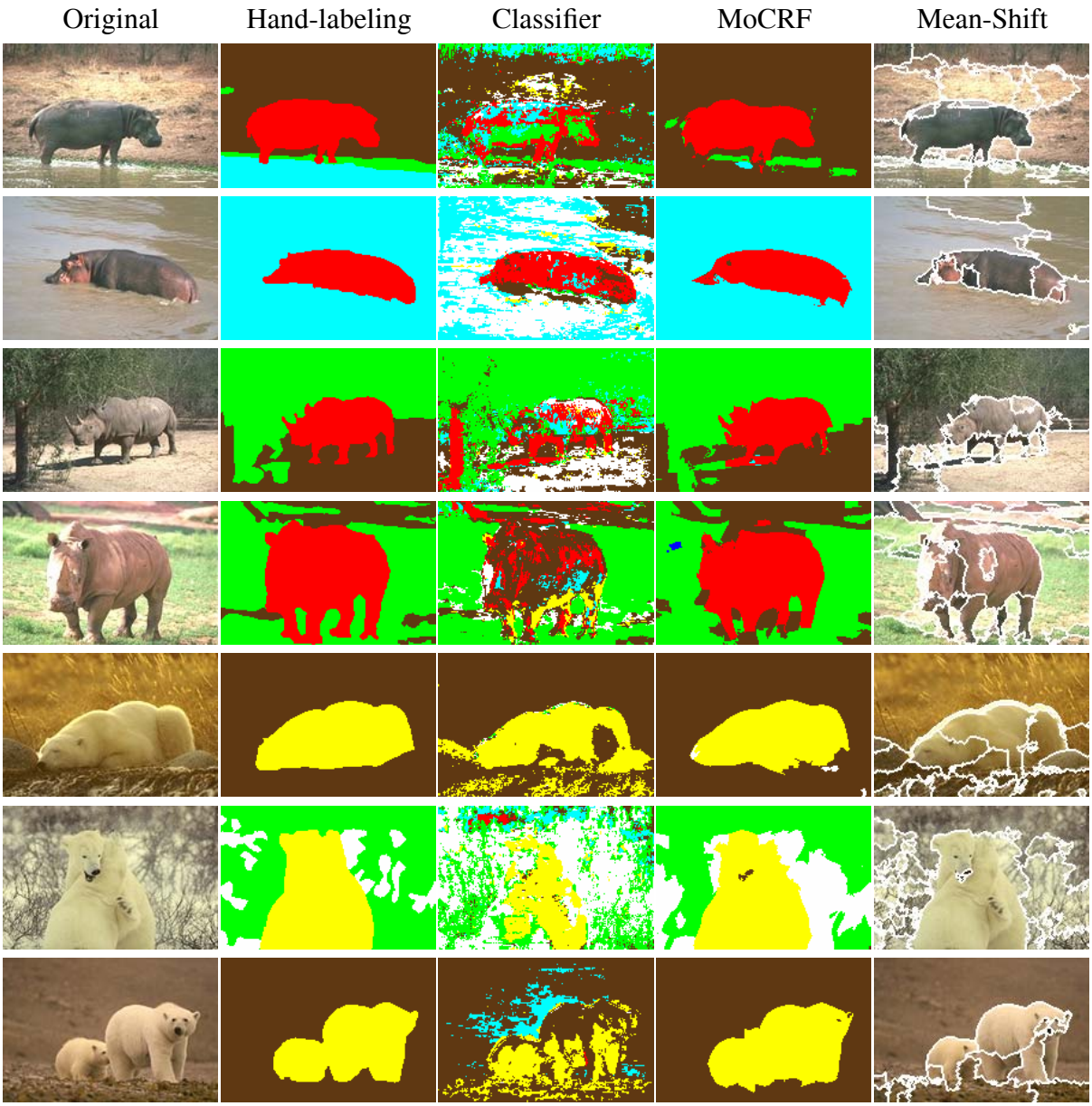

Figure 4.8: Some labeling results for the Corel datasets, using the pixel-wise classifier, CRF, MoCRF, and Mean Shift segmentation. The color keys for the labels are the same as Fig. 4.5.

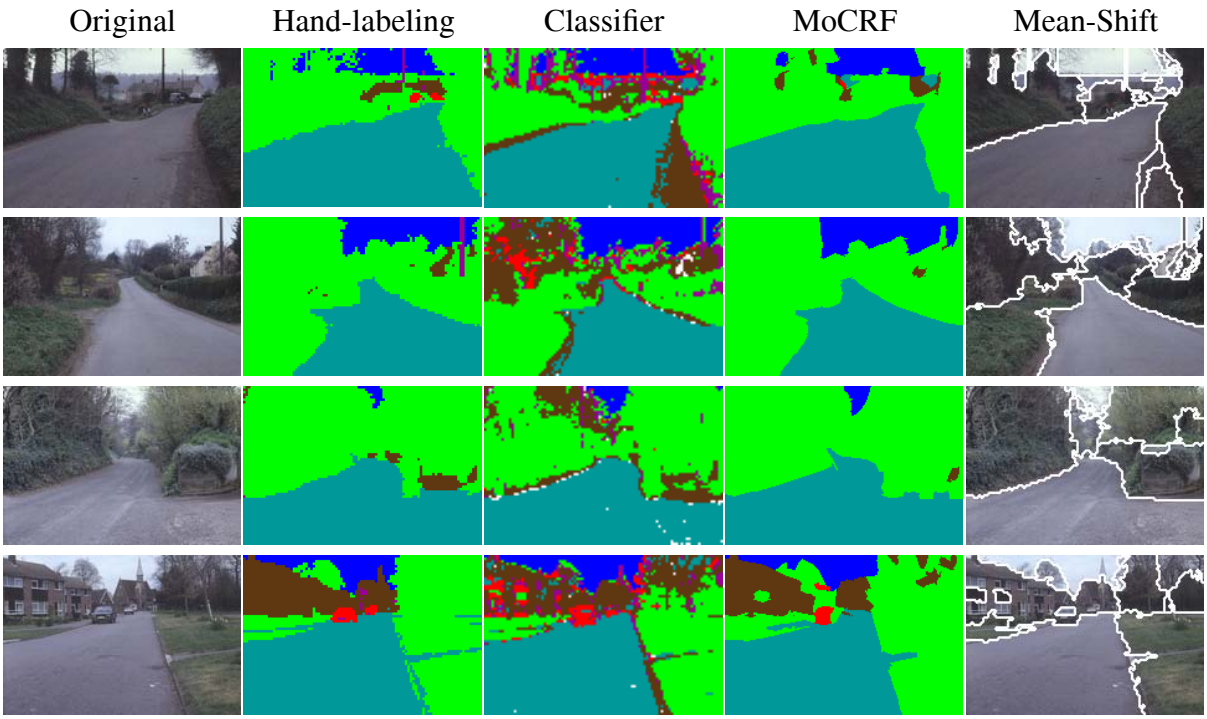

Figure 4.9: Some labeling results for the Sowerby dataset, using the pixel-wise classifier, CRF, MoCRF, and Mean Shift segmentation. The color keys for the labels are the same as Fig. 4.5.

Chapter 5

Topic Random Field and Learning with coarsely-labeled Data

5.1 Introduction

The discriminative approaches, such as Conditional Random Fields, have been successfully used in structural labeling tasks. In the previous chapters, we addressed the representation issue of context in the discriminative framework. One main limitation of discriminative methods is that they require a detailed-labeled data set to construct the models. When labeling information is unreliable, vague or missing, it is hard to apply them. It is nearly impossible to find a dataset where this is not the case in real-world vision applications. Labeling every pixel of an image manually is very tedious, and not practical for real-world datasets that include large numbers of images.

On the other hand, it is easier to obtain weakly-labeled image data, such as images with captions. While different types of weakly labeled datasets are available for image labeling, we focus here on image data with multiple levels of labels. In many cases, the label values have different levels of granularity, and they can be grouped into a label hierarchy based on their semantics. For example, a region with the label 'hippo' can also be labeled as 'animal' or 'animate object'. Using such more abstract or coarse labels requires less effort in labeling images, as the coarser label set often has a simpler structure than the detailed ones, and is easier to specify. Figure 5.1 shows a typical example where the coarse label configuration has simpler boundaries. We also consider another aspect of the coarseness in labeling, which means some image regions are unlabeled. Those regions are either not relevant to target problem, or too vague to be recognized. In practice, it is desirable to augment the existing detailed-labeled

Figure 5.1: Image labeling with detailed labels and coarse labels. Left: Original image. Middle: Detailed labeling. (Brown='plane', green='grass', grey='sky', olive='tree' and dark='void'.) Right: Coarse Labeling. (Brown='animate object', gray='static object', and dark='void'.)

images with those coarsely-labeled image to leverage the learning process.

In this chapter, we develop a hybrid approach to utilize both a small set of image data with detailed labels, and a bigger set of images with coarse labels. The variety of labelings requires a different approach to capture the context information. Instead of relying on learning context via patterns in the label field, we incorporate a generative image model for capturing the common image contexts. One widely used image model family is the Hidden Markov Random Fields. However, they usually include only local connectivity for computational efficiency. This limits the model capacity to capture the high-order interactions.

We consider a more flexible approach of generating data by extending a latent topic model, such as the Latent Dirichlet Allocation (LDA) [6]. In the original LDA model, topics capture co-occurring words, and are applied to the entire document. We use the topic model to model the appearance of image features, which can capture any feature configuration in the entire image. We also extend the topic model such that the topics are not just applied to input words, but also to labels. Given a topic, the model generates the input data, as well as a topic-dependent probabilistic mapping from input data to the output labels. With the discriminative component, we can apply the topic model to labeling tasks. We further introduce locality into the topic model, constraining topics to image features that occur together in some local spatial context. For example, the context can be regions in images. We refer to both types of extended topic models (with and without locality) as a latent topic random field (LTRF). Unlike the traditional Markov models, our approach has the flexibility of modeling context with different complexity, including higher-order patterns. Also, the topics with locality constraint can potentially capture image context with different scope in the image plane, which is useful for labeling objects with weakly constrained configurations.

We learn the model from a small detailed-labeled image set and a larger coarsely-labeled image set. The data with detailed labels provide precise information for learning the mapping from input to the outputs, whereas the data with coarse labels help the system to build a better topic model for image features. The coarsely-labeled set prevents the topics from overfitting a limited number of detailed-labeled images. In addition, those coarse labels give informative cues for learning the topics, which is valuable for the labeling task, but hard to achieve by using purely unlabeled data.

