

**STATISTICAL MODELS FOR TEXT QUERY-BASED IMAGE
RETRIEVAL**

A Dissertation Presented

by

SHAOLEI FENG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2008

Computer Science

© Copyright by Shaolei Feng 2008

All Rights Reserved

STATISTICAL MODELS FOR TEXT QUERY-BASED IMAGE RETRIEVAL

A Dissertation Presented

by

SHAOLEI FENG

Approved as to style and content by:

R. Manmatha, Chair

James Allan, Member

Allen R. Hanson, Member

Patrick A. Kelly, Member

Andrew G. Barto, Department Chair
Computer Science

To Xia

ACKNOWLEDGMENTS

I would like to thank , first and foremost, my advisor R. Manmatha for his guidance and support throughout this research and with my professional development. His advice, encouragement and support are always invaluable to me. I would also like to thank my committee members for their valuable input and interest in this research.

I am also grateful to all of my fellow students and colleagues at the Center for Intelligent Information Retrieval, particularly Toni Rath, Jiwoon Jeon, Jamie Rothfeder, and Natasha Mohanty. I greatly enjoyed working with them and miss the very pleasant discussions and collaboration. I would like to give special thanks to Victor Lavrenko for his insightful discussions and kindly help. My collaboration with him has been very fruitful.

Many thanks go to my family for their constant support and unconditional love. They are always my strong backing when times are rough. Special thanks go to my wife's family. They have provided so much help for taking care of my daughter during my hard time. Finally, I would like to thank my wife Xia for her love, understanding, patience and unending support during these years. I could not have done it without her.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the National Science Foundation under grant number IIS-9909073 and in part by SPAWARSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

ABSTRACT

STATISTICAL MODELS FOR TEXT QUERY-BASED IMAGE RETRIEVAL

MAY 2008

SHAOLEI FENG

B.S., SHANDONG UNIVERSITY OF TECHNOLOGY, CHINA

M.S., CHINESE ACADEMY OF SCIENCES, CHINA

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor R. Manmatha

Image indexing and retrieval has been an active research area for more than one decade. Although many accomplishments have been made in this domain, it is still a challenging problem and far from being solved. Traditional content-based approaches make use of queries based on image examples or image attributes like color and texture, and images are retrieved according to the similarity of each target image with the query image. However, image query based retrieval systems do not really capture the semantics or meanings of images well. Furthermore, image queries are difficult and inconvenient to form for most users.

To capture the semantics of images, libraries and other organizations have manually annotated each image with keywords and captions, and then search on those annotations using text retrieval engines. The disadvantage of this approach is the huge cost of annotating large number of images and the inconsistency of annotations by different people. In

this work, we focus on general image and historical handwritten document retrieval based on textual queries. We explore statistical model based techniques that allow us to retrieve general images and historical handwritten document images with text queries. These techniques are (i) image retrieval based on automatic annotation, (ii) direct retrieval based on computing the posterior of an image given a text query, and (iii) handwritten document image recognition. We compare the performance of these approaches on several general image and historical handwritten document collections.

The main contributions of this work include (i) two probabilistic generative models for annotation-based retrieval, (ii) a direct retrieval model for general images, and (iii) a thorough investigation of machine learning models for handwritten document recognition. Our experimental results and retrieval systems show that our proposed approaches may be applied to practical textual query based retrieval systems on large image data sets.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	vi
LIST OF TABLES	xii
LIST OF FIGURES	xv
 CHAPTER	
1. INTRODUCTION	1
1.1 Motivation	3
1.2 Contributions	6
1.2.1 General Image Retrieval	7
1.2.1.1 Two Probabilistic Generative Models For Annotation-Based Retrieval	7
1.2.1.2 A Direct Retrieval Framework	10
1.2.2 Historical Handwritten Document Recognition	11
1.2.2.1 Classification Models for Holistic Word Recognition	11
1.2.2.2 Sequence Models for Holistic Word Recognition	13
2. BACKGROUND AND RELATED WORK	15
2.1 Image Annotation and Retrieval	15
2.2 Historical Handwritten Document Recognition and Retrieval	22
2.3 Image Features	25
2.3.1 Features for Image Annotation and Retrieval	25
2.3.2 Features for Historical Handwritten Document Recognition	27

3. GENERAL IMAGE RETRIEVAL BASED ON ANNOTATION	29
3.1 Relevance-modeling Approach for Image Retrieval	29
3.1.1 Overview of Relevance-modeling Approach	29
3.1.1.1 Automated image annotation	32
3.1.1.2 Text-based ranked retrieval	32
3.1.1.3 Existing Relevance Models for Image Annotation and Retrieval	34
3.1.2 Modeling Image Content	35
3.1.2.1 Feature Generation Model	37
3.1.3 Modeling Captions	38
3.1.3.1 Multiple-Bernoulli word model	38
3.1.4 Multiple-Bernoulli Relevance Model	39
3.1.4.1 Simplifying the Computation	42
3.1.4.2 Estimating Parameters of the Multiple Bernoulli Model	44
3.1.4.3 Ranked Retrieval with the Bernoulli Model	47
3.1.5 Normalized CRM	48
3.1.5.1 Relation of Normalized CRM to MBRM	49
3.2 Experiments	52
3.2.1 Datasets	52
3.2.2 Features Used	54
3.2.3 Results of Automatic Image Annotation	55
3.2.4 Ranked Retrieval with Single Word Queries	61
3.2.5 Experiments Using the Retrieval Model for Multiple Word Queries	63
4. DIRECT RETRIEVAL USING MARKOV RANDOM FIELDS	69
4.1 Markov Random Field Framework for Image Retrieval	72
4.1.1 Framework Overview	72
4.2 Image Representation and Variants of MRF	74
4.3 Continuous Markov Random Fields for Image Retrieval	75

4.3.1	Clique Potentials	76
4.3.1.1	Full Independence (MRF-FI)	76
4.3.1.2	Nearest Region Dependence (MRF-NRD)	78
4.4	Discrete Markov Random Fields for Image Retrieval	80
4.4.1	Feature Quantization and Building a Large Visual Vocabulary	80
4.4.2	Full Independent Discrete MRF	82
4.4.2.1	Multinomial Vistern Model	83
4.4.2.2	Multiple Bernoulli Vistern Model	83
4.5	MRF Training for Image Retrieval	84
4.6	Experimental Results	85
4.6.1	Retrieval Results of Continuous MRFs	85
4.6.2	Retrieval Results for Discrete MRFs	92
5.	HISTORICAL HANDWRITTEN DOCUMENT RECOGNITION	96
5.1	Classification Models for Handwritten Word Recognition	97
5.1.1	Support Vector Machines	97
5.1.2	Conditional Maximum Entropy Models	98
5.1.2.1	Discrete Predicates	100
5.1.2.2	Continuous Predicates	101
5.1.3	Naive Bayes with Gaussian Kernel Density Estimate	102
5.2	Sequence Models for Word Recognition	103
5.2.1	Word Recognition with Discrete HMMs	104
5.2.1.1	Feature Probability Smoothing for HMMs	105
5.2.2	Conditional Random Fields Framework	106
5.2.2.1	Inference and Training in CRFs	107
5.2.2.2	Training and Inference with Beam Search	108
5.2.2.3	Word Recognition with CRFs	110
5.2.3	HMM with Gaussian Kernel Density Estimates	111
5.3	Experimental Results	112
5.3.1	Experimental Setup	112

5.3.2	Results on Different Classification Models	114
5.3.2.1	SVMs	114
5.3.2.2	Conditional Maximum Entropy Models	115
5.3.2.3	Naive Bayes with Gaussian Density Estimate	116
5.3.3	Tune and Compare Beam Search for Our CRF Model	116
5.3.4	Result Comparisons	118
6.	CONCLUSION	120
6.1	Summary	120
6.2	Future Work	123
6.2.1	Models for General Image Retrieval	123
6.2.2	Models for Historical Handwritten Documents	124
 APPENDICES		
A.	ANNOTATION WORDS	126
B.	IMAGE FEATURES	128
 BIBLIOGRAPHY		
		130

LIST OF TABLES

Table	Page
3.1 Performance comparison on the task of automatic image annotation on the small Corel dataset. CRM-Seg refers to CRM with segmentations while CRM is the same model with a grid. MBRM performs best outperforming CRM by a small amount. Symbol * indicates a significant improvement over the CRM-Seg, and † indicates a significant improvement over both CRM-Seg and CRM.	57
3.2 Performance comparison on automatic annotation between MBRM and the models reported in [9]. Three different measurements are used as in [9]: the prediction score (PR), the normalized classification score (NS) and the reduction of the KL-divergence from that computed using the empirical distribution (KL). Results show that MBRM consistently outperforms better than all the models reported in [9] on the automatic annotation task.	59
3.3 Mean recall and mean precision of MBRM on the large Corel data set [9] for the automatic annotation task.	60
3.4 Ranked retrieval results (in terms of mean average precision) based on one word queries. MBRM performs much better than the multinomial model [CRM]. The second row lists which words are used as queries. The last row gives the P-value produced by the sign test showing that the performance improvement is statistically significant.	63
3.5 Retrieval performance of different algorithms on the small video dataset for different query lengths. Symbol * indicates that the result is statistically significant better than CRM.	65
3.6 Retrieval performance of CRM, MBRM and Normalized-CRM on the large video dataset for different query lengths. Symbol * indicates that the result is statistically significant better than CRM.	66

4.1	Comparisons of retrieval performance of various models, which are normalized-continuous relevance model (N-CRM), full-independent MRF model(MRF-FI) and nearest region dependent MRF model(MRF-NRD). MRF-NRD-Exp1 and MRF-NRD-Exp2 are both nearest region dependent MRF models, but their parameters (λ_F, λ_N) are trained differently. For MRF-NRD-Exp1, we trained one identical set of parameters for all query words while for MRF-NRD-Exp2 we tuned parameters separately for each individual words. Symbol * indicates a statistically significant improvement over the N-CRM, and † indicates a statistically significant improvement over both N-CRM and MRF-FI.	86
4.2	Query examples on Corel Standard test set. MRF-NRD parameters (λ_F, λ_N) were separately tuned for each word.	87
4.3	Comparison of retrieval performance of different models on the small corel set of 5k images.	88
4.4	Performance comparison on automatic annotation between MRF-FI, MBRM and the models reported in [barnard:matching]. Results show that MRF-FI consistently outperforms the models in [barnard:matching].	91
4.5	Retrieval performance comparison between discrete MRF and other models on the 5k Corel set. The running time is measured for all the 371 words in the vocabulary.	93
4.6	Retrieval performance comparison between discrete MRF and N-CRM on the TRECVID03 set. The running time is measured for all the 75 query words.	94
5.1	Experimental results using SVMs. Word accuracy is reported for two different sets of words respectively – all words in the test set (with OOV) and the set without out-of-vocabulary (OOV) words included.	114
5.2	Performance Comparisons for maximum entropy models and features.	115
5.3	N-best Beam Search with different fixed beam widths	116
5.4	Ratio Threshold Beam Search with different K values	116
5.5	KL Divergence Beam Search with different ϵ in $KL \leq \epsilon$	116

5.6 Results comparing different models. The external corpora used for transition estimates consists of a large electronic collection of writings by George Washington and Thomas Jefferson. CRFs cannot really be used with the continuous features described here and so are not directly comparable with HMMs using continuous features. 117

LIST OF FIGURES

Figure	Page
1.1 Top images returned by Google Image in respond to query words “tiger river”	6
1.2 $P(Bill_Clinton image)$ under different models for two images with annotations of different lengths.	8
1.3 A scanned page from George Washington’s collection.	12
3.1 $P(Bill_Clinton image)$ under different models for two images with annotations of different lengths.	38
3.2 MBRM viewed as a generative process. The annotation \mathbf{w} is a binary vector sampled from the underlying multiple-Bernoulli model. First we randomly pick a training example J which generates two distributions $P(\mathbf{r} J)$ and $P(\mathbf{w} J)$. The image is produced by sampling a set of feature vectors $\{r_1^{\vec{}} \dots r_n^{\vec{}}\}$, each of which represents an image region. Resulting regions are tiled to form the image.	41
3.3 Top automatic annotations produced by the CRM and MBRM models, with ground truth words correctly predicted marked in blue. MBRM performs better than CRM for the first two images and the fourth image. For the third image the annotations are identical. Note that many video frames are annotated with the words <code>graphics_and_text</code> and <code>text_overlay</code> because of the station logos - difficult to see in these images. Interestingly, some of the automatic labels do not correspond to human labels but are perfectly reasonable e.g sky in the first image, sand in the second one and people in the third.	56
3.4 Examples of annotation and probabilities for CRM and MBRM, with words correctly predicated marked in blue.	57
3.5 Negative annotation examples of using MBRM over the Corel image set.	61
3.6 First 4 ranked results for the query “tiger” in the Corel dataset using a) CRM and b) MBRM.	62

3.7	First 4 ranked results for the query “addressing”. According to the ground truth the first and the third returned by the CRM are relevant, but the other two are not. For the MBRM, all the four images are relevant.	62
3.8	Recall/Precision graphs the large video dataset with 1- and 3-word queries	67
3.9	First 4 ranked results for the query “Outdoors, Snow, Person”.	67
3.10	<i>Recall-precision graphs for the NCRM, the machine translation (MT) model.</i>	68
4.1	The configuration of MRF models for image retrieval. The top line shows the original image and its regional representation. The bottom line shows the full-independence MRF variant(left) and the nearest region dependence MRF variant(right), where edges in red are determined by nearest region pairs. To obtain the nearest region pairs, we look for the nearest neighbor of each region in terms of their mass centers in the image (e.g. the nearest region of r_1 is r_4 , and of r_4 is r_5 in the figure), and then accordingly add an edge for each pair of these two regions in the MRF configuration	75
4.2	Recall-precision graphs of various models over the TRECVID set, which are the full-independent MRF model(MRF-FI), nearest region dependent MRF model(MRF-NRD), the NCRM, the machine translation model(MT) and the HMM model.	88
4.3	Examples of top 4 ranked images in response to query “birds” and query “train” using MRF-NRD over the small Corel set. Each image in this set is 192x128 and partitioned into 24 rectangular regions	89
4.4	Examples of top 4 ranked key frames in response to query “waterfall” and query “clock” using MRF_NRD over the TRECVID2003 set. Note that the 4th image bottom contains a clock on the wall at the top right corner. It is a relevant image to clock in the ground truth. Each frame (352x264) is partitioned into 35 rectangular regions.	89
4.5	Top ranked images in response to the query “boats” using MRF-NRD over the small Corel data set. The second and the third images are irrelevant.	90
4.6	Some supporting images for the word “boats” in the training set.	90
4.7	Curves of performance vs visual vocabulary size for multinomial visterm model and multiple Bernoulli visterm model.	92

4.8	Performance vs branch factors with 1M leaf nodes	93
4.9	5 top ranked images of the discrete MRF in the test set of the 5k Corel set in responding to the query word "birds"	94
4.10	5 top ranked images of the discrete MRF in the test set of the TRECVID03 set in response to the query word "sport_event"	95
5.1	A part of one segmented page in our dataset.	113
5.2	The histogram of the word frequency in our dataset, which is subject to a Zipf distribution.	114

CHAPTER 1

INTRODUCTION

Image retrieval is concerned with searching for useful images from large, unstructured image collections. Given a user query, the task of image retrieval is to find those images relevant to users' needs, without asking users to tediously browse an entire collection. Previous work assumes that users would provide image examples or visual attributes as queries which seriously complicates query formulation and impairs the practicability of image retrieval systems. In contrast, text query based image retrieval allows users to represent their needs using textual queries and thus makes image retrieval systems more practical. This work explores statistical modeling methods for text query based image retrieval. Specifically, we research statistical models to automatically associate words with images for retrieval purposes. Unlike commercial image search engines which rely on pre-existing text surrounding images and entirely ignore the image content, we develop and investigate models to generate words for images. This is done through learning the relationship between words and image features from a training collection of annotated images.

This work focuses on statistical modeling approaches rather than new feature extraction and image processing methods although the features used will be discussed in context. A common problem with much previous work is the use of different image sets and different features making comparisons difficult. To improve our understanding it is important to be able to compare models on standard datasets and features. The use of standard datasets has led to advances in such fields as text information retrieval. To ensure fair comparisons, we use the same dataset and the same feature with those used by the models we compare with, instead of developing our own features. For some datasets used by other models, we do not

even have the images because of licensing issues. This also prevents us from developing our own features on those datasets.

The first part of this work will be devoted to automatic annotation based retrieval for general images. Image annotation is a technique to automatically predict words for images through learning from an annotated image set. The predicted words are usually some key words depicting the high-level image content, e.g. the objects and background scene in the image. It is not necessary for automatic annotation techniques to predict the correspondence between annotation words and individual image regions, i.e. for image annotation we do not need to know which annotation word labels which image region except that all the annotation words loosely label the whole image. After automatically annotating each image in the test collection, text retrieval techniques (e.g. language models [121, 50, 74, 57]) may be applied to the annotation words for retrieving images from the test collection. We develop two generative models – the multiple-Bernoulli relevance model and the normalized continuous relevance model – for estimating the joint distribution of annotation words and image features, based on which images can be annotated and retrieved. These proposed models build on previous work in relevance modeling approaches for text and image retrieval but capture the special properties of the distributions of annotation words of images. We show how the newly developed models improve the retrieval performance and how they are related to each other theoretically.

In the second part, we develop a new direct retrieval framework for text query based image retrieval. Direct retrieval models do not involve an explicit annotation step and are trained by directly optimizing retrieval performance. Our direct image retrieval framework is built using Markov random fields which model the joint probabilities of query words and image features. This Markov random field based framework is reduced to a linear model for the retrieval task and flexible enough to model feature dependencies and combine different visual information. We show that performance improvements may be achieved through direct image retrieval and feature dependency modeling. We also build a discrete version

of the Markov random field for direct image retrieval which achieves fast retrieval while retaining comparable retrieval performance.

The third part of this work will be concerned with automatic recognition for historical handwritten document retrieval, which follows the main stream of research on handwritten document image analysis. We do a through investigation of applying different machine learning models for historical handwritten document recognition. The recognizers used in this work do not rely on character segmentation and instead recognize words holistically. After training over a labeled handwritten document set, recognizers automatically create transcriptions of all the handwritten documents in a collection. Then via standard text retrieval techniques, the transcriptions automatically generated may be used for finding lines or pages of the original manuscripts relevant to users' textual queries. We compare and analyze the recognition performance of various models.

1.1 Motivation

With the development of the Internet and digital imaging devices many large image collections are being created. Popular online photo-sharing sites like Flickr [1] contain hundreds of millions of diverse pictures. Many organizations, e.g. libraries, hospitals, governments and commerce have also been creating their large image databases by scanning paintings, manuscripts, prints and drawings. Searching and finding large numbers of images from a database is a challenging problem. Traditional content-based approaches make use of information directly extracted from image pixels. Such image retrieval systems use different kinds of user queries [111, 143, 7] which are usually categorized into one of three kinds: example images, sketches of images and image attributes like color and texture. Content-based image retrieval systems (e.g. [111, 143, 7]) rank images in a collection in proportion to the similarity between the image features from the target image and the query image. Such approaches suffer from a number of problems. They do not really capture

the semantics or meaning of images well. Furthermore, they often require people to pose queries using image examples, color or texture which is difficult for most people to do.

Image queries are even harder to create for historical handwritten document image retrieval. Although some research work [96, 95, 94, 71] uses word templates or writing samples as handwritten retrieval queries, it is generally impractical for users to look for example word images or templates to form every query. So text queries are more appropriate for historical handwritten document image retrieval. In this kind of approach, handwritten images in the dataset are first labeled through recognizers [120, 145, 156, 59, 36], then retrieved through applying standard text retrieval approaches to the recognition results. More recently, probabilistic models have been proposed for automatic word image annotation and handwriting images are retrieved based on the annotation results [126]. Probabilistic annotation models are used to estimate the distributions of each word given observed image features, and then standard language models are used for document retrieval.

In this work, we focus on text query based general image retrieval and historical handwritten document image recognition. Our approaches use statistical models to associate words with images by learning models using a training set of labeled images and then retrieving images based on the associated words.

The traditional “low-tech” approach to capturing the semantics of images is to annotate each image manually with keywords or captions and then search on those captions or keywords using a conventional text search engine. The rationale here is that the keywords capture the semantic content of the image and help in retrieving the images. This technique is also used by television news organizations to retrieve file footage from their videos. While “low-tech”, such techniques allow text queries and are successful in finding the relevant pictures. The main disadvantage with manual annotations is the cost and difficulty of scaling it to large numbers of images. The consistency of annotations by different people is also a problem for images, especially when the semantics of an image is not that self-evident. Another solution to associating words with images has been to try to recog-

nize objects in the images and then retrieve the recognition results. While some success has been achieved for objects like faces [136, 160, 170] much work still needs to be done to be able to recognize general objects. In addition, conventional object recognition techniques usually require that a recognizer be trained for each object and extensive manual intervention is required to create training sets.

Although keywords and captions are not easy to obtain, the advantages of text query-based image retrieval is apparent once the annotations are available. First, it supports semantic search. Second, it is easier to implement an image search engine and the searching process is usually much faster than one based on image matching. Third, text query-based image retrieval has the potential to achieve good retrieval performance and be successfully commercialized. For example, image search engines by Google, Lycos, Alta Vista and Yahoo are popular because they provide an efficient way to search web images using user provided text queries. A recent retrieval system by Rath *et al.* [126] for historical handwritten document images also shows promising results using word queries.

For image retrieval systems based on text queries, the key problem is how to get the metadata such as captions, titles or transcriptions. Manual annotation is not practical for large volumes of image sets. Commercial image search engines for the Internet, e.g. Google image and Yahoo image, use the text surrounding each image as its description. However, these search engines entirely ignore the visual content of the images and the surrounding text doesn't always relate to the visual content of an image. The consequence is that the returned images may be entirely unrelated to users' needs. Figure 1.1 shows an example of the top ranked images by Google Image in response to the query "tiger river", from which we can see that only one image (the third image in the bottom row) is probably relevant. Furthermore, such search engines can never retrieve images which do not have any surrounding text or captions even if their visual contents are relevant to users' queries.

Surveys [25] on user behavior while searching images have shown that, in practice users prefer using query words which are closely related to the visual information of images. To



Figure 1.1. Top images returned by Google Image in respond to query words “tiger river”.

support visual content based retrieval while retaining the advantages of a semantic search, this work develops statistical approaches for joint visual-text modeling on image content and annotation words. We investigate this problem from the perspectives of automatic image annotation, word image recognition and direct image retrieval models. The techniques presented in this work assume there is a set of annotated images available for training models, in which annotation words loosely label the entire image and not necessarily individual image regions/features. As a general requirement for supervised machine learning and statistical models, a training set may be obtained by manually annotating a portion of the image sets. By learning the semantics of a labeled training set of images, our statistical models estimate the probabilities of associating each annotation word to unlabeled images and retrieve those images based on the estimated probabilities.

1.2 Contributions

In this work we focus on statistical model based approaches for general image and handwritten historical document image retrieval that allows the user to use text as queries. The

techniques used here are (i) image retrieval based on automatic annotation, (ii) Markov random field based direct retrieval, (iii) handwritten document image recognition. We compare the performance of these approaches on several general image and handwritten document image collections.

The main contributions of this work include:

1. two probabilistic generative models for annotation-based retrieval,
2. a direct retrieval model for general images,
3. a thorough investigation of machine learning models for handwritten document recognition,
4. the first application of SVM, maximum entropy models and conditional random fields for historical handwritten document recognition
5. improved hidden Markov models for historical handwritten document recognition.

The techniques and contributions are briefly discussed as follows.

1.2.1 General Image Retrieval

We propose three models for general image retrieval with text queries—two probabilistic generative models for annotation based retrieval, and a direct retrieval framework.

1.2.1.1 Two Probabilistic Generative Models For Annotation-Based Retrieval

For general image retrieval based on automatic annotation, we propose two probabilistic generative models – the multiple Bernoulli relevance model and the normalized continuous relevance model. In our models, the images are partitioned into rectangles and features are computed over these rectangles. We then learn a joint probabilistic model for (continuous) image features and words called a relevance model and use this model to annotate test images which we have not seen.



		
Models	Bill_Clinton, books, greenery, text_overlay	Bill_Clinton
Multinomial	0.25	1.0
Bernoulli	1.0	1.0

Figure 1.2. $P(\text{Bill_Clinton} \mid \text{image})$ under different models for two images with annotations of different lengths.

Previous annotation models [32, 14, 66, 78] used the multinomial distribution to model annotation words as an analogy with the text retrieval world. However, annotation text for images has very different characteristics from full text in documents. For example, keyword annotations occur only once. So a direct application of the multinomial distribution does not capture these special characteristics. Therefore, we model the distribution of words in two different ways to solve this problem. The first model assumes that the word distribution is subjected to a multiple-Bernoulli distribution while the second assumes that they are distributed according to a normalized multinomial distribution. Experimental results show that they outperform previously reported results on other models which assume multinomial distributions.

A multinomial distributes the probability mass between multiple words. For example, in the first image in Figure 1.2 the image is annotated with four words including “Bill_Clinton” and with a perfect annotation the probability for each word is equal to 0.25. In particular, “Bill_Clinton” has a probability of 0.25. On the other hand, the second image is only annotated with the word “Bill_Clinton” and, therefore, with a perfect annotation the probability of “Bill_Clinton” is 1.0. If we want to rank order images with “Bill_Clinton” in terms of probability, the second image would be preferred and would be much further ahead in rank although in reality both images contain “Bill_Clinton” and there is no real reason for preferring one over the other. This problem arises because the annotation lengths

are very different for different images. A similar problem occurs in text retrieval when document lengths vary although the situation is not as extreme because of the large number of words each document contains. A Bernoulli model avoids this problem by making decisions about each annotation word independent of the other annotations. In the example above, all the four annotations in the first image would have the same probability (i.e. 1.0 - assuming perfect annotation) and the annotation in the second image would also have probability 1.0. Specifically, this would ensure that “Bill_Clinton” in both images receives probability 1.0 and both images would, therefore, be ranked equally in a ranking of images containing faces. We demonstrate that for both a standard Corel image dataset and also the TRECVID [2] news videos - ABC, CNN news videos - that the Bernoulli model outperforms a multinomial model on image annotation.

While the annotations from a multiple Bernoulli relevance model are good, retrieval using a Bernoulli retrieval model over the annotations gives poor results. One can get good retrieval results by applying a multinomial retrieval model to the Bernoulli annotations. But this is not a “clean model”, i.e. one assumes two different word distributions for annotation and retrieval respectively. An alternative is to use a normalized multinomial continuous relevance model (NCRM) which produces the same annotations and up to a constant scale factor the same annotation probabilities as a Bernoulli relevance model. This is achieved by forcing the annotation length to be constant (by padding training images with nulls if necessary). So, for the example images in Figure 1.2 if we assume that the vocabulary size is 4, each annotation word in the first image has 0.25 probability while the “Bill_Clinton” annotation in the second image also has probability 0.25 after padding with 3 nulls. Thus, the “Bill_Clinton” annotations in both images have the same 0.25 probability as is desirable. We show that the NCRM model has the same annotation performance as the multiple Bernoulli relevance model but may be used with a multinomial retrieval model to give good retrieval performance on textual queries.

In this work, we also investigate the relationships among three kinds of models, continuous relevance models (which uses a multinomial model), multiple Bernoulli models and normalized continuous relevance models and compare them with other models(e.g. machine translation models [32], latent Dirichlet allocation model (LDA) [15]). Another contribution of this work is that by using a rectangular grid instead of regions obtained using a segmentation algorithm we show that large improvements are achieved in annotation and retrieval performance.

Through our proposed models significant improvements are achieved in annotation and retrieval performance over a number of other models on standard datasets.

1.2.1.2 A Direct Retrieval Framework

Most previous work on text query based image retrieval involves an explicit annotation/recognition step and the models are trained through maximizing the annotation performance. However, retrieval and annotation are basically two different tasks. Image retrieval focuses on the order of the returned images in the ranking list while image annotation cares about the number of words correctly predicted given an image. So models trained by maximizing annotation performance cannot guarantee the best retrieval performance. Direct retrieval models do not involve an explicit annotation step and directly maximize the retrieval performance.

Little work has been done on the direct retrieval of images using text queries before. Jeon *et al.* [66] proposed to rank images according to the Kullback-Liebler divergence of the query model and the image model, which are respectively represented by the distributions of discrete visual features co-occurring with a particular word and features of the test image. In this work, we propose a Markov random fields based framework for direct retrieval over general images and handwritten document images. Direct retrieval here means that there isn't an explicit annotation or recognition procedure before retrieval. Instead, the retrieval is done directly through the posterior of an image given a query $P(I|Q)$. Under

this framework, parameters are trained through maximizing the mean average precision directly rather than maximizing the likelihood as done by most other models. This framework is based on Markov random fields and analogous to a similar framework by Metzler and Croft [103] for text retrieval. Another contribution of this framework is that it is flexible enough to model the visual feature dependency, which is important to capturing the image structure information. Our experiments show that modeling feature dependency can significantly improve the retrieval performance. By building a discrete Markov random field model, we also achieve greatly improved running times for training and retrieval.

1.2.2 Historical Handwritten Document Recognition

1.2.2.1 Classification Models for Holistic Word Recognition

For handwritten document image recognition and retrieval, we focus on the recognition and retrieval over historical handwritten manuscripts. Manmatha and his colleagues have done a lot of work [94, 95, 126, 79, 96] on this task and developed a retrieval system for historical handwritten manuscripts [126] based on continuous relevance models. They also proposed word spotting [94, 95, 96] and a word level hidden Markov model (HMM) [79] for this task. Figure 1.3 shows a scanned page from the corpus of George Washington's letters collected by the Library of Congress. On those degraded document images, it is very difficult to do correct character segmentation. One of the contributions of this work is that we take the manuscript recognition problem as a handwritten word classification problem and perform a thorough investigation of classification models for this problem. In particular, we test and compare support vector machines(SVM), conditional maximum entropy models and Naive Bayes with Gaussian kernel density estimate and explore their behaviors and properties when solving this problem. This is the first application of these models for historical handwritten document recognition. For conditional maximum entropy models, we explore the use of different predicates including both discrete predicates and continuous predicates.

270. Letters, Orders and Instructions. October 1755.

only for the publick use, unless by particular Orders from me. You are to send down a Barrel of Flints with the Arms, to Winchester, and about two thousand weight of Flour, for the two Companies of Rangers; twelve hundred of which to be delivered Captain Ashby and Company, at the Plantation of Charles Sellars - the rest to ^{Captain} Cook's Company, at Nicholas Reasmers.

October 26.

G.W.

th25. Winchester. October 28. 1755.

Parole Hampton.

The officers who came down from Fort Cumberland with Colonel Washington, are immediately to go Recruiting; and they are allowed until the 5th of December; at which time if they do not punctually appear at the place of Rendezvous assigned them, they will be tried by a Court Martial, for disobedience of Orders.

They are to wait upon the Aid de camp at one of the block to receive their Recruiting Instructions - Each Officer present, to give in a Return immediately of the number of men he has enlisted. - One Subaltern, one Sergeant, one Corporal, one Drummer, and twenty-five private men, are to mount Guard to-day, and to be relieved to-morrow at ten o'clock. - All Reports and Returns are to be made to the Aid de camp.

Figure 1.3. A scanned page from George Washington's collection.

