University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2011

Image annotation and retrieval based on multi-modal feature clustering and similarity propagation.

Mohamed Maher Ben Ismail 1979-
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Recommended Citation

Ismail, Mohamed Maher Ben 1979-, "Image annotation and retrieval based on multi-modal feature clustering and similarity propagation." (2011). *Electronic Theses and Dissertations.* Paper 662.
https://doi.org/10.18297/etd/662

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

IMAGE ANNOTATION AND RETRIEVAL BASED ON MULTI-MODAL FEATURE CLUSTERING AND SIMILARITY PROPAGATION

By

Mohamed Maher Ben Ismail
B.S., EE, National school of Engineering of Tunisia, 2002

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2011

Copyright 2011 by Mohamed Maher ben Ismail

All rights reserved

IMAGE ANNOTATION AND RETRIEVAL BASED ON MULTI-MODAL FEATURE CLUSTERING AND SIMILARITY PROPAGATION

By

Mohamed Maher Ben Ismail

A Dissertation Approved On

4/12/11

Date

by the Following Dissertation Committee:

Dissertation Director

Dr. Ayman El-Baz

Dr. Ming Ouyang

Dr. Dar-jen chang

Dr. Roman V. Yampolskiy

ACKNOWLEDGEMENTS

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family. I would like to express my deepest gratitude to my advisor, Dr. Hichem Frigui, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. I would like to thank the members of my committee, Dr. Ayman El-Baz, Dr. Ming Ouyang, Dr. Dar-jen chang, and Dr. Roman V. Yampolskiy, for their continual guidance and support. I would like to thank Ouiem Bchir, who as a good wife, was always willing to help and give her best suggestions. Many thanks to all researchers in the Multimedia Research Lab for their help. I would like to express my gratitude to my parents for their love, guidance, and understanding.

ABSTRACT

IMAGE ANNOTATION AND RETRIEVAL BASED ON MULTI-MODAL FEATURE CLUSTERING AND SIMILARITY PROPAGATION

Mohamed Maher Ben Ismail

May 13, 2011

The performance of content-based image retrieval systems has proved to be inherently constrained by the used lowlevel features, and cannot give satisfactory results when the user's high level concepts cannot be expressed by low level features. In an attempt to bridge this semantic gap, recent approaches started integrating both low level-visual features and high-level textual keywords. Unfortunately, manual image annotation is a tedious process and may not be possible for large image databases.

In this thesis we propose a system for image retrieval that has three mains components. The first component of our system consists of a novel possibilistic clustering and feature weighting algorithm based on robust modeling of the Generalized Dirichlet (GD) finite mixture. Robust estimation of the mixture model parameters is achieved by incorporating two complementary types of membership degrees. The first one is a posterior probability that indicates the degree to which a point fits the estimated distribution.

The second membership represents the degree of "typicality" and is used to indentify and discard noise points. Robustness to noisy and irrelevant features is achieved by transforming the data to make the features independent and follow Beta distribution, and learning optimal relevance weight for each feature subset within each cluster. We extend our algorithm to find the optimal number of clusters in an unsupervised and efficient way by exploiting some properties of the possibilistic membership function. We also outline a semi-supervised version of the proposed algorithm.

In the second component of our system consists of a novel approach to unsupervised image annotation. Our approach is based on : (*i*) the proposed semi-supervised possibilistic clustering; (*ii*) a greedy selection and joining algorithm (GSJ); (*iii*) Bayes rule; and (*iv*) a probabilistic model that is based on possibilistic memebership degrees to annotate an image.

The third component of the proposed system consists of an image retrieval framework based on multi-modal similarity propagation. The proposed framework is designed to deal with two data modalities: low-level visual features and high-level textual keywords generated by our proposed image annotation algorithm. The multi-modal similarity propagation system exploits the mutual reinforcement of relational data and results in a non-linear combination of the different modalities. Specifically, It is used to learn the semantic similarities between images by leveraging the relationships between features from the different modalities.

The proposed image annotation and retrieval approaches are implemented and tested with a standard benchmark dataset. We show the effectiveness of our clustering algorithm to handle high dimensional and noisy data. We compare our proposed image annotation approach to three state-of-the-art methods and demonstrate the effectiveness of the proposed image retrieval system.