This chapter is organized as follows. In Section 5.2, we describe the label hierarchy and the architecture of our latent topic random field model. Section 5.3 presents the inference algorithm for labeling a new image, and the inference used in learning the model. The learning procedure is detailed in Section 5.4. We compare our model with other approaches based on a real world image dataset in Section 5.5. Section 5.6 summarizes this chapter and discusses some further issues.

5.2 Model Architecture

5.2.1 Label Hierarchy

We consider a situation in which we have a set of label values that are not exclusive and can form a hierarchy according to their semantics. The label hierarchy could be derived from some taxonomy of concepts, as in WordNet [1], or from a hierarchical clustering process. For example, Figure 5.2 shows a hierarchy of objects that appear in the Microsoft Research Cambridge Image Dataset [2]. In such a tree structure, a parent label value includes its children as special cases. The label values at the leaf nodes correspond to a detailed labeling of images, whereas the internal nodes are used in coarse labeling. In this chapter, we assume that the hierarchy is given, and each training image is labeled at some level: either using detailed labels at the leaf level, or coarse labels from a single internal level. The construction of this type of label hierarchy may be problem specific, and will not be pursued here. Note that the label hierarchy gives two different senses of coarseness: 1) First, the internal label values are coarse in terms of their semantics; 2) Second, the spatial configuration of labeling can be coarser due to merging of subclasses (See Figure 5.1). In addition, to handle unlabeled regions, both types of labelings include the background as a "catch-all" label.

[1] See `http://wordnet.princeton.edu/`

[2] See `https://research.microsoft.com/vision/cambridge/recognition/default.htm`

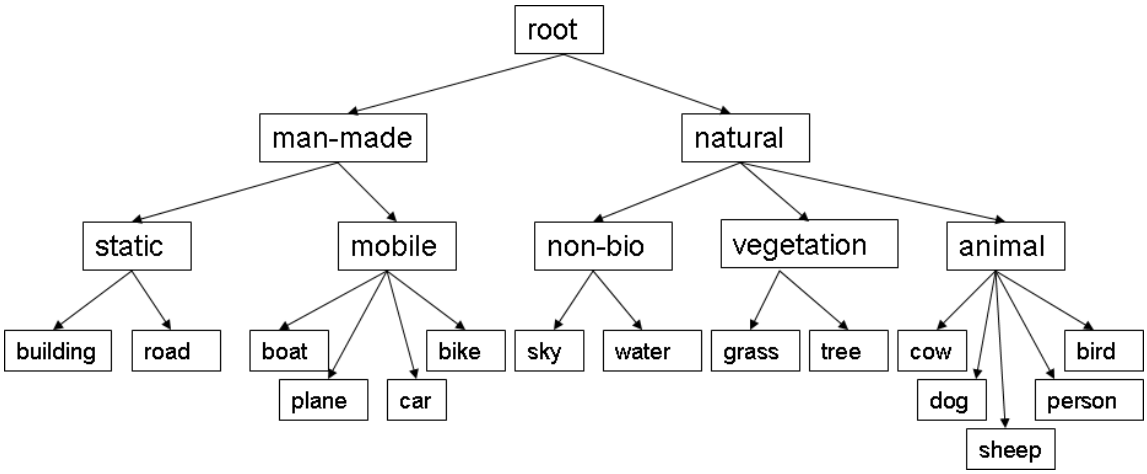

Figure 5.2: A label hierarchy of objects used in the Microsoft Research Cambridge Image Dataset. We construct the hierarchy based on the semantics of labels.

5.2.2 Topic Model with Labels

We start by building a generative topic model for images based on the Latent Dirichlet Allocation model. Let us introduce some notation first. Suppose each image is represented by a set of image features, and the ith feature has its appearance descriptor a_i in image location x_i. The appearance variable a_i takes values from a vocabulary of visual words. We associate two types of labels with each image feature: coarse label f_i and detailed label l_i. The detailed label variable l_i takes values from $\{1, ..., L\}$, and the coarse label f_i from $\{1, ..., L_c\}$. The graphical representation of the model is shown in Figure 5.3, and its details are described as follows.

We assume that the appearance of image feature a_i is generated from a finite set of hidden topics. Each image I is described by a multinomial distribution θ over the hidden topics. Let the number of topics be K, then θ is a K dimensional vector. To generate a new appearance a_i in an image, we start by first sampling a hidden topic z_i from the multinomial distribution θ corresponding to the image. Given the topic z_i, the distribution over appearances is multinomial with parameters β_{z_i}. As in the LDA model, the parameters between different images are tied by drawing θ of all images from a common Dirichlet prior parameterized by α. α is also a vector in K dimensional space.

We then incorporate the label variables into the latent topic model of images. Given image feature a_i, its location x_i and the corresponding topic z_i, the label pair $\{f_i, l_i\}$ is predicted from a conditional multinomial distribution. Viewing each topic as a context, we have a context dependent appearance model $p(a_i|z_i)$, and a context dependent label predictor $P(l_i, f_i|a_i, x_i, z_i)$.

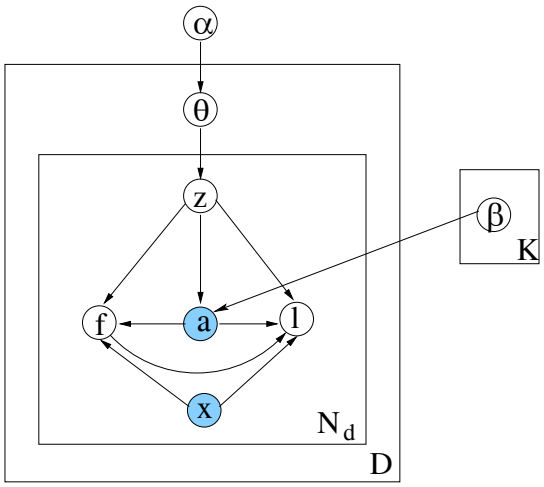

Figure 5.3: A graphical representation of the extended topic model with both image features and their labels. N_d is the number of image features in each image, and D denotes all the training data.

Note that the appearance model is position invariant, whereas the label predictor uses the position information. Thus, the joint distribution of the model can be written as

$$P(\mathbf{a}, \mathbf{l}, \mathbf{f}, \mathbf{z}, \theta|\alpha, \mathbf{x}) = P(\theta|\alpha) \prod_{i=1}^{N_d} P(l_i, f_i|a_i, x_i, z_i) P(a_i|z_i) P(z_i|\theta) \tag{5.1}$$

where N_d is the number of image features in image I_d.

We further parameterize the conditional distribution of the pair $\{f_i, l_i\}$ as follows: the coarse label f_i is first generated from a multinomial $P(f_i|a_i, x_i, z_i)$, given a_i, x_i and z_i. Conditioned on the coarse label, the detailed label l_i is then generated from $P(l_i|f_i, a_i, x_i, z_i)$ (see Figure 5.3). The coarse label predictors $P(f_i|a_i, x_i, z_i)$ are modeled by a set of topic dependent probabilistic classifiers: for each topic k, we have a classifier $P_k^c(f|a, x)$. In other words,

$$P(f_i|a_i, x_i, z_i) = P_{z_i}^c(f_i|a_i, x_i), \tag{5.2}$$

where we assume that the classifiers produce properly normalized distributions.

To build $P(l_i|f_i, a_i, x_i, z_i)$, we introduce another set of classifiers, in which $P_k^d(l|a, x)$ predicts the detailed label given the topic k and input (a, x). We also assume their outputs form normalized distributions. The conditional distribution of the detailed label l_i given other information is written as

$$\begin{aligned} P(l_i|f_i, a_i, x_i, z_i) &\propto P_{z_i}^d(l_i|a_i, x_i)\delta[l_i \in f_i] \\ &= \frac{P_{z_i}^d(l_i|a_i, x_i)\delta[l_i \in f_i]}{\sum_{l \in f_i} P_{z_i}^d(l|a_i, x_i)} \end{aligned} \tag{5.3}$$

where $\delta[l_i \in f_i]$ is 1 if l_i is a child node of f_i in the label hierarchy and 0 otherwise. Notice that by summing out the coarse label variables, the conditional distribution of the detailed label given input and topic can be written as

$$P(l_i|a_i, x_i, z_i) = P^d_{z_i}(l_i|a_i, x_i)\frac{P^c_{z_i}(f[l_i]|a_i, x_i)}{\sum_{l\in f[l_i]} P^d_{z_i}(l|a_i, x_i)} \tag{5.4}$$

where $f[l_i]$ is the parent coarse label of l_i in the label hierarchy.

The topic model essentially introduces a weak correlation between image features and their labels. To see this, we integrate out the Dirichlet variable θ, and the marginal distribution of the model can be written as

$$P(\mathbf{z}, \mathbf{a}, \mathbf{l}, \mathbf{f}|\alpha, \mathbf{x}) = \int_\theta P(\mathbf{a}, \mathbf{l}, \mathbf{f}, \mathbf{z}, \theta)d\theta \tag{5.5}$$

$$= \prod_i P(l_i, f_i|x_i, a_i, z_i)P(a_i|z_i)\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)}\frac{\prod_k \Gamma(\alpha_k + \sum_i \delta(z_i, k))}{\Gamma(\sum_k \alpha_k + N)} \tag{5.6}$$

where $\Gamma(\cdot)$ is the Gamma function and N is the total number of elements in $\mathbf{z}$.

Our model can be viewed as a hybrid structure of a generative topic model and a set of discriminative classifiers. While we introduce the label variables into the topic model, the basic assumption of the original LDA model, which views each image as a bag of features, remains the same. The advantage of this weak assumption is that a topic could correspond to any co-occurring feature pattern in images. However, those patterns may not be easy to interpret.

5.2.3 Topic Model with Labels and Locality

In certain problems, we may expect that topics are localized in the image plane: each topic is associated with a set of neighboring features. For instance, if a topic corresponds to an object part, its features are usually grouped together. We augment this locality property into our topic label model by introducing lateral connections between the hidden topic variables $\{z_i\}$.