1.2.2.2 Sequence Models for Holistic Word Recognition

Compared with the classification models mentioned above, sequence models have advantages in formulating term dependencies and incorporating language models. Handwritten document images have particular properties which make it easier to use sequence models. They are written sequentially. For example, in English they are written from left to right, while for some other languages, a different order may be used, i.e. from right to left in Arabic. Their ground truth transcriptions are text in natural languages. A hidden Markov model (HMM) is a kind of widely used sequence model for handwriting recognition [101, 158, 79, 36]. In this work, we investigate the application of sequence models for handwritten word recognition. We improve the performance of HMMs for handwriting recognition through proper smoothing techniques using HMMs with discrete features. We also employ non-parameter estimates for HMM generative probability estimation, and show that it significantly outperforms other HMMs.

Recent research on machine learning shows conditional random fields (CRFs) [73] have advantages over HMMs on some tasks which involve labeling sequence data. On this kind of task, generative models such as HMM define a joint probability over observation and label sequences which theoretically requires enumeration of all possible observation sequences. CRFs, as conditional undirected graphic models, model the conditional probabilities of label sequences given an observation sequence. In other words they do not involve generating a testing observation sequence. Furthermore, CRFs allow arbitrary dependencies on the observation sequence. We also investigate the use of CRFs for this task and compare them to HMMs and maximum entropy. This is the first use of CRFs in this domain. Our experiments, however, show that HMMs outperform CRFs for this task.

To elaborate the methods and contributions we discussed in this section, the remainder of this work is structured as follows. Chapter 2 will go over the background of this work and the related work in the areas of image annotation, historical handwritten document recognition and image features. Chapter 3 will be devoted to the annotation based general

image retrieval, where we propose two new relevance models – the multiple-Bernoulli relevance model and the normalized continuous relevance model – for image annotation and retrieval. Chapter 4 explores direct models for image retrieval based on text queries. In this chapter, we will introduce a new direct retrieval framework based on Markov random fields and show its advantages in directly optimizing the retrieval performance and in modeling feature dependency. Chapter 5 is dedicated to the investigation of various machine learning models for historical handwritten document recognition, analyzing and comparing their properties and performance on the task of word-level document recognition. Finally, we will conclude this thesis and suggest future work in Chapter 6.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter gives an overview of previously published work related to this work. The related work falls in the areas of image annotation and retrieval and handwritten document recognition and retrieval. Because this work emphasizes models rather than features, we only briefly discuss image features used in these areas after the discussion of related approaches.

2.1 Image Annotation and Retrieval

Since this dissertation focuses on image retrieval based on text queries, this section mainly reviews the related work on associating words with images and retrieving images using words. One can refer to [155, 29] for thorough surveys of traditional content-based image retrieval systems which are based on the similarity search of visual features.

Object recognition is a conceivable approach for text query based image retrieval. One can recognize all the objects in an image database and then search images by querying over the recognition results. As a classical computer vision problem, object recognition has been studied for decades [48, 85, 153, 108, 34, 136, 160, 5, 45, 115, 72, 123, 162, 46]. It is still an active research topic and its study has achieved much success for objects in specific situations, like geometric objects (e.g., polyhedrons), human faces and vehicles with well-defined backgrounds and poses. Nevertheless, it has not satisfactorily been solved in the more general case and the state of the art still leaves much to be desired.

Image annotation techniques automatically assign keywords to images for the purposes of indexing or retrieving. Unlike object recognition, which usually recognizes and local-

izes one or several kinds of target objects from images, image annotation tags the entire image with words from a large vocabulary. Similar to how a librarian manually tags images, automatic image annotation labels the entire image rather than specific image regions. Our image annotation models differ from traditional object recognition approaches in two respects. First, our annotation approaches model all annotation words together and have simpler training processes than those of most recognition models. Most object recognition approaches train a separate model for each object to be recognized. Although the form of the model may be the same, separate training runs are required for each object. Each run requires positive and negative examples for that particular object. In contrast, annotation models proposed in this work learn all the annotation words at the same time by learning some underlying joint distributions of annotation words with visual features from annotated training images, each of which usually has many annotations. Second, our annotation models require fewer constraints on training samples. Our proposed generative models can handle multiple objects in the same training image with arbitrary backgrounds and can also annotate backgrounds like sky and grass. In contrast, statistical model based object recognition approaches usually require well-defined training examples, in which the backgrounds are relatively simple and the target objects are manually segmented or labeled. Although some of the newer object recognition techniques [45, 46] do not require that, they still seem to require one object in each image.

One important attribute of our annotation models is that they utilize the context information of image regions for annotation through modeling the joint probability of a set of image regions and annotation words. For example, from training images they learn that an elephant is more likely to be associated with grass and water than a car. Thus any grass and water regions in a given image increase the probability of recognizing the object as an elephant. Statistical models for object recognition seldom use the context information of image backgrounds although some attempts to model background context have been made in [9, 8, 20, 109, 152, 37]. Carbonetto *et al.* [20] proposed a Markov random field based

model to map words to image regions where spatial context is employed for the estimation of the probability of an image region aligned to a particular word. Murphy *et al.* [109, 152] proposed using boosted random fields to exploit both local image information as well as contextual information from other objects for object detection. Epshtein and Ullman [37] proposed to use common context fragments to identify semantically equivalent parts of the same object class in images.

Compared with object recognition, image annotation is a relatively new research topic. In 1995, Picard and Minka [119] explored using texture similarity for interactive annotation. This approach required users to first select positive or negative examples for a label and then the label is propagated over image regions with similar texture. This interactive annotation was applied to the MIT Photobook image retrieval system. In 1999, Mori *et al.* [107] proposed a co-occurrence model to annotate new images by looking at the co-occurrence of annotation words and quantized image regions created from an annotated image collection using a regular grid. This model requires large numbers of training samples for reliable probability estimation and tends to associate frequent words with every image region.

Our annotation models are closer in spirit to other recently proposed annotation models. Barnard and Forsyth [8] proposed a hierarchical aspect cluster model for image annotation, which models the generation of an annotated image as a sampling process from the nodes on a path to the root in a binary tree. Duygulu *et al.* [32] adopted the classical statistical machine translation models for image annotation and labeling. Their machine translation models first create a vocabulary of blobs by quantizing image segments obtained via segmentation algorithm like normalized cuts. Then the translation models translate each of the blobs of an image to an keyword. Barnard *et al.* [9] applied a number of models for image annotation and labeling which include hierarchical clustering models, machine translation models, probabilistic latent semantic indexing and latent Dirichlet allocation, and investigated integrating explicit correspondence of regions and words with a hierar-

chical clustering model. Carbonetto *et al.* [115] presented a related contextual translation model which indicates that using rectangular regions gives better performance than using segmented regions. Carbonetto *et al.* [21] also proposed a shrinkage model which essentially allows for continuous features in a translation model unlike the discrete translation model used in [32]. Other models proposed include maximum entropy models by Jeon and Manmatha [67], inference nets by Metzler and Manmatha [104], support vector machines (SVM) [26] and ensembles of extremely randomized decision trees [99].

Hidden Markov models (HMMs) have also been applied for image annotation and the related task of image categorization. Xie *et al.* [163] used hierarchical HMM's on a video dataset to associate words from a speech transcript of video with temporal video patterns, while Li and Wang [83] used two-dimensional multi-resolution HMMs on the Corel dataset to categorize images into different concepts. Ghoshal *et al.* [52] proposed a HMM for automatic image annotation and retrieval where each state represents a conceptual word.

Recently, Sivic *et al.* [141] proposed the use of probabilistic Latent Semantic Analysis (pLSA) to detect objects and their locations in images. Fergus *et al.* [44] extended this model to a scale invariant pLSA (TSI-pLSA) and learned object categories over noisy training images returned by commercial image search engines. Shi *et al.* [140] proposed a Bayesian learning framework to characterize the hierarchical concept structure acquired from prior domain knowledge. Yavlinsky *et al.* [165] described a simple non-parametric framework for image annotation using global image features. To tackle the problem that kernel smoothing is not effective in high-dimensional space, they adopted the Earth Mover's Distance measure [134] for kernel estimates. Zhou *et al.* [171] first calculate the distributions of quantized visual features from each image-keyword set – a subset of training images containing the same annotation word. Given a new image, the distribution of its quantized visual features is compared with the pre-calculated distributions from each image-keyword set. For each new image, they selected a fixed number of keywords with the k -least K-L divergence as the annotation. Magalhães and Rügner [4] proposed a logistic

regression method for image annotation and retrieval, where the probability of a keyword given an image is defined as a generalized linear combination of codewords. Rasiwasia *et al.* [124] presented a query-by-semantic-example based image retrieval method, which extended the query-by-example for semantic image retrieval through mapping images in the dataset and the query image into a semantic simplex defined by the keyword vocabulary and then retrieve images according to their distances to the query image in the semantic space. Yu and Tian [166] proposed an optimal semantic subspace project to model images in non-linear subspaces related to concepts for image retrieval. Yuan *et al.* [167] investigated the application of support vector machines for concept learning from large scale imbalanced data set in video retrieval. To tackle the scale and imbalance problem, they developed a meta-algorithm called the support cluster machine which iteratively selects support and no support vectors from positive samples and clustered negative samples. Carneiro *et al.* [22] defined a semantic class as the set of images labeled with a common keyword, and treated image annotation and retrieval as classification problems of the semantic classes. In the training step they first estimated a Gaussian mixture for features of each image in the training set, then pooled all the mixtures within one semantic class into a density estimate to represent the corresponding feature distribution of that semantic class. Test images are annotated and retrieved based on the ordering of the probabilities of features from each test image generated by the semantic classes.

To alleviate the problem of lack of adequate annotated training samples, Fei-Fei *et al.* [39] proposed a Bayesian approach for learning object categories from just one or several images utilizing prior knowledge of learned categories. Fan *et al.* [38] proposed to use unlabeled samples for semi-supervised hierarchical semantic learning. Natsev *et al.* [110] proposed a combination of a nearest neighbor model and a support vector machine for semantic learning of concepts from a small number of examples for multimedia retrieval.

Statistical learning models were also applied to specific annotation tasks for image or video retrieval. Yang and Hauptmann [164] adopted logistic regression and support vector

machines to combine multi-modal features for annotating news video with locations, and achieved approximately 85% accuracy in location labeling of shots from the TRECVID dataset. Ozkan and Duygulu [114] proposed a graph based method to identify frequently appearing persons from large scale news videos, in which a similarity graph is constructed for all faces in some search space determined by the speech transcript text where the query name is mentioned. Then the densest subgraph is found corresponding to the query name. Zhao *et al.* [169] propose an automated method for person annotation of family photo album using social context, body and face information. Berg and Forsyth [11] proposed a voting method incorporating text, color, shape and texture for identifying categories of animals from the animal images in the Web.

Models inspired by text retrieval techniques were also proposed for image annotation and retrieval. Jeon *et al.* [66] proposed a cross-media relevance model (CMRM) for annotation based image retrieval, which is inspired by the cross-lingual relevance model for text retrieval [76]. Lavrenko *et al.* [78] extended this model to the continuous image feature space and proposed the continuous relevance model (CRM). Both CMRM and CRM used a doubly non-parametric model to estimate the joint probability of a set of image regions and words. In this work we propose two relevance model based annotation approaches for general images – the multiple Bernoulli relevance model (MBRM) and the normalized-continuous relevance model, which are related to the CMRM and the CRM. Our models and the continuous relevance model (CRM) are significantly different from CMRM in a number ways. First, CMRM is a discrete model and cannot take advantage of continuous features. To annotate images, CMRM has to quantize continuous feature vectors into a discrete vocabulary (similarly to many other annotation models like the translation [32] models). In contrast, our annotation models and CRM directly model continuous features without information loss caused by the quantization process. Second, CMRM relies on *clustering* of the feature vectors into *blobs*. The annotation quality of the CMRM is sensitive to clustering errors which are hard to correct in the later annotation phase. Large

visual vocabularies can alleviate these problems as we will show in the later part of this work on discrete Markov random field. However, the estimation of the joint distribution in CMRM involves a summation over products of probabilities and each of these probabilities approaches zero because the discrete feature space is sparse. This makes it difficult to use CMRM with large visual vocabularies. On the other hand, since CRM and our models use continuous features they do not suffer from the clustering errors and granularity issues. Our relevance models are different from CMRM and CRM in that our models use a multiple Bernoulli or normalized multinomial distribution for word generation while both CMRM and CRM use a multinomial distribution.

Our annotation models are also significantly different from the *GM-mixture* model by Blei and Jordan [14, 15]. Although they have a similar dependence structure among the random variables involved, the topological structures and word distribution assumptions are quite different. *GM-mixture* is a fully-parametric model estimated using the EM algorithm. It assumes that some "latent aspects" generate annotations and image regions. In contrast, our annotation models make no assumptions about the topological structure and are essentially non-parametric approaches. In our annotation models every individual image in the training set is a support point for region and annotation generation leading to a computation of the expectation over all the training images. In addition the *GM-mixture* model words use a multinomial process while our models use a multiple Bernoulli or normalized multinomial distribution.

It should be stressed that the difference between our MBRM, NCRM and previously discussed models is not merely conceptual. In section 3.2 we will show that MBRM and NCRM perform significantly better than previously proposed models on the tasks of image annotation and retrieval. To ensure a fair comparison, we show results on exactly the same data set and similar feature representations as used in [32, 66, 78].

Most previous work on image retrieval based on text queries are performed by first annotating images then retrieving image based on the annotation results, where the whole

system is optimized for the annotation performance. However, maximizing annotation performance doesn't guarantee the best retrieval performance, and this has been experimentally shown in text retrieval [106]. To tackle this problem we also propose a Markov random field based direct retrieval model for images and videos. Our direct retrieval model based on Markov random field (MRF) is analogous to the Markov random field framework proposed by Metzler and Croft [103] for text retrieval. It is quite different from the traditional applications of MRF for object recognition [20] in terms of both topological structure and training method. In our direct retrieval framework, a MRF is composed of a set of nodes each of which is either an image feature (e.g. color or texture features extracted from each image region or local features represented by the SIFT descriptor) or a query word. The links could be defined differently according to the dependency between those features and query words. The proposed MRF is directly trained to maximize the retrieval performance, i.e. mean average precision. For direct image retrieval based on text queries, Jeon *et al.* [66] proposed a framework based on the Kullback-Liebler divergence of the query model and the image model. Inspired by the cross-language latent semantic indexing in text retrieval, Hare *et al.* [55] proposed a linear algebraic method for learning the semantic structure of terms in an annotated training set of images. Without an explicit annotation step, they learned a term-matrix representing an aligned semantic space of terms and documents and projected unannotated images into the semantic document space with the learned matrix. Then images are ranked based on their relative positions to the query terms in the space.

In addition to the work on general image annotation and retrieval, we also investigate the problem of historical handwritten document recognition. In the next section, we will discuss the related work in that area.

2.2 Historical Handwritten Document Recognition and Retrieval

Handwriting recognition is a classical computer vision problem which generally can be categorized into online recognition and offline recognition, depending on whether the

recognition is performed online and synchronized with the actually writing. Benefiting from the dynamic information obtained in strokes using special input devices like tablets, online handwriting recognition has advanced to the level of commercial application. One can refer to [120] for a comprehensive survey of online handwriting recognition approaches. Historical handwritten document recognition is an offline recognition process given that no dynamic information is available.

Offline handwriting recognition [120, 145, 156] has only been successful in small-vocabulary and highly constrained domains, such as postal code recognition in automatic mail sorting and bank check reading [156, 90, 53]. Only very recently have people started to look at offline recognition of large vocabulary handwritten documents [157]. A Hidden Markov Model (HMM) is a popular model used for handwritten document recognition. Rath *et al.* [79] described an approach to recognizing historical handwritten manuscripts using simple HMMs with one state for each word. By adding word bigrams from similar historical corpora they showed that the performance could approach a word level recognition accuracy of 60%. Marti *et al.* [101] proposed the use of a HMM for modern handwriting recognition. Each character is represented using a Hidden Markov model with 14 states. Words and lines are modelled as a concatenation of these Markov models. A statistical language model was used to compute word bigrams and this improved the performance by 10%. Edwards *et al.* [36, 35] described an approach to recognizing medieval Latin manuscripts using generalized-HMMs, where each state is a character or the space between characters. A similar generalized-HMM was used by Chan and Forsyth [23] for Arabic printed and handwritten documents recognition.

Recently, other models like dynamic programming techniques [130] and boosted decision trees [59] have also been proposed for handwritten document recognition. Based on the unit of recognition, handwritten document recognition can be classified into segmentation based [101, 120, 145] and holistic analysis methods [90, 89, 79, 59]. The former relies on segmentation word images into smaller units, like characters, strokes and image

columns [120, 145, 88]. The latter [90, 89, 79, 59] takes word images as recognition units and requires no further segmentation. More details on handwritten recognition may be found in survey articles by Steinherz *et al.* [145], Plamondon *et al.* [120] and Vinciarelli [156].

Direct retrieval approaches have also been proposed for document image retrieval without involving an explicit recognition procedure. Manmatha *et al.* [94, 95, 96] proposed the word spotting idea for handwritten document retrieval. Word spotting first clusters words in a collection of handwritten documents via a word image matching algorithm [125, 133], then automatically selects candidate clusters for indexing. Rath and Manmatha [127, 129, 125] investigated a number of different approaches for word matching and clustering, including SSD, shape context [10], index approach [128] and dynamic time warping (DTW) [125]. Tan *et al.* [150] represent both textual queries and document image words with features, then retrieve printed documents by matching these features. Gatos *et al.* [51] proposed a similar technique which retrieves historical typewritten documents through matching synthetic word images created from query words with automated segmented document words. After the first run of retrieval for each query, they refined the ranked list through user's relevance feedback. Inspired by the use of eigenface in face recognition [153], Terasawa *et al.* [151] proposed an eigenspace method for matching word images which are represented as sequences of small slits. Balasubramanian *et al.* [3] proposed a DTW based word matching scheme for printed document image retrieval.

More recently, Rath *et al.* [126] proposed relevance model based probabilistic approaches for automatic annotation of historical handwritten document images. They developed the first automatic retrieval system for historical manuscripts based on the joint occurrence of annotation and word images in cross-modal retrieval models. Howe *et al.* [59] proposed boosted decision trees for historical handwritten word recognition. They augmented the word classes with a very few training samples to deal with the skewed distribution of class frequencies and substantially improved the recognition performance.

In this work, we explore recognition approaches for historical handwritten manuscript. Our recognition approaches consist of two main aspects. First, we thoroughly investigate different classification models for handwritten word recognition. In particular, we compare support vector machines, conditional maximum entropy models and Naive Bayes with Gaussian kernel density estimates [41]. Second, we explore the use of sequence models for whole word recognition, e.g. conditional random field models and HMMs [42].

2.3 Image Features

The performance of image annotation, recognition and retrieval is affected by image features to a large extent. Low level visual features have been studied for decades as a classical problem in computer vision and numerous features have been proposed. Usually, the choice of features depends on the properties of the problem to solve. Different features are desired for general image classification and for image search in narrow domains. For example, in the later case it is easier to utilize domain knowledge and well-defined visual features (e.g. flower and bird indexing proposed by Das *et al.* [28, 27]).

In this section, we only discuss features used in the areas of general image annotation and retrieval, and historical handwritten document recognition.

2.3.1 Features for Image Annotation and Retrieval

Traditional content-based image retrieval (CBIR) directly relies on the extracted visual features since images in the database are directly matched with a query image or query features in terms of visual similarity. Color, texture and shape are the most widely used features for CBIR. Swain and Ballard [147] proposed the use of color histograms for image indexing. Other color features were subsequently proposed, e.g. statistical color moments [146], color constancy [49, 47], and color correlograms proposed by Huang *et al.* [61] which incorporated the spatial distribution of colors based on the histograms. Tamura *et al.* [149] proposed six texture features corresponding to human visual perception: *coarse-*

ness, contrast, directionality, line-likeness, regularity, and roughness. Haralick *et al.* [54] proposed six texture features defined on gray level co-occurrence matrices. Manjunath and Ma [93] successfully applied Gabor filters to texture feature extraction for image retrieval. Wang *et al.* [161] used Daubechies' wavelets to extract texture features in the WBIIS system. Shape features are also used in image retrieval systems. The extraction of shape features from general images usually requires the segmentation of objects from backgrounds and thus depends on the quality of image segmentation. The state of the art of automatic image segmentation is not close to achieving coherent semantic segments, though some significant progress has been made, e.g. snake and region growing proposed by Zhu and Yuille [172] and normalized cuts proposed by Shi and Malik [139]. Mokhtarian [105] explored multi-scale contour models for shape representation. Del Bimbo and Pala [13] applied elastic matching to sketch-based image retrieval. Eakins *et al.* [33] investigated a number of different shape features and matching techniques for trademark image retrieval. Spatial relationships between objects or image regions were also studied for querying images. Smith and Chang [144] used 2D-strings to describe spatial relationship of image regions in image search. Petrakis and Faloutsos [117] use graphs to represent spatial relationships among objects in medical images.

Recently, local features extracted from regions around interesting points were proposed and showed their capability for representing image visual content. Schmid and Mohr [135] proposed view and occlusion invariant local features for image retrieval. Local patch-based salient features was proposed by Tuytelaars and van Gool [154] for stereo matching and image retrieval. Lowe [87, 86] proposed the scale invariant feature transform (SIFT) approach for detecting and extracting local feature descriptors that are invariant to scale, rotation, illumination, image noise, and small changes in viewpoint. Sivic and Zisserman [142] proposed an object and scene retrieval system for videos called "video Google" which adopted SIFT descriptors as indexing features. Nister and Stewenius [112] constructed a vocabulary tree to index SIFT descriptors of objects and objects are retrieved based on

term frequency-inverse document frequency (tf-idf) measures of the visual words. Philbin *et al.* [118] presented fast object retrieval approaches using large visual vocabularies constructed by an approximate k-means clustering method over SIFT descriptors. Other local features include Harris corners proposed by Harris and Stephens [56], and the Maximally Stable Extremal Regions (MSERs) proposed by Matas *et al.* [102]. For the purpose of fair comparisons between our models and others, this work on general image annotation and retrieval builds on color and texture features previously used for these datasets. features based on interest point

2.3.2 Features for Historical Handwritten Document Recognition

Recognition of historical manuscripts is quite different from object recognition in general images. First, historical manuscripts are often degraded and noisy. To correctly recognize the handwriting in historical manuscripts, a preprocessing stage is usually required to remove noise, enhance document images and localize text in document images. Second, because of the symbolic nature of text, features extracted from recognition units (words or characters) are more texture or shape related in order to capture the symbol forms of word or character images.

One can find the various pre-processing steps usually employed in handwritten document recognition searchable in previous surveys [145, 156]. In particular, to reverse the distortions caused by wrapping scan or slightly rotation of scanned pages, Hutchison and Barrett [62] proposed documents registration using Fourier-Mellin transform based affine warping. Cao *et al.* [19] wrapped scanned book pages according to a reconstructed orthonormal projection. Work on removing noise from scanned pages includes the removal of black margins and long lines (e.g. [97] proposed by Manmatha and Rothfeder), and the removal of bleed-through (e.g. [150] proposed by Tan *et al.*). Layout analysis techniques were employed to detect text regions from scanned pages [6, 17] and then segment these regions into lines or words [98, 97, 100, 92]. Page segmentation is very important for

recognition since the accuracy of the segmentation has direct impact on recognition performance. Manmatha and Srimal [98] proposed the use of anisotropic Laplacian of Gaussian filters to segment pages into word objects. They specialized the algorithm to select the scale at which the images can be filtered into connected word blobs. Manmatha and Rothfeder [97] refined this algorithm experimentally to avoid under- and over-word segmentation. Mahadevan and Nagabushnam [91], and Marti and Bunke [100] presented gap metric approaches for line segmentation of handwritten text. Feldbach and Tonnie [40] combined piecewise estimates for the location estimates of the lower baseline and the upper baseline and used it for detecting and separating lines of historical handwritten church registers. After lines and words are segmented from scanned pages, standard deskewing or deslanting techniques are employed to remove slant and skew from the segmented units. One can find deskewing techniques described in [16, 159] and deslanting techniques in [16, 70].

After the preprocessing step, features are extracted from each recognition units, e.g. word images or character images. Extensive research has been done on the design of features for handwritten document recognition and the amount of related literature is large. For features extracted from characters, [113, 137] are good reviews which investigate pixel based features, and features based on the distribution of pixels and geometrical and topological features. For feature representation of word images, Rath and Manmatha [129] described a set of features for holistic word matching and recognition in historical manuscripts, which includes six kinds of scalar features, such as word image height and width, number of descenders and ascenders in the word, and five kinds of profile based features, such as projection profiles, upper/lower word profiles, background to ink transitions and gray scale variance. Based on the Harris corner detection, Rothfeder *et al.* [133] proposed an effective matching approach for handwritten word images using corner feature correspondences. Given model comparison is what we focus on, this work on historical handwritten document recognition is mainly built on the features described in [129]. We will compare our results on the dataset reported in [80].

CHAPTER 3

GENERAL IMAGE RETRIEVAL BASED ON ANNOTATION

In this chapter, we focus on general image retrieval using probabilistic annotations. We describe two kinds of statistical models, multiple Bernoulli relevance models and normalized continuous relevance model, for general image annotation and retrieval and show experimental results using them.

3.1 Relevance-modeling Approach for Image Retrieval

3.1.1 Overview of Relevance-modeling Approach

As a formal way of doing query expansion in information retrieval, relevance models have been successfully used in text retrieval [81] and cross-lingual information retrieval [76]. Our relevance-modeling approaches are analogous to these applications. In the image annotation and retrieval scenario, relevance models exploit the image context. They give meaning to an image region in the context of that image, without which isolated pixels or regions are hard to interpret.

The fundamental assumption of relevance models is the generative relevance hypothesis [75](Page 22):

“The Generative Relevance Hypothesis (GRH): for a given information need, queries expressing that need and documents relevant to that need can be viewed as independent random samples from the same underlying generative model.”

This hypothesis implies an identical distribution for both queries and documents, and that the relevance of a document to a query is determined by their ”similarity”. However, it

seems an unreasonable assumption in many scenarios since queries and documents can have quite different representations. For example, in image retrieval via text queries, queries may consist of several keywords while documents are un-annotated images. It is not plausible that these two entirely different representations - words and images - have an identical underlying distribution. To handle this problem, relevance models assume that both queries and documents originate from a common representational space which is rich enough to represent all the attributes of both. But before being presented to us both queries and documents are transformed by some deterministic functions into the forms which we actually observe. For image retrieval based on text queries we assume that both queries and documents stem from some underlying common space of an infinite collection of images, each of which contains the textual description of its visual content. So a query embodies an image along with its textual description in the common representational space although actually it consists of several keywords, and an image contains a textual description although in reality the images may have no annotation associated. A query and its relevant documents are close in the representational space and sampled according to an identical underlying distribution. Before they are presented to us the query is stripped of the image portion and the documents of descriptions.

So the basic idea of the relevance-modeling approach in this work is that images and annotation words may be thought of as random samples from the same underlying generative process defined over a common representational space which is formed by all the annotated images. This generative process is formulated using a joint probability distribution $P(\mathbf{r}, \mathbf{w})$ where \mathbf{r} is some representation of the image content, and \mathbf{w} represents the annotation. With the estimated joint distribution, we can automatically annotate a given image with the most probable caption and retrieve the most relevant image in response to any text query.

The joint distribution $P(\mathbf{r}, \mathbf{w})$ is estimated using a completely non-parametric approach, as proposed in the cross-media relevance model (CMRM) whose details will be given in

section 3.1.1.3:

$$P(\mathbf{r}, \mathbf{w}) = \sum_{J \in \mathcal{T}} P(J)P(\mathbf{w}|J)P(\mathbf{r}|J) \quad (3.1)$$

where $P(\mathbf{r}|J)$ is the *content* component of the relevance model responsible for generating the image content \mathbf{r} from each training example J . Similarly, $P(\mathbf{w}|J)$ is the *language* component of the relevance model used to sample the word \mathbf{w} from annotations of each training example. The exact distributions of these two components depend on the representation for \mathbf{r} and \mathbf{w} and will be detailed in sections 3.1.2 and 3.1.3 respectively. $P(J)$ is a distribution that determines the relative importance of the various training images, which is assumed to be uniform in the absence of prior knowledge. In relevance models, the joint distribution $P(\mathbf{r}, \mathbf{w})$ is calculated as the expectation of generating the image content \mathbf{r} and annotation \mathbf{w} over all the training images $J \in \mathcal{T}$. This may also be viewed as a mixture over all the training images. Some other proposed methods which also use a joint distribution to annotate images assume some latent topological structure of the random variables. For example, the hierarchical aspect cluster model [8] and the machine translation model [32] define hidden variables from image clusters, and the GM-mixture model [14] assumes higher level “latent aspects”. Unlike those models, the relevance model makes no assumptions about the topological structure of the random variables and employs a non-parametric expectation approach over every individual point in the training set.

Relevance models assume that visual features are independent given an observed image. Under this assumption the content component $P(\mathbf{r}|J)$ in equation (3.1) is computed as:

$$P(\mathbf{r}|J) = \prod_{i=1}^n P(\vec{r}_i|J) \quad (3.2)$$

where \vec{r}_i denotes the feature vector extracted from an image region.

In the following sections 3.1.1.1 and 3.1.1.2, we show how to annotate new images and retrieve images relevant to a text query based on Equation (3.1).

3.1.1.1 Automated image annotation

We can use equation (3.1) to annotate an un-labeled image I . Suppose the image contents are represented using a feature vector \mathbf{r}_I and we are given a training set \mathcal{T} , the most likely annotation \mathbf{w} for image I is computed as:

$$\mathbf{w}^* = \arg \max_{\mathbf{w}'} \frac{P(\mathbf{w}', \mathbf{r}_I)}{\sum_{\mathbf{w}} P(\mathbf{w}, \mathbf{r}_I)} \quad (3.3)$$

In general, it is expensive to search over all possible annotations \mathbf{w} to find the optimized one \mathbf{w}^* . In practice we limit the search to single-word annotations and annotate I with the k most-probable words under equation (3.3). k could be a fixed number for all images (e.g. the average or maximum length of all the annotations in the training set) or a variable adapted to each image by trimming all the annotation words with probabilities less than a threshold.

3.1.1.2 Text-based ranked retrieval

Ranked retrieval is strongly related to automatic annotation but is a very different problem. The purpose of automatic annotation is to find the set of annotation words best describing the content of each image as defined in equation 3.3. In automatic annotation given an image I , the words compete for the probability of being its annotations. In contrast, the goal of ranked retrieval is: given a text query \mathbf{w}_q , rank all the images in an un-annotated collection \mathcal{C} such that the relevant images are ranked as high as possible. We consider an image I relevant to a query \mathbf{w}_q if a user would use all the words from \mathbf{w}_q in her annotation of image I . This suggests a straightforward solution to the retrieval problem based on the automatic annotations – first annotate each image with the k most likely words using the equation 3.3, then return those images with all words from \mathbf{w}_q present in the annotation. Unfortunately, this approach suffers some problems. First, it will drop many relevant images in the returned list because of the trimmed annotations for each image, and thus cause recall to be low. Especially for long queries \mathbf{w}_q the chance of finding all query words in

the same annotation is vanishingly small. Second, this approach simply classifies all images into two categories relevant and irrelevant and ignores the probability information for modeling the “relevance” of images to the query. Because *retrieval* and *annotation* are two different tasks, it is possible to achieve good *retrieval* performance for a given query, even if the *annotation* is poor. For example, assume an image I consists of a bird and the sky, and that the most probable annotation words are *sky* and *aircraft*. The annotation performance for image I will be poor, since the image does not contain aircraft. However, the poor annotations for image I may not affect the retrieval performance for the query *aircraft* as long as I is ranked lower than all aircraft images, which happens when the likelihood of *aircraft* given I is lower than the likelihood of *aircraft* given any image I' that does indeed contain aircraft.