TABLE OF CONTENTS

LIST OF TABLES **xii**

LIST OF FIGURES **xiv**

1 INTRODUCTION **1**

2 RELATED WORK **6**

2.1 Unsupervised Learning Algorithms 10

2.1.1 The Expectation Maximization (EM) Algorithm . . 10

2.1.2 The K-means Algorithm 13

2.1.3 The Fuzzy C-means (FCM) Algorithm 14

2.1.4 The Possibilistic C-means (PCM) Algorithm 16

2.1.5 The Competitive Agglomeration (CA) Algorithm . . 18

2.1.6 Simultaneous Clustering and Attribute Discrimination 19

2.1.7 Dirichlet Mixture Models 22

2.1.8 Generalized Dirichlet Mixture Models 26

2.2 Semi-supervised Clustering 29

2.2.1 Semi-supervised Mixture modeling 30

2.2.2 The Semi-supervised K-means Algorithm 30

2.2.3 The Semi-supervised Simultaneous Clustering and Attribute Discrimination (sSCAD) algorithm 31

2.3 Unsupervised Image Annotation 34

2.3.1 Statistics-based Models 34

2.3.2 Vector Space-Based Approaches 40

2.3.3 Classification-Based Approaches 41

2.3.3.1 Thesaurus Based Image Annotation 42

2.4 Major Contributions and Relation to Existing Work 43

3 DATA CLUSTERING BASED ON GENERALIZED DIRICHLET MIXTURE MODELS 45

3.1 Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models 46

3.2 Robust Unsupervised Learning of Finite GD Mixture Models with Feature Discrimination 53

3.3 Robust Unsupervised Learning of Finite GD Mixture Models with Feature Subset Selection 58

3.4 Semi-supervised Possibilistic Clustering and Feature Subset Weighting based on Robust GD Mixture Modeling 63

3.5 Finding the Optimal Number of Clusters 67

3.6 Experimental results 70

4 IMAGE ANNOTATION BASED ON CONSTRAINED REGION CLUSTERING 76

4.1 Image Database Organization 79

4.1.1 Image Segmentation 79

4.1.2 Feature Extraction and Representation 80

4.1.3 Constraints Formulation 81

4.1.4 Semi-supervised Clustering and Cluster Correlation Estimation 84

4.2 Image Annotation 88

4.2.1 Image Regions Assignment 88

4.2.1.1 Minimum Distance Image Region Assignment: 88

4.2.1.2 Probabilistic Image Region Assignment: . . 89

4.2.2 Identifying Independent Subsets of Image Regions . . 90

4.2.3 Image Annotation using a Semi-naive Bayesian Approach 91

4.2.4 Image Annotation using Membership based Cross Media Relevance Model 95

4.3 Experimental Results 100

4.3.1 Image Segmentation 103

4.3.2 Feature Representation 103

4.3.2.1 RGB Color Histogram: 103

4.3.2.2 HSV Color Moments: 105

4.3.2.3 LUV Color Moments: 105

4.3.2.4 Edge Histogram: 105

4.3.2.5 Wavelet Texture Features: 105

4.3.2.6 Shape Feature: 106

4.3.3 Constraint Formulation 106

4.3.4 Image Region Clustering 108

4.3.4.1 Minimum-distance based Clustering 108

4.3.4.2 Probabilistic Clustering 109

4.3.5 Image Annotation 113

4.3.6 Empirical Comparison with State-of-the-art Methods 125

5 IMAGE RETRIEVAL BASED ON MULTI-MODAL SIMILARITY PROPAGATION 129

5.1 Inter-modality Similarity Propagation 132

5.1.1 Convergence of the Algorithm 135

5.2 Image Retrieval Using Iterative Similarity Propagation . . . 137

5.3 Experimental Results 140

5.3.1 Comparison with Standard CBIR 142

5.3.2 Comparison with Hybrid Method 144

6 CONCLUSIONS AND POTENTIAL FUTURE WORK 150

6.1 Conclusions 150

6.2 Potential Future Work 153

6.2.1 System Scalability 154

6.2.2 User Relevance Feedback 154

REFERENCES 155

CURRICULUM VITAE 169

LIST OF TABLES

3.1 Parameters learned by RULe_GDM_FS for two Beta distributed clusters 71

3.2 Parameters learned by RULe_GDM_FS for a 4-dimensional data containing irrelevant features 71

3.3 Comparaison of the parameters used to generate the data to the parameters estimated using the method in [113] and RULe_GDM_FS. 74