To be specific, we first construct a graph on the topic variables by connecting the topic variables of neighboring image features in the image plane. Given the graph, we define a random field of topic variables with pairwise compatibilities. The random field then is combined with the original topic model multiplicatively to form the full prior of the topics. The locality of the topics is imposed by the random field, which encourages neighboring z_i's to share the same topic. Denoting the compatibility between z_i and z_j by $\phi(z_i, z_j)$, we can write the joint

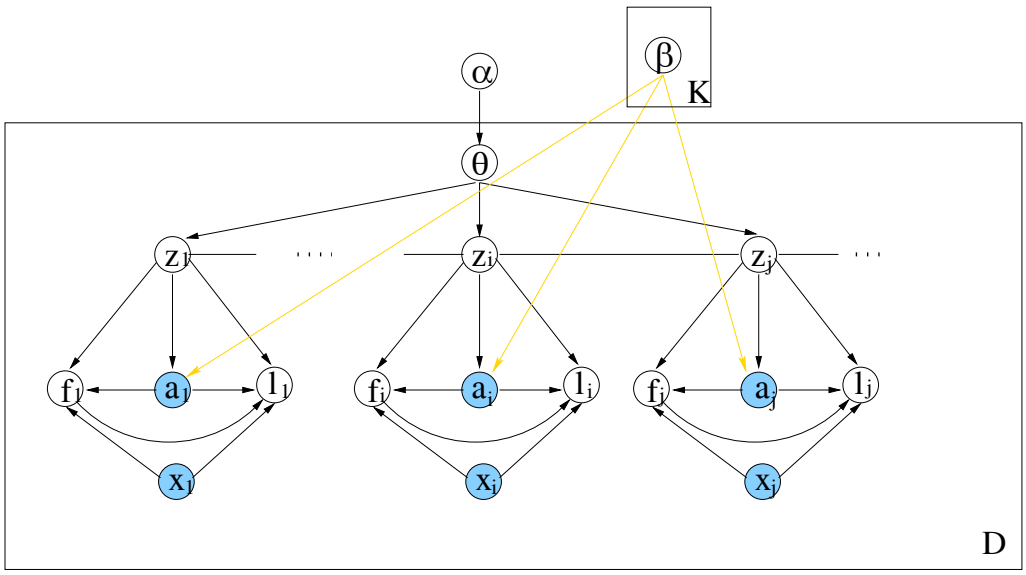

Figure 5.4: A graphical representation of the general Latent Topic Random Field with image features and their labels. D denotes all the training data.

distribution of the model as

$$P(\mathbf{a}, \mathbf{l}, \mathbf{f}, \mathbf{z}, \theta | \alpha, \mathbf{x}) = \frac{1}{Z} P(\theta|\alpha) \prod_i P(l_i, f_i | a_i, x_i, z_i) P(l_i|z_i) P(z_i|\theta) \prod_{i<j} \phi(z_i, z_j) \tag{5.7}$$

The compatibility $\phi(z_i, z_j)$ is associative, that is, it takes a larger positive weight for the same neighboring nodes than the different ones. Mathematically, $\phi(z_i, z_j) = \delta(z_i, z_j) + w(1 - \delta(z_i, z_j))$, where $0 < w \leq 1$. This prior will encourage each topic to form blob-like regions. Notice that the model with locality includes the one without locality in Section 5.2.2 as a special case with $w = 1$. Therefore, we refer to both types of these topic models with labels as Latent Topic Random Fields (LTRFs). Figure 5.4 shows the graphical representation of the general LTRF.

Note that LTRF can capture both pairwise and potential high-order interactions between image features. This can be seen by integrating out the Dirichlet variable θ, and the distribution of topic variables, labels and inputs is

$$P(\mathbf{z}, \mathbf{a}, \mathbf{l}, \mathbf{f} | \alpha) = \int_\theta P(\mathbf{a}, \mathbf{l}, \mathbf{f}, \mathbf{z}, \theta) d\theta \tag{5.8}$$

$$\propto \prod_i P(l_i, f_i | a_i, x_i, z_i) P(a_i|z_i) \prod_{i<j} \phi(z_i, z_j) \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \frac{\prod_k \Gamma(\alpha_k + \sum_i \delta(z_i, k))}{\Gamma(\sum_k \alpha_k + N)} \tag{5.9}$$

in which the prior over topic variables z_i includes both the associative term and the topic term.

5.3 Inference and Label Prediction

Given a new image $I = \{\mathbf{a}, \mathbf{x}\}$ and our topic model, we predict its labeling based on the Maximum Posterior Marginals (MPM) criterion:

$$(l_i^*, f_i^*) = \arg\max_{l_i, f_i} P(l_i, f_i|\mathbf{a}, \mathbf{x}), \tag{5.10}$$

where the marginal distribution $P(l_i, f_i|\mathbf{a}, \mathbf{x})$ can be computed by

$$P(l_i, f_i|\mathbf{a}, \mathbf{x}) = \sum_{z_i} P(l_i, f_i|z_i, a_i, x_i) P(z_i|\mathbf{a}, \mathbf{x}). \tag{5.11}$$

The key step in inference is to obtain the conditional distribution of the hidden topic variables $\mathbf{z}$ given observed data components. By integrating out the Dirichlet variable θ, we take a Gibbs sampling approach to estimate that distribution. From Equation 5.9, we can derive the posterior of each topic variable z_i given other variables as,

$$P(z_i = k|\mathbf{z}_{-i}, \mathbf{a}, \mathbf{x}) \propto P(a_i|z_i) \prod_{j \in N(i)} \phi(z_i, z_j)(\alpha_k + \sum_m \delta(\mathbf{z}_{-i,m}, k)) \tag{5.12}$$

where $\mathbf{z}_{-i}$ denotes all the topic variables in $\mathbf{z}$ except z_i, and $N(i)$ is the neighborhood of node i. Given the samples of the topic variables, we estimate their posterior marginal distribution $P(z_i|\mathbf{a}, \mathbf{x})$ by simply computing their normalized histograms. To be specific, given a set of samples $\{\mathbf{z}^{n,j}\}_{j=1}^J$ for image I^n, we can estimate the posterior distribution $P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n) \propto \sum_j \delta(z_i^{n,j}, k)$.

As we will see in the following section, the posterior of the hidden topic variables $\mathbf{z}$ is also required during the learning procedure. In training, we observe not only the image features, but also their labels at either the coarse or detailed level. Therefore, we want to compute the posterior distribution $P(\mathbf{z}|\mathbf{a}, \mathbf{x}, \mathbf{l}, \mathbf{f})$ or $P(\mathbf{z}|\mathbf{a}, \mathbf{x}, \mathbf{f})$, depending on which type of labeling is observed. We use the same Gibbs sampling approach to estimate these distributions. The conditional distributions used by the Gibbs sampler are

$$P(z_i = k|\mathbf{z}_{-i}, \mathbf{a}, \mathbf{x}, \mathbf{l}, \mathbf{f}) \propto P(l_i, f_i|a_i, x_i, z_i) P(a_i|z_i) \prod_{j \in N(i)} \phi(z_i, z_j)(\alpha_k + \sum_m \delta(\mathbf{z}_{-i,m}, k)), \tag{5.13}$$

$$P(z_i = k|\mathbf{z}_{-i}, \mathbf{a}, \mathbf{x}, \mathbf{f}) \propto P(f_i|a_i, x_i, z_i) P(a_i|z_i) \prod_{j \in N(i)} \phi(z_i, z_j)(\alpha_k + \sum_m \delta(\mathbf{z}_{-i,m}, k)). \tag{5.14}$$

5.4 Parameter Estimation

To build a LTRF model, we assume the following learning scenario: the training data include a small set of image data with detailed labeling, and a large set of image data with only coarse labeling. We will describe two learning procedures for different situations of data availability. The first model learns with the detailed labeling only, while the other model learns with both the detailed and coarse labeling. In the following discussion, we assume the weight w in the compatibility and the hyper-parameter α are fixed. We will determine their values through validation.

5.4.1 Learning with detailed-labeled Data

We first consider learning the model parameters with the detailed-labeled dataset. Let the training dataset be $D = \{(\mathbf{l}^n, \mathbf{f}^n, \mathbf{a}^n, \mathbf{x}^n)\}$. The coarse labeling $\mathbf{f}^n$ can be derived from the detailed labeling $\mathbf{l}^n$ according to the label hierarchy if they are not given. We denote all the parameters in our model as Θ. The learning objective function is the log likelihood of the image appearance and labeling:

$$\mathcal{L} = \sum_n \log P(\mathbf{a}^n, \mathbf{l}^n, \mathbf{f}^n | \mathbf{x}^n; \Theta) \tag{5.15}$$

We maximize the log likelihood by Monte Carlo EM algorithm based on the following lower bound of the objective function:

$$\begin{aligned}
\mathcal{L} &\geq \sum_n \langle \log P(\mathbf{a}^n, \mathbf{l}^n, \mathbf{f}^n, \mathbf{z}^n | \mathbf{x}^n) \rangle_{P(\mathbf{z}^n | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)} && (5.16)\\
&\geq \sum_{n,i} [\langle \log P(l_i^n | f_i^n, a_i^n, x_i^n, z_i^n) + \log P(f_i^n | a_i^n, x_i^n, z_i^n) \rangle_{P(z_i^n | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)} && (5.17)\\
&\quad + \langle \log P(a_i^n | z_i^n) \rangle_{P(z_i^n | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)}] + \texttt{Const}.
\end{aligned}$$

The expectation in the lower bound of the objective function is estimated by sampling the posterior of topic variables. It uses the Gibbs sampling procedure described by Equation 5.14 in Section 5.3. In the M step, we update the model parameters based on the following three separate optimization tasks:

Learning appearance model

The multinomial distribution in the appearance model $P(a|z)$ is computed by

$$\max \sum_n \sum_i \langle \log P(a_i^n | z_i^n) \rangle_{P(z_i^n | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)} \tag{5.18}$$

Taking the derivative of the objective, we have the following updating equation from its stationary point:

$$P(a_i = m|z_i = k) \propto \sum_n \sum_i \delta(a_i^n, m) P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \tag{5.19}$$

Learning classifiers for detailed labeling

The sub-problem of learning detailed label classifiers has the following objective

$$\begin{aligned} \mathcal{L}_d &= \sum_n \sum_i \langle \log P(l_i^n|f_i^n, a_i^n, x_i^n, z_i^n) \rangle_{P(z_i^n|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)} \\ &= \sum_n \sum_i \sum_k P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P(l_i^n|f_i^n, a_i^n, x_i^n, z_i^n) \end{aligned} \tag{5.20}$$

where $P(l_i^n|f_i^n, a_i^n, x_i^n, z_i^n)$ is specified by Equation 5.3. Although direct optimizing $\mathcal{L}_d$ w.r.t. the classifier parameters is feasible, the required normalization in the distribution $P(l_i|f_i, a_i, x_i, z_i)$ complicates learning of the detailed label classifier parameters. However, we notice that the output of a detailed label classifier can approximate the delta function $\delta[l_i \in f_i]$ well if it is initialized by pre-training on the labeled data. That is,

$$\delta[l_i \in f_i] \approx \sum_{l \in f[l_i]} P_{z_i}^d(l|a_i, x_i). \tag{5.21}$$

Using this simplification,, we adopt the following approximate objective in learning those classifiers:

$$\mathcal{L}_d \approx \sum_n \sum_i \sum_k P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^d(l_i^n|a_i^n, x_i^n) \tag{5.22}$$

As the classifiers have separate parameters, we maximize the following weighted log likelihood for the classifier belonging to topic k:

$$\sum_n \sum_i P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^d(l_i^n|a_i^n, x_i^n) \tag{5.23}$$

We implement this sub-learning problem by modifying a gradient-based algorithm. We weight the whole dataset according to the posterior distribution $P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)$; and then train the topic dependent coarse label classifiers using standard classifier learning algorithms. Notice that we need to run the gradient ascent for only a few steps at each iteration to save computation.