For ranked retrieval, we need to model the “relevance” of an image to the query, i.e. the possibility that an image is relevant to this query. Here we adopt a variant of the language-modeling approach to retrieval pioneered by Ponte and Croft [121], which estimates a language model M_I for each image in the collection, and then ranks the images by the probability of observing the query $\mathbf{w}_q = (v_1, \dots, v_m)$ when randomly sampling from the language model of each image. Specifically, the language modeling approach to retrieval is formulated as:

$$P(\mathbf{w}_q|M_I) = \prod_{v \in \mathbf{w}_q} P(v|M_I) = \prod_{v \in \mathbf{w}_q} \frac{P(v, \mathbf{r}_I)}{\sum_{v'} P(v', \mathbf{r}_I)} \quad (3.4)$$

where the product goes over all words v in the query \mathbf{w}_q . The language model M_I is computed as: $P(v|M_I) = \frac{P(v, \mathbf{r}_I)}{\sum_{v'} P(v', \mathbf{r}_I)}$, where the summation goes over all words v' in the vocabulary of the training set. Under equation 3.4 images that are more likely to generate a query are considered more relevant and ranked at the top in the returned list.

3.1.1.3 Existing Relevance Models for Image Annotation and Retrieval

In this section we give a brief review of existing relevance models for image annotation and retrieval, and in the following sections we will discuss the weakness of these models in modeling image content and annotations and introduce our new relevance models.

1. Cross Media Relevance Model

Jeon *et al.* [66] proposed the cross media relevance model (CMRM) for image annotation and retrieval. Analogous to the relevance model for cross language text retrieval proposed by Lavrenko *et al.* [76], CMRM treats an image as a document consisting of “visual words” or “visterms”, and estimates the joint distribution of the set of “visual words” with a set of textual words. The visterms are obtained through clustering continuous visual features extracted from image regions.

CMRM assumes multinomial distributions for both the annotation word probability and the visterm probability. Let v represent an annotation word and b a visterm. The probability of $P(v|J)$ is estimated using a smoothed likelihood:

$$P(v|J) = (1 - \alpha) \frac{\#(v, J)}{|J|} + \alpha \frac{\#(v, \tau)}{|\tau|} \quad (3.5)$$

where $|J|$ and $|\tau|$ are the numbers of annotation words in image J and in the training set τ respectively. α is a smoothing parameter over the background distribution to avoid zero probabilities for those words absent from the annotation of image J . This is a standard way of estimating the word probabilities in text retrieval [168].

Similarly, the probability $P(b|J)$ of a visual feature given image J is estimated as:

$$P(b|J) = (1 - \beta) \frac{\#(b, J)}{|J|} + \beta \frac{\#(b, \tau)}{|\tau|} \quad (3.6)$$

where $|J|$ and $|\tau|$ are the numbers of visterms in image J and in the training set τ respectively. β is a smoothing parameter for the visual word distribution.

CMRM is a discrete model and cannot take advantage of continuous features. It relies on clustering visual features into a discrete vocabulary and thus is sensitive to clustering errors. Furthermore, its performance heavily depends on the cluster granularity which is usually manually pre-selected. Too many clusters can result in a very sparse space and overfitting while a small number of clusters is insufficient to distinguish different objects.

2. Continuous Relevance Model

Lavrenko *et al.* [78] expanded CMRM into the continuous space of image features and proposed the continuous relevance model (CRM). Instead of quantizing continuous features into a discrete visual vocabulary, they directly estimate the probability of a continuous feature generated by an image through kernel density estimation. Like CMRM, CRM also makes an assumption that the word probability given an image is subject to some underlying multinomial distribution. In particular, the joint distribution of a set of continuous image features \mathbf{r} and a set of words \mathbf{w} is given by:

$$P(\mathbf{r}, \mathbf{w}) = \sum_{J \in \mathcal{I}} P(J) \prod_{i=1}^n P(\vec{r}_i | J) \prod_{v \in \mathbf{w}} P(v | J) \quad (3.7)$$

$P(v | J)$ is estimated as in CMRM (see Equation 3.5). $P(\vec{r}_i | J)$ is estimated using a Gaussian kernel density estimate, which we will discuss in 3.1.2.1.

3.1.2 Modeling Image Content

The way images are represented can strongly affect the overall annotation and retrieval performance. For the relevance modeling approach, the ideal case is that we obtain semantically-coherent regions for each image which correspond to the annotation words of that image. A number of previous approaches have used segmentation algorithms to automatically partition the image into regions [32, 9, 14, 66] in the expectation that these will produce semantic regions. However the performance of current segmentation algorithms

is still error-prone and leaves much to be desired in the sense of producing semantically meaningful regions. The errors introduced by segmentation usually cannot be corrected in the later annotation stage. Furthermore automatic segmentation is usually expensive and hence it is computationally impractical for large image datasets.

Instead of automatic segmentation, our current models first partition images into rectangular regions and then extract features from each of those regions. In our models, we impose a fixed-size rectangular grid on each image and represent the image as a set of tiles. Our experimental results show that, compared with using Normalized-cut based segmentation, the rectangular representation itself achieves a 38% improvement (in mean average precision) for the task of image retrieval on a dataset of the Corel images. On a similar object recognition task, Carbonetto and Freitas [115] show that grid partitions generate better recognition results. It is not surprising that simple rectangular partitions outperform automatic segmentation for image retrieval, given the poor performance of current segmentation algorithms. The algorithms for automatic image segmentation are performed on single images. When segmenting one image, these algorithms do not utilize the information present in other images. So they tend not to produce semantically coherent regions. Furthermore, the granularity of image partitions is important to train robust models for image annotation and retrieval. Our experiments show that with finer partitions of images, i.e. more number of regions per image up to some point, the models usually have better performance. Across all images in the training set, the models can collect more reliable statistics of the occurrences of image features with finer image partitions.

Compared with using segmentation, the benefits of using an image grid include a significant reduction in computational overhead and a simplification of the parameter estimation due to the fixed number of regions for each image. Furthermore, it is easier to incorporate structure information into the model using a grid representation. For example, the relative position may greatly help in distinguishing adjacent regions like sea and sky.

From each image region, we extract color and texture features similar to those used by [32, 8]. More details on the features used in the experiments is available in the experimental section 3.2.

3.1.2.1 Feature Generation Model

$P_{\mathcal{R}}(\cdot|J)$ is a density function responsible for generating the feature vectors $r_1^{\vec{}} \dots r_n^{\vec{}}$. It is estimated through a non-parametric kernel-based density. A kernel density estimate is essentially a kind of local regression method requiring little training. We use a non-parametric kernel density to estimate the probability $P_{\mathcal{R}}(\vec{r}|J)$ of generating the feature vector \vec{r} from image J . Let $\mathbf{r}_J = \{r_1^{\vec{}} \dots r_n^{\vec{}}\}$ be the set of region features of image J we estimate:

$$P_{\mathcal{R}}(\vec{r}|J) = \frac{1}{n} \sum_{i=1}^n \frac{\exp \left\{ -(\vec{r} - r_i^{\vec{}})^{\top} \Sigma^{-1} (\vec{r} - r_i^{\vec{}}) \right\}}{\sqrt{2^k \pi^k |\Sigma|}} \quad (3.8)$$

Equation (3.8) arises by placing a Gaussian kernel over the feature vector $r_i^{\vec{}}$ of every region of image J . Each kernel is parameterized by the feature covariance matrix Σ . As a matter of convenience we assumed $\Sigma = \rho \cdot I$, where I is the identity matrix¹. ρ plays the role of kernel *bandwidth*, which determines the degree of smoothing of $P_{\mathcal{R}}$ around the support point $r_i^{\vec{}}$. The value of ρ is selected empirically on a held-out portion of the training set \mathcal{T} . The actual value selected for ρ is related to the dimension and variance of features as well as the training set, and will be reported in the experiments section 3.2. The drawback of kernel estimation is that it has high online computational complexity.

¹We actually tried the diagonal matrix based on the variance of visterms corresponding to each annotation word, but the results are worse because there is not enough training data.



		
Models	Bill_Clinton, books, greenery, text_overlay	Bill_Clinton
Multinomial	0.25	1.0
Bernoulli	1.0	1.0

Figure 3.1. $P(\text{Bill_Clinton} \mid \text{image})$ under different models for two images with annotations of different lengths.

3.1.3 Modeling Captions

The continuous relevance model (CRM) assumes that annotation words for any given image are subject to a multinomial distribution. Here an alternative model, the multiple-Bernoulli model [43] is discussed and compared with the multinomial model.

3.1.3.1 Multiple-Bernoulli word model

A multinomial model distributes the probability mass between all the words in a given annotation, where each word may appear multiple times. The event space of a multinomial model is all the strings composed of words from a vocabulary. During the estimation process all the words will compete for the probability mass. Take as an example the two images in Figure 1.2 in the introduction section. For convenience, we repeat the figure here again (Figure 3.1). Both these two images contain Bill Clinton, but the first image is annotated with four key words “Bill_Clinton, books, greenery, text_overlay” and the second with only one “Bill_Clinton”. Using a multinomial model, the first image splits the probability mass equally between the four annotation words and $P(\text{Bill_Clinton} \mid I) = \frac{1}{4}$. On the other hand, the second image assigns all probability mass to the word “Bill_Clinton”, i.e. $P(\text{Bill_Clinton} \mid I) = 1$. The multinomial models capture the *prominence* of a word in the annotation and words with multiple occurrence are assigned higher probabilities. But arguably, since both images contain Bill Clinton, the probability of “Bill_Clinton” shouldn’t

be that different. Otherwise, the second image will be ranked much further ahead of the first image in response to the query “Bill_Clinton” although there is no real reason for preferring one over the other. This can be avoided by using a multiple-Bernoulli model.

A multiple-Bernoulli model explicitly focuses on the *presence* or *absence* of words in the annotation rather than on their prominence, which models the annotators’ behavior better: when people annotate an entire image, they care about what objects and environment exist in the image rather than their prominence. Using a multiple-Bernoulli model, both images will have the same probability of “Bill_Clinton” (equal to 1) since both of them contain Bill Clinton. By representing each word in the vocabulary as a binary variable, then each possible annotation of an image is a binary occurrence vector in $\{0, 1\}^V$, where V is the size of vocabulary. So the event space of the multiple-Bernoulli model is the set of all *subsets* of the given vocabulary. Individual components of each vector are assumed to be independent and identically (Bernoulli-) distributed given the particular image.

For most annotated image datasets, a Bernoulli model provides a closer match than the multinomial because of the following factors: i) no word is ever used more than once in a given annotation, so modeling word frequency is pointless. ii) most annotations have varying length, especially for those hierarchically annotated, e.g. the video datasets [2]. iii) words are usually assigned to the annotation based on the presence of an object in a image, not on its prominence. Our hypothesis is also supported by experimental results which will be discussed in section 3.2.

3.1.4 Multiple-Bernoulli Relevance Model

We describe the notations used in this section as follows:

1. \mathcal{V} : the annotation vocabulary.
2. \mathcal{T} : the training set of annotated images.
3. J : an image in \mathcal{T} .

4. $\mathbf{r}_J = \{r_1^\vec{\cdot} \dots r_n^\vec{\cdot}\}$: all feature vectors extracted from image regions of J .
5. $\mathbf{w}_J \in \{0, 1\}^\mathcal{V}$: the set of annotation words of J .
6. A : an un-annotated image A .
7. n_A : the number of image regions of A .
8. $\mathbf{r}_A = \{r_1^\vec{\cdot} \dots r_{n_A}^\vec{\cdot}\}$: the feature vectors of A .
9. \mathbf{w}_B : some arbitrary subset of \mathcal{V} .
10. n_B : the number of words in \mathbf{w}_B .

In our multiple-Bernoulli relevance model (MBRM), two distinct probability distributions $P_{\mathcal{V}}(\cdot|J)$ and $P_{\mathcal{R}}(\cdot|J)$ dominate the generation of J . According to the previous section, the annotation words \mathbf{w}_J of J are a subset of the whole vocabulary \mathcal{V} , represented as a binary occurrence vector. $P_{\mathcal{V}}(\cdot|J)$ is some underlying multiple-Bernoulli distribution from which every component of this binary vector is independently sampled. $P_{\mathcal{R}}(\cdot|J)$ is some underlying multi-variate density function from which the set of real-valued feature vectors r of dimension k is sampled. Each of the feature vectors represents an image region of \mathbf{r}_J .

Given an un-annotated image A represented as a set of region feature vectors \mathbf{r}_A , we would like to model the joint probability $P(\mathbf{r}_A, \mathbf{w}_B)$ of \mathbf{r}_A and some arbitrary word subset \mathbf{w}_B via an expectation over all images $J \in \mathcal{T}$. The overall process is as follows:

1. Pick a training image $J \in \mathcal{T}$ with probability $P_{\mathcal{T}}(J)$
2. Sample \mathbf{w}_B from a multiple-Bernoulli model $P_{\mathcal{V}}(\cdot|J)$.
3. For $a = 1 \dots n_A$:
 - (a) Sample a generator vector $r_a^\vec{\cdot}$ from the probability density $P_{\mathcal{R}}(\cdot|J)$.

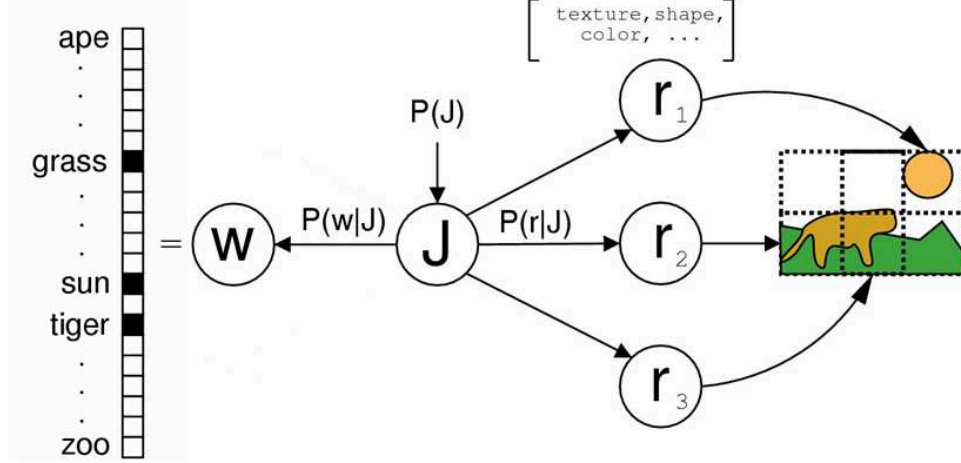


Figure 3.2. MBRM viewed as a generative process. The annotation \mathbf{w} is a binary vector sampled from the underlying multiple-Bernoulli model. First we randomly pick a training example J which generates two distributions $P(\mathbf{r}|J)$ and $P(\mathbf{w}|J)$. The image is produced by sampling a set of feature vectors $\{\vec{r}_1 \dots \vec{r}_n\}$, each of which represents an image region. Resulting regions are tiled to form the image.

Figure 3.2² shows a graphical dependency diagram for the generative process outlined above. We show the process of generating a simple image consisting of three regions and a corresponding 3-word annotation. Note that the number of words in the annotation n_B does not have to be the same as the number of image regions n_A . Formally, the probability of a joint observation $\{\mathbf{r}_A, \mathbf{w}_B\}$ is given by:

$$P(\mathbf{r}_A, \mathbf{w}_B) = \sum_{J \in \mathcal{T}} \left\{ P_T(J) \prod_{a=1}^{n_A} P_{\mathcal{R}}(\vec{r}_a | J) \times \prod_{v \in \mathbf{w}_B} P_V(v | J) \prod_{v \notin \mathbf{w}_B} (1 - P_V(v | J)) \right\} \quad (3.9)$$

Equation (3.9) allows us to annotate the image by finding that subset of vocabulary \mathbf{w}^* which is most likely to co-occur with the image:

$$\mathbf{w}^* = \arg \max_{\mathbf{w} \in \{0,1\}^V} \frac{P(\mathbf{r}_A, \mathbf{w})}{P(\mathbf{r}_A)} \quad (3.10)$$

²Modified from the MBRM diagram in [43]

In practice we only consider subsets of a fixed size. The maximization in equation (3.10) can be done very efficiently because of the factored nature of the Bernoulli component.

3.1.4.1 Simplifying the Computation

A simplification is made for the calculation of the joint probability of a single word with a set of region features $P(\mathbf{r}_A, w)$. This simplification is based on the assumption that when one associates a word with an image, he/she synthesizes all the situations of that word occurring in the image, i.e. all the possible word subsets containing that particular word.

Let \mathcal{V} again be the word vocabulary. Then a possible annotation is represented by a vector \mathbf{V} of length $|\mathcal{V}|$ where each element can be either 0 or 1 depending on whether that particular word is selected. We let Z be the entire annotation space obtained by taking the power set ($|Z| = 2^V$), and Z_w be the sub-space defined by word w . Z_w is noted as a set of word subsets S where $Z_w = \{S \in Z : w \in S\}$, and S is a subset of \mathcal{V} , represented as a binary vector and $s_i = 1$ if $v_i \in S$. That is Z_w is the set of all the subsets of \mathcal{V} which contain the word w . As mentioned above, we assume that when people annotate an image with one word w , they are actually annotating it with all the possible subsets of words, Z_w , each of which must contain the word w . Based on this assumption, we have³:

$$P(\mathbf{r}_A, w) = P(\mathbf{r}_A, Z_w) = P(\mathbf{r}_A, \bigcup_{S:w \in S} S)$$

Because P is additive in Z , which is the space of P , this gives:

$$P(\mathbf{r}_A, \bigcup_{S:w \in S} S) = \sum_{S:w \in S} P(\mathbf{r}_A, S)$$

³Courtesy of Victor Lavrenko for the discussion and an original proof.

Using equation (3.9), the RHS of the above equation may be written as:

$$\sum_J P(J) \prod_a P(\vec{r}_a|J) \sum_{S:w \in S} \prod_{v \in S} P(v|J) \times \prod_{v \notin S} (1 - P(v|J))$$

If we explicitly represent all the possible $S \in Z_w$ by enumerating whether each word occurs in S , we can rewrite the above expression as:

$$\begin{aligned} \sum_J P(J) \prod_a P(\vec{r}_a|J) & \sum_{s_1 \in \{0,1\}} \sum_{s_2 \in \{0,1\}} \dots \\ & \dots \sum_{s_w=1} \sum_{s_{w+1} \in \{0,1\}} \dots \\ & \dots \sum_{s_N \in \{0,1\}} \prod_{v=1}^N (P(v|J)^{s_v} (1 - P(v|J))^{1-s_v}) \end{aligned}$$

which may be expressed as:

$$\begin{aligned} \sum_J P(J) \prod_a P(\vec{r}_a|J) & \sum_{s_1 \in \{0,1\}} [P(v_1|J)^{s_1} (1 - P(v_1|J))^{1-s_1}] \\ & \sum_{s_2 \in \{0,1\}} [P(v_2|J)^{s_2} (1 - P(v_2|J))^{1-s_2}] \dots \\ & \dots \sum_{s_w=1} [P(v_1|J)^{s_w} (1 - P(v_1|J))^{1-s_w}] \dots \\ & \dots \sum_{s_N \in \{0,1\}} [P(v_N|J)^{s_N} (1 - P(v_N|J))^{1-s_N}] \end{aligned}$$

which can be rewritten as:

$$\begin{aligned} \sum_J & P(J) \prod_a P(\vec{r}_a|J) \prod_{i \notin w} \sum_{S_i \in \{0,1\}} [P(v_i|J)^{S_i} (1 - P(v_i|J))^{1-S_i}] \times P(w|J) \\ = & \sum_J P(J) \prod_a P(\vec{r}_a|J) \prod_{i \notin w} [P(v_i|J) + (1 - P(v_i|J))] \times P(w|J) \\ = & \sum_J P(J) \left(\prod_a P(\vec{r}_a|J) \right) P(w|J) \end{aligned}$$

Thus we have:

$$P(\mathbf{r}_A, \text{ "word } w\text{ "}) = \sum_J P(J) \left(\prod_a P(\vec{r}_a | J) \right) P(w | J) \quad (3.11)$$

We showed that the joint probability of a word and an image, which is represented as a set of visual features \mathbf{r}_A , is independent of the other words. Note the simplification arises from the assumption that when a person annotates an image with a word, she synthesizes all the situations of that words occurring in that image.

3.1.4.2 Estimating Parameters of the Multiple Bernoulli Model

In this section we will discuss simple but effective estimation techniques for the three components of the model: $P_{\mathcal{T}}$, $P_{\mathcal{V}}$ and $P_{\mathcal{R}}$. $P_{\mathcal{T}}(J)$ is the probability of selecting the underlying model of image J to generate some new observation \mathbf{r}_A , \mathbf{w} . In the absence of any task knowledge we use a uniform prior $P_{\mathcal{T}}(J) = 1/N_{\mathcal{T}}$, where $N_{\mathcal{T}}$ is the size of the training set. The distribution $P_{\mathcal{R}}(\cdot | J)$ is estimated by a Gaussian kernel density and the details are given in equation 3.8.

$P_{\mathcal{V}}(v | J)$ is the v 'th component of the multiple-Bernoulli distribution that is assumed to have generated the annotation \mathbf{w}_J of image $J \in \mathcal{T}$. Maximum likelihood estimates (MLE) are widely used for parameter estimations of various distributions including the multiple-Bernoulli distribution. However, the bias of MLE could be substantial with sparse training samples. According to the definition of the multiple-Bernoulli distribution, the subset of words \mathbf{w} associated with an image is distributed as:

$$P(\mathbf{w} | \theta) = \prod_{i=1}^{|\mathcal{V}|} \theta_i^{x_i} (1 - \theta_i)^{1-x_i} \quad (3.12)$$

where V is the size of the vocabulary and x_i indicates if word w_i occurs in a particular image and if so it is equal to 1 and 0 otherwise. The distribution P is parameterized by $\theta = \{\theta_1 \dots \theta_{|v|}\}$ for each word variable w_i . The MLE estimates of parameter θ_i are:

$$\begin{aligned}
\theta_i^{ML} &= \arg \max_{\theta_i} L(\theta_i) \\
&= \arg \max_{\theta_i} \log P(\mathbf{w}|\theta) \\
&= \arg \max_{\theta_i} (x_i \log \theta_i + (1 - x_i) \log(1 - \theta_i))
\end{aligned}$$

Maximizing by solving $\frac{\partial L}{\partial \theta_i} = 0$ gives $\theta_i^{ML} = x_i$, i.e. the binary indicator of occurrence of w_i in the image annotation.

The problem with this MLE estimation is that if a word doesn't occur in the annotation, the estimation predicts that it can never have been associated with that image. This could lead to a very biased estimation because it is not uncommon to see sparse data in the real world, especially for the image annotation task. The expense of manually annotating images, the annotator's preferences in describing an image, polysemy and synonyms, all these factors can prevent us from seeing sufficient data for an unbiased maximum likelihood estimation of θ .

Therefore, instead of an MLE we estimate $P_{\mathcal{V}}(v|J)$ using a Bayesian estimate, which solves the pathology through calculating the posterior $P(\theta|\mathbf{w})$ of the parameters θ given annotation \mathbf{w} with incorporation of priors of the parameters. A Beta distribution is selected to formulate the prior of θ_i because it is conjugate to the Bernoulli distribution, i.e. the posterior resulting from multiplying the prior and the likelihood is in the same family as the Beta prior.

The parameter θ is distributed according to a Beta distribution given by:

$$P(\theta) = \prod_i \left\{ \theta_i^{\alpha_i} (1 - \theta_i)^{\beta_i} \frac{\Gamma(\alpha_i + \beta_i + 2)}{\Gamma(\alpha_i + 1)\Gamma(\beta_i + 1)} \right\} \quad (3.13)$$

where Γ is the Gamma function. The parameters α_i and β_i act like ‘‘pseudo counts’’ of the occurrence and non-occurrence of word w_i in an un-observed Bernoulli sequence.

The Bayesian estimate of $P(\theta|\mathbf{w})$ is given by:

$$P(\theta|\mathbf{w}) = \frac{P(\mathbf{w}|\theta)P(\theta)}{\int_{\theta} P(\mathbf{w}|\theta)P(\theta)d\theta} \quad (3.14)$$

The numerator is equal to:

$$\prod_i \theta_i^{x_i} (1 - \theta_i)^{1-x_i} \theta_i^{\alpha_i} (1 - \theta_i)^{\beta_i} \frac{\Gamma(\alpha_i + \beta_i + 2)}{\Gamma(\alpha_i + 1)\Gamma(\beta_i + 1)} \quad (3.15)$$

The denominator is just the integral of the numerator and can be obtained by observing that:

$$\int_{\theta_i} \theta_i^{x_i + \alpha_i} (1 - \theta_i)^{\beta_i + 1 - x_i} d\theta_i = \frac{\Gamma(x_i + \alpha_i + 1)\Gamma(\beta_i + 2 - x_i)}{\Gamma(\alpha_i + \beta_i + 3)} \quad (3.16)$$

Therefore,

$$P(\theta|\mathbf{w}) = \prod_i \frac{\Gamma(\alpha_i + \beta_i + 3)}{\Gamma(\alpha_i + x_i + 1)\Gamma(\beta_i + 2 - x_i)} \theta_i^{\alpha_i + x_i} (1 - \theta_i)^{\beta_i + 1 - x_i} \quad (3.17)$$

Therefore, we have $P(\theta_i|w_i) \sim \text{Beta}(\alpha_i + x_i, \beta_i + 1 - x_i)$, which indicates that the observed occurrence or non-occurrence $x_i, 1 - x_i$ of w_i is accumulated on its "pseudo counts" α_i, β_i to form an updated Beta distribution. So far we obtained the distribution of the posterior $P(\theta|\mathbf{w})$. To collapse to a single point, it is common to pick the posterior mean. Based on the fact that the expected value of a standard Beta distribution $X \sim \text{Beta}(\alpha, \beta)$ is $E(x) = \frac{\alpha}{\alpha + \beta}$, we have:

$$P(w_i|J) = \tilde{\theta}_i = E[\theta_i|w_i] = \frac{\alpha_i + x_i}{\alpha_i + \beta_i + 1}$$

Since α and β act as "pseudo counts" of w_i in the annotation of an image, a plausible way is to select the parameter values based on the whole collection. We pick the parameter values as $\alpha_i = N_{w_i}/\mu$ and $\beta_i = ((N - N_{w_i})/\mu)$, where N_{w_i} is the number of training

images that contain w_i in the annotation and N is the number of images in the training set (see [168] for the Dirichlet case). μ here is a smoothing parameter. Substituting the selected values for the α_i and β_i in equation (3.18) yields:

$$P(w_i|J) = \frac{N_{w_i}/\mu + x_i}{N/\mu + 1} = \frac{\mu x_i + N_{w_i}}{\mu + N} \quad (3.18)$$

Let v be an arbitrary word in the vocabulary and $\delta_{v,J}$ indicate whether v occurs in J , the above equation could be rewritten as:

$$P(v|J) = \frac{\mu \delta_{v,J} + N_v}{\mu + N} \quad (3.19)$$

where $\delta_{v,J} = 1$ if the word v occurs in the annotation of image J and zero otherwise.

3.1.4.3 Ranked Retrieval with the Bernoulli Model

Relevance modeling approaches rank images according to the probability $P(\mathbf{w}_Q|M_I)$ of generating the query \mathbf{w}_Q by a language model M_I associated with image I . The calculation of the probability $P(\mathbf{w}_Q|M_I)$ differs significantly from model to model, mainly based on the representation of the query \mathbf{w}_Q . In a multinomial relevance model, the query is represented as a sequence of variables and so $P(\mathbf{w}_Q|M_I)$ is estimated under the multinomial language framework under equation 3.4. But in MBRM the query \mathbf{w}_Q is represented as a binary vector over the entire vocabulary, so the retrieval model for MBRM should be based on the multiple-Bernoulli language modeling approach proposed in [121]. According to this retrieval model, the images I are ranked by:

$$P(\mathbf{w}_Q|M_I) = \prod_{w \in \mathbf{w}_Q} P(w|M_I) \prod_{w \notin \mathbf{w}_Q} (1 - P(w|M_I)) \quad (3.20)$$

However, our experimental results in section 3.2 show that although MBRM produces better annotation results than CRM, using the MBRM with the Bernoulli retrieval model gives poor retrieval results. In practice, we found that a combination of MBRM with the multinomial retrieval model outperforms the purely multinomial model (CRM) for

retrieval. This is not a "clean" model because we assume that words follow a multiple-Bernoulli distribution for annotation but a multinomial distribution for retrieval. The next section shows how to construct a modified multinomial model (the normalized continuous relevance model) which has the annotation performance of the Bernoulli model and the retrieval performance of a multinomial model.

3.1.5 Normalized CRM

This section describes an alternative to the MBRM, called the normalized Continuous Relevance Model, which performs the same as MBRM on the task of annotation but achieves excellent retrieval performance.

The normalized CRM [77] is a modification of the CRM and bears the same formulation for the joint distribution $P(\mathbf{w}, \mathbf{r})$ as equations (3.1) and (3.10), where \mathbf{w} is an arbitrary set of words and $\mathbf{r} = \{r_1^{\vec{r}} \dots r_n^{\vec{r}}\}$ a given image with regions.

Under the multinomial assumption, the original CRM [78] estimated the annotation probability $P(w|J)$ based on the relative frequency of the word w in the annotation of image J :

$$P(w|J) = \lambda \frac{N_{w,J}}{N_J} + (1 - \lambda) \frac{N_w}{L} \quad (3.21)$$

Here $N_{w,J}$ is the number of times w occurs in the annotation of J and N_J the length of the annotation for image J . The probability is smoothed by the background probability of w over the whole training set. The degree of smoothing is denoted by the parameter λ . N_w is the total number of times w occurs in the training set, and L the aggregate length of all training annotations.

In section 3.1.3.1, we have described the problem of multinomial model for annotation words. For images which contain the same object described by the annotation word "w", if the lengths of their annotations vary a lot, the probabilities of word "w" given these images will be significantly different. Arguably, there is no real reason for the large difference between these probabilities since all the images contain "w". An alternative to using a

multiple-Bernoulli model is to (pad) all annotations to a fixed length $N^* = \max_J \{N_J\}$ by adding $(N^* - N_J)$ instances of a special “null” word to the annotation of image J . We refer to this variation of the model as *Normalized-CRM* and demonstrate that it achieves substantially better retrieval performance than the original CRM.

For the feature component of the Normalized-CRM, we also use a non-parametric kernel-based density estimate for the distribution $P(\vec{r}|J)$, just as the CRM and the multiple-Bernoulli model do in equation (3.8).

We can show that the joint distributions in MBRM and NCRM differ only by a constant factor - the length of each image annotation in the Normalized-CRM model, and hence they have the same annotation performance.

The multinomial retrieval model [31, 50, 57] is more appropriate for the CRM and Normalized CRM models since it assumes a multinomial distribution for the annotation words:

$$P(\mathbf{w}_Q|M_I) = \prod_{w \in \mathbf{w}_Q} P(w|M_I) \quad (3.22)$$

The language model $P(w|M_I)$ may be calculated using $P(w|M_I) = P(w, \mathbf{r}_I)/P(\mathbf{r}_I)$, where $P(w, \mathbf{r}_I)$ is the joint distribution for every word w and $P(\mathbf{r}_I)$ is calculated by marginalizing the joint distribution.

3.1.5.1 Relation of Normalized CRM to MBRM

We now demonstrate that the Normalized-CRM gives the same annotation performance as the MBRM and the annotation probabilities produced by the two models differ by a constant factor. In section 3.1.4.1, we showed that we could use MBRM to annotate each word separately. Both MBRM and normalized CRM, therefore, compute the joint distribution for a single word

$$P(w, \mathbf{r}) = \sum_{J \in \mathcal{I}} P(J)P(w|J) \prod_{\vec{r} \in \mathbf{r}} P(\vec{r}|J) \quad (3.23)$$

The feature probabilities $\prod_{\vec{r} \in \mathbf{r}} P(\vec{r}|J)$ are identical for both models and hence we only need to consider the word probabilities $P(w|J)$.

