1-1-2010

Conditional random field based image and video content analysis

Xiaofeng Wang
Ryerson University

Conditional Random Field based Image and Video Content Analysis

by

Xiaofeng Wang

BEng, Beijing University of Posts and Telecommunications, Beijing, Aug. 1998

Meng, Beijing University of Posts and Telecommunications, Beijing, Apr. 2001

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2010

September 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Conditional Random Field based Image and Video Content Analysis, Xiaofeng Wang Phd, Electrical and Computer Engineering, Ryerson University, 2010

Image and video content analysis is an interesting, meaningful and challenging topic. In recent years much of the research effort in the multimedia field focuses on indexing and retrieval. Semantic gap between low-level features and high-level content is a bottleneck in most systems. To bridge the semantic gap, new content analysis models need to be developed. In this thesis, algorithms based on a relatively new graphical model, called the conditional random field (CRF) model, are developed for two closely-related problems in content analysis: image labeling and video content analysis. The CRF model can represent spatial interactions in image labeling and temporal interactions in video content analysis. New feature functions are designed to better represent the feature distributions. The mixture feature functions are used in image labeling for databases with nature images, and the independent component analysis (ICA) mixture function is applied in sports video content analysis. The spatial dependence of image parts and the temporal dependence of video frames can be explored by the CRF model more effectively using new feature functions. For image labeling with large databases, the content-based image retrieval method is combined with the CRF image labeling model successfully.

Acknowledgments

It is my great pleasure to thank the many people who made this thesis possible.

I want to thank the Department of Electrical and Computer Engineering of Ryerson University for giving me the opportunity to start this thesis and to conduct the necessary research work that I enjoy.

I can not overstate my gratitude to my Ph.D. supervisor, Prof. Xiao-Ping Zhang. His enthusiasm, knowledge, encouragement and patience inspired me. He also provided good advice, good company, and lots of new ideas. I would not have finished this thesis without his help.

The greatest thing about this department is the helpful environment. I am deeply indebted to Prof. Sridhar (Sri) Krishnan, Prof. Alagan Anpalagan, and Prof. Ling Guan for their mentoring and kind assistance.

I am deeply appreciative of Epson Canada for providing me with financial support and research opportunities. I would also like to thank Research In Motion (RIM) for providing me with a Co-op opportunity in video coding. In particular, I would like to thank Mr. Ian Clarke and Mr. Yury Yakubovich of Epson Canada, Dr. Hui Zhou of Vixs systems, Dr. Longji Wang and Dr. Dake He of RIM for their valuable advice and help. Special thanks should be given to my colleagues of RIM, Dr. Guixing Wu, Dr. Xiaohan Wang, Dr. Xiang Yu, Dr. Tianyin Ji and others who helped me during

my work term.

I am indebted to my student colleagues of this department and all over the campus for providing a stimulating and fun environment in which I can learn and grow. I am especially grateful to my labmates (Junfeng Jiang, Tom Tsui, Alon Shalev, Ran Wu, Zhicheng Wei, Meng Zhang, Haibei Wu, Hui Zha and many others) for having shared many experiences, thoughts, encouragement and laughs with me throughout these years. Special thanks to Ning Zhang of Ryerson Multimedia Lab for our fruitful discussions and helping me collect sample videos. I am grateful to Diana Ning and all my colleagues of international student services for making my initial school years joyful.

I feel a deep sense of gratitude for my parents, my brother and my wife, whose love, encouragement and support are backbones of my life. One of the great experiences of my PhD was the birth of my son Matthew, who provided endless happiness and inspiration for the completion of my thesis and many wonderful things.

Acronyms

BN	Bayesian Network
BP	Belief Propagation
BW	Baum-Welch
CBIR	Content Based Image Retrieval
CCTV	Closed-Circuit Television
CLL	Conditional Log-Likelihood
CRF	Conditional Random Field
EM	Expectation Maximization
GMHMM	Gaussian Mixture Hidden Markov Model
GMM	Gaussian Mixture Model
HCRF	Hidden Conditional Random Field
HMM	Hidden Markov Model
ICA	Independent Component Analysis
ICAMHCRF	ICA Mixture Hidden Conditional Random Field
ICAMHMM	ICA Mixture Hidden Markov Model
IID	Identical Independent Distribution
KL	Kullback-Leibler
LBP	Loopy Belief Propogation

LM	Leung-Malik
LMM	Laplacian Mixture Model
LOG	Laplacian of Gaussian
MCMC	Markov Chain Monte Carlo
ME	Maximal Entropy
MRF	Markov Random Field
NN	Neural Network
PCA	Principal Component Analysis
RGB	Red, Green and Blue
SGD	Scholastic Gradient Descent
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine

List of Important Symbols

$\tilde{(\cdot)}$	Empirical distribution of a variable
$\hat{(\cdot)}$	Estimation of a function or value
$(\cdot)^*$	Estimation of a function or value
a	Mixture coefficient
b	Scale parameter of laplacian distribution
C	Clique
K	Number of features
M	Number of mixture components
N	Number of all nodes in a graph
V	Set of nodes
E	Set of edges
$G(V, E)$	Graph with nodes V and edges E
S	Independent source of ICA
μ	mean
σ^2	Variance
$\boldsymbol{x}$	Random observation variable of whole graph in CRF
$\boldsymbol{y}$	Random label variable of whole graph in CRF
$\boldsymbol{h}$	Random hidden state variable of whole graph in HCRF

$\mathcal{X}$	All possible observation sets of node observation
$\mathcal{Y}$	All possible label sets of nodes
$\mathcal{H}$	All possible hidden state sets of node label
$\mathcal{T}$	All training samples
$\mathcal{X}^N$	Set of observations at all N nodes
$\mathcal{Y}^N$	Set of labels at all N nodes
$\mathcal{C}$	All possible cliques of a whole graph in CRF
x	Instance of observation variable of a single node in CRF
y	Instance of label variable of a single node in CRF
A_c	Accuracy of classification
T_p	True positive
T_n	True negative
F_p	False positive
F_n	False negative
$E(\cdot), < \cdot >$	Expectation function
$f_i(\cdot)$	Feature function corresponding to the association potential at the label site i
$f_{ij}(\cdot)$	Feature function corresponding to the interaction potential between the current site i and its neighboring site j
$m(\cdot)$	Message function
$b(\cdot)$	Belief function
$Z(\cdot)$	Normalization factor
$\mathcal{L}(\cdot)$	Likelihood function
$\Psi_S(\cdot)$	Factor of subset
$\Psi_C(\cdot)$	Factor of the clique C

$\alpha(\cdot)$	Forward variable
$\beta(\cdot)$	Backward variable
λ	Parameters of maximum entropy model
θ	Parameters of HCRF
w	Parameters of CRF or HCRF
$\varphi_i(\cdot)$	Association potential between the observation data and the label site i
$\psi_{ij}(\cdot)$	Interaction potential between current site i and its neighboring site j
$\Psi_t(y_t, y_{t-1}, \boldsymbol{x})$	Factor of linear chain CRF or HCRF
$\frac{\partial \mathcal{L}(\mathcal{T}, \boldsymbol{w})}{\partial w_k}$	Partial derivative of the likelihood function with respect to w_k
η	Learning rate

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Image and Video Content Analysis

Image and video content analysis is an interesting and challenging topic in the multimedia signal processing field. In recent years, much of the research effort focuses on multimedia indexing and retrieval. The main driving force of image and video indexing and retrieval systems is their wide applications in many signal processing and computer vision fields. Some example applications of indexing and retrieval and content analysis related research are listed as follows.

- *Image and video search.* With the advance of World Wide Web and Internet search engines such as Yahoo and Google, the indexing and retrieval of large amount of information becomes more and more important. During the past two decades, content-based image and video retrieval dominated the multimedia signal processing research. The motivation is that the traditional keyword-based search is no longer suitable for large amount and more varieties of multimedia

content. A comprehensive content description of most content is nonexistent. The current state-of-art algorithms in computer vision and signal processing could not generate the kind of keywords automatically by computers. The manual labeling is time consuming and subjective. The image and video search systems using content-based information instead of keywords become the center stage of multimedia research.

- *Image and video editing.* New kinds of media content come up nearly every year, and new media interfaces encourage personalized and professional service for ordinary people. The public needs personalized media now more than any other time in the history. Personalized editing of multimedia content is one important application of content analysis.

- *Medical multimedia research.* To effectively and efficiently detect a kind of disease with new media tools is the goal of multimedia medical research. In medical applications, it is important to understand the images or other kinds of media for computer aided disease diagnosing. The new technology provides much more information for medical professionals. How to help them store and manage this information is critical to the advance of medical research. Medical media search and indexing is interesting and beneficial to human beings.

- *Security video search.* Video surveillance is in the center of research in safety and security due to its high importance applications. Usually, humans have to monitor the closed-circuit television (CCTV) screens all the time and often they need to pay full attention for 24 hours a day. It would be desirable to have surveillance systems to do this task automatically. Therefore abnormal event

detection is one application oriented research in video content analysis.

- *Robot vision and consumer electronics.* With the development of three-dimensional (3D) TV, the new media format beyond two dimensions gains the interest of both professionals and general audience. Interesting applications such as room decoration systems with 3D reconstruction are also appealing. The 3D signal processing needs the content information to generate useful results automatically.

Up to now, most systems have limit performance by using only a few low-level features such as color, texture, shape, and motion. There is a huge semantic gap between low-level features and high-level content. The most efficient and straightforward way to narrow the semantic gap is to better understand the image and video content. It is the task of content analysis.

Image and video content analysis is a combination and interconnection of many subjects such as machine learning, image processing, natural language processing and computer vision. Of primary interest in this thesis are two closely-related problems in content analysis: image labeling and video content analysis. Fig. 1.1 shows two examples of the problems that will be discussed in this thesis.

- **Image labeling**. Image labeling aims to automatically segment and recognize objects or regions in images [85, 34, 11, 47, 68]. Different from image segmentation, labeling is a high-level vision, which not only segments the images but also provides meaningful class labels to image pixels. For example, the indoor image labeling is to classify every pixel of an indoor image into a semantic category such as "floor","wall" and "ceiling". It is very useful in image annotation and further operations such as robot vision and 3D scene reconstruction [19, 84, 38].

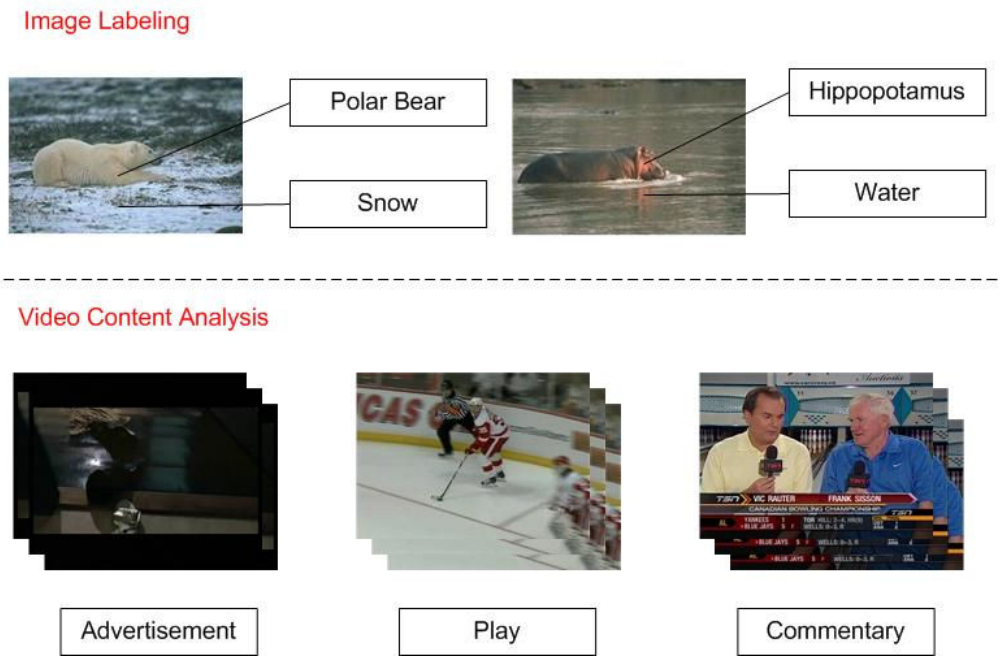


Figure 1.1: An illustration of image and video content analysis problem.

- **Video content analysis**. Video content analysis is to find meaningful structure and patterns from visual data for the purpose of efficient indexing and mining of videos. Video analysis tasks include video parsing, content indexing, abstraction, and representation. Video parsing is to segment video to different levels of segments. The early works focus on low-level parsing, *i.e.*, the video shot boundary detection [105, 31, 32]. An important and key technology in the process of content indexing, abstraction, and representation is content classification. After segmentation, camera shots need to be labeled, given meaningful names, and classified into different categories. One kind of video classification is video event classification, which classifies shots into different events. In recent years, event classification in sports videos has become a popular research topic. Our main interest is to automatically segment and recognize shot level events

or highlights in video sequences.

Although image labeling and video content analysis are two different problems, they share common methodology and philosophy. Both of them are fundamental problems of multimedia content analysis. For the image labeling problem, it is helpful to know the content around the object we plan to recognize. For example, when there is snow in an image, it is more likely to find a polar bear than a hippo. The images are composed of spatial coherent areas. For the video content analysis problem, videos are a series of images. The images before or after a current scene is helpful to determine the class that the current image belongs to. The videos are composed in a temporary coherent manner. Content recognition seems simple and straightforward for humans, however, it needs a lot of effort to make computers finish this task automatically. If images or videos could be well segmented effectively, one could have a better chance of recognizing the objects or events in the scene. On the other hand, if objects, events and their properties were known, one could segment the scene with better accuracy. The content ambiguity of both problems is the main difficulty of ongoing research. In this thesis, the conditional random field (CRF) model in machine learning is used to tackle both problems, by taking spatial structure of images and coherent temporal dynamics of videos into account.

1.2 Background Work in CRF for Image and Video Content Analysis

The CRF model was first proposed by Lafferty for labeling 1D sequential data such as speech [52]. It is a discriminant probabilistic graphical model which addresses

the limitations of a hidden Markov model (HMM). The CRF model finds successful applications for classifying structured data in various applications such as speech recognition [52, 29], diagram labeling [74], image labeling [85, 34, 28, 51], object recognition [89], video content and event analysis [95, 78], and image content analysis (recognizing manmade structures) [51, 50]. The CRF model incorporates neighborhood interactions in the labels and observed data, so has many advantages over traditional generative models. In most real and difficult cases the CRF model can model both spatial and temporal structures with better accuracy than other existing models because of its maximum entropy equivalence property [27].

There are spatial interactions in image labeling and temporal interactions in video content analysis. The CRF model is a powerful and efficient graphical model which can represent spatial or temporal interactions in these two problems. Also the CRF model has training and discriminant advantages. When using CRF to solve semantic content analysis problems, new models corresponding to different properties of different content analysis problems should be derived. This thesis does not discuss a general solution for content analysis. Several image and video analysis problems are formulated using a common CRF graphical model but with different feature functions. New semantic content analysis algorithms are proposed for automatic processing of images and videos. Nature images and sports videos exhibit strong spatial and temporal dependence separately and modeling these dependencies using modern machine learning and pattern recognition algorithms is crucial to achieve a good understanding of these contents.

1.2.1 CRF in Image Labeling

In image labeling, an image is first divided into regular grids such as pixels or rectangular regions, then features of these grids are extracted. The features may include color, texture and shape. The 2D grid is a graph where probabilistic graphical models could be applied. The current state-of-art CRF image labeling methods includes several PhD thesis and papers [85, 34, 28, 51]. In [51], the two class image labeling problem with CRF is presented and it is the baseline CRF in our discussion. In [85], the recognition problem is formulated using CRF with many kinds of features and potential functions. The complex multiscale CRF is discussed in [34]. The relative location information is added in [28]. In image labeling, we discuss two problems, the design of potential functions and the labeling of large databases.

Potential Functions in Image Labeling

The CRF model, which is a discriminant probabilistic graphical model, is built on 2D grid features and labels for training with association and interaction potential functions [51, 50]. The association potentials represent the likelihoods of the node label given the observation of the current node. The interaction potentials are the likelihoods of the interaction between neighboring grid labels given the observation of neighboring grid features. Both potentials may include many types of nonstructural classifiers depending on applications and types of feature data structures. Usually the potential functions in CRF are selected empirically and hand-tuned to achieve better performance.

In image labeling and object recognition, the potentials are designed using arbitrary discriminant classifiers such as logistic [51], probit [74], boosting [89], neural

network [37][34], and the combination of many types [85]. But these forms of potentials generally need hundreds of features to have reasonable results which makes the training and inference difficult. It is the responsibility of the CRF training algorithm to find the weights of different potentials. Sometimes the training fails to find the right parameters because the initial point is not well chosen. How to design these potential functions is essential to CRF image labeling.

Image Labeling for Large Labeled Databases

For large databases the problem becomes more complex. In image labeling for small or specific controlled databases, researchers usually set up a database with several classes under certain conditions. The performance is evaluated using specific databases, for example, the MSRC benchmark [3] with 21 classes and the Corel and Sowerby [2] with 7 classes. The generation of the database is generally troublesome and the hand labeling process is time consuming. One problem is that in the real world there are no such specific databases with limited classes to be used for classification and building probabilistic models. When a very large labeled database such as Labelme [83] is used, the image labeling result would not be effective because of the variety of images, class labels and label ambiguities. Therefore, the key problem is how to handle large labeled databases for the training and labeling with these ambiguities.

To reduce the content ambiguities, there is a growing trend to combine top-down information and bottom-up labeling. The top-down means using the information from high-level vision, for example, the object and the scene to infer pixel labels. The bottom-up means the pixel labeling process from low-level raw pixel features. Since the bottom-up is not accurate enough for image labeling, recently top-down

cues such as object information is incorporated to improve performance. Even for small database with only several categories the top-down content information is often used. Usually a probabilistic model is built for the content or integrated in the labeling model such as [35, 90, 56, 9, 30].

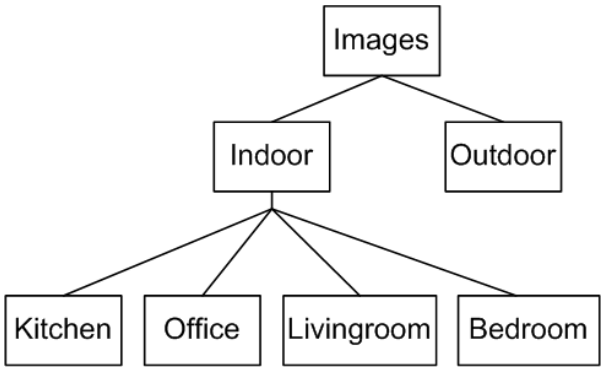


Figure 1.2: An example of hierarchical top-down model.

An example of hierarchical probabilistic model is shown in Fig.1.2. The model divides the images into concepts. For example the images are classified into two groups, indoor and outdoor. The indoor images could be further divided into several concepts: office, living, kitchen and so on. The concepts are grouped based on a tree-like structure. The concepts are then modeled with Bayesian or random field probabilistic models. In [35], top-down category-based information is used to help merge bottom-up segments into object components. The concept is reflected in the content dependent mixture of CRF model [52, 51]. The method is highly complex and used for small databases with a limited class size. The authors of [90] simplify the top-down approach with a single CRF by including the global features. Papers [56] and [9] combine the top-down example-based information and bottom-up segmentation

information. In [30] object arrangement rules are adopted as top-down information in the Bayesian model. Due to their purpose for limited controlled labeling databases, these models are still far from being used for large labeled databases. For large databases, the methods relying on content probabilistic modeling such as Fig. 1.2 do not work due to the complexity of the model structure and parameter learning. How to reflect top-down information in CRF image labeling is important for image labeling of large databases.

1.2.2 CRF in Video Content Analysis

Most of the previous research in video content analysis was based on video state models utilizing probabilistic graphical models such as a hidden Markov model (HMM) [53]. There is a large amount of literature that discusses the HMM in video analysis algorithms, *e.g.*, [100, 99, 60, 97, 40, 45, 6, 43, 25, 46, 101, 103, 14, 59, 102, 22, 26, 67]. In [99], unsupervised classification based on color ratio and motion in soccer domain is discussed and the observation model is Gaussian mixture. In [60], the audio features such as applause and cheering are modeled as HMM. In [14], baseball highlights are modeled as HMM using various kinds of features. It is extended to the maximum entropy model [26] which puts several shot features together for classification and does not use the useful temporal graph information. The hierarchical HMM presented in [67] is a more complex HMM model. In [107], based on the non-Gaussian property of visual features the ICA mixture [54] observation model is applied in HMM for golf video event classification. As mentioned in the HMM tutorial paper [76], there are certain limitations of HMM, the conditional independence of observations, the form of observation distribution and the Markov chain interaction.

CRF, which relaxes the conditional independence assumption of HMM, is more suitable to video content analysis tasks. But the full labeling of training sequences' states prevents it from applying to event analysis of videos directly. To solve this problem hidden conditional random field (HCRF), which is proposed by Quattoni for object recognition recently in [75] can be used. HCRF is a general extension of HMM which relaxes the independent observation and generative assumption. It has been applied to phone classification [29], gesture recognition [95] and meeting segmentation [78]. Although mentioned in the book [27, 32], it has not been used in sports video analysis.

1.3 New Approaches

To interpret the scenes contained in images or videos as a collection of meaningful entities is the fundamental task of content analysis. It is to interpret the information in the scene with different levels of meanings. For example in video analysis, we group frames to shot, events and stories from bottom-up, segment video to different levels, and recognize each levels from top-down. One may also be interested in understanding different regions of a single frame, *e.g.*, a person, a football or any thing in the scene which is the task of image labeling and object recognition. The problem is both an interesting and a challenging one.

Images are not random collections of pixels and videos are not random collections of image frames as well. To analyze these contents, the contextual information in the form of dependencies should be used. It is the main idea of this thesis. Various discriminative and generative models are discussed and the CRF model is selected for the image and video content analysis.

Based on analysis of both the image labeling and object detection problems, we apply the CRF model with new potential functions to image content analysis. For image labeling with small specific databases we use the mixture functions to model the features. We analyze the distributions of features of nature image parts and use mixture (Gaussian Mixture or Laplacian Mixture) to approach these distributions which reduces the number of features needed in CRF image labeling. The main approach is to use new local potential functions in the discriminant manner. The advantages are less training effort and better accuracy. With large databases we successfully combine CBIR and CRF. Since content labeling ambiguities exist in large labeled databases, we propose using CBIR to choose content similar images as the new database used for labeling. The top-down information is reflected in the CBIR process. The advantages of both CBIR and CRF are integrated to deal with the image labeling problem with large databases.

Unlike HMM, CRF is less studied in video content analysis. For video event analysis, based on previous work in video content analysis, we formulated a new HCRF model for event detection. HCRF is better than HMM because of its ability to model the temporal content dynamics more efficiently. The main reason is that CRF relaxes some strong assumptions of HMM model. The relaxations provide accuracy advantage of HCRF in video content analysis. To further enhance the HCRF framework for video analysis tasks, we model local observations as ICA mixtures.

In this work we present new CRF models with new feature functions to model interactions in images and videos. We take a modern approach using training samples (supervised learning) to build graphical models for image and video content analysis.

1.4 Main Contributions

New contributions are summarized as follows:

1. A novel mixture CRF model to improve the image labeling accuracy according to the feature analysis of small databases with nature images such as corel.

2. For a large image database such as LABELME, a new combination of CRF and CBIR to tackle the top-down learning of image content analysis.

3. A novel video analysis framework with hidden CRF (HCRF) model based on analysis of sport video frame features and their temporal structures.

1.5 Thesis Outline

As shown in Fig. 1.3 this thesis is structured as follows:

- The first part of the thesis (Chapter 2) which provides the background of the CRF discussion of this thesis consists of an introduction, review, and brief theory description of the CRF training and inference method, the insight of CRF's maximum entropy equivalence and modeling interaction in content analysis using CRF.

- In Chapter 3, we begin with image labeling of nature images. Based on an analysis of traditional image labeling models and nature image features, a new mixture CRF model is presented for supervised image labeling task with small database.

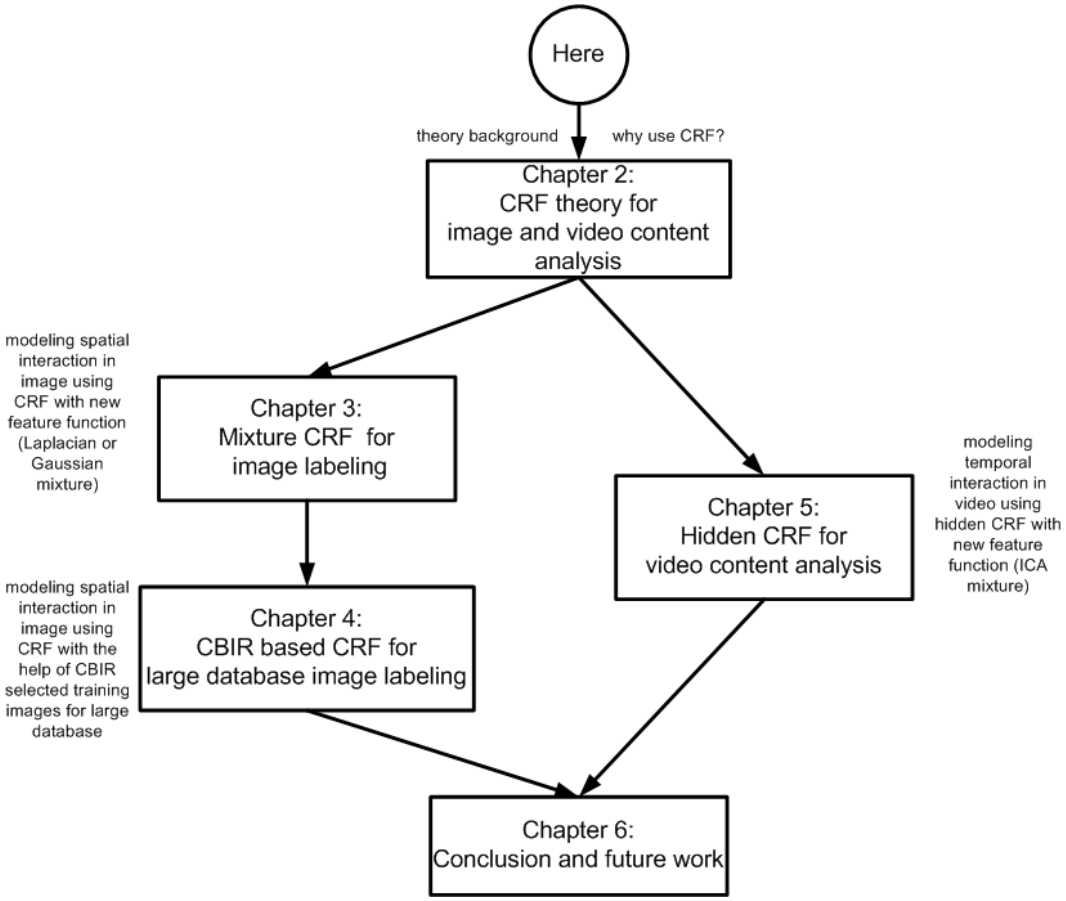


Figure 1.3: Road map of this thesis.

- In Chapter 4, we extend the image labeling discussion to the large and uncontrolled dataset which is often encountered in real circumstances. A new approach combining the advantages of CBIR algorithm and CRF model is presented for this challenging task.

- In Chapter 5, we further our discussion to video content analysis. Hidden CRF model with new mixture feature functions are given for better modeling the coherent structure of the video content.

- Finally in the last chapter, the main contributions of this thesis are summarized

with discussions on the challenges of image and video content analysis and their future research directions.

Chapter 2

CRF Model for Content Analysis

Problems of content analysis are usually solved using a two-step methodology:

1. Problems are analyzed and formulated using some rules or probability based optimization criteria.

2. Optimal solutions that best meet the criterion are found by function optimization or probabilistic reasoning.

The rule-based optimization, which is often called regularization, was originally developed by statisticians trying to fit models to data. The drawback is that the regularization methods may severely limit the solution space. The probabilistic analysis is performed on the probabilistic criterion to find the optimal solution. For example, in the video shot classification problem, the criterion is the probability of a shot class given the shot feature observations. Higher probability means there is a better chance that the features fit the class. For image and video content analysis, the existence of noise and uncertainty makes probabilistic models better suit the task.