Learning classifiers for coarse labeling

Based on the approximation in Equation 5.21, learning the classifiers for coarse labeling is simplified to maximizing the following objective

$$\begin{aligned}\mathcal{L}_c &= \sum_n \sum_i \langle \log P(f_i^n | a_i^n, x_i^n, z_i^n) \rangle_{P(z_i^n | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n)} \\ &= \sum_n \sum_i \sum_k P(z_i^n = k | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^c(f_i^n | a_i^n, x_i^n)\end{aligned} \tag{5.24}$$

Similar to detailed label classifiers, we maximize the following weighted log likelihood for the classifier belonging to topic k:

$$\sum_n \sum_i P(z_i^n = k | \mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^c(f_i^n | a_i^n, x_i^n). \tag{5.25}$$

The learning procedure is implemented by the same modified gradient-based algorithm used in the detailed labeling case.

5.4.2 Learning with coarsely-labeled Data

We now consider the full version of our problem, in which both detailed-labeled data and coarsely-labeled data are available. Let the training dataset have two subsets, $D = \{D_l, D_c\}$. The D_l denotes the image set with detailed labeling, whereas D_c is the image set with coarse labeling only. We learn the model by maximizing a weighted sum of log data likelihood, and objective function is written as:

$$\mathcal{L}_a = \sum_{n \in D_l} \log P(\mathbf{a}^n, \mathbf{l}^n, \mathbf{f}^n | \mathbf{x}^n) + \gamma \sum_{t \in D_c} \log P(\mathbf{a}^t, \mathbf{f}^t | \mathbf{x}^n) \tag{5.26}$$

where γ are the coefficients to control the influence of coarsely-labeled dataset. Note that the second term in the objective can be computed straightforwardly by marginalizing out the detailed label variables $\mathbf{l}$ from the model. Therefore, the coarsely-labeled data will only affect the learning of the appearance model and coarse label classifiers.

We use the same Monte Carlo EM algorithm, maximizing the following lower bound of the

objective iteratively:

$$\mathcal{L}_a \geq \sum_{n\in D_l} \langle \log P(\mathbf{a}^n, \mathbf{l}^n, \mathbf{f}^n, \mathbf{z}^n|\mathbf{x}^n)\rangle_{P(\mathbf{z}^n|\mathbf{a}^n,\mathbf{x}^n,\mathbf{l}^n,\mathbf{f}^n)} \tag{5.27}$$
$$+ \gamma \sum_{t\in D_c} \langle \log P(\mathbf{a}^t, \mathbf{f}^t, \mathbf{z}^t|\mathbf{x}^t)\rangle_{P(\mathbf{z}^t|\mathbf{a}^t,\mathbf{x}^t,\mathbf{f}^t)}$$
$$\geq \sum_{n,i} [\langle \log P(l_i^n|f_i^n, a_i^n, x_i^n, z_i^n) + \log P(f_i^n|a_i^n, x_i^n, z_i^n)\rangle_{P(z_i^n|\mathbf{a}^n,\mathbf{x}^n,\mathbf{l}^n,\mathbf{f}^n)} \tag{5.28}$$
$$+ \langle \log P(a_i^n|z_i^n)\rangle_{P(z_i^n|\mathbf{a}^n,\mathbf{x}^n,\mathbf{l}^n,\mathbf{f}^n)}] + \gamma \sum_{t,i} [\langle \log P(f_i^t|a_i^t, x_i^t, z_i^t)\rangle_{P(z_i^t|\mathbf{a}^t,\mathbf{x}^t,\mathbf{f}^t)}$$
$$+ \langle \log P(a_i^t|z_i^t)\rangle_{P(z_i^t|\mathbf{a}^t,\mathbf{x}^t,\mathbf{f}^t)}] + \texttt{Const}.$$

In the E step, the posterior distributions of the topic variables are estimated by the Gibbs sampling procedure in Section 5.3. In the M step, we update the model parameters in three subcomponents of the model as follows:

Learning appearance model

The multinomial distribution in the appearance model $P(a|z)$ is updated by

$$P(a_i = m|z_i = k) \propto \sum_{n\in D_l} \sum_i \delta(a_i^n, m) P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \tag{5.29}$$
$$+ \gamma \sum_{t\in D_c} \sum_i \delta(a_i^t, m) P(z_i^t = k|\mathbf{a}^t, \mathbf{x}^t, \mathbf{f}^t)$$

Learning classifiers for detailed labeling

The sub-problem of learning detailed label classifiers remains the same as in Equation 5.20, and we maximize the following weighted log likelihood for the classifier belonging to topic k:

$$\sum_{n\in D_l} \sum_i P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^d(l_i^n|a_i^n, x_i^n) \tag{5.30}$$

Learning classifiers for coarse labeling

The objective for learning the coarse label classifiers can be written as

$$\sum_{n\in D_l} \sum_i \sum_k P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^c(f_i^n|a_i^n, x_i^n) \tag{5.31}$$
$$+ \gamma \sum_{t\in D_c} \sum_i \sum_k P(z_i^t = k|\mathbf{a}^t, \mathbf{x}^t, \mathbf{f}^t) \log P_k^c(f_i^t|a_i^t, x_i^t).$$

Table 5.1: The sixteen classes and their proportions in the data set.

Labels	void	building	grass	tree	cow	person	sheep	sky
Proportion	0.20	0.09	0.20	0.08	0.03	0.01	0.02	0.09
Labels	boat	plane	water	dog	car	bike	road	bird
Proportion	0.01	0.01	0.07	0.02	0.03	0.03	0.07	0.01

And we maximize the following weighted log likelihood for the classifier belonging to topic k:

$$\sum_{n\in D_l}\sum_i P(z_i^n = k|\mathbf{a}^n, \mathbf{x}^n, \mathbf{l}^n, \mathbf{f}^n) \log P_k^c(f_i^n|a_i^n, x_i^n) \tag{5.32}$$
$$+\gamma \sum_{t\in D_c}\sum_i P(z_i^t = k|\mathbf{a}^t, \mathbf{x}^t, \mathbf{f}^t) \log P_k^c(f_i^t|a_i^t, x_i^t).$$

we run the gradient ascent for only a few steps at each iteration to save on computation.

5.5 Experimental Evaluation

5.5.1 Data Sets and Feature Extraction

Our experiments use the Microsoft Research Cambridge (MSRC) Image Database [3]. We choose a subset of the whole database that includes 415 detailed-labeled images and 16 different label classes as they have enough data instances in each class. Table 5.1 shows the meaning of those labels and their proportion in the dataset. We randomly split the dataset into four subsets: 10% data is used as the training dataset with detailed labels, 20% data is used for validation, and another 20% data is used as the test dataset. The remaining 50% data are kept as training dataset with coarse labels. The original dataset only have the detailed labeling. To obtain the coarse labeling, we use the label hierarchy in Figure 5.2. We choose the second level such that the coarse labels have three different values: 'void', 'man-made' and 'natural'.

We use the normalized cut segmentation algorithm to build the super-pixel representation of the images, in which the segmentation algorithm is tuned to generate approximately 1000 segments for each image on average. We extracted a set of basic image features, including color, edge and texture information, from each pixel site. For the color information, we transform the RGB values into CIE Lab* color space as in Chapter 3, which is perceptually uniform. The edge and texture are extracted by a set of filter-banks including a difference-of-Gaussian

[3] https://research.microsoft.com/vision/cambridge/recognition/default.htm

filter at 3 different scales, and quadrature pairs of oriented even- and odd-symmetric filters at 4 orientations (0; $\pi/4$; $\pi/2$; $3\pi/4$) and 3 scales. The color descriptor of a super-pixel is the average color over the pixels in that super-pixel. For edge and texture descriptors of a super-pixel, we use the histograms of the pixel features within the super-pixel. Take texture as an example. We first discretize the texture feature space by K-means, and use each cluster as a bin for histograms. Then we compute the normalized histgrams of the texture features within a super-pixel as the texture descriptor. In this experiment, we use 10 bins for edge information and 50 bins for texture information. In total, the image descriptor of a super-pixel has 63 dimensions. The image position of a super-pixel is the average image positions of its pixels.

5.5.2 Model specification

To compute the vocabulary of visual words used in the topic model, we use K-means to group the super-pixel descriptors into 500 clusters after the descriptors are centered and normalized. The cluster centers are used as visual words and all the descriptors are encoded by its word index. Note that the word indices are only used in the topic model; the topic-dependent classifiers use the original descriptors as its inputs. The size of vocabulary is chosen from $100, 200, 500, 800, 1000$ based on the model performance on the validation set.

Across all the experiments, we choose the number of hidden topics as 20 for our models based on the validation set. For the topic-dependent classifiers, we use a set of Multi-layer Perceptons (MLP) with one hidden layer. The classifiers for detailed labeling have 5 hidden units and the classifiers for coarse labeling are linear logistic regressors. Those classifiers are initialized by training them on the corresponding labeled dataset. The appearance model for topics is initialized randomly. The hyper-parameter α is set to $(0.3, ..., 0.3)$ and the weight w in the associative random field is set to 0.95 based on the validation performance. In the learning procedure, we carry out EM for 100 steps. The E step uses 1000 samples to estimate the posterior distribution of topics. In M step, we take 1 step in gradient ascent learning of the classifiers per iteration.