1. $P(w|J)$ in MBRM:

According to equation (3.19):

$$P(w|J) = \frac{\mu \delta_{w,J} + N_w}{\mu + N} \quad (3.24)$$

where μ is a smoothing parameter, $\delta_{w,J} = 1$ if the word w occurs in the annotation of image J and zero otherwise. N_w is the number of training images that contain w in the annotation and N is the total number of training images. Let $\lambda_b = \frac{\mu}{\mu+N}$, then $1 - \lambda_b = \frac{N}{\mu+N}$. Let $\delta_{w,C}$ represent the frequency of word w in the whole training set C , then $\delta_{w,C} = N_w$ since in image/video annotations, every word is either absent or occurs only once for each image or frame. We, therefore, have

$$P(w|J) = \lambda_b \delta_{w,J} + (1 - \lambda_b) \frac{\delta_{w,C}}{N} \quad (3.25)$$

2. $P(w|J)$ in Normalized-CRM

From equation (3.21),

$$P(w|J) = \lambda \frac{N_{w,J}}{N_J} + (1 - \lambda) \frac{N_w}{L} \quad (3.26)$$

In the NCRM all annotations are padded to the same length, say $|a|$, so for each image the annotation length N_J is the constant $|a|$ ⁴. Again, since for image/video

⁴Usually $|a|$ is the size of the vocabulary

annotations, a word is either absent or occurs exactly once in the annotation of an image, we have $N_{w,J} = \delta_{w,J}$, $N_w = \delta_{w,C}$ and $L = |a|N$. We, therefore, have:

$$P(w|J) = \frac{1}{|a|}[\lambda\delta_{w,J} + (1 - \lambda)\frac{\delta_{w,C}}{N}] \quad (3.27)$$

which is the same as for the MBRM model except for the constant factor $|a|$.

We have thus shown that the joint distributions in the two models differ only by a constant factor - the length of each image annotation in the Normalized-CRM model.

Assume that the joint probability for word w_i and image $\{J = \mathbf{r}\}$ using the Bernoulli model is, $P(w_i, \mathbf{r}) = k_i$. Then, the annotation probability using the Bernoulli model is given by:

$$P(w_i|\mathbf{r}) = \frac{P(w_i, \mathbf{r})}{P(w_i, \mathbf{r}) + P(\tilde{w}_i, \mathbf{r})} = k_i \quad (3.28)$$

where \tilde{w}_i stands for (not w_i) (for example not face for face).

The annotation probability for a word for the Normalized-CRM model is given by:

$$P(w_i|\mathbf{r}) = \frac{P(w_i, \mathbf{r})}{P(\mathbf{r})} = \frac{P(w_i, \mathbf{r})}{\sum_j P(w_j, \mathbf{r})} = \frac{k_i|a|}{\sum_j k_j|a|} = \frac{k_i}{\sum_j k_j} \quad (3.29)$$

Since $\sum_j k_j/|a| = 1$ we have $\sum_j k_j = |a|$ which is a constant and hence $P(w|\mathbf{r}) = k_i/|a|$.

Hence the annotations produced by the models are identical and the probabilities produced by the Normalized-CRM model differ from those produced by the multiple-Bernoulli relevance model by a constant factor given (usually) by the length of the vocabulary. This is a nice result. As a practical side effect, to compute the Normalized-CRM we just compute the annotations using the Bernoulli Relevance Model. If necessary, we can obtain the probabilities by dividing by the constant factor. For most practical purposes including retrieval we can usually ignore this constant factor. Formalizing the Normalized-CRM allows us to use these annotations with a multinomial retrieval model.

3.2 Experiments

This section discusses annotation and retrieval results comparing the annotation and retrieval performance of a number of different models on 5 different datasets. Subsection 3.2.1 mentions the details of the datasets used while subsections 3.2.3 and 3.2.4 discuss the results of running different algorithms.

3.2.1 Datasets

Five different datasets were used in all. The first two were from the Corel image data set and were exactly the same as that used in Duygulu *et al.* [32] and in Barnard *et al.* [9] respectively.⁵ One subset of video keyframes from NIST’s TRECVID 2003 [2] was also used for testing. Finally, a large scale test was done using the entire TRECVID 2003 dataset.

We first used the Corel dataset in [32] to be able to directly compare with their results. This dataset consists of 5000 images from 50 Corel Stock Photo CD’s. Each CD includes 100 images on the same topic, and each image is also associated with 1-5 keywords. Overall there are 371 keywords in the dataset. In the experiments, we divided this dataset into 3 parts: a training set of 4000 images, a validation set of 500 images and a test set of 500 images. The validation set is used to find model parameters. After finding the parameters, we merged the 4000 images in the training set and 500 images in the validation set to form a new training set. This corresponds to the training set of 4500 images and the test set of 500 images used by Duygulu *et al.* [32]. There are 260 words present in the test set. Refer Appendix A for these 260 annotation words. Most annotation words in this dataset denote objects, scenes or other directly visual-related concepts. Theoretically the models proposed in this work can be used for retrieving more general or abstract queries, e.g. “*peace, intentions*”. But it is still a well-known open problem on how to reduce the

⁵We thank Kobus Barnard for making the Corel datasets available at: http://www.cs.arizona.edu/people/kobus/research/data/eccv_2002 and http://kobus.ca/research/data/jmlr_2003/

semantic gap between computational image representation and human understanding of images. This work did not focus on developing high-level feature representations to tackle this problem, but based on statistical models it provides a way to associate the low level image features to semantic concepts.

For a more complete comparison with the models of Barnard et al in [9], we then tested our model on the exact same dataset as in [9], which is a larger Corel image dataset consisting of 160 CD's, each of which includes 100 images on the same topic. The configuration of the dataset for testing our model is also exactly the same as in [9]. From the 160 CD's, 80 CD's were drawn as a sample and further divided into training (75%) and "standard" held-out (25%) sets. The remaining CD's formed a more difficult "novel" held-out set. Final results were averaged over 10 such random samples. The vocabulary of each sample contains 150 ~ 170 words. To find system parameters for our model, we used the training set of the first sample. We trained our model on the first 4188 images of it and tuned parameters on the last 1000 images.

Another dataset we used is from NIST's TRECVID 2003 dataset. This dataset consists of a set of mpeg files. Each file is a 30 minute section of CNN or ABC news (including advertisements). NIST provided shot segmentations. The participants in TREC annotated a portion of the videos. The word vocabulary for human annotation is represented as a hierarchical tree with each annotation word as a node, which means many key frames are annotated hierarchically, e.g. a key frame can be assigned a set of words like "face, male_face, male_news_subject". This means that the annotation length for key frames may vary widely.

The first dataset (small video set) consists of 12 of the mpeg files. 5200 key frames were extracted using NIST provided shot segmentations. There are 137 keywords in the whole dataset after ignoring the audio annotations. The dataset was randomly divided into a training set (1735 key frames), a validation set (1735 key frames) and a test set (1730 key

frames). As for the Corel set, the validation set is used to find system parameters, and then merged into the training set after we find the parameters.

We then tested our algorithms on a larger subset from NIST’s TRECVID dataset. This set consists of 39 mpg files, 14,202 key frames in total. In this dataset the key frames for the training and test sets are obtained from different mpeg files. That is, the training and test sets are separated in time. There are 9,415 key frames from 26 mpg files in the training set and 4,787 key frames from the 13 other mpeg files in the test set.

Finally tests were run on NIST’s entire TRECVID development dataset containing 58 mpeg files of ABC World News Tonight and 57 mpeg files of CNN Headline News, about 44,100 key frames in total. In the final test of our algorithms, the set is divided into 45 hours of training data and 15 hours of test data, with 34,880 key frames for training and 9,220 key frames for test. Also the training set and the test set are separated in time. Each key frame in the TRECVID3 development dataset has been manually annotated with key words from about 100 semantic concepts, from which about 75 concepts are selected in our experiments to guarantee that each of them has more than 20 training examples in the development set.

3.2.2 Features Used

For the small Corel dataset, one experiment using CRM is run with the original segmentations and features as in Duygulu *et al.* [32] to make a direct comparison possible. For the large Corel dataset, all experiments used the segmentations and features provided by Barnard *et al.* [9] to fairly compare the performance of our model and their models in [9]. Images in both of these two Corel sets were automatically segmented using the Normalized-Cut algorithm in their original settings as in [32] and [9]. The feature set consists of 6 area, position and shape features, 24 color features and 16 texture features (see Barnard *et al.* [9]). Refer Appendix B for the feature details.

For all other experiments, every image is partitioned using a rectangular grid, and a feature vector is then calculated for every grid region. The number of rectangles is empirically selected using the training and validation sets. We tested different numbers and selected the one with the best performance on the validation set. We didn't do a thorough sweeping for tuning the number of rectangles per image because the model's computational complexity dramatically increases with the number of regions per image. The number of regions we used for the final test is 24 for the Corel set, and 35 for the video dataset sets. For the Corel dataset, 30 features are used (18 color features and 12 texture features). The color features for an image region include the average, the standard deviation and the skewness of the pixel values for each channel of the L^*a^*b color space. The texture features consist of Gabor energy computed over 3 scales and 4 orientations. Separate values of bandwidth β (equation 3.8) are used for the color features and for the texture features from the Corel set and the values are selected empirically using a validation set.

For the large video dataset and the entire TRECVID development dataset, the features used include 12 moments computed in the L^*a^*b color space and 20 gray level co-occurrence matrices ⁶ (refer Appendix B for the feature details).

3.2.3 Results of Automatic Image Annotation

In this section we evaluate and compare the performance of MBRM on automatic image annotation. Given an un-annotated image or key frame, we can calculate the probability of generating every candidate word in the vocabulary conditioned on the image. For the Corel set, we take the top 5 words (according to probability) as an automatic annotation of that image. For the video set, we take the top 6 words (the average length of human annotations over all key frames). Experiments are performed on the same dataset with identical preprocessing, features and training sets. Results show that MBRM performs better than CRM (although on average their performance is close to each other).

⁶The features were kindly supplied by Giridharan Iyengar at IBM Research





Image	CRM	MBRM	Groundtruth
	clouds plane jet water sky	water sky bridge boat tree	bridge tower water
	tree plane water zebra herd	sand water people beach umbrella	beach kauai people water
	non-studio_setting people sport_event basketball face	non-studio_setting people sport_event basketball face	non-studio_setting sport_event basketball
	face male_face indoors news_subject_monologue male_news_person	face indoors news_subject_monologue female_face female_news_person	indoors studio_setting face graphics_and_text_overlay monologue

Figure 3.3. Top automatic annotations produced by the CRM and MBRM models, with ground truth words correctly predicted marked in blue. MBRM performs better than CRM for the first two images and the fourth image. For the third image the annotations are identical. Note that many video frames are annotated with the words `graphics_and_text` and `text_overlay` because of the station logos - difficult to see in these images. Interestingly, some of the automatic labels do not correspond to human labels but are perfectly reasonable e.g `sky` in the first image, `sand` in the second one and `people` in the third.

Figure 3.3 shows examples of the automatic annotations obtained using CRM and MBRM models on the TREC Video set. These results are obtained on the same dataset with identical preprocessing, features and training sets. Note that MBRM performs better than CRM (although on average their performance is close to each other).

Figure 3.4 shows some examples of automatic annotations and probabilities obtained using CRM and MBRM models on the Corel dataset and TREC Video. For the first image, MBRM performs better than CRM. For the second image, they have similar performance.



Image	CRM		MBRM		Ground Truth
	tree	0.28	water	0.89	fox
	fox	0.18	fox	0.86	ice
	den	0.18	river	0.86	water
	elephant	0.12	arctic	0.86	river
	water	0.09	sky	0.03	
	
	graphics_and_text	0.21	graphics_and_text	0.95	outdoors
	text_overlay	0.20	text_overlay	0.95	crowd
	monologue	0.17	monologue	0.92	graphics_and_text
	crowd	0.13	crowd	0.91	text_overlay
	physical_violence	0.10	physical_violence	0.91	monologue
	riot	0.10	riot	0.91	physical_violence
	riot

Figure 3.4. Examples of annotation and probabilities for CRM and MBRM, with words correctly predicated marked in blue.

Models	Co-occurrence	Translation	CMRM	CRM-Seg	CRM	MBRM
#Words Recall > 0	19	49	66	107	119	122
Results on 49 best words, as in[32, 66]						
Mean Recall/Word	–	0.34	0.48	0.70	0.75*	0.78*
Mean Precision/Word	–	0.20	0.40	0.59	0.72*	0.74*
Results on all 260 words						
Mean Recall/Word	0.02	0.04	0.09	0.19	0.23*	0.25†
Mean Precision/Word	0.03	0.06	0.10	0.16	0.22*	0.24†

Table 3.1. Performance comparison on the task of automatic image annotation on the small Corel dataset. CRM-Seg refers to CRM with segmentations while CRM is the same model with a grid. MBRM performs best outperforming CRM by a small amount. Symbol * indicates a significant improvement over the CRM-Seg, and † indicates a significant improvement over both CRM-Seg and CRM.

However, in both cases the probabilities obtained from MBRM are easier to interpret. In CRM the probability of a word given an image must be compared with those of all other words to decide whether it is an important annotation because all words compete for the probability mass. In contrast, in MBRM the probability of a word given an image explicitly tells us if the word appears in that image. For example, it is not clear whether a 0.1 probability of water in the CRM model is significant or not while in the MBRM case if sky gets a probability of 0.03 it is clear that the possibility of sky in the image is very low.

The first evaluation on annotation is done as in [32, 66, 78] using recall and precision calculated for every word in the test set. For this part of the process we do not use the actual rankings. Let A be the number of images automatically annotated with a given word, B the number of images correctly annotated with that word, C the number of images having that word in the ground-truth annotation. Then $recall = \frac{B}{C}$, and $precision = \frac{B}{A}$. To evaluate the system performance, recall and precision values are averaged over the testing words. The first set of results are shown for the Corel dataset in Table 3.1. Results are reported for all (260) words in the test set. Annotation performance is also reported for the top 49 annotations to make a direct comparison with the results reported in [32] for IBM Translation Model 2. The three relevance model approaches are clearly much better than the translation model approach in [32] with MBRM outperforming all other models (4 times better than the translation model). Both CRM and CRM-Seg are identical except for the fact that CRM uses regions partitioned into rectangles while the regions in the CRM-Seg model are obtained using normalized cuts segmentation. As the results show this improves the performance significantly (almost 38% improvement in precision). Segmentation is a difficult error prone process in computer vision. Each image is segmented on its own without reference to any of the other training images. Since the probabilistic model deals with regions as entities, it cannot undo segmentation errors (if for example two distinct image regions are combined together in the segmentation). If we start from a rectangular partition (at a finer granular level), the probabilistic model which learns from multiple training images has a better chance of associating the rectangular regions with the correct words. We believe that this accounts for the better performance using a rectangular partition.

To have a fair and complete comparison with the models reported by Barnard *et al.* [9] on the automatic annotation task, we tested our MBRM on the same data set using the same features, data configurations and evaluation measurements as in [9]. To measure how well a model predicts words, [9] proposed using prediction score (PR) and normalized classification score (NS). The former is based on predicting the same number of words for each

Measurements	Held-out data			Novel data		
	PR	NS	KL	PR	NS	KL
Best Results in [9]	0.298	0.604	0.747	0.249	0.506	0.268
MBRM	0.371	0.647	1.129	0.255	0.514	0.274

Table 3.2. Performance comparison on automatic annotation between MBRM and the models reported in [9]. Three different measurements are used as in [9]: the prediction score (PR), the normalized classification score (NS) and the reduction of the KL-divergence from that computed using the empirical distribution (KL). Results show that MBRM consistently outperforms better than all the models reported in [9] on the automatic annotation task.

test image as the number of associated ground truth words with it, while the latter is based on predicting all words which exceed a certain probability threshold. PR is basically the prediction accuracy on the predicted words for a image and NS is basically the normalized correct and incorrect classifications. Let N be the vocabulary size, n the number of ground truth words for the image, c the number of correctly predicted words, and e the number of incorrectly predicted words. Then for an image, NS is calculated as $\frac{c}{n} - \frac{e}{N-n}$ and PR is calculated as $\frac{c}{n}$. To measure the quality of word posterior distribution, [9] proposed using the reduction of the KL-divergence (KL) between the computed predictive distribution and the target distribution from that between the empirical word distribution and the target distribution. For the target distribution, they simply assume that the ground truth words should be predicted uniformly and all other words should not be predicted at all. The average values of these performance measurements over all the images are reported. Table 4.4 reports the results of MBRM on the large Corel data set using those three measurements for automatic annotation. It should be noted that the best results by Barnard *et al.* [9] in Table 4.4 are not from the same model, e.g. the best NS in [9] for held-out data is from their "D-2"(D for dependent) model, while the best NS in [9] for novel data is from their "I-0"(I for independent) model and the best KL in [9] for novel data is from their "D-0" model. The comparison with the models in [9] shows our MBRM consistently outperforms all those models on both the "standard" held-out data and the "novel" data.

Unlike those performance measurements used in [9], which essentially measures the mean precision/recall per image, mean precision per word and mean recall per word are more retrieval-related, i.e. they measure how well the automatic annotation supports the retrieval task. The annotation measurements based on mean precision/recall per image sometimes can give misleading results. For example, simply assigning high frequent background words to each testing image can be a tricky way to achieve high mean precision/recall per image, although the annotation system may mess up most other object words for those testing images. Furthermore, the mean precision and recall per word have become the standard measurements for annotation performance. Since our ultimate goal is image retrieval using key words, we also report the precision and recall of the MBRM on this larger Corel dataset, and for all other experiments we only report the performance using recall and precision. Table 3.3 shows the mean precision and mean recall of MBRM on the large Corel dataset.

	Held-out data	Novel data
Average #Words in test sets	161	148
Average #Words recall > 0	123	62
Results on all words in test sets		
Mean Recall/Word	0.256	0.078
Mean Precision/Word	0.260	0.067
Results on all words with recall > 0		
Mean Recall/Word	0.336	0.185
Mean Precision/Word	0.344	0.158

Table 3.3. Mean recall and mean precision of MBRM on the large Corel data set [9] for the automatic annotation task.

Figure 3.5 shows some annotation examples of using MBRM over the Corel image set, in which the automatically predicted words are quite different from the ground truth annotation. The first image shows the failure because of the inconsistency of human annotation. Note the MBRM does generate a set of reasonable annotations for the image, e.g. *clouds*, *coast*, *sunset* and *water*, but the human annotator ignored these words and selected *light* and *shore* which are more rarely used in the training set. For the second image, we



			
MBRM	clouds beach coast sunset wa- ter		snow fox shrubs water sky
Ground truth	light shore		bulls elk field snow

Figure 3.5. Negative annotation examples of using MBRM over the Corel image set.

found that the contributions to the probabilities of *snow*, *fox* and *shrubs* are from a large number of training images which share very similar backgrounds but contain a *fox* rather than an *elk*. On the other hand, most training images of *elk* have different background so they contribute little to the probability of *elk* for the second image. This example shows the important role of background context in relevance models. Although they can provide very useful information to distinguish objects, sometime they make it difficult to correctly predict objects appearing in an unusual environment. For example, we mentioned at the beginning of this chapter that *tiger* rarely appears in an office environment. But if this does happen – assuming a *toy tiger* on the desk of an office, the MBRM may fail to predict the *tiger*. In this case we believe better image features and more training images will help solve this problem. Better image features may distinguish tiger from other objects easily, and larger training sets may train the model better to identify a common background and weight it less for probability contributions.

3.2.4 Ranked Retrieval with Single Word Queries

The annotation results reported above ignore rank order. That is, imagine that one wanted to find all car images. One would ideally like to rank these according to the probability of annotation and hope that the top ranked ones are all cars. In fact, in large databases most users are not likely to even want to see more than 10 or 20 images in response to a query. Rank order is, therefore, very important for such applications. Figures 3.6-3.7 show

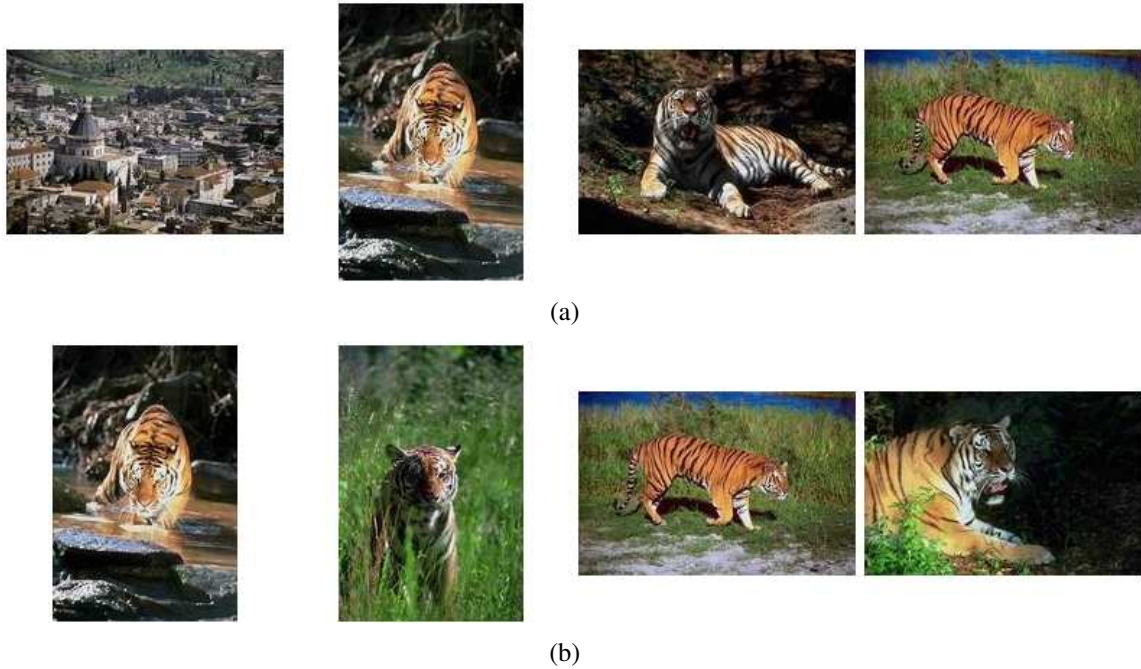


Figure 3.6. First 4 ranked results for the query “tiger” in the Corel dataset using a) CRM and b) MBRM.



Figure 3.7. First 4 ranked results for the query “addressing”. According to the ground truth the first and the third returned by the CRM are relevant, but the other two are not. For the MBRM, all the four images are relevant.

the performance of CRM and MBRM in response to one word text queries. The first image in Figure 3.6 is an Israeli city view. We found both CRM and MBRM wrongly predict

words “cat tiger” for this image because the training set lacks images with similar city or building views but has many tiger images with similar colors and texture. But its positions in the ranked lists are quite different for CRM and MBRM – CRM ranks it at the first place and MBRM ranks it at the ninth place. Although the annotation performance of the two models does not seem to be that different, the results below show that the retrieval performance can be very different.

Models	Corel Dataset		Small Video Dataset	
	All 260	Recall > 0	All 107	Recall > 0
CRM	0.26	0.30	0.25	0.29
MBRM	0.30	0.35	0.29	0.37
P-value	0.000037	0.0000082	0.000013	0.000000026

Table 3.4. Ranked retrieval results (in terms of mean average precision) based on one word queries. MBRM performs much better than the multinomial model [CRM]. The second row lists which words are used as queries. The last row gives the P-value produced by the sign test showing that the performance improvement is statistically significant.

We use a standard information retrieval metric called mean average precision to evaluate the retrieval performances. Average precision is the average of precision values at the ranks where relevant (here ‘relevant’ means that the ground-truth annotation of this image contains the query word) items occurs. This is further averaged over all queries to give mean average precision. Table 3.4 shows that for ranked retrieval the Bernoulli model substantially outperforms (by 15% for the Corel dataset and by 16% for the small Trecvid dataset) the multinomial model.

3.2.5 Experiments Using the Retrieval Model for Multiple Word Queries

For one word queries, images may be ranked according to the annotation probabilities of the query word given each image or keyframe. The previous section shows the experimental results of image ranking for single word queries directly according to the annotation probabilities. However, for multiple word queries the images should be ranked by applying language models over the annotation probabilities. The form of the language models depends on how we represent the queries. This section reports the retrieval performance

with one word or multiple-word queries using different retrieval models. Given a text query and a collection of un-annotated key frames, then our goal is to return all the relevant key frames, ranked according to the probabilities obtained using our retrieval model. In our retrieval experiments, we use three sets of queries ⁷ constructed from all 1-, 2-, 3- combinations of words which occur at least 10 times in the testing set. For each set of queries, we do comparative experiments using both CRM and the Bernoulli retrieval model. An image is considered relevant to a given query if its manual annotation contains all the query words. Evaluation metrics are precision at 5 retrieved key frames and non-interpolated average precision, averaged over the entire query set. These two different metrics are good measures suitable for the distinct needs of casual users and professional users.

Table 3.5 compares the performance of CRM and the Bernoulli model for three sets of queries. We observe that the Bernoulli retrieval model performs really poorly in comparison with the CRM although it has better annotation performance. What seems unusual is that the performance of MBRM on one word queries in Table 3.5 is much worse than the performance on one word queries reported in Table 3.4. Note that this is because in Table 3.4 the ranking is based directly on annotation probability while in Table 3.5 the values of MBRM is obtained by applying the multiple-Bernoulli retrieval model over multiple-Bernoulli annotation probabilities. Based on the comparisons, it may be inferred that the multinomial retrieval model is better in modeling a user's retrieval behavior: when a user retrieves images using a query, she/he doesn't care about whether the images contain the objects referred by the words which are not included in the query. But for multiple-Bernoulli retrieval models, they explicitly consider the presence or absence of every word in the images.

Closer examination reveals that the one word query retrieval used in Table 3.4 ranks the images according to the probability of the annotation of a single word. This is essentially

⁷Given that we used only a subset of TRECVID it did not make sense to use TRECVID queries

Query length	1 word	2 words	3 words
Number of queries	107	431	402
Precision at 5 retrieved key frames			
CRM	0.36	0.33	0.42
MBRM	0.21	0.16	0.17
Normalized CRM	0.49*	0.47*	0.58*
Mean Average Precision			
CRM	0.26	0.19	0.25
MBRM	0.08	0.07	0.09
Normalized CRM	0.30*	0.26*	0.32*

Table 3.5. Retrieval performance of different algorithms on the small video dataset for different query lengths. Symbol * indicates that the result is statistically significant better than CRM.

equivalent to using a multinomial retrieval model. On the other hand in Table 3.5 the multiple Bernoulli annotations are ranked using the Bernoulli retrieval model which takes the product of the probability of the query word and one minus the probability of words absent from the query. This means that words absent from the query dominate the model. It may also be a function of the fact that our estimate for many of these words is poor. That is, the Bernoulli retrieval model expects us to produce good estimates for all the annotations not only the most likely ones.

Clearly, this shows that for one word queries, the multinomial retrieval model performs a lot better with even the multiple-Bernoulli annotation model. Further, the multiple Bernoulli annotation model combined with the multinomial retrieval model does much better than using the purely multinomial (CRM) model for both retrieval and annotation. This was the rationale for creating the Normalized-CRM model.

Table 3.5 also reports the results of doing ranked retrieval using a multinomial retrieval model and the Normalized-CRM model. The results show that there is a substantial improvement in retrieval using the Normalized-CRM with the multinomial retrieval model. Specifically, Normalized-CRM outperforms CRM by 15%, 37% and 28% on the 1-, 2- and 3-words query sets respectively, For all three query sets the differences in precision are

statistically significant according to the sign test. The precision at 5 retrieved key frames also indicates that Normalized-CRM significantly outperforms CRM.

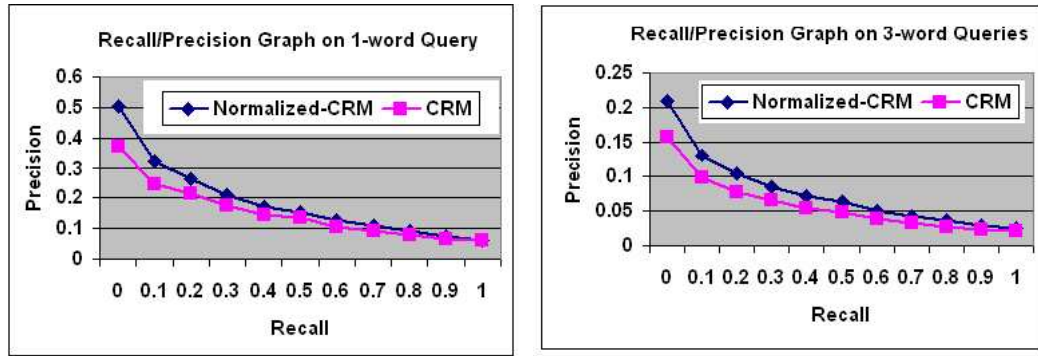
Performance on the larger video dataset (Table 3.6) where the training and test datasets are separated in time is also good. Normalized-CRM again outperforms CRM by 21%, 29% and 40% for 1-, 2- and 3-word queries respectively.

The Large Video Dataset			
Query length	1 word	2 words	3 words
Number of queries	115	431	682
Precision at 5 retrieved key frames			
CRM	0.18	0.11	0.09
MBRM	0.06	0.05	0.05
normalized CRM	0.28*	0.20*	0.15*
Mean Average Precision			
CRM	0.14	0.07	0.05
MBRM	0.03	0.02	0.02
Normalized-CRM	0.17*	0.10*	0.08*

Table 3.6. Retrieval performance of CRM, MBRM and Normalized-CRM on the large video dataset for different query lengths. Symbol * indicates that the result is statistically significant better than CRM.

While the average precision indicates how the algorithms perform on average, a recall-precision curve shows how the algorithms behave at different recall levels. Figure 3.8 shows the recall precision graphs for queries of different lengths. Again the graphs indicate that Normalized-CRM consistently outperforms CRM at all recall levels. These graphs also indicate that at the high ranks, the precision is quite high.

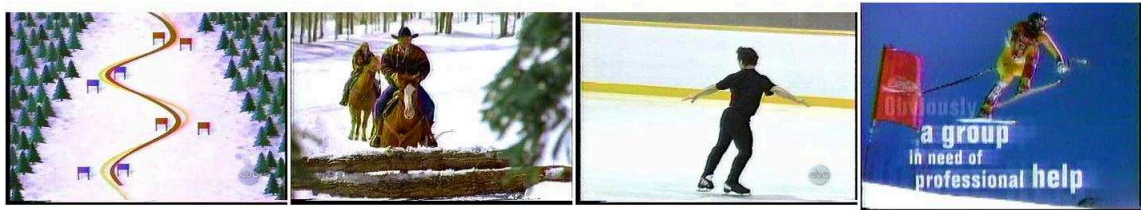
Figure 3.9 shows the top 4 images in rank order using both CRM and NCRM corresponding to the text query “outdoors snow person” on the small video set. Among the top 4 ranked images by the CRM the second and the fourth are relevant according to the ground truth. The first image is actually a picture of a ski run. The third one shows a person skating indoors. In contrast, all the top 4 ranked images by NCRM are relevant according to the human annotation.



(a)

(b)

Figure 3.8. Recall/Precision graphs the large video dataset with 1- and 3-word queries



(a) CRM Results



(b) Normalized CRM Results

Figure 3.9. First 4 ranked results for the query “Outdoors, Snow, Person”.