Suppose $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, $\boldsymbol{x}_i \in \mathcal{X}$, are N input variables which represent our observation knowledge, and $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$, $\boldsymbol{y}_i \in \mathcal{Y}$, are their corresponding classes which we wish to predict, where $\mathcal{X}$ is all possible observations and $\mathcal{Y}$ is a set of finite classes. In supervised learning, $\boldsymbol{y}$ is known for the training set and unknown for the testing set. The problem of content analysis could be formulated as finding the probability of class variables given observations $P(\boldsymbol{y}|\boldsymbol{x})$.

The organization of this chapter is as follows: First, the graphical models used in machine learning and content analysis are discussed in Section 2.1. Second, we focus on conditional random field (CRF) model formulation which is more suitable to deal with complex image, video content analysis problems and discuss its maximum entropy equivalence which leads to the success of the model in Section 2.2. Third, the training and inference methods of CRF model and CRF with hidden states are presented in details in Section 2.3 and Section 2.4. Finally, we discuss modeling the spatial and temporal dependence in image and video content using the CRF model in Section 2.5.

2.1 Graphical Models

2.1.1 Categories of Probabilistic Models in Machine Learning

Probabilistic models in machine learning could be divided into different categories based on different criteria as shown in Table 2.1. Based on whether structural information is used, machine learning probabilistic models could be roughly divided to two categories: nonstructural and structural methods.

Table 2.1: Classification of probabilistic models in machine learning.

	Nonstructural	Structural (Graphical)
Generative	Naive Bayes Gaussian Mixture[65][10][72] Laplacian Mixture[4] ICA Mixture[54]	Bayesian Network [42][86][15] Markov Random Field [24][58][77] Hidden Markov Model [76]
Discriminative	Neural Network [21][62] Support Vector Machines [91][17][87]	Conditional Random Field [52][51] Hidden Conditional Random Field [75]

Nonstructural models assume no correlation between parts of variables, *e.g.*, assume observations are identical and independently distributed (IID). It is an appropriate assumption in some applications, for example in predicting the weights of a group of people based on their heights. Non-structural methods include clustering, neural networks, support vector machines and so on.

Structural methods refer to graphical models. The graphical models are highly advantageous by using diagrammatic representations of probability distribution for applications which involves spatial or temporary interaction between class variables and observations. Because images are composed of spatial coherent parts and videos are composed of temporary coherent frames, their structural information provides additional useful information in their content analysis.

Based on the probability expression of the problem solution, the probabilistic models could be classified into discriminative models and generative models. A generative model is a full probabilistic generative process of all observations from the class variables, while a discriminative model targets only class variables conditional on the observations.

The generative models are based on the Bayes rule formulation $P(\boldsymbol{y}|\boldsymbol{x}) \propto P(\boldsymbol{y}, \boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})$, [8]. The model captures the causal process by which the observations are

generated by class variables. The generative models such as the hidden Markov model (HMM) are widely used traditionally because the conditional probability $P(\boldsymbol{x}|\boldsymbol{y})$ is easier to model than the posterior probability $P(\boldsymbol{y}|\boldsymbol{x})$ and there are well-established and well-engineered algorithms such as the expectation-maximization (EM) algorithm [20] and Baum-Welch (BW) algorithm [98]. However, there are several disadvantages to these generative models. To make the model tractable, the observation features are usually treated as independent components. However, it is unrealistic in most cases. More precisely, the observation at any given instance only depends on the label at its location. Another drawback of generative models is that full observations are expected for the model parameter learning because of the excessive modeling of $P(\boldsymbol{x}|\boldsymbol{y})$. The generative models must enumerate all observation cases. The effort is wasted in modeling the observation probability $P(\boldsymbol{x}|\boldsymbol{y})$ which is very complex sometimes.

The discriminative model models the conditional probability $P(\boldsymbol{y}|\boldsymbol{x})$ directly. One advantage of the discriminative model is that it does not waste effort on modeling observation and samples of observation that could be used for the training. It is similar to the maximum entropy model which only models the known variables and assumes the unknown variables as uniform as possible.

2.1.2 Definition of Graphical Models

One key idea of the new machine learning developed in recent years is the probabilistic graphical model, which is an interplay between probability and graph theory and plays a central role in uncertain and complex engineering problems [1, 93, 8, 71].

Graphical models originated from physics have broad applications in machine learning, image processing and computer vision. There are several advantages in

using probabilistic graphical models over the nonstructural methods:

- The use of graphs provides a simple way to visualize the structure of the probabilistic model. Its graphical visualization provides a useful way of designing and constructing new models.

- By carefully inspecting the graph representation the insight of conditional independence could be specified.

- The computational complexity could be reduced based on insight of these independence conditions. Sum and product rule [64][49] could be easily applied according to the graph node and edge structure.

A graph $G(V, E)$ comprises nodes V and edges E. Nodes are random variables, and edges/links represent relationships between these variables. Absence of an edge between two nodes represents conditional independence between them. Two random variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are called conditional independence given a third random variable $\boldsymbol{x}_3$, if they are independent in their conditional probability distribution, formally $p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{x}_3) = p(\boldsymbol{x}_1|\boldsymbol{x}_3)p(\boldsymbol{x}_2|\boldsymbol{x}_3)$. A graph can capture the interactions of the random variables, so the joint distribution of these random variables can be expressed in term of a product of factors. Conditional independence of nodes in a graph can be used to decompose complex probability distribution $P(\boldsymbol{x})$ into a product of factors, each consisting of a subset of corresponding random variables. A probabilistic graphical model is a diagrammatic representation of a probability distribution with factorized terms as follows,

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{S \subset G} \Psi_S(\boldsymbol{x}_S), \tag{2.1}$$

where S denotes a subset of the graph $G = (V, E)$, Ψ_S is a subset of factors, and $\boldsymbol{x}_S$ is a subset of observations. The normalization factor is

$$Z = \sum_{\boldsymbol{x} \in \mathcal{X}^N} \prod_{S \subset G} \Psi_S(\boldsymbol{x}_S), \tag{2.2}$$

where N is the number of all nodes in the whole graph. It should be noted that for all possible subsets S, each element in $\boldsymbol{x}_S$ belongs to the observation set $\mathcal{X}$ and the normalization factor Z is summed over all possible $\boldsymbol{x}_S$. For simplicity, it is denoted as $\boldsymbol{x} \in \mathcal{X}^N$, where $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$. This expression also applies for the summation of $\boldsymbol{y}_C$ of $\boldsymbol{y}$ in the rest of the thesis.

The graphical models could also be classified based on whether its nodes have parents or not. This means the arc of the graph have a direction or not. In the directed graphical model, (also called Bayesian network), edges of the graph have a particular direction indicated by arrows. The undirected graphical model, such as the Markov random field model, does not have direction on the graph. The directions on the graph denote the causal relationship of nodes. If no direction exists there are only soft constrains between the nodes. Or in other words, the directed models is just a subset of undirected models with one way interaction.

2.1.3 Directed Graphical Models

General graphical models are formulations for compactly expressing different types of conditional independences between an ensemble of random variables. The directed graphical models are those graphical models in which all the inter-node connections have a direction, usually indicated by an arrowhead. If a joint distribution $P(\boldsymbol{x})$ of a graphical model can be factorized to the product of distributions for each node i,

i.e.,

$$P(\boldsymbol{x}) = \prod_{i \in V} P(\boldsymbol{x}_i | \boldsymbol{x}_{\pi(i)}), \tag{2.3}$$

where the distribution of each node is conditioned on its set of parent nodes $\pi(i)$, this graphical model is called directed graphical models or Bayesian networks (Bayes nets).

2.1.4 Markov (undirected) Graphical Models

The most widely-used probabilistic model in the signal processing field is the Markov network also referred to the Markov graphical model. In Markov graphical models, a probability distribution can be represented by an undirected graphical model using a product of non-negative functions of the maximal cliques of $G = (V, E)$. This section introduces the Markov random field and its extension–conditional random field. Both of them could be formulated with hidden states.

For classification problems, vertexes $V = \mathcal{X} \bigcup \mathcal{Y}$ are depicted by circles in an independency graph G. Here $\mathcal{X}$ is the set of input observations and $\mathcal{Y}$ is the set of output labels. In this thesis, as in Fig. 2.1, $\mathcal{X}$ and $\mathcal{Y}$ are denoted by shaded circles and empty circles, respectively.

Markov Property and Factorization

In graphical models, the graph G can be used to impose constraints on random variables in two different ways: Markov property and factorization.

- *Markov property.* Observations $\boldsymbol{x}$ are Markov with respect to the graph G, if $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ are conditionally independent given $\boldsymbol{x}_S$, where S separates A and B. Here S, A and B are nodes in the graph G.

- *Factorization.* The distribution $P(\boldsymbol{x})$ can be factorized according to the graph G, if it can be expressed as a product over cliques:

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C). \tag{2.4}$$

The factors $\Psi_C > 0$ are so-called potential functions of the random variables $\boldsymbol{x}_C$ within a clique $C \in \mathcal{C}$, where $\mathcal{C}$ is the set of all cliques. The normalization factor is

$$Z = \sum_{\boldsymbol{x} \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C). \tag{2.5}$$

The relationship of Markov property and factorization could be described in the following theorem [7].

Theorem 2.1. *(Hammersley_Clifford) Suppose p is a strictly positive distribution, and G is an undirected graph that indexes the domain of p, then p is Markov with respect to G if and only if p factorizes according to G.*

It gives necessary and sufficient conditions that a positive distribution satisfies the Markov property with respect to an undirected graph. It means that a positive distribution has Markov properties according to an undirected graph if and only if its density can be factorized over the cliques of the graph.

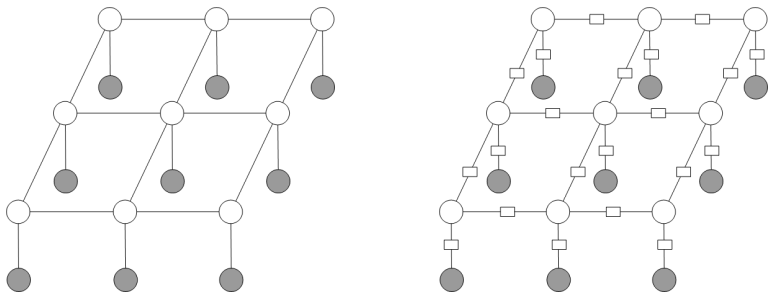
Markov Random Field

Markov random field (MRF) [24], an undirected graphical model, is popular in the physics and vision field. The traditional MRF model in a classification problem can

be formulated using the posterior probability as

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}) &= \frac{P(\boldsymbol{x},\boldsymbol{y})}{P(\boldsymbol{x})} \\ &\propto P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y}) \\ &= \prod_{i=1}^{N} P(\boldsymbol{x}_i|\boldsymbol{y}_i) \cdot \frac{1}{Z} \prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{y}_C). \end{aligned} \tag{2.6}$$

Fig 2.1 shows an example of a 2D MRF model and its factor graph representation. Shaded circles are the observed features at nodes, and empty circles represent labels. The interactions between these random variables are shown as edges. Factors are denoted by empty rectangulars in the factor graph expression.



(a) (b)

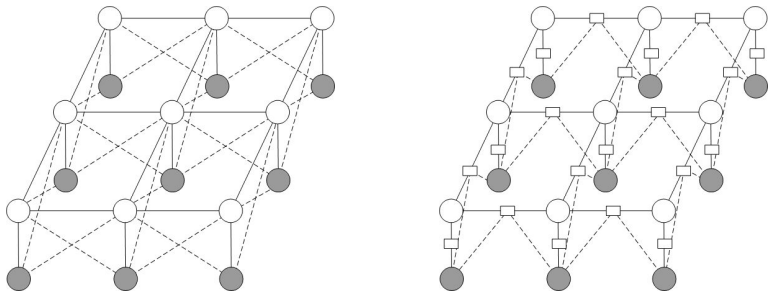
Figure 2.1: An example of 2D MRF model (a) and its factor graph representation (b)

MRF models incorporate both prior knowledge and local spatial relationship. Their performance can be evaluated in a natural way. MRF methods are based on pixels or regular shape neighbors and are widely explored in theoretical and practical research [58]. Note that the MRF assumes that the observations are conditionally independent of each other given the current labels. MRF makes the unwarranted independent assumption, which is not desirable for real-world applications with multiple

interacting features and long range dependencies.

Conditional Random Field

The CRF model [52] is an extension of MRF model. Fig. 2.2 shows an example of a 2D CRF model and its factor graph representation. Interactions between observations at nodes and their neighboring nodes' labels are displayed by dashed lines in Fig. 2.2. It relaxes the observation independence assumptions of MRF. There are interactions between the current observation and neighboring observations, so the conditional probability $P(\boldsymbol{x}|\boldsymbol{y})$ can not be written in the form of $\prod_{i=1}^{N} P(\boldsymbol{x}_i|\boldsymbol{y}_i)$ as equation (2.6) above. Moreover, since the labels $\boldsymbol{y}$ are related to observations $\boldsymbol{x}$ without the assumption of independence which is often the case in real-world applications, the expression of the prior probability $P(\boldsymbol{y})$ as $\frac{1}{Z}\prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{y}_C)$ in equation (2.6) is not appropriate here.



(a) (b)

Figure 2.2: An example of 2D CRF model (a) and its factor graph representation (b)

The definition of CRF is as follows [52][94],

Definition 2.1. *Let $G = (V, E)$ be a graph and the random variables $\boldsymbol{y} = (\boldsymbol{y}_i)_{i \in V}$, so that $\boldsymbol{y}$ is indexed by the vertices of G. Then $(\boldsymbol{x}, \boldsymbol{y})$ is a conditional random field in case when conditioned on $\boldsymbol{x}$, the random variables $\boldsymbol{y}$ obey the Markov property with respect to the graph. $P(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{V_{\setminus i}}) = P(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{N_i})$, where $V_{\setminus i}$ is the set of all nodes in the graph except the node i, N_i is the set of neighbors of the node i in G.*

The general model formulation of CRF models is

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{y}_C, \boldsymbol{x}_C), \tag{2.7}$$

where the normalization factor is

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}' \in \mathcal{Y}^N} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{y}'_C, \boldsymbol{x}_C) \tag{2.8}$$

and $\boldsymbol{y}'$ is all possible $\boldsymbol{y}$. The CRF model performs better than other graphical models in most real-world applications because it does not make the unwarranted independent assumption. Its theory has a equivalence to maximum entropy model.

2.2 The Maximum Entropy Original of CRF Model

The maximum entropy model [41][33] comes from two basic ideas.

- The first idea is to keep unknown variables as uniform as possible. A mathematical measurement of uniformity of a conditional distribution of $P(\boldsymbol{y}|\boldsymbol{x})$ is a conditional entropy

$$H(P) \approx -\sum_{\boldsymbol{x} \in \mathcal{X}^N, \boldsymbol{y} \in \mathcal{Y}^N} \tilde{P}(\boldsymbol{x}) P(\boldsymbol{y}|\boldsymbol{x}) \log P(\boldsymbol{y}|\boldsymbol{x}), \tag{2.9}$$

  where $\tilde{P}(\boldsymbol{x})$ is statistical (empirical) distribution of training samples, and N is the number of nodes in the graph G.

- The second idea is to keep the model factor comply with the known factor distribution of training samples,

$$P(f_k) = \tilde{P}(f_k) \;\; k = 1, \ldots, K, \tag{2.10}$$

where K is the total number of factors, $\{f_k\}$ are feature functions of random variable $\boldsymbol{x}$ and $\boldsymbol{y}$. Feature functions could be any kind of factor such as color and texture. Here $P(f_k)$ and $\tilde{P}(f_k)$ are,

$$P(f_k) \approx \sum_{\boldsymbol{x}\in\mathcal{X}^N, \boldsymbol{y}\in\mathcal{Y}^N} \tilde{P}(\boldsymbol{x}) P(\boldsymbol{y}|\boldsymbol{x}) f_k(\boldsymbol{x}, \boldsymbol{y}), \tag{2.11}$$

$$\tilde{P}(f_k) = \sum_{\boldsymbol{x}\in\mathcal{X}^N, \boldsymbol{y}\in\mathcal{Y}^N} \tilde{P}(\boldsymbol{x}, \boldsymbol{y}) f_k(\boldsymbol{x}, \boldsymbol{y}). \tag{2.12}$$

To find a solution to this constrained problem, one can define a Lagrange function with the Lagrange multipliers $\{\lambda_k\}$ as follows,

$$L(P, \boldsymbol{\lambda}) = H(P) + \sum_{k=1}^{K} \lambda_k (P(f_k) - \tilde{P}(f_k)), \tag{2.13}$$

where $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_K\}$. To get the optimal solution, one can calculate the partial derivative of $L(P, \lambda)$ with respect to P and set the value to zero, *i.e.*,

$$\begin{aligned} \frac{\partial L(P, \boldsymbol{\lambda})}{\partial P} &= -\tilde{P}(\boldsymbol{x}) \log P(\boldsymbol{y}|\boldsymbol{x}) - \tilde{P}(\boldsymbol{x}) P(\boldsymbol{y}|\boldsymbol{x}) \frac{1}{P(\boldsymbol{y}|\boldsymbol{x})} + \sum_{k=1}^{K} \lambda_k \tilde{P}(\boldsymbol{x}) f_k(\boldsymbol{x}, \boldsymbol{y}) \\ &= -\tilde{P}(\boldsymbol{x}) \log P(\boldsymbol{y}|\boldsymbol{x}) - \tilde{P}(\boldsymbol{x}) + \sum_{k=1}^{K} \lambda_k \tilde{P}(\boldsymbol{x}) f_k(\boldsymbol{x}, \boldsymbol{y}) \\ &= 0. \end{aligned} \tag{2.14}$$

The distribution becomes,

$$P(\boldsymbol{y}|\boldsymbol{x}) = \exp(-1 + \sum_{k=1}^{K} \lambda_k f_k(\boldsymbol{x}, \boldsymbol{y})). \tag{2.15}$$

To comply with normalization of the distribution, *i.e.*,

$$\begin{aligned} 1 &= \sum_{\boldsymbol{y}\in\mathcal{Y}^N} P(\boldsymbol{y}|\boldsymbol{x}) \\ &= \sum_{\boldsymbol{y}\in\mathcal{Y}^N} \exp(-1+\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y})), \end{aligned} \tag{2.16}$$

one can have

$$e^{-1} = \frac{1}{\sum_{\boldsymbol{y}\in\mathcal{Y}^N}\exp(\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y}))}. \tag{2.17}$$

The distribution that maximizes the Lagrangian function $L(P,\boldsymbol{\lambda})$ is,

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}) &= \exp(-1+\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y})) \\ &= e^{-1}\cdot\exp(\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y}))) \\ &= \frac{\exp(\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y})))}{\sum_{\boldsymbol{y}\in\mathcal{Y}^N}\exp\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y}))}. \end{aligned} \tag{2.18}$$

Let $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}\in\mathcal{Y}^N}\exp(\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y}))$, the posterior probability in the maximum entropy model is

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x},\boldsymbol{y}))}{Z(\boldsymbol{x})}. \tag{2.19}$$

According to Theorem 9.1 of [27], the maximum entropy (ME) model is equivalent to the conditional random field (CRF) model, and this is uniquely determined [73],

$$P_{me} = P_{CRF}. \tag{2.20}$$

Therefore, the CRF model can be factorized as,

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}) &= \frac{1}{Z(\boldsymbol{x})}\prod_{C\in\mathcal{C}}\Psi_C(\boldsymbol{y}_C,\boldsymbol{x}_C) \\ &= \frac{1}{Z(\boldsymbol{x})}\prod_{C\in\mathcal{C}}\prod_{k\in K_C}\exp\left(w_k f_k(\boldsymbol{y}_C,\boldsymbol{x}_C)\right), \end{aligned} \tag{2.21}$$

where $K_C \subset \{1, 2, \ldots, K\}$ is the index set of factors for the clique C, $\boldsymbol{w}$ is the weight coefficient vector of factors, and totally there are K factors. The normalization factor is:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}' \in \mathcal{Y}^N} \prod_{C \in \mathcal{C}} \prod_{k \in K_C} \exp\left(w_k f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C)\right). \tag{2.22}$$

Here one assume factors $\boldsymbol{f} = (f_1, \ldots, f_K)$ does not change across the cliques.

2.3 CRF Training and Inference

2.3.1 Training of CRF Models

For all types of CRF models, the maximum-likelihood training method can be used to estimate parameters $\boldsymbol{w}$ of the model [94, 48, 88, 92]. Suppose the training set is $\mathcal{T}$ and the estimation can be done by maximizing the following log-likelihood $\mathcal{L}(\mathcal{T}, \boldsymbol{w})$ with parameters $\boldsymbol{w} = \{w_1, \ldots, w_K\}$, which is

$$\begin{aligned}
\mathcal{L}(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \log P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) \\
&= \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \log \left(\frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \prod_{C \in \mathcal{C}} \prod_{k \in K_C} \exp\left(w_k f_k(\boldsymbol{y}_C, \boldsymbol{x}_C)\right) \right) \\
&= \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \log \frac{\prod_{C \in \mathcal{C}} \prod_{k \in K_C} \exp\left(w_k f_k(\boldsymbol{y}_C, \boldsymbol{x}_C)\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^N} \prod_{C \in \mathcal{C}} \prod_{k \in K_C} \exp\left(w_k f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C)\right)}.
\end{aligned} \tag{2.23}$$

To avoid overfitting, a penalty term $-\sum_{k=1}^{K} \frac{w_k^2}{2\delta^2}$ is added [16, 48]. The log-likelihood $\mathcal{L}(\mathcal{T}, \boldsymbol{w})$ becomes

$$\begin{aligned}
\mathcal{L}(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \log \left(\frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \prod_{C \in \mathcal{C}} \prod_{k \in K_C} \exp\left(w_k f_k(\boldsymbol{y}_C, \boldsymbol{x}_C)\right) \right) - \sum_{k=1}^{K} \frac{w_k^2}{2\delta^2} \\
&= \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \sum_{C \in \mathcal{C}} \sum_{k \in K_C} w_k f_k(\boldsymbol{y}_C, \boldsymbol{x}_C) - \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}} \log Z(\boldsymbol{x}, \boldsymbol{w}) - \sum_{k=1}^{K} \frac{w_k^2}{2\delta^2},
\end{aligned}$$

where δ^2 is a constant chosen to trade off between exact fitting of observation factors and squared norms of the weight vector $\boldsymbol{w}$ [48, 63]. The smaller the values are the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. Denote

$$\begin{aligned} \mathcal{L}_1(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{C\in\mathcal{C}} \sum_{k\in K_C} w_k f_k(\boldsymbol{y}_C, \boldsymbol{x}_C), \\ \mathcal{L}_2(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log Z(\boldsymbol{x}, \boldsymbol{w}). \end{aligned} \tag{2.24}$$

Gradient descent algorithm can be used in the training of CRF models, with an arbitrary initial values of the weight vector $\boldsymbol{w}_0$. To update weights in each iteration, we need to calculate the partial derivatives of $\mathcal{L}(\mathcal{T}, \boldsymbol{w})$ with respect to the weight w_k, $k = 1, \ldots, K$. The partial derivative of the first term with respect to w_k is

$$\frac{\partial \mathcal{L}_1(\mathcal{T}, \boldsymbol{w})}{\partial w_k} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}_C, \boldsymbol{x}_C). \tag{2.25}$$

The partial derivative of the second term with respect to w_k is

$$\begin{aligned} \frac{\partial \mathcal{L}_2(\mathcal{T}, \boldsymbol{w})}{\partial w_k} &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \frac{\partial Z(\boldsymbol{x}, \boldsymbol{w})}{\partial w_k} \\ &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} \left(\prod_{C\in\mathcal{C}} \prod_{k\in K_C} \exp\left(w_k f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C) \right) \right) \cdot \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C) \\ &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} \left(\frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \prod_{C\in\mathcal{C}} \prod_{k\in K_C} \exp\left(w_k f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C) \right) \right) \cdot \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C) \\ &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} P(\boldsymbol{y}'|\boldsymbol{x}) \cdot \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C). \end{aligned} \tag{2.26}$$

At last, the partial derivative of the third term with respect to w_k is

$$\frac{\partial}{\partial w_k} \left(-\sum_{k=1}^{K} \frac{w_k^2}{2\delta^2} \right) = -\frac{w_k}{\delta^2}. \tag{2.27}$$

The partial derivative of the first term with respect to w_k in equation (2.25) is the

empirical distribution of a feature f_k, *i.e.*,

$$\tilde{E}(f_k) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}_C, \boldsymbol{x}_C). \tag{2.28}$$

The partial derivative of second term with respect to w_k in equation (2.26) is the expectation under the model distribution, *i.e.*,

$$E(f_k) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} P(\boldsymbol{y}'|\boldsymbol{x}) \cdot \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C). \tag{2.29}$$

Therefore, the partial derivative of the log-likelihood $\mathcal{L}(\mathcal{T};\boldsymbol{w})$ with respect to w_k can be calculated as

$$\frac{\partial\mathcal{L}(\mathcal{T},\boldsymbol{w})}{\partial w_k} = \tilde{E}(f_k) - E(f_k) - \frac{w_k}{\delta^2}. \tag{2.30}$$

The empirical distribution of feature functions is supposed to be equal to its expected value on the model distribution. Denote the partial derivative $\partial\mathcal{L}(\mathcal{T},\boldsymbol{w})$ as Δw_k, the weight w_k of the CRF model is updated as $w_k - \Delta w_k$ after each iteration in the process of training, $k = 1,\ldots,K$. The iteration process of estimating the weight vector $\boldsymbol{w}$ stops when all differences $\{\Delta w_k\}$ are less than a predetermined threshold.

In CRF training, it is not possible to calculate $\sum_{\boldsymbol{y}'\in\mathcal{Y}^N} P(\boldsymbol{y}'|\boldsymbol{x}) \cdot \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C)$ in equation (2.26) directly, because of huge number of possible labels. For a general graph, even if all parameters $\boldsymbol{w}$ and factors $\boldsymbol{f}$ are known, one can only obtain an approximate value of it. In a general CRF, one can use the loopy belief propagation algorithm. For a special form of a CRF, the linear chain structure similar to hidden Markov model (HMM), the standard backward-forward algorithm can be used. Both algorithms are presented in the following section for the inference of CRF models.

2.3.2 Inference of CRF Models

There are three principle algorithms for probabilistic inference of graphical model [93], namely exact algorithm, sampling algorithm, and variational algorithm.

Exact inference algorithms include the elimination algorithm, the sum-product algorithm [64], and the junction tree algorithm [39]. They compute marginal probability by systematically exploiting the graphical structure. When the tree width is small exact algorithms are practical. HMM is an algorithm of this kind. Once the tree width is overly large, these algorithms are not viable.

Sampling algorithms, such as the Markov chain Monte Carlo [80, 81], provide a general methodology for inference. We can find solutions by approximating distribution such as Gibbs Sampling [27].

The general idea behind the variational algorithm is to characterize a probability distribution by solving a perturbed optimization problem. In early applications, it is formulated as the solution of Kullback-Leibler(KL) divergence. It can also be obtained using other ways such as mean field approach. The Bethe approximation approach involves retaining only consistency relations that arise from local neighborhood relationship in graphical model. Surprisingly the Bethe approximation is equivalent to the sum product algorithm for trees and for graph with loops [104].

Since this thesis is not about the fundamentals of machine learning theory, we only focus on one popular variational algorithm (loopy belief propagation) and the backward-forward inference algorithm for linear chain CRF models. Before introducing these two inference algorithms, the expression of CRF models in edge and node factors is presented here. The CRF can be expressed in the following edge and node

factors [50],

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{(i,j)\in E} \Psi_{(i,j)}(\boldsymbol{y}, \boldsymbol{x}) \prod_{i\in V} \Psi_i(\boldsymbol{y}, \boldsymbol{x}). \tag{2.31}$$

where $\Psi_{(i,j)}(\boldsymbol{y}, \boldsymbol{x})$ is the contribution of edge (i, j), and $\Psi_i(\boldsymbol{y}, \boldsymbol{x})$ is the contribution of node i as shown in the factor graph of Fig. 2.2. Suppose the set of all nodes is S, and the set $N_i \subset S$ is neighboring sites of the site i, the posterior probability $P(\boldsymbol{y}|\boldsymbol{x})$ in equation (2.31) usually is written as

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\{\sum_{i\in S} \varphi_i(y_i|\boldsymbol{x}) + \sum_{i\in S} \sum_{j\in N_i} \psi_{ij}(y_i, y_j|\boldsymbol{x})\}, \tag{2.32}$$

where $\varphi_i(\cdot)$ is the association potential between the observation data and the label of site i, and $\psi_{ij}(\cdot)$ is the interaction potential between current site i and its neighboring site j given the observed features. As shown in Fig. 2.2 usually φ_i represents the prediction of the label y_i based on the local feature vector $\boldsymbol{x}_i$ at site i and ψ_{ij} predicts the label y_i based on local compatibility between neighboring labels and features.

