Statistical Part-Based Models: Theory and Applications in Image Similarity, Object Detection and Region Labeling

Dong-Qing Zhang

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2005

Abstract

Statistical Part-Based Models:
Theory and Applications in Image Similarity,
Object Detection and Region Labeling

Dong-Qing Zhang

The automatic analysis and indexing of visual content in unconstrained domain are important and challenging problems for a variety of multimedia applications. Much of the prior research work deals with the problems by modeling images and videos as feature vectors, such as global histogram or block-based representation. Despite substantial research efforts on analysis and indexing algorithms based on this representation, their performance remains unsatisfactory.

This dissertation attempts to explore the problem from a different perspective through a part-based representation, where images and videos are represented as a collection of parts with their appearance and relational features. Such representation is partly motivated by the human vision research showing that the human vision system adopts similar mechanism to perceive images. Although part-based representation has been investigated for decades, most of the prior work has been focused on *ad hoc* or deterministic approaches, which require manual designs of the models and often have poor performance for real-world images or videos due to their inability to model uncertainty and noise. The main focus of this thesis instead is on incorporating statistical modeling and machine learning techniques into the paradigm of part-based modeling so as to alleviate the burden of human manual design,

achieve the robustness to content variation and noise, and maximize the performance by learning from examples.

We focus on the following three fundamental problems for visual content indexing and analysis : measuring the similarity of images, detecting objects and learning object models, and assigning semantic labels to the regions in images. We focus on a general graph-based representation for images and objects, called Attributed Relational Graph (ARG). We explore new statistical algorithms based upon this representation. Our main contributions include the following: First, we introduce a new principled similarity measure for ARGs that is able to learn the similarity from training data. We establish a theoretical framework for the similarity calculation and learning. And we have applied the developed method to detection of near-duplicate images. Second, we extend the ARG model and traditional Random Graph to a new model called Random Attributed Relational Graph (Random ARG) to represent an object model. We show how to achieve object detection through constructing Markov Random Fields, mapping parameters and performing approximations using advanced inference and learning algorithms. Third, we explore a higher-order relational model and efficient inference algorithms for the region labeling problem, using video scene text detection as a test case.

Contents

Contents **i**

List of Figures **iv**

List of Tables **vi**

Acknowledgements **vii**

1 Introduction **1**

1.1 Motivation 1

1.2 Introduction to Part-based Modeling 5

1.2.1 Part Detection in Images 6

1.2.2 Feature Extraction 7

1.2.3 Part-based Representation 8

1.3 Major Problems Addressed and Contributions 9

1.3.1 Image Similarity 10

1.3.2 Object Detection 12

1.3.3 Region Labeling 15

1.4 Overview of the Thesis 16

2 Learnable Image Similarity using Attributed Relational Graph **19**

2.1 Introduction 19

2.1.1 Related Work 20

2.1.2 Overview of the Proposed Method 24

2.2 Similarity by Transformation 25

2.2.1 Design of Transformation 29

2.3 Computing the Transformation Likelihood 30

2.3.1 Map the Parameters of Transformation to MRF 30

2.3.2 Computing the Partition Functions 35

2.3.3 Computing the Approximated Marginals 35

2.4 Learning the Transformation Parameters 38

2.5 Application to Image Near-Duplicate Detection and Retrieval 40

2.5.1 Implementation Issues 41

2.5.2 Experiments 42

2.6 Summary 47

3 Learning Random Attributed Relational Graph for Part-based Object Detection 51

3.1 Introduction 51

3.2 Literature Review and Our Contributions 52

3.3 The Random Attributed Relational Graph Model 57

3.3.1 Bayesian Classification under Random ARG Framework 58

3.3.2 Mapping the Random ARG parameters to the Association Graph MRF 60

3.3.3 Computing the Partition Functions 63

3.3.4 Learning Random Attributed Relational Graph 65

3.3.5 Spanning Tree Approximation for Spatial Relational Features 67

3.4 Extension to Multi-view Mixture Model 67

3.5 Experiments 69

3.6 Summary 73

4 Image Region Labeling with a Higher-order Statistical Relational Model 75

4.1 Introduction 75

4.2 Prior Work and Our Contributions 76

4.3 Region adjacency graph formation 78

4.4 Formulating text detection using MRF . . . 79

4.4.1 Pairwise MRF . . . 80

4.4.2 Higher-Order MRF with belief propagation . . . 81

4.5 Design of the Potential Functions . . . 84

4.5.1 The one-node potential . . . 84

4.5.2 The three-node potential . . . 85

4.6 Learning the Higher-Order MRF . . . 86

4.7 Modification of the potential functions for multiple text lines . . . 88

4.8 Experiments and results . . . 90

4.9 Summary . . . 93

5 Conclusion and Future work **94**

5.1 Summary of Contributions . . . 94

5.2 Future work . . . 96

5.2.1 Limitations and Possible Solutions . . . 96

5.2.2 Other Applications . . . 98

A Proofs of Chapter 2 **109**

A.1 Proof of Lemma 2.1 . . . 109

A.2 Proof of Lemma 2.2 . . . 110

A.3 Proof of Proposition 2.1 . . . 110

A.4 Proof of Theorem 2.1 . . . 112

B Proofs of Chapter 3 **114**

B.1 Proof of Lemma 3.1 . . . 114

B.2 Proof of Lemma 3.2 . . . 116

B.3 Proof of Theorem 3.1 . . . 117

C Higher-Order Belief Propagation **118**

C.1 Derivation of Higher-Order Belief Propagation . . . 118

List of Figures

1.1 (a) flickr.com (b) photo blog (c) cnn free news videos 2

1.2 Part-based modeling and analysis 5

1.3 Examples of part detection. (a) corner (interest point) detection by the Harris corner detector. (b) salient region detection by the Maximum Entropy Region detector (c) region segmentation by the mean-shift algorithm 7

1.4 Examples of part-based representation. Color in the ARG indicates different feature associated with the nodes 9

2.1 Attributed Relational Graph for Part-based Representation 25

2.2 The association graph upon which the MRFs are defined. 29

2.3 The generative model for ARG matching 30

2.4 Examples of Image Near-Duplicate 41

2.5 ROC curve of IND detection with different methods. (GRAPH-MC: the proposed method with Monte Carlo inference, GRAPH-LBP: the proposed method with Loopy Belief Propagation, GCH: Grid-based Color Histogram, COLOR: Color Histogram, MINE: MINimum Energy by energy minimization, LED: Local Edge Descriptor) 44

2.6 ROC curve of IND detection with different MRF node number (here, GCH is for reference) 46

2.7 The retrieval performance with different methods.(GRAPH-MC: the proposed method with Monte Carlo inference, GRAPH-LBP: the proposed method with Loopy Belief Propagation, GCH: Grid-based Color Histogram, COLOR: Color Histogram, MINE: MINimum Energy by energy minimization, LED: Local Edge Descriptor) 48

2.8 Correctly detected IND image pairs. The numbers shown below images are similarity values 49

2.9 Missed IND image pairs 49

2.10 Correctly rejected non-IND image pairs 50

2.11 Non-IND image pairs that are falsely detected as IND pairs 50

3.1 A generative process that generates the part-based representation of an image 57

3.2 ARG, Random ARG and the Association Graph. Circles in the image are detected parts 59

3.3 Spanning tree approximation, realized by first constructing a weighted graph (having the same topology as the Random ARG) with the weight $w_{ij} = |\Sigma_{ij}|$ in the edge $e_l = (n_i, n_j)$, then invoking the conventional minimum spanning tree(MST) algorithm, such as Kruskal's algorithm. Here $|\Sigma_{ij}|$ is the determinant of the covariance matrix of $f_{ij}(.)$ (the *pdf* of the relational feature in the Random ARG) associated with the edge $e_l = (n_i, n_j)$. 66

3.4 The Random ARG learned from the "motorbike" images. 70

3.5 The multi-view objects for learning and testing 72

4.1 Examples of scene text in images 77

4.2 Region segmentation and adjacency graph. 79

4.3 (Left) MRF with higher-order potential, node features, and clique-level relational features (Right) The message passing of the high-order belief propagation 82

4.4 The reinforcement of the beliefs as the number of characters increases 84

4.5 Examples of higher-order features 85

4.6 (Left) The miss in detecting multiple text lines due to cross-text-line (CTL) cliques. (Right) the results after potential function modification. 88

4.7 The potential functions and the modified version of $G_{111}(y_{ijk})$ 89

4.8 Precision recall curve, (Left) The comparison of ROC curve by using conventional pairwise MRF (dashed blue) and proposed method (red). (Right) The ROC curve of detection in set 1(green) set 2(blue) and average(red). . . 91

4.9 Example results from the higher-order MRF model (Brightness of nodes represents the probability as "text") 92

4.10 Output from the pairwise MRF model (brightness of nodes represents the probability as "text") 93

List of Tables

2.1 Average precision of IND detection by different methods 45

2.2 Average precision of IND detection with different MRF node number 46

2.3 Average ranks using different methods (The smaller the better. The perfect value is 1.0 47

3.1 Object detection performance and learning time of different methods (xi/yh means x iterations and y hours). * Background images are road images the same as [30]. 71

3.2 Multi-view object detection performance 72

Acknowledgements

First and foremost, I would like to express my deep gratitude to my advisor, Prof. Shih-Fu Chang, for his invaluable guidance and vision that direct me towards my research goals. I wouldn't forget that it was under his vision that I started the research on graphical models and statistical methods. His insistency in rigor and precise has also benefited me a lot in research styles.

I also would like to thank my thesis committee members, Prof. Tony Jebara, Prof. Paul Sajda, Prof. Dan Ellis and Prof. Xiaodong Wang for their many constructive suggestions regarding general and specific issues on machine learning and human vision system. I am fortunate to have the opportunity to discuss with them, but regretted to have too little time to extend our discussions.

I also benefit a lot from helpful interactions with other members in the DVMM group. Specifically, I would like to thank Dr. Shahram Ebadollahi for his introduction of Markov Random Fields in our paper reading group, and the discussions on part-based models. I would like to thank Dr. Lexing Xie for the discussions about machine learning and object detection. I would also like to thank other DVMM members: Dr. Qibin Sun, Dr. Ana Benitez, Dr. Alejandro Jaimes, Dr. Hari Sundaram, Tian-Tsong Ng, Eric Zavesky, Winston Hsu, Lyndon Kennedy, Barry A. Rafkind, Dr. Yong Wang, Jessie Hsu, Akira Yanagawa among others for their suggestions and friendship.

I would like to thank Dr. John R. Smith and Dr. Ching-Yung Lin in IBM T.J. Watson Research center for their supports and guidance during my internship in IBM. Special appreciation is towards Dr. Belle Tseng for her advice.

Finally, I would like to thank my parents. I would like appreciate their perpetual understanding and support of my study and work.

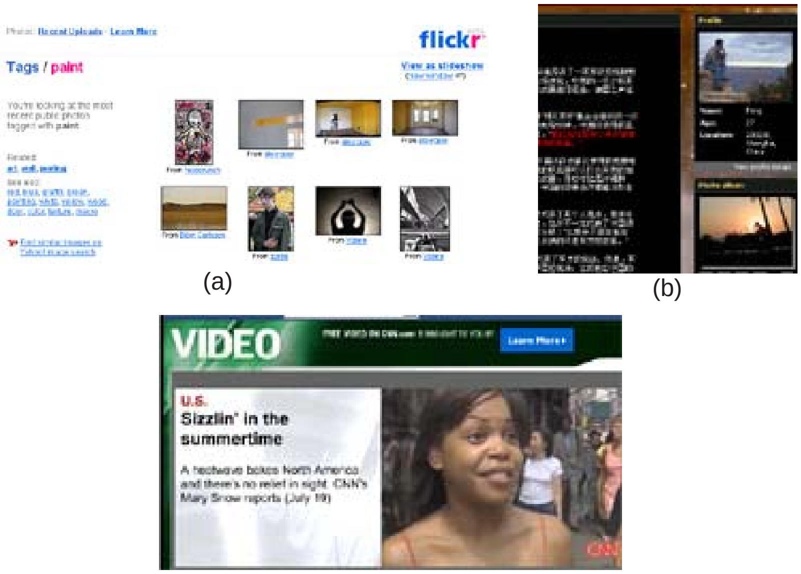
Chapter 1

Introduction

1.1 Motivation

Due to the popularity of digital cameras and camcorders, we have witnessed the dramatic increase of visual content such as photos and videos in recent years. The increased use of visual content has fostered new multimedia applications. For instance, flickr.com, a photo sharing web site, allows registered users to share their photos and tag keywords to the photos for categorization and browsing (Figure 1a). Photo blog is another popular application that enables users to publish their photos and add annotations (Figure 1b). In video domains, video streaming is becoming increasingly pervasive. Many major online media websites have established their free news video services, such as CNN.com (Figure 1c). The emergence of these applications call for the needs of managing vast amount of visual content in automatic ways.

In contrast to the significant advancement of the multimedia applications. technologies for automatic visual content analysis and indexing are still lacking. Until today, it remains difficult for a computer to understand the content in a natural image. The understanding of



A heatwave bakes North America and there's no relief in sight. CNN's Mary Snow reports (July 19)

Figure 1.1. (a) flickr.com (b) photo blog (c) cnn free news videos

visual content in an image is considered so difficult that people has coined a phrase called “semantic gap” to indicate the difficulty of translating the low-level features into high-level semantics.

Most of the traditional techniques for analyzing and indexing content are based on the classic pattern recognition theory [23]. This standard paradigm first represents an image or a segment of video as a feature vector, which is then followed by a classification machine to map the features (continuous or discrete variables) to semantic labels (symbolic variables). Behind this paradigm is the rigorous mathematical framework for automatic learning and decision. Research on automatic visual content analysis has been following this framework for many years. The advancement of the research on machine learning and pattern recognition has resulted in sophisticated statistical and learning algorithms. Indeed, we have seen that the performance of image retrieval and semantic concept detection has significantly increased due to the use of machine learning algorithms, such as Support Vector Machines (SVM), Boosting and Ensemble learning. There is no doubt that statistical and machine learning algorithm will continue to play indispensable roles in visual content analysis systems. Yet, despite the pervasive use of statistical methods with sophisticated optimization algorithms, the performance of visual content analysis and indexing by computers is far behind what can be done by their human counterparts.

Such discrepancy raises fundamental psychological and philosophical questions beyond technical designs of recognition algorithms : how does human being perceive images ? For decades, human vision researchers have been studying how human being perceives and understands images [101][8]. The psychological studies indicate that there is a dramatic difference of image understanding between human being and the standard paradigm mentioned above. It turns out that human image perception is goal-driven, active and partly sequential[81]. The process can be roughly divided into two subsystems: pre-attentive subsystems and attentive subsystems. In the pre-attentive subsystem, human eyes try to detect salient parts, such as corners, in the image. In the attentive subsystem, these detected parts are grouped into meaningful high-level entities such as objects. Feedback from the attentive subsystem could in turn assists the pre-attentive subsystem to locate the salient parts. Such mechanism motivates the use of similar approaches in computer vision or image retrieval, namely, decomposing images or objects into a collection of parts (visual primitives such as regions or corners) and exploring part attributes and relations for analysis and retrieval.

Image analysis by parts is not a new idea, and has been widely adopted by the computer vision researchers from its very beginning. For example, generalized cylinder is a part-based model proposed by Marr [67] for describing three-dimensional objects. Marr proposed that a three-dimensional object can be constituted by a set of generalized cylinders, whose parameters characterize individual parts. Similar representations for two-dimensional objects have also been developed. For instance, a well-known face recognition model called Elastic Bunch Graph [106] represents a face as a set of parts with spatial relations. Despite the success of these methods in constrained domains, such as synthetic images or photos under controlled laboratory conditions. It is difficult to apply them for real-world images or videos. The main problem is that irrelevant contents and noise are ubiquitous in real-world images. As a result, it is difficult to guarantee that parts relevant to the target object can be accurately detected and features are precise to describe the detected parts. Taking the example of the Elastic Bunch Graph model, it generally requires that the vertices of the graph be well registered to the corner points in the face. This could be achievable for the face images under good conditions, it is rather difficult to realize if faces are embedded in complex scenes, or with mustache and glasses.

On the other hand, if we look back at the pattern classification based methods, we will find that, in many circumstances, they can often achieve satisfactory results in real-world images or videos although their representations may be not optimal. One example is the SVM-based face detection system [73]. The system exhaustively slides a search window over image plane, extracts feature vectors within the windows and feeds them into SVM for classification. Although the computation cost of the system is extraordinary high, its performance has been shown to be satisfactory for face detection in natural images [73]. The main reason for this robustness is that the SVM model is able to accommodate and learn face variations from the training data, achieving high robustness to various conditions.

Based on the above analysis, a natural question is whether or not we can establish similar statistical framework for the part-based models. Namely, extend the pattern classification and machine learning methods to part-based representations. Compared with the traditional feature vector based approaches, pattern classification and machine learning on part-based representation or non-vector space models is a much less explored area. Currently, there is no formal theoretical framework for non-vector space models. Therefore combining the part-based model and statistical methods is not trivial. The non-triviality draws a clear boundary between the traditional deterministic part-based models and the new ones with statistical

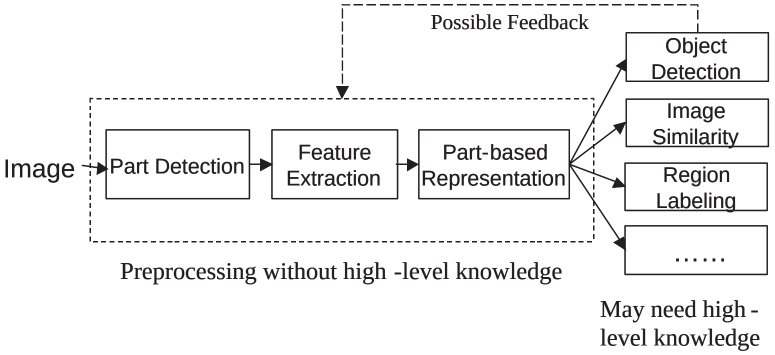


Figure 1.2. Part-based modeling and analysis

modeling. We may call the previous part-based models as classic part-based models and the new ones as statistical part-based models.

In order to make the following chapters more comprehensive, we first present a brief overview about the part-based modeling in the following section.

1.2 Introduction to Part-based Modeling

The part-based modeling procedure usually contains three major components: part detection, feature Extraction and part-based representation (Figure 1.2). Part detection locates the salient visual primitives in an image, such as corners or local image patches. Feature extraction is used to extract informative attributes for describing individual parts and inter-part relationships. Part-based representation is used to organize the detected parts and their attributes as an integrated entity, as an input to the part-based analysis subsystem, which could be similarity computation, object detection etc.

1.2.1 Part Detection in Images

Parts are elementary building blocks for part-based analysis. Here parts mean visual primitives in an image. Parts can be corners detected by corner detection algorithm, regions after color region segmentation, etc. The choice of specific part detector depends on characteristics of the applications. For example, for image similarity, our experiments have shown that corner point based representation results in better performance than region-based representation.

Corner detection is also known as interest point detection. Although there are several algorithms for corner detection, the principles behind those algorithms are very similar. The most widely used corner detection algorithm may be the Harris corner detector [39], which realizes corner detection by computing the second-order geometric features in a local image patch. Another popular corner detection algorithm is the SUSAN corner detector [92], which realize corner detection by calculating the second order spatial derivative around a pixel.

Region segmentation is intended to partition an image into a set of disjoint regions with homogenous properties. Region segmentation has been an active research topic in computer vision. Major algorithms include K-mean clustering based approach, Graph Cuts [88] and mean-shift algorithm [16]. Compared with interest point detection, the main limitation of region segmentation is its sensitivity to noise. Segmenting two images with the same visual content may end up with totally different region segmentation schemes because of noise. Furthermore, because the segmented regions are disjoint, the errors of region segmentation often cannot be corrected in the later analysis stage.

Another type of region detection algorithm, called salient region detector, yields overlapping regions rather than disjoint regions. The benefit of using overlapping region is to create an overcomplete representation of an image, so that in the analysis stage, high-level knowledge can be utilized to eliminate the irrelevant or noisy regions. A popular salient region

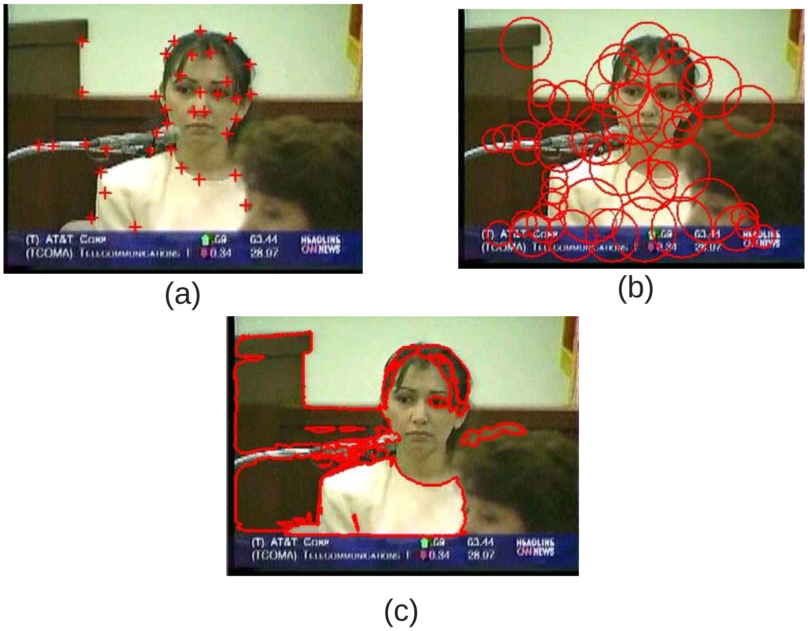


Figure 1.3. Examples of part detection. (a) corner (interest point) detection by the Harris corner detector. (b) salient region detection by the Maximum Entropy Region detector (c) region segmentation by the mean-shift algorithm

detector used in object detection is the maximum entropy region (MER) detector [54]. The MER detector scans over the image. In each location (uniformly sampled on the image plane, or at each pixel), the MER detector initialize a circle and increase the radius of the circle until the entropy of the content within the region reaches maximum. Previous research work [30] has shown that the MER detector performs very well in object detection.

Figure 1.3 shows examples of part detection using corner detection, region segmentation and salient region detection.

1.2.2 Feature Extraction

After part detection, feature extraction is applied to extract the attributes of the parts. Features can be spatial features such as the coordinates of the corner points or the centroid of the regions. They can be color features such as color histograms within the regions; They can be texture features such as Gabor wavelet coefficients [22] or steerable filter coefficients [31]. Recently, Lowe *et al.* [63] has proposed a new local descriptor that is extracted by Scale Invariant Feature Transform (SIFT), which has been demonstrated as the most robust feature against the common image transformations [69].

1.2.3 Part-based Representation

Part-based representation is intended to group the detected parts and their features all together into an integrated entity as an input to the analysis system. Part-based representation represents an image as a collection of individual elements. Relationship among these elements may or may not be modeled. If the relationship among parts are not modeled, the representation is a bag-based representation. The bag-based representation has been widely used in information retrieval, and computer vision. For example, bag-of-pixel model was proposed in [51] for image representation, bag-of-keypoints was used in object detection [21]. If the relations among the parts are modeled, then it is a graph-based representation.

There are different types of graph-based representations. The simplest one is the structural graph model, where the attributes of parts are not taken into account. Such representation can be used for modeling aerial images, maps and trademarks [77]. However, the structural graph model has limited representation power for general image data, such as natural images. Attributed Graph (AG) or Attributed Relational Graph (ARG) [42] is an augmented graph model, in which vertices and edges are associated with discrete or real-valued feature vectors to represent the properties of parts. Therefore, ARG can be used to represent general image data. There are different types of ARGs. If the attributes are integers or symbols, the model is called labeled graph. If there is no loop in the graph, the graph becomes Attributed Tree. Labeled graphs are important models for the research fields such as structural

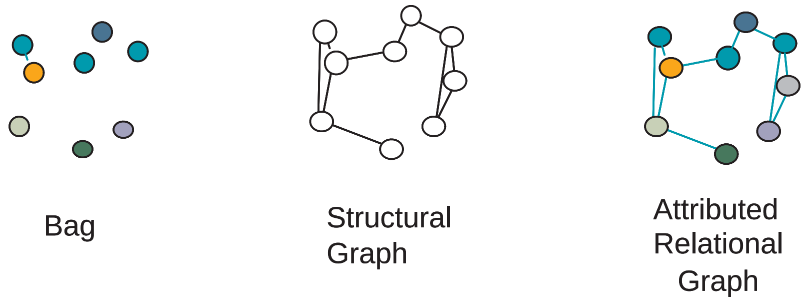


Figure 1.4. Examples of part-based representation. Color in the ARG indicates different feature associated with the nodes

chemistry or structural biology. Compared with general Attributed Relational Graph, labeled graphs also hold many special mathematical properties. Therefore they are also subjects of research on combinatorics, such as analytic combinatorics [76]. For image applications, labeled graphs are less useful due to its limited representation power. Figure 1.4 illustrates several examples of part-based representation.

There are two prominent distinctions between part-based representation and vector-based representation. First, in vector-based representation, every feature vector has the same number of components, i.e. the same dimension. Second, the orders of those components are fixed in vector-based representation. Part-based representation is more general than vector-based representation. In part-based representation, the number of parts could vary across different data, and there is no order associated with individual parts. These two distinctions result in the problems of finding correspondences of parts and dealing with the removed or inserted parts.

1.3 Major Problems Addressed and Contributions

This thesis exploits the statistical modeling and learning methods for part-based methods for solving a few problems in visual content analysis and indexing. Specifically, we investigate the following interesting problems. First, we exploit the learnable similarity of images in the context of part-based models. Second, we establish formal statistical models for describing the topological and attributive properties of a class of objects for object detection. Third, we study the problem of assigning labels to segmented regions in images through higher-order relation modeling. We elaborate each of these problems below and present a brief overview of the past work and our main contributions. In each of the subsequent chapters, we will provide more detailed review of the prior work.

1.3.1 Image Similarity

Measuring image similarity is a problem arising from many multimedia applications, such as content-based image retrieval, visual data mining, etc. Image similarity itself can be formulated in a straightforward way: given two images I_1, I_2, output a real-valued number $S(I_1, I_2)$ that measures their similarity. In the traditional content-based retrieval systems, similarity is simply calculated from the feature vectors that represent images, for example the inner product of two color histograms. Such similarity definition is simple, easy to compute and often holds good mathematical properties favorable for efficient algorithm designs. Yet this definition is far from optimal. For instance, two images with exactly identical color histogram could contain totally different contents.

We suggest that a good similarity measure should be the one that is consistent with human judgment. Namely, our proposed measure for image similarity should be motivated by the Human Vision System.

Regarding learning of image similarity, there has been much work on similarity learning based on vector-based presentation, for example [14][37]. It is difficult for these approaches to achieve good similarity due to the limitation of the feature vector based representation mentioned above.