Based on the performance comparisons above, we tested the retrieval performance of the NCRM on the entire development dataset of TRECVID 2003. The mean average precision achieved is 0.158 based on color and texture features. Figure 3.10 compares recall-precision graphs for NCRM and the IBM translation model one (this was shown to outperform other translation models) [122]. For low recall (i.e. for the top documents) the NCRM clearly performs substantially than the translation model. In the next chapter, we will update this figure with a more comprehensive comparison with Markov random field and other models.

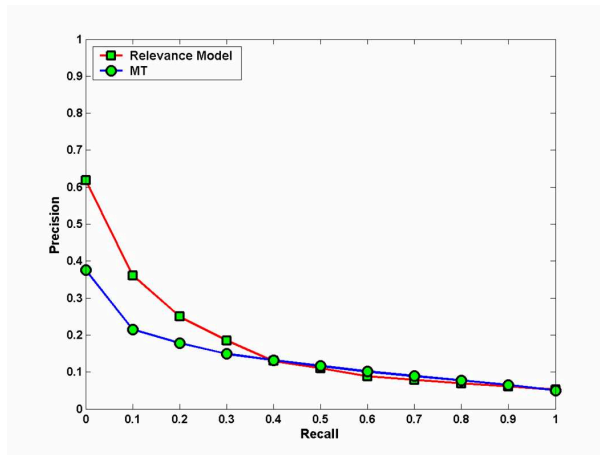


Figure 3.10. Recall-precision graphs for the NCRM, the machine translation (MT) model.

In this chapter, we proposed the Multiple-Bernoulli relevance model (MBRM) and the normalized continuous relevance model (NCRM) for image annotation and retrieval, and demonstrated that they outperform many previous models. However, all the relevance modeling approaches for images have some common limitations. Here we highlight two limitations which we consider to be important. First, relevance models assume that the visual features/visterms of a test image are independent given a training image and treat them as *a bag of visterms*. This assumption prevents relevance modeling approaches from utilizing the feature dependency information among images, which is important to image understanding. In the next chapter, we will propose a Markov random field based framework for image retrieval which models the feature dependency. Another limitation of relevance models is the relatively high computational expense due to the non-parameter kernel-based estimates, although they show statistically significant improvement in performance. The non-parameter kernel estimate does not assume any underlying probability distribution functions and needs to calculate an expectation over all the training images for each test image. Although accelerating methods (e.g. pruning the training set and selecting a subset of training image more likely to be similar to the test image) can reduce the computation, the problem is only alleviated to some extent.

CHAPTER 4

DIRECT RETRIEVAL USING MARKOV RANDOM FIELDS

Most retrieval approaches for un-annotated images require an explicit annotation or recognition procedure, then retrieve those images based on the annotation or recognition results. Models for automatic annotation based image retrieval are trained based on annotation performance rather than retrieval performance. However, maximizing annotation performance doesn't guarantee the best retrieval performance since annotation is about how well words are predicted for each image while retrieval emphasizes the order of images in a ranked list given a query. More precisely, annotation models are usually evaluated based on prediction accuracy and trained by maximizing the likelihood of generating the training set, while retrieval performance is evaluated using mean average precision. Mean average precision weights each query equally and so it is important to do well on as many of the queries as possible. However, annotation accuracy gives more weight to those words which occur more frequently in the dataset. For example in a database of nature images where sky or grass occurs much more frequently than animals such tigers or lions, finding sky or grass accurately improves the annotation results substantially - in fact one could have very good annotation performance based on just detecting sky and grass. It is almost as if stop words dominated the annotation results. It has also been experimentally shown in text retrieval that the likelihood surface is unlikely to correlate with the retrieval metric surface [106].

In this chapter, we propose a Markov random field (MRF) based framework for direct image retrieval. Our MRF framework is analogous to a framework proposed by Metzler and Croft [103] to capture feature dependence for text retrieval. Although Markov random field

models have been widely used in computer vision for low-level (e.g. edge detection and image segmentation) and high-level tasks (e.g. matching) [148, 20], our Markov random field based model is quite different from them in terms of the problems tackled, topological structure and training strategy. Many computer vision problems solved by MRF based methods are essentially problems involving labeling image sites with word labels, while in this chapter the proposed MRF models solve a problem of ranking images in response to a text query. So the goal of our proposed MRF model is to compute the joint probabilities of images and queries. The graph consists of a set of region nodes representing an image and a set of word nodes representing a query, with edges determining the dependency among these nodes. Unlike labeling problems, our proposed model neither requires that every training image is labeled region by region nor outputs annotations at the region level. Instead, it calculates the joint probability of a query word with the entire image in order to rank images. We also show that the proposed MRF model may be simplified to a linear form for the ranking task and trained through maximizing the retrieval performance - mean average precision. So here the “direct” has two meanings: 1) The model does not involve an explicit annotation step for retrieval. 2) The model is trained through directly maximizing retrieval performance.

Little previous work has been done on direct image retrieval based on text queries. Jeon *et al.* [66] directly ranked images according to the K-L divergence of visterm distributions of the query model and the document model. Their approach assumed that the query and a relevant image have similar visterm distributions. However, in the real world these distributions may be very different. Inspired by the cross-language latent semantic indexing (LSI) in text retrieval [30], Hare *et al.* [55] proposed a singular value decomposition (SVD) based approach to learn the semantic structure of the visterms and annotation words from the training set and retrieve images according to the positions of the text images in the semantic space. Like the LSI techniques in text retrieval, their approach assumed latent concepts linking the visual terms and annotation words.

Our direct retrieval framework doesn't make such assumptions. It directly estimates an underlying joint distribution of queries and images $P_{\Lambda}(Q, I)$ through modeling the dependency of annotation words and each image region among a test image, and formulates this as a Markov random field with a set of parameters Λ . Images are ranked according to the posterior $P_{\Lambda}(I|Q)$. Given a set of query and image pairs, the parameters Λ are estimated by directly maximizing mean average precision rather than the likelihood of the training data.

In this chapter, we discuss a Markov random field framework and a number of models. The specific contributions include:

1. Avoiding the problem of unbalanced/uneven human annotations in the training set. Annotation or recognition models usually do not perform very well when the labels are distributed very unevenly in the training set. Our direct retrieval framework directly ranks images given a query so it doesn't require an annotation step.
2. Overcoming the problem of *metric divergence* caused by applying parameters optimized over annotation performance for retrieval purpose. Parameters trained through maximizing the performance of annotation and recognition are not guaranteed to optimize the performance of retrieval. Without an explicit annotation step, we directly rank images according to the posterior probabilities of image features given a query in the general statistical framework. In our Markov random field based framework, the parameters are trained by maximizing mean average precision directly.
3. Investigating the incorporation of structure information present in images for retrieval. We show that by modeling feature dependencies in the continuous space the retrieval performance may be significantly improved. Although recently some annotation or classification models [68, 82] have modeled feature dependencies based on the statistics of co-occurrences of image parts or interest point descriptors, they have only yielded negative or small improvements on performance. In this chapter, we define and model one kind of region *bigram* to capture the dependence among image

regions. The region bigrams are defined over real-valued region features and significantly improve the retrieval performance after integration into the Markov random field model.

4. Investigation of Markov random fields using continuous features and discrete features respectively. We first formulate MRFs using continuous features then develop a discrete Markov random field which runs much faster than previous models while showing comparable retrieval performance.

4.1 Markov Random Field Framework for Image Retrieval

In this section we describe a Markov random field based model for text query based image and video retrieval. Markov random fields have been widely used in the machine learning domain to model joint distributions of random variables. In computer vision, MRFs have been used for image restoration, edge detection, texture analysis, image segmentation and image matching [84], where all these problems are formulated as a visual labeling problem which assigns a label from a label set to each image site which could be an image region, an edge or a pixel.

In this chapter we use a MRF to model the joint distribution $P_{\Lambda}(Q, I)$ over text queries Q and images I , parameterized by Λ . Based on the joint distribution, images are ranked according to the posterior probability of $P_{\Lambda}(I|Q)$ without an explicit annotation step. The parameter Λ is estimated by directly maximizing the retrieval performance over an annotated training/validation set.

4.1.1 Framework Overview

A Markov random field (MRF) is defined as an undirected graph modeling the joint distribution of a set of random variables. In a MRF, the nodes represent random variables, and the edges define the dependency between these random variables. Every node in a MRF is subject to the Markov property, i.e. the random variable represented by it is conditionally

independent of all other random variables given its neighborhood set. Our Markov random field framework is similar to one for textual retrieval proposed by Metzler and Croft [103]. In our MRF framework G for image retrieval based on text queries, the random variables are the key words $\{q_i\}$ in a query Q and the image I which is represented by a set of image regions $\{r_i\}$. The joint distribution $P(Q, I)$ of a query Q and image I is given by:

$$P(Q, I) = \frac{1}{Z_\Lambda} \prod_{c \in C(G)} \phi(c; \Lambda) \quad (4.1)$$

where $Q = q_1 \dots q_n$, I is represented by a set of regions of $\{r_i\}$, and $\phi(c; \Lambda)$ is a set of non-negative potential functions parameterized by Λ , one for each clique c in graph G . The normalizing constant Z_Λ is:

$$Z_\Lambda = \sum_{Q, I} \prod_{c \in C(G)} \phi(c; \Lambda) \quad (4.2)$$

which is usually expensive to compute because of the exponential number of summations.

The non-negative potential function usually has an exponential form:

$$\phi(c; \Lambda) = \exp[\lambda_c f(c)] \quad (4.3)$$

where $f(c)$ is some feature function over clique c , and λ_c is the weight of this particular feature function.

For textual query based image retrieval, we rank images according to the posteriors $P(I|Q)$ of each image I given the query Q :

$$\begin{aligned} P(I|Q) &= \frac{P_\Lambda(Q, I)}{P_\Lambda(Q)} & (4.4) \\ &\stackrel{rank}{=} \log P_\Lambda(Q, I) - \log P_\Lambda(Q) \\ &\stackrel{rank}{=} \sum_{c \in C(G)} \log \phi(c; \Lambda) \end{aligned}$$

where $\stackrel{rank}{\equiv}$ implies rank equivalence. By the definition of potential functions 4.3, the ranking function may be re-written as:

$$P_{\Lambda}(I|Q) \stackrel{rank}{\equiv} \sum_{c \in \mathcal{C}(G)} \lambda_c f(c) \quad (4.5)$$

4.2 Image Representation and Variants of MRF

The model proposed here has no specific requirements on the image representation. Each image is represented as a set of sites, which may be either local interest descriptors or image regions. In this chapter, an image I is represented as a set of regions $\{r_1, r_2, \dots, r_m\}$, either obtained through superimposing a rectangular grid or by automatic segmentation of the image. Real-valued visual features (color, texture) are first extracted from each region. Depending on whether the model uses continuous features or discrete features, the image representation is different. In the case of the model for continuous features, each region can be directly represented as a real-valued vector and an image is represented as a set of real-valued vectors, each of which is associated with the location information in the original image. On the other hand, if the model uses discrete features, we first quantize all the real-valued features to build a visual vocabulary, and then each region has a corresponding visterm (visual word). Consequently, the image will be represented as a set of visual words with the location information in the original image. In the following sections, continuous Markov random field models and discrete Markov random fields models will be proposed and discussed respectively.

We explore two variants of the MRF model, with different dependence assumptions. The first one is the fully independence variant (MRF-FI) and the other one is the nearest region dependence variant (MRF-NRD). The fully independent variant makes the assumption that all image regions are independent of each other given some query Q , which is made by many annotation or retrieval models for images and videos, e.g. relevance models and machine translation models. Under this assumption, the likelihood of one image re-

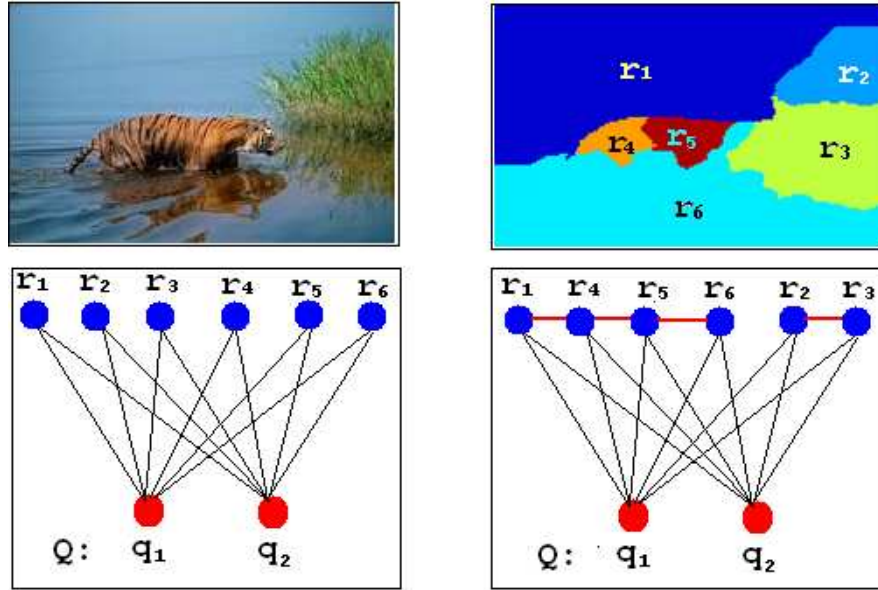


Figure 4.1. The configuration of MRF models for image retrieval. The top line shows the original image and its regional representation. The bottom line shows the full-independence MRF variant(left) and the nearest region dependence MRF variant(right), where edges in red are determined by nearest region pairs. To obtain the nearest region pairs, we look for the nearest neighbor of each region in terms of their mass centers in the image (e.g. the nearest region of r_1 is r_4 , and of r_4 is r_5 in the figure), and then accordingly add an edge for each pair of these two regions in the MRF configuration

gion is independent of others given the query. The nearest dependence variant assumes that only neighboring regions have a dependence on each other (given the query) and that two regions which do not neighbor each other are independent given the query. It is straightforward to generalize the MRF to model higher order dependencies but the computational load becomes higher and so they are not explored in this chapter. One can also easily model other kinds of features such as point features. Figure 4.1 illustrates the configurations of the fully independent MRF and the nearest region dependent MRF.

4.3 Continuous Markov Random Fields for Image Retrieval

This section presents the details of continuous Markov random fields for image retrieval, in which each region r_i is a real-valued feature vector.

4.3.1 Clique Potentials

We do not calculate the exact joint probability $P(Q, I)$. Instead, by choosing proper potential functions we try to approximate the joint distribution in a generalized exponential form. For example, the potential function should give a higher value for a clique including the query word “tiger” and an orange image region with black strips than a clique of the same query word and an image region in blue. The proposed MRF explicitly models the context information since each query word node is connected to all image regions or region pairs. As an example, given an image of a tiger in forest and the query word “tiger”, every region in this image will contribute to the energy function with a non-negative value depending on how compatible that region is with the query word. This is quite different from standard MRF based annotation or recognition methods which make hard decisions for each region by labeling it with one word.

For any clique which doesn’t involve an image region node, the potential is assumed to be one since it does not have an impact on ranking. In the fully-independent graph model the simplest clique is a 2-clique consisting of a query node w and an image region node r , while in the nearest dependent model there are cliques containing two image regions and a query word. Based on the region dependency, we define the potential functions over 2-cliques and 3-cliques in the graph respectively. The potential functions may be easily generalized to cliques of higher order.

4.3.1.1 Full Independence (MRF-FI)

The basic idea of the potential function for a 2-clique consisting of one image region r and one query word w is to formulate the possibility of predicting the query word w given the region r , weighted by the importance of this region r in the image I . Formally, the potential function is defined as (where the left-hand side is equal to $\lambda_c f(c)$ in equation (2)).

$$\varphi_F(w, r) = \lambda_F P(w|r)P(r|I) \quad (4.6)$$

where λ is the weight of this potential function and $P(r|I) = \frac{1}{|I|}$ with $|I|$ as the number of regions in the image I , and $P(w|r)$ is calculated using Bayes' rule as:

$$P(w|r) = \frac{P(w, r)}{P(r)} = \frac{P(w, r)}{\sum_w P(w, r)} \quad (4.7)$$

$\frac{1}{P(r)} = \frac{1}{\sum_w P(w, r)}$ acts like an “*inverse document frequency - idf*” in the continuous case, which measures the capability of a visual feature in distinguishing different words.

To calculate $P(w, r)$, where image region r is represented by a real-valued feature vector, we utilize a kernel-based estimate over all the training images:

$$P(w, r) = \sum_{J \in \tau} P(J)P(w|J)P(r|J) \quad (4.8)$$

where τ is the training set and J is an image in the training set. Note the estimation of $P(w, r)$ has a form very similar to the generative distribution estimation presented for the continuous relevance model [78, 43, 77], except that here it is a joint distribution of a single region with one query word. So we adopt the same estimate for $P(w|J)$ and $P(r|J)$ as in [77]:

$$P(r|J) = \frac{1}{m} \sum_{t=1}^m K\left(\frac{\|r - r_t\|}{\beta}\right) \quad (4.9)$$

where m is the number of regions in image J and t is an index over the set of bigrams in a training image J . This equation arises by placing a Gaussian kernel K over the feature vector r_t of every region of image J . β parameterizes each kernel and plays the role of kernel *bandwidth*. The value of β is selected empirically on a held-out portion of the training set.

The word probability $P(w|J)$ is estimated based on the relative frequency of the word w in the annotation of image J which has been padded to a fixed length with a special “null” word:

$$P(w|J) = \lambda \frac{N_{w,J}}{N_J} + (1 - \lambda) \frac{N_w}{N} \quad (4.10)$$

where N_J is the fixed length of annotations of training images, $N_{w,J}$ the number of occurrence of word w in image J , N_w the number of w in the whole training set and N the total number of annotation words in the training set.

4.3.1.2 Nearest Region Dependence (MRF-NRD)

In the nearest regional dependence variant, the 3-cliques consisting of two nearest regions and one query word basically capture the dependency between the query word and “region bigrams”. The potential function over the 3-cliques measures the compatibilities of the region bigrams and the query word. In this chapter, the region bigrams of an image I are determined in the vertical and horizontal directions respectively and form two different bigram sets. We obtain the vertical region bigrams from an image through the following steps:

1. For each image region r_i , first look for its nearest region r_j in the image in the vertical direction. The distance of two image regions is measured based on their mass centers.
2. Then rearrange these two regions according to their relative position always keeping them in the top-down order.
3. Finally remove repeated pairs from the image bigram set.

The horizontal bigrams are obtained similarly except that the region pairs are selected in the horizontal direction and rearranged in the left-right order. The bigrams selected in this way keep the relative position information.

The potentials of vertical bigrams and horizontal bigrams have the same functional form, but are calculated separately. In experiments we use one weight for the potentials of all bigrams. For the sake of the simplicity of notation, all bigrams in this chapter are from the same kind of bigrams hereafter, either vertical bigrams or horizontal bigrams. Without the loss of generalization, let (r_i, r_j) be one region bigram from image I . The potential function for a 3-clique of a region bigram and query word w is defined as:

$$\begin{aligned}
\varphi_N(w, (r_i, r_j)) &= \lambda_N P(w|(r_i, r_j)) P((r_i, r_j)|I) \\
&= \lambda_N \frac{P(w, (r_i, r_j))}{P((r_i, r_j))} \frac{1}{\sharp((r_g, r_h), I)}
\end{aligned} \tag{4.11}$$

where λ_N is the weight of the bigram potential functions, $\sharp((r_g, r_h), I)$ here is the number of region bigrams of image I , and $P((r_i, r_j)) = \sum_w P(w, (r_i, r_j))$.

The joint probability $P(w, (r_i, r_j))$ of a region bigram (r_i, r_j) with a query word w is calculated through a kernel estimate over all training images:

$$P(w, (r_i, r_j)) = \sum_{J \in \tau} P(J) P(w|J) P((r_i, r_j)|J) \tag{4.12}$$

where $P(w|J)$ is estimated using equation 4.10 and $P((r_i, r_j)|J)$ is estimated through a Gaussian kernel density estimate over each bigram of image J :

$$\begin{aligned}
P((r_i, r_j)|J) &= \frac{1}{m} \sum_{t=1}^m K\left(\frac{\|(r_i, r_j) - (r_i, r_j)_{Jt}\|}{\beta}\right) \\
&= \frac{1}{m} \sum_{t=1}^m K\left(\frac{\max(\|r_i - r_{i-Jt}\|, \|r_j - r_{j-Jt}\|)}{\beta}\right)
\end{aligned}$$

where m is the number of bigrams in image J , K a Gaussian kernel and β is the bandwidth of the density estimate and has the same value as in equation 4.9. In this equation the distance between two pairs of region bigrams in the feature space is the maximum distance of the two pairs of corresponding regions. Note that since the two regions of each bigram have been rearranged w.r.t their spatial location in the image, the feature distance is always calculated between regions with the same relative position. For example, feature distances are calculated between the two top regions and the two bottom regions respectively for vertical bigrams. The intuition behind using a “maximum” rather than an “average” is to impose a strict compatibility requirement for a bigram and a query word, that is, only when

both regions of a bigram have high correlations with the query word, the potential of the corresponding clique has a high value.

4.4 Discrete Markov Random Fields for Image Retrieval

This section proposes Markov random fields built on discrete image features, which are called discrete Markov random fields distinguished from the continuous models in the previous section. Using the continuous feature space is usually much more compute-intensive than using the discrete feature space. So one important purpose of developing discrete Markov random field models is to speed up the retrieval procedure. Experiments shows that our discrete Markov random fields run extremely fast on even very large datasets while having comparable retrieval performance to continuous models.

4.4.1 Feature Quantization and Building a Large Visual Vocabulary

Since a discrete Markov random field is build on discrete image features, the first step is to quantize image features into discrete visual words (visterms). Unsupervised clustering methods are usually employed for this purpose, e.g. K-means clustering or hierarchical clustering. Most clustering methods require that one pre-defines either the number of categories or some threshold controlling the number of categories. Each category corresponds to one visual word, so the number of categories is actually the size of the visual vocabulary. Recent literature on object or image matching [112, 118] using discrete features has shown that ¹ the size of visual vocabulary can substantially affect the matching performance and good performance requires large visual vocabularies. This is reasonable since large visual vocabularies are better at distinguishing different visual features. However, too large a visual vocabulary can also segregate features originating from the same objects. So selecting the appropriate size of visual vocabulary is very important, but usually very difficult without any domain knowledge. To test on different size of visual vocabularies, one requires a

¹Jeon and Manmatha also noticed this in unpublished work.

fast clustering approach which can deal with large-scale features in high dimension space. A flat K-means clustering or a hierarchical agglomerative clustering (e.g. single linkage clustering) doesn't meet this requirement. The time complexity of K-means is linear to the number of training features and the number of clusters, while the complexity of the hierarchical agglomerative clustering is at least square of the number of training features. It is impractical to use these two clustering methods to build large visual vocabularies (e.g. more than 1 million categories for more than 10 million features). For this reason, we adopt the hierarchical k -means for clustering in our work [112, 118].

Hierarchical k -means applies a tree structure for representation of the clustering results over a set of training features, where k defines the branch factor of the tree rather than the total number of the categories. Initially, the k -means algorithm partitions the training features into k categories/clusters, each of which forms a node in the tree consisting of feature vectors closest to a particular cluster center. The same k -means algorithm is then recursively applied to each node and splits each of them into k finer clusters. This process is recursively performed until the depth of the tree reaches a pre-defined level. So if the depth of the tree is d , the number of categories at the leaf level will be k^d . The computational cost of the hierarchical k -means is logarithmic in the number of leaf nodes, which is much smaller than that of non-hierarchical clustering methods.

The visual vocabulary tree is constructed by clustering all the feature vectors in the training set using the hierarchical k -means. Then the feature vectors of the test set will be clustered through an efficient search procedure, which propagates the vector down the tree till the leaf level by comparing the vector with the k candidates cluster centers at each level and selecting the closest one. This lookup only takes $\mathbf{O}(\log(n))$ compared with the complexity $\mathbf{O}(n)$ of a flat K-means for the same task, where n is the size of the visual vocabulary. In the case of an extremely large training set, the visual vocabulary tree may be constructed using a portion of the training feature vectors sampled from the whole training set. Then the corresponding visual words of the rest of training vectors are obtained through

searching over the tree as for the test vectors. So finally, an image is represented as a set of visual words (visterms) each of which corresponds to one image region, noted as $\{v_1, \dots, v_m\}$, where m is the number of the regions.

Our discrete MRFs have the same configurations as shown in figure 4.1 but feature functions for each clique are calculated in different ways.

4.4.2 Full Independent Discrete MRF

As in the continuous case, the potential function of the fully independent MRF is defined over an image region represented as a discrete visterm v and one query word w . Formally, it is formulated as:

$$\varphi_F(w, v) = \lambda_F \lambda_v P(w|v)P(v|I) \quad (4.13)$$

where λ_F is the weight of the full independent potential function and λ_v the weight of the visterm v . λ_F is identical to all the full independent potential functions and tuned over a validation set. λ_v is directly calculated as the *idf* (*inverse document frequency*) $\lambda_v = \log \frac{|D|}{\#(d_j \ v \in d_j)}$ which measures the general importance of the discrete visterm v . $P(v|I)$ is the probability of a visual word v observed in the test image I and $P(w|v)$ is the posterior probability of a query word w given a visual word v . So here the potential function basically represents the possibility of predicting query word w from the occurrences of visual word v in the test image I .

Estimating the probabilities $P(w|v)$ and $P(v|I)$ depends on the distributions of the words and the visterms. In our previous chapter on relevance models, we have shown that the normalized multinomial or multiple Bernoulli model is more suitable for annotation word distribution. So here, we utilize normalized multinomial distribution for word probability estimation. Without any prior knowledge about the discrete visterm distribution, we investigate both multinomial and multiple Bernoulli models for $P(v|I)$.

4.4.2.1 Multinomial Vistern Model

Based on a multinomial distribution assumption of the visterms from images, the probability of $P(v|I)$ is calculated as the frequency of the vistern v in image I :

$$P(v|I) = \frac{\#(v, I)}{\sum_v \#(v, I)} \quad (4.14)$$

where $\#(v, I)$ is the number of occurrences of the vistern v in image I .

The posterior probability $P(w|v)$ is calculated under the Bayesian framework:

$$P(w|v) = \frac{P(w, v)}{P(v)} = \frac{P(w, v)}{\sum_w P(w, v)} \quad (4.15)$$

As in the continuous case, the joint probability of $P(w, v)$ is calculated through an expectation over all training images:

$$P(w, v) = \sum_{J \in \tau} P(J)P(w|J)P(v|J) \quad (4.16)$$

where τ is the training set and J is an image in the training set. The word probability $P(w|J)$ is estimated under the normalized multinomial word distribution framework as in Equation 4.10. With a multinomial distribution, the vistern probability $P(v|J)$ of a vistern v generated by a training image J is estimated as:

$$P(v|J) = \frac{\#(v, J)}{\sum_v \#(v, J)} \quad (4.17)$$

4.4.2.2 Multiple Bernoulli Vistern Model

A multiple Bernoulli vistern model only considers if a particular discrete vistern occurs in the image or not and ignores the number of occurrences of that vistern if it does exist in the image. Correspondingly, the probability of $P(v|I)$ is estimated as:

$$P(v|I) = \delta_{v,I} \quad (4.18)$$

where $\delta_{v,I} = 1$ if the word v occurs in the annotation of image I and zero otherwise.

Similarly, the probability $P(w, v)$ is calculated using Equation 4.16 but $P(v|J)$ is computed based on the multiple Bernoulli distribution: $P(v|J) = \delta_{v,J}$.

So far we have talked about the fully-independent model which treats visterms independently and ignores the latent relationships among the image features. It is desirable that the model also discover the visterm dependency for retrieval. We have investigated multiple ways to incorporate visterm dependency in the discrete case including the nearest neighboring region dependency, local constrained region dependency and full dependency among regions. Although our MRFs in the continuous cases have shown that modeling feature dependency can significantly improve the performance, our MRFs in the discrete cases haven't achieved that through modeling discrete visterm dependency. The main reason we find is that in the discrete case, a very large visual vocabulary generates very sparse distributions of visterm bigram, i.e. most visterm bigrams observed in the test set have no occurrence in the training set at all. Although this problem can be alleviated through reducing the size of visual vocabulary dramatically, a smaller visual vocabulary loses the expressivity of representation of the images and our experiments show that it leads to low performance.

4.5 MRF Training for Image Retrieval

Discrete MRF models in the previous section are parameterized by the weights λ_R , λ_L and λ_N for particular feature functions. In standard training approaches for MRF, these parameters are set through maximizing the log likelihood or the posterior of observing a given annotated or labeled image.

However, as claimed in [103], there are several disadvantages using these standard approaches for training MRF models for retrieval purpose: First, the event space of $Q \times I$ is extremely large so that maximizing likelihood tends to generate a biased estimation of the distribution. Second, it is difficult to compute the normalization factor Z_λ given a large

training set. Third, it has been shown [103] that maximizing the likelihood of generating the training set doesn't guarantee maximizing the retrieval performance, say mean average precision. So based on the observation in [103] that the surface of the mean average precision is concave or nearly concave over an interesting range of parameter values, we directly maximize the mean average precision through coordinate-level hill climbing search.

4.6 Experimental Results

We used three different datasets in our experiments for comparison between our model and other models. These three datasets have been described in the experimental section [3.2] of Chapter 3. The first one is the small Corel image set which contains 5000 images and the second one is the 160-CD Corel image data set (including features) which is exactly the same as that used by Barnard *et al.* [9]. Models are also tested on the large scale data set consisting of the entire TRECVID 2003 development dataset and feature set used by [63]. The same features as described in Section 3.2 were used for the purpose of fairly comparing models.

4.6.1 Retrieval Results of Continuous MRFs

Given a test set of images or key frames, our goal is to rank them according to the posterior $P(I|Q)$ calculated in equation (4.5). An image or key frame is considered relevant to a given query if its true annotation contains all query words. For simplicity, in our experiments we only considered one word queries. One can easily generalize it to multiple-word query cases since given the linear form of the ranking equation given in 4.5 and the independence assumption among query words implies that the ranking score for a multiple-word query is equal to the sum over the scores of individual query words. So the retrieval system can calculate the ranking score for each key word through an offline procedure and store them for further use when retrieving arbitrary queries. To accelerate the score calculation, we adopted a voting scheme to first quickly determine a small subset of images

	Corel Standard		Corel Novel		TRECVID03	
	mAP	P@10	mAP	P@10	mAP	P@10
N-CRM	0.258	0.377	0.054	0.102	0.158	0.319
MRF-FI	0.269 (+4%)	0.389 (+3%)	0.055 (+2%)	0.107* (+5%)	0.175* (+11%)	0.397* (+25%)
MRF-NRD -Exp1	0.272* (+5%)	0.392 (+4%)	0.056* (+4%)	0.113* (+11%)	0.200† (+27%)	0.431† (+35%)
MRF-NRD -Exp2	0.285† (+11%)	0.420† (+11%)	0.059† (+9%)	0.137† (+34%)	0.216† (+37%)	0.449† (+41%)

Table 4.1. Comparisons of retrieval performance of various models, which are normalized-continuous relevance model (N-CRM), full-independent MRF model(MRF-FI) and nearest region dependent MRF model(MRF-NRD). MRF-NRD-Exp1 and MRF-NRD-Exp2 are both nearest region dependent MRF models, but their parameters (λ_F, λ_N) are trained differently. For MRF-NRD-Exp1, we trained one identical set of parameters for all query words while for MRF-NRD-Exp2 we tuned parameters separately for each individual words. Symbol * indicates a statistically significant improvement over the N-CRM, and † indicates a statistically significant improvement over both N-CRM and MRF-FI.

in the training set most similar to a test image, then calculate ranking scores based on the selected subset of images.