5.5.3 Experiment Design

We carried out two different sets of comparison experiments. The first set is to compare our approach with other discriminative approaches when only the detailed-labeled dataset is available. The baseline systems we used are a super-pixel-wise classifier and a basic CRF model. The super-pixel-wise classifier is an MLP with one hidden layer and predicting label for each

super-pixel independently. Based on validation performance, we choose a MLP with 50 hidden units. In the basic CRF, the distribution of label configuration $\mathbf{L}$ defined as the following form:

$$P_{CRF}(\mathbf{l}|\mathbf{a},\mathbf{x}) = \frac{1}{Z}\exp\{\sum_{i,j}\log\Psi_{ij}(l_i,l_j) + \alpha\sum_{i}\log\Phi_I(l_i|a_i,x_i)\} \tag{5.33}$$

where Φ_I is the output from the super-pixel classifier and Ψ_{ij} is the compatibility function. We trained the CRF model using the pseudo-likelihood algorithm, and label the image based on its marginal distribution of each label variable. The marginal distribution is calculated by the loopy belief propagation algorithm.

The second set of experiments compares our approach with the baseline methods when the coarsely-labeled data are added. To utilize the coarsely-labeled data in the baseline systems, we modify those models as follows. First, we keep the baseline systems trained using the detailed-labeled data and denote them as P_o. Then we train a separate set of baseline systems based on the coarsely-labeled data. The coarse models are denoted as P_n. The final labeling of a super-pixel i by the baseline systems is given by combining two sets of models:

$$(l_i^*, f_i^*) = \arg\max_{l_i,f_i} P_o(l_i|\mathbf{a},\mathbf{x})P_n(f_i|\mathbf{a},\mathbf{x})\delta[l_i \in f_i] \tag{5.34}$$

We evaluate the final performance by two different measures: one is the class accuracy and the other is the F1 measure [60]. F1 measure of a class is the harmonic mean of the precision and recall of the class, which considers both the false positives and miss rates. We only consider the prediction performance on the detailed labeling, and the class 'void' is not counted.

5.5.4 Experimental Results

The performance of LTRF is first evaluated based on learning from detailed-labeled data only. We compared the performance of LTRF to the super-pixel classifier (S_Class), and the CRF model. The average error rates of our model and the baselines are summarized in Figure 5.5. We also report the average pixel-level classification accuracies and F1 measures for each class over 10 different runs in Table 5.2. The overall accuracies and F1 measures in the table are computed by averaging pixel-level classification rates and by averaging class-level F1 measures, respectively.

We can see from the results that the LTRF model achieves almost the same performance as the CRF model with the small detailed-labeled training data, and has higher average accuracy and F1 score than the simple classifier. For individual classes, our model obtains better results on certain classes different from the CRF model. From the F1 scores, we notice that the CRF

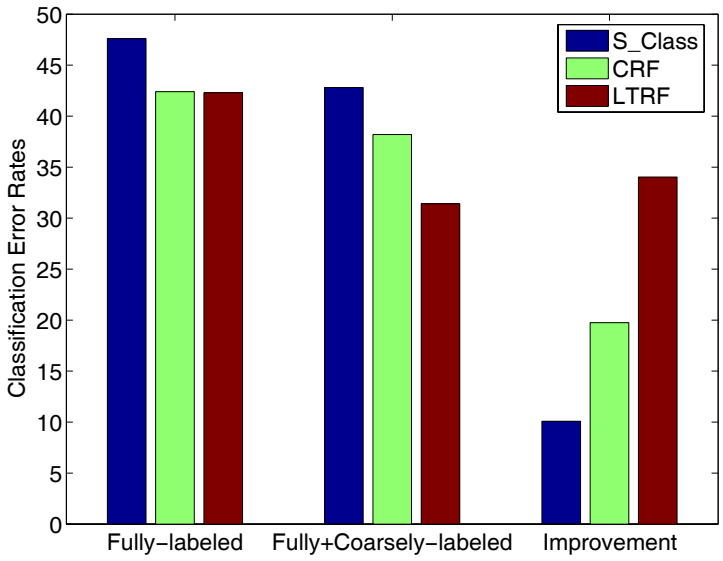

Figure 5.5: Classification error rates for the models: (Left) Based on the detailed-labeled data only; (Middle) Based on all the data; (Right) The accuracy improvement of three models trained with all the data compared to the baseline super-pixel classifier trained with detailed-labeled data only (in percentage). S_Class is the super-pixel level classifier, CRF is the simple CRF model, LTRF is the Latent Topic Random Field.

model is usually superior for the classes with large extent, such as grass and road, whereas our LTRF model is better for man-made objects with complex structure, such as planes and bikes. This may be caused by the fact that the CRF has a strong smoothing effect and the topics are good at picking up co-occurring structures. We notice that due to the imbalance of class sizes, small classes have low recognition accuracies.

Then we compare the performance of LTRF to the baseline systems when they are learned from both detailed-labeled data and coarsely-labeled data. In Figure 5.5, we show the summary of average error rates and the performance improvement of three models over the super-pixel classifier trained with detailed-labeled data only. We also report the best pixel-level classification accuracies and F1 measures for each class over 10 different runs in Table 5.3. For our model, we choose $\gamma = 0.2$ to control the influence from the coarsely-labeled set through validation, and our experiments show that the results are not very sensitive to that value. We can see that the LTRF model with additional coarsely-labeled data provides a significant improvement over itself with detailed-labeled data only, as well as the modified baseline methods. Even for

Table 5.2: A comparison of classification accuracy and F1 measure (in parenthesis) of LTRF model with super-pixel classifier and CRF model trained on detailed-labeled data. The average classification accuracy and F1 measure are at pixel-level and at class-level, respectively.

Label	building	grass	tree	cow	person	sheep	sky	boat
S_class	32.9(30.0)	81.8(83.0)	50.4(54.4)	16.6(21.2)	8.7(8.0)	16.8(22.2)	86.2(78.0)	0.3(0.2)
CRF	40.7(41.0)	82.9(84.2)	57.4(62.2)	22.8(28.4)	16.0(11.0)	35.5(40.6)	83.2(79.1)	0.0(0.0)
LTRF	44.1(43.3)	86.8(83.9)	55.5(62.6)	24.4(25.5)	8.6(11.5)	38.5(45.8)	83.5(78.2)	11.3(2.9)
Label	plane	water	dog	car	bike	road	bird	Overall
S_class	18.1(21.3)	28.3(37.3)	12.1(12.9)	26.9(28.1)	41.3(35.0)	43.1(46.6)	2.7(3.8)	52.4(32.1)
CRF	22.0(27.7)	32.5(42.9)	20.2(19.0)	43.1(42.2)	55.0(45.6)	52.7(53.7)	2.0(2.8)	57.6(38.7)
LTRF	57.0(42.1)	42.1(47.9)	3.5(4.6)	34.1(44.5)	70.2(53.3)	28.7(38.2)	3.5(5.8)	57.7(39.4)

Table 5.3: A comparison of classification accuracy and F1 measure (in parenthesis) of LTRF model with super-pixel classifier and CRF model trained on all data. The average classification accuracy and F1 measure are at pixel-level and at class-level, respectively.

Label	building	grass	tree	cow	person	sheep	sky	boat
S_class	38.7(33.1)	87.2(84.3)	58.2(56.5)	19.9(23.5)	4.5(4.7)	19.7(23.5)	89.4(77.9)	0.3(0.2)
CRF	46.8(45.5)	86.2(85.0)	60.2(62.3)	28.8(32.2)	13.5(11.8)	34.7(38.1)	87.6(80.2)	0.3(0.3)
LTRF	46.3(50.8)	92.9(84.2)	74.7(67.8)	27.2(27.1)	14.0(10.2)	50.6(44.4)	95.2(80.7)	6.4(3.9)
Label	plane	water	dog	car	bike	road	bird	Overall
S_class	20.0(22.1)	31.6(40.1)	10.6(10.9)	26.4(26.2)	46.2(36.3)	51.7(51.5)	4.7(5.4)	57.2(33.1)
CRF	26.0(30.1)	37.2(49.0)	4.0(4.3)	43.3(39.0)	71.6(49.0)	62.9(59.6)	7.6(8.0)	61.8(39.6)
LTRF	74.5(48.8)	56.6(63.2)	5.4(5.4)	64.6(65.2)	81.0(55.5)	43.2(49.2)	4.8(4.2)	68.6(44.0)

individual classes, our model is always better than, or comparable to other approaches, except on a few small classes.

Figure 5.6 shows the distributions of the topics over words, and the average histogram of ground truth labels for each topics. Most topics are more specialized in some subset of labels, and some topics, such as topic 6 and 16, are almost always focused on one object class. In Figure 5.7, we also show some examples of topics in the test images. Note that we take the MAP estimate of the topic variables in the display. While the topic instantiations are slightly noisy, they capture some co-occurred patterns, such as “sky-cow-grass”, “tree-grass” and “building-sky”. Further, we notice that those topics have a weak locality property. In Figure 5.8, we show the outputs of these methods on some test images. From those results,

we can see that LTRF works better than other approaches in terms of accuracy, while the CRF method gets smoother labelings. Those classes with small regions usually have much poorer accuracy than others.

5.6 Conclusion and Discussion

In this chapter, we have presented a hybrid approach that relaxes a limitation of discriminative labeling models due to their reliance on detailed training data. Our method integrates a generative topic model with discriminative label classifiers for image labeling. One main contribution of our approach is that the extended topic model, LTRF, is capable of utilizing both a small set of detailed-labeled image data and a bigger set of coarsely-labeled images. This partially saves the tedious labeling effort in preparing the training data, and presents some promise of the system extending to large databases of images.

The proposed framework is able to capture high-order image contexts, in that the topics model any regularly co-occurring configuration of image features in the entire image. We also incorporate locality into the topic model, such that topics can correspond to blob-like regions in images. Our learning method uses the coarsely-labeled data to impose regularization to the topic model, which would otherwise easily overfit the small detailed-labeled set. Compared to purely unlabeled data, the coarse labels also provide more informative cues to learn the topics for the labeling task.