In comparison, part-based representation provides more comprehensive information about the visual scene in an image. It cannot only capture the appearance features of an image but also can characterize the relationship among parts. In the thesis, we establish a learnable similarity framework for part-based representation. We focus on the most general representation, i.e. Attributed Relational Graph (ARG). The similarity is defined as the probability ratio (a.k.a odds) of whether or not one ARG is transformed from the other. This definition is partly inspired by the relevance model [79] used in information retrieval, where document retrieval is realized by calculating the probability ratio of whether or not the query is relevant to the document.

Transformation based similarity definition is not new, but most of the prior work is based on computing the cost of transformation. Typical examples include string or graph edit distance [78], Earth Mover Distance (EMD) [80], etc. It is yet unknown how to establish formal learning methods on top of these deterministic frameworks. On the other hand, probabilistic methods for matching the vertices of ARGs are also not new. Bayesian methods have been proposed and extended for matching structural or labeled graphs in previous papers [77][19]. However, although the Bayesian method is formal and principled for matching vertices, how to define the similarity and aggregate the vertex-level similarity to the graph level remains a problem. Moreover, without the definition of the graph-level similarity, learning parameters often has to resort to vertex level annotations (vertex matching between two ARGs). Annotating vertex correspondence is very time-consuming since typically the vertex number of an ARG for representing an image ranges from 50 to 100.

One of the main contributions of this thesis is the development of a principled similarity measure for ARGs. We show how this similarity measure relates to the partition functions of the Markov Random Fields (MRFs) that is used for matching vertices. The log convexity of the partition functions of MRFs leads to dual representations (a.k.a variational representation) of the partition functions, which are linear and tractable. The dual representation allows us to develop a maximum likelihood estimation method for learning parameters.

The developed approach has been successfully applied to detection of Near-Duplicate images in image databases. Image Near-Duplicate (IND) refers to two images that look similar but have variation in content, such as object insertion and deletion, or change of imaging conditions,such as lighting change, slight view point change, etc. Near Duplicate images are abundant in Internet images and broadcast news videos. Detecting INDs are useful for copyright violation identification and visual data mining. We compare the learning-based method with the traditional energy minimization based method for ARG similarity computation. The experimental results demonstrate that the learning-based method performs far better than the traditional methods.

1.3.2 Object Detection

Object detection is a classic problem in computer vision. Here, the term "object detection" refers to a general problem that encompasses both object class detection (answering the question if an image contain or not contain the specified object), and object recognition (answering the question what the object is after it has been detected). Taking an example of face detection and recognition, face detection is an object class detection problem, whereas face recognition is an object recognition problem. Technically, the object class detection problem is a binary classification problem, while the recognition problem is a multi-class classification problem.

Research on object detection has evolved from simple template matching to very sophisticated statistical methods in the past decades. Template matching [68] may be the earliest exercise by computing the distance between the predetermined templates and the subimages within an image. The poor performance of this approach implies that object detection needs better representation and modeling of noise rather than simple pixel level distance. Appearance based methods are similar to template matching but include additional mechanism to take into account content uncertainty and noise. Principle Component Analysis (PCA) [97] was a classic example of appearance based method by applying techniques from linear algebra. PCA consists of two components, learning and recognition. In learning, PCA discovers the subspace of the image matrix, whose dimension in general is much smaller than that of the original space. In recognition, PCA projects the incoming image onto the subspace and compute the distance between the resulting vector and the template vectors. PCA is not merely a dimension reduction technique. Its real advantage against template matching is the capability of separating the signal part (the projected feature vector) and the noise (the remainder of the projection). This leads to a more formal generative model called probabilistic PCA [95], which explicitly formulate an image as the addition of an intrinsic image and a noise image. Those two images are generated from certain probability density functions. The advantage of the probabilistic PCA is its capability of modeling the noise more accurately. Beyond the generative model, researchers also realized that for a classification problem, although Bayesian decision based on generative models is optimal, it remains difficult to choose good probability density functions and estimate the parameters accurately if the training data set is small. Therefore, it is more favorable to directly optimize the classification performance itself, rather than learn the density function indirectly. Linear Discriminant Analysis (LDA) [27] and Support Vector Machines (SVMs) [73] are two of the discriminative algorithms that are widely used in the context of appearance models.

Another category of object detection methods is the feature-based method [66][17] (here ”feature” means the component of an object. Therefore, its meaning is different from the ”feature” in feature-based representation) . This model is very close to the part-based approach. In feature-based methods, object recognition is realized by detecting the features (such as eye corners in faces) and measuring their relations. For example, in face recognition, feature-based approach finds the locations of the eyes, nose, and computes the distances between those features as feature vectors. Feature-based approach can be considered the precursor of the part-based models. The method have then evolved into more formal graph-based representation. One of the extension is the well-known Elastic Bunch Graph (EBG) model for face recognition. In EBG, vertices of the graph represent the parts of the face and edges represent their relations. Gabor coefficients (also referred to as Gabor Jets) are extracted to describe the local appearances of the parts. Matching between the graph and the images are realized by energy minimization. Other similar ideas have been developed. For example, pictorial structure represents an object as a star-like graph [29]. More recently, the constellation model[102][30] has been proposed for object detection. Constellation model differs from the graph-based models by modeling the spatial locations of the parts as a joint Gaussian function. We will review these models in more details in Chapter 3.

Apart from above mentioned two methodologies. Another less used approach is the shape-based method [57][6]. In this approach, the shape features are extracted from the contour of an object and sent to classifiers for learning and recognition. The main problem of shape-based approach is that the extracted contour needs to be accurate. However, this is often difficult to achieve for objects in natural images.

Inspired by these previous work, we have developed a new method for object detection called Random Attributed Relational Graph (Random ARG). The model extends the traditional Random Graph [25] by associating the vertices and edges of a graph with random variables. The random variables thereby are used to characterize the variations of part appearances and relationships. Compared with the pictorial structural model, the advantage of the Random ARG model is its ability to model the part occlusion statistics. Compared with the constellation model, the Random ARG model has the capability of modeling a class of objects under different views. Furthermore, we show that the learning of Random ARG can be achieved by a variational Expectation-Maximization algorithm. Our experiments show that the Random ARG model achieves comparable performance with the Constellation model for detecting single-view objects, but uses much fewer learning iterations towards convergence. In the thesis, we further extend the single Random ARG model to a mixture model to achieve better accuracy for modeling multi-view objects.

1.3.3 Region Labeling

Region labeling refers to assigning semantic labels to regions that are extracted by region segmentation algorithms.

Markov Random Field (MRF) [35][62] is a traditional approach to the region labeling problem. However many of the previous approaches for region labeling only consider pairwise relationships (such as relative locations) between regions[70]. For most object categories, pairwise relationship is sufficient. However, for some object categories, such as visual text, it is important to model higher-order relations. Visual text refers to text that is overlaid on images or videos or embedded in image scenes. In images, characters in a text line usually form a straight-line or smooth curve. The straight-line constraint is a higher-order relational constraint that cannot be accommodated by pairwise MRFs. In this thesis, we transform the high-order relational rules into a probabilistic model described by a higher-order MRF (MRF with higher-order potential functions). The formulation reduces the text detection problem to a probabilistic inference problem. We developed a Belief Propagation algorithm for the higher-order MRF to realize efficient statistical inference. Regarding text detection in images and videos, most of the prior work deal with the problems with *ad hoc* approaches, lacking a systematic framework for learning parameters and performance optimization.

1.4 Overview of the Thesis

In summary, the main contributions of this thesis are as follows:

1. A new learning-based algorithm for computing the similarity of Attributed Relational Graphs and its application to measuring image similarity and detecting Image Near-Duplicate.
2. The Random Attributed Relational Graph model with its learning method, and its application to object detection and object model learning.
3. Region labeling by Higher-Order Markov Random Fields with a learning method based on Higher-order Belief Propagation algorithm for higher-order MRFs, and application to scene text detection in images and videos.

The remainder of the thesis is organized as follows:

Chapter 2: Learnable Image Similarity using Attributed Relational Graph

This chapter establishes a learning-based framework for computing the similarity of Attributed Relational Graphs, and its application to detecting Image Near-Duplicate in image databases.

The chapter starts with the prior work on Attributed Relational Graph matching and similarity. We then propose a general principle for measuring the similarity of data, called similarity by statistical transformation. The key idea is to define the similarity as the probability ratio (odds) of whether or not one ARG is transformed from the other. We then present the method of designing the transformation for ARG similarity. We show how to map the transformation into a Markov Random Field (MRF) for calculating the probability ratio. Importantly, we show that the probability ratio is related to the partition functions of the MRFs. The log convexity of the partition functions allows us to develop approximation methods for the similarity calculation and maximum likelihood learning through variational approximation. Finally, we present the application of this method to detecting Image Near-Duplicate from distributed sources. The performance of this method is compared with previous methods for IND detection and the traditional method for ARG matching.

Chapter 3: Learning Random Attributed Graph for Part-based Object Detection

This chapter is the extension of the chapter 2. In chapter 2 we model a generative process that transforms one ARG to the other. In this chapter, the generative process models the generation of one ARG from a Random Attributed Relational Graph (Random ARG). We start with the definition of the Random ARG model. Then we establish the Bayesian decision framework through a generative model. Similar to the Chapter 2, we realize the vertex matching and likelihood computation through rigorous design of Markov Random Fields (MRFs). We exploit the properties of the MRFs and show that the probability ratio of object detection relates to the partition functions of the MRFs, which leads to variational inference and learning schemes. We also exploit the object model under different views by incorporating a mixture model. The chapter ends with experiments to compare our proposed methods with the Constellation model, which is considered as the state-of-the-art model.

Chapter 4: Image Region Labeling with Higher-order Statistical Relation Model

In this chapter, we deal with the problem of assigning segmented regions with labels through higher-order relational model. Specifically, we are interested in the scene text detection problem. Namely, we want to label a segmented region in the image scene either as “text” or “non-text”. The problem is modeled as a probabilistic inference problem, where we need to compute the marginal probability of the label assigning to each region given the observations and labels of all other regions. This is a bit different from chapter 2 and 3, which deal with the problem of computing probability ratio, where probabilistic inference is an implicit problem embedded in the variational approximation.

The chapter starts with the formation of Region Adjacency Graph, upon which a Markov Random Field is defined to model the higher-order relations among the regions. We study the efficient inference algorithm under the higher-order Markov Random Field (MRF) model. The Belief Propagation (BP) algorithm is extended to the higher-order MRF using similar derivations developed in [111], which views the BP algorithm as an optimization method for solving the variational approximation problem. In experiments, we exploit the empirical performance of the higher-order relational model by comparing it with the pairwise Markov Random Field and demonstrates promising performance gain.

Chapter 5: Summary and Future Work

This chapter summarizes the results and contributions of the thesis. We discuss the main limitations of the current work, and explore several possible directions that could lead to future advances. We further mention the potential future applications or extensions beyond computer vision.

Chapter 2

Learnable Image Similarity using Attributed Relational Graph

2.1 Introduction

In this chapter, we investigate a novel image similarity measure using the Attributed Relational Graph (ARG) representation. We represent an image as an Attributed Relational Graph (ARG), a mathematical model that extends the ordinary graph in graph theory by attaching discrete or continuous feature vectors to the vertices and edges of the graph. Image similarity therefore can be defined by the corresponding ARG similarity.

There are essentially two fundamental problems related to ARG models: matching vertices and computing the similarity of two ARGs. The vertex matching problem has been extensively investigated previously. However, relatively little attention has been paid to the definition and computation of ARG similarity. Yet, the problem is important and challenging for image/video indexing in unconstrained domain, such as news videos, where similarity is often subjective and domain specific. It is therefore of great interest to develop a principled

ARG similarity framework that is able to learn the similarity from training data provided through subjective labeling.

This chapter presents a novel statistical similarity measure for ARGs that is able to learn from training data. We define the similarity of two ARGs as the probability ratio (also known as “odds” in statistics) of whether or not one ARG is transformed from the other. The transformation then is reduced to a Markov Random Field which is defined on the association graph between two ARGs. We shall prove that the probability ratio of the transformation equals to a ratio of the partition functions of the MRFs. This important property then leads to an approximation scheme, called variational approximation [48], to compute the probability ratio and learn the parameters of the transformation. More importantly, the proposed learning scheme is able to learn the parameters in an unsupervised manner, meaning that we only need to annotate whether or not two ARGs are similar without having to annotate the vertex correspondence. The definition using probability ratio, the learning framework and the matching algorithm are novel compared with the previous methods. We apply the algorithm to detecting Image Near-Duplicate (IND), which is defined as two similar images that often correspond to the same event and scene but involve changes of content and capturing conditions, in a broadcast news video database. The experiment results have shown that our proposed approach significantly outperforms prior approaches that are based on energy minimization or image feature matching.

2.1.1 Related Work

Research on ARG matching can be dated back to the seminal work of Barrow [42], who proposed the basic concept of modeling image scenes using relational structures. Since then, the basic ideas have been extended. Researchers have been focusing on three major problems:

similarity definition, vertex matching, and optimization method. We review related prior work in the following.

Early work on ARG or graph similarity can be found in the structural pattern recognition literature. Similarity in these papers was defined in an *ad hoc* manner. For instance, Shapiro and Haralick [85] define similarity by counting consistent subgraphs of two graphs. The similarity is then refined by Eshera and Fu [26], Sanfeliu and Fu [82] by extending the edit distance of strings to graphs. Bunke *et al.* [40] then showed that the edit distance is related to the size of the maximal common subgraph. Similarity defined by these approaches consider mainly the about structural graph without accounting for attributes. In order to extend the method to attributed graphs. Wilson and Hancock [77] proposed a similarity measure for labeled graphs using the aggregated Hamming distance between the node labels together with the size difference of the graphs. The idea is extended by Myers *et al.* [78] to combine the edit distance with Wilson and Hancock's Bayesian framework. Another method for computing graph similarity is based on the information theoretic measure. For instance, Wong and You [108] have tried to define an entropy measure for matching structural graphs. For ARGs, Shi and Malik [87] has attempted to connect the ARG similarity with the eigenvalue of the affinity matrix of the graph, where the two ARGs have to be of the same size. For general ARGs associated with real-valued multi-variate features, ARG similarity is often defined by the minimum value of the energy function used to associated with the vertex matching process [42].

Vertex matching is another important problem, based on which the similarity can be computed. There are two approaches to solving this problem: energy minimization and Bayesian formulation. Energy minimization approaches reduce the vertex matching problem to a cost optimization problem. For instance, in [42], the energy function is defined as the sum of distance between the vertices in the two input graphs. Other formulations include linear programming and integer programming[4][74],which are used in matching weighed graphs and shock graphs respectively. Another framework, Bayesian formulation, was first proposed by Wilson and Hancock [77] and further refined by others in [19][78]. The idea is to cast the graph matching problem as the maximization of the posterior probability of the vertex correspondence, which is defined as a functional that maps the vetexes of one ARG to the other. In [77], the probability calculation is realized by constructing a dictionary of super-clique mapping and defining the probabilities for the entries. To overcome the complexity issues associated with the dictionary construction procedure, the method has been improved by Myers *et al.* [78] using the edit distance concept. The framework was also extended to hierarchical graphical structures in [105]. Markov Random Field method [61] can be considered as another Bayesian formulation for ARG matching, which copes with the ARGs associated with multi-variate features. The optimal vertex matching is realized by a *Maximum A Posteriori* method, which is equivalent to minimizing an energy function.

Optimization is a central problem for vertex matching and similarity computation. The optimization methods can be classified into two categories: discrete optimization and relaxed continuous optimization. Discrete optimization attempts to directly minimize the energy function by varying the vertex matching configurations. Examples include Simulated Annealing [41], Genetic Algorithm [20], and tabu search [104]. Although some of the methods in this category guarantee to find the local minima, its computational cost is prohibitive for practical applications. In comparison, relaxed continuous optimization is more computationally favorable. The method relaxes the original equivalent integer programming problem to a real-valued problem that can be solved by existing optimization techniques. For instance, Pelillo *et al.* [74][75] find a solution that adopts the methods based on differential equations to minimize the energy function. Almohamad and Duffuaa[4] used a linear programming based approach. More recently, Schelleald and Schnorr [83] formulated the subgraph isomorphism problem by semidefinite programming. Spectral method is another widely used method, which reduces the optimization problem to finding eigen vectors. Thresholding the principal eigen vector yields an approximate solution for the integer programming problem. Examples include methods proposed by Umeyama [98], Shapiro and Haralick[86], Carcassoni *et al.* [12], and Scott *et al.* [84]. Another related relaxation method is iterative refinement based approach, or Expectation-Maximization algorithms. This includes the methods proposed by Gold and Rangarajan [36], Luo and Hancock[9] and Cross and Hancock [19].

Image similarity is a fundamental problem for image retrieval and search. There has been substantial work on image similarity from the image retrieval community. Global feature extraction and distance computation [119][7][90][14] are traditional methods for image retrieval. In order to achieve better representations of images, region-based image similarity is another promising direction that has gained significant interests in recent years [91],[72],[15]. Among different representation methods for region-based image retrieval, Attributed Relational Graphs have been widely used for representing images by regions [77][61][87][58].

As image similarity is often a subjective measure and it is hard to define its computable form manually, learning image similarity is often favorable for an image retrieval system. Learning-based methods for vector-based representation have been extensively investigated in [37][13]. However, there still lacks a systematic and principled method to learn the similarity of images based on region-based or part-based representations.

Based on the review of the literature, we note that despite much work on ARG vertex matching and similarity, there still lacks a principled similarity measure for general ARGs, in which the vertices of ARGs are associated with real-valued and multivariate feature vectors. Computing the similarity of such types of ARGs are useful for many problems, particularly for images, videos, which are difficult to be represented as structural or labeled graphs. On the other hand, the similarity of such ARGs are often subjective and domain-specific. Therefore, it is important to develop a method to learn the similarity measure directly from training data rather using measures handcrafted by researchers.

2.1.2 Overview of the Proposed Method

The methods for ARG matching developed in this thesis is motivated by a practical problem called Image-Near-Duplicate (IND) detection, which aims at finding two similar images that are often captured at the same scene/event but have slight difference in content and capturing conditions. Whether or not two images are Near-Duplicate is often subjective, and even confuses human judgers. Therefore it is highly desirable to have a learning-based approach which learns the required parameters from the training data.

Our proposed approach first constructs a transformation that transforms the model ARG (the first ARG) to the data ARG (the second ARG). The ARG similarity is then defined as the probability ratio of whether or not the data graph is transformed from the model graph. The idea is partly inspired by the relevance model [79] used in information retrieval, where the probability ratio of whether or not a document is relevant to the query is taken for ranking documents. To distinguish our approach from the Bayesian framework[77], we refer to the proposed approach as *stochastic ARG matching*[115] in that the transformation can be deemed as a stochastic process comprised of multiple actions. The main difference between our method and the Bayesian method [77] is the Bayesian method focuses on the vertex matching problem, while computing the ARG similarity by aggregation of vertex similarity in an *ad hoc* manner. In contract, our approach focuses on the similarity problem while treating the vertex matching as an implicit problem. Such top-down approach allows us to learn ARG similarity in an unsupervised manner, which has not been shown feasible by any prior work using the Bayesian framework.

In order to calculate the probability ratio, we map the parameters of the transformation to pairwise Markov Random Fields (MRFs). We show that the probability ratio of the transformation is related to the partition functions of the MRFs. The partition function involves the summation over all possible vertex matching schemes. The resulting similarity there-

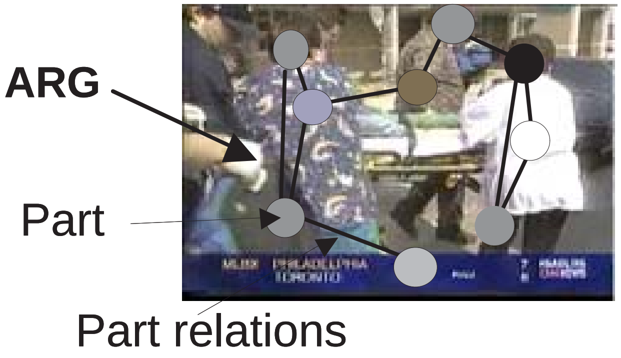


Figure 2.1. Attributed Relational Graph for Part-based Representation

fore takes into account the contributions from all possible matchings rather than only the one with the maximum likelihood. The convexity of the log partition function allows us to use variational methods to approximate the similarity and learn the parameters.

2.2 Similarity by Transformation

Attributed Relational Graph (ARG) generalizes the notion of "graph" by attaching multivariate feature vectors to the vertices and edges. The incorporation of attributes allows ARGs to model not only the topological properties of real-world entities but also their attributive properties. Formally, ARG is defined as follows.

Definition 2.1. *An Attributed Relational Graph (ARG) is defined as a triplet $G = (V, E, A)$, where V is the vertex set , E is the edge set, and A is the attribute set containing attribute y_i attached to each node $n_i \in V$, and attribute y_{ij} attached to each edge $e_k = (n_i, n_j) \in E$.*

Figure 2.1 illustrates the use of ARG to represent an image. Note that while the ARG shown is not fully connected, most of the previous work in computer vision assume that the ARG is fully connected. This assumption is legitimate in that the connectivity of two vertices itself can considered as a binary attribute defined at edges. Therefore, without loss of generality, we only focus on fully-connected ARGs throughout this chapter. Also notice that although we only define the relational features up to the second order, higher-order relational features can be easily incorporated by adding attributes defined over cliques.

Unlike the conventional feature vector based representation, where inner product or normed distance can be used to measure the similarity, the definition of ARG similarity poses two significant challenges. First, the correspondences of the vertices of the two ARGs are unknown *a priori*, and the number of possible correspondences is exponential in the product of the ARG sizes. Second, the two input ARGs could be of different sizes.

We propose a principle for similarity computation called *similarity by transformation*. The approach is to design a transformation (could involve multiple actions) to transform the model graph to the data graph, and then use the probability ratio of the transformation as the similarity measure. The idea is related to the edit distance [26][82] used in the structural graph matching and string matching, which measures the similarity by the cost of transformation (editing). In comparison, we extend the notion of cost to the likelihood of a stochastic transformation. Conceptually, the likelihood function is related to the transformation cost, with higher cost the less likelihood. We use probability ratio instead of positive probability because decision based on the probability ratio is optimal in terms of Bayesian classification theory. The use of probability ratio is consistent with the probability models [**?**][79] used in information retrieval.

We follow the previous papers [77] to use the following conventional notation in the subsequent sections: G_m denotes model graph with N vertices, and G_d data graph with M vertices. We consider two hypotheses of the transformation. $H = 1$ indicates that the data graph G_d is transformed from the model graph G_m; $H = 0$ otherwise. The use of negative hypothesis allows the algorithm to exploit the characteristic of negative training data. The similarity $S(G_m, G_d)$ is defined as the following

$$S(G_m, G_d) = \frac{p(H = 1|G_m, G_d)}{p(H = 0|G_m, G_d)} \tag{2.1}$$

By applying Bayes rule, we have the following

$$\frac{p(H = 1|G_m, G_d)}{p(H = 0|G_m, G_d)} = \frac{p(G_d|H = 1, G_m)\frac{p(H=1|G_m)}{p(G_d|G_m)}}{p(G_d|H = 0, G_m)\frac{p(H=0|G_m)}{p(G_d|G_m)}} = \frac{p(G_d|H = 1, G_m)p(H = 1|G_m)}{p(G_d|H = 0, G_m)p(H = 0|G_m)} \tag{2.2}$$

For some applications, we may need to use the distance between two ARGs instead of the similarity, in which case we can take $d(G_m, G_s) = \frac{1}{S(G_m,G_d)}$ as the distance between G_d and G_m. According to Eq.(2.2), it is apparent that such distance is positive and symmetric. However, it is not necessary to satisfy the triangular inequality. For instance, considering two degenerated ARGs G_1, G_2, each of them only has one vertex and assume that the similarity between them is zero. Then let us form the third ARG by a union operator $G_3 = G_1 \bigcup G_2$. By the definition we should have $S(G_3, G_1) > 0$ and $S(G_3, G_2) > 0$, therefore $d(G_3, G_1) + d(G_3, G_2) < d(G_1, G_2)$, which violates the triangular inequality. Therefore, it may be more appropriate to call the inverse similarity as divergence instead of distance. However, this fact does not mean that the proposed similarity measure is invalid. In general, the distance between two data points in a non-linear space is not necessary to satisfy the triangular inequality, for instance, data points on a curved manifold.

We assume that we do not have prior bias on either hypotheses (i.e. $p(H = 1|G_m)/p(H = 0|G_m) = 1$). Therefore, we have

$$S(G_m, G_d) = \frac{p(G_d|G_m, H = 1)}{p(G_d|G_m, H = 0)} = \frac{p(G_d|G_m, H = 1)}{p(G_d|H = 0)} \tag{2.3}$$

Here we assume that G_d is independent of G_m if G_d is not transformed from G_m. And the term $p(G_d|G_m, H = 1)$ characterizes the transformation from G_m to G_d, which will be referred to as *transformation likelihood*.

Let Y_i, Y_{ij} with $i, j \leq N$ denote the features of the model graph G_m defined at its nodes and edges, and Y_u, Y_{uv} with $u, v \leq M$ denote the attributes of the data graph G_d. For the negative likelihood function $p(G_d|H=0)$, we assume that the the node and edge attributes y_u and y_{uv} are independent, i.e.

$$p(G_d|H=0)=\prod_u p(y_u|H=0)\prod_{uv} p(y_{uv}|H=0)=\prod_u f_{B_1}(y_u)\prod_{uv} f_{B_2}(y_{uv}) \qquad (2.4)$$

where $f_{B_1}(\cdot)$ is the *pdf* characterizing the appearance statistics of an arbitrary image part, and $f_{B_2}(\cdot)$ is the *pdf* characterizing the part relations. Therefore, $f_{B_1}(\cdot)$ and $f_{B_2}(\cdot)$ with learned parameters characterize the image statistics within certain domain (e.g. news video). In our experiments, the parameters of $f_{B_1}(\cdot)$ and $f_{B_2}(\cdot)$ are learned from the negative training set.

For the *transformation likelihood* $p(G_d|G_m,H=1)$, it is difficult to factorize because the node correspondences between the two ARGs are unknown. Yet we can treat the correspondence as missing variables, and marginalize over all possible correspondences. To do this, we introduce another variable X to represent the correspondence. X consists of a set of binary random variables x_{iu}, with $x_{iu}=1$ if the node i in the model graph corresponds to the node u in the data graph; $x_{iu}=0$ otherwise. Therefore, we have $X=\{x_{11},...,x_{iu},...,x_{NM}\}$. Furthermore, if we associate each x_{iu} with a node, then all of these nodes together with their inter-connections form a complete graph, called the association graph (Figure 2.2). And the nodes together with the binary random variables form an undirected graphical model (a.k.a Markov Random Field(MRF)), whose joint probability (called Gibbs distribution) can be designed by specifying the potential functions of the MRF. By introducing X, we can achieve the following factorization for the *transformation likelihood*

$$p(G_d|G_m,H=1)=\sum_X p(G_d|X,G_m,H=1)p(X|G_m,H=1) \qquad (2.5)$$

This factorization can be illustrated as a graphical model shown in figure 2.3. To specify the two terms $p(G_d|X,G_m,H=1)$ and $p(X|G_m,H=1)$ in the factorization, we first need to design the transformation.