The Loop Belief Propagation Inference of CRF Model for General Graph

For graphs with loops such as those in image labeling, there is no exact inference algorithm. The inference can be done using approximate loopy belief propagation (BP) [104], and gradient descent [92] can be used as the training method.

When the CRF model is expressed in the form of equation (2.31), the belief propagation algorithm can be applied for model inference. Denote a message variables such as $m_{ij}(\boldsymbol{y}, \boldsymbol{x})$ from node i to node j. It can be intuitively understood as a message from a node i to a node j about what node j should be like. It is proportional to how likely node i thinks that node j will be of certain value. In the BP algorithm, that belief at a node i is proportional to the product of the local evidence at that node

$\psi_i(\boldsymbol{y}, \boldsymbol{x})$ and all message coming into node i, *i.e.*,

$$b_i(\boldsymbol{y}, \boldsymbol{x}) = k\Psi_i(\boldsymbol{y}, \boldsymbol{x}) \prod_{j \in N_i} m_{ji}(\boldsymbol{y}, \boldsymbol{x}) \tag{2.33}$$

where k is a normalization constant to force the summation of beliefs to be 1. The messages are determined self-consistently by the message update rules as follows,

$$m_{ji}(\boldsymbol{y}, \boldsymbol{x}) = \sum_{\boldsymbol{y}_j \in \mathcal{Y}} \Psi_i(\boldsymbol{y}, \boldsymbol{x}) \Psi_{(j,i)}(\boldsymbol{y}, \boldsymbol{x}) \prod_{k \in N_j \setminus i} m_{kj}(\boldsymbol{y}, \boldsymbol{x}), \tag{2.34}$$

where $N_j \setminus i$ denotes the neighboring set of node j except the node i. Here we take the product of all messages going into node j except the one coming from node i as shown in Fig. 2.3. The summation is done by all possible labels of node j. It is easy

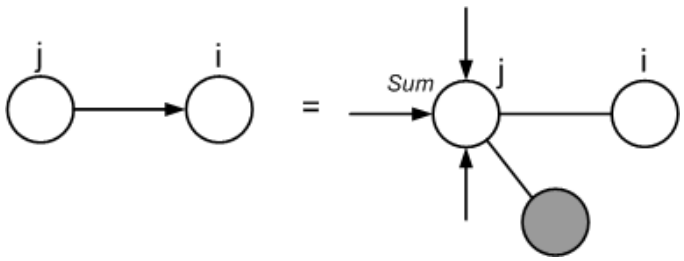


Figure 2.3: An illustration of message updating rules in belief propagation. The sum indicate summation of all messages coming to node j except the one from i.

to testify the BP updating rule in graph without loops as shown in Appendix A. For a graph with loops, it may not converge with some parameter setting. However, the BP has already been used in graphs with loops successfully in many applications such as computer vision and error control coding.

The Backward-forward Inference of Linear Chain CRF Model

For graphs without loops, *e.g.*, the CRF model structured as a linear chain, there are exact inference algorithms exist such as the dynamic programming [48]. Here the

commonly-used linear chain CRF model, which is a counterpart of HMM model, is discussed. In the linear chain CRF model, the posterior probability can be simplified as

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \boldsymbol{x}), \tag{2.35}$$

where t is an index of time, T is the length of the sequence, $\boldsymbol{x}_t$ denotes a vector which include observation features at time t, y_t denotes the label at time t, and the factor at time t is

$$\Psi_t(y_t, y_{t-1}, \boldsymbol{x}) = \exp\left(\sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, \boldsymbol{x})\right). \tag{2.36}$$

Define the forward and backward variables for CRF as follows,

$$\begin{aligned}
\alpha_t(j) &\propto P(y_t = j|\boldsymbol{x}_{<1\ldots t>}) \\
&= \sum_{y_{<1,\ldots,t-1>} \in \mathcal{Y}^{t-1}} \Psi_t(j, y_{t-1}, \boldsymbol{x}) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, \boldsymbol{x}), && (2.37) \\
\alpha_{t+1}(j) &= \sum_{i \in S} \Psi_t(j, i, \boldsymbol{x}) \alpha_{t-1}(i), && (2.38) \\
\beta_t(i) &\propto p(y_t = i|\boldsymbol{x}_{<t+1\ldots T>}) \\
&= \sum_{y_{<t+1,\ldots,T>} \in \mathcal{Y}^{T-t}} \prod_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, \boldsymbol{x}), && (2.39) \\
\beta_t(i) &= \sum_{j \in S} \Psi_{t+1}(j, i, \boldsymbol{x}) \beta_{t+1}(j) && (2.40)
\end{aligned}$$

and the initial points,

$$\beta_T(end) = \alpha_0(start) = 1, \tag{2.41}$$

where $\boldsymbol{x}_{<1,\ldots,t>}$ denotes $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$. One can compute the margin probability needed in gradient computation as

$$P(y_{t-1}, y_t|\boldsymbol{x}) \propto \alpha_{t-1}(y_{t-1}) \Psi_t(y_t, y_{t-1}, \boldsymbol{x}) \beta_t(y_t). \tag{2.42}$$

So, the calculation of $E(f_k)$ in equation (2.30) can be implemented efficiently as

follows,

$$
\begin{aligned}
E(f_k) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} P(\boldsymbol{y}'|\boldsymbol{x}) \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{x}_C) \\
&= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \frac{1}{Z(\boldsymbol{x})} \sum_{t=1}^{T} \sum_{(i,j)\in E} \sum_{i\in V} f_k(j,i,\boldsymbol{x})\alpha_{t-1}(i)\Psi_t(j,i,\boldsymbol{x})\beta_t(j). \quad (2.43)
\end{aligned}
$$

Also the normalize factor can be efficiently formulated as backward or forward as

$$Z(\boldsymbol{x}) = \beta_0(start) = \alpha_T(end). \quad (2.44)$$

Therefore, the most probable assigned labels $\boldsymbol{y}^*$ are those that maximize the posterior probability $P(\boldsymbol{y}|\boldsymbol{x})$, *i.e.*,

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}\in\mathcal{Y}^N} P(\boldsymbol{y}|\boldsymbol{x}). \quad (2.45)$$

Dynamic programming is applied here to obtain the optimal label solutions. Define a quantity $\delta_t(i)$ as the highest score along a path at time t given observations,

$$\delta_t(i) = \max_{y_{<1,\ldots,t-1>}\in\mathcal{Y}^{t-1}} P(y_{<1,\ldots,t-1>}, y_t = i|\boldsymbol{x}). \quad (2.46)$$

A Viterbi algorithm is used to calculate the optimal labels. Here ξ_t is used to keep track the label values. Steps of the Viterbi algorithm are listed as follows:

- Initialization:

$$
\begin{aligned}
\delta_1(i) &= \Psi_1(i, \text{start}, \boldsymbol{x}), \quad (2.47)\\
\xi_1(i) &= \text{start}; \quad (2.48)
\end{aligned}
$$

- Induction:

$$
\begin{aligned}
\delta_t(i) &= \max_j \delta_{t-1}(j)\Psi_t(j,i,\boldsymbol{x}), \quad (2.49)\\
\xi_t(i) &= \arg\max_j \delta_{t-1}(j)\Psi_t(j,i,\boldsymbol{x}); \quad (2.50)
\end{aligned}
$$

- Calculating probability:

$$P(\boldsymbol{y}|\boldsymbol{x}) \propto \max_j \delta_T(j). \tag{2.51}$$

$$\boldsymbol{y}_T^* = \arg\max_j \delta_T(j); \tag{2.52}$$

- Path backtracking: choosing the optimal path using the track keeping values γ_t. The optimal solution at time t is

$$y_t^* = \xi_{t+1}(y_{t+1}^*). \tag{2.53}$$

2.4 HCRF and its Training and Inference

2.4.1 HCRF Models

Recently there has been a growing interest in CRF with latent variables. Original works on CRF focus fully on observed training data which is difficult in cases such as video content analysis. Additionally it is a troublesome work to label all states manually. The introduction of the hidden states in graphical model simplifies the complex joint distribution by breaking them into simpler components. Similar to hidden Markov model (HMM) that is a 1D chain Markov random field with hidden variable, hidden conditional random field (HCRF), a relative new graphical model, is chain CRF with hidden variables [75]. Similar to HMM which is widely used in event detection *e.g.*, [14, 25, 43] the HCRF has been developed for event and object recognition *e.g.*, [29, 95].

An illustration of the HMM, CRF and HCRF model is shown in Fig. 2.4. As shown in Fig. 2.4, for video event detection using HMM, a specific model should be set up for each specific event y. For example in the golf event detection, there

are three events: the full swing (event 1), the non-full swing (event 2), and others (event 3), so $y \in \mathcal{Y} = \{1, 2, 3\}$. There are three models corresponding to three events. During the training, the parameters are learned for each model. The class label for testing a sequence is inferred by finding the most probable model for a sequence.

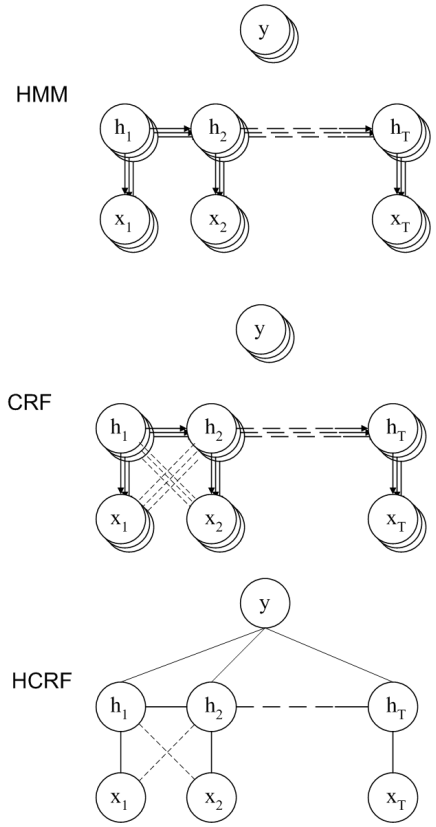


Figure 2.4: An illustration of HMM, CRF and HCRF model structure for video content analysis.

CRF can also be used for video event analysis. It is similar to HMM except that there are links between current label and neighboring observations. CRF needs labels for all hidden states for the training set, and it is difficult and time consuming. In HCRF video event analysis, there is only one model and weights of different factors

to serve as coefficients to classify the sequences. During the training process, weights $\boldsymbol{w}$ are learned from training sequences. Estimated parameters are used to label the events in the testing process.

There are several differences between the two models.

1. There are direct links between y and hidden states sequence in HCRF, while HMM does not have this useful structure.

2. Links of HMM have direction. This is the generative nature of the model. Observations are "children" of states, and generated by states. So full observation is needed for the training. The HCRF relaxes this assumption.

3. In HMM the observations are independent and only depend on their own state. A HCRF model can have links between the current observation and other states beside its current state.

These properties make HCRF a better tool for complex video event detection problems.

The HCRF formulation is as follows,

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \sum_{\boldsymbol{h} \in \mathcal{H}^N} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{y}_C, \boldsymbol{h}_C, \boldsymbol{x}_C). \tag{2.54}$$

Unlike the node labels $\boldsymbol{y}$, the unknown hidden states $\boldsymbol{h}$ is summed over in the equation. The normalize factor is,

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}' \in \mathcal{Y}^N} \sum_{\boldsymbol{h} \in \mathcal{H}^N} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{y}'_C, \boldsymbol{h}_C, \boldsymbol{x}_C), \tag{2.55}$$

where $\boldsymbol{y}'$ are possible labels for a sequence.

2.4.2 HCRF and its Inference and Training

The training and inference can be done similarly to the ordinary CRF model except the summation of hidden variables,

$$\begin{aligned}
\mathcal{L}(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}) \\
&= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log \sum_{\boldsymbol{h}\in\mathcal{H}^N} P(\boldsymbol{y},\boldsymbol{h}|\boldsymbol{x};\boldsymbol{w}) \\
&= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log \frac{1}{Z(\boldsymbol{x},\boldsymbol{w})} \sum_{\boldsymbol{h}\in\mathcal{H}^N} \prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{y}_C,\boldsymbol{h}_C,\boldsymbol{x}_C). \qquad (2.56)
\end{aligned}$$

Similarly to CRF models, there is a penalty term $-\sum_{k=1}^{K}\frac{w_k^2}{2\delta^2}$ that is added to avoid overfitting. The function becomes

$$\begin{aligned}
\mathcal{L}(\mathcal{T}, \boldsymbol{w}) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log \frac{1}{Z(\boldsymbol{x},\boldsymbol{w})} \sum_{\boldsymbol{h}\in\mathcal{H}^N} \prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{y}_C,\boldsymbol{h}_C,\boldsymbol{x}_C) - \sum_{k=1}^{K}\frac{w_k^2}{2\delta^2} \\
&= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{h}\in\mathcal{H}^N} \exp\left(\sum_{C\in\mathcal{C}}\sum_{k\in K_C} w_k f_k(\boldsymbol{y}_C,\boldsymbol{x}_C)\right) \\
&- \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \log Z(\boldsymbol{x},\boldsymbol{w}) \\
&- \sum_{k=1}^{K}\frac{w_k^2}{2\delta^2}. \qquad (2.57)
\end{aligned}$$

Let $\mathcal{L}_l(\mathcal{T},\boldsymbol{w})$ denote the lth term of the log-likelihood $\mathcal{L}(\mathcal{T},\boldsymbol{w})$, $l = 1,2,3$. The partial derivative of the first term with respect to w_k is,

$$\frac{\partial\mathcal{L}_1(\mathcal{T},\boldsymbol{w})}{\partial w_k} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{h}\in\mathcal{H}^N} P(\boldsymbol{h}|\boldsymbol{y},\boldsymbol{x};\boldsymbol{w}) \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}_C,\boldsymbol{h}_C,\boldsymbol{x}_C). \qquad (2.58)$$

The partial derivative of the second term with respect to w is,

$$\frac{\partial\mathcal{L}_2(\mathcal{T},\boldsymbol{w})}{\partial w_k} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} \sum_{\boldsymbol{h}\in\mathcal{H}^N} P(\boldsymbol{y}',\boldsymbol{h}|\boldsymbol{x};\boldsymbol{w}) \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C,\boldsymbol{h}_C,\boldsymbol{x}_C). \qquad (2.59)$$

Therefore, the partial derivative of the log-likelihood (with a penalty) with respect to w_k is

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathcal{T}, \boldsymbol{w})}{\partial w_k} &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{h}\in\mathcal{H}^N} P(\boldsymbol{h}|\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{w}) \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}_C, \boldsymbol{h}_C, \boldsymbol{x}_C) \\ &- \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} \sum_{\boldsymbol{h}\in\mathcal{H}^N} P(\boldsymbol{y}', \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{w}) \sum_{C\in\mathcal{C}} f_k(\boldsymbol{y}'_C, \boldsymbol{h}_C, \boldsymbol{x}_C) \\ &- \frac{w_k}{\delta^2}. \end{aligned} \tag{2.60}$$

This requires calculation of two marginalized distribution which can be calculated using belief propagation. For linear HCRF models, the backward-forward inference algorithm can be used in its inference process also.

2.5 Modeling Spatial and Temporal Interaction with CRF Model

2.5.1 The Interaction in Image and Video Content

In image labeling, the interactions include the smoothness of the region labels and the complex interaction of the observed features. In video shot classification, the interactions include the smoothness of shot frame labels and the interactions of the observed features of frames. Smoothness of labels means that the neighboring sites tend to have similar labels except at the group boundary. The complex interactions of features tend to regulate the labels. The features and label of one site depends on its neighbors' labels and features. In the ideal case, we would like to find a model that can incorporate these interactions and learn the dependence in a consistent way using the training data in the supervised learning.

2.5.2 Modeling the Interaction using CRF

These proceeding properties make graphical models in particularly CRF idea to solve the content analysis problem. The CRF model relaxes the independent assumption of observation data which is more suitable to model the complex date and label interactions in image and video content. When modeling the interactions of this content, it is also important to take the statistical variations of the feature observations in each class and other uncertainties such as noise into account. In the following chapters, we present new CRF models with new feature functions according to different characteristics of specific content analysis problems. The task is to infer the labels or classes using CRF models with coefficients learned from training samples.

In this chapter, we provide the mathematical formulation of the conditional random field model which is fundamental to the following discussion. Graphical models are very popular for content analysis due to their many advantages over non-structural models. The HMM is one widely used graphical model. The HMM has many limitations such as conditional independence of observations and only tractable for limited types of distributions. CRF is proposed to overcome these problems. The training of CRF could be done efficiently by maximizing the log-likelihood of training data. The loopy belief propagation is an approximate inference method that is effective for general graphs of CRF. For linear chain CRF the backward and forward method could be applied the same as HMM. HCRF is a direct extension of CRF and HMM which is favorable for observations with hidden states. The training and inference could be performed the same as CRF with the summation of all possible hidden states. We will discuss how to use these mathematics in real content analysis tasks, *e.g.*, image labeling and video content analysis in the following chapters.

Chapter 3

Mixture Conditional Random Field for Image Labeling

A new conditional random field (CRF) model based on mixture feature functions is proposed for multi-class image labeling in this chapter.

In image labeling, an image is first divided into a grid of pixels or rectangular regions, and then features of these grids are extracted. These observed features may include color, texture and shape. The 2D grid of an image is a graph where the probabilistic graphical model can be applied. The CRF image labeling model is built on the 2D grid with association and interaction potential functions [51, 50] as in Equation (2.32). The association potential for each site i, $\varphi_i(y_i|\boldsymbol{x})$, represents the log-likelihoods of the label y_i at site i given the observation feature vector $\boldsymbol{x}_i$, *i.e.*,

$$\varphi_i(y_i|\boldsymbol{x}) = \log P(y_i|\boldsymbol{x}_i). \tag{3.1}$$

The interaction potentials $\psi_{ij}(y_i, y_j|\boldsymbol{x})$ at two neighboring sites i and j are the log-likelihoods of the interaction between neighboring grid labels y_i and y_j given the

observation feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, *i.e.*,

$$\psi_{ij}(y_i, y_j|\boldsymbol{x}) = \log P'(y_i, y_j|\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3.2}$$

Using multiple features, both potentials can be factorized into the weighted feature function forms, *i.e.*, $\sum_k w_k f_k(y|\boldsymbol{x})$. It is the responsibility of the CRF training algorithm to find the weights $\{w_k\}$ for different potential feature functions.

In image processing, image features can be modeled as mixtures [72, 70]. In [85], many potentials are used and the color association potentials of CRF are modeled as Gaussian mixtures. Based on the distributions of features of different classes in nature images (nature images mean images with nature scene), we present a new nature image labeling method in this chapter, based on the mixture CRF model that chooses a Gaussian or Laplacian mixtures as feature functions $\{f_k(\cdot)\}$. By using Gaussian or Laplacian mixtures to approach the distributions of features of nature image parts, the number of features needed in the CRF image labeling can be reduced. Instead of modeling many potentials differently, all potentials are placed on a common form of Gaussian or Laplacian mixtures [70, 4]. By taking advantage of the feature distribution properties, the number of features needed for CRF is greatly reduced. To evaluate the performance we apply the new model to the nature image labeling problem. The performance of Gaussian and Laplacian mixture CRF is evaluated with commonly used 7 class Corel database. The Laplacian mixture is a suitable choice for nature images because the distributions of their features can be better approximated by a Laplacian distribution than by a Gaussian distribution [70]. The experimental results show that the new model with Laplacian mixture achieves best labeling accuracy, compared with the Gaussian mixture CRF, the baseline CRF [51, 28] and the nonstructural SVM model, with the same number of features. The new

model with only several features shows comparable results with the state-of-the-art CRF models with at least around 100 features and complex potential structures.

The chapter is organized as follows. The solution of image labeling problem by CRF is formulated in Section 3.1. Then a new CRF model based on (Gaussian and Laplacian) mixture feature functions is introduced in Section 3.2. After that in Section 3.3, detail steps of the mixture model for image labeling are given. In Section 3.4, the new image labeling model is applied to 7 class Corel database and the simulation results are shown. This chapter ends with some discussions in Section 3.5.

3.1 Formulation of CRF for Image Labeling

A CRF model is used to learn the conditional distribution over the class labels given an image [85]. CRF image labeling is a supervised learning process. The parameters $\boldsymbol{w}$ of CRF are learned from training images with known labels and feature functions. With these parameters, the labels of an input image with unknown node labels can be inferred. Here the features are node and edge features, which could be any meaningful filter response of the site, such as color, texture and shape. The interaction and association are defined on the graph. The task of CRF image labeling is to infer the most probable labels given an input image based on the model parameters which are learned from the training images.

Let $\boldsymbol{x} = \{x_i\}_{i \in S}$ denote the observation data (features) from the input image. S is a set of image sites which could be pixels or a group of adjacent pixels with regular or irregular shapes. The observation at the site i, x_i is a set of observation features. The image has a corresponding labels $\boldsymbol{y} = \{y_i\}_{i \in S}$ where $y_i \in \mathcal{Y}$ is the label for site i.

Here $\mathcal{Y}$ is a set of all possible labels. For example the binary image labeling problem, $\mathcal{Y} = \{-1, 1\}$, 1 represents the object and -1 the background. The labeling problem is to infer the underlying labels $\boldsymbol{y}$ given the image features $\boldsymbol{x}$ and parameters of the model. The probabilistic expression of the problem is to maximize the conditional probability $P(\boldsymbol{y}|\boldsymbol{x})$.

3.2 New Mixture CRFs

3.2.1 Mixture CRFs

The potentials in CRF are usually nonstructural discriminative classifiers such as boosting and logistic. To let the features select themselves simplifies CRF design routine. But it usually needs hundreds of features to converge to a reasonable result and the convergence speed becomes slower with more features. Since most weights of these features are zero or near zero, we could safely select features and feature functions to reduce the complexity and improve the labeling accuracy. Selecting feature functions in potentials that better reflect the distribution of the dominant features could reduce the need for more features and increase the convergence speed. In image processing, mixture models are widely-used nonstructural classifiers. The use of mixture models as potentials for CRF image labeling has not been widely investigated. In this chapter we discuss a new mixture potential solution, namely mixture CRFs, for nature image labeling.

The potentials $P(\boldsymbol{y}|\boldsymbol{x})$ of Equation (2.32) could be factorized in following feature

function forms,

$$\begin{aligned} \varphi_i(y_i|\boldsymbol{x}) &= \sum_{k\in K_i} w_{ik} f_{ik}(y_i|\boldsymbol{x}), \\ \psi_{ij}(y_i, y_j|\boldsymbol{x}) &= \sum_{k\in K_{ij}} w_{ijk} f_{ijk}(y_i, y_j|\boldsymbol{x}), \end{aligned} \tag{3.3}$$

where k is the index of features, w_{ik} and w_{ijk} are weights for these mixtures, K_i and K_{ij} are numbers of mixtures. In this chapter we choose one mixture for each feature. The features are represented by log-likelihood functions, *i.e.*,

$$\begin{aligned} f_{ik}(y_i|\boldsymbol{x}) &= \sum_{l\in\mathcal{L}} \delta(y_i - l) \log \sum_{m\in M} a_{my_i} P_i(x_{ik}|y_i, m), \\ f_{ijk}(y_i, y_j|\boldsymbol{x}) &= \sum_{l\in\mathcal{L}} \sum_{l'\in\mathcal{L}} \delta(y_i - l)\delta(y_j - l') \\ &\cdot \quad \log \sum_{m\in M} a_{my_iy_j} P_{ij}(x_{ik}, x_{jk}|y_i, y_j, m), \end{aligned} \tag{3.4}$$

where m is the index of the mixture component, and M is the number of components. Here a_{y_im} and $a_{y_iy_jm}$ are mixture coefficients. The function

$$\delta(y - l) = \begin{cases} 1 & \text{if } y = l, \\ 0 & \text{otherwhere,} \end{cases}$$

where $l \in \mathcal{L}$ is the index of image classes, and $\mathcal{L}$ is the set of all classes. For a Laplacian mixture, the conditional probabilities are

$$P_i(x_{i_k}|y_i, m) = \frac{\exp(-\frac{|x_{i_k} - \mu_{y_im}|}{b_{y_im}})}{2b_{y_im}}, \tag{3.5}$$

$$P_{ij}(x_{i_k}, x_{j_k}|y_i, y_j, m) = \frac{\exp(-\frac{|(x_{i_k} - x_{j_k}) - \mu_{y_iy_jm}|}{b_{y_iy_jm}})}{2b_{y_iy_jm}}. \tag{3.6}$$

While for a Gaussian mixture, these probabilities are

$$P_i(x_{i_k}|y_i, m) = \frac{\exp(-\frac{(x_{i_k}-\mu_{y_i m})^2}{2\sigma^2_{y_j m}})}{\sqrt{2\pi}\sigma_{y_j m}}, \tag{3.7}$$

$$P_{ij}(x_{i_k}, x_{j_k}|y_i, y_j, m) = \frac{\exp(-\frac{(x_{i_k}-x_{j_k}-\mu_{y_i y_j m})^2}{2\sigma^2_{y_i y_j m}})}{\sqrt{2\pi}\sigma_{y_i y_j m}}. \tag{3.8}$$

In the following mixture parameter learning discussion we discuss only the association potential $\varphi_i(y_i|\boldsymbol{x})$, and the interaction potential $\psi_{ij}(y_i, y_j|\boldsymbol{x})$ can be derived in a similar manner. For a simple expression, parameters $b_{y_i m}$, $\mu_{y_i m}$, $a_{y_i m}$, and $\sigma_{y_i m}$ are replaced by b_m, μ_m, a_m, and σ_m, respectively.

Features in nature images follow certain statistical distributions. An example is shown in Fig. 3.1. The rows are 7 classes in the Corel image database and the columns are five different features: 3 Lab colors and 2 positions (horizontal and vertical offset from the image center). Although they are different, it could not be classified correctly using traditional non-structural classifiers. Any distribution could be approximated using a mixture of Gaussian [5], so one can use Gaussian mixtures as feature functions, but usually more mixture components are needed if the distribution is far from Gaussian. From Fig. 3.1, we find that the Lab color and location feature distribution of nature image are more likely to be Laplacian mixture rather than a Gaussian mixture.