The results of applying our method to a real-world image dataset suggest that this integrated approach may extend to a variety of image types and databases. The labeling system consistently out-performs alternative discriminative approaches, such as a standard classifier and a standard CRF. An important limitation of our model concerns the bag-of-features assumption. While the topics can potentially detect high-order configurations of features, the model is unable to learn and utilize spatial relations between parts of an object in the labeling procedure. In addition, our attempt is the first step towards using weakly labeled data for learning labeling models. The coarsely-labeled data still require considerable human effort to collect. There are other types of weakly-labeled images that are easier to obtain, including sparsely labeled images or captioned images. We are exploring an extension of the model that can handle those types of weakly-labeled images.

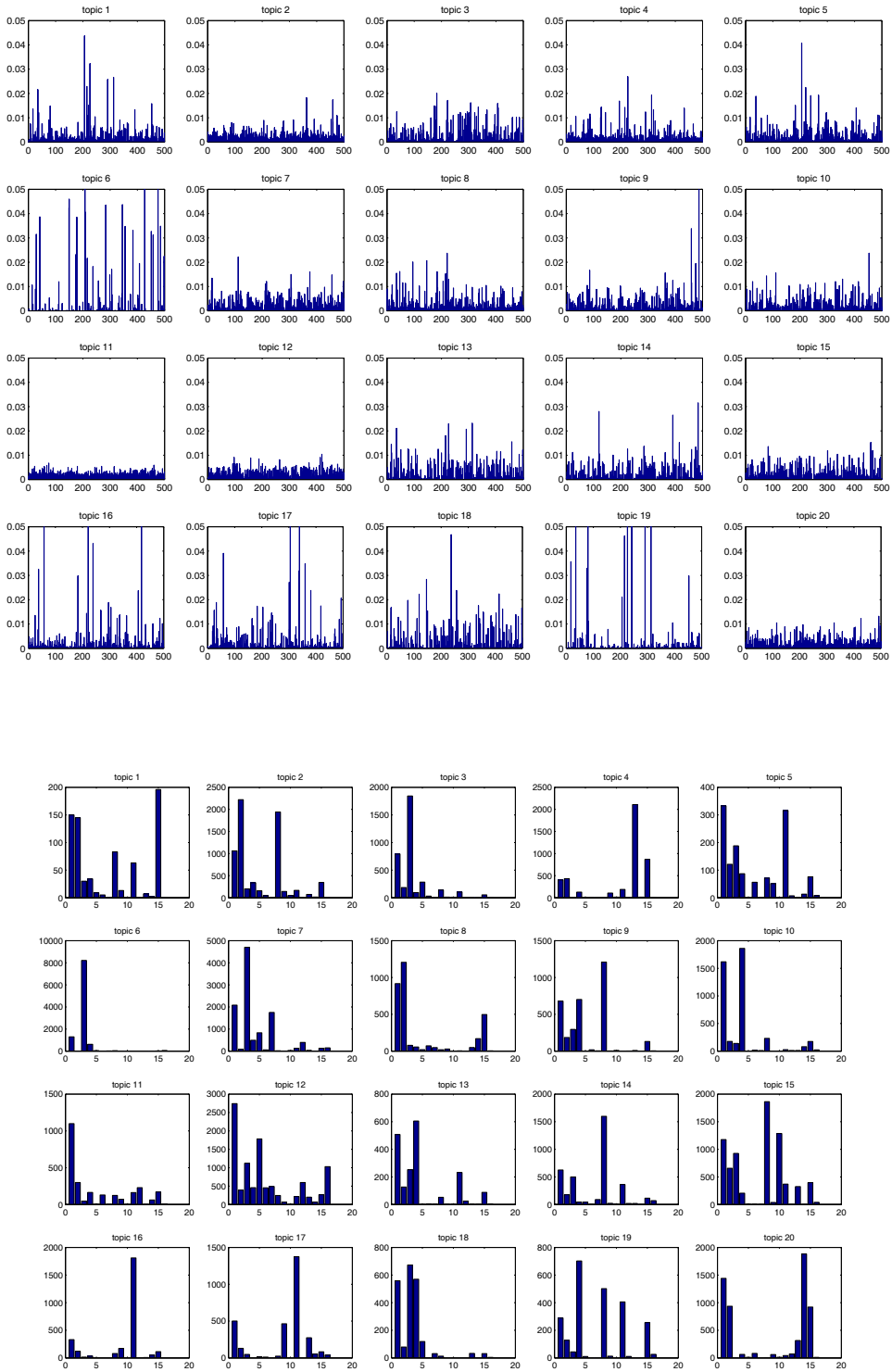

Figure 5.6: Top: the word distributions associated with each topic. Bottom: the histogram of ground truth labels for each topic in the test set.

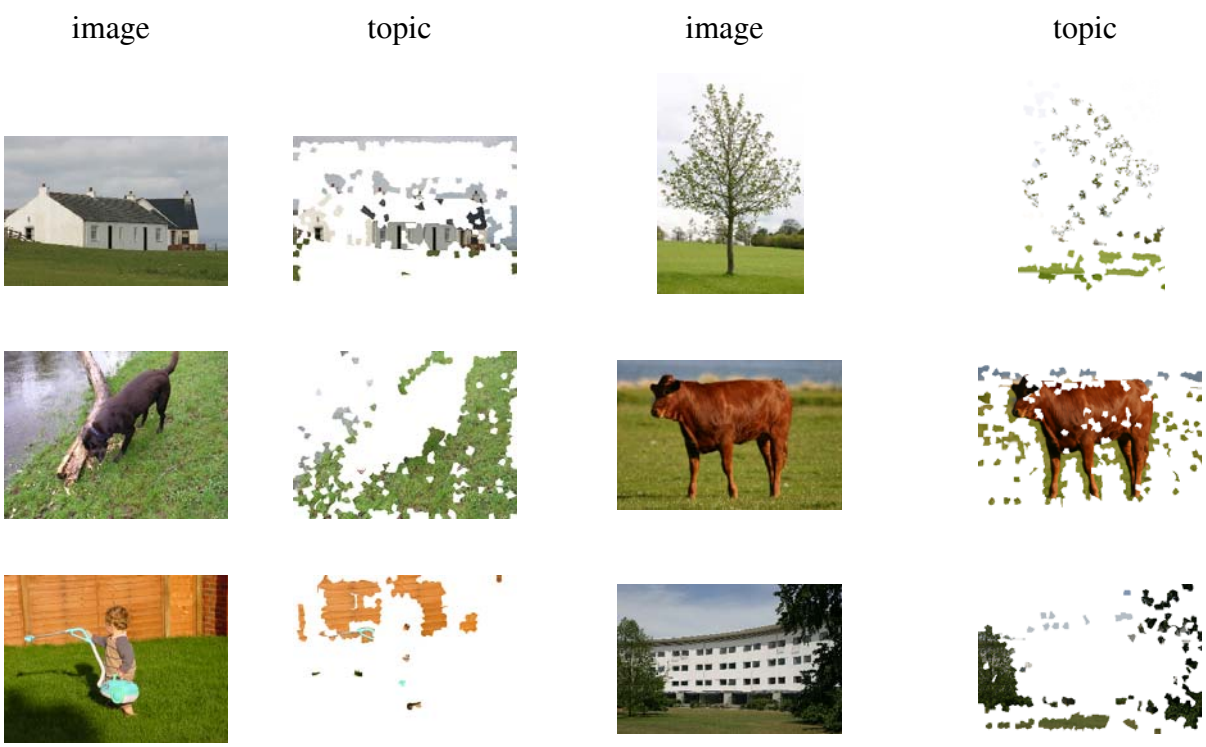

Figure 5.7: Examples of topics in the test images. The regions corresponding to a given topic are masked out.

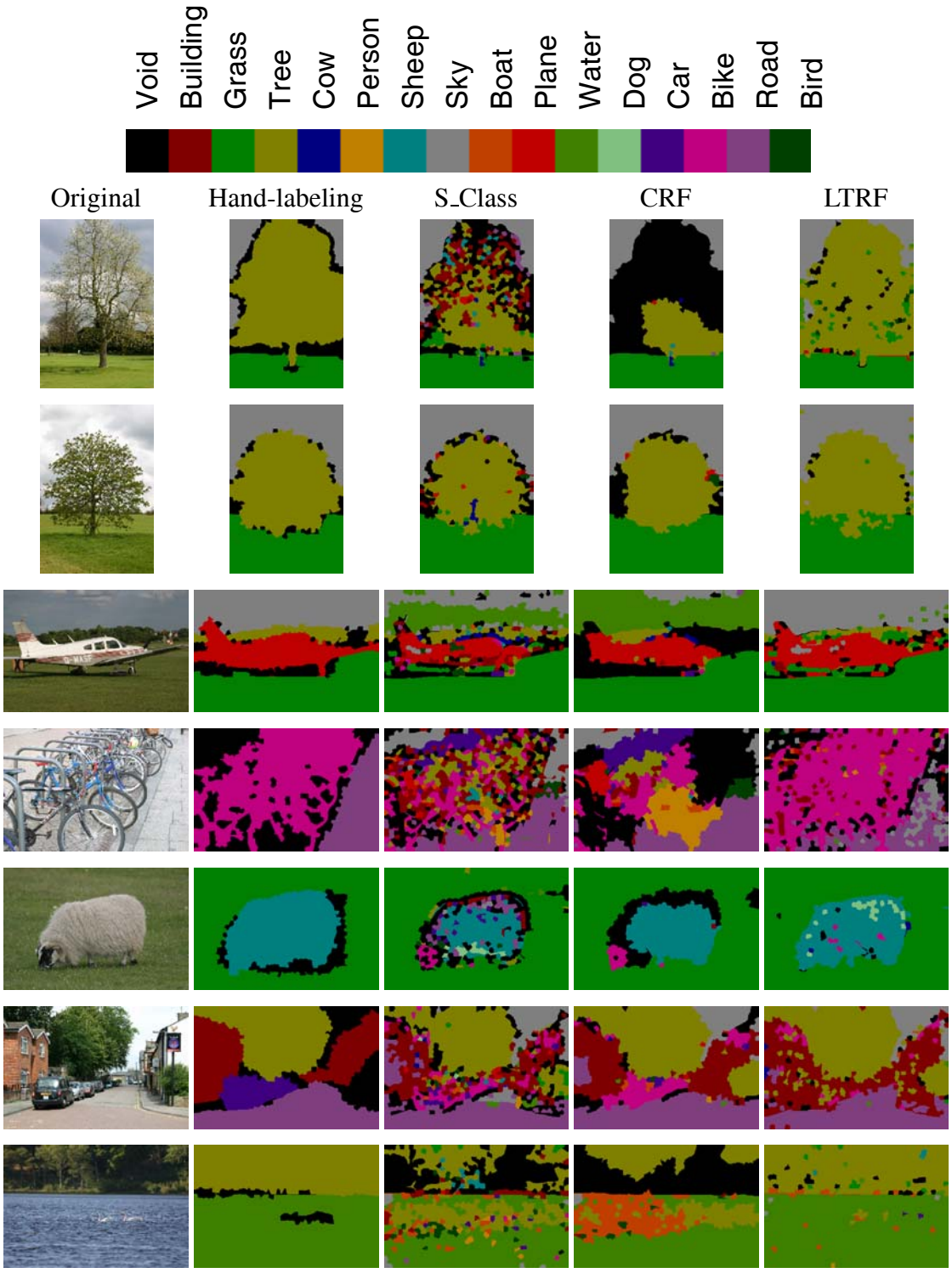

Figure 5.8: Some labeling results for the MSRC datasets, using the super-pixel-wise classifier, CRF, and LTRF. (Top): The color keys for the labels.