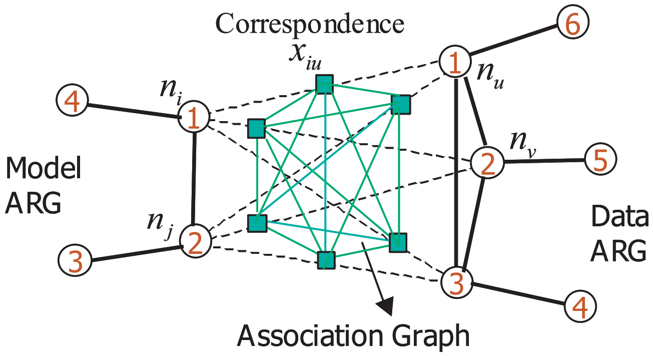


Figure 2.2. The association graph upon which the MRFs are defined.

2.2.1 Design of Transformation

The design of the transformation is largely dependent on specific applications. We focus on the application of image matching, where each vertex of the ARG denotes one part (region, interest point etc.) of the image. The content change in the two images usually involves the additions and occlusions of objects, object movements and appearance changes, and other variations such as lighting condition change. In summary, two types of variations could be considered : First, we use topology change to model part occlusion and addition. For simplicity, let's first model part occlusion. We assume that the parts of the model graph are occluded with probability $1 - r$. In other word, r is the probability of the vertices of the model graph being copied to the data graph. For brevity, we refer to r as *vertex copy probability*. Second, we use attribute change at vertices and edges to model part appearance change, part movement and other variations. For the attributive transformation, if the vertex u in G_d corresponds to the vertex i in G_m, then we let the vertex attribute y_u to be sampled from a Gaussian *pdf* with mean y_i and variance Σ_1, denoted as $f_1(y_u; y_i)$. This means that the node attribute in G_d is the copy of the attribute in G_m plus the distortion characterized by the Gaussian *pdf*. In Near-Duplicate images, this distortion is caused by object movement, part appearance change or other reasons. Likewise, the edge attribute y_{uv} is sampled from

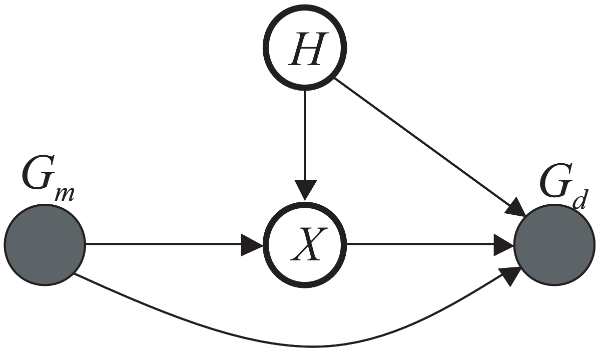


Figure 2.3. The generative model for ARG matching

a Gaussian *pdf* with mean y_{ij} and variance Σ_2, denoted as $f_2(y_{uv}; y_{ij})$. The transformation characterizes the relative position changes between parts in two Near-Duplicate images.

The above defined transformation may be too simplistic for general image transformation. However, it is not difficult to incorporate more sophisticated transformation, such as camera transforms, which has been investigated in [33][53] as an image generative model. In our current system, the camera transform is not used because our experiments have shown that camera transforms have little contribution to performance improvement.

2.3 Computing the Transformation Likelihood

We have specified the pdfs of the transformation as well as their parameters. Since the computation of the *transformation likelihood* has to be conducted in the MRFs, we need to map the parameters of the transformation to the MRFs.

2.3.1 Map the Parameters of Transformation to MRF

We have introduced the MRFs in the section 2.1.2. In the MRF, each node n_{iu} (note we use double index to tag the nodes in MRF) encodes a possible vertex matching between the node i in the model graph and the node u in the data graph. In practice, the resulting MRF is often very large (NM nodes). In order to reduce the computational cost, we can prune the MRF by discarding the vertex n_{iu} if feature distance between the corresponding two nodes i and u is too large. As a result, the node i in G_m is only allowed to match d_i number of vertices in G_d, with $d_i \ll M$. We call the resulting MRF as pruned MRF. We will discuss the performance degradation due to punning in the experiment section.

In the following, we will relate the *transformation likelihood* to the quantities defined in the MRFs. Before that, let us map the *pdf*s and their parameters of the transformation to the MRF.

The factorized *transformation likelihood* in Eq.(2.5) involves two terms, the prior probability of correspondence $p(X|G_m, H = 1)$ and the conditional density function $p(G_d|X, G_m, H = 1)$.

The prior probability of correspondence $p(X|G_m, H = 1)$ is designed so as to serve two purposes : enforce the one-to-one matching constraint (one node in the model graph can only be matched to one node in the data graph) and encode the *vertex copy probability*. The MRF is designed with the following Gibbs distribution

$$p(X|G_m, H = 1) = \frac{1}{Z} \prod_{iu,jv} \psi_{iu,jv}(x_{iu}, x_{jv}) \prod_{iu} \phi_{iu}(x_{iu})$$

where Z is a normalization constant, also called partition function. The 2-node potential function $\psi_{iu,jv}(x_{iu}, x_{jv})$ is used to enforce the constraint, designed as

$$\begin{aligned} \psi_{iu,jv}(x_{iu} = 1, x_{jv} = 1) &= 0, \text{ for } i = j \text{ or } u = v; \\ \psi_{iu,jv}(x_{iu}, x_{jv}) &= 1, \text{ otherwise} \end{aligned} \tag{2.6}$$

The first line of the above equations is the repulsive constraint used to enforce the one-to-one correspondence. The explanation of this potential function is straitforward: if the one-to-one correspondence constraint is violated, the probability of correspondence will drop to zero. The 1-node potential function is defined as

$$\phi_{iu}(0) = 1, \quad \phi_{iu}(1) = z$$

where z is a real number related to the *vertex copy probability*. We set $\phi_{iu}(0) = 1$ because it is not difficult to show that any $\phi_{iu}(1)$ and $\phi_{iu}(0)$ with the same ratio $\phi_{iu}(1)/\phi_{iu}(0)$ would result in the equivalent MRF distribution (with different partition functions). The partition function Z of the MRF is one of the most important quantities of the Markov Random Field. And it can be used to derive several important properties of MRF. In the above MRF, Z is the function of M,N and z. The following Lemma relates the *vertex copy probability* to the MRF parameter z

Lemma 2.1. *z and r are related by the following equation*

$$r = \frac{z}{N}\frac{d\mathrm{ln}Z}{dz} \tag{2.7}$$

which is true for both pruned and original MRFs.

This relationship is useful to derive the equations for learning r. It also turns out that there is no closed form solution for directly learning z from training data. After r is obtained, we need to map r back to z for MRF computation. We found that the closed form for mapping is also unavailable. However, we can resort to the following approximation

Lemma 2.2. *The log partition function of the original MRF satisfies the following inequality*

$$\mathrm{ln}Z(N;M;z) \leq N\ln\big(1 + Mz\big) \tag{2.8}$$

For the pruned MRF, the log partition function satisfies the following inequality

$$\mathrm{ln}Z(N;d_1,d_2,...,d_N;z) \leq N\ln\big(1 + max_i\{d_i\}z\big) \tag{2.9}$$

where d_i is the number of the vertices in the data graph that are possibly matched to the vertex i in the model graph after pruning.

Therefore, using the upper bound as the relaxed version of the log partition function, we can map r back to z using the following approximated relationship $z \approx \frac{r}{(1-r)max_i\{d_i\}}$.

If both graphs are very large, the following proposition can be considered for approximation

Proposition 2.1. *Let $N \leq M$. When $N \rightarrow \infty$, and $M - N < \infty$ The log partition function $\ln Z$ tends to*

$$N\big[\ln(z) + c\big] \tag{2.10}$$

where c is a constant, which can be calculated as

$$c = \lim_{N\rightarrow\infty} \frac{1}{N} \text{log}\Big[\sum_{i=1}^{N} \binom{N}{i}\binom{N}{i} i!\Big]$$

The next step is to design the conditional density $p(G_d|X, G_m, H = 1)$. If we assume that node attributes and edge attributes are independently generated, then we have the following factorization

$$p(G_d|G_m, X, H = 1) = \prod_{u} p(y_u|X, G_m, H = 1) \prod_{uv} p(y_{uv}|X, G_m, H = 1)$$

Furthermore, according to the design of the transformation, the feature of a node and edge in the data graph should only depend on its matched node and edge in the model graph. And if there is no node in the model graph matched to a node in the data graph, the feature of that node should obey the background *pdf*, i.e. $f_{B_1}(\cdot)$, likewise for the edge features. Therefore, we should have the following identities

$$\begin{aligned}
p(y_u|x_{11}=0,...,x_{iu}=1,...,x_{NM}=0,G_m,H=1) &= f_1(y_u;y_i) \\
p(y_{uv}|x_{11}=0,...,x_{iu}=1,x_{jv}=1,...,x_{NM}=0,G_m,H=1) &= f_2(y_{uv};y_{ij}) \\
p(y_u|x_{1u}=0,...,x_{iu}=0,...,x_{Nu}=0,G_m,H=1) &= f_{B_1}(y_u) \\
p(y_{uv}|x_{1u}=0,...,x_{iu}=0,x_{jv}=0,...,x_{Nu}=0,G_m,H=1) &= f_{B_2}(y_{uv}) \qquad (2.11)
\end{aligned}$$

Based on these setups, we can further reduce the probability ratio computation to a new MRF model, as shown in the following theorem.

Theorem 2.1. *The transformation probability ratio relates to the partition functions of the MRFs by*

$$S(G_m,G_d)=\frac{p(H=1|G_d,G_m)}{p(H=0|G_d,G_m)}=\frac{Z'}{Z} \qquad (2.12)$$

where Z is the partition function of the Gibbs distribution $p(X|G_m,H=1)$. Z' is the partition function of the Gibbs distribution that is equivalent to the posterior probability $p(X|G_d,G_m,H=1)$, which has the following forms

$$p(X|G_d,G_m,H=1)=\frac{1}{Z'}\prod_{iu,jv}\varsigma_{iu,jv}(x_{iu},x_{jv})\prod_{iu}\eta_{iu}(x_{iu})$$

where the one-node and two-node potential functions have the following form

$$\eta_{iu}(1)=zf_1(y_u;y_i)/f_{B_1}(y_u);\quad \varsigma_{iu,jv}(1,1)=\psi_{iu,jv}(1,1)f_2(y_{uv};y_{ij})/f_{B_2}(y_{uv}) \qquad (2.13)$$

all other values of the potential functions are set to 1 (e.g. $\eta_{iu}(x_{iu}=0)=1$).

This theorem reduces the problem of computing the probability ratio to computing the partition function Z and Z'.

2.3.2 Computing the Partition Functions

For the partition function Z, we can compute it using numerical methods in polynomial time (Eq.(A.1) in Appendix A), or using Lemma 2.2 for approximation. For the partition function Z', it is a summation over all possible matchings, whose number is exponential in NM. Therefore, approximate methods have to be developed. Fortunately, computing partition function Z' is a standard problem in machine learning. Since the partition function Z' is log convex, it can be represented as a variational form, or in the form of Jensen's inequality as the following [111][100]:

$$\ln Z' \geq \sum_{iu,jv} \sum_{x_{iu},x_{jv}} \hat{q}(x_{iu}, x_{jv}) \ln \varsigma_{iu,jv}(x_{iu}, x_{jv}) + \sum_{iu} \sum_{x_{iu}} \hat{q}(x_{iu}) \ln \eta_{iu}(x_{iu}) + \mathcal{H}(\hat{q}(X)) \tag{2.14}$$

where $\hat{q}(x_{iu})$ and $\hat{q}(x_{iu}, x_{jv})$ are the approximated one-node and two-node marginals of $p(X|G_m, G_d, H = 1)$, also known as beliefs. $\mathcal{H}(\hat{q}(X))$ is the entropy of $\hat{q}(X)$, which is the approximated distribution of $p(X|G_m, G_d, H = 1)$. Computing the entropy again is intractable for the loopy graphical model (here the MRF is fully connected). Yet the entropy $\mathcal{H}(\hat{q}(X))$ can be approximated using Bethe approximation [111], in the following form

$$\mathcal{H}(\hat{q}(X)) \approx - \sum_{iu,jv} \sum_{x_{iu},x_{jv}} \hat{q}(x_{iu}, x_{jv}) \ln \hat{q}(x_{iu}, x_{jv}) + \sum_{iu} (MN - 2) \sum_{x_{iu}} \hat{q}(x_{iu}) \ln(\hat{q}(x_{iu}))$$

Note that the variational approximation becomes exact if the approximated marginal is exact and the entropy function is correct.

2.3.3 Computing the Approximated Marginals

To realize likelihood calculation and learning, the key problem is to compute the approximated marginals, a procedure known as probabilistic inference. There are two major approaches to calculating the approximated marginals: variational method and Monte Carlo method. Variational method maximizes the lower bound in Eq.(2.14) with respect to the marginals. An algorithm known as Loopy Belief Propagation (LBP)[111] can be thought of as an approach to maximizing the lower bound through fixed point iteration[111] (called message passing in AI terminology). The main advantage of LBP is its high efficiency for some graphical models. In complete graphs, the LBP algorithm has complexity $O\big((N \times M)^2\big)$. The modified message passing rules for complete graph are listed below

$$\begin{aligned} m_{iu,jv}^{(t+1)}(x_{jv}) &= k \sum_{x_{iu}} \varsigma_{iu,jv}(x_{iu}, x_{jv}) \eta_{iu}(x_{iu}) \mathcal{M}_{iu}^{(t)}(x_{iu}) / m_{jv,iu}^{(t)}(x_{iu}) \\ \mathcal{M}_{iu}^{(t+1)}(x_{iu}) &= \exp\big(\sum_{kw|kw \neq iu} \ln(m_{kw,iu}^{(t+1)}(x_{iu}))\big) \end{aligned}$$

where $m_{iu,jv}$ is the message passing from the node iu to jv, t is the iteration index, k is a normalization constant to avoid overflow. After the message update terminates, the one node and two node beliefs $\hat{q}(x_i)$ and $\hat{q}(x_{iu}, x_{jv})$ can be computed as following

$$\begin{aligned} \hat{q}_{iu}(x_{iu}) &= k\eta_{iu}(x_{iu})\mathcal{M}_{iu}(x_{iu}) \\ \hat{q}(x_{iu}, x_{jv}) &= k\varsigma_{iu,jv}(x_{iu}, x_{jv})\eta_{iu}(x_{iu})\eta_{jv}(x_{jv})\mathcal{M}_{iu}(x_{iu})\mathcal{M}_{jv}(x_{jv})/\big(m_{jv,iu}(x_{iu})(m_{iu,jv}(x_{jv})\big) \end{aligned}$$

where k is a normalization constant.

However, due to the particularity of the potential functions (Eq.(2.6)) used in our algorithm, LBP messages often do not converge, but oscillate among multiple states. This problem in the past was approached by using momentum-based approach [71], i.e. replacing the messages that were sent at time t with a weighted average of the messages at time t and $t - 1$. However, in our experiments, we observe that the momentum-based method fails to learn correct parameters and result in incorrect vertex matching and similarity. To solve this problem, we use a different approach. In our approach, we do not manipulate the passing messages. Instead, we compare the lower bound values in Eq.(2.14) across message update iterations and select the set of approximated marginals that results in the largest lower-bound.

Specifically, let $\hat{q}^{(t)}(x_{iu})$ and $\hat{q}^{(t)}(x_{iu}, x_{jv})$ be one-node and two-node marginals at iteration t, and let $\hat{q}^{(t)} = \left\{\hat{q}^{(t)}(x_{iu}), \hat{q}^{(t)}(x_{iu}, x_{jv}), 1 \leq i, j \leq N, 1 \leq u, v \leq M\right\}$, then the lower bound in Eq.(2.14) is a functional of $\hat{q}^{(t)}$ denoted as $\mathcal{F}(\hat{q}^{(t)})$. In our approach, after several iterations (5-15) of message passing, we let the final marginals be

$$\hat{q} = \hat{q}^{(\hat{t})}, \quad \hat{t} = argmax_t \mathcal{F}(\hat{q}^{(t)})$$

This approach is consistent with the optimization view of the LBP algorithm proposed in [111]. Namely, we shall select the messages resulting in the largest lower bound instead of averaging them across iterations. Such approach results in satisfactory accuracy of vertex matching and ARG similarity computation in our experiments.

Monte Carlo methods approximate the marginals by drawing samples from certain probability distribution and summing over the obtained samples. We use a special Monte Carlo approach known as Gibbs Sampling (GS) [5]. GS is realized by iteratively sampling the state of each node in the MRF from the conditional probability of the state given the states of all other nodes in the MRF. Concretely, assuming that in the tth iteration we have obtained the sample $x^t = \{x_1^t, x_2^t, ..., x_{MN}^t\}$, we draw the sample in the $(t+1)$th iteration by using following procedure:

Step 1. sample x_1^{t+1} from $p(x_1|x_2^t, x_3^t, ..., x_{MN}^t)$
Step 2. sample x_2^{t+1} from $p(x_2|x_1^{t+1}, x_3^t, ..., x_{MN}^t)$
...
Step MN. sample x_{MN}^{t+1} from $p(x_{MN}|x_1^{t+1}, x_2^{t+1}, ..., x_{MN-1}^{t+1})$

After sampling, the approximated marginal is computed by averaging the corresponding samples. For instance, let $x^1, x^2, ..., x^S$ be the S samples drawn by Gibbs Sampling. Then

$$\hat{q}_{iu}(x_{iu}=1) = \frac{\sum_{s=1}^{S} \mathbf{1}(x_{iu}^s)}{S} \tag{2.15}$$

where $\mathbf{1}(x_{iu}^s)$ is the indicator function with $\mathbf{1}(x_{iu}^s = 1) = 1$ and $\mathbf{1}(x_{iu}^s = 0) = 0$.

Apart from LBP and Monte Carlo methods, new algorithms for inference developed more recently can also be used. For example, the semidefinite relaxation method developed in [100] is a new approach for inference through convex optimization (specifically, log determinant maximization as in [100]). In our experiments, we also implemented the semidefinite relaxation approach. However, the complexity of this approach is at least $O\big((N \times M)^3\big)$, which is too high to be practical in our current application. Therefore, only the LBP and Monte Carlo method are utilized in our current performance evaluation.

2.4 Learning the Transformation Parameters

Conventionally, for example in the previous MRF based approach[61], learning the parameters requires annotations at the vertex level. Namely, the annotators have to mark the correspondences of the vertices in the model graph and data graph. This is a time-consuming procedure since an ARG for image representation typically contains 30-100 vertices. In order to reduce human supervision, it is necessary to develop an unsupervised approach, where annotators only need to annotate whether two ARGs are similar or not.

Unsupervised learning is realized by maximizing the *transformation likelihood* with respect to the parameters. However, directly maximizing the *transformation likelihood* is difficult due to the intractable summation in the partition function. Fortunately, we have shown that the partition function has a variational representation, which has a tractable form and can be used for maximization. Note that if the marginals in Eq.(2.14) is exact, maximizing the lower bound is equivalent to directly maximizing the *transformation likelihood*, regardless of the accuracy of the entropy approximation.

Using the variational approximation, the parameter estimation is a process composed of two iterative steps: (1) probabilistic inference to obtain the approximated marginals, and (2) maximization of the lower bound with respect to the parameters. This is known as the variational Expectation-Maximization (E-M) scheme described as follows:

E Step: Compute $\hat{q}(x_{iu})$ and $\hat{q}(x_{iu}, x_{jv})$ using Loopy Belief Propagation or Gibbs Sampling.

M Step: Maximize the lower bound in Eq.(2.14) by varying parameters. This is realized by differentiating the lower bound with respect to the parameters, resulting in the following update equations

$$\begin{aligned}
\xi_{iu}^{k} &= \hat{q}(x_{iu}^{k}=1); \quad \xi_{iu,jv}^{k} = \hat{q}(x_{iu}^{k}=1, x_{jv}^{k}=1) \\
\Sigma_1 &= \frac{\sum_k \sum_{iu} (y_u^k - y_i^k)(y_u^k - y_i^k)^T \xi_{iu}^k}{\sum_k \sum_{iu} \xi_{iu}^k} \\
\Sigma_{11} &= \frac{\sum_k \sum_{iu,jv} (y_{uv}^k - y_{ij}^k)(y_{uv}^k - y_{ij}^k)^T \xi_{iu,jv}^k}{\sum_k \sum_{iu,jv} \xi_{iu,jv}^k}
\end{aligned} \tag{2.16}$$

where k is the index of the training ARG pairs.

For the *vertex copy probability* r, it can also be obtained by maximizing the *transformation likelihood*. Through Theorem 2.1, we have:

$$\hat{r} = argmax_r \frac{Z'(r)}{Z(r)} p(O|H=0) = argmax_r \Big[\ln\big(Z'(r)\big) - \ln\big(Z(r)\big) \Big] \tag{2.17}$$

By differentiating the term inside the $argmax$ operator and invoking Lemma 2.1, we get

$$r = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{iu} \hat{q}(x_{iu}^k = 1) \tag{2.18}$$

where K is the number of the training instances, and N_k is the node number of the kth model ARG.

For the negative hypothesis, we need to learn the parameters μ_0,Σ_0,μ_{00} and Σ_{00}. The estimates of these parameters are the sample means of the corresponding feature vectors according to the Maximimum Likelihood Estimation (MLE).

2.5 Application to Image Near-Duplicate Detection and Retrieval

We apply the proposed method to detecting and retrieving Image Near-Duplicates (IND) in a news video database. Image Near-Duplicate is defined as two similar images that are often captured from the same site/event, but with variations due to content changes, camera parameter changes, and digitization conditions. Figure 2.4 shows several examples of IND in a typical news video database.

IND detection problem is to determine whether or not a pair of images are near-duplicate. Therefore, IND detection is a binary classification problem. On the other hand, IND retrieval problem is similar to Content-Based Image Retrieval (CBIR). Given a query image, IND retrieval finds the near-duplicate images to the query image. Therefore, the outcome of IND retrieval is a rank list of images. And ideally the near-duplicate images would be at the top of the rank list.

The transformation between one image to the other in IND usually cannot be accommodated by linear transforms such as the affine transform. We represent each image as an ARG, where each part of the visual scene corresponds to one vertex in the ARG. The parts in the image are extracted using interest point detector (Harris corner detector [30]). The number of feature points in an image ranges from 10-50. Local feature descriptors are then extracted around the interest points to represent the appearances of the parts. Each descriptor consists of eleven components: three for RGB colors (average RGB color in the $15x15$ block), two

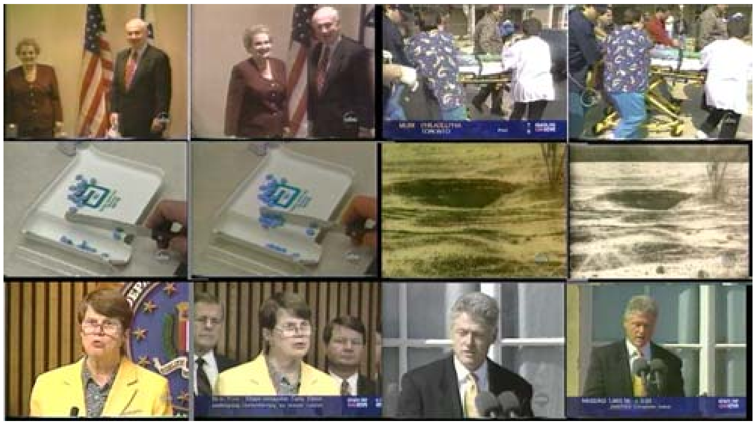


Figure 2.4. Examples of Image Near-Duplicate

for spatial locations and six for Gabor filter coefficients. Edge relational features are spatial coordinate difference. All ARGs are fully connected graphs.

IND detection therefore is realized by thresholding the ARG similarity, i.e. the probability ratio. Varying the threshold, we can obtain different recall and precision performance.

2.5.1 Implementation Issues

From the approximate likelihood equations, it is noted that the probability ratio is not invariant to the size of the input graphs. To reduce the sensitivity of the size variation, we use the normalized probability ratio $S(G_m, G_d)/(MN)$ instead of the original probability ratio $S(G_m, G_d)$.

Furthermore, in order to reduce the computational cost, we use the pruned MRF. The pruning process is carried out by discarding the MRF nodes whose corresponding 1-node potential function (i.e. $\eta_{iu}(1)$ in the Eq.(2.13)) is less than a threshold, which can be determined empirically. The lower number of MRF nodes will result in lower computational cost but degraded detection accuracy. We will discuss the trade-off between MRF pruning and performance degradation in the experiment.

Another method of saving computational cost is to reduce the Monte Carlo sample number. In general, Monte Carlo sample number has to be determined empirically. For general ARGs, such as labeled ARGs, it may be necessary to use a large number of Monte Carlo samples. However, for IND detection and retrieval, we found that the detection and retrieval performance remains excellent even if we use a small number of samples. In our experiments, we have found 40-100 samples are sufficient for achieving good results. Using a large number of samples does not gain significant performance improvement. However, in the learning stage, we suggest to use more sample numbers so that the parameters can be more accurately estimated.

In the learning stage, the *vertex copy probability* r is initialized as 0.9. And the parameters of the transformation are initialized by setting all ξ_{iu}^{k} and $\xi_{iu,jv}^{k}$ to 1.0 and invoking Eq.(2.16) to calculate the parameters. Such configuration is faster than random initialization. In general, the learning process takes 6 to 10 iterations to converge.

2.5.2 Experiments

We conduct the IND detection experiments in a public news video database, known as TREC-VID data set(year 2003)[1]. This data set is chosen because it is a widely adopted data set for visual concept (object,scene,event) detection and video search. Furthermore, there are abundant IND examples in the video database in that the same news story is often reported in different times and across different channels and the corresponding video segments often contain Image Near-Duplicates. Detecting INDs in news videos is useful for various applications, for example tracking news videos of the same topic across multiple channels [117].

TREC-VID 2003 data set consists of broadcast news videos during the period from January 1998 to June 1998. The news programs are from two major US broadcast channels: CNN and ABC. We browse through the keyframes (provided by TREC-VID) of the videos. IND image pairs for positive data set are discovered manually using the news topic ground truth from TDT2. TDT, known as Topic Detection and Tracking, is another benchmark data set that contains the groundtruth labels of the news video topics in the TREC 2003 data set. Video stories annotated with the same label are related to the same topic or event. Searching IND frames within the video stories with the same topical label can dramatically reduce the annotation time. Non-IND frames are randomly sampled from the keyframe data set.

The IND detection data set consists of 300 images (150 pairs) and 300 non-duplicate images. The training set consists of 30 IND pairs and 60 non-duplicate images, which are randomly extracted from the data set. The testing set contains the rest of the images, i.e. 120 IND pairs and 240 non-duplicate images. Therefore, there are $480 \times 479/2 - 120 = 114840$ non-IND pairs.