Assume that there are N training features used in mixture parameter estimation, $\{x^{(1)}, \cdots, x^{(n)}, \cdots, x^{(N)}\}$, where $x^{(n)}$ could be one feature or a set of several features. The class labels $\{y_i\}$, in equation (3.4), for each site i of the training images are known. The index of sites i and features k in equation (3.4) are omitted to simplify expression. Based on the experimental distributions of color and location features, we suppose these features follow Gaussian mixture distributions, *i.e.*, $x^{(n)}|m \sim \mathcal{N}(\mu_m, \sigma_m)$, or

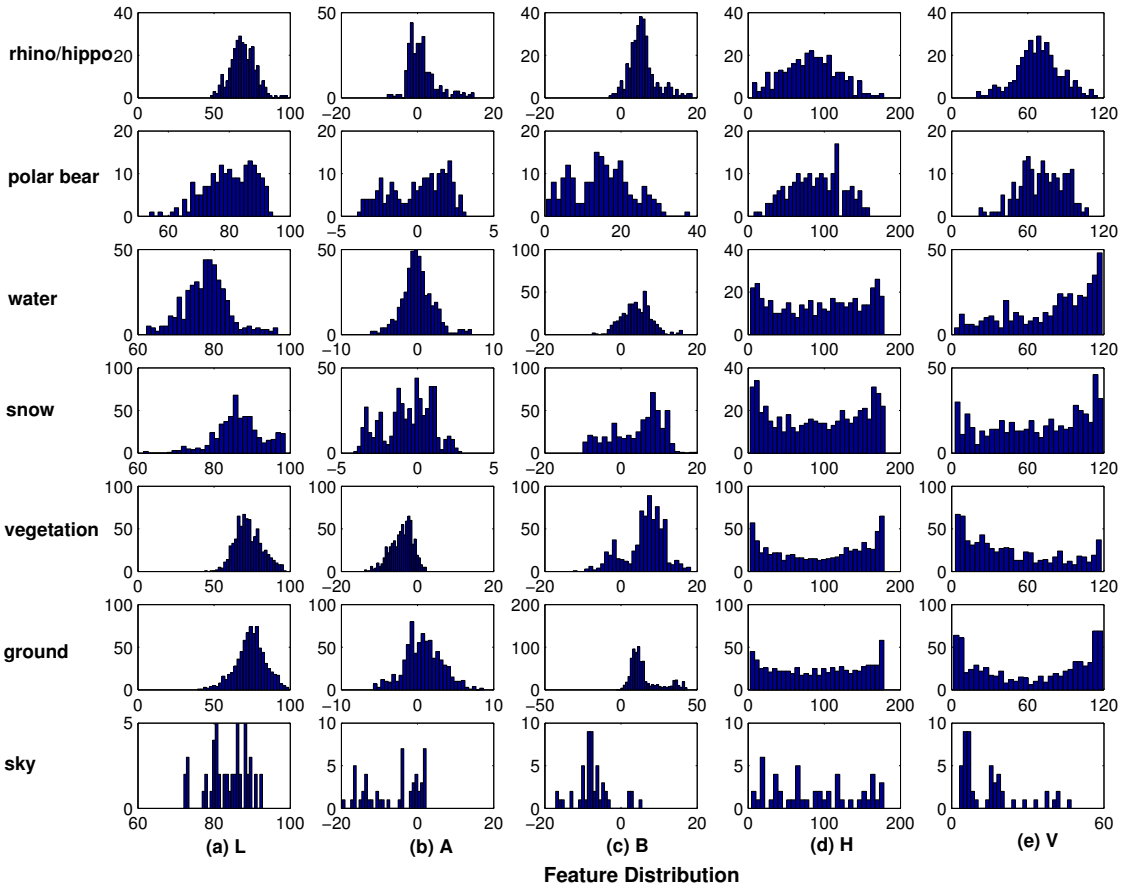


Figure 3.1: Feature distribution of 7 classes of Corel image database. The columns correspond to five different features: Lab colors (L: lightness, A,B: color-opponent dimensions) and positions (H: horizontal and V: vertical offset from the image center), from left to right.

Laplacian mixture distributions, *i.e.*, $x^{(n)}|m \sim \mathcal{L}(\mu_m, b_m)$, where $m = 1, \ldots, M$, $\mathcal{N}(\cdot)$ and $\mathcal{L}(\cdot)$ denote Gaussian and Laplace distribution, respectively. Parameters of the Gaussian distributions $\boldsymbol{\theta}_m$ include the mean μ_m and the variance σ_m. Parameters of the Laplacian distribution $\boldsymbol{\theta}_m$ include the mean μ_m and the scale parameter b_m. For each training features $x^{(n)}$, suppose the hidden variable $\boldsymbol{z}^{(n)} = (z_1^{(n)}, \ldots, z_m^{(n)}, \ldots, z_M^{(n)})$, $n = 1, \ldots, N$. If the data is generated from the mth component of the mixture, all elements of $\boldsymbol{z}^{(n)}$ are zeros except the mth element, which equals one.

To find the most probable parameters for a certain number of mixtures, the log likelihood of the joint distribution needs to be maximized. In most cases when the parameter learning process with joint distribution is not tractable, the EM (Expectation-Maximization) algorithm provides an effective solution. The EM algorithm is an iterative process with two steps in each iteration: expectation calculation step (E-step) and maximization step (M-step). We formulate EM algorithm [20] to estimate Gaussian and Laplacian mixture parameters as follows (The calculation details of Laplacian mixture parameter estimation is shown in Appendix B). The detail steps using EM algorithm for parameters training in a Gaussian mixture can be found in [65][10].

- **E-step**. In this step likelihood functions are calculated with initial guess of parameters or from previous maximization step,

$$P(x^{(n)}|z_m^{(n)} = 1; \boldsymbol{\theta}_m) = \begin{cases} \frac{\exp(-\frac{(x^{(n)}-\mu_m)^2}{2\sigma_m^2})}{\sqrt{2\pi}\sigma_m} & \text{Gaussian;} \\ \frac{\exp(-\frac{|x^{(n)}-\mu_m|}{b_m})}{2b_m} & \text{Laplacian.} \end{cases}$$

  $m = 1, \ldots, M$ and $n = 1, \ldots, N$, and then the expectation values of $z_m^{(n)}$ with

respect to likelihood function are calculated as follows,

$$\langle z_m^{(n)} \rangle = \frac{P(x^{(n)}|z_m^{(n)}=1;\boldsymbol{\theta}_m)a_m}{\sum_{j=1}^{M} P(x^{(n)}|z_j^{(n)}=1;\boldsymbol{\theta}_j)a_j}, \tag{3.9}$$

where $\langle \cdot \rangle$ represents the expectation.

- **M-step**. In this step parameters that maximize the expectation are found,

$$\hat{\mu}_m^{(l)} = \begin{cases} \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle x^{(n)}}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle} & \text{Gaussian;} \\ \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{x^{(n)}}{|x^{(n)}-\hat{\mu}_m^{(l-1)}|}}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{1}{|x^{(n)}-\hat{\mu}_m^{(l-1)}|}} & \text{Laplacian.} \end{cases}$$

$$\begin{cases} \hat{\sigma}_m = \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle (x^{(n)}-\hat{\mu}_m^{(l)})^2}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle} & \text{Gaussian;} \\ \hat{b}_m^{(l)} = \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle |x^{(n)}-\hat{\mu}_m^{(l)}|}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle} & \text{Laplacian.} \end{cases}$$

$$\hat{a}_m^{(l)} = \frac{1}{N}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle,$$

where $m = 1, \ldots, M$, $l = 1, \ldots, L$, and L is the number of iterations. The parameter $\hat{\mu}_m^{(l)}$ is the estimation of μ_m after the lth iteration, based on the estimate value $\hat{\mu}_m^{(l-1)}$ after the previous $(l-1)$th iteration. The E-step and M-step are performed alternatively until the parameter estimation has converged.

With known class labels for each site of the training images, one can group the features of the same label and use the EM algorithm to calculate parameters of the label. When class labels are known for one site and its neighboring site, the parameters can be learned for their label interaction. Knowing these parameters the feature function of the model can be calculated using the Equation (3.4) for both training and inference of CRF. Once these functions are known, the belief propagation (BP) inference and stochastic gradient descent (SGD) weight learning can be applied in this new model.

3.2.2 Mixture CRF Training and Inference

The belief propagation and stochastic gradient descent are used for inference and training of the new model. In the belief propagation algorithm [104], the $\exp(\varphi_i(y_i|\boldsymbol{x}))$ and $\exp(\psi_{ij}(y_i, y_j|\boldsymbol{x}))$ are the node belief and the edge belief of the message passing, respectively, these both depend on labels. The parameters of each mixture for different labels are known before CRF training. As long as the weights $\boldsymbol{w} = \{w_{ik}, w_{ijk}\}$ are given, the beliefs for each type of features can be calculated .

All weights of the CRF model $\boldsymbol{w}$ are obtained using stochastic gradient descent. Here all weights are assumed the same for all sites, this approach is a tangible training solution. Since the new model is log-linear, one can use the stochastic gradient descent to maximize the conditional log-likelihood (CLL). The parameters are updated based on a batch of training examples each time. In our experiment, the number of training images in a batch is set to be 3. There is one weight for each mixture in the new CRF model. The partial derivative of the conditional log-likelihood $\log P(\boldsymbol{y}^{(n)}|\boldsymbol{x}^{(n)}; \boldsymbol{w})$ with respect to the weight w_k (that could be w_{ik} or w_{ijk}) is calculated as follows [23].

$$
\begin{aligned}
&\frac{\partial}{\partial w_k} \log P(\boldsymbol{y}^{(n)}|\boldsymbol{x}^{(n)}; \boldsymbol{w}) \\
&= f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) - \frac{\partial}{\partial w_k} \log Z(\boldsymbol{x}^{(n)}, \boldsymbol{w}) \\
&= f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) \\
&\quad - \tfrac{1}{Z(\boldsymbol{x}^{(n)}, \boldsymbol{w})} \textstyle\sum_{\boldsymbol{y}^{(n)\prime}} \frac{\partial}{\partial w_{k'}} \exp \sum_{k'} w_{k'} f_{k'}(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)\prime}) \\
&= f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) \\
&\quad - \textstyle\sum_{\boldsymbol{y}^{(n)\prime}} f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)\prime}) \frac{\exp \sum_{k'} w_{k'} f_{k'}(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)\prime})}{\sum_{\boldsymbol{y}^{(n)\prime\prime}} \exp \sum_{k''} w_{k''} f_{k''}(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)\prime\prime})} \\
&= f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) - \langle f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)\prime}) \rangle_{P(\boldsymbol{y}^{(n)\prime}|\boldsymbol{x}^{(n)}; \boldsymbol{w})}.
\end{aligned}
\tag{3.10}
$$

Here n is the current training example and both $\boldsymbol{y}^{(n)\prime}$ and $\boldsymbol{y}^{(n)\prime\prime}$ represents the possible

labels. The $f_k(\cdot)$ ($f_{ik}(\cdot)$ or $f_{ijk}(\cdot)$) are the feature functions in the equation (3.4) and $P(\boldsymbol{y}^{(n)'}|\boldsymbol{x}^{(n)};\boldsymbol{w})$ is the conditional probability of label $\boldsymbol{y}^{(n)'}$ given the weights $\boldsymbol{w}$ and features $\boldsymbol{x}^{(n)}$. According to this partial derivative, the weights w_k are updated iteratively as

$$w_k - \eta(f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) - \langle f_k(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\rangle), \tag{3.11}$$

where η is the learning rate. The weight change is proportion to the value of the feature function for the known label $\boldsymbol{y}^{(n)}$ minus the average value of the feature function for all possible labels $\boldsymbol{y}^{(n)'}$. Here a penalty term should be added as in Equation (2.27). Since the belief propagation method is used for inference, the probability of all alternatives $\boldsymbol{y}^{(n)}$ for each node and edge can be obtained during this process.

3.3 Mixture CRF based Image Labeling

Based on previous analysis, the new CRF model with a Gaussian or Laplacian mixture can be applied in image labeling tasks. Since the CRF model is computational intensive, this new model is apply on superpixels instead of pixels, to reduce the complexity. Images are first oversegmented to superpixels. Then features are generated for both training images and the image to be labeled. Distribution parameters are learned from training features, and the feature functions are calculated for training data. Then weights are learned using stochastic gradient descent, and the feature functions are generated for test images. The image label inference is done with belief propagation.

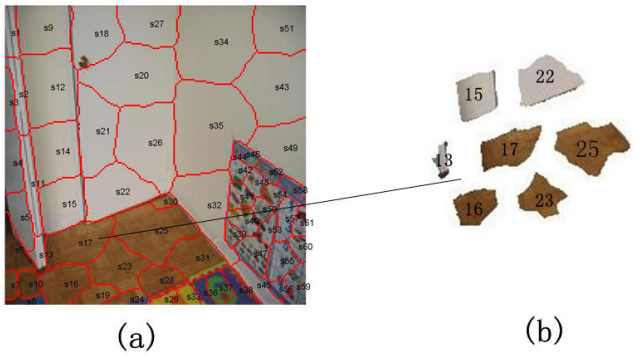


Figure 3.2: An example of superpixel.

3.3.1 Superpixel

The task of image labeling is to find an appropriate content label for every image pixel. This approach is highly redundant, since most likely a pixel belongs to the same object category as the neighboring pixel. Here the mixture CRF model is built on small homogenous segments called superpixels [34, 36, 28], which is a composition of a small group of neighboring similar pixels [79, 66]. Bottom-up normalized cut, which is an oversegmentation of images, is utilized to generate this image representation by multiple superpixels. With a large number of small regions, the potential error induced by such a oversegmentation is relatively small. Although the superpixel graph of an image is irregular in nature (see an example in Fig. 3.2(a)), this model is built by making the pairwise relationship compatible with the irregular shape. Fig. 3.2(b) shows that the superpixel 17 has six neighbors 13, 15, 16, 22, 23 and 25. These six superpixels form the set of neighbors for superpixel site 17. Although a superpixel is still a small part of a image, the number of nodes of the graphical model used is greatly reduced. After reducing the number of node used in the graphical model, the computational burden of the CRF training and inference is much relaxed.

3.3.2 Steps of Mixture CRF Image Labeling

The basic steps of image labeling using a mixture model are listed as follows.

- *Step 1: generating the training data superpixel graph and features.* Training images are oversegmented to superpixels and features of each superpixel are generated. Various kinds of features are used. Of particular interests are those features that can be modeled by a Gaussian or Laplacian mixture. In nature images, many features such as color and position follow a mixture model especially the Laplacian mixture as shown in Fig. 3.1.

- *Step 2: learning mixture parameters.* Superpixel features are grouped by training data superpixel classes. EM algorithm is used to compute the parameters of Gaussian and Laplacian distributions. After obtaining features from training images, the mixture parameters for each class and neighboring class combination can be calculated. Parameters of the mixture for each class are used to calculate the associate potential feature functions. Parameters for each class combination are for interaction potential feature functions. Parameters of these mixtures are learned by the EM algorithm from the training data before the CRF training. Each feature function for each class of association potential is represented by two component mixtures. Each feature function of each neighboring class combination for interaction potential is also represented by two component mixtures.

- *Step 3: training the CRF.* The potential feature functions are calculated by using parameters learned in Step 2. In this step the stochastic gradient decent training is performed iteratively. The CRF weight parameters for potential

feature functions are learned.

- *Step 4: generating superpixel graph and features of testing images.* The potentials are computed using mixture parameters learned from Step 2 for inference.

- *Step 5: performing the inference of the testing image superpixel labels using belief propagation.* In belief propagation, the messages are passed from one node to another based on the probability calculation of nodes and edges.

3.4 Experimental Results

To evaluate the performance of this new CRF model with Gaussian and Laplacian mixtures, image labeling experiments were conducted on the commonly-used 100-image subset of the Corel image database [2]. There are seven classes, rhino/hippo, polar bear, water, snow, vegetation, ground, and sky. The task is to recognize and segment these 7 classes. The database has 100 images, the images have 180×120 pixels. In the experiment, the database is divided randomly to 50 training and 50 testing. Due to the fact that the pixel-based CRF is computationally intensive, the new mixture CRF is built on superpixels, similar to [34, 28]. Each image is segmented to roughly 60 superpixels. The number of superpixels is chosen for all image labeling experiments in this thesis because of the image size used. The number affects the performance [28] but it does not affect our comparison.

The features used in the new model are constructed from low-level descriptors. For each superpixel, a feature vector with five components (Lab color and locations) is computed. The exact feature value of a superpixel is the mean over all pixel feature values of this superpixel. A bias term 1 is always added to the feature vector.

Altogether the association potential has six original features. The interaction feature vector is calculated as the absolute difference of the two neighboring superpixels' features. Adding the bias term, it is also a original feature vector with six components. With the original superpixel features of the training images, parameters of mixtures for both association and interaction potentials can be calculated accordingly.

With these estimated mixture parameters, the mixture CRF training and inference are performed for image labeling. Here the learning rate η is fixed to be 0.0001. Starting with random weights, the stochastic gradient descent algorithm converges after about 10 iterations for the Laplacian mixture CRF. Table 3.1 shows the confusion matrix of the new model comparing to the baseline CRF using logistic potential feature functions and SVM classifier with the same number of features [50]. Note that in the baseline CRF the quadratic expansion of the features is used and has desirable results. For every class, the performance of the Laplacian mixture CRF is better than the Gaussian mixture CRF and the baseline CRF.

Fig. 3.3 shows the receiver operating characteristic (ROC) curve comparison between the mixture CRFs and the baseline CRF. The ROC curves plot the false positive rate versus the true positive rate. True positive rate is the rate of classifying positive instances correctly among all positive samples available during the test. False positive rate is the rate of classifying negative instances wrongly among all negative samples. Note that average values of detection rates for multiple classes are used in ROC computation for this multi-class case. If the true class and predicted class are the same, it is called a match. Suppose for the lth ($l = 1, \ldots, 7$) class of images with 7 classes,

- the number of correct matches (classification) is the true positives T_{pl},

Table 3.1: Confusion matrix of new mixture CRF model on Corel dataset

	rh/hi	pb	wa	sn	ve	gr	sk
rhino/hippo	**79.9** {70.4} (73.2) [48.7]	0 {0} (3.6) [0]	5.6 {0.6} (14.8) [14.1]	0 {0.4} (1.9) [1.9]	4.3 {11.4} (2.8) [25.2]	10.2 {8.6} (3.7) [10.3]	0 {0} (0) [0]
polar bear	0 {0} (1.7) [1.0]	**72.5** {71.6} (56.8) [0]	0 {0} (5.1) [8.3]	14.8 {14.0} (17.3) [16.7]	5.1 {0.9} (9.8) [5.6]	7.6 {5.5} (9.3) [68.5]	0 {0} (0) [0]
water	5.8 {15.1} (6.5) [2.4]	0.2 {1.9} (0.7) [0]	**67.6** {65.5} (62.3) [46.7]	3.7 {4.6} (14.1) [21.7]	13.0 {6.3} (12.7) [13.2]	7.4 {3.9} (0.5) [16.0]	2.3 {2.8} (3.2) [0]
snow	0.2 {3.4} (0.2) [0]	0.9 {1.1} (0.7) [0]	8.2 {13.2} (21.1) [12.2]	**86.1** {77.6} (71.9) [76.0]	2.7 {3.2} (0.3) [3.8]	1.7 {1.4} (0.5) [6.9]	0.2 {0} (5.3) [0.1]
vegetation	4.6 {14.2} (6.1) [3.2]	1.9 {2.7} (3.4) [0]	4.9 {5.3} (13.8) [4.3]	6.7 {9.6} (9.6) [7.8]	**67.4** {59.6} (54.5) [74.1]	11.7 {5.9} (7.3) [8.1]	2.8 {2.5} (5.3) [2.6]
ground	6.4 {21.5} (12.4) [3.8]	4.1 {5.5} (6.6) [0]	4.7 {4.2} (12.3) [4.3]	3.9 {6.7} (11.5) [9.6]	12.5 {10.5} (9.4) [10.1]	**67.9** {51.5} (47.1) [72.3]	0.5 {0.1} (5.7) [0]
sky	0 {0} (0) [0]	0 {0} (0) [0]	0 {0} (0) [0]	0 {0} (0) [38.5]	10 {20.0} (11.8) [15.4]	3.3 {0} (2.0) [0]	**86.7** {80.0} (86.2) [46.2]

Note: Accuracy of the Laplacian mixture CRF on the 7-class Corel database. The confusion matrix shows the pixel-wise recall accuracy (%) for each class and is row normalized. Row labels are the true classes and column labels are the predicted classes. The second number in braces in each cell shows the Gaussian mixture CRF result. The third number in parentheses in each cell shows the baseline CRF result. The fourth number in square bracket in each cell shows the nonstructural SVM classifier result.

- the number of matches that were not correctly detected is the false negatives F_{nl},
- the number of non-matches that were not correctly rejected is the false positives F_{pl},
- the number of non-matches that were correctly rejected is the true negatives T_{nl}.

Then the average true positive rate and false positive rate are

$$T_p = \frac{\sum_{l=1}^{7} T_{pl}}{\sum_{l=1}^{7}(T_{pl}+F_{nl})}, \tag{3.12}$$

$$F_p = \frac{\sum_{l=1}^{7} F_{pl}}{\sum_{l=1}^{7}(F_{pl}+T_{nl})}. \tag{3.13}$$

The average accuracy rate of this image labeling task is defined as the percentage of image pixels assigned to the correct labels for all seven classes in the Corel database, *i.e.*,

$$A_c = \frac{\sum_{l=1}^{7}(T_{pl}+T_{nl})}{\sum_{l=1}^{7}(T_{pl}+F_{nl}+F_{pl}+T_{nl})}. \tag{3.14}$$

With only 5 features the overall accuracy classification rate of our Laplacian mixture model is 75.4% which is comparable with the state-of-the-art results (in the range of $70\% - 80\%$) in [85, 34, 28]. Previous papers usually have at least around 100 features and many different types of potential feature functions which prevents efficient learning and increases the difficulties for reproducing the results. With the same number of features the accuracy of the baseline CRF is 64.6% and Gaussian mixture mode is 68.04%. The classification performance of support vector machine (SVM) using LIBSVM [13] which is one of the best nonstructural model is also known. The accuracy of SVM classification is 61.7% which is less than CRF structural model. This proves

the performance advantage of applying CRF graphical model in image labeling over nonstructural model. The results indicate that the use of Laplacian mixture and CRF significantly improves the classification performance.

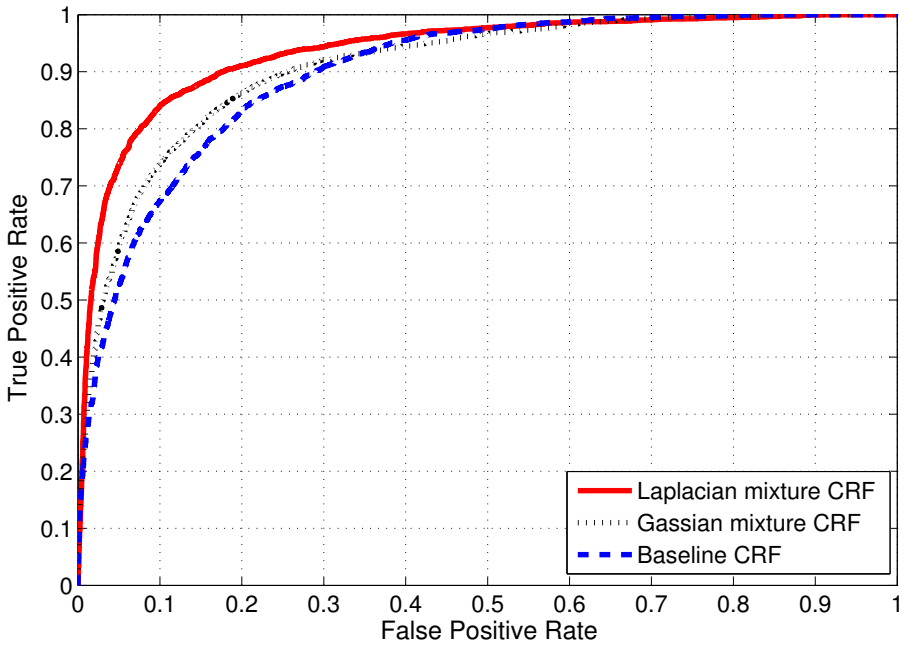


Figure 3.3: ROC curve of the Laplacian mixture CRF, Gaussian mixture CRF, and the baseline for Corel 7-class database.

In Fig. 3.3, the red solid curve indicates the ROC of the Laplacian mixture CRF, the black dotted curve indicates that of the Gaussian mixture CRF, and the blue dashed curve shows that of the baseline CRF. It is evident from Fig. 3.3 that the ROC plot of the Laplacian mixture CRF is closest to the upper left corner than that of the Gaussian mixture CRF and the baseline CRF. Therefore, the Laplacian mixture CRF model has a highest overall accuracy. The AUCs (Area under ROC curve) of these three methods are: Laplacian mixture CRF, 93.65%, Gaussian mixture CRF, 90.95%, and Baseline CRF, 89.91%. AUC is the area under the ROC curve which is a

usually used performance indicator. It means the probability that when we randomly pick one positive and one negative example, the classifier will assign higher score to the positive example than the negative example. It can also be observed from Fig. 3.3 that the performance of the Gaussian mixture CRF is better than that of the baseline CRF.

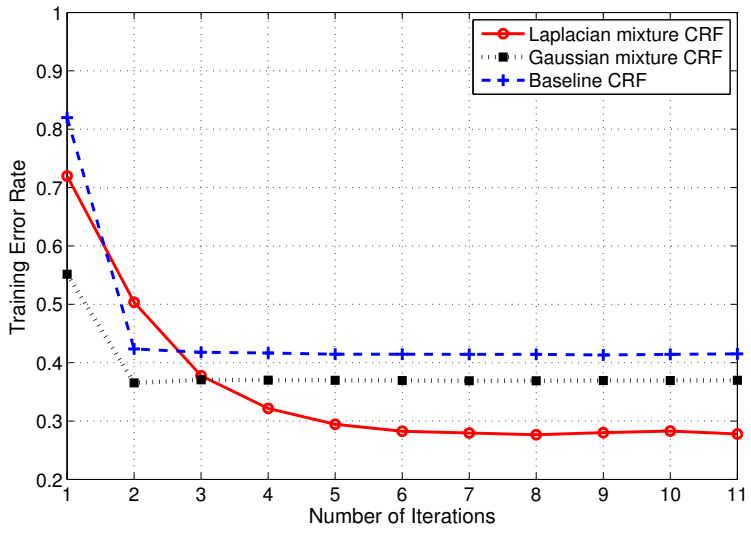


Figure 3.4: Learning curves of the Gaussian mixture CRF, Laplacian mixture CRF and the baseline CRF, using Corel 7-class database.

Fig. 3.4 compares the learning curves of the Laplacian mixture CRF (red solid curve), the Gaussian mixture CRF (black dotted curve), and the baseline CRF (blue dashed curve). The learning curves show the test errors as a function of iterations in the training process. The Laplacian mixture CRF achieves lower test errors after 4 iterations compared with the baseline CRF and Gaussian mixture CRF.

To have a qualitative analysis, the performance is shown in Fig. 3.5. These figures show that with the same features both qualitative and quantitative results of both the Laplacian mixture CRF and the Gaussian mixture CRF perform better than those

of the baseline CRF.

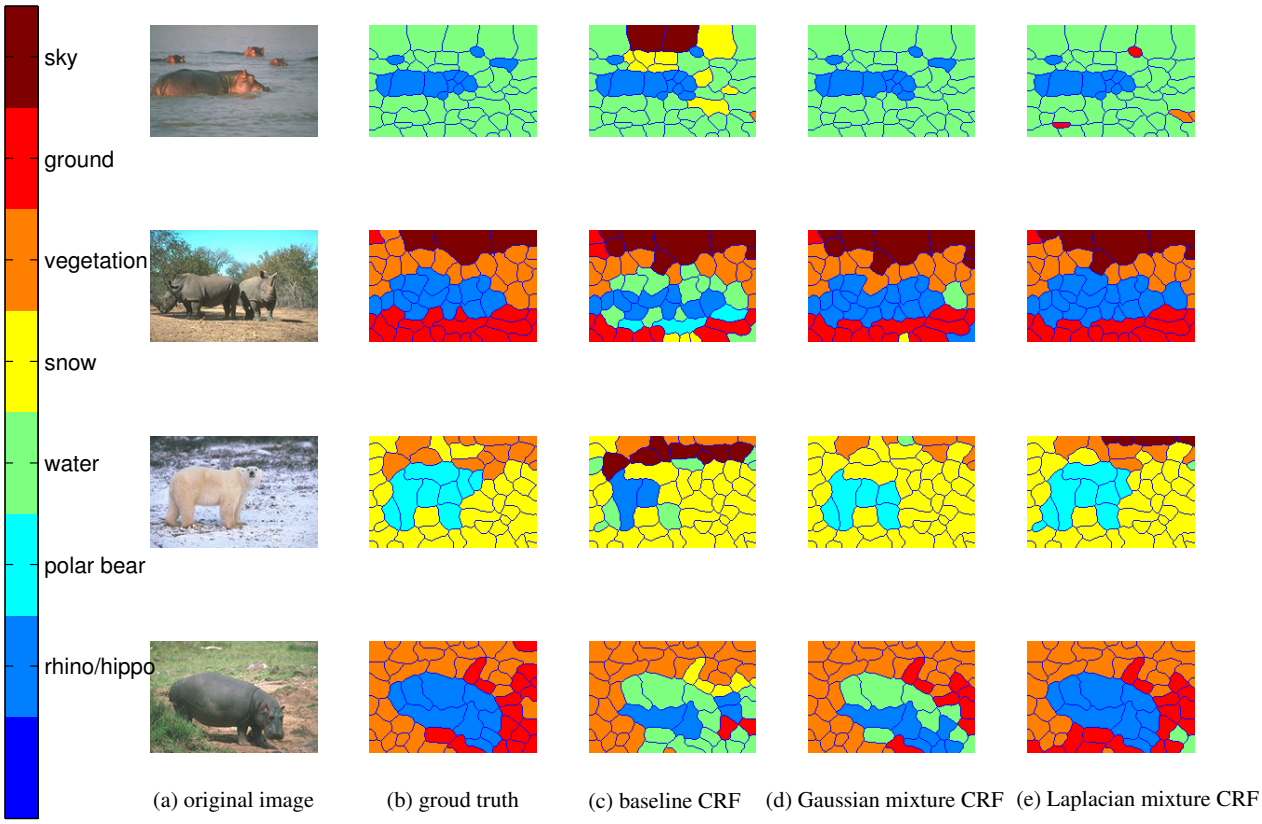


Figure 3.5: Some labeling results for nature images in the Corel dataset using the Gaussian mixture, Laplacian mixture and baseline CRF.