Chapter 6

Conclusion

This thesis has developed and implemented a series of structured prediction models for image labeling with object classes. This final chapter summarizes the main contributions of the thesis, and closes with suggestions of possible directions for future work.

6.1 Primary Contributions

Image labeling has wide application in computational vision: many image and object properties can be viewed as a certain type of labels, and our task is to infer those label values from images. A key issue in image labeling is to exploit the context information within each image, as local evidence is insufficient to determine the label value. This thesis addresses two main problems in incorporating context information into the labeling process: 1) what are the efficient representations of contexts for labeling? and 2) how do we specify the context representations for a labeling task from data automatically?

We adopt a structured probabilistic approach to the image labeling problem, in which the context interactions are captured by the dependency between label variables and image features. The advantages of the probabilistic approach come from three aspects: First, the context representation problem can be formulated as choosing suitable model structure and parameterization; Second, the labeling process is implemented as probabilistic inference in the structured model, which implicitly incorporates the context information into labeling. No ad hoc procedure is required to combine the context and local information. Finally, for a given labeling task, we can learn the probabilistic model automatically from a labeled image dataset, without requiring tuning or combining model parts heuristically.

We first consider the situation in which we have fully labeled data for building a probabilis-

tic labeling model in Chapter 3. Based on conditional random fields, we develop a discriminative labeling framework in which the context information is represented by feature functions. The original form of CRFs has several limitations, due to its local connectivity and linear feature functions. Our discriminative framework extends the original CRF and has the following novelties: First, it uses multiscale feature functions to model the image/label structures at several spatial scales. Those feature functions affect the labeling as contexts from local to global levels: some aspects of the contexts concern co-occurrence of objects in the image, while other aspects concern the geometric relationship between objects. Also, it introduces hidden variables into the model to facilitate representing structures and inference. When marginalizing out the hidden variables, the model is essentially a general CRF family with nonlinear feature functions. Furthermore, it develops an efficient learning strategy based on Contrastive Divergence algorithms.

Chapter 4 further develops the discriminative labeling framework that integrates bottom-up and top-down cues. A chief contribution of our model is modularity: images in a database are classified as to their context, and separate sub-models are learned for the different contexts. This modularity presents some promise of the system extending to large databases of images. The top-down cues include categorical information of image regions and are learned in a context-specific manner. The system integrates them with bottom-up cues, which are utilized in several ways: they define a higher-level image representation, super-pixels, which significantly simplify the model structure; they determine probabilities of local boundaries between super-pixels, which are used to constrain and guide labeling; and they enable context classification. With respect to image segmentation, our approach extends top-down cues to include a considerably wider range of object classes than earlier methods.

The discriminative models provide the current state-of-the-art method for labeling problems. However, we usually have to face the data availability issue in learning those models. To address this problem, we consider learning a structured labeling model from coarsely labeled data in Chapter 5. Our method integrates a generative topic model with discriminative label classifiers for image labeling. The generative topics are able to capture high-order image contexts, i.e., any regularly co-occurring configuration of image features in the entire image. Given a topic, the model generates the input data, as well as a topic-dependent probabilistic mapping from input data to the output labels. A main contribution of our model is the resulting extension of a topic model that is capable of utilizing both small sets of fully-labeled image data, and bigger sets of coarsely labeled images. The coarsely labeled data help the system to build a better topic model by regularizing the topics, which would easily overfit the fully-labeled data

otherwise.

The image labeling problem in this thesis integrates the traditional image segmentation and region categorization into a single task, which can exploit the interactions between bottom-up and top-down information in several aspects. With additional weak categorical information, the segmentation process can merge regions with different appearance, and utilize the contextual information from other co-occurring objects. With detailed boundary prediction, the categorization process can not only detect the object classes in an input image, but also localize them at the pixel level. The main difference between our approach and object categorization is that we focus on region segmentation and classification, instead of identifying object instances. Therefore, our methods mainly rely on bottom-up image features and the weak category-level information without explicit object models. The advantage of this tradeoff is that we improve the region segmentation performance at the expense of requiring a moderate amount of high-level information.

On the other hand, as we do not address the issue of recognizing object instances, our models do not explicitly incorporate the invariance properties with respect to common object transformations. In particular, we do not directly cope with the translation, rotation of objects, occlusion, shape deformation and the changes of object scale, camera viewpoint or object articulation. All three models will tolerate certain degrees of those transformations, as long as the image features and label patterns are stable with respect to them. Also, because we mainly use appearance-based features as the model inputs, our models are moderately sensitive to the change of object appearances. To achieve higher object classification accuracy and complete invariance, it is necessary to incorporate more object-specific high-level information into the models. In addition, our approaches heavily rely on the availability of labeled data. Two of our models, mCRF and MoCRF, have to be learned from fully-labeled images, so they are not capable of handling incomplete labels. The third model, LTRF, relaxes that constraint by utilizing both fully-labeled and coarsely-labeled images, which provides a promising way to handle missing labels. Finally, while mCRF is limited in coping with large image database due to its model complexity, both MoCRF and LTRF can potentially scale up to a large number of images, classes and contexts.

6.2 Future Directions

The discussion in the previous chapters has suggested the specific directions in which we would like to extend the three structured image labeling models. In this section, we describe more

general directions for future work.

6.2.1 Flexible Structures in Scene Modeling

The random fields with local connections and coarse global features can only impose weak constraints on scene structures. In many situations, the effect of adding contextual information reduces to a simple smoothing of the predictions given by local evidence. As our labels are related to object categories, it will be more effective to represent object and scene structures explicitly. Random fields with flat structure have limited capacity for such representation. Therefore, we need to explore other model architectures to capture more informative higher-order patterns of contexts.

A natural extension of our context model is to introduce a flexible object model for certain label classes. The object model will deliver the shape information, which is very important for labeling many object classes. However, to add such object-specific information, we need an efficient representation such that we can afford to have many different categories. A potential feasible way is to use popular part-based representation in object recognition, in which different classes can share parts. Modeling relationships between parts can also give more flexibility in modeling the context. Another issue is how we combine the object-specific prior with other weaker region-based priors. Due to deformation and occlusion, we may expect the importance of the object information will be adjusted dynamically according to individual image. This will require the model to have a dynamic structure as in the Dynamic Tree model [1].

We may also want to model the structure of scenes. As several researchers have pointed out, one effective way to incorporate both scene and objects is to build a hierarchical model. The difficulty, however, is to have a model capable of accounting for an unknown number of object categories and object instances. The nonparametric Bayesian models [70] provide a promising framework to handle such situation, but it is still hard to build a full nonparametric hierarchical model from data. Another direction is the grammar-based hierarchial models, which combine a rule-based model with probabilistic features. A challenging issue in such an approach is to induce the grammar automatically from data.

6.2.2 More Informative Image Features

Our implementation of labeling models mostly relies on region-based features, such as color and texture. While region-based features are effective to describe certain natural objects with stable and distinctive appearance, they are insufficient to represent many object classes with

varying appearance but distinctive shapes. In addition, the region-based features are sensitive to shading and illumination changes. Therefore, for image labeling with object classes, it is desirable to extract image features reflecting local shape information. Interest-point features, such SIFT [48], have been widely used in object categorization, and are capable of describing local structures of images. We would like to incorporate such image features into our models. As the interest-point features have non-uniform positions and multiple scales, we need to find a way to combine them with region-based features. A possible solution to this issue is to use all image patches and encode them with the SIFT descriptor.

Although the interest-point features may correspond to intermediate structures of objects, such as parts, most of those features are low-level and each individual is not very informative for labeling. A more distinctive type of features is a composite one formed by a small group of neighboring low-level features. Such composite features have been widely used in specific object recognition systems [48, 31]. We could use a latent topic model to learn those higher-order features. An important issue in learning is to find those distinctive configurations buried in background clutter.

The super-pixel representation simplifies our model structure. However, it also introduces additional labeling errors that cannot be corrected by our models. An interesting problem is to combine the pixel-level labeling with the super-pixel-level labeling, such that we could recover some errors in pre-segmentation. To address this problem, we may want to add more pixel-level features, such as the contour information. Using the continuation property of object boundaries, we could correct some mistakes induced by super-pixels.

6.2.3 Learning Issues in Image Labeling

The Contrastive Divergence algorithm and the approximate learning with Loopy BP have been successfully applied to our models. However, as they may not maximize the learning objectives, there is no guarantee that those methods will work in the same way if we apply them to other cases without further tuning. On the other hand, learning in structured models is usually hard: the exact learning is infeasible for the models with complex structures. We would like to explore other approximations in learning structured models based on variational and sampling techniques. Ideally, we want to maximize a lower bound of the data log-likelihood with an efficient approximate algorithm. An interesting direction is to use the convexity property of objective functions, and to approximate a complicated graphical model by a set of tractable models [81].