The experiment is intended to evaluate the effectiveness of the learning-based ARG matching algorithm for the IND detection problem. We compare the proposed method with the energy minimization (MINE) based approach, as well as other methods based on global or grid-based features, including grid-based color histogram (GCH), local edge descriptor (LED), and global color histogram (COLOR). For energy minimization, we implemented a randomized search algorithm that is able to approximately find the minimal energy of the energy function, which is the sum of distance of the vertex and edge features [42][87]. The overall computation time of MINE is a little bit longer than the ARG matching method. We did not compare with other ARG matching methods, such as that in [77], because it is difficult to implement their algorithms and there is no publicly available implementation. For the proposed graph matching algorithm, we have implemented two inference algorithms:

Monte Carlo method (GRAPH-MC) and Loopy Belief Propagation (GRAPH-LBP). We used 40 sample points for GRAPH-MC. GRAPH-LBP is a bit faster than GRAPH-MC although their performance are almost the same. For COLOR, we use the color histogram with 256 bins and HSV color space (16 bins for H, 4 bins for S, and 4 bins for V), which is a standard configuration used in content-based image retrieval. For GCH, we use the standard block partition used in image retrieval with 5×5 blocks. The histogram within each block has 32 bins (8 bins for H, 2 bins for S, 2 bins for V). For LED, we use the same method as that in [38], in which LED has been shown to outperform all other approaches in Exact Duplicate Detection (a problem similar to IND detection, except that there is no content variations of the two duplicate images). The results are shown in Figure 2.5. Table 2.1 shows the average precision of IND detection with different methods. The average precision measure is computed by sampling multiple precision-recall points (50 samples) in the ROC curves and then averaging the precision values.

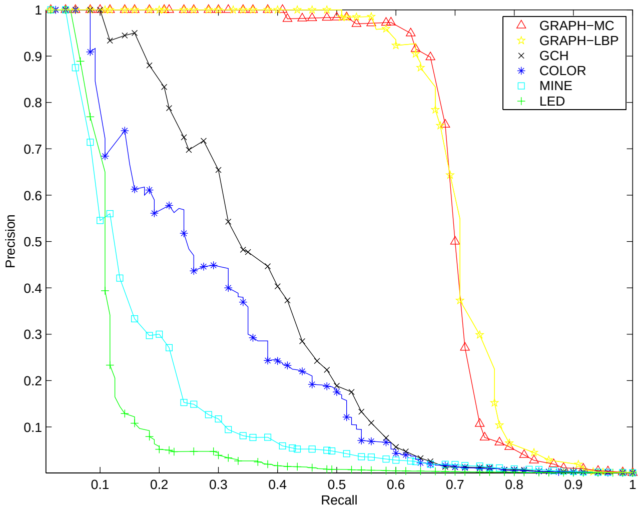


Figure 2.5. ROC curve of IND detection with different methods. (GRAPH-MC: the proposed method with Monte Carlo inference, GRAPH-LBP: the proposed method with Loopy Belief Propagation, GCH: Grid-based Color Histogram, COLOR: Color Histogram, MINE: MINimum Energy by energy minimization, LED: Local Edge Descriptor)

GRAPH-MC	GRAPH-LBP	GCH	COLOR	MINE	LED
0.69	0.7	0.351	0.267	0.158	0.103

Table 2.1. Average precision of IND detection by different methods

In practical applications, we need to determine the threshold value for IND detection. The threshold can be easily learned from training data or determined empirically. Figure 2.8-2.11 show example IND detection results while setting the threshold as 5.0. It can be observed that misses occur (Figure 2.9) when there is large camera zooming. And false alarms may happen (Figure 2.11) when the appearances and spatial structures of two images look similar. From Figure 2.10, it can be observed that our method is able to correctly reject non-IND images that may have similar color compositions.

The main limitation of our algorithm is its high computational cost. However the binary representation of vertex correspondence allows us to aggressively prune the resulting MRF so as to significantly save the computational cost. Figure 2.6 shows the IND detection performance with different MRF node number threshold to indicate the trade-off between MRF node number and the detection accuracy. Table 2.2 shows the average precision measure of IND detection under different MRF node numbers. It can be observed from the ROC curve and the average precision comparison that the degradation of accuracy is insignificant when we reduce the MRF node number from 50 to 20, while considerably reducing the computational cost as the complexity of the LBP algorithm is $O\big(N_{MRF})^2\big)$, where N_{MRF} is the node number of the MRF. However, the degradation becomes significant when we reduce the MRF node number from 20 to 10. The trade-off curves indicate that the detection accuracy is well-tolerant to the MRF node pruning. And the node pruning method is very effective for reducing computation time. Currently, when the MRF node number is set to 50, the speed of detection is about 10-20 image pairs per second.

To further speed up the algorithm, low cost algorithms such as GCH can be applied prior

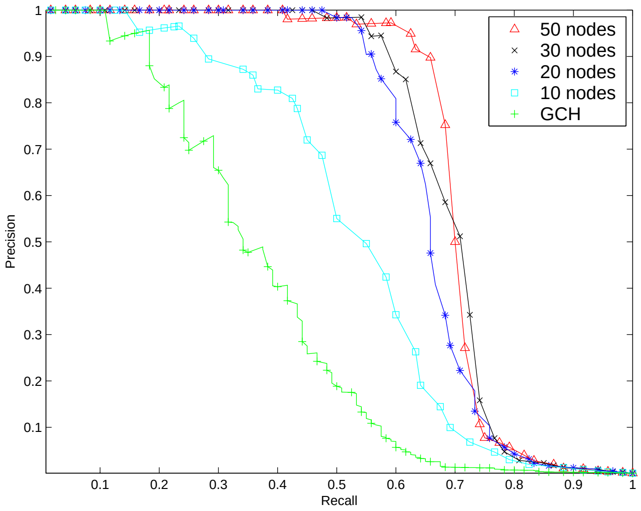


Figure 2.6. ROC curve of IND detection with different MRF node number (here, GCH is for reference)

node=50	node=30	node=20	node=10	GCH
0.69	0.676	0.652	0.517	0.351

Table 2.2. Average precision of IND detection with different MRF node number

to the ARG matching algorithm to filter out the false alarms. In fact, in a large scale database, this prefiltering procedure is often indispensable for realistic computation.

We also evaluated the IND retrieval performance. For each query image, in our test data set, there is one and only one corresponding duplicate image in the resulting rank list. Average rank of the corresponding IND in response to a query is a metric for evaluating how well the retrieval algorithm performs. Average rank is computed by averaging the ranks of the retrieved duplicated images. Specifically, if R_i is the position of the corresponding ground truth duplicate image to the query image I_i, and we have Q duplicate pairs in the test set. Then the average rank (AR) is simply calculated by

$$AR = \frac{1}{2Q}\sum_{i=1}^{2Q} R_i$$

Ideally, if the retrieval algorithm is perfect, the average rank would be 1.

Another evaluation method is to compute the empirical probability of retrieving duplicate images within certain return size, then plot the probability-scope curve. This is to measure the probability that the IND of a query image can be found in the top K returned results. Let $\mathbf{1}(\cdot)$ be an indicator function. Then the empirical probability $P(K)$ with respect to the search scope is calculated by

$$P(K) = \frac{1}{2Q}\sum_{i=1}^{2Q}\mathbf{1}(R_i \leq K)$$

Figure 2.7 shows the retrieval curve corresponding to different methods. Table 2.3 lists the Average Ranks of the different methods. The curves and Average Ranks indicate that the proposed method significantly outperforms all other approaches. It is also interesting to note that the minimum energy distance method (MINE) performs better than global or grid-based visual features when $K > 30$. It can be observed that the MINE approach performs much poorer than GCH and COLOR in IND detection, while it performs better in IND retrieval. This discrepancy indicates that IND detection and retrieval are two different problems. IND retrieval assumes that there is at least one duplicate image for the query image in the data set. While IND detection has no such assumption, and needs to reject many negative pairs in which both images do not have duplicate versions in the data set.

GRAPH-MC	GRAPH-LBP	GCH	COLOR	MINE	LED
14.41	17.14	33.25	36.00	28.35	66.10

Table 2.3. Average ranks using different methods (The smaller the better. The perfect value is 1.0

2.6 Summary

We have presented a learning-based framework for ARG vertex matching and similarity. The framework allows the ARG similarity to be learned from training data in an unsupervised manner without the need of annotating the vertex correspondence. We applied the approach

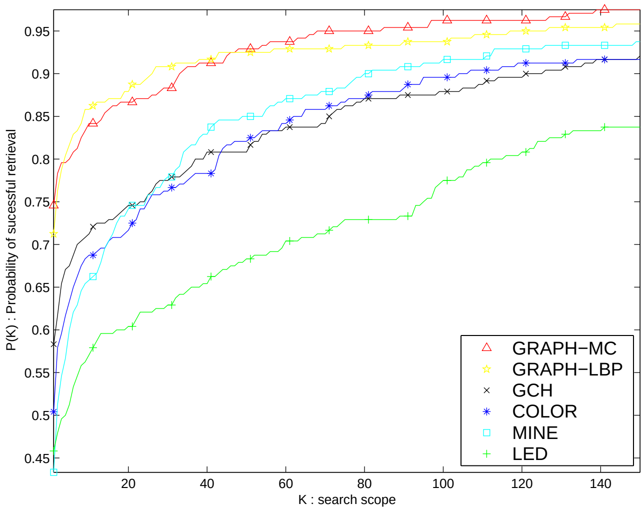


Figure 2.7. The retrieval performance with different methods.(GRAPH-MC: the proposed method with Monte Carlo inference, GRAPH-LBP: the proposed method with Loopy Belief Propagation, GCH: Grid-based Color Histogram, COLOR: Color Histogram, MINE: MINimum Energy by energy minimization, LED: Local Edge Descriptor)

to measuring image similarity and detecting Image-Near-Duplicate in a news video database. The experiments have shown dramatic performance improvement over the traditional energy minimization based methods and other visual feature based approaches. These results confirm the advantage of the presented framework for computing ARG similarity and IND detection.

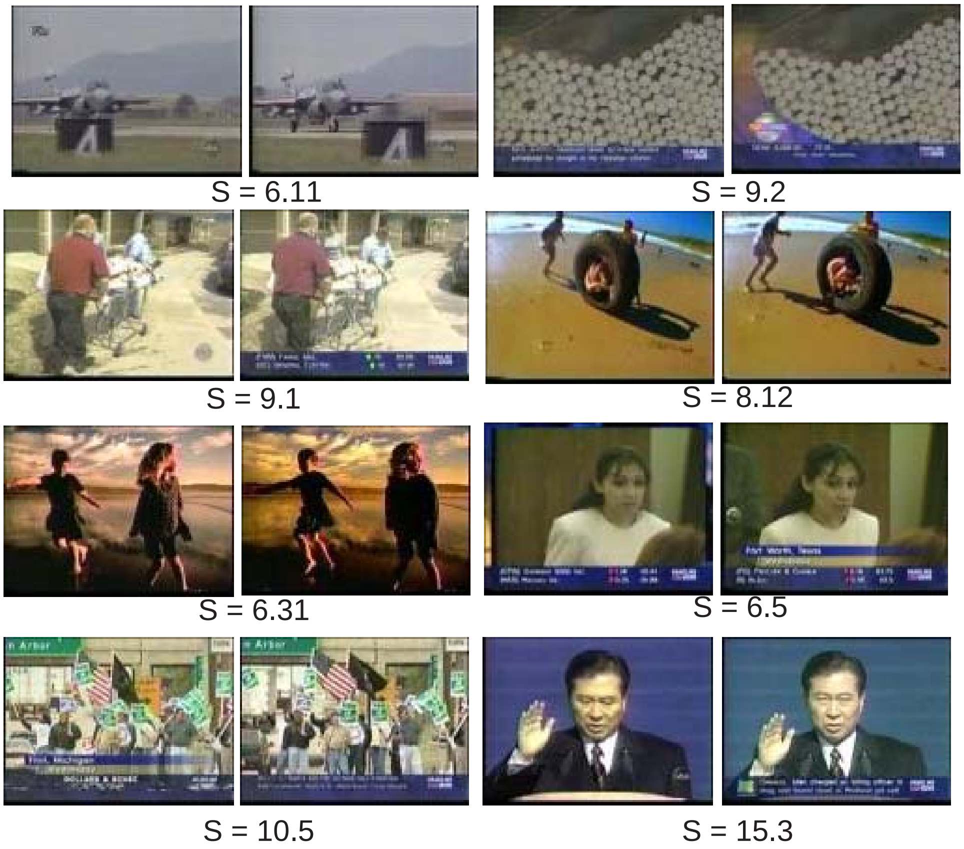


Figure 2.8. Correctly detected IND image pairs. The numbers shown below images are similarity values

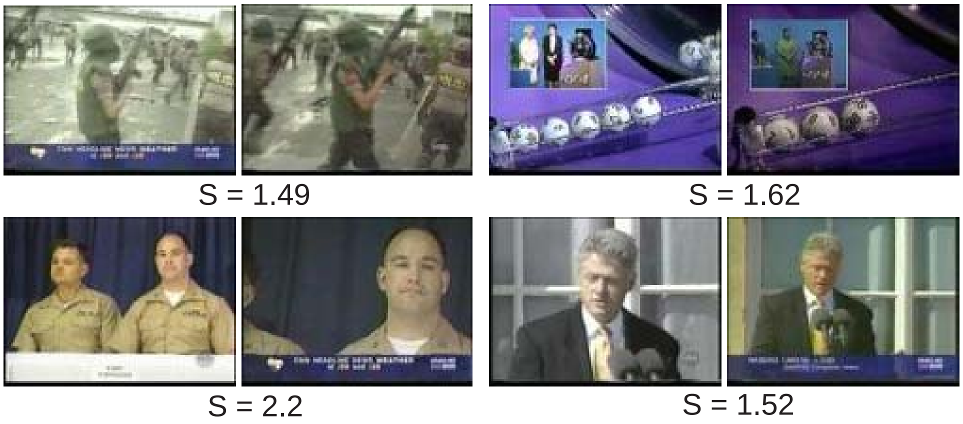


Figure 2.9. Missed IND image pairs

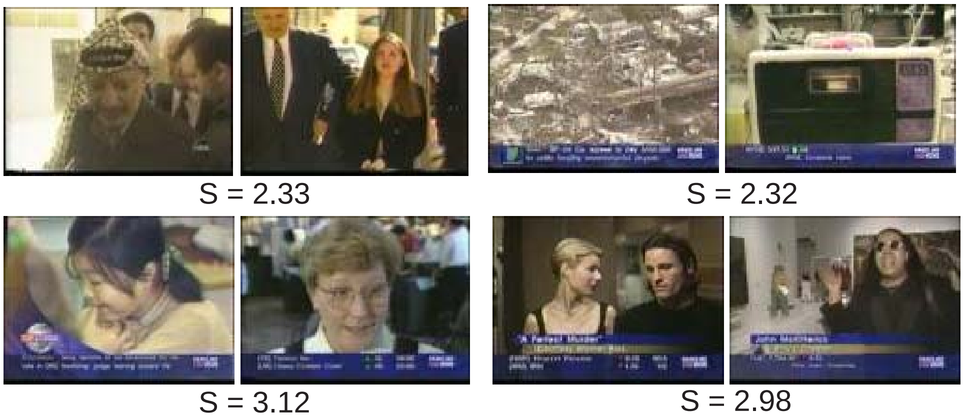


Figure 2.10. Correctly rejected non-IND image pairs

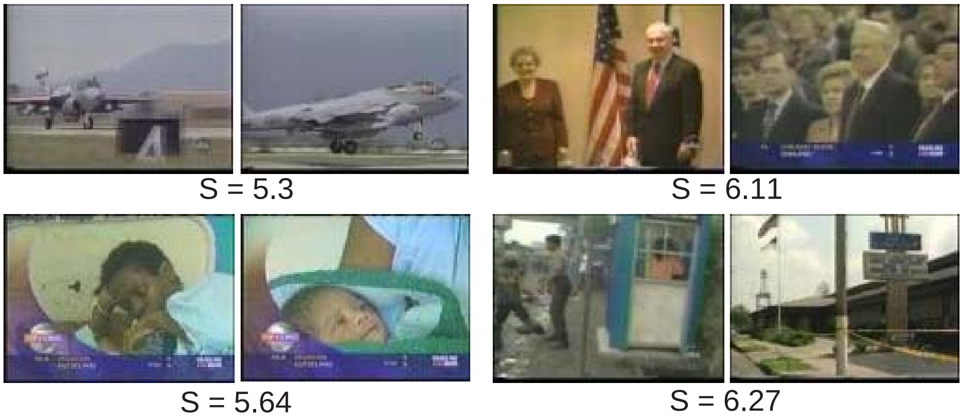


Figure 2.11. Non-IND image pairs that are falsely detected as IND pairs

Chapter 3

Learning Random Attributed Relational Graph for Part-based Object Detection

3.1 Introduction

This chapter deals with the object detection problem. Although object detection has been studied for decades, we specifically focus on the problem of generic object detection based on machine learning methods.

The generic or learning-based object detection paradigm recognizes objects in images by learning statistical object models from a corpus of training data. Among many solutions, the part-based approach represents the object model as a collection of parts with constituent attributes and inter-part relationships. Recently, combination of advanced machine learning techniques with the part-based model has shown great promise in accurate detection of a broad class of objects.

Although there has been much prior work on part-based object detection by automatic learning, there still lacks a general and formal statistical framework that is able to handle the issues of part occlusion, uncertainty of part matching, and multi-view object variations. We present our new model for part-based object detection, called Random Attributed Relational Graph (Random ARG) model. Under the Random ARG framework, an object or image, represented as Attributed Relational Graph (ARG), is considered as an instance generated from the Random ARG. We show how to compute the generation likelihood by constructing Markov Random Fields (MRF) and how to handle part occlusion by designing the MRFs. Importantly, we show that the object detection likelihood ratio is related to the partition functions of the MRFs. And the log convexity of the partition functions allows us to use variational approximations to approximate the likelihood and develop a variational E-M scheme to learn the model. This makes the object model learning more accurate and faster in convergence. We further extend the single Random ARG model to a mixture model to represent multi-view objects more accurately.

3.2 Literature Review and Our Contributions

Part-based model is a classic paradigm for object detection. The earliest part-based model may be Marr's generalized cylinder model [67] for 3D object representation. Earlier research on part-based object detection has been focused on deterministic approaches or Bayesian approaches with energy minimization. Examples include Elastic Bunch Graph for face recognition [106], pictorial structure for locating object parts [29] and finding the correspondence between the parts in image and model by Markov Random Field [61].

Statistical and learning methods for part-based object detection becomes popular only recently. The main differences between the learning-based object detection and traditional object detection paradigm include the following : (1) learning-based methods focus on general objects instead of specific object such as face. (2) The focus of the learning-based object detection is on the statistical modeling of object classes and the associated learning methods.

There are two categories of methods for part-based object detection with learning: generative approaches and discriminative approaches.

Generative approaches are based on the Bayesian classification theory, and compute the likelihood of the object under different object classes (for example “bike” and “background”). Likelihood ratio is generally used to determine whether or not an image contains the object. Generative approaches learn the positive and negative classes separately.

One of the well-known generative models is the constellation model, which is proposed and extended in [30][102]. The Constellation model models the appearance variations of object parts by Principle Component Analysis (PCA) and Gaussian probability density functions (*pdf*s), and relations among parts as a global joint Gaussian *pdf*. Unsupervised learning of the object model has been developed using a E-M like algorithm. Such unsupervised methods do not need the annotation of parts and the locations of objects). In order to increase the generalization capability and alleviate the overfitting problem, Bayesian learning algorithm has been incorporated into the model in [59]. The limitation of the constellation model is its global modeling of the part constellation. This potentially limits its power of modeling multi-view objects. Mixture model could be a solution to the multi-view object detection problem in which each component cover one view of the object. However, this solution will significantly increase the computational cost of the system.

Another type of generative model, pictorial structure, models an object as a graph-like entity (typically star-graph) in which nodes represent object parts and edges represent the relations among parts. Learning pictorial structure is realized by estimating the parameters of the Gaussian density functions using the annotated parts in the training images. The pictorial structure originally was used to locate object parts. But recently, it has been extended to deal with the object detection problem. The star-graph structure has also been extended to more general K-fan graphs [18]. Yet, the method is still limited in several aspects, including lacking the capability of unsupervised learning and modeling part occlusion. The lack of part occlusion modeling potentially limits its applications in multi-view object detection, as for objects under different views, some parts are not visible. The main advantage of the pictorial structure model against the constellation model is that part relations are modeled locally by the edges of the graph instead of as a joint Gaussian. Furthermore, translations of the objects can be implicitly handled by relative coordinates, while in the constellation model, centroid calibration has to be performed in order to achieve translation invariance.

Another category of object detection methods is the discriminative method. Unlike generative models that model the statistics and variations of the objects in an object class, discriminative approaches directly model the decision boundary between two object classes. This in theory would result in better detection performance than that using generative models since some of the assumptions in generative models may be inaccurate, for example the Gaussian *pdf* assumption. Currently, the most widely used discriminative model is the boosting method. Viola and Jones first propose boosting for face detection in [99]. Their method achieves excellent performance with real-time speed. Boosting then is extended to general object class detection by Opelt *et al.*[2]. The basic idea of boosting-based object detection is to let each part in the model a weak classifier. The decision from individual weak classifiers is computed by thresholding the distance between the part features in the model and part features in the image. The main advantage of boosting is its ability to learn and detect multi-view objects. The drawback is is inability to model the relations among parts.

More recently, there have been research efforts to combine the generative and discriminative approaches in an attempt to obtain the advantages of the two methods. For example, Holub and Perona [46] has developed Fisher kernels for the constellation model. The fisher kernel allows object detection and learning to be conducted using Support Vector Machine (SVM), a widely used discriminative classifier. Bar-Hillel *et al.* [45][44] has presented a boosting-based learning method for their generative model, which is similar to the constellation model. Combing discriminative and generative classifiers has been also explored [49][52] in the machine learning community, and been shown to be more advantageous than using the generative models alone.

We have been focusing on generative models for object detection because generative modeling offers a principled framework for incorporating human knowledge and physics-based information. Our model basically follows the pictorial structure, while we enhance it with the modeling of the topological variations of the graph. When the topological variations are taken into account, the new model resembles the conventional random graph [25], in which the existence of an edge connecting two nodes is determined by sampling a binomial random variable (indicating whether the link is present). The random graph can be further augmented by attaching general random variables to its nodes and edges rather than only the connectivity random variables. This ends up with a random graph with associated random attributes, which is called Random Attributed Relational Graph (Random ARG). The Random ARG model has the advantages of the constellation model: part occlusion modeling and unsupervised learning, but with enhancement in modeling of the part occlusion and more accurate learning algorithm by variational E-M. On the other hand, Random ARG holds the advantages of the pictorial structure model : modeling the relations among parts locally and handling the translation invariance implicitly. Such unique features make it possible to model variations of multi-view objects.

In the Random ARG model, we model an object instance as an ARG, with nodes in the ARG representing the parts in the object. An image containing the object is an instance generated from the Random ARG plus some patches generated from the background model, resulting in an ARG representation. In order to realize likelihood computation, we reduce the computation to a MRF model and we show that there is an elegant mathematical relationship between the object detection likelihood ratio and the partition functions of the MRF. This discovery enables the use of variational inference methods, such as Loopy Belief Propagation or Belief Optimization, to estimate the part matching probability and learn the parameters by variational E-M.

We compare our proposed Random ARG model with the constellation model developed in [30], which also provides a publicly available benchmark data set . Our approach achieves a significant improvement in learning convergence speed (measured by the number of iterations and the total learning time) with comparable detection accuracy. The learning speed is improved by more than two times if we use a combined scheme of Gibbs Sampling and Belief Optimization, and more than five times if we use Loopy Belief Propagation. The improved efficiency is important in practical applications, as it allows us to rapidly deploy the method to learning general object classes as well as detection of objects with view variations.

We extend the Random ARG model to a Mixture of Random ARG (MOR) model to capture the structural and appearance variations of the objects with different views of the same object class. Through a semi-supervised learning scheme, the MOR model is shown to improve the detection performance against the single Random ARG model for detecting objects with continuous view variations in a data set consisting of images downloaded from the web. The data set constructed by us can be used as a public benchmark for multi-view object detection.

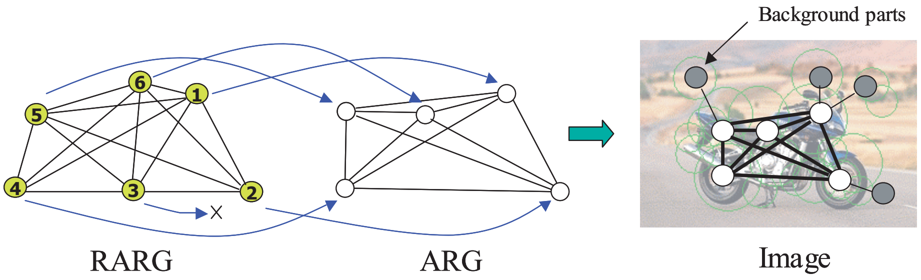


Figure 3.1. A generative process that generates the part-based representation of an image

3.3 The Random Attributed Relational Graph Model

The representation method used by us follows the previous work in [42][62][24], where an object instance or image is represented as an Attributed Relational Graph [42], formally defined as

Definition 3.1. *An Attributed Relational Graph(ARG) is defined as a triplet $O = (V, E, Y)$, where V is the vertex set, E is the edge set, and Y is the attribute set that contains attribute y_u attached to each node $n_u \in V$, and attribute y_{uv} attached to each edge $e_w = (n_u, n_v) \in E$.*

For an object instance, a node in the ARG corresponds to one part in the object. attributes y_u and y_{uv} represent the appearances of the parts and relations among the parts. For an object model, we use a graph based representation similar to the ARG but attach random variables to the nodes and edges of the graph, formally defined as a Random Attributed Relational Graph

Definition 3.2. *A Random Attributed Relational Graph (Random ARG) is defined as a quadruple $R = (V, E, A, T)$, where V is the vertex set, E is the edge set, A is a set of random variables consisting of A_i attached to the node $n_i \in V$ with pdf $f_i(.)$, and A_{ij} attached to the edge $e_k = (n_i, n_j) \in E$ with pdf $f_{ij}(.)$. T is a set of binary random variables, with T_i attached to each node (modeling the presense/absence of nodes).*

$f_i(.)$ is used to capture the statistics of the part appearance. $f_{ij}(.)$ is used to capture the statistics of the part relation. T_i is used to handle part occlusion. $r_i = p(T_i = 1)$ is referred to as the *occurrence probability* of the part i in the object model. An ARG hence can be considered as an instance generated from Random ARG by multiple steps: first draw samples from $\{T_i\}$ to determine the topology of the ARG, then draw samples from A_i and A_{ij} to obtain the attributes of the ARG and thus the appearance of the object instance. In our current system, both Random ARG and ARG are fully connected. However, in more general cases, we can also accommodate edge connection variations by attaching binary random variables T_{ij} to the edges, where $T_{ij} = 1$ indicates that there is an edge connecting the node i and node j, $T_{ij} = 0$ otherwise.