To demonstrate the effectiveness of the new feature selection, performances of using Lab color feature only (green dashed curve), position feature only (black dotted curve) and all features (red solid curve) are compared in Fig. 3.6. The Lab color features are more useful than the position features in the image labeling of nature images. Obviously, the combination of two kinds features are better than any single set of features.

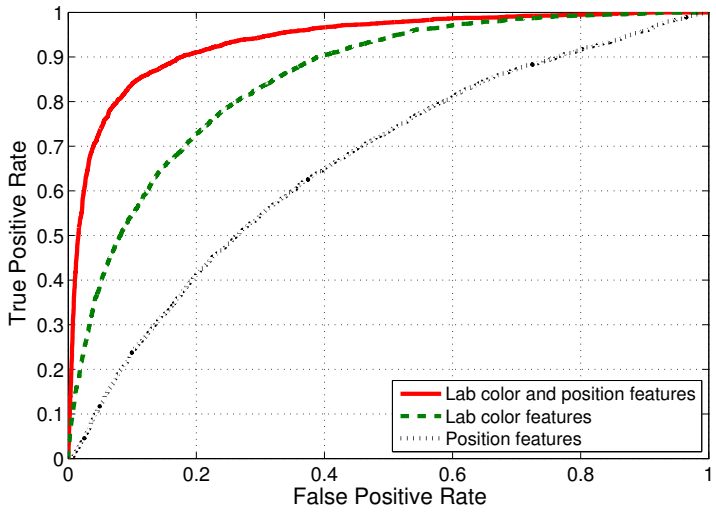


Figure 3.6: ROC curve of the Laplacian mixture CRF with different features for Corel 7-class database.

The better performance of the Laplacian mixture model are due to three reasons. First, the nature image features most likely follow Laplacian mixture distribution as shown in Fig. 3.1. Second, the EM algorithm effectively extracts the parameters of the distribution. Third, the feature selection is effective. Both the class and class combination feature functions contribute the increasing accuracy of classification. The combination of Laplacian mixture and CRF provides effective and efficient solution for image labeling.

3.5 Discussions

A new image labeling model based on mixture CRFs is introduced in this chapter. After analyzing the distribution of features of nature images, we apply Gaussian or Laplacian potential feature functions to model associations and interactions in CRF. The model takes advantage of both the unstructured Gaussian or Laplacain

feature distribution and structural discriminative CRF model. The combination of the two provides a new framework for nature image labeling. Also instead of modeling different potential feature functions in CRF differently we represent all potential feature functions in the mixture format. The training of the new CRF is performed by stochastic gradient descent. Belief propagation inference is used to infer the most probable labels. To test the effectiveness of the new model it is applied to nature image labeling of seven class Corel database, with the same number of features as the new Laplacian mixture CRF shows performance improvement over the Gaussian mixture CRF and the baseline CRF. Our model with only several features shows comparable result with the other CRF models with hundreds of features. Although the results are preliminary, not superior to other complex models and more simulations need to be done for other databases, the new mixture CRFs put the image labeling problem in a new way which is not seen in other literatures as far as we know. The new mixture CRF model is a general framework with the advantage of classification accuracy rate and training simplicity, which can be applied to other applications related to multimedia content analysis. Future works include improving the performance by incorporating more relevant features, testing the method for other more complex databases and further reducing the overall computational complexity by using other approximate learning algorithms.

Chapter 4

CIBR-Based CRF Model for Image Labeling of Large Database

In this chapter, the problem of image labeling with large training databases is investigated. A new image labeling approach that implicitly incorporates top-down information using content-based image retrieval (CBIR) with conditional random field (CRF) model is presented.

While providing more information, large labeled training databases with various kinds of images posses new challenges such as content ambiguities. It is difficult to extract content probabilistic model from a large image database. To reduce the content ambiguities and increase the recognition accuracy, large image databases are reduced to small relevant ones by using the content-based image retrieval (CBIR) models in this thesis.

CBIR is a querying system using image content such as low-level features and high-level semantic content [82, 57, 12, 18]. It finds applications in computer vision and becomes popular because it could be applied in mining digital images in

large databases. The CBIR system provides a solution by examining image content. Content-based analyzes of the content that can be found from the image itself and may refer to colors, shapes, textures, or other information. It overcomes disadvantage of the traditional search, which is based on metadata such as keywords and is laborious, expensive and sometimes subjective.

The proposed method in this chapter is devised for large labeled training databases by learning the top-down content information with CBIR and integrating CBIR retrieval information with the CRF model. This system has two parts: CBIR image retrieval and CRF image labeling classifier.

- A small content similar training set for CRF labeling is built using retrieved CBIR matches from a large image database. The top-down content information is learned using CBIR features, and the content of the input image is used to select the several most probable content similar images in the labeled database. Since the search is content-based, the top-down information is reflected in the image retrieval results. Content similar images are used as the training set for the image labeling process. The retrieval scores (similarity measure) are used as weights for the global factor in the CRF labeling model in order to reflect the scene similarity.

- In CRF-based image labeling, each node represents a random variable whose labels is to be inferred, and each edge represents a dependency between two random variables, labels and observations. To achieve global consistency of image labeling, we present a novel superpixel-based CRF probabilistic model with a revised global factor. The use of superpixels reduces the bottom-up calculation burden. The loopy belief propagation [104, 69] and stochastic gradient descent

[92] are the inference and training algorithm used in our experiments.

The new image labeling framework based on CBIR and CRF is tested using the Labelme database which has a very large number of images, these images where labeled by other researchers under a uncontrolled environment. Test results show that the new image labeling model based on CBIR and CRF demonstrates promising results, compared with the CRF approach without retrieval.

This chapter is organized as follows. First the idea of applying CBIR to image labeling is presented in Section 4.1. Then in Section 4.2 we propose a new superpixel CRF model which incorporates CBIR top-down information from Section 4.1. After that simulation results are given to prove our analysis in Section 4.3. Finally we conclude the chapter with discussions and future research directions in Section 4.4.

4.1 CBIR for Image Labeling

In image labeling, the content of the training database plays a central role for accurately labeling the input image. For simplicity the layout and precisely labeled images are in a small and specific database, it is appropriate to let the learning methods understand the content themselves. For large and uncontrolled circumstances, the problem becomes troublesome because of the content ambiguities and training image labeling errors. With the increase of the database size, the semantic meaning of the content gets more ambiguous and the labeling error increases. There are two ways to deal with this problem. The first is to build a superior machine learning algorithm. This approach is not realistic based on the current technology because of the computational complexity. The secondary is to select a subset of relevant training images to train the image labeling model. In this chapter we take the second

approach. We present a new method in which CBIR is used to choose the relevant images.

Traditional retrieval methods are based on text search, *e.g.*, keyword search. For example, if one wants to label the floor area of an input image from a large database, the "floor" is the keyword in the database search. Keyword search usually returns a lot of images, sometimes over thousands, from a large database, and most of them are different from the input image to be labeled. If all images chosen by the keyword search are used to train the image labeling classifier, parameters learned will be far from the model that the input image actually belongs to or the model is too complex to be inferred. The keyword search provided is a high level that includes concept ambiguities. Keyword could mean something totally different such as the floor lamp. Even with the same meaning the floor has different appearances and follows a different model in different kinds of scenes in large databases. For example, the floor in the kitchen is different from the floor in the hallway. Traditionally the topic or the scene content is retrieved and a (Bayesian or random field) probabilistic model with a hierarchical structure is built to solve the content ambiguity [35, 90, 56, 9, 30]. However, this kind of method is highly complex and only useful for small databases with limited classes.

Without dealing with a very high level abstract concept, we present a new approach to provide a better training set for image labeling of a large image database. A new retrieval system for content-based image retrieval (CBIR) that reflects top-down information, is built for the purpose of improving labeling accuracy. The CBIR system takes a single input image, retrieves content similar images from the database, and uses these images as the training set for the image labeling. The search by the

CBIR system is based on the content information of the images. Since the large labeled image database also has the keyword information, one can use keyword search as the preprocessing step of a CBIR system for image labeling. A CBIR system can be divided into two components: signature extraction to describe an image mathematically and similarity measure to assess the similarity of two images given the abstracted expression of the images.

Signature Extraction

In this chapter, color, texture and salient features are used in the CBIR system.

- *Color features.* The color histogram of red, green, and blue (RGB) color space is applied in our system. With 10 bins for each color there are 30 color features. Although RGB may not be effective as other color spaces with respect to other applications, in large database it exhibited good global indicator for similarity of images.

- *Texture features.* We first transform a RGB image to gray scale, then apply the Leung-Malik (LM) filters [55], then take mean response of the image, to get texture features. The LM filter bank set is 48 filters in multiple scales and multiple orientations. It includes first and second derivatives of Gaussians at 6 orientations and 3 scales which makes a total of 36, 8 Laplacian of Gaussian (LOG) filters, and 4 Gaussians. Altogether there are 48 texture features.

- *Interest point features.* We use the scale invariant feature transform (SIFT) [61] feature vector which is proved to be very useful in object recognition. Each interest point has a SIFT feature of the size 128. The principle components

are obtained using principle component analysis (PCA) [44] to indicate interest point features. For an image we have a salient point feature vector with 128 elements.

Altogether there are 206 features used in the CBIR system. Color features are designed to define the overall color distribution of the images. Texture features are used to reflect the global texture of the images. Interest point features are used to gain the support of object recognition. These features carry top-down meaning of the whole image. CBIR finds the meaning of the content information implicitly. Given an input image, the feature vector needs to be computed and compared with the signatures of images in the database.

Similarity Measure

A multivariate Gaussian similarity measure (retrieval score) as in paper [18] is used in the CBIR system. In the CBIR retrieval system, given an input image feature vector $\boldsymbol{v}_I$, a retrieval score for each image in the database is defined as

$$D(\boldsymbol{v}) = \exp\left(-\frac{(\boldsymbol{v}-\boldsymbol{v}_I)^T\boldsymbol{\Sigma}(\boldsymbol{v}-\boldsymbol{v}_I)}{2}\right) \tag{4.1}$$

where $\boldsymbol{v}$ is the global feature vector for an image in the image database, and the superscript ^{T} denotes the transpose of a vector. Here $\boldsymbol{\Sigma}$ is the similarity matrix with the adjustable weights on specific color, texture and salient features. The retrieval score $D(\boldsymbol{v})$ is a multivariate Gaussian distance measure that reflects the similarity between retrieved images and the input image. If the retrieved image is the same as the input image, the retrieval score will achieve its highest score, *i.e.*, 1. Each feature in the vector is normalized to zero mean before the distance calculation to reduce the disparity among features. In our experiment the $\boldsymbol{\Sigma}$ is set to be the inverse covariance

matrix of features. For each feature we assume the feature of $\boldsymbol{v}_I$ is the mean value. The $\boldsymbol{\Sigma}$ is calculated based on sample images from the database. The Gaussian like distance measure is easy to be applied to CRF-based image labeling presented in the next section.

With the feature extraction and similarity measure the CBIR system can find the content similar images in the database based on the ranking of distance measures. Images with highest retrieval scores are retrieved from the large database and used in the labeling process. The number of images can be determined by the distance or percentage. For example, top ten percents of retrieval results are selected as the matched image data set for labeling. The training set is now determined by CBIR.

4.2 A New Superpixel CRF Model with CBIR Top-down Information

The conditional random field (CRF) model is widely used in image labeling. To solve the labeling problem with large database, we present a new CRF image labeling model to incorporate the CBIR similarity score as a weight for the new global factor. To further reduce the complexity of pixel-based CRF, the new model is built on small homogenous segments called superpixels, which is introduced in Section 3.3.1.

4.2.1 A New Superpixel CRF Model with Global Feature weighted by CBIR Score

Both top-down information from CBIR and other superpixel information are used in a new CRF model to merge superpixels into semantic meaningful labeled segments.

Since the similarity between the retrieved image and the input image for labeling is relevant to labeling task, a new global factor is added accordingly in the CRF model to reflect the global similarity. Adding a new term to equation (2.32), the posterior probability in the new CRF model becomes

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{i\in S} \varphi_i(y_i|\boldsymbol{x}) + \sum_{i\in S}\sum_{j\in N_i} \psi_{ij}(y_i, y_j|\boldsymbol{x}) + \sum_{i\in S} \psi_v(y_i|\boldsymbol{x})\right), \quad (4.2)$$

where $\psi_v(\cdot)$ is the new global factor based on CBIR retrieval scores. This new model reflects both the local and the global factor of the image in the retrieved group.

The unary, pairwise and global factors in the new CRF model are defined as follows in factorized form,

$$\varphi_i(y_i|\boldsymbol{x}) = \sum_{k\in\mathcal{K}_u} w_{uk} f_{ik}(y_i|\boldsymbol{x}), \quad (4.3)$$

$$\psi_{ij}(y_i, y_j|\boldsymbol{x}) = \sum_{k\in\mathcal{K}_p} w_{pk} f_{ijk}(y_i, y_j|\boldsymbol{x}), \quad (4.4)$$

$$\psi_v(y_i|\boldsymbol{x}) = \sum_{k\in\mathcal{K}_v} w_{vk} f_{vk}(y_i|\boldsymbol{x}). \quad (4.5)$$

The $\{f_{ik}(\cdot)\}$ and $\{f_{ijk}(\cdot)\}$ are feature factors corresponding to association potentials and interaction potentials, respectively. The k is the index of features, $\mathcal{K}_u$, $\mathcal{K}_p$ and $\mathcal{K}_v$ are sets of all possible indexes k for unary, pairwise and global feature functions. The unary, pairwise and global feature factors have K_u, K_p and K_v features, respectively. The w_{uk}, w_{pk}, w_{vk} are weights for the three kinds of factors.

Color and texture features are used in local unary factors, and the difference of two neighboring superpixels' color and texture are used in local pairwise interaction functions. For a binary classification problem, define the label set as $\mathcal{Y} = \{-1, 1\}$, the featured functions can be defined as $f_{ik}(y_i|\boldsymbol{x}) = y_i x_{i_k}$ for $i \in S$ and $k \in K_u$, and $f_{ijk}(y_i, y_j|\boldsymbol{x}) = y_i y_j \cdot |x_{i_k} - x_{j_k}|$ for $(i, j) \in E$ and $k \in K_p$. Variables x_{i_k} and x_{j_k} are kth features of site i and site j, respectively. To avoid training diverge problem, a

constant scalar 1 is included in the feature vector $\boldsymbol{x}$, which is replaced by $(1, \boldsymbol{x})$ then.

The new global factor $f_{\boldsymbol{v}k}$ is based on location potential in the new CRF model. It is proved that location is one very useful feature globally [28]. Our global feature is defined as

$$f_{vk}(y_i|\boldsymbol{x}) = D(\boldsymbol{v})H_k(y_i, l_i), \tag{4.6}$$

where l_i represents the normalized position of the superpixel inside the image. The function $H_k(y_i, l_i)$ is the global position potential which is the possibility of the superpixel belonging to a certain class given the position. It is calculated from the training data. Fig. 4.1 shows an example of estimated position potential function H for four classes: floor, window, ceiling and wall. The $D(\boldsymbol{v})$ is the similarity measure used in the CBIR. This function is used to weight the global factor for the purpose of better training the model. The global factor is more important during the training for the image which is similar to the input image we want to label. The function $f_{vk}(\cdot)$ indicates the scene similarity affects the confident of the location potential global factor.

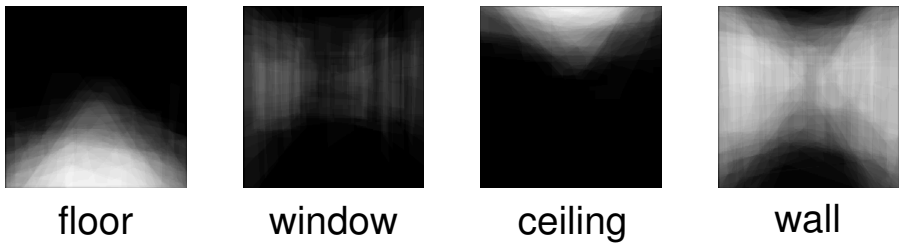


Figure 4.1: Position potentials.

The new CRF model is a linear weighted summation of all local unary, pairwise

interaction and global feature factors as follows in a log-linear form,

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}) &= \frac{1}{Z(\boldsymbol{x},\boldsymbol{w})}\exp\{\sum_{i\in S}\sum_{k\in\mathcal{K}_u} w_{ik}f_{ik}(y_i|\boldsymbol{x}) \\ &+\sum_{i\in S}\sum_{j\in N_i}\sum_{k\in\mathcal{K}_p} w_{ijk}f_{ijk}(y_i,y_j|\boldsymbol{x}) \\ &+\sum_{i\in S}\sum_{k\in\mathcal{K}_v} w_{vk}f_{vk}(y_i|\boldsymbol{x})\}. \end{aligned} \tag{4.7}$$

The $Z(\boldsymbol{x},\boldsymbol{w})$ is a normalizing factor

$$Z(\boldsymbol{x},\boldsymbol{w}) = \sum_{\boldsymbol{y}'\in\mathcal{Y}^N} P(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w}), \tag{4.8}$$

with the weight vector $\boldsymbol{w} = (w_{ik}\ w_{ijk}\ w_{vk})$ to be learned for the feature factors, where N is the number of superpixels in the input image.

The weights are learned from the training set, a subset of large image database, in which all retrieved images have higher retrieval scores relative to other images in the database. For the irregular graph, the problem is ill-posed if the weights are different for different sites since the graph structure is different from one image to another. To approach a tangible solution we assume all weights are the same for all sites. Since the model is assumed log-linear, the stochastic gradient descent algorithm is used to maximize the conditional log-likelihood (CLL) [23] in the CRF model. Parameters are updated based on a batch of training examples for each iteration. There is one weight for each feature in CRF. The partial derivative of the log-likelihood function with respect to each weight w_k is calculated as

$$\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w})}{\partial w_k} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}_s} \left(f_k(\boldsymbol{y},\boldsymbol{x}) - \langle f_k(\boldsymbol{y}',\boldsymbol{x})\rangle_{P(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})}\right), \tag{4.9}$$

where $\mathcal{T}_s$ is a subset of the observation set T. Here $\boldsymbol{y}'$ represents the possible labels and $P(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})$ is the conditional probability of label $\boldsymbol{y}'$ given the weights and features. The feature function $f_k(\cdot)$ can be f_{ik}, f_{ijk}, or f_{vk} depending whether the index k

belongs to the set $\mathcal{K}_u$, $\mathcal{K}_p$ or $\mathcal{K}_v$.

The weight updating rule is as follows,

$$w_k \leftarrow w_k - \eta \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}_s} \left(f_k(\boldsymbol{y},\boldsymbol{x}) - \langle f_k(\boldsymbol{y}',\boldsymbol{x})\rangle_{P(\boldsymbol{y}'|\boldsymbol{x};\boldsymbol{w})} \right), \tag{4.10}$$

where η is the learning rate. The weight changes according to partial derivative of the feature function for the known label minus the average value of the feature function for all possible $\boldsymbol{y}'$. The probability of all alternative $\boldsymbol{y}'$ for each node and edge is obtained in the belief propagation inference. Note that a penalty term should be added also as in Equation (2.27) to overcome overfitting here. The parameter learning is done iteratively.

4.2.2 Steps of CBIR-based CRF Image Labeling

The new labeling approach based on CRF and CBIR provides a better solution for labeling large labeled databases. A flowchart of this algorithm is shown in Fig.4.2. First, the CBIR algorithm, which implicitly provides top-down information for CRF and is performed for the input image. The content search is based on features of both the input image and images in the large training set. The top retrieved matches are served as the real training set for CRF labeling. Second, a new superpixel CRF labeling model is used as classifier to label the input image. The new model is different from the traditional CRF model by adding a position potential global feature. The position potential is weighted by the similarity measures from CBIR and reflects how the similarity affects the global information in CRF. In this way the similarity information is integrated into the CRF model, and the model is simplified to a log-linear form with many feature factors. The parameters of the model are learned from training sets using gradient descent. The general model can be used in common image

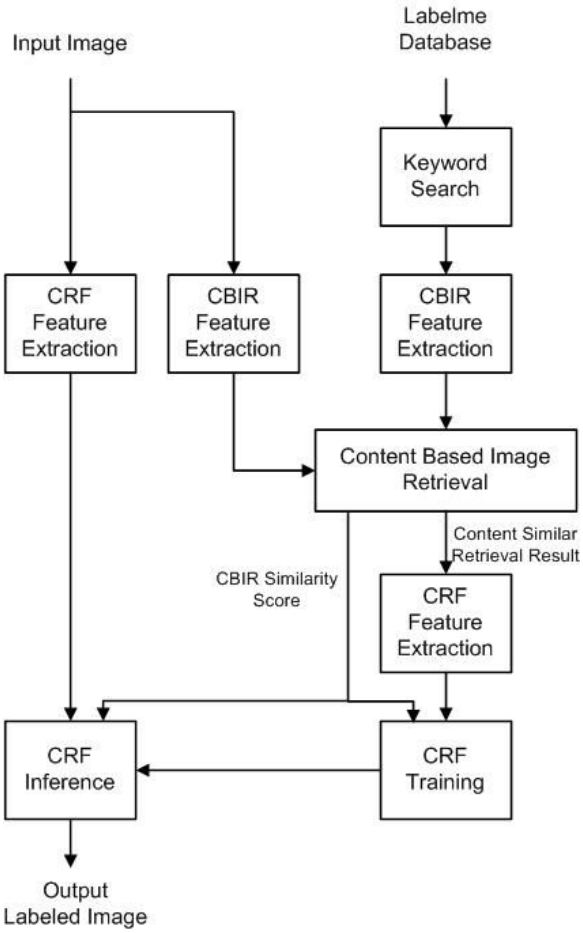


Figure 4.2: Flowchart of CBIR-based CRF image labeling.

labeling tasks.

Besides, the new CBIR-based CRF model can also incorporate the method presented in the previous chapter, by representing all potential feature functions in a Laplacian or Gaussian mixture format.

4.3 Simulation Results

The proposed model is applied to floor area labeling of indoor images. Automatic detection of the floor area is useful for scene understanding and 3D reconstruction of room setting [19, 96]. We evaluate the model using the commonly used Labelme database. The keyword "floor" is used to set up a new smaller database. The new database has 1124 images in total, which have the floor area labeled by other researchers.

The performance of the new CBIR-based CRF model is compared with a traditional CRF model without CBIR. If all 1124 images are used as a training set for the CRF model without CBIR, the CRF training fails to learn parameters properly because of the varieties of database contents. Only 53 representative images are selected as the training set for the traditional CRF learning without retrieval. Some sample images of the new database are shown in Fig. 4.3.

4.3.1 CBIR Results

The CBIR is used in searching the large database to retrieve content similar images. Fig. 4.4 shows four examples of retrieved images using CBIR. Images at the first column are input images. The retrieved top 4 matches are shown accordingly with the similarity score listed above. The similarity matrix $\mathbf{\Sigma}$ is estimated based on 53 images, the top 10 images are used for CRF training. From these examples, one can see that the hallway and room concepts are separated implicitly in the retrieved set. Therefore, it is unnecessary to build a probabilistic model to learn these concepts separately. The top-down information is learned through the CBIR system.

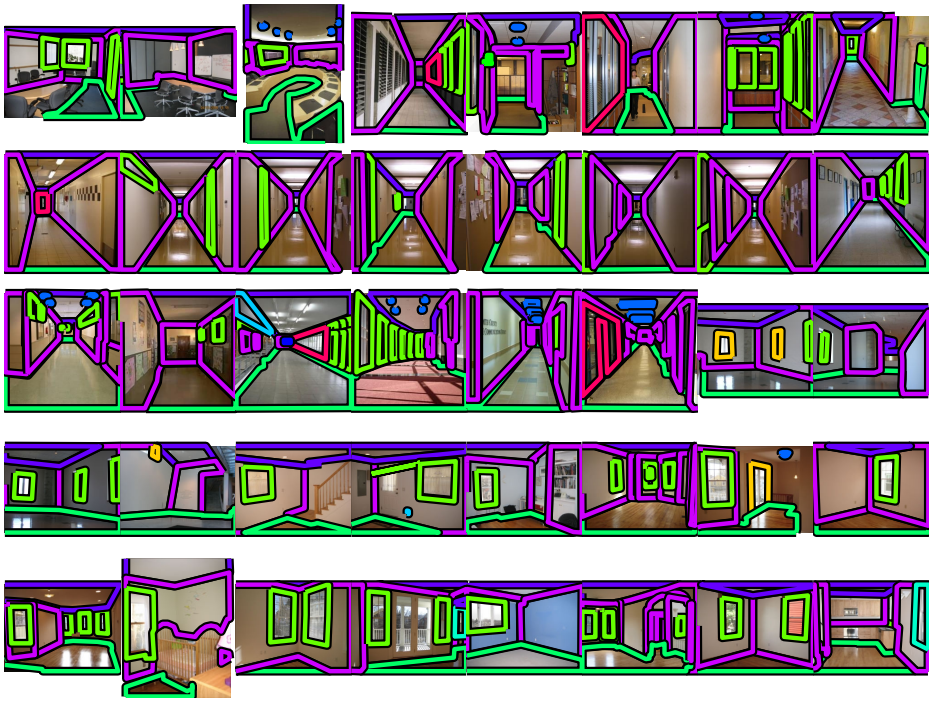


Figure 4.3: Sample images from Labelme database with keyword floor.

4.3.2 CBIR-Based CRF Labeling Results

Fig. 4.5 shows image labeling results of four input images using the new CBIR-based CRF model. The first column contains the four test input images. The second column contains the ground truth segmentation results, which is labeled by other researchers using Labelme [83] tool. The third column contains the CRF labeling result using all 53 images for training. The fourth column contains the CBIR-based CRF result, which uses the CBIR retrieved images for training. The fifth column contains the CBIR-based Laplacian mixture CRF result. The features used in CRF models are the same which includes CIELab color features, edge percentage of the superpixel, as texture features for both the unary and pairwise feature factor. For CRF models combined with CBIR, additional position potential weighted by CBIR retrieval scores

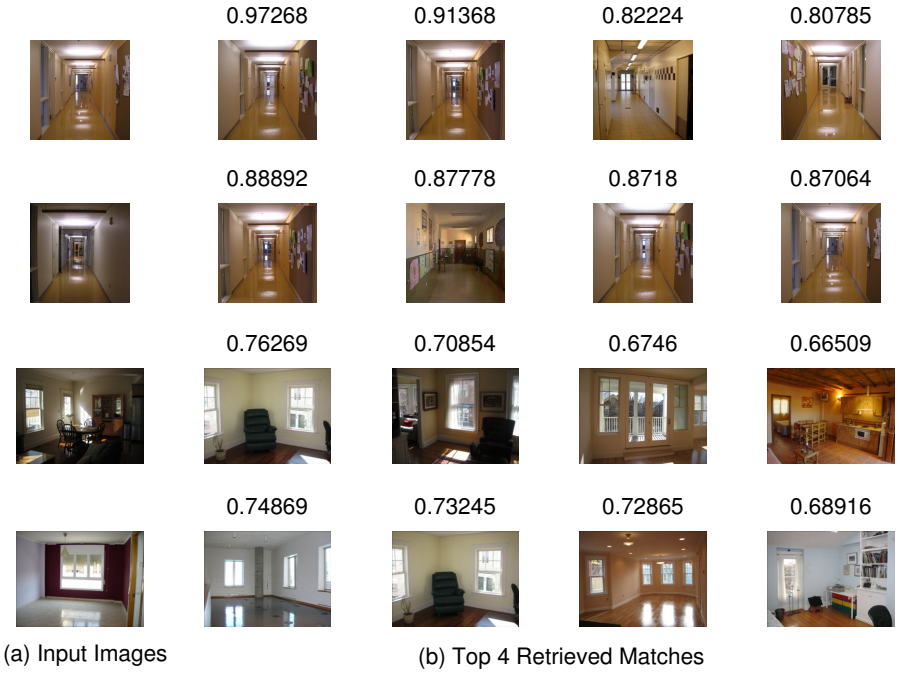


Figure 4.4: Example CBIR results and corresponding retrieval scores.

is used as a global feature. Fig. 4.5 also shows the superpixel structures. The number of sueprpixels in each image is set to be around 60. The number of iterations for training is set to be 100, and the learning rate η is 0.001.