To evaluate the performance of a retrieval system, we use the standard metric - mean average precision (mAP). Average precision is calculated for each query as the average of precision values at correctly returned points, which is further averaged over all queries to get a reliable measurement. Since for large datasets, it is more likely that users are only interested in the top 10 or 20 ranked images, we also report the average precision at the 10th ranked images (P@10). In our experiments, we take each word present in the test set as a single query and report average precision over all the queries. For the 5k Corel image set, the queries are 260 annotation words which occur in the test set. The average numbers of queries are 161 and 134 for the 10 samples of Corel standard held-out set and the 10 samples of the Corel novel held-out set respectively. For the TRECVID, each of the 75 selected concepts is a single query.

Table 4.1 shows the retrieval performance of various models over the Corel standard held-out set, Corel novel held-out set and the TRECVID set. For the nearest region de-

pendency MRF, we performed two different experiments. The first one (MRF-NRD-Exp1 in the table) tuned one identical set of parameters (λ_F, λ_N) for all query words, while the second one (MRF-NRD-Exp2) tuned separate sets of (λ_F, λ_N) for each individual query words. A statistical significance test was implemented using a t-test with $p \leq 0.05$. The table shows that MRF-NRD-Exp2 consistently performs better than other models over all these datasets. From the Table 4.1, we can see that:

query	AP of MRF-FI	AP of MRF-NRD	(λ_F, λ_N)
clouds	0.407	0.407	(1.0,0.0)
tree	0.359	0.361	(0.9, 0.1)
woman	0.546	0.55	(0.9, 0.1)
sunset	0.465	0.483	(0.8, 0.2)
dunes	0.527	0.578	(0.7, 0.3)
boats	0.275	0.289	(0.6, 0.4)
tiger	0.550	0.573	(0.5, 0.5)
bear	0.427	0.474	(0.4, 0.6)
tulip	0.091	0.173	(0.3, 0.7)
sailboats	0.312	0.329	(0.2, 0.8)

Table 4.2. Query examples on Corel Standard test set. MRF-NRD parameters (λ_F, λ_N) were separately tuned for each word.

1. The fully independent MRF model is comparable in performance with the normalized continuous relevance model on the Corel datasets and significantly better than it on the TRECVID dataset. We believe one reason is that the TRECVID has a larger training set.
2. Nearest region dependent MRF models outperform the fully independent MRF and the normalized-continuous relevance models, which shows that appropriately incorporating structure information could significantly improve retrieval performance.
3. Separately tuning parameters (λ_F, λ_N) for individual query words outperforms using one identical parameter for all query words. Table 4.2 shows some query examples from the Corel standard dataset with the tuned parameters for the MRF-NRD model,

	MBRM	Carneiro <i>et al.</i> [22]	MRF-FI	MRF-NRD-Exp1	MRF-NRD-Exp2
mAP	0.30	0.31	0.30	0.31	0.34

Table 4.3. Comparison of retrieval performance of different models on the small corel set of 5k images.

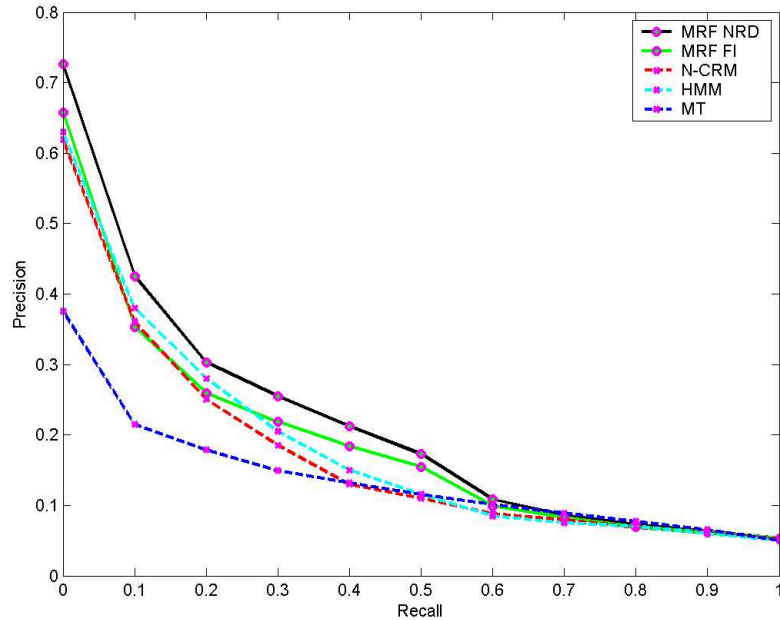


Figure 4.2. Recall-precision graphs of various models over the TRECVID set, which are the full-independent MRF model(MRF-FI), nearest region dependent MRF model(MRF-NRD), the NCRM, the machine translation model(MT) and the HMM model.

which shows that different words require different weights for independent regions and region bigrams.

Table 4.3 shows retrieval performance comparisons over the 5k Corel dataset, our best results are obtained using MRF-NRD with separately tuned weights for each word, with a mean average precision (mAP) of 0.34 which outperforms the best result (0.31) published before (achieved by Carneiro *et al.* [22]).

Figure 4.2 shows the recall-precision curves of various models over the TRECVID3 set, from which we can see that MRF-NRD outperforms other models consistently over the whole curve (the NCRM and translation model curves are from [63] while the HMM

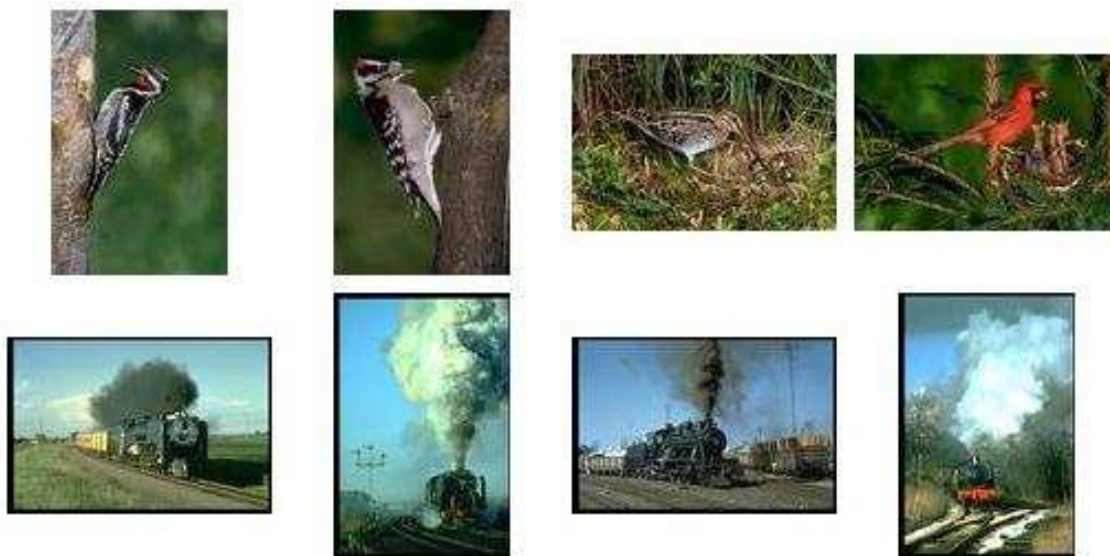


Figure 4.3. Examples of top 4 ranked images in response to query “birds” and query “train” using MRF-NRD over the small Corel set. Each image in this set is 192x128 and partitioned into 24 rectangular regions



Figure 4.4. Examples of top 4 ranked key frames in response to query “waterfall” and query “clock” using MRF_NRD over the TRECVID2003 set. Note that the 4th image bottom contains a clock on the wall at the top right corner. It is a relevant image to clock in the ground truth. Each frame (352x264) is partitioned into 35 rectangular regions.



Figure 4.5. Top ranked images in response to the query “boats” using MRF-NRD over the small Corel data set. The second and the third images are irrelevant.



Figure 4.6. Some supporting images for the word “boats” in the training set.

curve is from [52]). Figure 4.4 and Figure 4.3 show some examples of top ranked images in response to queries using the MRF-NRD model. All the images shown in these two figures are relevant to their corresponding queries.

Figure 4.5 shows the top ranked images in response to query “boats” using MRF-NRD over the small Corel data set. Both the second and the third are irrelevant to the query according to the ground truth, although they are highly ranked. Again these false positives are caused by the significant contributions from the background contexts of the “boats” in the training images. Although the second image doesn’t contain a boat, its background is very close to the backgrounds of some supporting images to “boats” in the training set. The first two images in Figure 4.6 show two such examples. The boat in the second supporting image (under the sun) is very tiny and hard to identify, but the human annotator did annotate it with “boats”. It is the same reason for the third image in 4.5 being highly ranked – note that the last two supporting examples in Figure 4.6 have very similar backgrounds with it.

Remember that this problem also happens when using relevance modeling approaches for annotation and retrieval (see Section 3.2.3). This is not surprising because in both MRF and relevance models the context information plays an important role. The context information is obtained through allowing every image region to contribute to the probability of a query word. As we suggested before, better image features and larger training sets will help solve this problem. Better image features may distinguish the target object from other objects easily, and larger training sets may train the model better to identify a common background and weight it less for probability contributions.

Although our MRF models are proposed for image retrieval, we can also apply them for image annotation. We annotate each image in this way: first calculate the ranking scores for each word based on equation 4.1, then annotate each image with those words in descending order of the ranking scores. We compared the annotation performance of our full-independent MRF model with MBRM and all the models proposed in [9] and Table 4.4 presents the results. We used the same measurements with those used by [9], which have been briefly described in the experimental section (Section 3.2.3) of the previous chapter. Table 4.4 shows both MRF models and MBRM consistently outperform the models proposed in [9] on image annotation. The difference between MRF-FI and MBRM shown in this table is not statistically significant.

Measurements	Corel standard set			Corel Novel set		
	PR	NS	KL	PR	NS	KL
Best Results in [9]	0.298	0.604	0.747	0.249	0.506	0.268
MBRM	0.371	0.647	1.129	0.255	0.514	0.274
MRF-FI	0.372	0.650	1.126	0.264	0.518	0.270

Table 4.4. Performance comparison on automatic annotation between MRF-FI, MBRM and the models reported in [barnard:matching]. Results show that MRF-FI consistently outperforms the models in [barnard:matching].

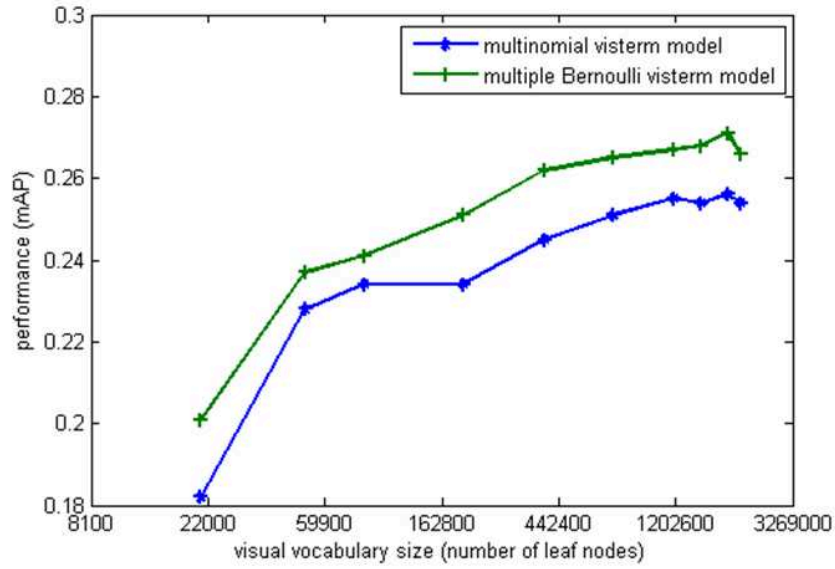


Figure 4.7. Curves of performance vs visual vocabulary size for multinomial visterm model and multiple Bernoulli visterm model.

4.6.2 Retrieval Results for Discrete MRFs

In our experiments on discrete MRFs, images are first partitioned into rectangular regions and the discrete visterms are build from those regions. Since we do not have the original images of the 100-CD Corel set, our experiments on discrete MRFs are conducted over the 5k Corel image set and the TRECVID03 set.

On the 5k Corel image set, we tested the effects of the size of the visual vocabulary on the retrieval performance and compared the multinomial visterm model and the multiple Bernoulli visterm model (see Figure 4.7). From which we can see that the mean average precision dramatically increases with vocabulary size and then flattens out. We can also see that the multiple Bernoulli visterm model works better than the multinomial visterm model by a small margin.

We also observed that with the same visual vocabulary size, the performance increases with the branching factor (see Figure 4.8). We believe this is because of the property of the hierarchical k-means clustering. Since the visual vocabulary is constructed from a

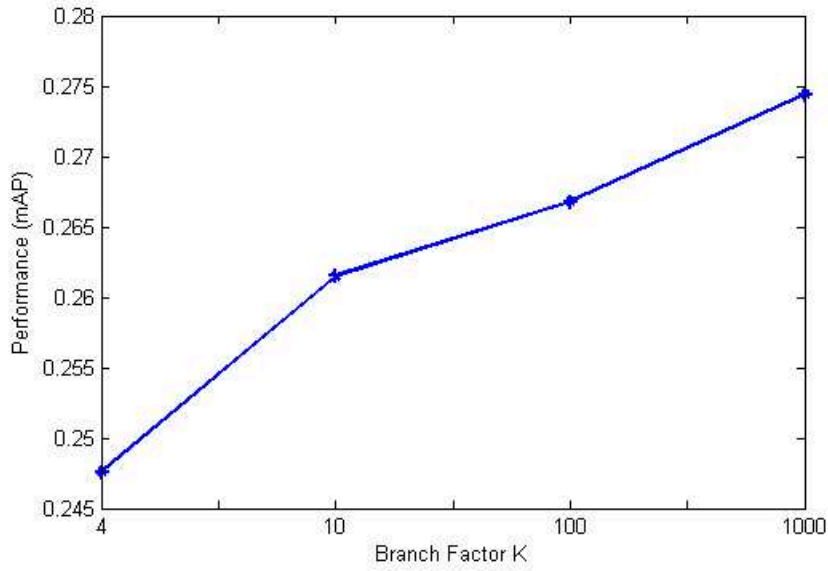


Figure 4.8. Performance vs branch factors with 1M leaf nodes

	CMRM	CRM	N-CRM	SML [22]	Discrete MRF
mAP	0.14	0.26	0.30	0.31	0.28
RunningTime(secs)	10	660	660	-	16

Table 4.5. Retrieval performance comparison between discrete MRF and other models on the 5k Corel set. The running time is measured for all the 371 words in the vocabulary.

hierarchical k-means tree, then any errors in clustering made at the higher level will be propagated to the lower level and cannot be corrected. A tree with a larger branch factor reduces the chances of propagation of the clustering error. In our experiments we also tested the discrete visterm sets constructed from the multiple-scale space of the original image via the image Gaussian pyramid. However, the results didn't show any apparent improvement over the single scale setting.

The best results of our discrete MRF is from the multiple Bernoulli visterm model with a 2085136 (38^4) vocabulary size over 16x16 rectangular partitions, which achieves mAP of 0.28 and precision@10 of 0.198. Table 4.5 shows the comparison with other models, from which we can see that our discrete MRF achieves better results than the cross-media relevance models (CMRM), the continuous relevance model (CRM) and is comparable

	N-CRM	Discrete MRF
mAP	0.158	0.152
P@10	0.319	0.335
RunningTime	6.8(hrs)	90(secs)

Table 4.6. Retrieval performance comparison between discrete MRF and N-CRM on the TRECVID03 set. The running time is measured for all the 75 query words.



Figure 4.9. 5 top ranked images of the discrete MRF in the test set of the 5k Corel set in response to the query word "birds"

to normalized-continuous relevance models and Carneiro’s hierarchical Gaussian mixture model [22]. However our discrete MRF is much more efficient than continuous models in terms of running time. The implementation used sparse matrix techniques to accelerate the probability calculation.

Finally, our discrete MRF model was tested on the TRECVID03 dataset. Each keyframe in this dataset is partitioned into 32x32 rectangular regions, and then color and texture features are extracted from each region. We randomly sampled 1/10-th of the training features to construct a visual vocabulary of size 2085136 (38^4). Then the remaining training features and all test features are determined by looking up using the tree to obtain their corresponding discrete visterms. Our results are shown in table 4.6, from which we can see that the mean average precision of our discrete models on this dataset is very close to the NCRM and the precision@10 is slightly better. Compared to the 6.8 hours of running time measured for the NCRM, our discrete model only takes 1.5 minutes to complete the whole procedure after the discrete visterm set is obtained.

Figure 4.9 shows the 5 top ranked images in the returned rank list of the discrete MRF in response to the query word "birds" over the 5k Corel set. Note the third image does not contain



Figure 4.10. 5 top ranked images of the discrete MRF in the test set of the TRECVID03 set in response to the query word "sport_event"

a bird (seagull) although it is quite small and the ground-truth annotation of that image does have the word "birds". Although the ground-truth annotation of the first image doesn't contain the word "birds" (instead it has "albatross"), our model correctly associates word "birds" with it. Figure 4.10 shows a retrieval example for the discrete MRF in response to the query word "sport_event" over the TRECVID03 dataset.

CHAPTER 5

HISTORICAL HANDWRITTEN DOCUMENT RECOGNITION

In this chapter, we discuss historical handwritten document recognition. We describe and compare the application of various classification and sequence models to the recognition task.

In particular, our historical handwritten document recognition is performed at the word level, i.e. the recognition units are word images. Character segmentation on degraded historical handwritten document is still a difficult problem. To avoid the errors introduced by character segmentation, recognition in this work is directly done on the word level where historical handwritten documents are first segmented into word images. Word segmentation is much easier to do than character segmentation. Once features are extracted from each word image, machine learning techniques may be employed to label each feature by learning over transcribed documents. For the purpose of fair comparisons between different models, this chapter employs the same dataset and the same features for experiments with those used by [79]. The dataset consists of 20 pages from a collection of letters by George Washington. Each page is segmented into individual word images, from which scalar image features and profile-based image features are extracted. One can refer to the experimental section 5.3 for more details about the dataset and features.

This chapter is devoted to the investigation of various statistical models on word image recognition for historical handwritten documents. Most of the models discussed in this chapter, including support vector machines, maximum entropy, naive Bayes with kernel density estimate, and conditional random field, have been employed for this task for the first time. Although HMMs have been widely used for this problem, we proposed two

different ways to improve the recognition performance with HMM's. One is the smoothing of the probabilities of feature generation and the other is the combination of HMM and kernel density estimates. The recognition results may be used for retrieval. To do this, we can directly use a text retrieval model over the automatically generated transcripts of the handwritten images.

In this chapter, classification models are first discussed followed by sequence models. Finally, experiments over these models are reported and compared.

5.1 Classification Models for Handwritten Word Recognition

We investigate a number of different classification models for historical handwritten word recognition - both discriminative and generative. The model selection is based on their general classification performance and their usefulness to our recognition problem.

5.1.1 Support Vector Machines

Originally introduced as a binary linear classifier, support vector machines (SVMs) attempt to find an oriented hyper-plane which separates the linear separable space defined by the training data while maximizing the margin. The margin is the distance of each training instance to the hyperplane.

To extend this to classifying nonlinear separable data, SVM uses a *kernel* function K to map the training data to a higher Euclidean space, in which the data may be linearly separable. The kernel function is defined as : $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where $\phi(x)$ is some mapping. For dealing with nonseparable data and avoiding overfitting, SVM's usually use a soft margin which allows some instances to be misclassified. A SVM classifier solves the optimization problem:

$$\min_{\xi, \mathbf{w}, b} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^N \xi_i \quad (5.1)$$

such that $y_i(\langle \mathbf{w}, \phi(x_i) \rangle + b) \geq 1 - \xi_i$, and $\xi_i \geq 0$. Here \mathbf{w} is a vector pointing perpendicularly to the separating hyper-plane and b is the offset parameter of this hyper-

plane. $y_i \in \{-1, 1\}$ is the label of instance x_i . The slack variable ξ_i measures the degree of misclassification of the datum x_i and the capability C determines the cost of margin constraint violations.

SVM has shown a powerful classification capability in many applications [60]. In the handwritten word recognition scenario, x_i is a feature vector representing a word image and $y_i \in \{-1, 1\}$ is an indicator of whether that word image is labeled as a particular word. To adapt the original SVM to the multiple-class case, we adopt the max-win strategy, which uses k binary SVMs for the k -class problem. Each of these binary SVMs separates the word images of a particular class/label w and the non- w word images. For a test word image, the class/label with the highest value wins. Since the max-win has to train k binary SVMs for k classes (labels), the complexity of SVM training is high for recognition at the word level, since the vocabulary may be very large.

5.1.2 Conditional Maximum Entropy Models

Maximum entropy models have been recently widely applied in domains involving sequential data learning, e.g. natural languages [12, 132], biological sequence analysis [18], and very promising results have been achieved. Since maximum entropy models utilize information based on the entire history of a sequence, unlike HMMs whose predictions are usually based only on a short fixed length of prior emissions, we expect maximum entropy models to work well for handwritten document recognition problems since in our case each page may be taken as a long sequence of words, each of which emits a set of observations represented as word image features.

The goal of conditional maximum entropy models is to estimate the conditional distribution of label y given data x , say $P(y|x)$. The framework is fairly straightforward. It basically specifies that the modeled distribution should be as uniform as possible, while being consistent with the constraints that are given by the features of the training data.

Given a set of predicates¹ $f_i(x, y)$, which may be real or binary values and represent some observation properties (e.g. co-occurrence) of the input x and output y , the constraints are that for each predicate its expectation value under the model $P(y|x)$ should be the same as its expectation under the empirical joint distribution $\tilde{P}(x, y)$, i.e.

$$\sum_{x,y} \tilde{P}(x)P(y|x, \lambda)f(x, y) = \sum_{x,y} \tilde{P}(x, y)f(x, y) \quad (5.2)$$

With these constraints, the maximum conditional entropy principle picks the model maximizing the conditional entropy:

$$H(P) = - \sum_{x \in X, y \in Y} \tilde{P}(x)P(y|x, \lambda) \log P(y|x, \lambda) \quad (5.3)$$

It has been shown [131] that there is always a unique distribution that satisfies the constraints and maximizes the conditional entropy. This distribution has the exponential form:

$$P(y|x, \lambda) = \frac{1}{Z} e^{\sum_i \lambda_i f_i(x,y)} \quad (5.4)$$

where Z is a normalization constant such that $\sum_y P(y|x, \lambda) = 1$ and λ_i is the weight of predicate f_i in the model.

The maximum entropy model's flexibility comes from the ability to use arbitrary predicate definitions as constraints. These feature definitions represent knowledge learned from the training set. So our test of conditional maximum entropy modeling on our task focuses on the aspect of predicate definitions and their effects on performance. Both discrete predicates and continuous predicates are investigated in our work.

¹We use the term predicates rather than features to differentiate these from image features.

5.1.2.1 Discrete Predicates

We do a linear vector quantization (VQ) on the original continuous features measured from the images and discretize each of them into a fixed number of bins. We define two types of binary predicates for the maximum entropy model based on the discrete features extracted from word images and the corresponding label sequence:

1. **Unigram Predicates** The frequency of a discrete feature x and the current word w :

$$f_i^u(x, w) = \begin{cases} 1 & \text{if the feature set of } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

2. **Bigram Predicates** We define two sets of bigram predicates, which intuitively represent the statistical properties of a word and the features of this word's neighboring word images. For example, if in the training set the word "*force*" always follows the word "*Fredericksburgh's*", then in the test set it will increase the probability of the current word being recognized as "*force*" given that its previous word image is very long. One set of bigram predicates we define is the joint frequency of the word w and a discrete feature x which appears in the feature set of the word image preceding word w :

$$f_i^{bf}(x, w) = \begin{cases} 1 & \text{if the feature set of the word image preceding } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

and the other set is the frequency of word w and a discrete feature x which appears in the feature set of the word image following word w :

$$f_i^{bb}(x, w) = \begin{cases} 1 & \text{if the feature set of the word image following } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

5.1.2.2 Continuous Predicates

Since the VQ process causes loss of information from the raw continuous features and there is little literature on maximum entropy models using continuous predicates, we are interested in investigating continuous features for maximum entropy models.

1. **Raw Predicates** In theory, the raw continuous features can be directly fed into the maximum entropy models, defining the predicate as the values of feature φ (feature name, e.g. width, height...) of word w :

$$f_i^{cr}(\varphi, w) = x \tag{5.8}$$

i.e. the feature φ of w has value x

2. **Distance Predicates** Maximum entropy models have a problem with the raw predicates. Note that in Equation 5.4, the conditional probability of a label given the observed features is formulated as an exponential function of the predicates. This exponential form implies that the conditional probability is basically monotonically non-decreasing over each predicate. In the case of raw predicates, since each predicate is an observed feature, the label probability is monotonically non-decreasing with each feature. However we know that the distribution of a visual feature over a specific class/label is usually not a single monotonic function but more complicated (e.g. it may be modeled as a Gaussian mixture which is not monotonically non-decreasing). For example, word image length is a real-valued visual feature widely used for word recognition. In the case of raw predicates, the probability of a word is non-decreasing with the increasing of the word image length, which is apparently not in accordance with reality. To solve this problem, we use distance predicates.

The intuition of using distance predicates is to convert the raw features to some predicates with whose increasing the label probability is truly non-decreasing. Assuming

that each kind of visual features of a label is subject to a certain Gaussian distribution whose mean is the average value of the features, then the larger the distance of an observed feature distance to this mean, the smaller the probability of that label given that observed feature. So one approach to computing distance predicates is to first calculate the centers of each set of features which is labeled as the same word, then use the distances of each raw feature to all these centers as the new feature set substituting for the original raw feature. However if the number of class labels (the size of the vocabulary in our case) is too large, this method will generate too many features to let the maximum entropy models finish all the runs in reasonable time.

An alternative idea is to use k-means to cluster each feature into a fixed number of categories (for simplicity, 5 in our experiments) and calculate the distances of each feature to the centers of every category instead of to the centers of each word class. This distance predicate for maximum entropy is defined as the negative of the distance of each feature of each word to every center:

$$f_i^{cd}(\varphi_c, w) = -d \tag{5.9}$$

i.e. the negative of the distance of feature φ of w to the c -th center is d .

5.1.3 Naive Bayes with Gaussian Kernel Density Estimate

Since our dataset is from letters which use natural language it is unbalanced. That is, since word frequencies follow a Zipfian-like distribution [58] their frequencies vary widely. On the other hand the dataset also provides us with reasonable prior probabilities of words in the document corpus. So instead of discriminative models like SVMs and maximum entropy, we want to use some kind of generative probability density model like Naive Bayes.

The Naive Bayes framework is pretty simple:

$$P(w|\mathbf{x}) = \frac{P(\mathbf{x}|w)P(w)}{\sum_w P(\mathbf{x}|w)P(w)} \quad (5.10)$$

where \mathbf{x} is the feature vector of a word image and w a word. We estimate the prior probability of word w directly as its relative frequency in the training set. We calculate the probability of the visual features of a word image given a word w using a non-parametric Gaussian kernel density estimate:

$$P(\mathbf{x}|w) = \frac{1}{\|w\|} \sum_{i=1}^{\|w\|} \frac{\exp\{-(\mathbf{x} - \mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}_i)\}}{\sqrt{2^k \pi^k |\Sigma|}} \quad (5.11)$$

This equation arises out of placing a Gaussian kernel over the feature vector \mathbf{x}_i of every word image labeled as word w . $\|w\|$ denotes the number of word images labeled as "w". \mathbf{x}_i is a feature of the i -th word image labeled as "w". Each kernel is parameterized by the feature covariance matrix Σ . We assumed $\Sigma = \beta \cdot I$, where I is the identity matrix and β plays the role of kernel bandwidth, which determines the smoothness of $P(\mathbf{x}|w)$ around the support points \mathbf{x}_i . The value of β is selected empirically on a validation set.

5.2 Sequence Models for Word Recognition

This section investigates machine learning sequence models for the task of handwritten document recognition. Although HMMs have been widely used for this task, most of them estimate the probability of an image feature generated by a label using a Gaussian distribution in the continuous feature space. Instead, we discussed the probability estimation in the discrete feature space and also propose combining kernel density estimates with HMMs for more accurate estimates in the continuous feature space. The conditional random field (CRF) is a recently developed graphical model for information extraction. This work introduces CRFs for historical handwritten document recognition for the first time. This is also the first application of CRF on a large state space.

5.2.1 Word Recognition with Discrete HMMs

We first test a HMM based on discrete features. As a generative model, a HMM estimates the joint probability of a hidden state sequence and a given observation sequence, which in our task are a sequence of words $S = \langle s_1, s_2, \dots, s_T \rangle$ and a sequence of discrete feature vectors $O = \langle o_1, o_2, \dots, o_T \rangle$ extracted from word images respectively:

$$P(S, O) = \prod_{t=0}^T P(s_t | s_{t-1}) P(o_t | s_t) \quad (5.12)$$

where T denotes the length of the sequences, and both transition probabilities $P(s_t | s_{t-1})$ and generative probabilities $P(o_t | s_t)$ are assumed to be subject to multinomial distributions. For each discrete feature in the feature vector $o_t = \langle o_{t1}, o_{t2}, \dots, o_{tm} \rangle$ extracted from the t -th word image, we assume it is independent of others given a hidden word s_t . Thus we have $P(o_t | s_t) = \prod_{i=0}^m P(o_{ti} | s_t)$. Given labeled handwritten documents as a training set τ , these probabilities can be easily computed using maximum likelihood estimation (MLE). Let w and v be two arbitrary words from vocabulary V , the transition probabilities are calculated as:

$$P(s_t = w | s_{t-1} = v) = \frac{\#(\text{word pair}(v, w) \text{ occurs in } \tau)}{\#(\text{word } v \text{ occurs in } \tau)} \quad (5.13)$$

Let I_w denotes all the word images labeled as w in the training set τ , then the generative probabilities are calculated as

$$P(o_{ti} | s_t = w) = \frac{\#(o_{ti} \text{ occurs as a feature of } I_w)}{\#(\text{all features of } I_w)} \quad (5.14)$$

The estimation of transition probabilities is done as in [79] and includes an averaging over the background distributions of these labels to smooth the probabilities:

$$\hat{P}(s_t = w | s_{t-1} = v) = \frac{1}{2}P(s_t = w | s_{t-1} = v) + \frac{1}{2}P(s_t = w) \quad (5.15)$$

where $P(s_t)$ is the background probability of label s_t in the collection τ and calculated as:

$$\hat{P}(s_t = w) = \frac{1}{2} \cdot \frac{\#(w \text{ in } \tau)}{\#(\text{all words in } \tau)} + \frac{1}{2} \cdot \frac{1}{|V|} \quad (5.16)$$

where $|V|$ is the size of the whole vocabulary.

Experiments in section 5.3 show this model doesn't perform that well.

5.2.1.1 Feature Probability Smoothing for HMMs

We explore using smoothing techniques to improve the performance of our original HMM model in section 5.2.1 - note that we are smoothing the features here not just the words as is usually done. The maximum likelihood estimate for generative probabilities in equation 5.14 is prone to be biased when the sample size is relative small. To alleviate this bias we smooth the generative probabilities using background probabilities of discrete features. Instead of a direct averaging as in [79], we tune the weight to optimize the likelihood on a held-out portion of a training sample. The formulation for feature probability smoothing has a linear form as follows:

$$\hat{P}(o_{ti} | s_t) = (1 - \lambda)P(o_{ti} | s_t) + \lambda P(o_{ti}) \quad (5.17)$$

where $P(o_{ti})$ is the background probability of discrete feature o_{ti} in the training set τ , directly calculated as the frequency of o_{ti} in τ . λ is the parameter of this linear smoothing and tuned through optimizing the likelihood on a validation set created from a portion of the training sample. Note that we use one identical smoothing parameter λ for all the generative probabilities.