3.3.1 Bayesian Classification under Random ARG Framework

Conventionally, object detection is formulated as a binary classification problem with two hypotheses: $H = 1$ indicates that the image contains the target object (e.g. bike), $H = 0$ otherwise. Let O denote the ARG representation of the input image. Object detection problem therefore is reduced to the following likelihood ratio test

$$\frac{p(O|H=1)}{p(O|H=0)} > \frac{p(H=0)}{p(H=1)} = \lambda \tag{3.1}$$

Where λ is used to adjust the precision and recall performance. The main problem is thus to compute the positive likelihood $p(O|H = 1)$ and the negative likelihood $p(O|H = 0)$. $p(O|H = 0)$ is the likelihood assuming the image is a *background* image without the target object. Due to the diversity of the *background* images, we adopt a simple decomposable *i.i.d.* model for the background parts. We factorize the negative likelihood as

$$p(O|H=0) = \prod_u p(y_u|H=0) \prod_{uv} p(y_{uv}|H=0) = \prod_u f^-_{B_1}(y_u) \prod_{uv} f^-_{B_2}(y_{uv}) \tag{3.2}$$

where $f^-_{B_1}(\cdot)$ and $f^-_{B_2}(\cdot)$ are *pdf*s to capture the statistics of the appearance and relations of the parts in the *background* images, referred to as background *pdf*s. The minus superscript

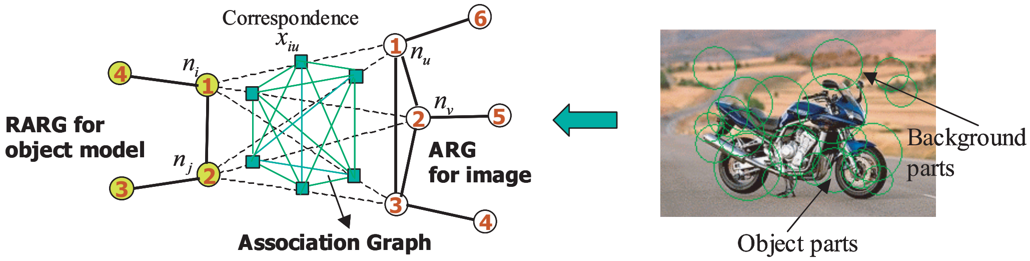


Figure 3.2. ARG, Random ARG and the Association Graph. Circles in the image are detected parts

indicates that the parameters of the *pdf*s are learned from the negative data set. To compute the positive likelihood $p(O|H=1)$, we assume that an image is generated by the following generative process (Figure 3.1): an ARG is first generated from the Random ARG, additional patches, whose attributes are sampled from the background *pdf*s, are independently added to form the final part-based representation O of the image. In order to compute the positive likelihood, we further introduce a variable X to denote the correspondences between parts in the ARG O and parts in the Random ARG R. Treating X as a hidden variable, we have

$$p(O|H=1) = \sum_X p(O|X, H=1)p(X|H=1) \tag{3.3}$$

Where X consists of a set of binary variables, with $x_{iu} = 1$ if the part i in the object model corresponds to the part u in the image, $x_{iu} = 0$ otherwise. If we assign each x_{iu} a node, then these nodes form an Association Graph as shown in Figure 3.2. The Association Graph can be used to define an undirected graphical model (Markov Random Field) for computing the positive likelihood in Eq. (3.3). In the rest of the paper, iu therefore is used to denote the index of the nodes in the Association Graph. A notable difference between our method and the previous methods [30][61] is that we use a binary random representation for the part correspondence. Such representation is important as it allows us to prune the MRF by discarding nodes associated with a pair of dissimilar parts to speed up part matching, and readily apply efficient inference techniques such as Belief Optimization[100][103].

3.3.2 Mapping the Random ARG parameters to the Association Graph MRF

The factorization in Eq. (3.3) requires computing two components $p(X|H=1)$ and $p(O|X,H=1)$. This section describes how to map the Random ARG parameters to these two terms as well as construct MRFs to compute the likelihood ratio.

The computational method developed in this section is similar to that in the chapter 2. The main difference is that we model the transformation between two ARGs in chapter 2, but in this chapter we model the "object model" itself instead of the transformation. In the Random ARG model each vertex is associated with a set of parameters, while in the similarity framework, there is only one set of parameters to specify the transformation. Consequently, the learning algorithms for these two problems are distinct.

First, $p(X|H=1)$, the prior probability of the correspondence, is designed so as to satisfy the one-to-one part matching constraint, namely,one part in the object model can only be matched to one part in the image, vice versa. Furthermore, $p(X|H=1)$ is also used to encode the *occurrence probability* r_i. To achieve these, $p(X|H=1)$ is designed as a binary pairwise MRF with the following Gibbs distribution

$$p(X|H=1) = \frac{1}{Z} \prod_{iu,jv} \psi_{iu,jv}(x_{iu}, x_{jv}) \prod_{iu} \phi_{iu}(x_{iu}) \tag{3.4}$$

Where Z is the normalization constant, a.k.a the partition function. $\psi_{iu,jv}(x_{iu}, x_{jv})$ is the two-node potential function defined as

$$\psi_{iu,jv}(1,1) = \varepsilon, \quad for \quad i = j \;\; or \;\; u = v; \qquad \psi_{iu,jv}(x_{iu}, x_{jv}) = 1, \;\; otherwise \tag{3.5}$$

where ε is set to 0 (for Gibbs Sampling) or a small positive number (for Loopy Belief Propagation). Therefore, if the part matching violates one-to-one constraint, the prior probability would drop to zero (or near zero). $\phi_{iu}(x_{iu})$ is the one-node potential function. Adjusting $\phi_{iu}(x_{iu})$ affects the distribution $p(X|H=1)$, therefore it is related to the *occurrence probability* r_i. By designing $\phi_{iu}(x_{iu})$ to different values, we will result in different r_i. For any iu, we have two parameters to specify for $\phi_{iu}(.)$, namely $\phi_{iu}(1)$ and $\phi_{iu}(0)$. Yet, it is not difficult to show that for any iu, different $\phi_{iu}(1)$ and $\phi_{iu}(0)$ with the same ratio $\phi_{iu}(1)/\phi_{iu}(0)$ would result in the same distribution $p(X|H = 1)$ (but different partition function Z). Therefore, we can just let $\phi_{iu}(0) = 1$ and $\phi_{iu}(0) = z_i$. Note here that z_i only has the single indice i. meaning the potential function for the correspondence variable between part i in the model and part u in the image does not depend on the index u. Such design is for simplicity and the following relationship between z_i and r_i.

Lemma 3.1. *r_i and z_i is related by the following equation:*

$$r_i = z_i \frac{\partial \mathrm{ln} Z}{\partial z_i}$$

where Z is the partition function defined in Eq. (3.4).

The above lemma leads to a simple formula to learn the *occurrence probability* r_i (section 2.4). However, lemma 1 still does not provide a closed-form solution for computing z_i given r_i. We resort to an approximate solution, through the following lemma.

Lemma 3.2. *The log partition function satisfy the inequality*

$$\mathrm{ln} Z \leq \sum_{i=1}^{N} \mathrm{ln}(1 + M z_i)$$

and the equality holds when N/M tends to zero (N and M are the numbers of parts in the object model and image respectively). For the pruned MRF, the upper bound is changed to

$$\mathrm{ln} Z \leq \sum_{i=1}^{N} \mathrm{ln}(1 + d_i z_i)$$

where d_i is the number of the nodes in the ARG that could possibly correspond to the node i in the Random ARG after pruning the Association Graph.

Since the closed form solution for mapping r_i to z_i is unavailable, we use the upper bound as an approximation. Consequently, combining lemmas 3.1 and 3.2 we can obtain the following relationship for the pruned MRF. $z_i = r_i/((1-r_i)d_i)$.

The next step is to derive the conditional density $p(O|X, H=1)$. Assuming that y_u and y_{uv} are independent given the correspondence, we have

$$p(O|X, H=1) = \prod_{uv} p(y_{uv}|x_{1u}, x_{1v}, ..., x_{Nu}, x_{Nv}, H=1) \prod_{u} p(y_u|x_{1u}, ..., x_{Nu}, H=1)$$

Furthermore, y_u and y_{uv} should only depends on the Random ARG nodes that are matched to u and v. Thus

$$\begin{aligned} p(y_u|x_{11}=0, ..., x_{iu}=1, ..., x_{NM}=0, H=1) &= f_i(y_u) \\ p(y_{uv}|x_{11}=0, ..., x_{iu}=1, x_{jv}=1, ..., x_{NM}=0, H=1) &= f_{ij}(y_{uv}) \end{aligned} \quad (3.6)$$

Also, if there is no node in the Random ARG matched to u, then y_u,y_{uv} should be sampled from the background *pdf*s, i.e.

$$\begin{aligned} p(y_u|x_{11}=0, x_{iu}=0, ..., x_{NM}=0, H=1) &= f^{+}_{B_1}(y_u) \\ p(y_{uv}|x_{11}=0, x_{iu}=0, ..., x_{NM}=0, H=1) &= f^{+}_{B_2}(y_{uv}) \end{aligned} \quad (3.7)$$

where $f^{+}_{B_1}(\cdot)$ and $f^{+}_{B_2}(\cdot)$ is the background *pdf* trained from the positive data set. Note that here we use two sets of background *pdf*s to capture the difference of the background statistics in the positive data set and that in the negative data set.

Combining all these elements together, we would end up with another MRF (to be described in theorem 3.1). It is important and interesting to note that the likelihood ratio for object detection is actually related to the partition functions of the MRFs through the following relationship.

Theorem 3.1. *The likelihood ratio is related to the partition functions of MRFs as the following*

$$\frac{p(O|H=1)}{p(O|H=0)} = \sigma \frac{Z'}{Z} \tag{3.8}$$

where Z is the partition function of the Gibbs distribution $p(X|H=1)$. Z' is the partition function of the Gibbs distribution of a new MRF, which happens to be the posterior probability of correspondence $p(X|O,H=1)$, with the following form

$$p(X|O,H=1) = \frac{1}{Z'} \prod_{iu,jv} \varsigma_{iu,jv}(x_{iu}, x_{jv}) \prod_{iu} \eta_{iu}(x_{iu}) \tag{3.9}$$

where the one-node and two-node potential functions have the following forms

$$\eta_{iu}(1) = z_i f_i(y_u)/f^{+}_{B_1}(y_u); \quad \varsigma_{iu,jv}(1,1) = \psi_{iu,jv}(1,1) f_{ij}(y_{uv})/f^{+}_{B_2}(y_{uv}) \tag{3.10}$$

all other values of the potential functions are set to 1 (e.g. $\eta_{iu}(x_{iu} = 0) = 1$). σ is a correction term

$$\sigma = \prod_u f^{+}_{B_1}(y_u)/f^{-}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}(y_{uv})/f^{-}_{B_2}(y_{uv})$$

3.3.3 Computing the Partition Functions

Theorem 3.1 reduces the likelihood ratio calculation to the computation of the partition functions. For the partition function Z, it has a closed form(See Appendix B.1) and can be computed in a polynomial time or using the lemma 2.2 for approximation. The main difficulty is to compute the partition function Z', which involves a summation over all possible correspondences, whose size is exponential in MN. Fortunately, computing the partition function of the MRF has been studied in statistical physics and machine learning [100]. It turns out that, due to its convexity, $\ln Z'$ can be written as a dual function, a.k.a. variational representation, or in the form of the Jensen's inequality [111].

$$\ln Z' \geq \sum_{(iu,jv)} \hat{q}(x_{iu}, x_{jv}) \ln \varsigma_{iu,jv}(x_{iu}, x_{jv}) + \sum_{(iu)} \hat{q}(x_{iu}) \ln \eta_{iu}(x_{iu}) + \mathcal{H}(\hat{q}(X)) \tag{3.11}$$

Where $\hat{q}(x_{iu})$ and $\hat{q}(x_{iu}, x_{jv})$ are known as one-node and two-node beliefs, which are the approximated marginal of the Gibbs distribution $p(X|O, H = 1)$. $\mathcal{H}(\hat{q}(X))$ is the approximated entropy, which can be approximated by Bethe approximation[111], as below

$$\mathcal{H}(\hat{q}(X))\approx -\sum_{iu,jv}\sum_{x_{iu},x_{jv}} \hat{q}(x_{iu}, x_{jv}) \ln \hat{q}(x_{iu}, x_{jv}) + \sum_{iu}(MN-2)\sum_{x_{iu}}\hat{q}(x_{iu}) \ln(\hat{q}(x_{iu}))$$

Apart from Bethe approximation, it is also possible to use more accurate approximations, such as semidefinite relaxation in [100].

The RHS in the Eq. (3.11) serves two purposes, for variational learning and for approximating $\ln Z'$. In both cases, we have to calculate the approximated marginal $\hat{q}(x_{iu})$ and $\hat{q}(x_{iu}, x_{jv})$. There are two options to approximate it, optimization-based approach and Monte Carlo method. The former maximizes the lower bound with respect to the approximated marginal. For example, Loopy Belief Propagation (LBP) is an approach to maximizing the lower bound through fixed point equations[111]. However, we found that, LBP message passing often does not converge using the potential functions in Eq.(3.10). Nonetheless, we found that if we select the marginal that corresponds to the larger lower bound in Eq.(3.11) across update iterations, we can achieve satisfactory inference results and reasonably accurate object models.

Another methodology, Monte Carlo sampling[5] , approximates the marginal by drawing samples and summing over the obtained samples. Gibbs Sampling, a type of Monte Carlo method, is used in our system due to its efficiency. In order to reduce the variances of the approximated two-node beliefs, we propose a new method to combine the Gibbs Sampling with the Belief Optimization developed in [103], which proves that there is a closed-form solution (through Bethe approximation) for computing the two-node beliefs given the one-node beliefs and the two-node potential functions (Lemma 3.1 in [103]). We refer to this approach as Gibbs Sampling plus Belief Optimization (GS+BO).

3.3.4 Learning Random Attributed Relational Graph

We use Gaussian models for all the *pdf*s associated with the Random ARG and the background model . Therefore, we need to learn the corresponding Gaussian parameters μ_i,Σ_i,μ_{ij},Σ_{ij};$\mu^+_{B_1}$,$\Sigma^+_{B_1}$,$\mu^+_{B_2}$,$\Sigma^+_{B_2}$;$\mu^-_{B_1}$,$\Sigma^-_{B_1}$,$\mu^-_{B_2}$,$\Sigma^-_{B_2}$. and the *occurrence probability* r_i.

Learning the Random ARG is realized by Maximum Likelihood Estimation (MLE). Directly maximizing the positive likelihood with respect to the parameters is intractable, instead we maximize the lower bound of the positive likelihood through Eq.(3.11), resulting in a method known as Variable Expectation-Maximization (Variational E-M). **Variational E-Step:** Perform GS+BO scheme or Loopy Belief Propagation to obtain the one-node and two-node beliefs.

M-Step: Maximize the overall log-likelihood with respect to the parameters

$$L = \sum_{k=1}^{K} \ln p(O_k|H=1) \tag{3.12}$$

where K is the number of the positive training instances. Since direct maximization is intractable, we use the lower bound approximation in Eq.(3.11), resulting in the following equations for computing the parameters

$$\begin{array}{ll}
\xi^k_{iu} = \hat{q}(x^k_{iu}=1), \;\; \xi^k_{iu,jv} = \hat{q}(x^k_{iu}=1, x^k_{jv}=1); & \bar{\xi}^k_{iu} = 1-\xi^k_{iu}, \;\; \bar{\xi}^k_{iu,jv} = 1-\xi^k_{iu,jv} \\
\mu_i = \dfrac{\sum_k \sum_u \xi^k_{iu} y^k_u}{\sum_k \sum_u \xi^k_{iu}} & \Sigma_i = \dfrac{\sum_k \sum_u \xi^k_{iu}(y^k_u - \mu_i)(y^k_u - \mu_i)^T}{\sum_k \sum_u \xi^k_{iu}} \\
\mu_{ij} = \dfrac{\sum_k \sum_{uv} \xi^k_{iu,jv} y^k_{uv}}{\sum_k \sum_{uv} \xi^k_{iu,jv}} & \Sigma_{ij} = \dfrac{\sum_k \sum_{uv} \xi^k_{iu,jv}(y^k_{uv} - \mu_{ij})(y^k_{uv} - \mu_{ij})^T}{\sum_k \sum_{uv} \xi^k_{iu,jv}} \\
\mu^+_{B_1} = \dfrac{\sum_k \sum_u \bar{\xi}^k_{iu} y^k_u}{\sum_k \sum_u \bar{\xi}^k_{iu}} & \Sigma^+_{B_1} = \dfrac{\sum_k \sum_u \bar{\xi}^k_{iu}(y^k_u - \mu_i)(y^k_u - \mu_i)^T}{\sum_k \sum_u \bar{\xi}^k_{iu}} \\
\mu^+_{B_2} = \dfrac{\sum_k \sum_{uv} \bar{\xi}^k_{iu,jv} y^k_{uv}}{\sum_k \sum_{uv} \bar{\xi}^k_{iu,jv}} & \Sigma^+_{B_2} = \dfrac{\sum_k \sum_{uv} \bar{\xi}^k_{iu,jv}(y^k_{uv} - \mu_{ij})(y^k_{uv} - \mu_{ij})^T}{\sum_k \sum_{uv} \bar{\xi}^k_{iu,jv}}
\end{array} \tag{3.13}$$

The *occurrence probability* r_i is derived from Lemma 3.1 using maximum likelihood estimation.Using the lower bound approximation in Eq.(3.11), we have the approximated overall

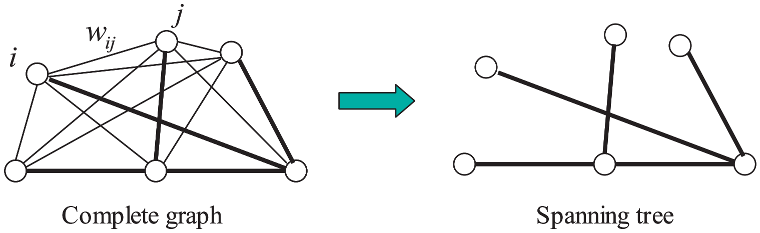


Figure 3.3. Spanning tree approximation, realized by first constructing a weighted graph (having the same topology as the Random ARG) with the weight $w_{ij} = |\Sigma_{ij}|$ in the edge $e_l = (n_i, n_j)$, then invoking the conventional minimum spanning tree(MST) algorithm, such as Kruskal's algorithm. Here $|\Sigma_{ij}|$ is the determinant of the covariance matrix of $f_{ij}(.)$ (the *pdf* of the relational feature in the Random ARG) associated with the edge $e_l = (n_i, n_j)$.

log-likelihood

$$L \approx \sum_{k=1}^{K} \sum_{iu} \hat{q}(x_{iu}^k = 1) \ln z_i - K \ln Z(N; M; z_1, z_2, ..., z_N) + \alpha \tag{3.14}$$

where α is a term independent on the *occurrence probability* $r_1, r_2, ..., r_N$. To minimize the approximated likelihood with respect to z_i, we compute the derivative of the Eq.(3.14), and equates it to zero

$$\begin{aligned} &\frac{\partial}{\partial z_i}\Big[\sum_{k=1}^{K} \sum_{iu} \hat{q}(x_{iu}^k = 1) \ln z_i\Big] - K \frac{\partial}{\partial z_i} \ln Z(N; M; z_1, z_2, ..., z_N) \\ &= \sum_{k=1}^{K} \sum_{iu} \hat{q}(x_{iu}^k = 1) \frac{1}{z_i} - K \frac{r_i}{z_i} = 0 \end{aligned} \tag{3.15}$$

We used Lemma 3.1 in the last step. Since $z_i \neq 0$, the above equation leads to the equation for estimating r_i

$$r_i = \frac{1}{K} \sum_{k} \sum_{u} \hat{q}(x_{iu}^k = 1) \tag{3.16}$$

For the background parameters $\mu_{B_1}^-$,$\Sigma_{B_1}^-$,$\mu_{B_2}^-$,$\Sigma_{B_2}^-$, the maximum likelihood estimation results in the sample mean and covariance matrix of the part attributes and relations of the images in the negative data set.

3.3.5 Spanning Tree Approximation for Spatial Relational Features

Our approaches described so far assume the relational features y_{ij} are independent. However, this may not be true in general. For example, if we let y_{ij} be coordinate differences, they are no longer independent. This can be easily seen by considering three edges of a triangle formed by any three parts. The coordinate difference of the third edge is determined by the other two edges. The independence assumption therefore is not accurate. To deal with this problem, we prune the fully-connected Random ARG into a tree by the spanning tree approximation algorithm, which discards the edges that have high determinant values in their covariance matrix of the Gaussian functions (Figure 3.3). This assumes that high determinant values of the covariance matrix indicate the spatial relation has large variation and thus is less salient.

In our experiments, we actually found a combined use of fully-connected Random ARG and the pruned spanning tree is most beneficial in terms of the learning speed and model accuracy. Specifically, we use a two-stage procedure: the fully-connected Random ARG is used to learn an initial model, which then is used to initialize the model in the 2*nd*-phase iterative learning process based on the pruned tree model. In the detection phase, only spanning-tree approximation is used.

3.4 Extension to Multi-view Mixture Model

The above described model assumes the training object instances have consistent single views. In order to capture the characteristic of an object class with view variations. We develop a Mixture of Random ARG (MOR) model, which allows the components in the MOR to capture the characteristic of the objects with different views.

Let R_t denotes the Random ARG to represent a distinct view t. The object model thereby is represented as $\Re = \{R_t\}$ along with the mixture coefficients $p(R_t|\Re)$. The positive likelihood then becomes

$$p(O|H = 1) = \sum_t p(O|R_t)p(R_t|\Re)$$

The maximum likelihood learning scheme to learn the mixture coefficients and the Gaussian *pdf* parameters therefore is similar to that of the Gaussian Mixture Model (GMM), consisting of the following E-M updates

E-step: Compute the assignment probability

$$\zeta_k^t = p(R_t|O_k, \Re) = \frac{p(O_k|R_t)p(R_t|\Re)}{\sum_t p(O_k|R_t)p(R_t|\Re)} \tag{3.17}$$

M-step: Compute the mixture coefficients

$$p(R_t|\Re) = \frac{1}{N}\sum_k \zeta_k^t \tag{3.18}$$

and update the Gaussian parameters for each component t(We omit the index t except for ζ_k^t for brevity):

$$\xi_{iu}^k = \hat{q}(x_{iu}^k = 1), \quad \xi_{iu,jv}^k = \hat{q}(x_{iu}^k = 1, x_{jv}^k = 1); \quad \bar{\xi}_{iu}^k = 1 - \xi_{iu}^k, \quad \bar{\xi}_{iu,jv}^k = 1 - \xi_{iu,jv}^k$$

$$\mu_i = \frac{\sum_k \zeta_k^t \sum_u \xi_{iu}^k y_u^k}{\sum_k \zeta_k^t \sum_u \xi_{iu}^k}, \qquad \Sigma_i = \frac{\sum_k \zeta_k^t \sum_u \xi_{iu}^k (y_u^k - \mu_i)(y_u^k - \mu_i)^T}{\sum_k \zeta_k^t \sum_u \xi_{iu}^k}$$

$$\mu_{ij} = \frac{\sum_k \zeta_k^t \sum_{uv} \xi_{iu,jv}^k y_{uv}^k}{\sum_k \zeta_k^t \sum_{uv} \xi_{iu,jv}^k}, \qquad \Sigma_{ij} = \frac{\sum_k \zeta_k^t \sum_{uv} \xi_{iu,jv}^k (y_{uv}^k - \mu_{ij})(y_{uv}^k - \mu_{ij})^T}{\sum_k \zeta_k^t \sum_{uv} \xi_{iu,jv}^k}$$

$$\mu_{B_1}^+ = \frac{\sum_k \zeta_k^t \sum_u \bar{\xi}_{iu}^k y_u^k}{\sum_k \zeta_k^t \sum_u \bar{\xi}_{iu}^k}, \qquad \Sigma_{B_1}^+ = \frac{\sum_k \zeta_k^t \sum_u \bar{\xi}_{iu}^k (y_u^k - \mu_i)(y_u^k - \mu_i)^T}{\sum_k \zeta_k^t \sum_u \bar{\xi}_{iu}^k}$$

$$\mu_{B_2}^+ = \frac{\sum_k \zeta_k^t \sum_{uv} \bar{\xi}_{iu,jv}^k y_{uv}^k}{\sum_k \zeta_k^t \sum_{uv} \bar{\xi}_{iu,jv}^k}, \qquad \Sigma_{B_2}^+ = \frac{\sum_k \zeta_k^t \sum_{uv} \bar{\xi}_{iu,jv}^k (y_{uv}^k - \mu_{ij})(y_{uv}^k - \mu_{ij})^T}{\sum_k \zeta_k^t \sum_{uv} \bar{\xi}_{iu,jv}^k}$$

The above equations are supposed to automatically discover the views of the training object instances through Eq.(3.17). However, our experiments show that directly using the E-M updates often results in inaccurate parameters of Random ARG components. This is because in the initial stage of learning, the parameters of each Random ARG component is often inaccurate, leading to inaccurate assignment probabilities (i.e. inaccurate view assignment). To overcome this problem, we can use a semi-supervised approach. First, the parameters of the Random ARG components are initially learned using view annotation data (view labels associated to the training images by annotators). Mathematically, this can be realized by fixing the assignment probabilities using the view labels during the E-M updates. For instance, if an instance k is annotated as view t, then we let $\zeta_k^t = 1$. After the initial learning process converges, we use the view update equation in Eq.(3.17) to continue the E-M iterations to refine the initially learned parameters. Such a two-stage procedure ensures that the parameters of a Random ARG component can be learned from the object instances with the view corresponding to the correct Random ARG component in the beginning of the learning process.

3.5 Experiments

We compare the performance of our system with the system using the constellation model presented in [30]. We use the same data set, which consists of four object classes - *motorbikes*, *cars*, *faces*, and *airplanes*, and a common *background* class. Each of the classes and the *background* class is randomly partitioned into training and testing sets of equal sizes. All images are resized to have a width of 256 pixels and converted to gray-scale images. Image patches are detected by Kadir's Salient Region Detector[54] with the same parameter across all four classes. Twenty patches with top saliency values are extracted for each image. Each extracted patch is normalized to the same size of 25×25 pixels and converted to a 15-dimensional PCA coefficient vectors, where PCA parameters are trained from the image patches in the positive data set. Overall, the feature vector at each node of the ARG is of 18 dimensions: two for spatial coordinates, fifteen for PCA coefficients, and one for the scale

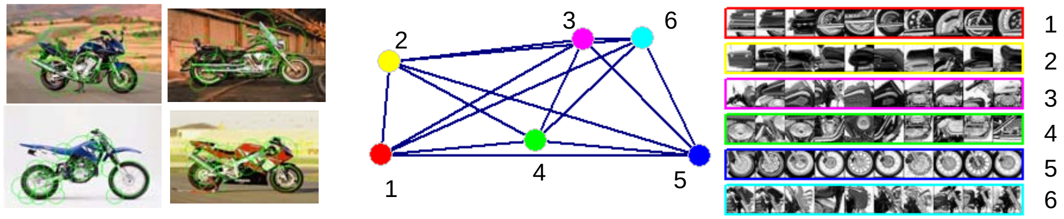


Figure 3.4. The Random ARG learned from the “motorbike” images.

feature, which is an output from the part detector to indicate the scale of the extracted part. Feature vectors at the edges are the coordinate differences.