The results prove that using the CBIR to select a small training set improves the image labeling accuracy. With 53 images to train the model it could not find the floor area for two of four images. For all four images, the performance of the new method is better than the CRF model without CBIR.

The average accuracy rate of the new CBIR-based CRF model for 1124 images, as shown in Table 4.1, is much higher than the average accuracy without CBIR. The average accuracy of floor means the percentage of floor pixels that is correctly labeled as floor. The labeling error is significantly reduced by using CBIR-based selective

Figure 4.5: Example results of floor labeling.

training and weighted global factor. By introducing Laplacian mixture into the CRF model, the performance of CBIR-based Laplacian CRF model is further improved, especially for the third input figure shown in Fig. 4.5.

Fig. 4.6 shows the learning curve of CRF labeling for input image 1 of the Fig. 4.5. Start from random weights, the initial error rate is high, and after several iterations the error rate of both methods reduces to a stable level. This also proves the convergence and effectiveness of the log-linear simplification. The CRF training finds a model with better parameters for input image using CBIR than the method

Table 4.1: The confusion matrix of 1124 images for floor area labeling using CBIR-based CRF (in bold) and CRF without CBIR (in parentheses) with 53 images for training.

	floor	other
floor	**79.34%** (52.94%)	**20.66%** (47.06%)
other	**12.71%** (10.98%)	**87.29%** (89.02)

Note: Row labels are the true classes and column labels the predicted classes.

without CBIR. Based on the results, it is reasonable to believe that the top-down information could be learned by CBIR and the new labeling model based on CBIR and CRF could have better performance for large image labeling database.

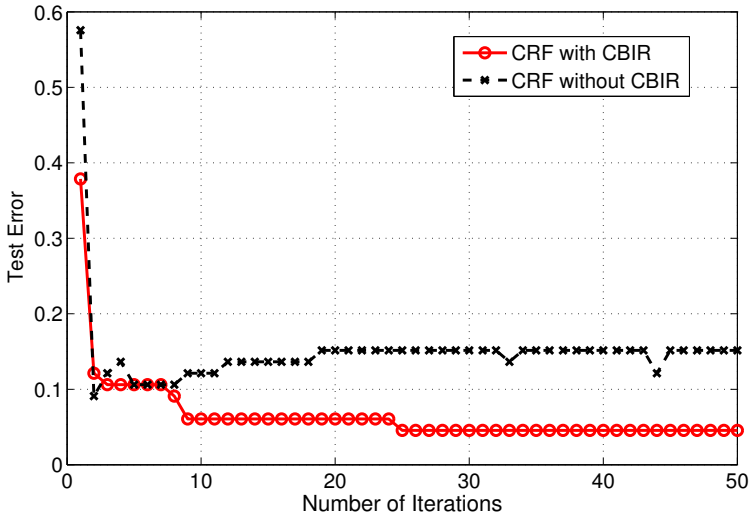


Figure 4.6: Learning curve of an example image.

Although the new CBIR part is added in the new algorithm, The computational cost of CRF with CBIR is not much higher than the one without CBIR. It is because we find a smaller training set for CRF. With smaller and similar training set, the

efficiency of CRF is increased. The computational cost could be further reduced by efficient CRF training algorithms and increased CBIR retrieval speed.

4.4 Conclusion

A new CRF image labeling framework for large labeled database which incorporates top-down information from CBIR-based retrieval was presented. The method is more suitable to deal with large labeled databases in real circumstances with a large number of ambiguous concepts. The main difference between the new approach and previous methods was that no hierarchical probabilistic model was built explicitly for an image labeling problem with specific and well controlled image database. CBIR is used to reduce top-down content ambiguities for large databases. CBIR provides a smaller content similar database for the purpose of input image labeling. Due to the content similarity of the retrieved matches, the new CRF model is better for the input image labeling task. By combining the top-down and bottom-up approach by transferring the scores from the retrieval part to the CRF model, the model is simplified to a log-linear form to reduce the computational complexity of inference and training. The stochastic gradient descent training and belief propagation inference are applied accordingly. To test the new method, a floor area labeling task was performed using the Labelme database. The new image labeling framework demonstrates better results than the one without CBIR, which proves our analysis. Though the simulation result are not complete, it can be believed that the retrieval based method is suitable to deal with content ambiguities in image labeling in large real-world databases. Future work may include finding the optimal retrieval threshold, adding more relevant features to the CRF model and combining CBIR and CRF more effectively and efficiently.

Overall, there are three new contributions of this chapter.

1. CBIR is applied to find a better and smaller training set for image labeling, and is useful for labeling using large uncontrolled image database such as Labelme.

2. The new CRF model incorporates CBIR retrieval scores as weights to strengthen global factor based on scene similarity. CBIR similarity measure is naturally integrated into the CRF model as a global weight factor which reflects the training image's similarity with the input image in the data set.

3. The new CBIR-based CRF model is simplified in a log-linear form to help reduce the training complexity.

Chapter 5

Hidden Conditional Random Field for Video Analysis

The previous two chapters discussed the image labeling problem using CRF model with new potential functions and CBIR selected training set. In content analysis, videos and images are two closely related problems. The temporal interactions of video frames are similar to spatial interactions in images. To deal with these interactions, the same principle can be applied to these two problems. This chapter extends the discussion to video content analysis. To improve the performance, a formulation is developed to solve the video problem using hidden CRF and design new potential functions following the same methodology as in image labeling.

Videos have rich structural information that can be explored for the usage in indexing and retrieval. Video content analysis finds meaningful structures and patterns from visual data for the purpose of efficient indexing and mining of videos. The primary focus of this chapter is the video event classification and its application in sports event detection. The first aspect of video content analysis is modeling the

temporal dynamic of video sequence using graph connections. The second aspect is the statistical modeling of local observations which is related to potential function selection.

The HMM is widely used for temporal interaction modeling in the literatures [100, 99, 60, 97, 14, 59, 102, 26, 67]. Unlike the HMM, the hidden CRF (HCRF) is a discriminative model that does not depend on the conditional independence assumption of observations, this makes it more suitable for video content analysis. For complex interactions in sports video frames, HCRF can be applied to model sports event and improve the classification accuracy rate compared with HMM. The HCRF has been applied to phone classification [29], gesture recognition [95] and meeting segmentation [78]. The later two papers are related to video. In [95], the gesture recognition problem is modeled using a HCRF. Relevant features are generated from gesture videos accordingly and then applied to gesture classification. In [78], multimodal features are extracted and a HMM like backward and forward algorithm is applied in HCRF to meeting event segmentation.

To address the local observation statistical modeling, the Gaussian mixture equivalent is employed in HMM and HCRF in [29] and [78]. However in sports videos the observations of features usually follow distributions other than Gaussian and Gaussian mixture, so it is more suitable to use the independent component analysis (ICA) mixture model [54] rather than the Gaussian mixture model. In [107], based on the non-Gaussian property of visual features, the ICA mixture observation model can be applied in HMM for golf video event classification.

This chapter presents a new ICA mixture HCRF (ICAMHCRF) model for sports video analysis. This new model takes advantage of discriminant power of HCRF

and the representing power of nonstructural ICA mixture model. The likelihoods of ICA mixture components are used as feature functions in the new model. The new ICAMHCRF is applied to bowling and golf event detection, and simulation results show that it has better performance than existing HMM models. It is also tested with one high activity sport hockey and simulation results show that it has comparable performance with existing HMM models.

This chapter is organized as follows. First, Section 5.1 presents a brief introduction of HCRF and formulates the video analysis problem using HCRF. Then the new HCRF model based on ICA mixture local observation is given in Section 5.2. Section 5.3 outlines general steps of the video content analysis using the ICAMHCRF model. In Section 5.4, the new ICAMHCRF model is applied to three kinds of sports (bowling, golf and ice hockey) video analysis, and numerical performance is given. Finally this chapter is concluded with summaries and future research directions in Section 5.5.

5.1 Hidden Conditional Random Field

5.1.1 Problem Formulation

The task of video content analysis is to assign the chunks of digital video data to content categories such as sports highlight, news anchor and snow mountain landscape. For a given video, the objective is to first identify the event boundaries and then classify each video segments into one of the possible known events. To simplify the problem, we assume that the beginning and ending frame of a video event are located at the shot boundaries. A video segment (one or a group of video shots) consists of a sequence of video frames which follows a chain structure. This is similar

to HMM, HCRF graphical model that can be applied to address this problem. Unlike gesture and meeting segmentation in which the backgrounds are simple, sports event detection with real scene settings is more challenging. Fortunately sports videos that consist of a set of predefined actions in a certain order fit the requirement of probabilistic graphical models.

For a video event sequence with video frames, we define each frame as a node in a graphical model. A linear chain graph is formed by linking nodes in the video playing order. A feature vector $\boldsymbol{x}_i \in \mathcal{X}$ with several features is extracted from i-th video frame in a video sequence. The feature vector is one observation of a frame in the multiple-dimensional feature space $\mathcal{X}$. A video event is defined as the meaning of one video segment (one or a group of video shots in our discussion) with several consecutive video frames. Let $y \in \mathcal{Y}$ denote a possible event where $\mathcal{Y}$ is all possible event set of a certain kind of sports video. The video event analysis task is to find the most probable y for the given observation sequence $\boldsymbol{x}$. The problem could be formulated as the conditional probability $P(y|\boldsymbol{x};\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the underlying parameter vector of the model. Here $\boldsymbol{\theta}$ instead of $\boldsymbol{w}$ is used to conform with the traditional expression in HCRF. The highest conditional probability $P(y|\boldsymbol{x};\boldsymbol{\theta})$ means that the video sequence most likely belongs to the event class y.

We formulate the video analysis problem using the hidden conditional random field (HCRF) model, an extension of conditional random field (CRF) model. The linear chain CRF model, shown in Fig. 5.1 is a commonly-used graphical model for labeling sequential data in computer vision. The structural interaction between different components of the data is reflected by a graph. The probabilistic model is build on the graph. In Fig. 5.1 the shaded circles are the observed features at nodes.

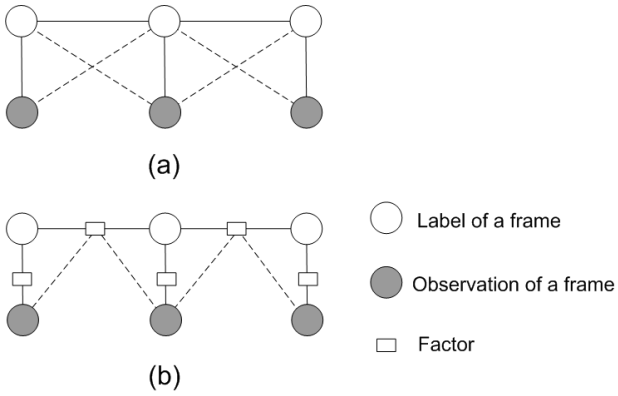


Figure 5.1: An example of 1D CRF model and its factor graph representation for video analysis.

Empty circles represent labels, which are unknown for the testing data and known for the training data. The interactions between these random variables are shown as edges. The corresponding probabilistic function of the model could be factorized to node and edge factors. Fig. 5.1(b) is the factor graph [49] representation of the model shown in Fig. 5.1(a).

Video event analysis estimates the probability $P(y|\boldsymbol{x};\boldsymbol{\theta})$ for a segment of video with a sequence of frames given the model parameter vector $\boldsymbol{\theta}$. CRF needs a label h_i for every node (frame), it prevents a CRF model from being directly applied to video content analysis. A hidden CRF model that does not require labeling for every node is applied to video analysis in the next section.

5.1.2 A New HCRF-Based Video Content Analysis Framework

In the video shot event classification, usually the states of nodes (frames) are hidden. It is a troublesome work to label all states manually. We formulate the video event

analysis task using HCRF, which is a better alternative to hidden Markov model for video event detection.

The hidden CRF (HCRF) model is first developed for object recognition [75]. Assume the observation variable $\boldsymbol{x} = \{x_i\}_{i \in V}$ has an associated labels $\boldsymbol{h} = \{h_i\}_{i \in V}$, where h_i is the label for site $i \in V$. The labeling problem is to infer the underlying labels $\boldsymbol{h}$ given the image features $\boldsymbol{x}$ and parameters of the model. In video event detection it is a troublesome work to label all states $\boldsymbol{h}$ in a sequence of video frames manually. Since we want a label for the whole sequence, we let labels of all sites be unknown hidden states. Therefore the formulation of the posterior probability $P(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{w})$ in a CRF model is replaced by $P(y|\boldsymbol{x}, \boldsymbol{\theta})$ in a hidden CRF model as in equation (2.54), which is a summation of exponentials of potential functions over all possible labels $\boldsymbol{h}$ as follows,

$$P(y|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\theta})} \sum_{\boldsymbol{h}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta})\}, \tag{5.1}$$

with potential functions

$$\Psi(y, \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i \in V} \varphi(y, h_i, \boldsymbol{x}, \boldsymbol{\theta}) + \sum_{(i,j) \in E} \psi(y, h_i, h_j, \boldsymbol{x}, \boldsymbol{\theta}).$$

Here $y \in \mathcal{Y}$ is a label for a whole sequence, and $\mathcal{Y}$ is the set of all possible labels. For example, in the binary event detection, $\mathcal{Y} = \{-1, 1\}$, where 1 represents the existence of the event and -1 nonevent. In a hidden CRF model, the observation-dependent normalization factor becomes

$$Z(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{y' \in \mathcal{Y}} \sum_{\boldsymbol{h}} \exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta})\}, \tag{5.2}$$

where y' is a possible label for the sequence.

Here a restricted form of the function $\Psi(y, \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta})$, as shown in factor graph Fig.

5.2, is chosen for video event analysis,

$$\begin{aligned}\Psi(y, \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{\theta}) &= \sum_{i \in V} [\boldsymbol{\phi}(y, \boldsymbol{x}_i)\boldsymbol{\theta}(h_i) + \delta(y, h_i)\boldsymbol{\theta}(y, h_i)] \\ &+ \sum_{(i,j) \in E} \delta(y, h_i, h_j)\boldsymbol{\theta}(y, h_i, h_j), \end{aligned} \tag{5.3}$$

where $\boldsymbol{\phi}(y, \boldsymbol{x}_i)$ is an observation function vector with the label y at site i, $\delta(y, h_i)$ equals 1 if the label y and hidden state h_i occur together else 0, and $\delta(y, h_i, h_j) = 1$ if the label y and hidden state h_i and h_j occur together else 0. Here $\boldsymbol{\theta}(h_i)$ is a parameter vector for associate potential of the hidden state h_i, $\boldsymbol{\theta}(y, h_i)$ is a compatibility parameter vector of the sequence label y and the hidden state h_i, and $\boldsymbol{\theta}(y, h_i, h_j)$ is a compatibility parameter vector of the label and the interaction edges. In equation (5.3), the first term $\boldsymbol{\phi}(y, \boldsymbol{x}_i)\boldsymbol{\theta}(h_i) + \delta(y, h_i)\boldsymbol{\theta}(y, h_i)$ is a simplification of $\varphi(y, h_i, \boldsymbol{x}; \boldsymbol{\theta})$, and the second term $\delta(y, h_i, h_j)\boldsymbol{\theta}(y, h_i, h_j)$ is an implementation of $\psi(y, h_i, h_j, \boldsymbol{x}; \boldsymbol{\theta})$. The task of the HCRF training is to learn parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}(h_i)\ \theta(y, h_i)\ \theta(y, h_i, h_j)]$, and the task of the inference is to find the label for a given input using these parameters which is also the main purpose of video content analysis.

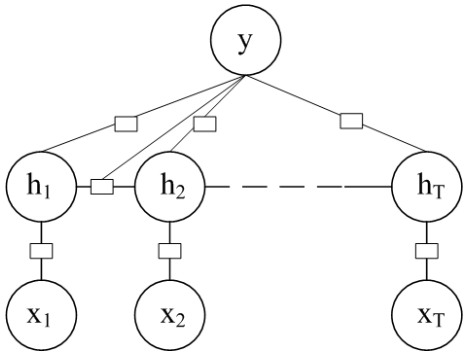


Figure 5.2: Factor graph of the new HCRF model for video analysis.

The observation feature function vector $\boldsymbol{\phi}(y, \boldsymbol{x}_i)$ is a feature statistics vector, *i.e.*,

$$\boldsymbol{\phi}(y, \boldsymbol{x}_i) = [f_1(y, \boldsymbol{x}_i), \cdots, f_{k_1}(y, \boldsymbol{x}_i), \cdots, f_{K_1}(y, \boldsymbol{x}_i)],$$

which is weighted by the parameter vector

$$\boldsymbol{\theta}(h_i) = [\theta_1, \cdots, \theta_{k_1}, \cdots, \theta_{K_1}]$$

in a HCRF model. Here K_1 is the total number and k_1 is the index of the feature function. The functions $f_{k_1}(\cdot)$ could be features themselves or functions of features. Note that we only consider the local observations $\boldsymbol{x}_i$ and the sequence label y in feature functions.

Similarly the function $\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})$ in equation (5.3) could also be written in the following feature function form,

$$\begin{aligned} \Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}) &= \sum_{i \in V} \sum_{k_1 \in \mathcal{K}_1} \theta_{k_1} f_{k_1}(y, \boldsymbol{x}_i) \\ &+ \sum_{i \in V} \sum_{k_2 \in \mathcal{K}_2} \theta_{k_2} f_{k_2}(y, h_i) \\ &+ \sum_{(i,j) \in E} \sum_{k_3 \in \mathcal{K}_3} \theta_{k_3} f_{k_3}(y, h_i, h_j), \end{aligned} \tag{5.4}$$

where $f_{k_2}(y, h_i)$ is one node feature function which represents $\delta(y, h_i)$, and $f_{k_3}(y, h_i, h_j)$ is one edge feature function which represents $\delta(y, h_i, h_j)$ for general expressions. The θ_{k_1}, θ_{k_2} and θ_{k_3} are the node and edge coefficients included in $\boldsymbol{\theta}$. The sets $\mathcal{K}_1 = \{1, \ldots, K_1\}$, $\mathcal{K}_2 = \{1, \ldots, K_2\}$ and $\mathcal{K}_3 = \{1, \ldots, K_3\}$ are sets of all possible indexes k_1, k_2 and k_3 of feature functions. This form is a linear expression so it simplifies the processe of training and reference.

5.2 ICA Mixture Hidden Conditional Random Field Model

In the traditional form, the feature vector $\boldsymbol{x}_i$ is directly used as the observation function $\boldsymbol{\phi}(y, \boldsymbol{x}_i)$ at site i in HCRF. It usually includes hundreds of features which make the learning process slow, and in addition the algorithm may not find the optimal value in a reasonable time period. In image analysis, mixture models are widely used in nonstructural classifiers. The usage of mixture models as observation functions for a HCRF model is not widely investigated except the Gaussian mixture mentioned in [29, 78]. In this chapter a new independent component analysis (ICA) mixture HCRF (ICAMHCRF) model for video event classification is developed. The feature function $f(\cdot)$ is defined as the log likelihood of the feature $\boldsymbol{x}_i$ belonging to a mixture model component. Since the log likelihood carries certain probabilistic meanings, the function could better reflect the local observation model.

5.2.1 From Gaussian Mixture to ICA Mixture for Local Observation Function

A mixture model, that commonly uses Gaussian distributions as kernel functions, is more expressive than a non-mixture. Mixture means the observation could be divided into mutual exclusive components, and obviously a mixture model can be applied to video analysis since video frames are often comprised in an interlaced manner. A mixture of Gaussian can approximate any distribution [5]. If the observation show non-Gaussian characteristics, however, more Gaussian mixture components are needed to fit the distribution. In this case, the distribution would better be decomposed into

independent mixture components.

Suppose observation $\boldsymbol{x}_i$ be expressed as a ICA mixture, *i.e.*, $\boldsymbol{x}_i \in C_k$ where C_k denotes the kth component of the mixture, $k \in \{1, 2, \ldots, K\}$. Write

$$\boldsymbol{x}_i = \boldsymbol{M}_k \boldsymbol{s}_k + \boldsymbol{\mu}_k, \tag{5.5}$$

where $\boldsymbol{M}_k$ is the mixing matrix, $\boldsymbol{s}_k$ are independent sources for kth component of mixture, and $\boldsymbol{\mu}_k$ is the bias. Then the conditional probability of seeing observation $\boldsymbol{x}_i$ given the sequence label y can be expressed as

$$\begin{aligned} P(\boldsymbol{x}_i|y) &= \sum_{k=1}^{K} P(\boldsymbol{x}_i|y, C_k) P(C_k|y) \\ &= \sum_{k=1}^{K} P(C_k|y) \exp[\log P(\boldsymbol{s}_k) - \log(|\boldsymbol{M}_k|)], \end{aligned} \tag{5.6}$$

where $|\boldsymbol{M}_k|$ denotes the determinant of the matrix $\boldsymbol{M}_k$.

5.2.2 HCRF Model with ICA Mixture Feature Function

The log likelihood of each observation belongs to a mixture component is chosen as a feature function,

$$f_{k_1}(y, \boldsymbol{x}_i) = \log P(C_k|y) P(\boldsymbol{x}_i|y, C_k), \tag{5.7}$$

$i \in V$ and $k = 1, 2, \ldots, K$ is the index of the mixture component. Here $\boldsymbol{x}_i$ is represented by conditional probability $P(\boldsymbol{x}_i|y)$ locally, and the feature functions are computed using mixture components. The number of feature functions K_1 is greater than that of mixture component K, since several groups of features such as color and texture can be included in a feature vector $\boldsymbol{x}_i$. In our experiments, only one group of feature is used so the number of feature functions K_1 is equal to the number of mixture components K.

The probability $P(C_k|y)$ in equation (5.7) is a mixture coefficient for the kth component. During a training process, the given class label y, parameters of ICA ($\boldsymbol{s}_k$, $\boldsymbol{M}_k$ and $\boldsymbol{\mu}_k$) and mixture components $P(C_k|y)$ and $P(\boldsymbol{x}_i|y, C_k)$ can be learned using a modified standard ICA algorithm presented in [54]. Major steps of this iterative algorithm are listed below,

- Compute log-likelihood of the data $\boldsymbol{x}_i$ given mixture component, as a function of current estimations of parameter $\boldsymbol{s}_k$ and $\boldsymbol{M}_k$,

$$\log P(\boldsymbol{x}_i|y, C_k) = \log P(\boldsymbol{s}_k) - \log(|\boldsymbol{M}_k|). \tag{5.8}$$

- Calculate the probability $P(C_k|y, x_i)$ with known observations $\boldsymbol{x}_i$ and the previous estimated mixture coefficient $P(C_k|y)$,

$$P(C_k|y, \boldsymbol{x}_i) = \frac{P(\boldsymbol{x}_i|y, C_k)P(C_k|y)}{\sum_{k=1}^{K} P(\boldsymbol{x}_i|y, C_k)P(C_k|y)}, \tag{5.9}$$

  and the new estimate of $P(C_k|y)$ is

$$P(C_k|y) = \frac{1}{N}\sum_{i \in V} P(C_k|y, \boldsymbol{x}_i), \tag{5.10}$$

  where N is total number of observation nodes.

- Estimate the change of the mixing matrix $\boldsymbol{M}_k$,

$$\Delta \boldsymbol{M}_k = \frac{P(\boldsymbol{x}_i|y, C_k)P(C_k|y)}{\sum_{k=1}^{K} P(\boldsymbol{x}_i|y, C_k)P(C_k|y)} \frac{\partial \log P(\boldsymbol{x}_i|y, C_k)}{\partial M_k}, \tag{5.11}$$

  and new bias

$$\boldsymbol{\mu}_k = \frac{\sum_{i \in V} P(C_k|y, \boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i \in V} P(C_k|y, \boldsymbol{x}_i)}. \tag{5.12}$$

The new ICAMHCRF model derived above provides a new way to model both the local and temporal interactions for sequence labeling tasks. ICA mixture is used in both training and testing processes. Unlike the Gaussian assumption [29, 78], a non-Gaussian model is used as local feature functions for observations. This new function

better represents complex distributions of complex features such as those in video frames. Since real scenes such as sports video consist of non-Gaussian components, which could not be represented by Gaussian mixture with only a few components [107], the proposed ICA mixture feature function is more suitable for video content analysis. When ICA mixture is combined with a HCRF model, it can adapt to statistical and temporal probabilistic structure of the data simultaneously.

5.3 ICAMHCRF based Sports Event Classification

The video event detection includes model identifications and calculating the conditional likelihood of an event. Semantic video events are represented as model parameters learned from the training video shots with known classes. This is used to train the model using video events with known labels. The parameter vector $\boldsymbol{\theta}$ is learned from the training process. After obtaining these parameters, it was possible to calculate the probability of each input video event segment belonging to a certain kind of event. In order to compute the likelihood of each event given the model $P(y|\boldsymbol{x};\boldsymbol{\theta})$. The sequence is classified as an event, whose probability produces the largest likelihood.

The new ICAMHCRF video event detection system with training and testing is shown in Fig. 5.3. In the preprocessing step, videos are divided to shots using shot boundary detection technique [106]. During training, features of frames are extracted, then ICA-based feature dimension reduction is used to reduce the computational complexity. Then compact features are modeled as a ICA mixture using the ICA

algorithm in [54] and with the assumption of Laplacian source, and parameters of the ICA mixture are learned. The log likelihood of each feature or feature group belonging to a mixture component is then calculated as the feature function in the HCRF model. Parameters of the HCRF model are learned to maximize the following likelihood objective function,

$$\mathcal{L}(\mathcal{T};\boldsymbol{\theta}) = \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \mathcal{L}(\boldsymbol{\theta}|y,\boldsymbol{x}) - \frac{\|\boldsymbol{\theta}\|^2}{2\delta^2}, \tag{5.13}$$

where

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}|y,\boldsymbol{x}) &= \log P(y|\boldsymbol{x};\boldsymbol{\theta}) \\ &= \log \frac{\sum_{\boldsymbol{h}\in\mathcal{H}} \exp\{\Psi(y,\boldsymbol{h},\boldsymbol{x};\boldsymbol{\theta})\}}{Z(\boldsymbol{x};\boldsymbol{\theta})}. \end{aligned} \tag{5.14}$$

Here $\mathcal{T}$ is the training data set, $\|\boldsymbol{\theta}\|^2$ denotes the square of the 2-norm of $\boldsymbol{\theta}$, *i.e.*, $\|\boldsymbol{\theta}\|^2 = \sum_{d=1}^{K_1+K_2+K_3} \theta_d$, and δ is the standard deviation of parameters $\boldsymbol{\theta}$. The objective function $\mathcal{L}(\mathcal{T};\boldsymbol{\theta})$ is the summation of log-likelihood of all training data minus a regulation factor. The term $\mathcal{L}(\boldsymbol{\theta}|y,\boldsymbol{x})$ is the log-likelihood of one training data belonging to the model with parameter $\boldsymbol{\theta}$. The second term in equation (5.13) is a regulation factor when parameters are assumed to be Gaussian distributed with variance δ^2.

The gradient descent method [75, 95] is used for training. The optimal estimation of the parameter $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{T};\boldsymbol{\theta}). \tag{5.15}$$

The details of training process are presented in Appendix C.