5.2.2 Conditional Random Fields Framework

A CRF [73] is defined as an undirected graphical model used to calculate the probability of a possible label sequence conditioned on the observation sequence. The structure of random fields is basically an arbitrary graph obeying the Markov property. Let $O = \langle o_1, o_2, \dots, o_T \rangle$ and $S = \langle s_1, s_2, \dots, s_T \rangle$ denote the observation sequence and the label sequence respectively (In general CRFs, T need not be the same for O and S). A CRF formulates the conditional probability of S given O as:

$$P_{\theta}(S|O) = \frac{1}{Z_{\theta}(O)} \prod_q \left(\exp \left(\sum_k \lambda_k f_k(\mathbf{s}_q, \mathbf{o}_q) \right) \right) \quad (5.18)$$

where feature functions $\{f_k\}$ are defined on any subset of the random variables in the sequences $\mathbf{s}_q \subset S$, $\mathbf{o}_q \subset O$, λ_k is a learned weight for each feature function, and Z is a normalization factor over all possible state sequences:

$$Z_{\theta}(O) = \sum_{S \in S^T} \prod_q \left(\exp \left(\sum_k \lambda_k f_k(\mathbf{s}_q, \mathbf{o}_q) \right) \right) \quad (5.19)$$

In the simplest case, the graph is an undirected linear chain among output states, where CRFs make a first-order Markov independence assumption. Under this configuration, equation (5.18) is rewritten as:

$$P_{\theta}(S|O) = \frac{1}{Z_{\theta}(O)} \exp \left(\sum_{t=1}^T \sum_k \lambda_k f_k(s_t, s_{t-1}, O, t) \right) \quad (5.20)$$

A feature function (as distinct from an image feature) is defined over the current state, the previous state and image features computed over the whole observation sequence. Usually feature functions are binary predicates. For example, assume that the only image feature used is the length of the current word image. Then, the feature function f_k is 1 if the current state corresponds to “Washington”, the previous state to “Colonel” and the length

of the current word is 8, else $f_k = 0$. Note that even in the simplest case the number of weights is $\mathbf{O}(|\mathcal{S}|^2 T)$, where T is the sequence length and $|\mathcal{S}|$ is the size of the state space.

To reduce the number of parameters estimated, we further simplify the model into a conditionally-trained hidden Markov model, in which all the incoming transitions into a state will share the same weight and only at each separate step of the sequence we create weights for the current state and observation. In this case, the conditional probability becomes:

$$P_{\theta}(S|O) = \frac{1}{Z_{\theta}(O)} \exp \left(\sum_{t=1}^T \left(\sum_k (\lambda_k f_k(s_t, O, t)) + \sum_l (\mu_l g_l(s_t, s_{t-1})) \right) \right) \quad (5.21)$$

The number of parameters becomes $\mathbf{O}(|\mathcal{S}|T + |\mathcal{S}|^2)$.

5.2.2.1 Inference and Training in CRFs

Inference in CRFs is done as follows: Given an observation sequence \tilde{O} , from all possible label (state) sequences find the one \tilde{S} with the largest conditional probability over the distribution of $P(S|\tilde{O})$. This distribution is defined by the undirected graphic structure and the set of weights. Note the number of possible state sequences is exponential in the sequence length T . For an arbitrarily-structured CRF, it is intractable to calculate the normalization factor in equation (5.19). In HMM-Style CRFs, the normalization factor becomes:

$$Z_{\theta}(O) = \sum_{S \in \mathcal{S}^T} \exp \left(\sum_{t=1}^T \left(\sum_k (\lambda_k f_k(s_t, O, t)) + \sum_l (\mu_l g_l(s_t, s_{t-1})) \right) \right) \quad (5.22)$$

A dynamic programming algorithm like Viterbi decoding can be used to efficiently calculate the normalization factor.

The parameters $\theta = \{\lambda_k, \mu_l\}$ are estimated by optimizing the model over a training set consisting of labeled sequences, $D = \{O^{(i)}, S^{(i)}\}_{i=1}^N$, i.e. by trying to find the set of weights that maximize the log-likelihood of the labeled sequences in the training set:

$$L = \sum_{i=1}^N \log(P(S^{(i)}|O^{(i)})) - \sum_k \frac{\lambda_k^2}{2\sigma^2} \quad (5.23)$$

where the second term is a Gaussian prior over parameters smoothing over the training data [24].

Iterative scaling [73] is a general method to optimize parameters of CRFs and other exponential models. Sha and Pereira [138] use the limited memory quasi-Newton (L-BFGS) [69] method instead, which is shown to be several orders of magnitude faster than iterative scaling. Like iterative scaling, L-BFGS is also a gradient based optimization procedure but only requires the first-derivative of the objective function.

Here, every word is taken as a state so there could be thousands of states. We apply beam search for CRFs to significantly speed up the forward-backward procedure.

5.2.2.2 Training and Inference with Beam Search

The basic idea of beam search is simple. At each stage of the inference before passing any message to the next stage we first purge the states at this stage and keep only a small fraction of them. The number of states kept is usually called the beam width. So when using beam search for forward-backward procedure, the number of outgoing transitions from the current stage to the next stage will dramatically drop. Our goal is to prune as many states as possible while minimizing the performance loss. To determine the states to eliminate, we need some criteria (see below). Based on different criteria for purging states the beam search method works differently. Note that we talk about the probabilities of states here, but the actual implementations of inference and the forward-backward algorithm uses costs which are equal to the negative logarithm of the probabilities. In the implementation of beam search these costs need to be converted into probabilities.

1. N-best Beam Search

The simplest way to do beam search is to sort all the states in the current stage according to their probabilities in descending order. Then only the top N states are kept and the other states are eliminated.

2. Ratio Threshold based Beam Search

At stage i , we first determine the maximal probability P_i^m of all the states:

$$P_i^m = \max_s P_i(s) \quad (5.24)$$

Then a dynamic threshold is calculated based on the value P_i^m :

$$\tau_i = \frac{P_i^m}{K} \quad (5.25)$$

where K is a empirically selected constant. Then all states s' at this stage whose $P_i(s') < \tau_i$ will be eliminated.

This method doesn't have a fixed width of the beam at each stage and the criterion for purge is based on the individual probability of every state. This method is widely used with HMMs in speech recognition [65].

3. K-L Divergence based Beam Search

Pal *et al.* [116] recently present a novel beam search method based on K-L Divergence. The basic idea is to approximate single variable potentials with a constrained adaptively sized sum of Kronecker delta functions and minimize the KL divergence between the approximated distribution and its original. At each stage of the trellis for Viterbi inference or forward-backward procedure, the probabilities of all the states form some arbitrary discrete probability distribution, say p . Any subset of these states, indexed with $I = \{1, \dots, k\}$, forms some other distribution which could be

approximated as a sum of weighted and normalized Kronecker deltas, say q . The goal is to find the subset of these states which minimize the K-L divergence between p and q . Pal *et al.* [116] show this K-L divergence is equal to the negative logarithm of the sum of the probabilities of the subset states. More formally, suppose we want to find the minimal subset of states such that the K-L divergence $KL(q||p) \leq \epsilon$, then that implies minimizing $|I|$ s.t.

$$KL(q||p) = -\log \sum_{i \in I} p_i \leq \epsilon \quad (5.26)$$

The solution involves sorting the states according to their probabilities in a descending order and then selecting the states from the top until the sum of their probabilities satisfies equation (5.26).

5.2.2.3 Word Recognition with CRFs

Using CRFs for word recognition is straightforward when the data are given as labeled handwritten documents. Handwritten documents are segmented into word images. Each word image is an observation and its corresponding label is the value of its state in CRFs.

The ideal case in each instance is a labeled sentence. However this is intractable for degraded handwritten documents because important clues for sentences such as punctuations are faded or connected with words and hence hard to detect. In our case each sequence instance is a completely labeled page, with a length between 200 to 300 words. The drawback of using pages as training instances is that unreliable transitions between connections of two separate sentences will be involved and learned by the model.

Because both the size of the state space and the length of sequences in our project are large, we use the HMM-Style CRFs described by equation (5.21) in section 5.2.2.

Continuous image features are first extracted from each word image based on its scale and shape. Each continuous feature is quantized into a fixed number of bins. The set of

discretized features of each word image is its observation representation. Details on image features are given in section 5.3.

The model features are defined in a straightforward way. For example $f_k(s_t, O, t)$ is equal to 1 if the word image at position t is labeled as "Fredericksburgh" and its length is at level 10 (the highest level for our discretized image features), otherwise it is zero. The transition features $g_k(s_t, s_{t-1})$ are defined in a similar manner. For example $g_k(s_t, s_{t-1})$ is equal to 1 if the word image at position t is labeled as "Fredericksburgh" and the previous word is "defend", otherwise zero.

5.2.3 HMM with Gaussian Kernel Density Estimates

Both HMMs and CRFs discussed so far in this section use discrete word image features. Discrete features are easy to use, but could cause information loss when doing vector quantization. To utilize real-valued continuous features, we employ a Gaussian kernel density estimate for probabilities of generating features from a word state.

The HMM framework used here is just the same as in Section 5.2.1 except that now the observation sequence $O = \langle o_1, o_2, \dots, o_T \rangle$ is a sequence of real-valued feature vectors and the generative probability $P(o_i|w_i)$ is now estimated through a Gaussian kernel density function:

$$P(o_i|w_i) = \frac{1}{\|w_i\|} \sum_{j=1}^{\|w_i\|} \frac{\exp\{-(o_i - o_j)^T \Sigma^{-1} (o_i - o_j)\}}{\sqrt{2^k \pi^k |\Sigma|}} \quad (5.27)$$

As in the equation 5.11, each kernel is parameterized by the feature covariance matrix Σ and assumed $\Sigma = \beta \cdot I$, where I is the identity matrix and β plays the role of kernel bandwidth, which determines the smoothness of $P(o_i|w_i)$ around the support points o_j . The value of β is selected empirically on a validation set.

5.3 Experimental Results

5.3.1 Experimental Setup

Our evaluation dataset consists of 20 pages from a collection of letters by George Washington. This is a publicly available standard dataset used by [79]). Each page is accurately segmented into individual word images, each of which has been manually transcribed. We do not lowercase transcribed words, so “*region*” and “*Region*” are taken as two different words. There are 4865 words in the corpus in total and 1187 of them are unique. Figure 5.1 shows a part of a segmented page in our dataset.

For the purpose of fair comparisons between different models, we used the same features with those used by [79]. For a quick reference, the word images features are briefly described here. One can refer to [79] for more detailed information about the extracted features. Two kinds of features are extracted from each word image: scalar image features and profile-based image features. Scalar features consist of 6 different coarse measurements on each word image. Given a word image with a tight bounding box, the scalar features are respectively: the height h of the image in pixels, the width w of the image, the aspect ratio w/h , the area $w \cdot h$, the number of descenders in the word image, and the number of ascenders in the word. Profile-based features are computed from different profiles of the word image. This work uses projection, upper word and lower word profiles, each of which has the same length as the image width. The project profile is computed by summing over the pixel values at each image column of the word. The upper/lower profile is calculated as the distance from the upper/lower boundary of the word image to the closest ink pixel at each image column. After these profiles are computed, the Discrete Fourier Transform (DFT) is applied to each of them and a fixed number of lower-order DFT coefficients are used to represent each profile. So given 3 profiles and 7 DFT coefficients for each of them, there are 21 profile-based features. Plus the 6 scalar features, 27 features are used for each word image.

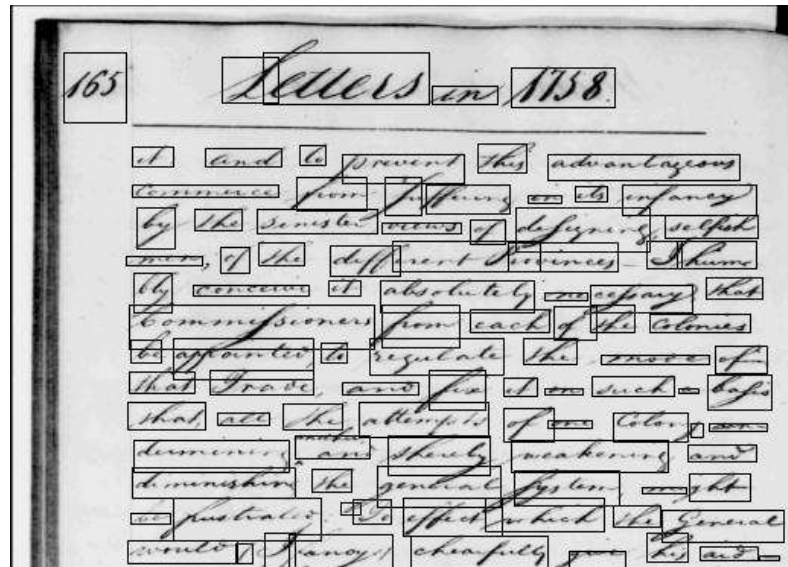


Figure 5.1. A part of one segmented page in our dataset.

We use word accuracy rate as our performance measure, i.e. the proportion of the words that are recovered exactly as they were in the manual transcript. 20-fold cross-validation is used to get a stable performance evaluation for each model. Each iteration leaves one page for test, and trains the model over the other 19 pages. We use the mean accuracy rate as the final evaluation measure. Since our dataset is relatively small, many words in the test set do not occur in any training pages - these are called out-of-vocabulary(OOV) terms as in [79] and cause errors of the recognition, we use two types of mean accuracy rate – mean accuracy rate with OOVs and mean accuracy rate without OOVs.

Since our data are from a collection of natural language documents (letters), the frequency of words can be approximated by a Zipf distribution. As Figure 5.2 shows, a few words have very high frequencies, however most words occur very infrequently. Over our whole dataset, 681 words have only one occurrence; 1008 words have less than 5 occurrences each but 30 words have 1856 occurrences in total. The unbalance and sparsity of training data for different words make the multi-classification problem intractable for some standard classifiers such as decision trees and neural networks as shown in [64].

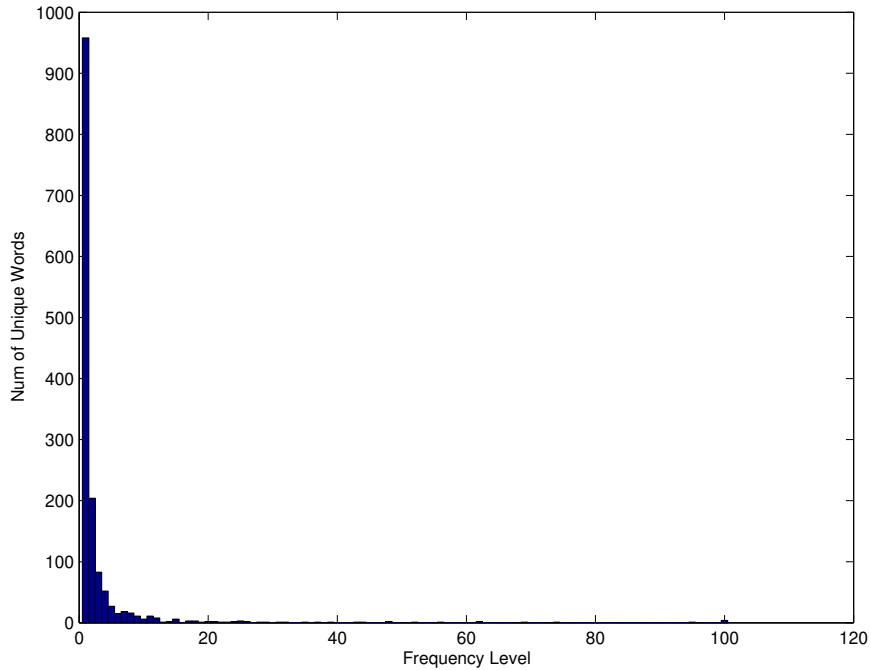


Figure 5.2. The histogram of the word frequency in our dataset, which is subject to a Zipf distribution.

5.3.2 Results on Different Classification Models

5.3.2.1 SVMs

We use the *MATLAB Support Vector Machine Toolbox* developed by Gavin Cawley to build the SVM model on the data. By using the 'max wins' algorithm, we tried linear kernels and polynomial kernels of degree 2 on the data.

Accuracy	with OOV	w/o OOV
Linear Kernel	0.3827	0.4642
Polynomial d-2	0.4463	0.5281

Table 5.1. Experimental results using SVMs. Word accuracy is reported for two different sets of words respectively – all words in the test set (with OOV) and the set without out-of-vocabulary (OOV) words included.

Table 5.1 shows the experimental results using support vector machines, from which we see that the polynomial kernel performs much better than the linear kernel. This is unsurprising since the kernel function plays a crucial role in SVMs. The kernel determines

the mapping of instances to a high dimensional space and whether the space is separable or not. However, it is not generally easy to locate the proper kernel. In other words, deciding to which space the original data should be projected requires a deeper understanding of the data - usually background knowledge is needed. In our case, both the linear kernel and the polynomial kernel of degree 2 do not work very well on the data. Other kernels that project the data into higher dimension spaces might help in this case but there is no simple way to determine these short of trying all of them.

5.3.2.2 Conditional Maximum Entropy Models

We use the maximum entropy toolkit from <http://homepages.inf.ed.ac.uk/s0450736/maxent-toolkit.html>, which was developed in C++ based on the java version at <http://maxent.sf.net>. To extract unigram and bigram discrete predicates in section 5.1.2, we linearly quantize each of the 27 continuous features into 19 bins. To test the influence of different numbers of bins into which the raw features is quantized, we also gradually changed the number of bins and re-ran the maximum entropy model. The performance varies only slightly with the change in the number of bins except at 100 bins the performance drops sharply.

Accuracy	with OOV	w/o OOV
Discrete Unigram	0.4164	0.4939
Discrete Unigram + Bigram	0.4432	0.5234
Raw Continuous	0.4161	0.5259
Distance Continuous	0.4454*	0.5629*
Raw + Dist	0.4367	0.5515

Table 5.2. Performance Comparisons for maximum entropy models and features

Table 5.2 shows the results of the maximum entropy model using discrete predicates and continuous predicates, from which we can see the distance features outperform all other features. Significance testing with the t-test shows that the difference between the results from distance features and raw continuous feature are significant with P-value of 0.03, while the P-value for distance feature vs unigram + bigram is 0.003. These numbers also show that using both unigram and bigram information outperforms only using unigram

Fixed Beam Width	10	80	105	106	107	132	264
Accuracy w/o OOV	0.001	0.001	0.001	0.645	0.645	0.645	0.645
Run Time (in Secs)	60	131	140	142	142	153	2944

Table 5.3. N-best Beam Search with different fixed beam widths

K in Equation (5.25)	1.0001	1.001	1.01	1.1	1.2	1.5	2
Accuracy w/o OOV	0.505	0.518	0.645	0.645	0.645	0.645	0.645
Run Time (in Secs)	97	99	107	1127	1238	1340	1527

Table 5.4. Ratio Threshold Beam Search with different K values

ϵ in $KL \leq \epsilon$	0.9	0.8	0.79	0.77	0.75	0.5	0
Accuracy w/o OOV	0.001	0.001	0.475	0.584	0.645	0.645	0.645
Run Time (in Secs)	62	70	75	87	91	209	2944

Table 5.5. KL Divergence Beam Search with different ϵ in $KL \leq \epsilon$

information by a small margin. Note that the concept of bigram here is defined between label states and features unlike that in HMMs where it is depicted as the dependency between label spaces. Since our dataset is relatively small and the vocabulary is huge, it is more difficult to capture useful bigram information for maximum entropy.

5.3.2.3 Naive Bayes with Gaussian Density Estimate

The mean accuracies achieved using the Naive Bayes model with Gaussian kernel density estimates are 0.542 with OOVs and 0.640 without OOVs. It is not surprising that Naive Bayes achieves good results on our task for at least two reasons. One is that the model provides prior probabilities of the words - that is the frequency of the words. This corresponds to unigram language model information used in [79] where it was shown to improve performance. Another is that the Gaussian density emphasizes the local information provided by each instance, which has been shown to be very useful in multimedia data analysis.

5.3.3 Tune and Compare Beam Search for Our CRF Model

All the three kinds of Beam Search in section 5.2.2.2 require us to experimentally decide the parameters controlling the width of beam. For this purpose, we select two pages

Accuracy Rate	with OOV	w/o OOV
SVM with polynomial d-2 kernel	0.446	0.528
ME with unigram (in [41])	0.416	0.494
ME with unigram and bigram	0.443	0.523
ME with distance predicates	0.445	0.563
Naive Bayes with Gaussian-KDE	0.542	0.640
CRFs with Ratio threshold beam search	0.417	0.503
CRFs with K-L divergence beam search	0.428	0.525
HMM with discrete features	0.336	0.404
HMM with discrete features after smoothing	0.504	0.595
HMM with continuous features (in [79])	0.497	0.586
HMM with Gaussian-KDE	0.583	0.688
HMM with continuous features + external corpora (in [79])	0.551	0.651
HMM with Gaussian-KDE + external corpora	0.611	0.723

Table 5.6. Results comparing different models. The external corpora used for transition estimates consists of a large electronic collection of writings by George Washington and Thomas Jefferson. CRFs cannot really be used with the continuous features described here and so are not directly comparable with HMMs using continuous features.

from our handwritten documents and use them as a training and a test example respectively. Even in this small dataset with 486 words in total, there are 264 states (unique words). On average there are less than 2 instances for each state, which means our model has been very starved for training data. But for this project, this is the largest unit we can use for tuning, including more pages results in a sharp increase in run time.

Tables 5.3, 5.4 and 5.5 show the results of using different values of tuning parameters for the N-best, ratio threshold and K-L divergence based beam search respectively. As the tables show the accuracy changes non-linearly with the tuning parameters. In certain regions it is relatively insensitive while in others it is very sensitive. The tables only show some values of the parameters - mostly those where very large changes occur.

Since with comparable accuracy N-best runs much slower than the other two methods, we did not do experiments using N-best over the whole dataset. We select $K = 1.01$ and $\epsilon = 0.75$ as the parameters of ratio threshold and KL-divergence respectively when testing over the whole dataset.

5.3.4 Result Comparisons

Table 5.6 compares the models we tested. To make the comparison fair, we report two set of results of the HMMs from a recent paper [79]. The first of the HMM models includes word bigrams obtained from the training set but not from the external corpora (the Naive Bayes model as well as the other models here do not use any bigrams). The second uses an external George Washington and Thomas Jefferson electronic text corpus for transition estimation. We can see that, with external text corpus both HMMs from [79] and our HMMs improve the performance significantly. From this table, we see that HMMs with a Gaussian density estimate achieved the best performance on our task. The t-test shows that the HMM with a Gaussian density estimate outperforms other HMMs significantly by a P-value of 0.01. With a Gaussian kernel density estimate, even naive Bayes can achieve very good results. The good performance of naive Bayes in our experiments shows that the prior probabilities (unigram information) is important for analysis on natural language document corpus (especially heavily unbalanced datasets). In contrast, prior distribution information is difficult to utilize in other discriminative models such as maximum entropy and SVM. Gaussian density estimates also show that localized models and local information are preferable for handwriting recognition. Such local information is suitable for many multimedia data problem in which each category could be a mixture of different patterns.

Table 5.6 also shows the results using CRFs with ratio threshold based beam search and K-L divergence based beam search respectively and HMMs with discrete features. For the maximum entropy model with unigram predicates, the model features are defined as those in CRFs for observational-state pairs, only observation and state at the same position are considered (see [41] for details). From the results, CRFs with a K-L divergence based beam search outperforms that with a ratio threshold based beam search by a small margin. Both CRFs outperform the maximum entropy model with unigram predicates, showing the importance of transition information. The HMM with discrete features where the features are not smoothed does not perform that well (the words are smoothed for all HMMs). HMM

performance can be improved substantially by also smoothing the features and as can be seen this makes them better than the CRF's. For reference, CRFs and Maximum Entropy use some kind of Gaussian prior for smoothing [24]. However, we believe that the poorer performance of CRFs is due to the substantially larger number of parameters that need to be estimated. In addition all the parameters are estimated at the same time while the probabilities for HMM's are estimated separately in this special case. More training data might improve the results but there are significant difficulties in using more training data. First, creating large amounts of training data is labor intensive and expensive. Second, CRFs are much slower and hence this would also require large amounts of computation. An alternative approach to increasing the amount of training data required would be to drastically reduce the state space. This would probably require dropping the whole word paradigm and moving to a character based approach with its attendant segmentation difficulties.

We have so far compared all techniques on the same features. Continuous features can substantially improve performance. However, using continuous features directly for CRFs is still problematical. CRFs require the continuous features to have a monotonic distribution. Most of the continuous features used in the paper here are not monotonic and in general it is non-trivial to find such features. Using the existing non-monotonic continuous features with CRFs leads to poor performance.

CHAPTER 6

CONCLUSION

In this dissertation, we presented our work on using statistical models for text query based on general image retrieval and historical handwritten manuscript recognition. We tackle these problems using automatic annotation based image retrieval, direct retrieval models for image retrieval and historical handwritten document recognition. Since the main goal of this work is to develop and compare different statistical models for the task of image retrieval, we have investigated the properties of various models, analyzed their drawbacks and benefits in modeling image contents and annotations, and developed new models more suitable for image annotation and retrieval. Besides the theoretical analysis of the effectiveness of our proposed models in text query based image retrieval, we have empirically demonstrated that our proposed annotation, recognition and retrieval models meet or exceed the state-of-the-art performance of previously proposed techniques. We summarize our findings and suggest future research directions to improve the retrieval performance.

6.1 Summary

The development of the new annotation based retrieval models was based on the examination of the word probabilities for image annotations employed by previous models. We observed and claimed that the multiple-Bernoulli model and normalized multinomial model are more suitable for formalizing the word distribution of annotated images although previous models for image annotation and retrieval used multinomial distributions. Based on this observation and the previous work on using relevance modeling approach for text retrieval

and image retrieval, we developed the multiple-Bernoulli relevance model to automatically annotate images and then retrieve images based on the annotation results. As a generative model, the multiple- Bernoulli relevance model estimated the joint distribution of visterms (image features or quantized image features) and annotations. This joint distribution is computed as an expectation over each image in the training set, leading to a non-parametric estimation. By comparing with other models using multinomial distribution for word probabilities, we demonstrated that the multiple-Bernoulli model has (statistically significant) better annotation performance. However, we found that the retrieval performance was poor when a multiple-Bernoulli language model was used for retrieval along with a multiple-Bernoulli annotation model. A multinomial language model combined with the multiple-Bernoulli annotation model resulted in superior retrieval performance. This inspired another annotation and retrieval model – the normalized continuous relevance model – which achieves the same performance on image annotation but much better retrieval performance than the multiple-Bernoulli relevance model. The normalized continuous relevance models padded all the annotations for each image to a fixed length using a special “null” word and then estimates the word probabilities using a multinomial model. Theoretically, we have shown the relationships between the continuous relevance model, the multiple-Bernoulli relevance model and the normalized continuous relevance model, and shown that the normalized continuous relevance model and the multiple-Bernoulli relevance model have the same annotation performance. Our experimental results demonstrated that the multiple-Bernoulli relevance model and the normalized continuous relevance models outperform most of the previous models.

Direct retrieval models were proposed to retrieve images without involving an explicit annotation step so that the models were trained by directly maximizing the retrieval performance rather than annotation performance. Our proposed direct retrieval model is based on the Markov random field and is flexible enough to model feature dependencies. We showed that the Markov random field model may be reduced to a linear combination of

different feature functions for the task of image retrieval. That makes it easy to incorporate different kinds of image features and the dependency between image features. We explored MRF based direct retrieval using both continuous image features and discrete features. Experimental results showed that our MRF with continuous features outperforms previous models for image retrieval. By modeling feature dependency it further improves the retrieval performance. Using an MRF model with discrete image features showed that large visual vocabularies improve the retrieval performance. In our discrete MRF model, large visual vocabularies are obtained by using hierarchical K-means to cluster image features. We demonstrated that the discrete MRF model runs much faster while having comparable retrieval performance with the continuous models.

For historical handwritten manuscripts, we mainly focused on statistical models for automatic recognition. The automatic recognition results can be used for retrieving the original manuscripts. We adopted the holistic word recognition approach to avoid the character segmentation problem which is one of the most challenging problems in handwritten document analysis. We tackled the holistic word recognition problem from two aspects. One was to take this problem as a multiple-classification problem using different machine learning classifiers. Specifically, we introduced and compared support vector machine (SVM), maximum entropy and naive Bayes models. The other aspect used graphical models for historical handwritten document recognition at the word level. Compared with non-graphical classifiers, graphical models can take the advantages of language modeling to capture the word dependency/transitions from the manuscript corpus. In particular, we studied the application of hidden Markov model (HMM) and conditional random fields (CRF) for this task. Although HMMs have been widely used for handwriting recognition, we explored ways to improve HMMs through probability smoothing of discrete features and kernel density estimation for generative probability estimation. It was also the first time that the CRF has been applied to historical handwritten document recognition. To deal with the large state space problem, we investigated different pruning techniques for CRFs. We compared

all these models and the experimental results showed that a HMM with a kernel density estimate outperforms other models.

6.2 Future Work

The focus of this work is on the exploration of new statistical models for the task of the text query based image retrieval. Appropriate statistical models are critical to effectively learn image semantics in order to achieve excellent retrieval performance. The remainder of this chapter will briefly outline promising directions for improving text query based image retrieval and handwritten document recognition and retrieval.

6.2.1 Models for General Image Retrieval

The models proposed in this work for general image retrieval have demonstrated comparable or superior performance compared with previously proposed techniques. We believe the models can be further improved in many ways:

1. Improving the feature dependency modeling. When modeling the feature dependency in our continuous Markov random field, we only considered the nearest region pairs. The MRF retrieval framework allows arbitrary dependency structures in principle. More advanced feature dependency, e.g. dependency of a group of features, may improve the learning of image semantics and benefit image retrieval.
2. Modeling word dependency. The current MRFs only considered the feature dependency and assumed that words are independent of each other. However, word independence may not be true for real image annotation and retrieval. For example, the annotation word "airplane" may be dependent on the annotation word "sky" and exclude the word "fish" most of the time, although an airplane and a fish may bear some visual similarities. Another example is that in the case of hierarchical annotation, a word always has strong dependencies with its children and parent annotation words. So modeling word dependency may be useful for annotating and retrieving images.

3. Incorporating different visual features. The proposed MRF models may be extended to incorporate different visual information, e.g. features from segmented/partitioned regions or local interest point based features. Different features describe the image from different aspects. For example, local descriptors focus on salient parts of an images which are usually captured in object corner or edge regions. However, local descriptors may ignore a lot of image regions which have relatively uniform texture or color information, such as image regions corresponding to sky, grass, beach, water. These regions provide important context information for object recognition or annotation. So combining different visual features in the retrieval model may improve the retrieval performance.

6.2.2 Models for Historical Handwritten Documents

Our investigation of historical handwritten document recognition is focused on holistic word recognition. Although this avoided the challenges of character segmentation, it suffers some problems e.g. out-of-vocabulary words cannot be recognized and it is sensitive to over- or under-segmentation. It is worth investigating sub-word modeling techniques to do segmentation and recognition simultaneously. A sub-word model could be a character, bi-character (a pair of consecutive characters) or tri-character (a triple of consecutive characters) model. Although there is a lot of work employing HMMs for handwritten recognition at the character level, one common issue with them is that similarity information among characters of an word image is generally ignored when decoding characters. One possible way is that we can first train a set of HMMs for single characters, then train and refine the models for bi-characters or tri-characters through tying their parameters based on similarities between the models. When decoding a line of handwritten image, we use the similarity information between sub-word parts of the test image as a constraint over the decoding algorithm. i.e. more similar pairs of sub-word parts of a test image should have

higher probabilities of being decoded into the same label while dissimilar pairs should have a lower probability of being labeled the same.