To keep the model size consistent with that in [30], we set the number of nodes in Random ARG to be six, which gives a good balance between detection accuracy and efficiency. The maximum size of the Association Graph therefore is 120 (6x20). But for efficiency, the Association Graph is pruned to 40 nodes based on the pruning criteria described in Section 2.1. In the learning process, we tried both inference schemes, i.e. GS+BO and LBP. But, in detection we only use GS+BO scheme because it is found to be more accurate. In LBP learning, relational features are not used because it is empirically found to result in lower performance. In GS+BO scheme, the sampling number is set to be proportional to the state space dimension, namely $\alpha \cdot 2 \cdot 40$ (α is set to 40 empirically). The *occurrence probability* r_i is computed only after the learning process converges because during the initial stages of learning, r_i is so small that it affects the convergence speed and final model accuracy. Besides, we also explore different ways of applying the background models in detection. We found a slight performance improvement by replacing B^+ with B^- in the detection step(Eq. (3.8)). Such an approach is adopted in our final implementation. Figure 3.4 shows the learned part-based model for object class “motorbike” and the image patches matched to each node. Table 1(next page) lists the object detection accuracy, measured by *equal error rate* (definition is in[30]), and the learning efficiency.

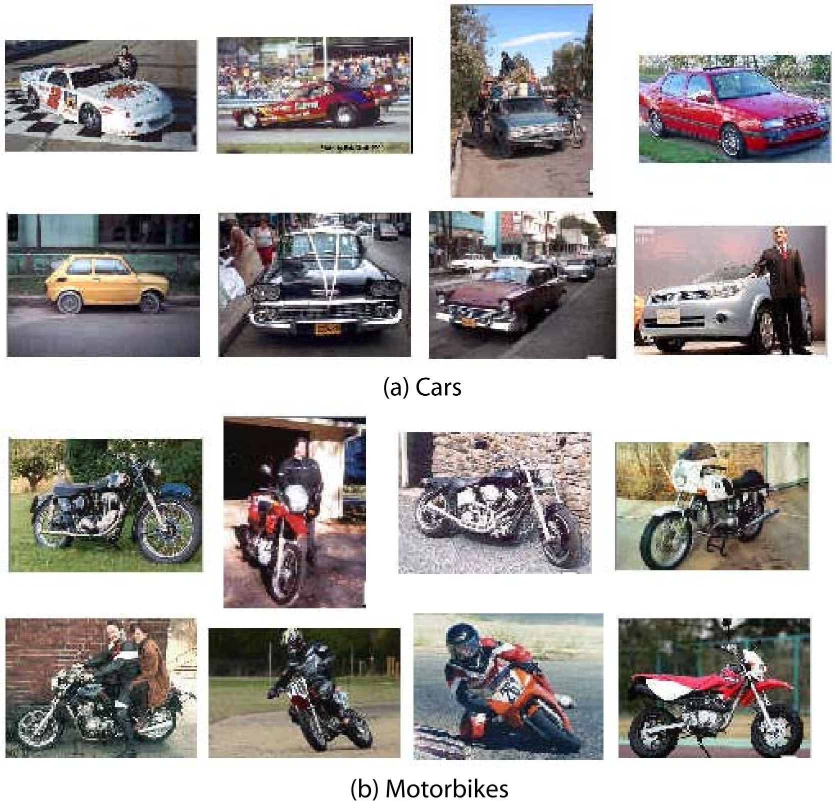
Dataset	GS+BO	LBP	Oxford
Motorbikes	91.2%	88.9%	92.5%
Faces	94.7%	92.4%	96.4%
Airplanes	90.5%	90.1%	90.2%
Cars(rear)	92.6%*	93.4%	90.3%

Dataset	GS+BO	LBP	Oxford
Motorbikes	23i/18h	28i/6h	24-36h
Faces	16i/8h	20i/4h	
Airplanes	16i/16 h	18i/8h	40-100i
Cars(rear)	16i/14h	20i/8h	

Table 3.1. Object detection performance and learning time of different methods (xi/yh means x iterations and y hours). * Background images are road images the same as [30].

The most significant performance impact by our method is the improvement in learning speed - two times faster than the well-known method [30] if we use GS+BO for learning; five times speedup if we use LBP learning. Even with the large speedup, our method still achieves very high accuracy, close to those reported in [30]. The slightly lower performance for the *face* class may be because we extracted image patches in the lower resolution images. We found that small parts such as eyes cannot be precisely located in the images with a width of 256 pixels only. We decided to detect patches from low-resolution images because the patch detection technique from [54] is slow (about one minute for one original face image). The improved learning speed allows us to rapidly learn object models in new domains or develop more complex models for challenging cases.

For multi-view object detection, we have built up our own data sets using google and altavista search engines (Figure 3.5). The data sets contain two object classes: “cars” and “motorbikes”. Each data set consists of 420 images. The objects in the data sets have continuous view changes, different styles and background clutters. The variations of the objects in the images roughly reflects the variations of the objects in web images, so that we can assess the performance of our algorithm for classifying and searching the web images. Before learn-



(a) Cars

(b) Motorbikes

Figure 3.5. The multi-view objects for learning and testing

ing and detection, the images first undergo the same preprocessing procedures as the case of the single view detection. To save the computation cost, we only use two mixture components in the Mixture of Random ARG model. The performances using different learning schemes are listed below.

dataset	sing	manu	auto	relax
Cars	74.5%	73.5%	76.2%	76.3%
MotorBikes	80.3%	81.8%	82.4%	83.7%

Table 3.2. Multi-view object detection performance

The baseline approach is the single Random ARG detection (”sing”), namely we use one Random ARG to cover all variations including view changes. Three different multi-view learning methods are tested against the baseline approach. In learning based on automatic view discovery (“auto”), Eq.(3.17) is used to update the assignment probability (i.e. the view probability) in each E-M iteration. In learning based on manual view assignment(“manu”), update through Eq.(3.17) is not used in E-M. In stead, the assignment probability is computed from the view annotation data and fixed throughout the learning procedure. In learning by combining view annotation and view discovery (“relax”), we first learn each component Random ARG model using view annotations. The automatic view discovery is then followed to refine the parameters. The view annotation procedure is realized by letting annotators inspect the images and assign a view label to each image. Here, because we only have two components, each image is assigned with either “side view” or “front view”. Although there are objects with “rear view”, they are very rare. Besides, we do not distinguish the orientations of the objects in “side view”.

From the experiments, it is observed that the “manu” mode performs worse than the “auto” and “relax” mode. This is because the continuous view variations in the data set makes the view annotations inaccurate. Overall, the “relax” model performs best. This is consistent with our theoretical analysis: learning based on view annotation ensures the component Random ARGs can be learned correctly, and the following refinement by automatic view discovery optimizes the parameters of the component Random ARGs as well as view assignments which could be inaccurate by manual annotations.

3.6 Summary

We have presented a new statistical part-based model, called Random ARG, for object representation and detection. We solve the part matching problem through the formulation of an Association Graph that characterizes the correspondences between the parts in the image and nodes in the object model. We prove an important mathematical property relating the likelihood ratio for object detection and the partition functions of the MRFs defined on the Association Graph. Such discovery allows us to apply efficient inference methods such as Gibbs Sampling and Loopy Belief Propagation to achieve significant performance gain. We further extend the single Random ARG model to a mixture model for multi-view object detection, which improves the detection accuracy achieved by the single Random ARG model.

Chapter 4

Image Region Labeling with a Higher-order Statistical Relational Model

4.1 Introduction

In this chapter, we deal with the region labeling problem using probabilistic methods. Region labeling refers to assigning semantic labels to the regions generated by region segmentation algorithms. The aim is to locate objects or interpret the visual scene in an image.

More specifically, we deal with the visual text detection problem. Visual text is text in images or videos that is overlaid during the editing process (overlay text) or embedded during the visual scene (scene text, e.g. the road sign). We treat the text detection problem as a region labeling problem by first segmenting an image into regions and then assigning a binary label(text or non-text) to each region. We focus on scene text detection because overlay text detection is a relatively easy problem.

Scene text detection in natural 3-D scenes is an important but challenging problem. Scene text provides semantic information about the scene in an image or events in a video segment. Therefore, the detected and recognized scene text can be used as informative features for search and retrieval. Figure 4.1 shows several examples of scene text, illustrating the variations of the shape, lighting and color of the scene text in real-world images.

4.2 Prior Work and Our Contributions

There have been much prior work on text detection, but most of them use *ad hoc* rules, lacking a systematic framework. Such approaches are difficult to generalize and achieve robust performance. They can be classified as texture based [60][3], region based [50][89], or hybrid [34][118][96]. Spatial layout analysis is also used in some of the systems in a rule based setting.

Text lines or words can be modeled as multi-part objects, where characters are disconnected parts. There has been some prior work on parts-based object detection and motion analysis. For example, in [11][30], a part constellation model is proposed to detect multi-part object with supervised and unsupervised learning. Spatial relations of parts are modeled using covariance matrix. In [29], objects are modeled as trees. Detecting objects is realized by matching model trees and input pictures. In [93], human motion detection is realized by a parts-based approach, where the parts modeling is limited to triangulated decomposable graphs. In [47], a parts-based approach is proposed to detect human body. Boosting is applied to combine weak classifiers corresponding to different body part assemblies. In [113],

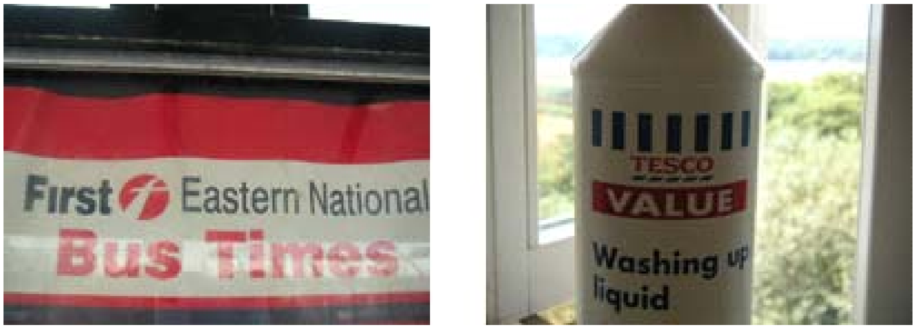


Figure 4.1. Examples of scene text in images

a graph partitioning approach is developed to group individual parts into objects. However, no probabilistic structure is presented to support systematic learning.

Markov Random Field (MRF) is an undirected graphical model, having widespread applications in computer vision. MRF with pairwise potential and belief propagation has been applied in many low-level vision applications [32] and high-level applications [70]. However, in order to detect multi-part objects, pairwise potential is often inadequate since it only captures two-node constraints. For example, in the text detection task, the pairwise potential cannot capture the unique spatial relationship that every three characters should be aligned on a straight line or a smooth curve. Another limitation of the previous pairwise MRF model [32] is that the state potential function does not incorporate the observed features. This makes it difficult to model the parts relations for general applications. For example, if we need to enforce that the “land” region should locate below the “sky” region in a natural image, the coordinate difference of the two regions is necessary to be taken into account.

In this chapter [116], we propose a parts-based object detection system via learning a high-order MRF model. The methodology is applied to detect scene text in images. The problem is formulated as calculating the beliefs (the marginalized probability) at nodes that correspond to automatically segmented regions. In order to realize efficient probabilistic inference, a variational method similar to Bethe approximation [111] is developed, which is converted into higher-order belief propagation equations. Supervised learning of this high-order MRF model is realized by maximum likelihood estimation.

Compared with prior systems, the proposed statistical framework incorporates higher-order constraints and takes advantage of the efficient inference algorithms. The proposed higher-order MRF model is also unique in that it uses potential functions considering inter-part relational attribute.

In the experiments, the higher-order MRF model is evaluated against the pairwise MRF model using a set of public benchmark images. The experiments show a substantial performance improvement accredited to the adoption of the higher-order statistical model. Moreover, the results also show that the presented method is extraordinarily robust even for text in severely cluttered background or with significant geometric variations. These evidences confirm the advantage of the higher-order MRF model for parts-based detection of scene text and probably broader categories of objects.

4.3 Region adjacency graph formation

Region adjacency graph (RAG) is used to model the properties of parts and parts relations. In this model, each node represents a segmented region, and each edge represents the likely relations between two regions. Region detection is realized by a mean-shift segmentation algorithm [16].

The edges between nodes are established according to the spatial positions of the regions. An edge is established only if the minimum distance between two regions is less than a predetermined threshold. The value of the minimum distance threshold (MDT) should allow three consecutive characters form a three-clique (i.e. triangle). Larger MDT would yield

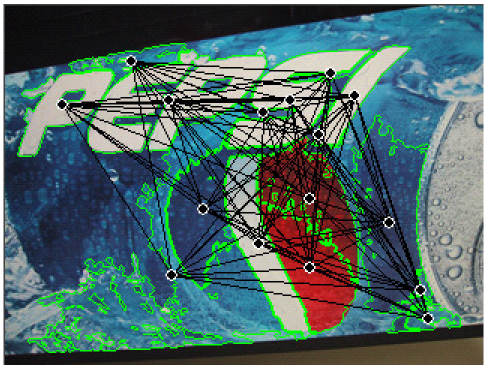


Figure 4.2. Region segmentation and adjacency graph.

denser graph and more cliques, resulting in more computation cost. The optimal selection of MDT remains an unsolved issue for future exploitation. A straightforward method is to use a multi-pass detection procedure, in which a small MDT is started and subsequently increased until text is detected.

Nested regions, such as a bounding box and its encompassed characters, would not be connected by edges, in order to prevent unnecessary computation. Moreover, the regions that touch image boundaries are assumed to be background. They are therefore eliminated to save computation resources. One example of RAG is shown in the Figure 4.2.

4.4 Formulating text detection using MRF

Based on a RAG, the corresponding Markov Random Field (MRF) is constructed by attaching each node i a state random variable X_i taking value from a label set. In text detection, the label set consists of two labels: “text” ($X_i = 1$) or “non-text” ($X_i = 0$). The observed features include one-node features y_i extracted from each region i, and three-node features y_{ijk} extracted from every three connected regions (or a three-clique in RAG). Text detection therefore can be modeled as the probabilistic inference problem given all observation features. The overall relations can be modeled as a joint probability $p(x,y)$, with $x = \{x_i | 1 \leq i \leq N\}$ and $y = \{y_i, y_{ijk} | 1 \leq i,j,k \leq N\}$ where N is the region number. Text detection is therefore the problem of computing the marginal (or belief)

$$p(x_i|y) = \sum_{x \backslash x_i} p(x,y)/p(y) \tag{4.1}$$

Labeling a region as text or non-text is realized by likelihood ratio rest of the two opposite hypotheses ($x_i = 1$,text;$x_i = 0$,non-text):

$$\frac{p(x_i = 1|y)}{p(x_i = 0|y)} = \frac{p(x_i = 1, y)}{p(x_i = 0, y)} \geq \lambda \tag{4.2}$$

where λ is a threshold, which can be adjusted to vary the precision and recall rate.

4.4.1 Pairwise MRF

Pairwise MRF has been applied in a variety of low-level vision applications [32]. The joint probability of a pairwise MRF can be written as

$$p(x,y) = \frac{1}{Z} \prod_{ij} \psi_{ij}(x_i, y_j) \prod_i \phi_i(x_i, y_i) \tag{4.3}$$

where Z is the normalization constant, $\psi_{ij}(x_i, y_j)$ is the state comparability function, $\phi_i(x_i, y_i)$ captures the compatibility between the state and observation. The marginal probability of MRF can be calculated by Belief Propagation[111].

For multi-part object detection in cluttered background, one needs to identify the parts and group them into assemblies by accommodating the relations of the parts. This requires identifying structures in the adjacency graph, not only verifying the compatibility between two nodes. For example, in text detection, we need to verify if three regions are aligned on a straight line approximately. These constraints cannot be addressed by pairwise potentials and require functions involving more than two nodes.

4.4.2 Higher-Order MRF with belief propagation

To overcome the limitation of the pairwise MRF, we attempt to utilize MRF model with higher-order potentials while retaining the computational efficiency of the belief propagation algorithm.

We adopt a unique generative model accommodating higher-order constraints, as visualized in Figure 4.3(Left), in which the observation features are not only defined at node but also three-cliques. Here we omit two-node potentials in order to simplify the computation and due to the fact that two-node constraints can be also incorporated in the three-node potentials if the graph is dense. It is not difficult to show that this model can be factorized as following:

$$p(x,y) = \frac{1}{Z}\prod_{ijk}\psi_{ijk}(x_i,x_j,x_k)p(y_{ijk}|x_i,x_j,x_k)\prod_{i}p(y_i|x_i) \tag{4.4}$$

Where y_i is the observation feature vector at node n_i. $y_{ijk}$is the clique-level relational feature, which is extracted from the entire set of nodes in the clique and is used to characterize the attribute relations of the three nodes in the same clique. Examples of clique features may include the relations of locations, shapes, and symmetry among the nodes. The higher-order potentials and clique features allow this model perform local pattern matching and evolve towards higher-scale hidden structures. The potential function containing the clique features is crucial for multi-part relationship modeling. $\psi_{ijk}(x_i,x_j,x_k)$ is the potential imposing prior constraint,and $p(y_{ijk}|x_i,x_j,x_k)$,$p(y_i|x_i)$ is the probability density functions at three-cliques and nodes respectively. Here we implicitly assume that the observation features y_{ijk},y_i are independent.

By combining the prior constraints and emission probabilities, this model is equivalent to the following MRF with inhomogeneous potentials (potential functions that vary wrt different sites):

$$p(x,y) = \frac{1}{Z} \prod_{ijk} \psi'_{ijk}(x_i, x_j, x_k, y_{ijk}) \phi'_i(x_i, y_i) \tag{4.5}$$

where $\psi'_{ijk}(x_i, x_j, x_k, y_{ijk})$ and $\phi'_i(x_i, y_i)$ are the inhomogeneous potential functions.

In the rest of the chapter, we use shorthand $\psi_{ijk}(x_i, x_j, x_k)$ and $\phi_i(x_i)$ for $\psi'_{ijk}(x_i, x_j, x_k, y_{ijk})$ and $\phi'_i(x_i, y_i)$ to simplify notations.

It has been shown that the belief propagation (BP) in pairwise MRF is equivalent to the Bethe approximation [111], a type of variational approximation. For higher-order MRF, we can use a similar variational approximation to obtain a higher-order version of the belief propagation. The detailed derivation is described in the Appendix. We could also obtain the same result by using Kikuchi approximation developed in [112][111], which is a more accurate method for probabilistic inference.

The message passing rule for higher-order BP is as following (also illustrated in Figure 4.3(Right))

$$\begin{aligned} m_{jki}(x_i) \longleftarrow \lambda \sum_{x_j} \sum_{x_k} \phi_j(x_j)\phi_k(x_k)\psi_{ijk}(x_i, x_j, x_k) \\ \prod_{(l,n)\in N_p(k)\backslash(i,j)} m_{lnk}(x_k) \prod_{(l,n)\in N_p(j)\backslash(i,k)} m_{lnj}(x_j) \end{aligned} \tag{4.6}$$

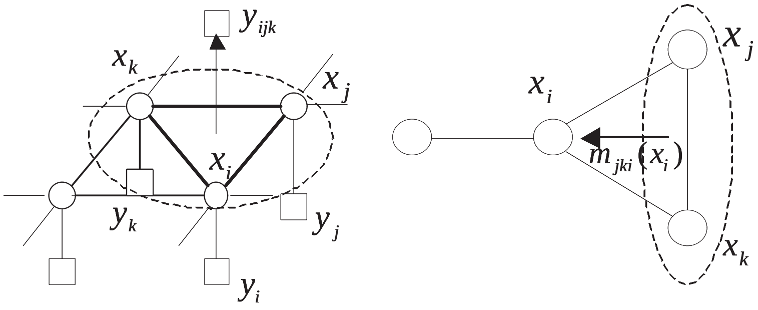


Figure 4.3. (Left) MRF with higher-order potential, node features, and clique-level relational features (Right) The message passing of the high-order belief propagation

where λ is a normalization factor so that the message computation will not cause arithmetic overflow or underflow. $N_p(i)$ is the node pair set in which every node pair forms a three-clique with the node i. Once the messages converge, the beliefs are computed using

$$b_i(x_i) = k\phi_i(x_i) \prod_{(j,k)\in N_p(i)} m_{jki}(x_i) \tag{4.7}$$

Where k is a normalization factor. Messages are uniformly initialized as a constant, typically 1.

Besides using the proposed higher-order BP, an alternative approach is to reduce the higher-order MRF to a pairwise MRF by clustering nodes and inserting additional nodes [114]. This process needs careful redesign of the potential functions and has to introduce extra delta-function-like potential functions, which may cause unstable message updates. It is therefore more straightforward to use the above higher-order version of belief propagation to perform inference.

Intuitively, the higher-order BP rules perform local pattern matching (by three-node potential with clique-level relational features) and pass around the evidences to the neighboring nodes to enhance or diminish the beliefs. To show this, Figure 4.4 shows the inference results from inputs with different numbers of characters. The brightness of each node (corresponding to a character) shown in the figure represents the belief of being "text" object. We note that more characters result in higher beliefs of the individual characters due to the interactions of the nodes.

Because the region adjacency graph is automatically generated, the topology of the graph is often loopy. Thus, in theory, the convergence of the BP cannot be guaranteed. However, our experiments on actual images so far have not observed significant divergences of message updates. This is probably due to the appropriate designs of the potential functions, or because the magnitudes of oscillations are too small to be observed.

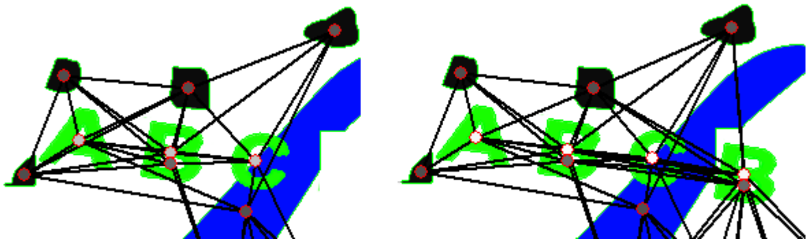


Figure 4.4. The reinforcement of the beliefs as the number of characters increases

4.5 Design of the Potential Functions

In order to effectively detect text, we need to carefully design the potential functions and emission probabilities in Eq. (4.4). The prior potentials are discrete probability mass functions. For emission probabilities, we have to adopt parametric probability density functions so that the model can be properly learned. In our system, we assume that the $p(y_{ijk}|x_i, x_j, x_k)$,$p(y_i|x_i)$ both have the form of Gaussian function or mixture of Gaussians.

In the following, we describe a few features for one-node and 3-node potential functions. Note the functions are general and other features can be added when useful, not only limited to the set we currently include in the implementation.

4.5.1 The one-node potential

In our current implementation, only aspect ratio is used as the feature for one-node potential. The distribution of the aspect ratio is modeled as Gaussian functions. There are two Gaussian *pdf*s: one for state 0 and another one for state 1, denoted as $G_0(y_i) = \mathcal{N}(\mu_0, \Sigma_0)$ and $G_1(y_i) = \mathcal{N}(\mu_1, \Sigma_1)$ respectively.

This model is accurate in the absence of segmentation errors. However, in many cases, multiple character regions may be merged due to poor region segmentation. To accommodate

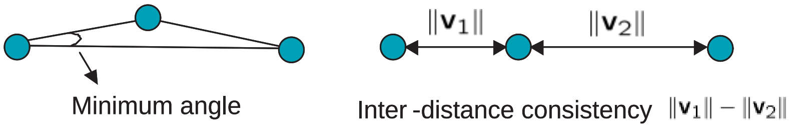

Minimum angle
$\|\mathbf{v}_1\|$
$\|\mathbf{v}_2\|$
Inter -distance consistency $\|\mathbf{v}_1\| - \|\mathbf{v}_2\|$

Figure 4.5. Examples of higher-order features

the mixed types of regions (single character regions and merged regions), we can use mixture of Gaussians to model the distribution.

4.5.2 The three-node potential

Three-node potential functions are used to enforce the spatial and visual relationship constraints on the cliques. The clique feature vector is extracted from every three-clique, the component of this vector is described as follows.

a) Minimum Angle

The feature is defined as the sinusoid of the minimum angle of the three-clique, i.e.:

$$y_{ijk}(1) = \sin(min_m \theta_m), m = 1, 2, 3.$$

where θ_m is one of the angles of the three-clique (illustrated in Figure 4.5). For a text line, the minimum angle should be close to 0. For text on a non-planar surface, the angle is assumed to be small (e.g., text on a cylindrical surface). Note that the statistical modeling approach allows for soft deviation from a fixed value, and thus non-planar text with small angles can also be detected.

b) Consistency of the region inter-distance

For most scene text in an image, the difference of the character inter-distance is approximately the same. The feature is defined as ,

$$y_{ijk}(2) = \|\mathbf{v}_1\| - \|\mathbf{v}_2\|$$

where $\mathbf{v}_1$,$\mathbf{v}_2$ are the two laterals with the maximum angle in the triangle (illustrated in Figure 4.5).

c) Maximum color distance

The feature is defined as the maximum pairwise color distance of the three regions. The use of this feature is based on the fact that the text regions in a text line have near uniform color distribution. The color distance is defined in the HSV space. For greyscale images, we can replace the color distance with the intensity difference although it may not be as robust as using color.

d) Height consistency of the character

The constraint enforces that the heights of the three character regions are approximately the same. The height divergence ratio is defined as

$$y_{ijk}(4) = (h_{max} - h_{min})/h_{min}$$

where h_{min} and h_{max} are the minimum and maximum height of the three regions. English characters usually are written with fixed discrete levels of height. Thus a mixture of Gaussian model would be adequate.

4.6 Learning the Higher-Order MRF

Learning the Higher-Order MRF is realized by the maximum likelihood estimation. Suppose M images are used in training. We want to estimate the optimal parameter set $\hat{\theta}$ to maximize the likelihood of the whole set of images.

$$\hat{\theta} = argmax_{\theta} \sum_{m=1}^{M} \ln p(x^m, y^m|\theta) \tag{4.8}$$

x^m,y^m is the state vector and observation feature vector in the mth image, where x^m is labelled by annotators. According to Eq.(4.4), the joint log likelihood of x,y in one image can be factorized as

$$\begin{aligned} \ln p(x, y) = \sum_{ijk} \ln \psi(x_i, x_j, x_k|\theta_x) + \\ \sum_{ijk} \ln p(y_{ijk}|x_i, x_j, x_k, \theta_{y3}) + \sum_{i} \ln p(y_i|x_i, \theta_{y1}) - \ln Z \end{aligned} \tag{4.9}$$

Where θ_x is the parameter for the state prior probability mass function. θ_{y3} is the parameter of the probability density function for the three-clique relational feature. θ_{y1} is for the one-node observation density. Since these three functions have independent parameters, and the partition function Z is independent on θ_{y3},θ_{y1} due to full factorization, the learning process can be carried out separately for each term. The maximum likelihood estimates of θ_{y3},θ_{y1} are obtained by simply calculating the mean and variance (or covariance matrix) of the Gaussian functions using labeled data. θ_x is the parameter for the prior distribution. Currently the prior distribution is assumed to be uniform.