During testing, the compact features are computed and log-likelihood feature functions are calculated using the parameters from ICA mixture learned during training. Therefore using belief propagation method [104], the most probable class label y^* of

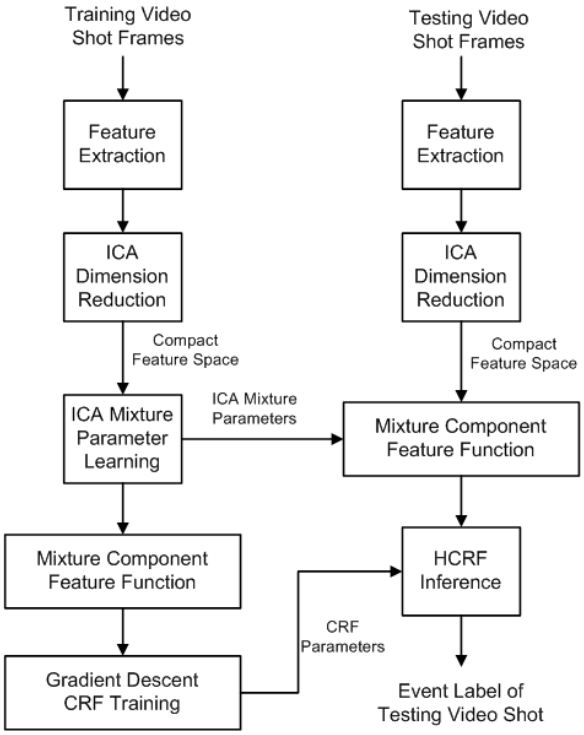


Figure 5.3: The flowchart of ICAMHCRF model for video event classification.

the testing sequence is

$$y^* = \arg\max_{y \in \mathcal{Y}} P(y|\boldsymbol{x}; \boldsymbol{\theta}^*). \tag{5.16}$$

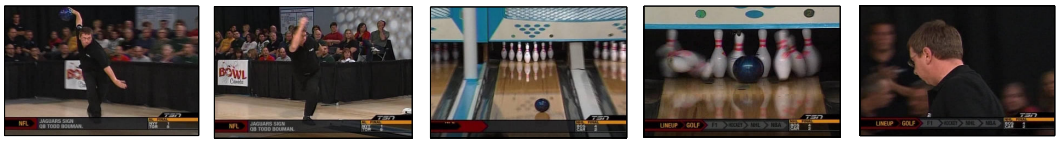
5.4 Simulation Results

The new ICAMHCRF system is applied to two types of video content analysis tasks. One is low activity sports including bowling activity recognition and golf video event analysis. The other is one high activity sport, ice hockey. In both cases, ICAMHCRF, Gaussian Mixture HCRF (GMHCRF), ICAMHMM and Gaussian mixture HMM (GMHMM) are compared. Note choosing the number of mixture components and

that of hidden states of HMM and HCRF are both non-trivial. However both of them could be optimized using training or validation set. In our experiment, the numbers are initially chosen in the range from 2 to 4 and the numbers, which maximize the classification accuracy rate in these methods, are selected. Because of the existence of hidden states, the optimization is no longer convex. So best result with random parameter initialization is chosen.

Sports videos are first segmented to shots before shot event classification. The ICA dimension reduction is applied to 256 illumination-invariant color histogram of frame features to reduce the feature vector to 2 dimension for each frame, which is also used in the event classification as original features. The cuts and gradual transitions detection is performed on this ICA subspace using an iterative clustering algorithm based on adaptive thresholding as in [106].

5.4.1 Bowling Activity Detection



(a) Event 1: a bowling shot event

(b) Event 2: other events

Figure 5.4: An example selected frames of bowling events. There are two events, (a) bowling shot event and (b) other event (an advertisement event is shown).

An ICAMHCRF model is used to recognize the bowling shot event and compared

with GMHCRF, ICAMHMM and GMHMM. A 30 minutes professional bowling TV program is used in the experiment. The video is divided to 232 video shots, in which there are 65 shots containing bowling shot events. Other irrelevant shots are comments, commercials, player's preparation and players after the shot.

An example bowling shot event sequence is shown in Fig. 5.4(a). The event consists of the following activities: bowler preparing to release his ball toward the pins, bowler dropping the ball on the lane, the ball striking the pins and finally the camera turning back to the player.

ICA mixture parameters and HCRF parameters are learned from one training shot of both events (bowling and irrelevant). Two hidden states and two mixture components are used in the experiment. Mixture components of an bowling shot event and irrelevant event are shown in Fig. 5.5 (a) and (b), respectively. The mixture components shown in Fig. 5.5 provide the possible feature distribution of these two categories of events in the ICA subspace.

Fig. 5.6 shows the receiver operating characteristic (ROC) curves of final event detection using Gaussian mixture HMM (blue dashed curve), ICA mixture HMM (red dotted curve), Gaussian mixture HCRF (green dash-dotted curve) and ICA mixture HCRF (black solid curve). The ROC curves plot the false positive rate versus the true positive rate. It can be observed from Fig. 5.6 that the ROC plot of the ICA mixture HCRF is closest to the upper left corner than that of the Gaussian mixture HCRF, Gaussian mixture HMM and ICA mixture HMM, so the ICA mixture HCRF model has a highest overall accuracy rate of detection than the other three models. The AUCs (Area under ROC curve) of those four methods are: ICA mixture HCRF, 86.04%, ICA mixture HMM, 82.95%, Gaussian mixture HCRF, 81.15%, and Gaussian

mixture HMM, 78.29%.

The confusion matrices of these two events, using the ICA mixture HCRF (shown in bold) and ICA mixture HMM (shown in parentheses), are given in Table 5.1. The rows are the true classes and the columns are detected classes.

Table 5.1: The confusion matrix for bowling event classification using ICA mixture HCRF (in bold) and ICA mixture HMM (in parentheses).

	bowling shot	other event
bowling shot	**49** (36)	**16** (29)
other event	**18** (21)	**149** (146)

Note: Row labels are the true classes and column labels the predicted classes.

The accuracy rate is defined as the ratio between the correctly labeled event and the total number of events. The detection accuracy rate using an ICAMHCRF model is given in Table 5.2. The performance of ICAMHCRF is about 7.4% better than the GMHCRF, 6.8% better than the ICAMHMM and 9.4% better than the GMHMM.

Table 5.2: Classification accuracy rate of bowling event classification.

Method	Accuracy
ICA mixture HCRF	**85.28%**
Gaussian mixture HCRF	77.92%
ICA mixture HMM	78.45%
Gaussian mixture HMM	75.86%

5.4.2 Golf Event Classification

In the process of golf video event detection, an hour long professional golf video from the authors of [107] is used. The procedure was identical to the bowling except one event consists of three shots, for fair comparison with ICAMHMM and better representation of golf event. Three example events are shown in Fig. 5.7. These three events are used for training the model, the total number of events was 202. These events where manually annotated to three categories, full swing, non-full swing and other irrelevant events. The event where very recognizable with recurrent patterns as in Fig. 5.7. The golf shot includes activities: Player prepares for the shot, followed by a player hitting the ball, then the camera follows the ball quickly. The final scene features the golf court and/or players with low activity.

The mixture components of a full swing shot, non full swing shot, and other event are displayed in Fig. 5.8 (a), (b) and (c), respectively. The mixture components shown in Fig. 5.8 provide the possible feature distribution these three categories of events in the ICA subspace.

The confusion matrices of these three events, using ICA mixture HCRF (shown in bold) and ICA mixture HMM (shown in parentheses), are shown in Table 5.3. The ICA mixture HCRF is better than ICA mixture HMM in both full swing and non-full swing classification. However, the performance of ICA mixture HCRF is not better than ICA mixture HMM for other irrelevant events, because only one training sample from the other event may not be representative for the other event class. The reasoning is that only one training sample is used here is to provide a fair comparison with results of ICA mixture HMM presented in paper [107]. Detection performance of golf event classification might be improved when more training samples are used.

Table 5.3: The confusion matrix for golf event classification using ICA mixture HCRF (in bold) and ICA mixture HMM (in parentheses).

	full swing	non full swing	others
full swing	**33** (26)	**20** (27)	**1** (1)
non full swing	**16** (23)	**109** (104)	**7** (5)
others	**2** (2)	**8** (1)	**6** (13)

Note: Row labels are the true classes and column labels the predicted classes. The ICAMHMM results shown in parentheses for comparison are cited from paper [107].

The overall accuracy rate of ICAMHCRF is 2.5% better than ICAMHMM as shown in the third row of Table 5.4.

Table 5.4: Classification accuracy rate of golf event classification.

Method	Accuracy
ICA mixture HCRF	**73.28%**
Gaussian mixture HCRF	64.68%
ICA mixture HMM	70.79%
Gaussian mixture HMM	56.93%

5.4.3 Ice Hockey Event Classification

Ice hockey event classification was used to test the proposed model in high activity sports video. A 30 minutes professional ice hockey game is used in this study. The ice hockey video is divided to 235 video shots. Three events: ice hockey shooting, ice hockey non-shooting and other irrelevant. The ice hockey shooting is usually a sequence of frames, which features following activities: the player catching the puck,

Table 5.5: The confusion matrix for ice hockey event classification using ICA mixture HCRF (in bold) and ICA mixture HMM (in parentheses).

	shooting	non-shooting	others
shooting	**22** (22)	**3** (6)	**3** (0)
non-shooting	**15** (45)	**127** (117)	**20** (0)
others	**25** (5)	**0** (8)	**20** (32)

Note: Row labels are the true classes and column labels the predicted classes. The ICAMHMM results shown in parentheses for comparison.

the player shooting the puck toward the net, the goaltender trying to catch the puck, the camera focusing on the goaltender or goal net and then the global situation of the arena. The ice hockey non-shooting includes other activities in a hockey play. Irrelevant event sequences include commercial advertisements, the scene of audience, etc. There are 28 hockey shooting events, 162 hockey non-shooting events and 45 irrelevant events in the selected video.

The classification confusion matrices of these three events, using ICA mixture HCRF (shown in bold) and ICA mixture HMM (shown in parentheses), are shown in Table 5.5. The ICA mixture HCRF is better than ICA mixture HMM in non-shooting event classification. And it is the same as ICA mixture HMM in shooting event classification. However, the performance of ICA mixture HCRF is not better than ICA mixture HMM for other irrelevant events. One possible reason is only one training sample from the irrelevant event may not be representative for the irrelevant event class.

Although for high activity video the overall accuracy of ICAMHCRF is a little lower than the accuracy with ICAMHMM as shown in Fig. 5.6, the performance of the

new model is still comparable with ICAMHMM model. That is mainly because the quick activity of ice hockey is reflected by interactions of hidden states. The complex hidden states and label interaction does not play the main role. The ICA mixture HMM captures the main factors in hockey events. To improve the performance of ICAMHCRF model for high activity sports, we need to include more features or change the graph model structure.

Table 5.6: Classification accuracy rate of ice hockey event classification.

Method	Accuracy
ICA mixture HCRF	**71.91%**
Gaussian mixture HCRF	58.72%
ICA mixture HMM	72.77%
Gaussian mixture HMM	59.15%

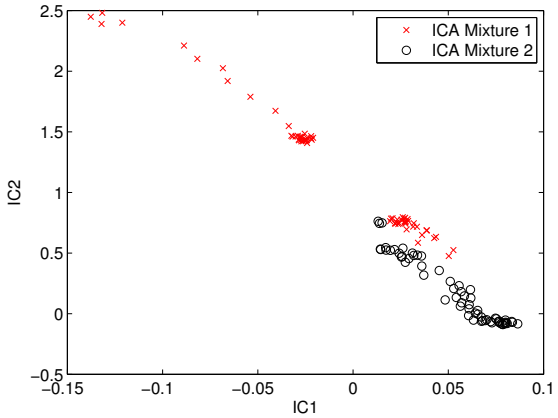
5.4.4 Discussion

In golf and bowling event classification cases, the new ICAMHCRF exhibits higher classification accuracy rate than HMM models, this is due to two main factors. First, the ICA mixture can approach the non-Gaussian distribution of compacted features of video frames. As shown in Fig. 5.5 and Fig. 5.8, a strong non-Gaussian character of compacted video features is observed. Second, comparing with HMM the relaxed assumption of HCRF model is more effective with limited training data. The feature distribution is characterized by ICA mixture and the chain temporal information is captured by HCRF. The new ICAMHCRF combines the good properties of the two and shows good performance in two low activity sports vent detection tasks over existing HMM models. It also shows comparable results for high activity sports video

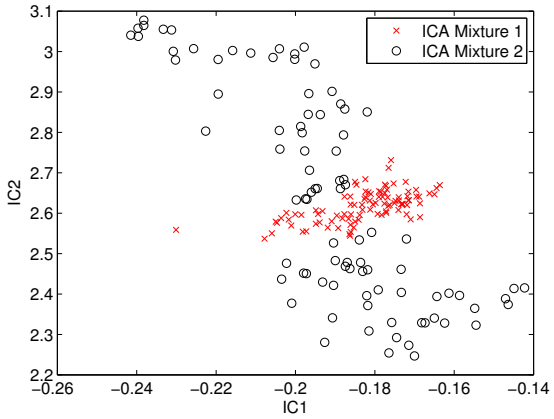
event analysis. In term of computational cost, the new HCRF framework is 2 times slower than the traditional HMM in my implementation without optimization. With better approximate training and inference algorithms, the computational efficiency is expected to be increased for the new HCRF model.

5.5 Conclusion

A new HCRF model is formulated for sports event classification in this chapter. With non-Gaussian property, the local observations of each event category are modeled as ICA mixtures. By introducing a new kind of feature function we successfully combine ICA mixture with HCRF. It is proved by experiments with bowling and golf event classification that the new model has better discriminant power than other HMM-based methods for low activity videos. The results also demonstrate the advantage of using ICA mixture over Gaussian mixture for non-Gaussian features. Future work may include extending the method to multi-modality and other kinds of features, adding links between current observation and other hidden states and investigating new model structures for high activity sports.



(a) bowling shot event



(b) irrelevant event

Figure 5.5: Two ICA mixture components for bowling shot event and irrelevant event. Axes (IC1 and IC2) are two ICA features of compacted feature space.

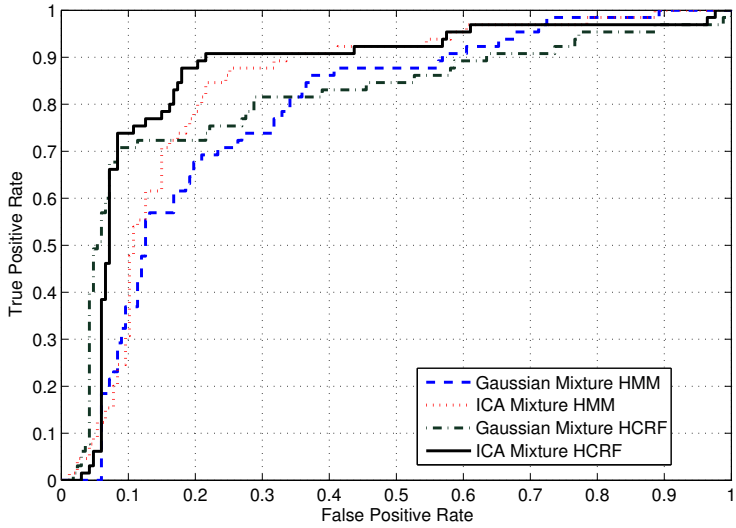


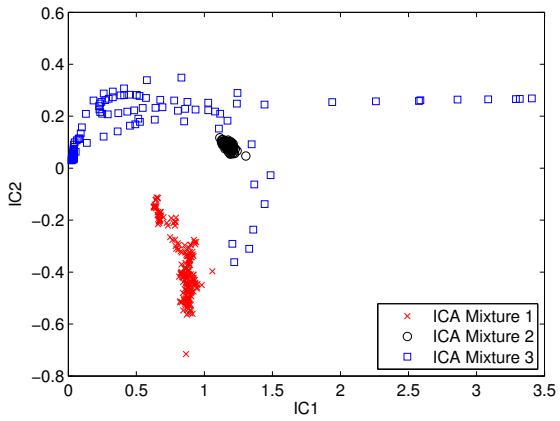
Figure 5.6: ROC performance of bowling shot classification.

(a) Event 1: a full swing shot

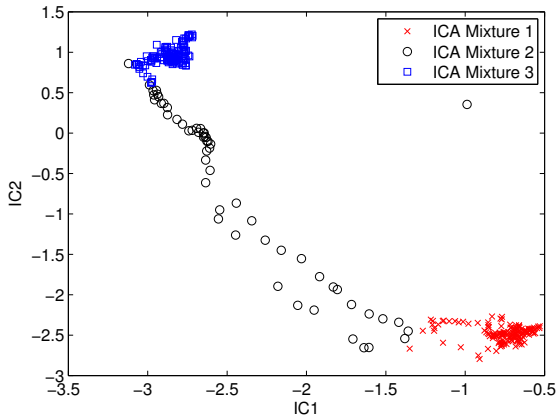
(b) Event 2: a non full swing shot

(c) Event 3: other event

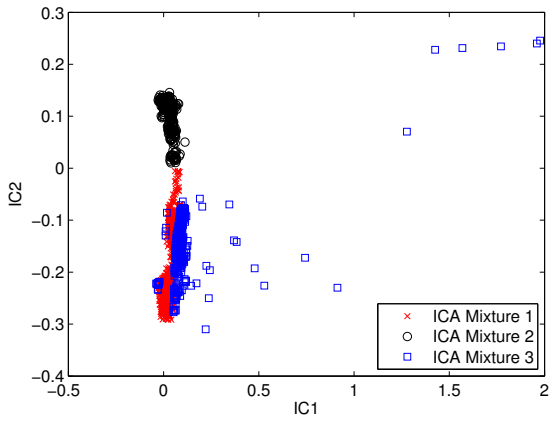
Figure 5.7: An example selected frames of golf events. There are three events, (a) full swing, (b) non-full swing, (c) other event.



(a) full swing event



(b) non full swing event



(c) other event

Figure 5.8: Three ICA mixture components of a full swing event, a non full swing event and other event, in golf video. Axes (IC1 and IC2) are two ICA features of compacted feature space.

Chapter 6

Conclusion

This thesis addressed the image and video content analysis problem based on their spatial and temporal dynamics, motivated by the importance of content analysis. Multimedia signal processing research is experiencing rapid surge because of the advance of new consumer electrical devices and the Internet, the indexing and retrieval research is dominant in this area. Most systems have limited performance, this is due to that fact that they are only using a few low-level features such as color, texture, shape, and motion. The reasoning is that there has semantic gap between high-level meanings and low-level features. The direct solution of this problem is to understand the multimedia content and bridge the semantic gap.

Image and video content analysis using graphical models especially the CRF model is the main focus of this thesis. CRF models are used for their ability to encapsulate the spatial and temporal structure of the multimedia content, the graphical model is the best solution to content analysis problems. The CRF was applied to two main content analysis problems, image labeling for both specific and general uncontrolled

databases and video shot classification. By analyzing the feature distribution of image regions in image labeling for specific databases, a new mixture feature function was introduced for better representation of the local association and interaction potential function of CRF. The new mixture CRF image labeling model was tested with commonly used Corel database, which showed superior performance than the baseline CRF. Then the discussion was extended to image labeling for large labeled databases. To reduce the content ambiguity and incorporate top-down information, a novel method combined the CBIR and CRF. The new method with CBIR was tested with the Labelme database to solve the floor labeling problem in real circumstances. The results and visual effects verified our theory analysis that the new CRF with CBIR provides better labeling accuracy. Video analysis was discussed with the same methodology as in image labeling and modeled temporal interactions in videos using hidden CRF. A new HCRF model with ICA mixture feature functions was applied to golf, bowling and hockey shot analysis with promising results.

The main contributions of this thesis include new feature functions for both CRF and HCRF, the formulation of the video content analysis problem with HCRF and the combination of CBIR and CRF for image labeling with large databases. Image and video content analysis is a large area which deserves more attention for its applications and importance for advancing human knowledge. For computers to achieve the same kind of intelligence of human being, there is much work to be done. Several challenges of image labeling and video content analysis are listed as follows:

- How to reduce the computational complexity of CRF? Although the computer technique has advanced to a level we could not have expected, but still needs extensive effort to reduce the complexity of the CRF model for the purpose of

effective use. The CRF model is built based on many features and graph structures. More features mean more complexity for CRF training and inference. Complex graph structure also means higher computational cost. Selecting good features and building a reasonable simplified graph structure will reduce the computation burden and achieve better accuracy.

- Image labeling for large databases with uncontrolled environment still needs more investigation. Most current systems have limited performance because of the limitation of feature representation and complexities of the real scene. Though graphical model such as CRF could incorporate spacial interactions, it is still limited to small database and simple graph structure. For small database such as Corel, the best model could only achieve accuracy of around 80% with fully labeling of image parts. The problem of large database image labeling is extremely difficult. Our initial work combining CBIR and CRF on floor area labeling is promising. But it still needs more research effort on other databases and multiple classes.

- How to build a generic model for all sorts of videos? Every kind of sports video is different from others. Building a generic graphical model that is appropriate to all kinds is a challenging task. The problem is still laying on the better understanding of the videos.

- How to define event categories of video content for real problems? This is an application driven problem. Each problem has its own characters. We can not find a common definition of events for all of them. It is also involves other branches of science such as psychology.

- How to combine multiple features of videos for analysis? Multimedia is a combination of multiple form of content. Multi-modality needs to be investigated thoroughly for representing video content. It is also very important to understand the role of each feature and modality for combining them naturally in graphical models.

Appendix A

Formulation of Belief Propagation for Conditional Random Field

We apply belief propagation (BP) to CRF inference. The CRF formulation is as follows,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z} \prod_{(i,j)\in E} \Psi_{(i,j)}(\boldsymbol{y}, \boldsymbol{x}) \prod_{i \in V} \Psi_i(\boldsymbol{y}, \boldsymbol{x}). \tag{A.1}$$

In BP, we estimate marginal probabilities called beliefs. In BP algorithm, messages are updated until convergence, then calculate beliefs. The standard BP is the application of the sum-product rule to estimate marginals. The standard BP solution for CRF problem is,

$$b_i(\boldsymbol{y}, \boldsymbol{x}) = k\Psi_i(\boldsymbol{y}, \boldsymbol{x}) \prod_{j \in N_i} m_{ji}(\boldsymbol{y}, \boldsymbol{x}), \tag{A.2}$$

$$m_{ji}(\boldsymbol{y}, \boldsymbol{x}) = \sum_{y_j} \Psi_i(\boldsymbol{y}, \boldsymbol{x}) \Psi_{(j,i)}(\boldsymbol{y}, \boldsymbol{x}) \prod_{k \in N_j - i} m_{kj}(\boldsymbol{y}, \boldsymbol{x}). \tag{A.3}$$

Given the message-update rule m_{ji} and belief b_i, we could compute the exact marginal probability if CRF is singly connected (That means the graph of CRF has

no loop).

An example is shown in Fig. A.1 It is not difficult to show that the message-update

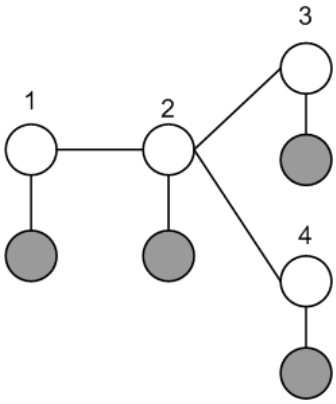


Figure A.1: Example of BP.

rule gives the exact marginal probability for node 1 as follows for the example,

$$b_1(\boldsymbol{y}, \boldsymbol{x}) = k\Psi_1(\boldsymbol{y}, \boldsymbol{x}) \sum_{y_2} \Psi_{21}(\boldsymbol{y}, \boldsymbol{x})\Psi_2(\boldsymbol{y}, \boldsymbol{x}) \sum_{y_3} \Psi_{32}(\boldsymbol{y}, \boldsymbol{x})\Psi_3(\boldsymbol{y}, \boldsymbol{x}) \sum_{y_4} \Psi_{42}(\boldsymbol{y}, \boldsymbol{x})\Psi_4(\boldsymbol{y}, \boldsymbol{x}). \tag{A.4}$$

It is also easy to convince that BP gives the exact marginal probability for all nodes in CRF without loop.

In classic point of view, for graphs with loops, the exactness of the BP breaks down. There is no restriction that forbids us to use BP for graph with loops. There are some circumstances the algorithm fails to converge. But it has been used in many research areas successfully. Because it is equivalent to an approximation of Bethe free energy of statistical physics [104]. When there is no loop, BP gives exact solution the same as dynamic algorithm such as Viterbi. For graph with loops, BP provides an approximate (but usually good) solution.

Appendix B

The EM Algorithm for Laplacian Mixture

The problem of parameter learning in Laplacian mixture model is to estimate model parameters μ_m and b_m as well as the prior probability a_m. Denote $\boldsymbol{\theta} = \{\boldsymbol{\theta}_m\}_{m=1}^{M}$ and M is the number of features, where $\boldsymbol{\theta}_m = (\mu_m, b_m, a_m)$, the problem is to find the optimal $\boldsymbol{\theta}$ that maximizes the likelihood $\mathcal{L}(\boldsymbol{\theta}) = \prod_{n=1}^{N} p(x^{(n)}; \boldsymbol{\theta})$ where N is the number of training data. When the parameters $\boldsymbol{\theta}$ known, it is most likely the data could be generated by this model. Since there has no close-form mathematical solution for the problem of this likelihood maximization of likelihood, the EM algorithm is applied in this case.

In the EM algorithm, instead of trying to maximize the likelihood $\mathcal{L}(\boldsymbol{\theta})$, one maximizes the likelihood of the joint distribution $\mathcal{L}_c(\boldsymbol{\theta}) = \log \prod_{n=1}^{N} p(x^{(n)}, z^{(n)}; \boldsymbol{\theta})$,

which can be written as

$$\begin{aligned}
\mathcal{L}_c(\boldsymbol{\theta}) &= \log \prod_{n=1}^{N} p(x^{(n)}, z^{(n)}; \boldsymbol{\theta}) && \text{(B.1)} \\
&= \log \prod_{n=1}^{N} \prod_{m=1}^{M} [p(x^{(n)}|z_m^{(n)} = 1; \boldsymbol{\theta}_m) p(z_m^{(n)} = 1)]^{z_m^{(n)}} \\
&= \sum_{n=1}^{N} \sum_{m=1}^{M} z_m^{(n)} \log p(x^{(n)}|z_m^{(n)} = 1; \boldsymbol{\theta}_m) \\
&\quad + \sum_{n=1}^{N} \sum_{m=1}^{M} z_m^{(n)} \log a_m && \text{(B.2)}
\end{aligned}$$

Taking the expectation in term of z, one can get

$$\begin{aligned}
\langle \mathcal{L}_c(\boldsymbol{\theta}) \rangle &= \sum_{n=1}^{N} \sum_{m=1}^{M} \langle z_m^{(n)} \rangle \log p(x^{(n)}|z_m^{(n)} = 1; \boldsymbol{\theta}_m) \\
&\quad + \sum_{n=1}^{N} \sum_{m=1}^{M} \langle z_m^{(n)} \rangle \log a_m, && \text{(B.3)}
\end{aligned}$$

in the E-Step, where the probability

$$p(x^{(n)}|z_m^{(n)} = 1; \boldsymbol{\theta}_m) = \frac{1}{2b_m} e^{-\frac{|x^{(n)} - \mu_m|}{b_m}}. \qquad \text{(B.4)}$$

In the M-Step, parameters $\boldsymbol{\theta}$ maximize the expectation of complete log-likelihood $\langle \mathcal{L}_c(\boldsymbol{\theta}) \rangle$ as defined in equation (B.3) are optimal solutions for the Laplacian mixture model.

B.0.1 Learning Parameter μ_m

Take a partial derivative of $\mathcal{L}_c(\boldsymbol{\theta})$ with respect to μ_m and let it equal to zero, one can have

$$\begin{aligned} 0 &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial \log p(x^{(n)}|z_m^{(n)}=1;\boldsymbol{\theta}_m)}{\partial \mu_m} \\ &= \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial(-\frac{|x^{(n)}-\mu_m|}{b_m}-\log 2b_m)}{\partial \mu_m} \\ &= \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial(|x^{(n)}-\mu_m|)}{\partial \mu_m}. \end{aligned} \tag{B.5}$$

A close-form solution does not exist. Replacing $|x^{(n)}-\mu_m|$ by $\frac{(x^{(n)}-\mu_m)^2}{|x^{(n)}-\hat{\mu}_m|}$, where $\hat{\mu}_m$ is the estimated value of μ_m at previous iteration, then one has

$$\begin{aligned} 0 &= \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial(\frac{(x^{(n)}-\mu_m)^2}{|x^{(n)}-\hat{\mu}_m|})}{\partial \mu_m} \\ &= \sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{(x^{(n)}-\mu_m)}{|x^{(n)}-\hat{\mu}_m|}. \end{aligned} \tag{B.6}$$

From equation (B.6), one has

$$\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{x^{(n)}}{|x^{(n)}-\hat{\mu}_m|} = \sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{\mu_m}{|x^{(n)}-\hat{\mu}_m|}. \tag{B.7}$$

The estimation of μ_m after the lth iteration is

$$\hat{\mu}_m^{(l)} = \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{x^{(n)}}{|x^{(n)}-\hat{\mu}_m^{(l-1)}|}}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \frac{1}{|x^{(n)}-\hat{\mu}_m^{(l-1)}|}}, \tag{B.8}$$

where $\hat{\mu}_m^{(l-1)}$ is the estimation of μ_m after the $(l-1)$th iteration.