4.1 Relevancy between pairs of keywords annotating the images in Fig 4.4 83

4.2 Relevancy between pairs of keywords annotating the images in Figure 4.5 84

4.3 List of words used to label the training images 101

4.4 Values of the constants used in the clustering process using sSCAD 108

4.5 Feature relevance weights for the 6 clusters displayed in Fig. 4.12 109

4.6 Feature relevance weights for the 6 clusters displayed in Fig. 4.13 111

4.7 Average accuracy of the proposed images annotation approaches when 1, 3, 5, and 7 words are used to annotate each image 120

4.8 Average per-image precision of the proposed image annotation methods when 1, 3, and 5 words are used to annotate each image 121

4.9 Average per-image recall of the proposed image annotation methods when 1, 3, and 5 words are used to annotate each image 122

4.10 Average accuracy of 5 image annotation methods 127

5.1 Optimal values of the similarity propagation parameters . . 142

LIST OF FIGURES

2.1 Overview of a typical CBIR system 7

2.2 Overview of a typical automatic image annotation system . 9

3.1 Finding the optimal number of clusters. (a) data set containing two Beta distributions, (b)-(f) Possibilistic membership of every point in the feature space in the 5 identified clusters. 69

3.2 Clustering 2 Beta distributions corrupted with uniform noise. (a) data set, (b) partition obtained with the method in [113], (c) partition obtained using Gaussian mixture model as described in [85], and (d) partition obtained using RULe_GDM_FS. Identified noise points are displayed with a '+' sign 73

3.3 Comparison of the accuracy of the data clustered with the method in [113] and RULe_GDM_FS when the dimensionality of the feature space is fixed to 30 and the noise rate is varied from 10 to 50 % 74

3.4 Comparison of the accuracy of the data clustered with the method in [113] and RULe_GDM_FS when the noise rate is fixed at 30% and the dimensionality of the feature space is varied from 2 to 40. 75

4.1 Overview of the proposed image annotation system 77

4.2 Examples of globally annotated images 79

4.3 Visual feature representation 80

4.4 Example of two correlated images. The first image is annotated by keywords “Plane” and “Sky”, while the second image is annotated by keywords “Sky” and “Bird”. 83

4.5 Example of two uncorrelated images. The first image is annotated by keywords “Beach”, “Sky”, “Sand” and “Tree” while the second image is annotated by keywords “car”, “road ”and “grass”. 84

4.6 Block diagram of the proposed approach to estimate correlation between clusters of image regions 87

4.7 Block diagram of the proposed image annotation approach based on Semi-naive Bayesian Model 94

4.8 Block diagram of the proposed image annotation approach using Membership based Cross Media Relevance Model . . . 99

4.9 frequency of the keywords used to label the set of training images 102

4.10 Example of images from the training set segmented using the CA clustering algorithm 104

4.11 Constraint Formulation Example 107

4.12 Representative regions of six sample clusters obtained by sSCAD 110

4.13 Representative regions of six sample clusters obtained by sRULe_GDM_FSS 112

4.14 Effect of the parameter t used in the GSJ on the annotation results using (a) sSCAD algorithm, and (b) sRULe_GDM_FSS algorithm. 115

4.15 Word accuracy obtained using semi-naive Bayesian Model with (a) sSCAD algorithm, (b) sRULe_GDM_FSS algorithm. 118

4.16 Word accuracy obtained using membership based CMRM with (a) sSCAD algorithm, (b) sRULe_GDM_FSS algorithm. 119

4.17 Image Annotation Samples (1) 123

4.18 Image Annotation Samples (2) 124

4.19 Comparison of the Precision vs. Recall curves of the two proposed sRULe_GDM_FSS based methods and three other existing methods 126