The features presented in Section 4.5 require the potential functions of each clique invariant to permutation of label assignments of the states in the same clique. For a three-clique, there are 8 different configurations, but due to the permutation invariance, there are only 4 different configurations $(x_i, x_j, x_k) = (111)$,$(x_i, x_j, x_k) = (110)$,$(x_i, x_j, x_k) = (100)$, $(x_i, x_j, x_k) = (000)$. As an example, $(x_i, x_j, x_k) = (111)$ means all three nodes in the clique are text regions. Correspondingly, we have Gaussian *pdf*s:

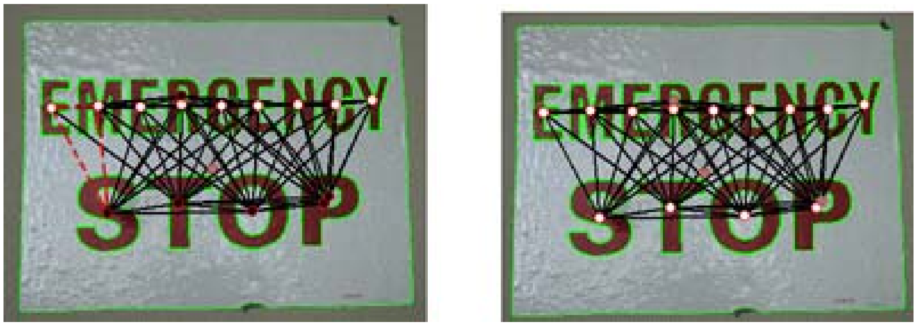


Figure 4.6. (Left) The miss in detecting multiple text lines due to cross-text-line (CTL) cliques. (Right) the results after potential function modification.

$$G_{111}(y_{ijk}) = p(y_{ijk}|x_i = 1, x_j = 1, x_k = 1) = \mathcal{N}(\mu_{111}, \Sigma_{111})$$

$$G_{110}(y_{ijk}) = p(y_{ijk}|x_i = 1, x_j = 1, x_k = 0) = \mathcal{N}(\mu_{110}, \Sigma_{110})$$

$$G_{100}(y_{ijk}) = p(y_{ijk}|x_i = 1, x_j = 0, x_k = 0) = \mathcal{N}(\mu_{100}, \Sigma_{100})$$

$$G_{000}(y_{ijk}) = p(y_{ijk}|x_i = 0, x_j = 0, x_k = 0) = \mathcal{N}(\mu_{000}, \Sigma_{000})$$

4.7 Modification of the potential functions for multiple text lines

The above detection algorithm works well when the image only contains single text line or the text lines are apart far away. However, if two or more text lines are close to one another, the algorithm will miss one or more text lines, as shown in the Figure 4.6. Such miss of detection is due to the negative constraint produced by the cross-text-line cliques (marked as dashed red lines in the Figure 4.6(Left)). In this case, the value of $G_{110}(y_{ijk})$,$G_{100}(y_{ijk})$,$G_{000}(y_{ijk})$ may be much larger than $G_{111}(y_{ijk})$ for a cross-text-line clique. The one-dimensional illustration of this situation is shown in the Figure 4.7, where the red (blue) curve indicates the potential function trained from "text"-"text"-"non-text" (text-text-text) cliques. Consequently, assigning the "non-text" label to one of the nodes in the

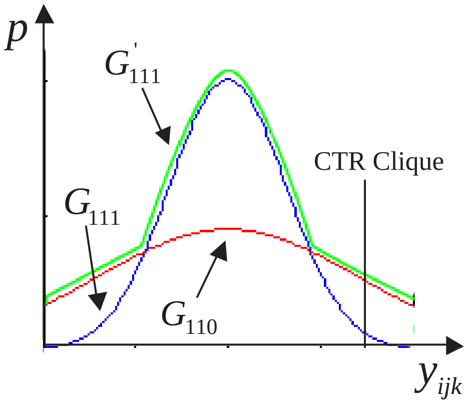


Figure 4.7. The potential functions and the modified version of $G_{111}(y_{ijk})$

cross-text-line three-clique will yield higher overall likelihood (as shown in the dashed line). One way to fix this problem is to modify the $G_{111}(y_{ijk})$ potential function such that it only has positive constraint effect within the desired feature range by the following operator

$$G'_{111}(y_{ijk}) = \sup\{G_{110}(y_{ijk}), G_{100}(y_{ijk}), G_{000}(y_{ijk})\}$$

The resulting function is shown in Figure 4.7. Therefore if the three-node feature is far from the mean of the Gaussian, it no longer gives higher value for $G_{110}(y_{ijk})$,$G_{100}(y_{ijk})$,$G_{000}(y_{ijk})$ compared with $G_{111}(y_{ijk})$. This modification shows very significant improvement in the experiments while it does not significantly impact the non-text regions. Figure 4.6(Right) shows the results by using the modified potentials. One potential drawback of the above modification is that it may raise the belief of the non-text region and thus increase false alarms. However, if the text line has enough characters, the likelihood ratio test with higher threshold will correctly reject those non-text regions. Another problem is that some singleton regions disconnected with any other region may exist in image. No three-node potential constraint is imposed on these nodes. Consequently, the beliefs are totally determined by the one-node potential function, which is often inaccurate.

To handle this problem, we can let the one-node potential only give negative constraint to non-text region if the features are quite different from the learned Gaussian mean. Thus, the one-node potential is modified using:

$$G'_1(y_i) = \sup\{G_1(y_i), G_0(y_i))\}$$

4.8 Experiments and results

To evaluate the proposed approach, we evaluate the performance using a public dataset used in the scene text detection competition in ICDAR 2003 [64]. The dataset contains 20 images with different natural conditions, for example, outdoor/indoor, background clutter, geometric variations, lighting variation, etc. All are colored images in the RGB format.. The resolution of these images is very high with a typical size 1280x960. To reduce the computation cost, we resize these images to about 640x480. This test set is limited since only images containing text are included. In order to evaluate the capability of the system in rejecting false regions in the cluttered images, another ten images with cluttered background but without text are added to the data set.

A cross-validation procedure is used to test the algorithm: the data is divided into two subsets, each of which alternates as training and testing set in a two fold cross-validation process. In the learning stage, each image first segmented by the mean-shift algorithm, and the segmented regions are manually labeled as text or non-text. Cross-text-line cliques are excluded from training to avoid confusion. We measure the precision and recall of the text region detection. Recall is the percentage of the ground truth text regions that are detected, while precision is the percentage of the correct text regions in the detected regions. The accuracy is measured at the character level.

We use the MRF model with pairwise potential as the baseline for comparison. The

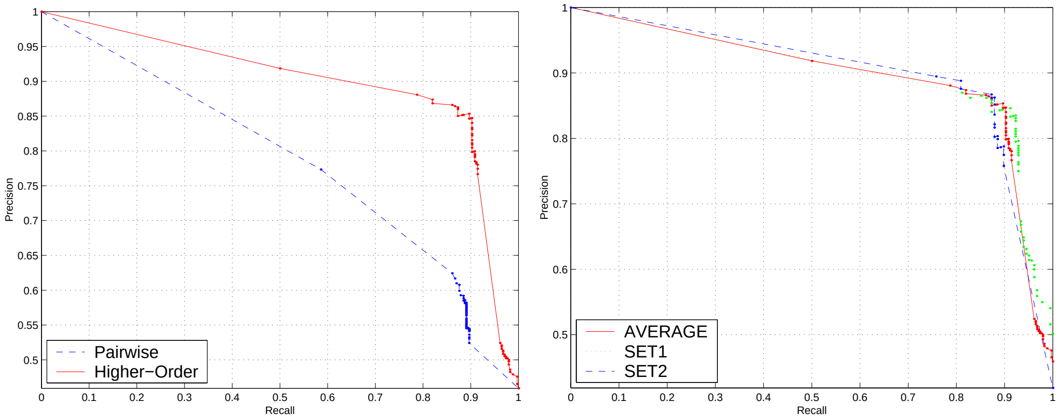


Figure 4.8. Precision recall curve, (Left) The comparison of ROC curve by using conventional pairwise MRF (dashed blue) and proposed method (red). (Right) The ROC curve of detection in set 1(green) set 2(blue) and average(red).

relational features are added into the pairwise model. It uses two features in the two-node potential - the color difference and height consistency. The one-node potential is the same as that used in the proposed higher-order MRF. The potentials are learned from labeled data. Inference is realized by standard belief propagation. A precision-recall curve (ROC curve) is generated by varying the threshold of the likelihood ratio, as shown in Eq.(4.2).

The performance comparison is shown in Figure 4.8(Left), which indicates that the higher-order MRF model significantly outperforms MRF with pairwise potential. Note interestingly there seems to be a turning point at 0.85/0.85 as precision/recall. The performance variance when using the cross-validation process is shown in Figure 4.8(Right), showing that the proposed method is stable over different training/testing partitions. Unfortunately, to the best of our knowledge, there is no public-domain performance data over the same benchmark set that we can compare.

Note that these results have not included the text regions missed in the automatic segmentation process. The miss rate of region detection is about 0.33. This makes the optimal recall (including segmentation and detection) about 0.67. The region detection miss is mainly

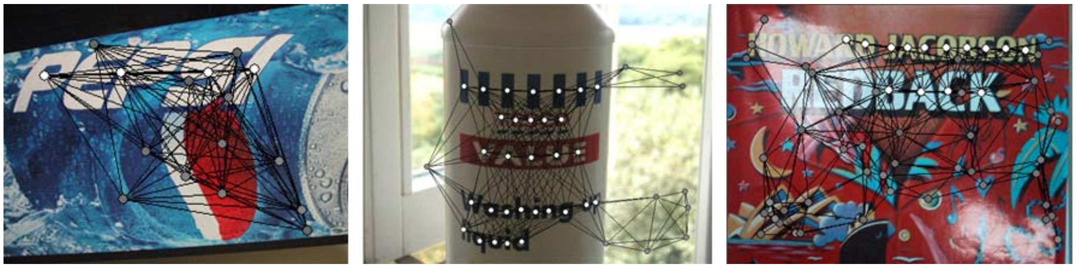


Figure 4.9. Example results from the higher-order MRF model (Brightness of nodes represents the probability as “text”)

due to the small text size. The inference computation speed excluding segmentation varies from 0.5 second to 30 second per image on a Pentium III 800MHz PC depending on the number of the cliques. The average inference speed is 2.77 second per image. The speed of segmentation and region formation is about 0.2 second to 5 second per image, depending on the image size and the content complexity of the image. The speed is improvable, since no code optimization and look-up-table is used currently.

Figure 4.9 shows some detection results by the proposed higher-order MRF model. The results show that the method is very robust to background clutteredness and geometric variations, and is able to detect text on curved as well as planar surfaces. Detecting text on curved surfaces is hard to achieve by conventional systems using fixed rules, where hard constraints are usually used. Our system achieves improved performance in this aspect by using soft constraints captured by the statistical method. Furthermore, the issue of character merging is successfully handled if the merged regions remain on the same planar or curve surfaces. To compare with MRF with pairwise potential, Figure 4.10 shows its output, which illustrates that without using the higher-order constraints, the pairwise MRF is very vulnerable to the clutter.

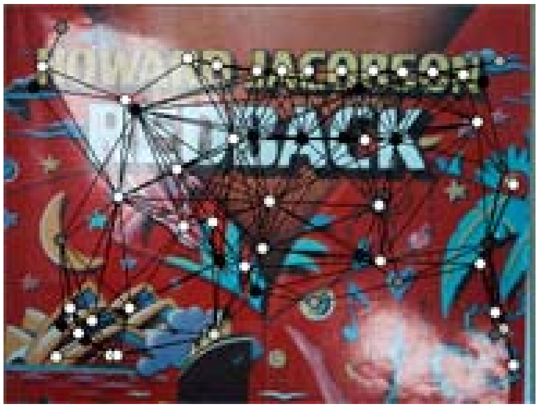


Figure 4.10. Output from the pairwise MRF model (brightness of nodes represents the probability as “text”)

4.9 Summary

We have presented a statistical method to detect text on planar or non-planar surface with limited angles in natural 3D scenes. We propose a MRF model with higher-order potential and incorporate intra-part relational features at the clique level. The proposed method is systematic, learnable, and robust to the background clutters and geometric variations. The system can be readily modified for the general multi-part object detection, for instance human body detection.

Chapter 5

Conclusion and Future work

This chapter summarizes the contributions of the thesis, discusses the limitations of the developed methods, and presents thoughts about future potential applications.

5.1 Summary of Contributions

The thesis explores the use of statistical methods for part-based modeling and learning. We have addressed three major problems arising from visual content analysis and indexing: image similarity measurement,object detection and region labeling.

For image similarity, our proposed method is motivated by human perception models and machine learning approaches. We propose a general framework for measuring the similarity of the data that are difficult to be represented as feature vectors. Specifically, we define the similarity as the probability ratio of whether or not one image is transformed from another. This framework is not only applicable to graph similarity, but also to other types of data. One example is the similarity of contours that are detected by active contour [56] algorithms or related methods. Like the case for graphs, it is also difficult to define the similarity between two contours. However, the proposed principle can be applied to compute the probability ratio of deforming one contour to another and learn the similarity to maximize the performance of retrieval or detection.

For Attributed Relational Graphs (ARG), the thesis has shown how to model the transformation between two ARGs. Especially, we show how to reduce the transformation to the specification of Markov Random Fields, which are defined on top of the association graph between two input ARGs. This work therefore extends the previous work on using association graph for Graph Isomorphism [43], which is concerned with matching vertices of structural graphs. More importantly, we show that there is an elegant relationship between the transformation probability ratio and the partition functions of the MRF. The log convexity of the partition function allows us to explore the approximation algorithms developed in convex optimization [10] and machine learning. For instance, we can use semidefinite relaxation [100] to approximate the similarity, which can be shown as an extension of the conventional semidefinite programming based approach for the graph isomorphism problem. The connection between the probability ratio and the partition functions also allows us to utilize maximum likelihood learning via the variational E-M algorithm. The benefit in practice is that the vertex-level annotation is no longer necessary in the learning stage.

For object detection, we have developed a new model called Random Attributed Relational Graph, which extends the conventional Random Graph that can only model the topological variations. In addition, the Random Attributed Relational Graph model is more general than the prior work on part-based object detection, such as the constellation model or pictorial structure, as our approach models part occlusion more accurately and provides a more effective learning scheme. For inference and learning, we have shown that the object detection likelihood ratio is related to the partition functions of the Markov Random Field (MRF). This make it possible to use variational Expectation-Maximization (E-M) in the learning procedure, instead of the E-M like algorithm used in the constellation model. We show how to model the occlusion of the individual parts via the design of the prior Markov Random Field.

We also extend the single Random Attributed Relational Graph model to a mixture model in order to detect multi-view object class more accurately. We have shown that although it is intuitive to use mixture model to represent multi-view object class, the computation cost and scarcity of training examples make this approach less attractive as anticipated. The low performance difference of using multi-view mixture model and the single Random Attributed Relational Graph model indicates that Random ARG alone can be used to model multi-view object classes.

For the region labeling problem, we explore the higher-order statistical relational model for the scene text detection problem. In order to realize efficient statistical inference, we extend the original Belief Propagation algorithm to a high-order version for the higher-order Markov Random Field. The derivation of the higher-order Belief Propagation is based on the duality between Belief Propagation and variational approximation discovered in [111].

5.2 Future work

In this section, we discuss the limitations of the developed method and present ideas about potential solutions. We also discuss the future applications of our developed methods beyond computer vision.

5.2.1 Limitations and Possible Solutions

The main limitation of our model is its high computation complexity. This problem is especially acute if the size of the ARGs or the number of the images in a database is large.

Here, we only focus on ARG similarity and Random Attributed Relational Graph, since for the region labeling problem, the LBP algorithm is often very efficient. For the ARG matching or Random ARG matching problem, the complexity is similar to that of the traditional graph matching problem as in the most general case we have to exhaustively try every matching scheme. This makes the computational complexity of our algorithms quite high. Fortunately, the Monte Carlo algorithm provides a scalable scheme for the trade-off of the performance and computation. For the case of the image similarity problem, in the experiments, we have observed that we can achieve very good performance even if we use only very few Monte Carlo samples. Furthermore, we can consider the following options for decreasing the complexity.

First, for the ARG similarity in the image database applications, we can use prefiltering schemes to first eliminate the less similarity image pairs using low-level features, such as color histogram, prior to ARG similarity computation. This prefiltering scheme is often very effective, since highly similar image pairs in an image database are usually scarce. And the data size after the prefiltering step is often small.

Second, it is possible to develop a hierarchical scheme for ARG similarity computation. This is realized by clustering the nodes of the ARG into super-nodes. Each super-node represents one vertex cluster. This approach can significantly reduce the vertex number of an ARG so as to reduce the overall computational cost. Similar ideas have been adopted in [105]. However, it is still unclear how the clustering process would affect the overall performance. It is also not straightforward to aggregate attributes at the nodes to the super-nodes.

Third, if the vertices of the graphs are associated with high-dimensional vectors, it is possible to use dimension reduction techniques to reduce the dimension of feature vectors so that the computation of the potential functions in the MRFs could be conducted more efficiently. One of possible choices of dimension reduction is geometric hashing [107], which has been widely used in prior work on image similarity.

Apart from the speed problem, another problem is the optimality of the classification. The methods developed so far are generative models. Generative models are optimal in the Bayesian classification framework under the assumption that we can accurately model the distribution of each class and we have sufficient training data for learning the model. In reality, both conditions are difficult to satisfy. Therefore, it would be important to extend the generative framework to incorporate discriminative approaches so that we can directly optimize the detection accuracy instead of maximizing the positive and negative likelihood separately.

5.2.2 Other Applications

Although we have mostly tested the proposed techniques in computer vision applications, they are generally applicable to many problems related to data analysis and data mining.

Attributed Relational Graph or Attributed Graph is a general model for representing relational data. For example, in Computed Aided Design(CAD), the design diagrams can be modeled as Attributed Relational Graphs. Matching and computing similarity of design diagrams are useful for searching engineering data. Another application is in structural chemistry and biology. Chemical compound database is widely used in structural chemistry and drug design. Searching chemical compounds in chemical databases is usually the first step in drug design. Designers need to find chemical compound having certain topological structure or certain substructure. This can be formulated as an ARG matching and similarity problem, where the vertices of the ARGs represent the atoms in chemical compounds and edges represent the bonds among atoms. In computational biology, ARG can be used to model 3-D protein structures. Each vertex of ARG can be used to model one amino acid of the protein. The edges of ARG can be used to model the spatial and topological relationship among amino acids. Here, the topological relationship basically is the connectivity between two amino acid.

Since Random Attributed Relational Graph is an extension to the Attributed Relational Graph, it can be applied to many problems related to ARGs, particularly data mining problems. Learning Random Attributed Relational Graph basically can be considered a data mining method that extracts common structures from relational data. However, different from the traditional data mining or graph mining approaches, the Random Attributed Relational Graph model provides another dimension for exploration : the statistical properties of the discovered structures, which convey more information than the deterministic structures discovered through the traditional graph mining methods [94]. Graph mining and graph-based learning has recently gained significant attention in machine learning and data mining domains, e.g., chemical compound mining [109][110], graph kernels [65][55], etc. The Random Attributed Relational Graph model potentially could be extended to develop new approaches for graph mining and graph learning problems.

Bibliography

[1] http://www-nlpir.nist.gov/projects/trecvid/. *Web site*, 2004.

[2] A. Pinz A. Opelt, M. Fussenegger and P. Auer. Generic object recognition with boosting. In *Technical Report TR-EMT-2004-01, EMT, TU Graz, Austria,*, 2004.

[3] L. Agnihotri and N. Dimitrova. Text detection for video analysis. In *Workshop on Content Based Image and Video Libraries*, pages 109–113, January Colorado, 1999.

[4] H.A. Almohamad and S.O.Duffuaa. A linear programming approach for the weighted graph matching problem. *IEEE Trans. PAMI*, 15(5):522–525, 1993.

[5] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. In *Machine Learning, vol. 50, pp. 5–43, Jan. - Feb.*, 2003.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. In *Technical Report UCB//CSD00 -1128, UC Berkeley*, January 2001.

[7] A. B. Benitez, M. Beigi, and S.-F. Chang. Using relevance feedback in content-based image metasearch. In *IEEE Internet Computing*, volume 2(4), pages 59–69, 1998.

[8] I. Biederman. Recognition-by-components: A theory of human image understanding. In *Psychological Review*, volume 94, pages 115–147.

[9] B.Luo and E.R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. PAMI*, 23(10):1120 – 1136, Oct. 2001.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[11] M. Burl and P. Perona. Recognition of planar object classes. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*. IEEE, 1996.

[12] M. Carcassoni and E.R.Hancock. Point pattern matching with robust spectral correspondence. *Computer Vision and Pattern Recognition*, 1:649–655, 2000.

[13] E. Y. Chang and B. Li. On learning perceptual distance functions for image retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002, Orlando.

[14] S.-F. Chang, G. Auffret, J. Foote, C.-S. Li, B. Shahraray, T. F. Syeda-Mahmood, and H. Zhang. Multimedia access and retrieval: the state of the art and future directions (panel session). In *ACM Multimedia*, pages 443–445, 1999.

[15] T.-S. Chua, S.-K. Lim, and H. K. Pung. Content-based retrieval of segmented images. In *ACM Multimedia*, pages 211–218, 1994.

[16] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, San Juan, Puerto Rico, June 1997.

[17] Ingemar J. Cox, Joumana Ghosn, and Peter N. Yianilos. Feature-based face recognition using mixture-distance. In *International Conference on Computer Vision and Pattern Recognition*, pages 209–216. IEEE Press, 1996.

[18] D. Crandall and P. Felzenszwalb. Spatial priors for part-based recognition using statistical models. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.

[19] A.D.J. Cross and E.R. Hancock. Graph matching with a dual-step em algorithm. *IEEE Trans. PAMI*, 20(11):1236–1253, Nov.1998.

[20] A.D.J. Cross, R.C. Wilson, and E.R. Hancock. Inexact graph matching with genetic search. *Pattern Recognition*, 30(6):953–970, 1997.

[21] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proceedings of the European Conference on Computer Vision*, Prague, Czech Republic, May 2004.

[22] J.D. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. In *IEEE Trans. Acoustics, Speech, and Signal Processing*, volume 36, pages 1169–1179, 1988.

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc, 2000.

[24] S. Ebadollahi, S.-F. Chang, and H. Wu. Automatic view recognition in echocardiogram videos using parts-based representation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 2–9, June, 2004.

[25] P. Erdös and J. Spencer. *Probabilistic Methods in Combinatorics*. New York: Academic Press, 1974.

[26] M.A. Eshera and K.S.Fu. An image understanding system using attributed symbolic representation and inexact graph matching. *J.Assoc. Computing Machinery*, 8(5):604–618, 1986.

[27] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. In *JOSA-A*, volume 14, pages 1724–1733, 1997.

[28] H. Exton. Handbook of hypergeometric integrals: Theory, applications, tables, computer programs. *Chichester, England: Ellis Horwood*, 1978.

[29] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 66–75, 2003.

[30] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 66–73. IEEE, 2003.

[31] W. T. Freeman and E. H. Adelson. Steerable filters for early vision, image analysis, and wavelet decomposition. In *IEEE International Conference on Computer Vision*, Osaka, Japan, 1990.

[32] W.T. Freeman, E.C.Pasztor, and O.T.Carmichael. Learning low-level vision. In *International Journal of Computer Vision,Vol 40, Issue 1*, pages 24–57, October 2000.

[33] B. Frey and N. Jojic. Learning graphical models of images, videos and their spatial transformations. In *Uncertainty in Artificial Intelligence*, Palo Alto, USA, 2000.

[34] J. Gao and J. Yang. An adaptive algorithm for text detection from natural scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Hawaii,2001.

[35] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *IEEE-PAMI*, volume 6, pages 721–741, 1984.

[36] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. PAMI*, 18(4):377–388, Apr. 1996.

[37] G.-D. Guo, A.K. Jain, W.-Y. Ma, and H.-J. Zhang. Learning similarity measure for natural image retrieval with relevance feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 731–736, 2001.

[38] A. Hampapur. Comparison of distance measures for video copy detection. In *Proceedings of ICME 2001*, pages 188–192. IEEE, August 2001.

[39] C.G. Harris and M.J. Stephens. A combined corner and edge detector. In *Proceedings Fourth Alvey Vision Conference, Manchester*, pages 147–151, 1998.

[40] H.Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. PAMI,* 21(9):917–922, Sept. 1999.

[41] L. Herault, R. Horaud, F. Veillon, and J.J.Niez. Symbolic image matching by simulated anealing. *Proceeding of British Machine Vision Conference*, pages 319–324, 1990.

[42] H.G.Barrow and R.J.Popplestone. Relational descriptions in picture processing. In *Machine Intelligence*, pages 6:377–396, 1971.

[43] H.G.Barrow and R.M.Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters,* 4:83–84, 1996.

[44] A. B. Hillel, T. Hertz, and D. Weinshall. Efficient learning of relational object class models. In *IEEE International Conference on Computer Vision (ICCV),* 2005.

[45] A. B. Hillel, T. Hertz, and D. Weinshall. Object class recognition by boosting a part based model. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR),* 2005.

[46] A. Holub and P. Perona. A discriminative framework for modeling object class. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.

[47] S. Ioffe and D.A. Forsyth. Probabilistic methods for finding people. In *International Journal of Computer Vision,Volume 43, Issue 1,* pages 45–68, June 2001.

[48] T. Jaakkola. Tutorial on variational approximation methods. *In Advanced mean field methods: theory and practice. MIT Press, 2000.*, 2000.

[49] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. Technical report, Dept. of Computer Science, Univ. of California, 1998.

[50] A.K. Jain and B.Yu. Automatic text location in images and video frames. In *Pattern Recognition,vol.31,no.12*, pages 2055–2076, 1998.

[51] T. Jebara. Images as bags of pixels. In *International Conference on Computer Vision*, 2003.

[52] T. Jebara. *Discriminative, Generative and Imitative Learning . PhD Thesis*. Media Laboratory, MIT, December 2001.

[53] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *International Conference on Computer Vision (ICCV)*, 2003.

[54] T. Kadir and M. Brady. Scale, saliency and image description. *In International Journal of Computer Vision*, pages 45(2):83–105, 2001.

[55] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML),* pages 321–328. AAAI Press, 2003.