This work investigated historical handwritten document recognition separately and suggests retrieval based on the recognition results. Since scanned historical handwritten documents can be taken as a special kind of general image, we believe our direct retrieval models may also be applied to them. As is the case for general image retrieval, our MRF based direct retrieval model can avoid the *metric divergence* between recognition performance and retrieval performance. It can model the feature dependency between different local characteristics for handwritings, which should be very important considering that words are meaningful permutations of individual characters.

APPENDIX A

ANNOTATION WORDS

We provide here the 260 annotation words present in the human annotations of the test set of the 5k-images Corel dataset:

water sky tree people grass buildings mountain flowers snow clouds rocks stone street plane field bear sand birds beach boats jet leaf cars plants house bridge valley polar garden hills close-up ruins statue tracks horses sun ice wall ocean cat train temple tiger scotland coral swimmers coast window branch pool foals sunset sculpture frost nest head fox forest mare city railroad ground shops petals horizon arch reefs palace reflection park desert skyline locomotive shore pillar castle town river road deer waves smoke sea church tower market zebra sign light coyote courtyard bush village pyramid landscape fence door roofs black tundra shadows elk display island rodent harbor grizzly flight stems runway woman turn tulip palm man dunes antlers restaurant formula fish white-tailed kauai buddha hut herd wood formation food museum oahu indian ships prototype prop lizard hillside hats flag farms bengal gate frozen face moose log caribou canyon bulls buddhist baby arctic tables night hotel fountain costume stairs path lawn hawaii giraffe meadow maui land cubs crystals booby windmills tusks sphinx mule monastery lake reptile monks marine iguana elephant cottage clothes ceremony anemone tails pots girl fruit f-16 albatross horns fly cow athlete shrubs relief post entrance crab column antelope vines vegetation sunrise slope plaza goat fan squirrel mosque lion glass blooms barn architecture vineyard sheep monument mist lynx interior dress detail cathedral canal african terrace silhouette outside kit den decoration cactus balcony art truck store porcupine nets needles marsh lighthouse dance

basket whales trunk peaks dock cave vendor snake festival doorway crafts butterfly vehicle
sidewalk calf cafe sails orchid cougar

APPENDIX B

IMAGE FEATURES

We list here the 36 features extracted from each segmentation for the large Corel dataset (160CD) and the experiments of CRM with the segmentations for the 5K-images Corel set (Features are kindly provided by Kobus Barnard, please refer to Barnard *et al.* [9]). The 36 features are listed below, where the number in each parenthesis indicates the dimension of each kind of features:

1. Area, x-location, y-location, boundary $_lengh^2/area$, convexity, moment-of-inertia (6)
2. Average RGB (3)
3. Average RGB (3, duplicated to increase RGB color's weight)
4. RGB standard derivative (3)
5. Average L*a*b (3)
6. Average L*a*b (3, duplicated to increase Lab color's weight)
7. L*a*b standard derivative (3)
8. Mean oriented energy, 30 degree increments (12)

We list here the 30 features we extracted from each rectangular regions for the 5K Corel images:

1. Average RGB (3)
2. Average RGB (3, duplicated to increase RGB color's weight)

3. RGB standard derivative (3)
4. Average L*a*b (3)
5. Average L*a*b (3, duplicated to increase Lab color's weight)
6. L*a*b standard derivative (3)
7. Gabor Energy in 3 scales and 4 directions (12)

32 Features are used for the TRECVID dataset, kindly provided by Giridharan Iyengar at IBM Research. They are listed here:

1. Average L*a*b (3)
2. Average L*a*b (3, duplicated to increase Lab color's weight)
3. L*a*b standard derivative (3)
4. Gray-level co-occurrence matrices (GLCM)(20)

BIBLIOGRAPHY

- [1] <http://www.flickr.com/>.
- [2] Trec video conference. <http://www-nlpir.nist.gov/projects/trecvid>.
- [3] A. Balasubramanian, M. Meshesha and C. V. Jawahar. Retrieval from Document Image Collections. In *Proceedings of Seventh IAPR Workshop on Document Analysis Systems (2006)*, pp. 1–12.
- [4] aes, J. Magalh and Ruger, S. Logistic regression of generic codebooks for semantic image retrieval. In *In CIVR06 (2006)*.
- [5] Agarwal, S., and Roth, D. Learning a sparse representation for object detection. In *Proc. ECCV (2002)*, pp. 113–130.
- [6] Antonacopoulos, A., Gatos, B., and Karatzas, D. Icdar 2003 page segmentation competition. In *In Proc. of the 7th Int'l Conf. on Document Analysis and Recognition (Edinburgh, Scotland, August 3-6 2003)*, vol. 2, pp. 688–692.
- [7] A.Pentland, R.W.Picard, and S.Sclaroff. Photobook: Tools for content-base manipulation of image databases. In *Storage and Retrieval of Image & Video Database II, Proceedings of SPIE 2185 (1994)*, pp. 34–47.
- [8] Barnard, K., and Forsyth, D. Learning the semantics of words and pictures. In *Proc. ICCV (2001)*, vol. 2, pp. 408–415.
- [9] Barnard, Kobus, Duygulu, Pinar, de Freitas, Nando, Forsyth, David, Blei, David, and Jordan, Michael I. Matching words and pictures. *JMLR 3 (2003)*, 1107–1135.
- [10] Belongie, S., Malik, J., and Puzicha., J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24:24 (2002)*, 509–522.
- [11] Berg, Tamara, and Forsyth, David. Animals on the web. In *In CVPR 2006 (NYC, 2006)*, pp. 1460–1470.
- [12] Berger, A., Pietra, S. D., and Pietra, V. D. A maximum entropy approach to natural language processing. In *Computational Linguistics (March 1996)*, pp. 39–71.
- [13] Bimbo, A. Del, and Pala, P. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Analysis and Machine Intelligence 19, 2 (1997)*, 121–132.

- [14] Blei, D. M., and Jordan, M. I. Modeling annotated data. In *Proc. of the 26th Annual Int'l ACM SIGIR Conf.* (Toronto, Canada, July 28-August 1 2003), pp. 127–134.
- [15] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [16] Bozinovic, R.M., and Srihari, S.N. Off-line cursive script word recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 11, 1 (1989), 68–83.
- [17] Breuel, T.M. An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis. In *In Proc. of the 7th Intl Conf. on Document Analysis and Recognition* (Edinburgh, Scotland, August 3-6 2003), pp. 66–70.
- [18] Buehler, Eugen C, and H.Ungar, Lyle. Maximum entropy methods for biological sequence modeling. In *Workshop on Data Mining in Bioinformatics of KDD01* (2001).
- [19] Cao, H., Ding, X., and Liu, C. Rectifying the bound document image captured by the camera: A model based approach. In *Proc. of the 7th Int'l Conf. on Document Analysis and Recognition* (Edinburgh, Scotland, August 3-6 2003), vol. 1, pp. 71–75.
- [20] Carbonetto, de Freitas, N., and Barnard., K. A statistical model for general contextual object recognition. In *Proc. ECCV* (2004).
- [21] Carbonetto, de Freitas, N., Gustafson, P., and Thompson., N. Bayesian feature weighting for unsupervised learning, with application to object recognition. In *Proc. of the 9th Int'l Workshop on Artificial Intelligence and Statistics* (2003).
- [22] Carneiro, G., Chan, A. B., Moreno, P. J., and Vasconcelos, N. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. PAMI* 29(3) (2007), 394–410.
- [23] Chan, J., Ziftci, C., and Forsyth, D. Searching off-line arabic documents. In *CVPR 2006* (NYC, 2006).
- [24] Chen, S.F., and Rosenfeld., R. A gaussian prior for smoothing maximum entropy models. Tech. rep., 2001.
- [25] Cunningham, S.J., and Masoodian, M. Looking for a picture: An analysis of everyday image information searching. In *the 6th ACM/IEEE-CS Joint Conference on Digital Libraries 2006* (2006), pp. 198–199.
- [26] Cusano, C, Ciocca, G, and Scettini, R. Image annotation using svm. In *Proceedings of the SPIE, Volume 5304* (2003), pp. 330–338.
- [27] Das, Madirakshi, and Manmatha, R. Automatic segmentation and indexing in a database of bird images. In *ICCV 2001, the 8th IEEE International Conf. on Computer Vision* (Vancouver, 2001), vol. 2, pp. 351–358.

- [28] Das, Madirakshi, Manmatha, R., and Riseman, Edward M. Indexing flowers by color names using domain knowledge-driven segmentation. In *Proc. of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)* (Princeton, NJ, Oct 1998), pp. 94–99.
- [29] Datta, Ritendra, Joshi, Dhiraj, Li, Jia, and Wang, James Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys* (2007).
- [30] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., and Harshman, R.A. Indexing by latent semantic analysis. In *Journal of the American Society of Information Science* (1990).
- [31] D.Miller, T.Leek, and R.Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd International ACM SIGIR conference* (1999), pp. 214–221.
- [32] Duygulu, P., Barnard, K., de Freitas, N., and Forsyth, D. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. ECCV* (2002), pp. 97–112.
- [33] Eakins, J.P., Riley, K.J., and Edwards, J.D. Shape feature matching for trademark image retrieval. In *Image and Video Retrieval* (2003).
- [34] Edwards, G., Cootes, T., and Taylor, C. Face recognition using active appearance models. In *ECCV* (1998).
- [35] Edwards, J., and Forsyth, D. Searching for character models. In *the Proc. of NIPS 2005* (2005).
- [36] Edwards, J., Teh, Y. Whye, Forsyth, D., Bock, R., Maire, M., and Vesom, G. Making latin manuscripts searchable using ghms. In *To appear in the Proc. of NIPS 2004* (2004).
- [37] Epshtein, B., and Ullman, S. Identifying semantically equivalent object fragments. In *the proceeding of Conference on Computer Vision and Pattern Recognition 2005* (San Diego, 2005), pp. 2–9.
- [38] Fan, Jianping, Luo, Hangzai, and Gao, Yuli. Learning the semantics of images by using unlabeled samples. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005).
- [39] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE Tran. on PAMI* 28, 4 (2006), 594–611.
- [40] Feldbach, M., and Tonnies, K. D. Line detection and segmentation in historical church registers. In *In Proc. of the 6th Intl Conf. on Document Analysis and Recognition* (2001).

- [41] Feng, S.L., and Manmatha, R. Classification models for historical documents recognition. In *In the Proc. of ICDAR'05* (2005), pp. 528–532.
- [42] Feng, S.L., and Manmatha, R. Exploring the use of conditional random field models and hmms for historical handwritten document recognition. In *the Proceedings of the 2nd IEEE International Conference on Document Image Analysis for Libraries (DIAL 06)* (2006), pp. 30–37.
- [43] Feng, S.L., Manmatha, R., and Lavrenko, V. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2004), pp. 1002–1009.
- [44] Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. Learning object categories from google's image search. In *Proc. ICCV* (2005).
- [45] Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. In *Proc. of CVPR'03* (2003), vol. II, pp. 264–271.
- [46] Fergus, R., Perona, P., and Zisserman, A. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR* (2005), vol. 1, pp. 380–387.
- [47] Finlayson, G. Color in perspective. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 10 (1996), 1034–1038.
- [48] Fischler, M. A., and Elschlager, R. A. The representation and matching of pictorial structures. *IEEE Trans. Comput.* 22, 1 (1973), 67–92.
- [49] Forsyth, D. A. A novel algorithm for color constancy. *International Journal of Computer Vision* 5, 1 (1990), 5–36.
- [50] F.Song, and W.B.Croft. A general language model for information retrieval. In *Proceedings on the 22nd annual international ACM SIGIR conference* (1999), pp. 279–280.
- [51] Gatos, B., Konidakis, T., Ntzios, K., Pratikakis, I., and Perantonis, S. A segmentation-free approach for keyword search in historical typewritten documents. In *In ICDAR 2006* (2006).
- [52] Ghoshal, A., Ircing, P., and Khudanpur, S. Hidden markov models for automatic annotation and content-based retrieval of images and videos. In *Proc. ACM SIGIR* (2005).
- [53] Gorski, N., Anisimov, V., Augustin, E., Baret, O., Price, D., and Simmon, J. A2ia check reader: A family of bank check recognition system. In *Proc. of the 5th Int'l Conf. on Document Analysis and Recognition* (1999), pp. 523–526.
- [54] Haralick, R. M. Statistical and structural approaches to texture. In *the IEEE* (1979), vol. 67, pp. 786–804.

- [55] Hare, J. S., Lewis, P. H., Enser, P.G.B., and Sandom, C. J. A linear-algebraic technique with an application in semantic image retrieval. In *In CIVR06* (2006).
- [56] Harris, C.J., and Stephens, M. A combined corner and edge detector. In *proc. 4th Alvey Vision Conference* (1988), pp. 147–151.
- [57] Hiemstra, D. Using language models for information retrieval. *PhD Thesis, University of Twente* (2001).
- [58] Houghton-Mifflin. *The Psychobiology of Language*. 1935.
- [59] Howe, N., Rath, T., and Manmatha, R. Boosted decision trees for word recognition in handwritten document retrieval. In *The 28th Annual international ACM SIGIR Conference* (2005), pp. 377–383.
- [60] <http://www.clopinet.com/isabelle/Projects/SVM/applist.html>.
- [61] Huang, J., Kumar, S. Ravi, Mitra, M., Zhu, W.-J., and Zabih, R. Spatial color indexing and applications. *Int. J. Computer Vision* 35 3 (1999), 245–268.
- [62] Hutchison, L. A. D., and Barrett, W. A. Fast registration of tabular document images using the fourier-mellin transform. In *In Proc. of the Int'l Workshop on Document Image Analysis for Libraries* (2004).
- [63] Iyengar, G., Duygulu, P., Feng, S., Ircing, P., Khudanpur, S., Klakow, D., Krause, M., Manmatha, R., Nock, H., Petkova, D., Pytlik, B., and Virga, P. Joint visual-text modeling for automatic retrieval of multimedia documents. In *Proc. ACM Multimedia* (2005).
- [64] Japkowicz, N., and Stephen, S. The class imbalance problem: A systematic study. In *Intelligent Data Analysis* (2002).
- [65] Jelinek, F. *Statistical Methods for Speech Recognition*. MIT Press, 2001.
- [66] Jeon, J., Lavrenko, V., and Manmatha, R. Automatic image annotation and retrieval using cross-media relevance models. In *Proc. of the 26th Annual Int'l ACM SIGIR Conf.* (Toronto, Canada, July 28-August 1 2003), pp. 119–126.
- [67] Jeon, J., and Manmatha, R. Using maximum entropy for automatic image annotation. In *Proceedings of the 3rd International Conference on Image and Video Retrieval* (2004), pp. 24–32.
- [68] Jeon, J., and Manmatha, R. Using maximum entropy for automatic image annotation. In *Proc. CIVR* (2004), pp. 24–32.
- [69] J.Norcedal, and Wright., S.J. *Numerical Optimization*. Springer, 1999.
- [70] Kavallieratou, E., Fakotakis, N., and Kokkinakis, G. A slant removal algorithm. *Pattern Recognition* 33, 7 (2000), 1261–1262.

- [71] Kolcz, A., Alspector, J., Augusteijn, M., Carlson, R., and Popescu, G. V. A. line-oriented approach to word spotting in handwritten documents. In *Pattern Analysis and Applications* (2000), pp. 153–168.
- [72] K.Sung, and Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. PAMI* 20(1) (Jan 1998), 39–51.
- [73] Lafferty, J., McCallum, A., and Pereira., F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML, 2001* (2001).
- [74] Lafferty, J., and Zhai, C. Document language models, query models, and risk minimization for information retrieval. In *the 24th annual international ACM SIGIR conference* (2001), pp. 111–119.
- [75] Lavrenko, V. A generative theory of relevance. *Ph.D Dissertation* (2004).
- [76] Lavrenko, V., Choquette, M., and Croft, W.B. Cross-lingual relevance models. In *the 25th annual international ACM SIGIR conference* (2002), pp. 175–182.
- [77] Lavrenko, V., Feng, S. L., and Manmatha, R. Statistical models for automatic video annotation and retrieval. In *Proc. ICASSP* (2004), pp. 1044–1047.
- [78] Lavrenko, V., Manmatha, R., and Jeon, J. A model for learning the semantics of pictures. In *In Proceedings of NIPS03* (2003).
- [79] Lavrenko, V., Rath, T., and Manmatha, R. Holistic word recognition for handwritten historical documents. In *the Proc. of DIAL'04* (2004), pp. 278–287.
- [80] Lavrenko, V., Rath, T. M., and Manmatha, R. Holistic word recognition for handwritten historical documents. In *Proc. of the Int'l Workshop on Document Image Analysis for Libraries* (Palo Alto, CA, January 23-24 2004), pp. 278–287.
- [81] Lavrenko, Victor, and Croft, W.B. Relevance-based language models. In *the 24th annual international ACM SIGIR conference* (2001), pp. 120–127.
- [82] Lazebnik, S., Schmid, C., and Ponce, J. A maximum entropy framework for part-based texture and object recognition. In *Proc. ICCV* (2005), pp. 832–838.
- [83] Li, Jia, and Wang, James Z. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on PAMI* 25, 9 (2003), 1075–1088.
- [84] Li, S.Z. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag Telos, 2001. 2Rev Ed edition.
- [85] Lowe, D. Three-dimensional object recognition from single two-dimensional images, 1987.
- [86] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* (2004), 91–110.

- [87] Lowe, David G. Object recognition from local scale-invariant features. In *International Conference on Computer Vision* (Corfu, Greece, 1999).
- [88] Lu, Y., and Shridhar, M. Character segmentation in handwritten words - an overview. *Pattern Recognition* 29 (1996), 77–96.
- [89] Madhvanath, S., and Govindaraju, V. Using holistic features in handwritten word recognition. In *Proc. of the U.S. Postal Service Advanced Technology Conf.* (Washington, DC, November 30 - December 2 1992), p. 183C199.
- [90] Madhvanath, S., and Govindaraju, V. The role of holistic paradigms in handwritten word recognition. *Trans. on Pattern Analysis and Machine Intelligence* 23, 2 (2001), 149–164.
- [91] Mahadevan, U., and Nagabushnam, R. C. Gap metrics for word separation in handwritten lines. In *Proc. of the 3rd Intl Conf. on Document Analysis and Recognition* (Montreal, Canada, August 14-15), volume = 1, pages = 124–127 1995).
- [92] Mahadevan, U., and Nagabushnam, R.C. Gap metrics for word separation in handwritten lines. In *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition* (Montreal, Canada, August 14–15 1995), pp. 124–127.
- [93] Manjunath, B., and Ma, W.Y. Texture features for browsing and retrieval of image data. In *IEEE Trans. Pattern Analysis and Machine Intelligence* (1996), vol. 18, pp. 837–842.
- [94] Manmatha, R., and Croft, W. B. Word spotting: Indexing handwritten manuscripts. *Intelligent Multimedia Information Retrieval* (1997), 43–64.
- [95] Manmatha, R., Han, C., and Riseman, E. M. Word spotting: A new approach to indexing handwriting. In *Proc. of the Conf. on Computer Vision and Pattern Recognition* (San Francisco, CA, June 1996), pp. 631–637.
- [96] Manmatha, R., Han, C., Riseman, E. M., and Croft, W. B. Indexing handwriting using word matching. In *Digital Libraries 96: 1st ACM Intl Conf. on Digital Libraries* (Bethesda, MD, March 1996), pp. 151–159.
- [97] Manmatha, R., and Rothfeder, J.L. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2005). (2005).
- [98] Manmatha, R., and Srimal, N. Scale space technique for word segmentation in handwritten manuscripts. In *Proc. of the Second Intl Conf. on Scale-Space Theories in Computer Vision* (Corfu, Greece, September 26-27 1999), pp. 22–33.
- [99] Maree, R., Geurts, P., Piater, J., and Wehenkel, L. Random subwindows for robust image classification. In *Proc. CVPR* (2005), pp. 34–40.

- [100] Marti, U.-V., and Bunke, H. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In *Proc. of the 6th Int'l Conf. on Document Analysis and Recognition* (Seattle, WA, September 10-13 2001), pp. 159–163.
- [101] Marti, U.-V., and Bunke, H. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *the Jnl. of Pattern Recognition and Artificial Intelligence* 15 (2001), 65–90.
- [102] Matas, J., Chum, O., Martin, U., and Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions,. In *Proceedings of the British Machine Vision Conference* (2002), pp. 384–393.
- [103] Metzler, D., and Croft, W. B. A markov random field model for term dependencies. In *The 28th Annual international ACM SIGIR Conference* (2005), pp. 472–479.
- [104] Metzler, D., and Manmatha, R. An inference network approach to image retrieval. In *Proc. CIVR* (2004), pp. 42–50.
- [105] Mokhtarian, F. Silhouette-based isolated object recognition through curvature scale space. *IEEE Trans. Pattern Analysis and Machine Intelligence* 17, 5 (1995), 539–544.
- [106] Morgan, W., Greiff, W., and Henderson., J. Direct maximization of average precision by hill-climbing with a comparison to a maximum entropy approach. *Technical report, MITRE* (2004).
- [107] Mori, Y., Takahashi, H., and Oka, R. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM'99 First International Workshop on Multimedia Intelligent Storage and Retrieval Management* (1999).
- [108] Murase, H., and Nayar, S. isual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision* (1994).
- [109] Murphy, K., Torralba, A., and Freeman, W. T. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *in NIPS03* (2003).
- [110] Natsev, A., Naphade, M. R., and Tešić, J. Learning the semantics of multimedia queries and concepts from a small number of examples. In *In ACM Multimedia 06* (Singapore, November 6–11 2006).
- [111] Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., and Taubin., G. The qbic project: Querying images by content using color, texture, and shape. In *Pocceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases* (San Jose, CA, February 1993), pp. 173–187.
- [112] Nister, D., and Stewenius, H. Scalable recognition with a vocabulary tree. In *Proc. CVPR 2006* (2006).

- [113] O. D. Trier, A. K. Jain, and Taxt, T. Feature extraction methods for character recognition - a survey. *Pattern Recognition* 29, 4 (1996), 641–662.
- [114] Ozkan, D., and Duygulu, P. Finding people frequently appearing in news. In *In CIVR06* (2006).
- [115] P. Carbonetto, N. de Freitas. Why can't jos read? the problem of learning semantic associations in a robot environment. In *Proc. HLT Workshop on Learning Word Meaning from Non-Linguistic Data* (2003).
- [116] Pal, C., Sutton, C., and McCallum, A. Constrained kronecker deltas for fast approximate inference and estimation. In *UAI 2005* (2005).
- [117] Petrakis, E., and Faloutsos, A. Similarity searching in medical image databases. *IEEE Trans. Knowledge and Data Engineering* 9, 3 (1997), 435–447.
- [118] Philbin, J., Chum, Ondrej, Isard, M., Sivic, J., and Zisserman, Andrew. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR 2007* (2007).
- [119] Picard, R.W., and T.P.Minka. Vision texture for annotation. *ACM/Springer Journal of Multimedia Systems* 3 (1995), 33–14.
- [120] Plamondon, R., and Srihari, S. N. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (2000), 63–84.
- [121] Ponte, J. M., and Croft, W. B. A language modeling approach to information retrieval. In *the 21st annual international ACM SIGIR conference* (1998), pp. 275–281.
- [122] P.Virga, and Duygulu., P. Systematic evaluation of machine translation methods for image and video annotation. *To appear at the Conference on Video and Image Retrieval, CIVR'05* (2005).
- [123] Quattoni, A., Collins, M., and Darrell, T. Conditional random fields for object recognition. In *In NIPS2004* (2004).
- [124] Rasiwasia, N., Vasconcelos, N., and Moreno, P.J. Query by semantic example. In *In CIVR06* (2006), pp. 51–60.
- [125] Rath, T., and Manmatha, R. Word image matching using dynamic time warping. In *Proceedings of CVPR'03* (2003), vol. 2, pp. 521–527.
- [126] Rath, T., Manmatha, R., and Lavrenko., V. A search engine for historical manuscript images. In *the Proceedings of SIGIR'04* (2004).

- [127] Rath, T. M., Kane, S., Lehman, A., Partridge, E., and Manmatha, R. Indexing for a digital library of george washingtons manuscripts: A study of word matching techniques. *Tech. rep., Center for Intelligent Information Retrieval Univ. of Massachusetts Amherst, 2000.* (2000).
- [128] Rath, T. M., and Manmatha, R. Word spotting for historical documents. *International Journal on Document Analysis and Recognition* 9 (2007), 139–152.
- [129] Rath, T. M., and Manmatha, R. Features for word spotting in historical manuscripts. In *Proc. of the 7th Intl Conf. on Document Analysis and Recognition* (Edinburgh, Scotland, August 3-6 2003), pp. 218C–222.
- [130] Rath, T.M., and Manmatha, R. Lower-bounding of dynamic time warping distances for multivariate time series. *Tech. rep., Center for Intelligent Information Retrieval, Univ. of Massachusetts Amherst, 2003.* (2003).
- [131] Ratnaparkhi, A. A simple introduction to maximum entropy models for natural language processing. *Tech. rep., 1997.*
- [132] Rosenfeld, R. A maximum entropy approach to adaptive statistical language modelling. In *Computer, Speech and Language* (1996), pp. 187–228.
- [133] Rothfeder, J.L., Feng, S.L., and Rath, T.M. Using corner feature correspondences to rank word images by similarity. In *In Proc. of the Workshop on Document Image Analysis and Retrieval* (Madison, WI, June 2003).
- [134] Rubner, Y. The earth-mover’s distance as a metric for image retrieval. In *Technical Report STAN-CS-TN-98-86, Stanford University, 1998.* (1998).
- [135] Schmid, C., and Mohr, R. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence* 19, 5 (1997).
- [136] Schneiderman, H., and Kanade, T. A statistical method for 3d object detection applied to faces and cars. *Proc. IEEE CVPR ’00* (2000).
- [137] Schomaker, L., and Segers, E. Advances in handwriting recognition. In *S.-W. Lee, ed., vol. 34, World Scientific, 1999* (1999).
- [138] Sha, F., and Pereira., F. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology, NAACL, 2003* (2003).
- [139] Shi, Jianbo, and Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.
- [140] Shi, R., Chua, T., Lee, C., and Gao, S. Bayesian learning of hierarchical multinomial mixture models of concepts for automatic image annotation. In *Proc. of CIVR06* (2006), pp. 102 – 112.

- [141] Sivic, J., Russell, B., Efros, A., Zisserman, A., and Freeman, W. Discovering object categories in image collections. In *Technical Report A.I. Memo 2005-005, MIT* (2005).
- [142] Sivic, J., and Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV 2003* (2003).
- [143] Smith, J., and Chang, S. Querying by color regions using the visualseek content-based visual query system, 1996.
- [144] Smith, J., and Chang, S.-F. Integrated spatial and feature image query. *IEEE Trans. Knowledge and Data Engineering* 9, 3 (1997), 435–447.
- [145] Steinherz, T., Rivlin, E., and Intrator, N. Offline cursive script word recognition - a survey. *Int'l Journal on Document Analysis and Recognition* 2 (1999), 90–110.
- [146] Stricker, M. A., and Orengo, M. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)* (1995), pp. 381–392.
- [147] Swain, M., and Ballard, B. Color indexing. *Int. Jnl. Computer Vision* 7, 1 (1991), 11–32.
- [148] S.Z.Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag Telos; 2Rev Ed edition, 2001.
- [149] Tamura, H., Mori, S., and Yamawaki., T. Textural features corresponding to visual perception. In *IEEE Trans. Systems, Man, and Cybernetics*, 8(6):460–72 (June 1978).
- [150] Tan, C. L., Cao, R., and Shen, P. Restoration of archival documents using a wavelet technique. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24, 10 (2002), 1399–1404.
- [151] Terasawa, Kengo, Nagasaki, Takeshi, and Kawashima, Toshio. Eigenspace method for text retrieval in historical document images. In *Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR05)* (2005).
- [152] Torralba, A., Murphy, K., and Freeman, William. Contextual models for object detection using boosted random fields. In *Proceedings of NIPS'04* (2004).
- [153] Turk, M.A., and Pentland, A.P. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience* (1991), vol. 3, pp. 71–86.
- [154] Tuytelaars, T., and van Gool, L. Content-based image retrieval based on local affinity invariant regions. In *In Proc. VISUAL. 1999* (1999).
- [155] Veltkamp, Remco, and Tanase, Mirela. Content-based image retrieval systems: A survey. *Technical Report UU-CS-2000-34* (October 28 2002).

- [156] Vinciarelli, A. A survey on off-line cursive word recognition. *Pattern Recognition* 35 (2002), 1433–1446.
- [157] Vinciarelli, A., Bengio, S., and Bunke, H. Offline recognition of large vocabulary cursive handwritten text. In *Proc. of ICDAR'03* (2003), pp. 1101–1105.
- [158] Vinciarelli, A., Bengio, S., and Bunke, H. Offline recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE Trans. Pattern Anal. Mach. Intelligence* 26, 6 (2004), 709–720.
- [159] Vinciarelli, A., and Luettin, J. Off-line cursive script recognition based on continuous density hmm. In *In Proc. of the 7th Int'l Workshop on Frontiers in Handwriting Recognition* (Amsterdam, The Netherlands, September 11-13 2000), pp. 493–498.
- [160] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR'01* (2001), pp. 511–518.
- [161] Wang, J., Wiederhold, G., Firschein, O., and Wei, S. Content-based image indexing and searching using daubechies wavelets. In *Int. J. Digital Libraries* 1, 4, 311–328. (1998).
- [162] Winn, J., Criminisi, A., and Minka, T. Object categorization by learned universal visual dictionary. In *Proc. ICCV* (2005), pp. 1800–1807.
- [163] Xie, L., Kennedy, L., Chang, S.-F., Divakaran, A., Sun, H., and Lin, C.-Y. Discovering meaningful multimedia patterns with audio-visual concepts and associated text. In *Proc. ICPR* (2004), pp. 2383–2386.
- [164] Yang, J., and Hauptmann, A. Annotating news video with locations. In *In CIVR06* (2006), pp. 153–162.
- [165] Yavlinsky, A., Schofield, E., and Rger, S. Automated image annotation using global features and robust nonparametric density estimation. In *Proc. of CIVR05* (2005), pp. 507–517.
- [166] Yu, J., and Tian, Q. Learning image manifolds by semantic subspace projection. In *In ACM Multimedia 06* (2006), pp. 297–306.
- [167] Yuan, J., Li, J., and Zhang, B. Learning concepts from large scale imbalanced data sets using support cluster machines. In *In ACM Multimedia 06* (2006), pp. 441–450.
- [168] Zhai, C., and Lafferty., J. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems* 22(2) (April 2004), 179–214.
- [169] Zhao, Ming, Teo, YongWei, Liu, Siliang, Chua, Tat-Seng, and Jain, Ramesh. Automatic person annotation of family photo album. In *In CIVR 2006* (2006), pp. 163–172.

- [170] Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P.J. Face recognition: A literature survey. *ACM Computing Surveys* (2003), 399–458.
- [171] Zhou, X., Chen, L., Ye, J., Zhang, Q., and Shi, B. Automatic image semantic annotation based on image-keyword document model. In *In CIVR05* (2005), pp. 184–193.
- [172] Zhu, S.-C., and Yuille, A. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 9 (1996), 884–900.