B.0.2 Learning Parameter b_m

By setting the partial derivative of $\mathcal{L}_c(\boldsymbol{\theta})$ with respect to the parameter b_m, one can have

$$0 = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial \log p(x^{(n)}|z_m^{(n)}=1;\boldsymbol{\theta}_m)}{\partial b_m}. \tag{B.9}$$

Substituting the probability $p(x^{(n)}|z_m^{(n)}=1;\boldsymbol{\theta}_m)$ in equation (B.4) into the equation (B.9), one has

$$\begin{aligned} 0 &= \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle \partial(-\frac{|x^{(n)}-\mu_m|}{b_m} - \log 2b_m)}{\partial b_m} \\ &= \sum_{n=1}^{N}\langle z_m^{(n)}\rangle(\frac{|x^{(n)}-\mu_m|}{{b_m}^2} - \frac{1}{b_m}), \end{aligned}$$

then the estimation of b_m is

$$\hat{b}_m = \frac{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle |x^{(n)}-\hat{\mu}_m|}{\sum_{n=1}^{N}\langle z_m^{(n)}\rangle}, \tag{B.10}$$

where $\hat{\mu}_m$ is the estimation value of μ_m in the current iteration.

B.0.3 Learning Parameter a_m

The optimal parameter a_m is the solution of the following constrained maximization problem,

$$\begin{aligned} \max_{a_m} \quad & \langle \mathcal{L}_c(\boldsymbol{\theta})\rangle \\ s.t. \quad & \sum_{m=1}^{M} a_m = 1. \end{aligned} \tag{B.11}$$

The constraint is added to ensure that the summation of prior probability equals 1. Denote a Lagrange multiplier as λ, the Lagrangian function is

$$\mathcal{L}'_c(\boldsymbol{\theta}) = \langle \mathcal{L}_c(\theta)\rangle - \lambda(\sum_{m=1}^{M} a_m - 1). \tag{B.12}$$

Calculate the partial derivative of $\mathcal{L}_c'(\boldsymbol{\theta})$ with respect to a_m and let it equal to zero, one can have

$$\begin{aligned} 0 &= \frac{1}{a_m}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle - \lambda \\ &= \sum_{n=1}^{N}\langle z_m^{(n)}\rangle - \lambda a_m, \end{aligned} \tag{B.13}$$

$m = 1, \ldots, M$. Summing over all M possible mixtures, one has

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle - \lambda\sum_{m=1}^{M} a_m = 0,$$

then

$$\begin{aligned} \lambda &= \sum_m\sum_n \langle z_m^{(n)}\rangle \\ &= N. \end{aligned}$$

Therefore, the optimal parameter a_m is

$$a_m = \frac{1}{N}\sum_{n=1}^{N}\langle z_m^{(n)}\rangle. \tag{B.14}$$

To estimate parameters μ_m, b_m and a_m, one needs to calculate the expectations $\langle z_m^{(n)}\rangle$, which is

$$\begin{aligned} \langle z_m^{(n)}\rangle &= p(z_m^{(n)} = 1|x^{(n)};\boldsymbol{\theta}) \\ &= \frac{p(x^{(n)}|z_m^{(n)} = 1;\theta_m)a_m}{\sum_j^M p(x^{(n)}|z_j^{(n)} = 1;\theta_j)a_j}. \end{aligned} \tag{B.15}$$

$m = 1, \ldots, M$ and $n = 1, \ldots, N$.

Appendix C

HCRF Training

The training of HCRF model could be done the same as the ordinary CRF model except the summation of hidden variables.

$$\begin{aligned}
\mathcal{L}(\mathcal{T}, \boldsymbol{\theta}) &= \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \mathcal{L}(\boldsymbol{\theta}|y,\boldsymbol{x}) - \frac{\|\boldsymbol{\theta}\|^2}{2\delta^2} \\
&= \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \log P(y|\boldsymbol{x};\boldsymbol{\theta}) - \frac{\|\boldsymbol{\theta}\|^2}{2\delta^2} \\
&= \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \log \sum_{\boldsymbol{h}\in\mathcal{H}} P(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta}) - \frac{\|\boldsymbol{\theta}\|^2}{2\delta^2} \\
&= \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \log \frac{\sum_{\boldsymbol{h}\in\mathcal{H}} \exp\{\Psi(y,\boldsymbol{h},\boldsymbol{x};\boldsymbol{\theta})\}}{Z(\boldsymbol{x};\boldsymbol{\theta})} - \sum_{k=1}^{K} \frac{\theta_k^2}{2\delta^2},
\end{aligned}$$

where we suppose there are K parameters in penalty term $-\sum_{k=1}^{K} \frac{\theta_k^2}{2\delta^2}$ added to avoid overfitting. It includes all parameters in $\boldsymbol{\theta}$. There are three components of

$\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})$, two node terms and one edge term, as follows,

$$\begin{aligned}\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}) &= \sum_{i \in V} \sum_{k_1 \in \mathcal{K}_1} \theta_{k_1} f_{k_1}(y, \boldsymbol{x}_i) \\ &+ \sum_{i \in V} \sum_{k_2 \in \mathcal{K}_2} \theta_{k_2} f_{k_2}(y, h_i) \\ &+ \sum_{(i,j) \in E} \sum_{k_3 \in \mathcal{K}_3} \theta_{k_3} f_{k_3}(y, h_i, h_j), \end{aligned} \tag{C.1}$$

Note that this function is a general form and it is formulated in this way for simplicity.

To estimate parameters $\boldsymbol{\theta}$, one can take a partial derivative of $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, y)$ with respect to each parameter θ_k, $k = 1, \ldots, K_1 + K_2 + K_3$. For the parameter θ_{k_1} only appearing in the node term of $\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})$, the partial derivative is

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{\theta}|y, \boldsymbol{x})}{\partial \theta_{k_1}} &= \frac{\partial \log \frac{\sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{Z(\boldsymbol{x}; \boldsymbol{\theta})}}{\partial \theta_{k_1}} \\ &= \frac{\partial \log \frac{\sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\sum_{y' \in \mathcal{Y}} \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}}{\partial \theta_{k_1}} \\ &= \frac{\partial \log \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\partial \theta_{k_1}} \\ &- \frac{\partial \log \sum_{y' \in \mathcal{Y}} \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\partial \theta_{k_1}}. \end{aligned}$$

The first term of the above partial derivative is

$$\begin{aligned}&\frac{\partial \log \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\partial \theta_{k_1}} \\ &= \sum_{\boldsymbol{h} \in \mathcal{H}} \left\{ \frac{\exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}} \frac{\partial \Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})}{\partial \theta_{k_1}} \right\} \\ &= \sum_{\boldsymbol{h} \in \mathcal{H}} \left\{ P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta}) \sum_{i \in V} f_{k_1}(y, \boldsymbol{x}_i) \right\} \\ &= \sum_{s \in \mathcal{H}} \sum_{i \in V} P(y, h_i = s|\boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_1}(y, \boldsymbol{x}_i). \end{aligned} \tag{C.2}$$

Here $s \in \mathcal{H}$ is a hidden state and the $\sum_{s \in \mathcal{H}}$ is the summation of all possible states of h_i at site i, $i \in V$.

Similarly the second term of the above partial derivative is

$$\begin{aligned}
&\frac{\partial \log \sum_{y' \in \dagger} \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\partial \theta_{k_1}} \\
= & \sum_{y' \in \mathcal{H}} \sum_{\boldsymbol{h} \in \mathcal{H}} \left\{ \frac{\exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}}{\sum_{y' \in \mathcal{Y}} \sum_{\boldsymbol{h} \in \mathcal{H}} \exp\{\Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})\}} \frac{\partial \Psi(y', \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})}{\partial \theta_{k_1}} \right\} \\
= & \sum_{y' \in \mathcal{Y}} \sum_{\boldsymbol{h} \in \mathcal{H}} \left\{ P(\boldsymbol{h} | y', \boldsymbol{x}; \boldsymbol{\theta}) \sum_{i \in V} f_{k_1}(y', \boldsymbol{x}_i) \right\} \\
= & \sum_{y' \in \mathcal{Y}} \sum_{s \in \mathcal{H}} \sum_{i \in V} P(h_i = s | y', \boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_1}(y', \boldsymbol{x}_i). \qquad \text{(C.3)}
\end{aligned}$$

Therefore, the partial derivative of $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, y)$ with respect to θ_{k_1} is

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, y)}{\partial \theta_{k_1}} \\
= & \sum_{s \in \mathcal{H}} \sum_{i \in V} P(y, h_i = s | \boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_1}(y, \boldsymbol{x}_i) \\
- & \sum_{y' \in \mathcal{Y}} \sum_{s \in \mathcal{H}} \sum_{i \in V} P(h_i = s | y', \boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_1}(y', \boldsymbol{x}_i) \\
= & g_{k_1}(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}).
\end{aligned}$$

Similarly, for the parameter θ_{k_2} only appearing in the node term of $\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})$, the partial derivative is

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, y)}{\partial \theta_{k_2}} \\
= & \sum_{s \in \mathcal{H}} \sum_{i \in V} P(y, h_i = s | \boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_2}(y, h_i = s) \\
- & \sum_{y' \in \mathcal{Y}} \sum_{s \in \mathcal{H}} \sum_{i \in V} P(h_i = s | y', \boldsymbol{x}; \boldsymbol{\theta}) \cdot f_{k_2}(y', h_i = s) \\
= & g_{k_2}(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}).
\end{aligned}$$

Similarly, for the parameter θ_{k_3} only appearing in the edge term of $\Psi(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta})$,

the partial derivative is

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, y)}{\partial \theta_{k_3}} \\
= &\sum_{s\in\mathcal{H}}\sum_{s'\in\mathcal{H}}\sum_{(i,j)\in E} P(y, h_i = s, h_j = s'|\boldsymbol{x};\boldsymbol{\theta}) \\
&\cdot f_{k_3}(y, h_i = s, h_j = s') \\
- &\sum_{y'\in\mathcal{Y}}\sum_{s\in\mathcal{H}}\sum_{s'\in\mathcal{H}}\sum_{(i,j)\in E} P(h_i = s, h_j = s'|y', \boldsymbol{x};\boldsymbol{\theta}) \\
&\cdot f_{k_3}(y, h_i = s, h_j = s') \\
= &\; g_{k_3}(y, \boldsymbol{h}, \boldsymbol{x};\boldsymbol{\theta}).
\end{aligned}$$

Since belief propagation is used in this algorithm, all four probabilities $P(y, h_i = s|\boldsymbol{x};\boldsymbol{\theta})$, $P(h_i = s|y', \boldsymbol{x};\boldsymbol{\theta})$, $P(y, h_i = s, h_j = s'|\boldsymbol{x};\boldsymbol{\theta})$ and $P(h_i = s, h_j = s'|y', \boldsymbol{x};\boldsymbol{\theta})$ can be found straightforward.

Parameter are updated as follow,

$$\begin{aligned}
\theta_{k_1}^{(l)} &= \theta_{k_1}^{(l-1)} - \mu \frac{\partial \mathcal{L}(\mathcal{T}, \boldsymbol{\theta})}{\partial \theta_{k_1}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(l-1)}} \\
&= \theta_{k_1}^{(l-1)} - \mu \left\{ \sum_{(\boldsymbol{x},y)\in\mathcal{T}} g_{k_1}(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}^{(l-1)}) - \frac{\theta_{k_1}^{(l-1)}}{\delta} \right\}, \\
\theta_{k_2}^{(l)} &= \theta_{k_2}^{(l-1)} - \mu \frac{\partial \mathcal{L}(\mathcal{T}, \boldsymbol{\theta})}{\partial \theta_{k_2}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(l-1)}} \\
&= \theta_{k_2}^{(l-1)} - \mu \left\{ \sum_{(\boldsymbol{x},y)\in\mathcal{T}} g_{k_2}(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}^{(l-1)}) - \frac{\theta_{k_2}^{(l-1)}}{\delta} \right\}, \\
\theta_{k_3}^{(l)} &= \theta_{k_3}^{(l-1)} - \mu \frac{\partial \mathcal{L}(\mathcal{T}, \boldsymbol{\theta})}{\partial \theta_{k_3}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(l-1)}} \\
&= \theta_{k_3}^{(l-1)} - \mu \left\{ \sum_{(\boldsymbol{x},y)\in\mathcal{T}} g_{k_3}(y, \boldsymbol{h}, \boldsymbol{x}; \boldsymbol{\theta}^{(l-1)}) - \frac{\theta_{k_3}^{(l-1)}}{\delta} \right\},
\end{aligned}$$

where l is the index of iterations, and $\boldsymbol{\theta}^{(l)}$ is the lth estimation of the parameter vector $\boldsymbol{\theta}$.

References

[1] A brief introduction to graphical models and bayesian networks. http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html.

[2] Corel and sowerby database. http://www.cs.toronto.edu/~hexm/label.htm.

[3] Microsoft research cambridge object recognition image database. http://research.microsoft.com/en-us/projects/objectclassrecognition/.

[4] T. Amin, M. Zeytinoglu, and Guan Ling. Application of laplacian mixture model to image and video retrieval. *IEEE Trans. on Multimedia*, 9(7):1416–1429, 2007.

[5] B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

[6] M. Baillie, J.M. Jose, and C.J. van Rijsbergen. HMM model selection issues for soccer video. In *Proc. Int. Conf. Content-based image and video retrieval*, pages 70–78, 2004.

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

[8] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] E. Borenstein and S. Ullman. Combined top-down/bottom-up segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(12):2109–2125, 2008.

[10] C. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/~bouman, April 1997.

[11] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Trans. on Image Processing*, 3(2):162–177, March 1994.

[12] K.W. Bowyer and P.J. Flynn. A 20th anniversary survey: Introduction to content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(12):1348–1348, December 2000.

[13] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[14] P. Chang, M. Han, and Y. Gong. Extract highlights from baseball game video with hidden Markov models. In *Proc. Int. Conf. Image Processing*, volume 1, pages 609–612, 2002.

[15] E. Charniak. Bayesian networks without tears. *AI MAGAZINE*, 12(4):50–63, 1991.

[16] S. F. Chen and Rosenfeld R. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, 2000.

[17] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[18] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[19] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *IEEE conference on computer vision and pattern recognition*, volume 2, pages 2418–2428, New York, 2006.

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

[21] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2001.

[22] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Trans. on Image Processing*, 12(7):796–807, July 2003.

[23] Charles Elkan. Lecture notes: Log-linear models and conditional random fields. `http://www-cse.ucsd.edu/~elkan/250B/cikmtutorial.pdf`.

[24] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.

[25] X. Gibert, Huiping Li, and D. Doermann. Sports video classification using HMM. *Proc. Int. Conf. on Multimedia and Expo*, 2:345–348, 2003.

[26] Y. Gong, M. Han, W. Hua, and W. Xu. Maximum entropy model-based baseball highlight detection and classification. *Comput. Vis. Image Underst.*, 96(2):181–199, 2004.

[27] Y. Gong and W. Xu. *Machine Learning for Multimedia Content Analysis*. Springer-Verlag, 2007.

[28] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[29] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. Hidden conditional random fields for phone classification. In *Interspeech*, pages 1117–1120, 2005.

[30] F. Han and S. Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(1):59–72, 2009.

[31] A. Hanjalic. Shot-boundary detection: Unraveled and resolved. *IEEE Trans. on Circuit and System for Video Technology*, 12(2):90–104, 2002.

[32] A. Hanjalic. *Content-based Analysis of Digital Video*. Kluwer Academic Publishers, 2004.

[33] P. Harremoas and F. Topsae. Maximum entropy fundamentals. *Entropy*, 3(3):191–226, 2001.

[34] X. He. *Learning Structured Prediction Models for Image Labeling*. PhD thesis, University of Toronto, 2007.

[35] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *Proc. 9th European Conf. on Computer Vision*, volume 1, pages 338–351, 2006.

[36] X. He, R.S. Zemel, and D. Ray. *Learning and Incorporating Top-Down Cues in Image Segmentation*. Springer Berlin/Heidelberg, 2006.

[37] Xuming He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Computer Vision and Pattern Recognition*, volume 2, pages 695–702, 2004.

[38] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):219–237, October 2007.

[39] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225 – 263, 1996.

[40] J. Huang, Z. Liu, and Y. Wang. Joint video scene segmentation and classification based on hidden Markov model. In *Proc. Int. Conf. on Multimedia and Expo*, pages 1551–1554, New York, 2000.

[41] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review Online Archive (Prola)*, 106(4):620–630, May 1957.

[42] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Information Science and Statistics. Springer, 2001.

[43] G.Y. Jin, L.M. Tao, and G.Y. Xu. Hidden Markov model based events detection in soccer video. In *Proc. Int. Conf. on Image Analysis and Recognition*, pages I: 605–612, 2004.

[44] I. T. Jolliffe. *Principal Components Analysis*. Springer-Verlag, New York, 1986.

[45] E. Kijak, L. Oisel, and P. Gros. Hierarchical structure analysis of sport videos using HMMs. In *Proc. Int. Conf. Image Processing*, pages II: 1025–1028, 2003.

[46] H.-G. Kim, S. Roeber, A. Samour, and T. Sikora. Detection of goal events in soccer videos. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5682, pages 317–325, December 2004.

[47] I.Y. Kim and H.S. Yang. Efficient image labeling based on Markov random field and error backpropagation network. *Pattern Recognition*, 26(11):1695–1707, November 1993.

[48] R. Klinger and K. Tomanek. Classical probabilistic models and conditional random fields. Technical report, Department of Computer Science, Dortmund University of Technology, North Rhine-Westphalia, Germany, 2007.

[49] F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47:498–519, 1998.

[50] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *Advances in Neural Information Processing Systems 16*, 2004.

[51] S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–201, 2006.

[52] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th Int'l. Conf. Machine Learning*, pages 282–289, 2001.

[53] G. Lavee, E. Rivlin, and M. Rudzsky. Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Tran. on Systems, Man, and Cybernetics- PART C: Applications and Reviews*.

[54] T. Lee, M. S. Lewicki, and T. J. Sejnowski. ICA mixture models for unsupervised classification and automatic context switching. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(10):1078–1089, 2000.

[55] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[56] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. 9th European Conf. on Computer Vision*, volume 1, pages 581–594, 2006.

[57] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[58] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, New York, 2001.

[59] S. Liu, M. Xu, H. Yi, L. Chia, and D. Rajan. Multimodal semantic analysis and annotation for basketball video. *EURASIP J. Appl. Signal Process.*, 2006:182–182, January.

[60] Z. Liu, J. Huang, and Yao Wang. Classification of tv programs based on audio information using hidden Markov model. In *IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, pages 27–32, 1998.

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

[62] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2003.

[63] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random field. *BMC Bioinformatics*, 6, May 2005.

[64] R. McEliece and S. M. Aji. The generalized distributive law. *IEEE Trans. Inform. Theory*, 46:325–343, 2000.

[65] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[66] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *IEEE conference on computer vision and pattern recognition*, volume 2, pages 326–333, 2004.

[67] M.R. Naphade and T.S. Huang. Extracting semantics from audiovisual content: The final frontier in multimedia retrieval. *IEEE Trans. on Neural Networks*, 13(4):793–810, 2002.

[68] R. Neher and A. Srivastava. A bayesian mrf framework for labeling terrain using hyperspectral imaging. *IEEE Trans. Geoscience and Remote Sensing*, 43(6):1363–1374, June 2005.

[69] I. Nwogu and J. J. Corso. Labeling irregular graphs with belief propagation. In *international Workshop on Combinatorial Image Analysis*, volume LNCS 4958, pages 295–305, 2008.

[70] H. Park and T. Lee. Capturing nonlinear dependencies in natural images using ICA and mixture of laplacian distribution. *Neurocomput.*, 69(13-15):1513–1528, 2006.

[71] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[72] H. Permuter, J. Francos, and I.H. Jermyn. A study of gaussian mixture models of colour and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[73] V. D. Pietra and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:380–393, 1995.

[74] Y. Qi, M. Szummer, and T. P. Minka. Diagram structure recognition by bayesian conditional random fields. In *IEEE conference on computer vision and pattern recognition*, volume 2, pages 191–196, 2005.

[75] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(10):1848–1852, 2007.

[76] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[77] A. Rangarajan, R. Chellappa, and B. Manjunath. Markov random fields and neural networks with applications to early vision. In I.K.Sethi and A.K.Jain, editors, *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections*. Elsevier Science Publishers, 1991.

[78] S. Reiter, B. Schuller, and G. Rigoll. Hidden conditional random fields for meeting segmentation. In *Proc. Int. Conf. on Multimedia and Expo*, pages 639–642, 2007.

[79] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. 9th ICCV*, volume 1, pages 10–17, 2003.

[80] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer-Verlag New York, Inc, New York, NY, USA, 2004.

[81] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, NY, USA, 2007.

[82] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999.

[83] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008.

[84] A. Saxena, M. Sun, and A. Y. Ng. Make 3d: Learning 3-d scene structure from a single still image. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(5):824–840, 2008.

[85] J. Shotton. *Contour and Texture for Visual Recognition of Object Categories*. PhD thesis, University of Cambridge, 2008.

[86] P. Smyth. Belief networks, hidden markov models, and markov random fields: a unifying view. *Pattern Recognition Letters*, 18:1261–1268, 1998.

[87] I. Steinwart and A. Christmann. *Support vector machines*. Springer, 2008.

[88] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[89] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems 17*, pages 1401–1408, 2005.

[90] T. Toyoda, K. Tagami, and O. Hasegawa. Integration of top-down and bottom-up information for image labeling. In *IEEE conference on computer vision and pattern recognition*, volume 1, pages 1106–1113, 2006.

[91] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[92] S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proc. 23rd Int'l. Conf. Machine Learning*, pages 969–976, 2006.

[93] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

[94] H.M Wallach. Conditional random fields: An introduction. Technical report, University of Pennsylvania, 2004.

[95] S. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1521–1527, 2006.

[96] X. Wang, X. Zhang, I. Clarke, and Y. Yakubovich. A new room decoration assistance system based on 3d reconstruction and integrated service. In *Proc. Int. Conf. Content-based image and video retrieval*, pages 627–634, 2008.

[97] Y. Wang, Z. Liu, and J. Huang. Multimedia content analysis using both audio and visual cues. *IEEE Signal Processing Magazine*, 2000.

[98] L. R. Welch. Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4), 2003.

[99] L. Xie and S. Chang. Structure analysis of soccer video with hidden Markov models. In *Pattern Recognition Letters*, pages 767–775, 2002.

[100] L. Xie, H. Sundaram, and M. Campbell. Event mining in multimedia streams. *Proceedings of the IEEE*, 96(4):623–647, 2008.

[101] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pages 401–404, 2003.

[102] G. Xu, Y.F. Ma, H.J. Zhang, and S.Q.A. Yang. An HMM-based framework for video semantic analysis. *IEEE Trans. on Circuit and System for Video Technology*, 15(11):1422–1433, November 2005.

[103] Y. Yang, S.X. Lin, Y.D. Zhang, and S. Tang. Statistical framework for shot segmentation and classification in sports video. In *Proc. Asian Conference on Computer Vision*, pages II: 106–115, 2007.

[104] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, pages 689–695, 2001.

[105] H.J. Zhang, A. Kankanhalli, and S.W. Somaliar. Automatic partition of full-motion video. *Multimedia Systemes*, 1(1):11–28, 1993.

[106] J. Zhou and X. Zhang. Video shot boundary detection using independent component analysis. In *Proceedings. (ICASSP '05). IEEE International Conference onAcoustics, Speech, and Signal Processing*, volume 2, pages 541–544, March 2005.

[107] J. Zhou and X. Zhang. An ICA mixture hidden Markov model for video content analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1576–1586, 2008.

VITA

NAME:	Xiaofeng Wang
PLACE OF BIRTH:	Nong'an CHINA
POST-SECONDARY	Beijing University of Posts and Telcom.
DEGREES:	Beijing, CHINA
	BENG, MENG
HONORS AND AWARDS:	Graduate Research Excellence Award, 2008, Ryerson
	Graduate Research Excellence Award, 2007, Ryerson
	Excellent Student, 1997, BUPT
	Qualcomm-BUPT academic scholarship, 1997, BUPT
	Academic Scholarship, 1996, BUPT
	Academic Scholarship, 1995, BUPT
WORK EXPERIENCE:	Siemens Ltd. CHINA
	System Engineer 2002-2004
	Beijing Founder Linkair INC, CHINA
	System Engineer 1999-2002

PUBLICATIONS

1. Xiaofeng Wang and Xiao-Ping Zhang, "On optimal look-up table based data hiding," in IET Signal Processing, to appear.

2. Xiaofeng Wang and Xiao-Ping Zhang, "A new laplacian mixture conditional

random field model for image labeling," in Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, Dallas TX, Mar. 2010.

3. Xiaofeng Wang, Xiao-Ping Zhang, "Hockey shot event modeling with mixture hidden Markov model" in Proc. of ACM Multimedia 2009 Workshop on Events in Multimedia, Beijing China, Oct. 2009, pp. 25-32.

4. Xiaofeng Wang, Xiao-Ping Zhang, Ian Clarke and Yury Yakubovich, "A new gaussian mixture conditional random field model for indoor image labeling" in Proc. of ACM Multimedia 2009 Workshop on Interactive Multimedia for Consumer Electronics, Beijing China, Oct. 2009, pp. 51-56.

5. Xiaofeng Wang, Xiao-Ping Zhang, "ICA mixture hidden conditional random field model for sports event classification" in Proc. of IEEE Workshop on Video-Oriented Object and Event Classification in Conjunction with ICCV 2009, Kyoto Japan, Sep. 2009.

6. Xiaofeng Wang, Xiao-Ping Zhang, Ian Clarke and Yury Yakubovich, "A new image labeling method based on content-based image retrieval and conditional random field ," in Proc. International Symposium on Image and Signal Processing and Analysis, Salzburg Austria, Sep. 2009, pp. 221-226.

7. Xiaofeng Wang and Xiao-Ping Zhang, "A new localized superpixel Markov random field for image segmentation," in Proc. IEEE Int. Conf. Multimedia and Expo, New York NY, June 2009, pp. 642-645.

8. Xiaofeng Wang, Xiao-Ping Zhang, Ian Clarke and Yury Yakubovich, "A new room decoration assistance system based on 3D reconstruction and integrated

service," in Proc. ACM International Conference on Image and Video Retrieval, Niagara Falls ON, July 2008, pp. 627-633

9. Xiao-Ping Zhang, Kan Li, and Xiaofeng Wang, "A novel look-up table design method for data hiding with reduced distortion," IEEE Trans. Circuits Syst. Video Technol.,Volume 18, Issue 6, June 2008, pp. 769-776.

10. Xiaofeng Wang and Xiao-Ping Zhang, "A new implementation of trellis coded quantization based data hiding," in Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, Las Vegas NV, April 2008, pp. 1689-1692.

11. Xiaofeng Wang and Xiao-Ping Zhang, "Generalized trellis coded quantization for data hiding," in Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, Honolulu HI, April 2007, vol. 2, pp. 269-272.

12. Xiaofeng Wang and Xiao-Ping Zhang, "Minimum distortion look-up table based data hiding," in Proc. IEEE Int. Conf. Multimedia and Expo, Toronto ON, July 2006, pp. 1337-1340.

PATENTS

1. XuePeng Zhang and Xiaofeng Wang, "METHOD FOR MULTICODE SPREADING AND DESPREADING BY LS CODE" PCT/CN02/00436

2. Yao Zhao and Xiaofeng Wang, "DETECTING THE USER NUMBER FOR THE TDSCDMA DOWN LINK" Siemens Ltd CHINA 2003.3.10