Many labeling approaches assume a flat structure of labels, but the label values can form a hierarchical structure. We have introduced a label hierarchy for learning from coarsely labeled data. There are other ways to incorporate such information into the learning process. A straightforward extension is to use the label hierarchy to handle the inconsistently labeled data in learning. For instance, different labelers may assign labels from different levels to image regions belonging to the same class. The label hierarchy can help us to find the constraints between those labels, and use all the labeling information in the learning algorithm. A further issue concerns learning such a label hierarchy from data, either in an unsupervised way or by optimizing the labeling performance.

In real world applications, many annotated image datasets have much weaker labeling information [4]. A common situation is that images are only associated with a set of caption keywords. As there is no correspondence between the labels and image regions, it is difficult to use such information for building a structured labeling model. The previous approaches mainly rely on simple classification models trained by Multiple Instance Learning (MIL) algorithms (e.g., see [91]). MIL algorithms simplify the learning by treating each image as a bag of image features. We would like to extend the MIL algorithms to include the interaction between objects within each image. To start, we may use bottom-up detectors trained by MIL method and add a top-layer of random field to capture keyword interactions. An EM-like learning algorithm can be used to estimate the full model.

6.2.4 Other Applications

In this thesis, we focus on the image labeling problem. However, the probabilistic models we developed can be applied to other domains that requires context modeling. A few examples are listed in the following.

1) Sequence labeling. In sequence analysis, context may also exist at multiple scales. For example, a text annotation task may require information form words, phrases, and sentences to disambiguate a label assignment. Our model can be easily applied to sequence labeling problems by decreasing the dimension of the underlying graph. In [68], we have proposed a variant of mCRF for citation labeling.

2) Information retrieval. The performance of image retrieval will be greatly improved if we can achieve good performance on labeling images. We may provide more detailed information about the retrieved image than just its content, such as the object configuration. The difficulty is to scale up our methods to a large amount of images. We could also apply our methods to

webpage classification in which the interactions between linked webpages are informative for labeling individual ones. The webpage summaries can be used as a weak label for training process.

3) Video processing. Compared to static images, video sequences provide more information about scenes and the objects inside. However, analyzing video data is also a more challenging task as it requires incorporating priors on both spatial and temporal structures. Extending the current framework to video analysis problems, such as event recognition, will be an interesting direction to explore.

4) Automatic navigation. A basic task in vision-based automatic navigation is to recognize different regions in input images so that the vehicle can be kept on the road and avoid obstacles (other vehicles and pedestrians). This problem is highly domain-specific, so we can use image labeling methods to parse the input frames. In practice, we need more efficient inference algorithms for real-time processing. Also, the model should be able to infer the depth information as well as the object categories from images.

Bibliography

[1] Nicholas J. Adams and Christopher K. I. Williams. Dynamic trees: Learning to model outdoor scenes. In *Proceedings of the 7th European Conference on Computer Vision*, 2002.

[2] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings of 20th International Conference on Machine Learning*, 2003.

[3] Yasemin Altun, Thomas Hofmann, and Mark Johnson. Discriminative learning for label sequences via boosting. In *Advances in Neural Information Processing Systems 14*, 2002.

[4] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[5] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of Royal Statistical Society B*, 36(2):192–236, 1974.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proceedings IEEE Workshop of Perceptual Organization in Computer Vision*, 2004.

[8] Charles Bouman and Michael Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, 1994.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1–18, 2001.

[10] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. Interpreting image databases by region classification. *Pattern Recognition*, 30:555–563, 1997.

[11] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Proceedings of the 8th European Conference on Computer Vision*, 2004.

[12] Miguel Á. Carreira-Perpiñan and Geoffrey Hinton. On contrastive divergence learning. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[13] Paul B. Chou and Christopher M. Brown. The theory and practice of Bayesian image labeling. *International Jounal of Computer Vision*, 4(3):185–210, 1990.

[14] William J. Christmas, Josef Kittler, and Maria Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.

[15] Dorin Comaniciu and Peter Meer. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[16] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley&Sons, second edition, 2001.

[17] Xiaojuan Feng, Christopher K. I. Williams, and Stephen Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):467–483, 2002.

[18] David A. Forsyth and Jean Ponce. *Computer Vision - A Modern Approach*. Prentice Hall, 2003.

[19] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International Jounal of Computer Vision*, 40(1):25–47, 2000.

[20] Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using 2-layer networks. In *Advances in Neural Information Processing Systems 4*, volume 4. MIT Press, 1992.

[21] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[22] E.R. Hancock and J.V. Kittler. Edge-labeling using dictionary-based relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):165–181, February 1990.

[23] A.R. Hanson and E.M. Riseman. Segmentation of natural scenes. In *Computer Vision Systems*, 1978.

[24] A.R. Hanson and E.M. Riseman. VISIONS: A computer system for interpreting scenes. In *Computer Vision Systems*, 1978.

[25] Xuming He, Richard Zemel, and Miguel Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[26] Xuming He, Richard S. Zemel, and Debajyoti Ray. Learning and incorporating top-down cues in image segmentation. In *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[27] G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 282–317. MIT Press, 1986.

[28] G.E. Hinton, S. Osindero, and K. Bao. Learning causally linked Markov random fields. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[29] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

[30] R. A. Jacobs, M. I. Jordan, S. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:1–12, 1991.

[31] M. Jamieson, A. Fazly, S. Dickinson, S. Stevenson, and S. Wachsmuth. Learning structured appearance models from captioned images of cluttered scenes. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, 2007.

[32] Michael I. Jordan. *Introduction to Probabilistic Graphical Model*. unpublished draft, 2005.

[33] Sham Kakade, Yee Whye Teh, and Sam Roweis. An alternate objective function for Markovian fields. In *Proceedings of 19th International Conference on Machine Learning*, 2002.

[34] J Kittler and J Illingworth. Relaxation labelling algorithms-A review. *Image and Vision Computing*, 3(4):206–216, 1986.

[35] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639, 2002.

[36] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[37] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000.

[38] M. Pawan Kumar, Philip H. S. Torr, and A. Zisserman. OBJ CUT. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[39] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2003.

[40] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005.

[41] Jean-Marc Laferte, Patrick Perez, and Fabrice Heitz. Discrete markov image modeling and inference on the quadtree. *IEEE Transactions on Image Processing*, 9(3):390–404, 2000.

[42] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, 2001.

[43] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[44] M.D. Levine. A knowledge-based computer vision system. In *Computer Vision Systems*, 1978.

[45] Stan Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., 2001.

[46] L. Liu and S. Sclaroff. Region segmentation via deformable model-guided split and merge. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, 2001.

[47] Nicolas Loeff, Himanshu Arora, Alexander Sorokin, and David Forsyth. Efficient unsupervised learning for localization and detection in object categories. In *Advances in Neural Information Processing Systems 18*, 2006.

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

[49] D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 175–204. Kluwer Academic Press, 1998. [Optional].

[50] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, 2003.

[51] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:530–549, 2003.

[52] Kevin Murphy, Antonio Torralba, and William T.F. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Information Processing Systems 15*, 2003.

[53] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[54] M. Peterson and B. Gibson. Shape recognition contributions to figure-ground organization in three-dimensional displays. *Cognitive Psychology*, 25:383–429, 1993.

[55] Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[56] Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[57] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems 17*, 2005.

[58] X. Ren, C.C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005.

[59] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2003.

[60] Berthier Ribeiro-Neto and Ricardo Baeza-Yates. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[61] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):420–433, June 1976.

[62] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: A database and web-based tool for image annotation. Technical report, MIT AI Lab Memo AIM-2005-025, 2005.

[63] B.J. Schachter, A. Lev, S.W. Zucker, and A. Rosenfeld. An application of relaxation to edge reinforcement. *IEEE Transactions on Systems, Man and Cybernetics*, 7:813–816, 1977.

[64] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL*, 2003.

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

[66] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[67] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005.

[68] Liam Stewart, Xuming He, and Richard S. Zemel. Learning flexible features for conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, To appear, 2007.

[69] Amos J. Storkey and Christopher K. Williams. Image modelling with position-encoding dynamic trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):859–871, 2003.

[70] Erik Sudderth, Antonio Torralba, William Freeman, and Alan Willsky. Describing visual scenes using transformed Dirichlet processes. In *Advances in Neural Information Processing Systems 18*, 2006.

[71] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, 2005.

[72] R. H. Swendsen and J. S. Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.

[73] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems 15*, 2003.

[74] J.M. Tenenbaum and H.G. Barrow. Experiments in interpretation guided segmentation. *Artificial Intelligence*, 8(3):241–274, June 1977.

[75] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2003.

[76] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation inNeural Systems*, 14:391–412, 2003.

[77] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems 17*, 2005.

[78] Luz Abril Torres-Mndez and Gregory Dudek. Reconstruction of 3d models from intensity images and partial depth. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, 2004.

[79] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673, 2002.

[80] M. Wainwright, T. Jaakkola, and A. Willsky. Tree-based reparameterization for approximate inference on loopy graphs. In *Advances in Neural Information Processing Systems 14*, 2002.

[81] Martin Wainwright, Tommi Jaakkola, and Alan Willsky. A new class of upper bounds on the log partition function. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence*, 2002.

[82] Hanna Wallach. Efficient training of conditional random fields. Master's thesis, Division of Informatics, University of Edinburgh, 2002.

[83] Max Welling, Michal Rosen-Zvi, and Geoffrey Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 17*, 2005.

[84] Max Welling and Charles Sutton. Learning in Markov random fields with contrastive free energies. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[85] Max Welling, Richard Zemel, and Geoffrey Hinton. Self supervised boosting. In *Advances in Neural Information Processing Systems 15*, 2002.

[86] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[87] W. A. Wright. Image labeling with a neural network. In *Proceedings of the Fifth Alvey Vision Conference*, 1989.

[88] W. A. Wright, W.P.J. Mackeown, and P. Greenway. The use of neural networks for region labeling and scene understanding. In J.G. Taylor, editor, *Neural Networks*, pages 165–192. 1995.

[89] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–269. Morgan Kaufmann, 2003.

[90] S. Yu and J. Shi. Object-specific figure-ground segregation. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[91] Qi Zhang, Sally A. Goldman, Wei Yu, and Jason Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of the 19th International Conference on Machine Learning*, 2002.

[92] SongChun Zhu, YingNian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.

[93] S.W. Zucker, R.A. Hummel, and A. Rosenfeld. An application of relaxation labeling to line and curve enhancement. *IEEE Transactions on Computers*, 26(4):394–403, April 1977.