[56] M. Kass, A. Witkin, and D. Terzopoulos. Snakes - active contour models. *International Journal of Computer Vision(IJCV)*, 1(4):321–331, 1987.

[57] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR03*, pages II: 409–415, 2003.

[58] C.-Y. Li and C.-T. Hsu. Region correspondence for image retrieval using graph-theoretic approach and maximum likelihood estimation. In *Proc. of ICIP*, Oct. 2004, Sigapore.

[59] F.-F. Li, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[60] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. In *IEEE Trans. on Image processing,Vol 9, No. 1*, January 2000.

[61] S. Z. Li. A markov random field model for object matching under contextual constraints. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 866–869, Seattle, Washington, June 1994.

[62] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Series: Computer Science Workbench, 2001.

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

[64] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *International Conference on Document Analysis and Recognition*, pages 682 – 687, 2003.

[65] P. Mah, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, page 70, New York, NY, USA, 2004. ACM Press.

[66] B. S. Manjunath, R. Chellappa, and C. von der Mals-burg. A feature based approach to face recogn-tion. In *IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition (CVPR)*, 1992.

[67] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *RoyalP*, volume B-200, pages 269–294, 1978.

[68] M.P.C. McQueen. A generalization of template matching for recognition of real objects. *Pattern Recognition*, 13(2):139–145, 1981.

[69] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of Computer Vision and Pattern Recognition*, June 2004.

[70] J. W. Modestino and J. Zhang. A markov random field model-based approach to image interpretation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(6):606–615, 1992.

[71] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence (UAI) 1999.*, pages 467–475.

[72] A. Natsev. *Thesis: Multimedia retrieval by regions, concepts, and constraints*. Duke University, Computer Science 2001.

[73] E. Osuna, R. Freund, and F. Girosi. Training support vector machines:an application to face detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[74] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, pages 1933 – 1955.

[75] M. Pelillo, K. Siddiqi, and S.W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. PAMI*, 21(11):1105–1120, Nov. 1999.

[76] P.Flajolet and R.Sedgewick. *Analytic Combinatorics*. August, 2005.

[77] R.C.Wilson and E.R.Hancock. Structual matching by discrete relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:1–2, 1997.

[78] R.C.Wison R.Myers and E.R.Hancock. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. PAMI*, 21(9):917–922, Sept. 1999.

[79] S. E. Robertson. The probability ranking principle in ir. *Journal of the American Society for Information Science*, 1997.

[80] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. In *Internat. J. Comput. Vision*, volume 40, pages 99–121, 2000.

[81] I. A. Rybak, V. I. Gusakova, A.V. Golovan, L. N. Podladchikova, and N. A. Shevtsova. A model of attention-guided visual perception and recognition. In *Vision Research*, volume 38, pages 2387–2400, 1998.

[82] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Systems, Man, and Cybernetics*, 13(3):353–362, May 1983.

[83] C. Schellewald and C. Schnöhrr. Subgraph matching with semidefinite programming. *Proc. Int. Workshop on Combinatorial Image Analysis (IWCIA'03),* Palermo, Italy, May 14-16 2003.

[84] G.L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of 2 images. *Proc. Royal Soc. London Series B*, 244(1309):21–26, 1991.

[85] L.G. Shapiro and R.M. Haralick. A metric for comparing relational descriptions. *IEEE Trans. PAMI*, 7(1):90–94, Jan.1985.

[86] L.S. Shapiro and R.M. Haralick. Feature-based correspondence - an eigenvector approach. *Image and Vision Computing*, 10:283, 1992.

[87] J. Shi and J. Malik. Self inducing relational distance and its application to image segmentation. *Lecture Notes in Computer Science*, 1406:528, 1998.

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

[89] J.C. Shim, C. Dorai, and R. Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *Proc. 14th International Conference on Pattern Recognition,vol. 1*, pages 618–620, Brisbane, Australia,August 1998.

[90] J. R. Smith and S.-F. Chang. Visualseek: A fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996.

[91] J. R. Smith and C.-S. Li. Image classification and querying using composite region templates. *Journal of Computer Vision and Image Understanding*, 1999.

[92] S.M. Smith and J.M. Brady. Susan - a new approach to low level image processing. In *International Journal of Computer Vision*, volume 23, pages 45–78, May 1997.

[93] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition,Vol I*, pages 810–817, Hilton Head Island, South Carolina, June 2000.

[94] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson Addison Wesley, May, 2005.

[95] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. In *Journal of the Royal Statistical Society, Series B*, volume 61(3), pages 611–622, 1999.

[96] B.L. Tseng, C.-Y. Lin, and D.-Q. Zhang. Improved text overlay detection in videos using a fusion-based classifier. In *IEEE Conference of Multimedia and Expo (ICME)*, 2003.

[97] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–C591.

[98] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Trans. PAMI*, 10:31–42, 1976.

[99] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision - to appear*, 2002.

[100] M. J. Wainwright and M. I. Jordan. Semidefinite methods for approximate inference on graphs with cycles. In *UC Berkeley CS Division technical report UCB/CSD-03-1226*, 2003.

[101] A. B. Watson. *Digital images and human vision.* Cambridge MA: MIT Press, 1993.

[102] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 101–109. IEEE, 2000.

[103] M. Welling and Y. W. Teh. Belief optimization for binary networks: A stable alternative to loopy belief propagation. In *Uncertainty of Artificial Intelligence*, 2001, Seattle, Washington.

[104] M.L. Williams, R.C. Wilson, and E.R. Hancock. Deterministic search for relational graph matching. *Pattern Recognition*, 32(7):1255–1271, 1999.

[105] R.C. Wilson and E.R. Hancock. Graph matching with hierarchical discrete relaxation. *Pattern Recognition Letter*, 20(10):1041–1052, October 1999.

[106] L Wiskott, J-M Fellous, N Krüger, and C Malsburg. Face recognition by elastic bunch graph matching. In Gerald Sommer, Kostas Daniilidis, and Josef Pauli, editors, *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, number 1296, pages 456–463, Heidelberg, 1997. Springer-Verlag.

[107] H.J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Comput. Sci. Eng.*, 4(4):10–21, 1997.

[108] A.K.C. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Trans. PAMI*, 7(5):509–609, Sept. 1985.

[109] X. Yan, X. J. Zhou, and J. Han. Mining closed relational graphs with connectivity constraints. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 324–333, New York, NY, USA, 2005. ACM Press.

[110] X. Yan, X. J. Zhou, and J. Han. Mining closed relational graphs with connectivity constraints. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 324–333, New York, NY, USA, 2005. ACM Press.

[111] J. S. Yedidia and W. T. Freeman. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*, pages pp. 239–236,Chap. 8,, Jan. 2003.

[112] J.S. Yedidia, W.T.Freeman, and Y.Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, pages Vol 13, pps 689–695, December 2000.

[113] S. X. Yu and J. Shi. Object-specific figure-ground segregation. In *Computer Vision and Pattern Recognition (CVPR)*, Madison, Wisconsin, June 16-22 2003.

[114] Y.Weiss and W.T.Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. In *Neural Computation, Vol 13*, pages 2173–2200, 2001.

[115] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM conference of Multimedia (ACM MM)*, 2004.

[116] D.-Q. Zhang and S.-F. Chang. Learning to detect scene text using a higher-order mrf with belief propagation. In *IEEE Workshop on Learning in Computer Vision and Pattern Recognition, in conjunction with CVPR (LCVPR)*, Washington DC, June 2004.

[117] D.-Q. Zhang, C.-Y. Lin, S.-F. Chang, and J.R. Smith. Semantic video clustering across sources using bipartite spectral clustering. In *Proceeding of IEEE Conference of Multimedia and Expo (ICME)*, June 2004.

[118] D.-Q. Zhang, B.L. Tseng, C.-Y. Lin, and S.-F. Chang. Accurate overlay text extraction for digital video analysis. In *International Conference on Information Technology: Research and Education (ITRE)*, 2003.

[119] H. Zhang and D. Zhong. Scheme for visual feature-based image indexing. storage and retrieval for image and video databases. In *SPIE*, 1995.

Appendix A

Proofs of Chapter 2

A.1 Proof of Lemma 2.1

Proof. By enumerating the admissible configurations, we have

$$Z(N;M;z)=\sum_{i=0}^{N}\binom{M}{i}i!\binom{N}{i}z^i \tag{A.1}$$

Its derivative is

$$\begin{aligned}\frac{dZ}{dz}&=\sum_{i=1}^{N}\binom{M}{i}\binom{N}{i}i!iz^{i-1}\\&=N\sum_{i=1}^{N}\binom{M}{i}\binom{N-1}{i-1}i!z^{i-1}=MN\sum_{i=0}^{N-1}\binom{M-1}{i}\binom{N-1}{i}i!z^i\end{aligned}$$

By enuneration, the *vertex copy probability* r can be written as

$$r=\frac{zM\sum_{i=0}^{N-1}\binom{M-1}{i}\binom{N-1}{i}i!z^i}{Z}=\frac{z}{N}\frac{dZ}{Zdz}=\frac{z}{N}\frac{d(\ln Z)}{dz} \tag{A.2}$$

For the pruned MRF, the term $\binom{M}{i}i!=M(M-1)(M-2)...$ would be changed to $d_1(d_2-V_1)(d_3-V_2)...$, where $V_1,V_2,...$ are the terms resulted from the invalid configurations. Using similar proof sequence, we will obtain the same conclusion. □

A.2 Proof of Lemma 2.2

Proof.

$$\begin{aligned} Z(N;M;z) &= \sum_{i=0}^{N} \binom{M}{i} i! \binom{N}{i} z^i = \sum_{i=0}^{N} M(M-1)...(M-i+1)\binom{N}{i} z^i \\ &\leq \sum_{i=0}^{N} M^i \binom{N}{i} z^i = \sum_{i=0}^{N} \binom{N}{i} (Mz)^i = (1+Mz)^N \end{aligned} \tag{A.3}$$

In the last step of the proof, we used the *binomial theorem*. Therefore, we have

$$\ln Z(N;M;z) \leq N \ln(1+Mz) \tag{A.4}$$

For the pruned MRF, the term $\binom{M}{i} i! = M(M-1)(M-2)...$ would be changed to $d_1(d_2 - V_1)(d_2 - V_2)...$, where $V_1, V_2, ...$ are the terms resulted from the invalid configurations. Therefore, we can use similar proof sequence and the inequality changes to

$$\ln Z(N;d_1,d_2,...,d_N;z) \leq N \ln(1 + max_i\{d_i\}z) \tag{A.5}$$

□

A.3 Proof of Proposition 2.1

Proof. We note that the partition function $Z(N;M;z)$ can be represented as a generalized hypergeometric function

$$Z(N;M;z) = \sum_{i=0}^{N} \binom{N}{i} \binom{M}{i} i! z^i = {}_2F_0(-N;-M;;z) \tag{A.6}$$

${}_2F_0(a;b;;z)$ is a generalized hypergeometric function[28], written as:

$${}_2F_0(a;b;;z) = \sum_{n=0}^{\infty} (a)_n (b)_n \frac{z^n}{n!} \tag{A.7}$$

where $(a)_p, (b)_p$ are rising factorials, written as $(a)_p = a(a+1)\cdots(a+p)$. It is not difficult to show that ${}_2F_0(a; b; ; z)$ is the solution of the following Ordinary Differential Equation (ODE):

$$z^2 y'' + [(a+b+1)z - 1]y' + aby = 0 \tag{A.8}$$

Therefore the partition function $Z(N; M; z)$ is the solution of the following ODE:

$$z^2 Z'' + [(1-N-M)z - 1]Z' + MNZ = 0 \tag{A.9}$$

To solve this ODE, we make a change of variable to let $Z = e^{Nw}$, then we have

$$\begin{aligned} Z' &= e^{Nw} N w' \\ Z'' &= N w'' e^{Nw} + N^2 (w')^2 e^{Nw} \end{aligned}$$

plug into Eq. (A.9), we get

$$z^2 N w'' + z^2 N^2 (w')^2 + (1-N-M) z N w' + MN = 0$$

Divide the above equation by N^2, then when $N \to \infty$ and $M \to \infty$, the 2nd-order ODE tends to the following 1st-order ODE

$$z^2 (w')^2 - 2zw' + 1 = 0$$

which yields $w' = \pm\frac{1}{z}$. But since z is positive and Z has to be monotonically increasing with respect to z, w' must be positive. Therefore we have solution $w = \ln(z) + \lambda$, where λ is a constant. Accordingly, we have

$$Z = e^{N\lambda} z^N$$

Therefore, when $N \to \infty$, the log partition function $\ln Z$ tends to

$$N[\ln(z) + \lambda]$$

To obtain the constant λ, we can let $z = 1$, then

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \ln \Big[\sum_{i=0}^{N} \binom{N}{i} \binom{N}{i} i! \Big]$$

Here the constant λ may be calculated numerically, which approximately equals to 4.1138. □

A.4 Proof of Theorem 2.1

Proof. We start from the posterior probability $p(X|G_d, G_m, H = 1)$. According to the Bayes rule

$$p(X|G_d, G_m, H = 1) = \frac{1}{C} p(G_d|G_m, X, H = 1) p(X|G_m, H = 1)$$

where C is the normalization term, which happens to be the positive likelihood function:

$$C = \sum_X p(G_d|G_m, X, H = 1) p(X|G_m, H = 1) = p(G_d|G_m, H = 1) \tag{A.10}$$

And the likelihood function is

$$p(X|G_d, G_m, H = 1) = \frac{\prod_i f_{B_1}(y_u) \prod_{ij} f_{B_2}(y_{uv})}{CZ} \frac{p(G_d|G_m, X, H = 1) p(X|G_m, H = 1) Z}{\prod_i f_{B_1}(y_u) \prod_{ij} f_{B_2}(y_{uv})} \tag{A.11}$$

Note that we assume the observations are independent given the correspondence. Also, plugging in the identities (Eq.(2.11)) into Eq.(A.11). We can see that for any configuration X, we have

$$\frac{p(G_d|G_m, X, H = 1) p(X|G_m, H = 1) Z}{\prod_i f_{B_1}(y_u) \prod_{ij} f_{B_2}(y_{uv})} = \prod_{ij,uv} \varsigma_{ijuv}(x_{iu}, x_{jv}) \prod_{ij} \eta_{iu}(x_{iu})$$

And the domain of $p(X|G_d, G_m, H = 1)$ and the MRF Gibbs distribution is the same. Therefore, the normalization constant should be also equal, therefore

$$\frac{CZ}{\prod_i f_{B_1}(y_u) \prod_{ij} f_{B_2}(y_{uv})} = Z'$$

Therefore the likelihood ratio is

$$\frac{p(H=1|G_d,G_m)}{p(H=0|G_d,G_m)} = \frac{Z'}{Z} \tag{A.12}$$

□

Appendix B

Proofs of Chapter 3

B.1 Proof of Lemma 3.1

Proof. To simplify the notation, we assume $N \leq M$. It is easy to extend to the case when $N > M$. The partition function can be calculated by enumerating the admissible matching (matching that does not violate the one-to-one constraint) as the following

$$Z(N; M; z_1, z_2, ..., z_N) = \sum_X \prod_{iu,jv} \psi_{iu,jv}(x_{iu}, x_{jv}) \prod_{iu} \phi_{iu}(x_{iu}) = \sum_{admissible \quad X} \prod_{iu} z_i$$

To calculate the above summation, we first enumerate the matchings where there are i nodes $n_{I_1}, n_{I_2}...n_{I_i}$ in the RARG being matched to the nodes in ARG, where $1 \leq i \leq N$,and $I_1, I_2...I_i$ is the index of the RARG node. The corresponding summation is

$$M(M-1)(M-2)...(M-i+1)z_{I_1}z_{I_2}...z_{I_i} = \binom{M}{i} i! z_{I_1}z_{I_2}...z_{I_i}$$

For all matchings where there are i nodes being matched to RARG, the summation becomes

$$\binom{M}{i} i! \sum_{1 \leq I_1 < I_2 < ... < I_i \leq N} z_{I_1}z_{I_2}...z_{I_i} = \binom{M}{i} i! \Pi_i(z_1, z_2, ..., z_N)$$

Where

$$\Pi_i(z_1,z_2,...,z_N)=\sum_{1\le I_1<I_2<...<I_i\le N} z_{I_1}z_{I_2}...z_{I_i}$$

is known as *Elementary Symmetric Polynomial*. By enumerating the index i from 0 to N, we get

$$Z(N;M;z_1,z_2,...,z_N)=\sum_{i=0}^{N}\binom{M}{i}i!\Pi_i(z_1,z_2,...,z_N) \tag{B.1}$$

Likewise, for the *presence probability* r_i, we enumerate all matchings in which the node i in the RARG is matched to a node in the ARG, yielding

$$\begin{aligned} r_i &= \frac{1}{Z}M\sum_{j=0}^{N-1}\binom{M-1}{j}j!z_i\Pi_{j|i}(z_1,z_2,...,z_N) \\ &= \frac{1}{Z}z_i\sum_{j=0}^{N-1}\binom{M}{j}j!\Pi_{j|i}(z_1,z_2,...,z_N) \\ &= z_i\frac{1}{Z}\partial Z/\partial z_i = z_i\partial\ln(Z)/\partial z_i \end{aligned}$$

Where, we have used the short-hand $\Pi_{j|i}(z_1,z_2,...,z_N)$, which is defined as

$$\Pi_{j|i}(z_1,z_2,...,z_N)=\sum_{1\le I_1<I_2<...I_p,...<I_j\le N;I_p\neq i,\forall p\in\{1,2,...,j\}} z_{I_1}z_{I_2}...z_{I_j}$$

For the pruned MRF, which is the more general case, we can separate the summation into two parts, the summation of the terms containing z_i and the summation of those not

$$Z(N;M;z_1,z_2,...,z_N)=V_1(z_1,z_2,...,z_i,...z_N)+V_2(z_1,z_2,...,z_{i-1},z_{i+1}...z_N)$$

Then the *presence probability* r_i is

$$r_i=\frac{V_1}{Z}=\frac{z_i\frac{\partial Z}{\partial z_i}}{Z}=z_i\frac{\partial\ln Z}{\partial z_i}$$

Where we have used the fact that V_1 and Z is the summation of the monomials in the form of $z_{I_1}z_{I_2}...z_{I_i}$, which holds the relationship

$$z_{I_1}z_{I_2}...z_{I_i}=z_{I_k}\frac{\partial}{\partial z_{I_k}}(z_{I_1}z_{I_2}...z_{I_i}), \qquad \forall I_k\in\{I_1,I_2,...,I_i\}$$

□

B.2 Proof of Lemma 3.2

Proof. We have obtained the closed-form of the partition function Z in the proof of Lemma 1, therefore it is apparent that Z satisfies the following inequality

$$Z = \sum_{i=0}^{N} M(M-1)...(M-i+1)\Pi_i(z_1, z_2, ..., z_N) \leq \sum_{i=0}^{N} M^i \Pi_i(z_1, z_2, ..., z_N) \tag{B.2}$$

The equality holds when N/M tends to zero. And we have the following relationships

$$\sum_{i=0}^{N} \Pi_i(z_1, z_2, ..., z_N) = 1 + z_1 + z_2 + ... + z_N + z_1 z_2 + ... + z_{N-1} z_N + ... = \prod_{i=1}^{N} (1 + z_i)$$

and

$$M^i \Pi_i(z_1, z_2, ..., z_N) = \Pi_i(Mz_1, Mz_2, ..., Mz_N)$$

Therefore, the RHS in equation (7) can be simplified as the following

$$\sum_{i=0}^{N} M^i \Pi_i(z_1, z_2, ..., z_N) = \sum_{i=0}^{N} \Pi_i(Mz_1, Mz_2, ..., Mz_N) = \prod_{i=1}^{N} (1 + Mz_i)$$

The above function in fact is the partition function of the Gibbs distribution if we remove the one-to-one constraints. Likewise, for the pruned MRF, the partition function is upper-bounded by the partition function of the Gibbs distribution if we remove the one-to-one constraints, which, by enumerating the matchings, can be written as

$$1 + d_1 z_1 + d_2 z_2 + ... + d_N z_N + d_1 d_2 z_1 z_2 + ... = \prod_{i=1}^{N} (1 + d_i z_i)$$

Therefore we have

$$\ln Z \leq \prod_{i=1}^{N} (1 + d_i z_i)$$

□

B.3 Proof of Theorem 3.1

Proof. We start from the posterior probability $p(X|O, H = 1)$. According to the Bayes rule

$$p(X|O, H = 1) = \frac{1}{C} p(O|X, H = 1)p(X|H = 1)$$

where C is the normalization term, which happens to be the positive likelihood $p(O|H = 1)$:

$$C = \sum_{X} p(O|X, H = 1)P(X|H = 1) = p(O|H = 1) \tag{B.3}$$

Next, let us rewrite the posterior probability $p(X|O, H = 1)$ as the following

$$p(X|O, H = 1) = \frac{\prod_u f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}(y_{uv})}{CZ} \frac{p(O|X, H = 1)p(X|H = 1)Z}{\prod_u f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}(y_{uv})} \tag{B.4}$$

Using the independence assumption

$$p(O|X, H = 1) = \prod_{uv} p(y_{uv}|x_{1u}, x_{1v}, ..., x_{Nu}, x_{Nv}, H = 1) \prod_{u} p(y_u|x_{1u}, ..., x_{Nu}, H = 1)$$

and plugging in the parameter mapping equations in Eq.(3.6) and Eq.(3.7). Comparing the term in Eq.(B.4) and the term in the Gibbs distribution in Eq.(3.9), we note that for any matching X, we have

$$\frac{p(O|X, H = 1)p(X|H = 1)Z}{\prod_u f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}(y_{uv})} = \prod_{iu,jv} \varsigma_{iu,jv}(x_{iu}, x_{jv}) \prod_{iu} \eta_{iu}(x_{iu})$$

Furthermore, the posterior probability $p(X|O, H = 1)$ and the Gibbs distribution in Eq.(3.9) have the same domain. Therefore, the normalization constant should be also equal, i.e.

$$\frac{CZ}{\prod_u f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}(y_{uv})} = Z'$$

Therefore the positive likelihood is

$$p(O|H = 1) = C = \frac{Z'}{Z} \prod_{u} f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2} \tag{B.5}$$

and the likelihood ratio is

$$\frac{p(O|H = 1)}{p(O|H = 0)} = \frac{\prod_u f^{+}_{B_1}(y_u) \prod_{uv} f^{+}_{B_2}}{\prod_u f^{-}_{B_1}(y_u) \prod_{uv} f^{-}_{B_2}} \frac{Z'}{Z} = \sigma \frac{Z'}{Z} \tag{B.6}$$

□

Appendix C

Higher-Order Belief Propagation

C.1 Derivation of Higher-Order Belief Propagation

Let $b_i(x_i)$ denotes the one-node belief and $b_{ijk}(x_i, x_j, x_k)$ denotes three-node belief. Let $N_p(i)$ be the node pair set in which every node pair forms a three-clique with the node i.

The energies associated with nodes and cliques can be define as

$$E_i(x_i) = -\ln\phi_i(x_i)$$

$$E_{ijk}(x_i, x_j, x_k) = -\ln\psi_{ijk}(x_i, x_j, x_k) - \ln\phi_i(x_i) - \ln\phi_j(x_j) - \ln\phi_j(x_j).$$

Then the Gibbs free energy [111] is

$$G = \sum_{ijk} \sum_{x_i x_j x_k} b_{ijk}(x_i, x_j, x_k)\big(E_{ijk}(x_i, x_j, x_k) + \ln b_{ijk}(x_i, x_j, x_k)\big) - \sum_i (q_i - 1) \sum_{x_i} b_i(x_i)\big(E_i(x_i) + \ln b_i(x_i)\big)$$

Where q_i is the degree of the node i. Therefore the Lagrangian multipliers and their corresponding constraints are

$$r_{ijk}: \sum_{x_i,x_j,x_k} b_{ijk}(x_i,x_j,x_k) - 1 = 0, \quad r_i: \sum_{x_i} b_i(x_i) - 1 = 0$$
$$\lambda_{jki}(x_i): \quad b_i(x_i) - \sum_{x_j}\sum_{x_k} b_{ijk}(x_i,x_j,x_k) = 0$$

The Lagrangian L is the summation of the G and the multiplier terms. To maximize L, we have

$$\frac{\partial L}{\partial b_{ijk}(x_i,x_j,x_k)} = 0 \quad \Rightarrow$$
$$\ln b_{ijk}(x_i,x_j,x_k) = E_{ijk}(x_i,x_j,x_k) + 1 + \lambda_{jki}(x_i) + \lambda_{kij}(x_j) + \lambda_{ijk}(x_k) + r_{ijk}$$
$$\frac{\partial L}{\partial b_i(x_i)} = 0 \quad \Rightarrow$$
$$\ln b_i(x_i) = -E_i(x_i) + \frac{1}{q_i - 1} \sum_{(j,k)\in N_p(i)} \lambda_{jki}(x_i) + r'_i$$

where r'_i is the rearranged constant.

By using change of variable or defining message as:

$$\lambda_{jki}(x_i) = \ln \prod_{(l,n)\in N_p(i)\backslash(j,k)} m_{lni}(x_i)$$

We obtain the following equations

$$b_i(x_i) = k\phi_i(x_i) \prod_{(j,k)\in N_p(i)} m_{jki}(x_i),$$
$$b_{ijk}(x_i,x_j,x_k) = k\psi_{ijk}(x_i,x_j,x_k)\phi_i(x_i)\phi_j(x_j)\phi_k(x_k)$$
$$\prod_{l,n\in N_p(i)\backslash j,k} m_{lni}(x_i) \prod_{l,n\in N_p(j)\backslash i,k} m_{lnj}(x_j) \prod_{l,n\in N_p(k)\backslash i,j} m_{lnk}(x_k)$$

Apply the constraint $b_i(x_i) = \sum_{x_j}\sum_{x_k} b_{ijk}(x_i,x_j,x_k)$, we obtain

$$m_{jki}(x_i) \longleftarrow \lambda \sum_{x_j}\sum_{x_k} \phi_j(x_j)\phi_k(x_k)\psi_{ijk}(x_i,x_j,x_k)$$
$$\prod_{(l,n)\in N_p(k)\backslash(i,j)} m_{lnk}(x_k) \prod_{(l,n)\in N_p(j)\backslash(i,k)} m_{lnj}(x_j) \qquad \text{(C.1)}$$

Which is exactly the message passing rule in Eq. (4.6) and Eq. (C.1).

Apply the constraint $b_i(x_i) = \sum_{x_j} \sum_{x_k} b_{ijk}(x_i, x_j, x_k)$, we obtain

$$\begin{aligned} m_{jki}(x_i) \longleftarrow & \lambda \sum_{x_j} \sum_{x_k} \phi_j(x_j) \phi_k(x_k) \psi_{ijk}(x_i, x_j, x_k) \\ & \prod_{(l,n) \in N_p(k) \backslash (i,j)} m_{lnk}(x_k) \prod_{(l,n) \in N_p(j) \backslash (i,k)} m_{lnj}(x_j) \end{aligned} \tag{C.2}$$