4.20 Comparison of the accuracy of the most frequent words using 5 different annotation methods 127

4.21 Word average accuracy vs. number of labeling keywords . . 128

5.1 Block Diagram of the Proposed Image Retrieval System . . 130

5.2 Illustration of the Multi-modal similarity propagation. Images A and B have similar visual features. Images B and C have similar textual features. Images B would be used as a bridge to enhance the similarity between images A and C. . 131

5.3 Illustration of the Similarity propagation process 133

5.4 Block diagram of the proposed image retrieval using multi-modal similarity propagation 139

5.5 Illustration of the Multi-modal Similarity Propagation. (a) Similarity before propagation, and (b) Similarity after propagation. 141

5.6 Precision Vs Number of Iterations when 20 images are retrieved. 143

5.7 Recall Vs Number of Iterations when 20 images are retrieved. 144

5.8 Comparison of the precision values for the proposed system and the baseline system versus the number of retrieved images. 145

5.9 Comparison of the recall values for the proposed system and the baseline system versus the number of retrieved images. . 146

5.10 Comparison of the precision values for the proposed system and the baseline system versus recall values 147

5.11 Top 25 representative from 4 typical clusters generated by (a) the proposed method, (b) the baseline method. 148

CHAPTER 1

INTRODUCTION

The widespread use of digital cameras, mobile phones with built-in cameras, and the storage of personal computers reaching a level of hundreds of gigabytes generated huge amounts of non-textual information, such as images stored in digital libraries. Meanwhile, photo sharing communities [1, 2] through the internet are becoming more and more popular. This exponential growth in image databases has demonstrated that simply increasing information quantity and its availability could be counterproductive if this is not coupled with automated tools for storing, searching, and retrieving.

Consequently, Content-Based Image Retrieval (CBIR) emerged as a new research field [3, 4]. CBIR involves the development of automated methods that are able to recognize the visual features of the images such as texture, color and shape, to characterize the salient information in the image, and to make use of this information in the indexing and retrieval processes. Building an efficient CBIR system requires tools from different disciplines. During the past few years, several CBIR systems have been proposed [22, 23] and research has focused on various topics such as system design [5], feature extraction [58], high dimensional indexing structures [6], similarity measures [7], perception analysis [8], semantic analysis [11], relevance feedback [21], user interfaces and user studies [66].

Unfortunately, after almost two decades of research in this field, the performance of most CBIR systems has proved to be inherently constrained by the used low-level features, and cannot give satisfactory results when the user's high level concepts cannot be expressed by low level features. In an attempt to bridge this semantic gap [24] and make the retrieval systems more accurate and efficient, few approaches that integrate low level visual features and textual features, used as caption to annotate images, have been proposed [5, 6, 7]. Unfortunately, manual image annotation is subjective and labor intensive since image databases can be very large. Moreover, region labeling may be needed, which makes the process more tedious. To address this issue, few algorithms that can annotate images/regions in an unsupervised (or semi-supervised) manner have been proposed recently.

Learning image semantics can be posed as either a supervised or unsupervised learning problem. The earliest efforts in this area were directed towards the reliable extraction of simple semantics, e.g., differentiating indoor from outdoor scenes [8], and cities from landscapes [9]. These efforts posed the problem of semantic extraction as one of supervised learning. That is, a set of training images with and without the concept of interest are collected and a binary classifier is trained to detect that concept. The classifier is then applied to each image in the database to annotate it with respect to the presence or absence of the concept.

Recently, there has been an effort to solve the annotation problem in greater generality by resorting to unsupervised learning. In fact, researchers turned to machine learning algorithms to build automatic annotation systems [10, 12, 13, 14]. Some image annotation approaches [15, 16] treat the problem in two independent stages. First categorizing the images, and then associating labels to them using the top ranked categories. Others, rely on the basic idea that the visual features corresponding to the same keyword are coherent. These methods rely on image segmentation and identifying homogeneous image regions which share the same semantics. An example of this approach is the method proposed by Duygulu et al. [17] which treats the problem as a translation of image regions to words. Another approach, proposed by Mori et al. [18], uses co-occurrence statistics of fixed image grids and words to model the associations. More recently, constrained clustering followed by semi-naive Bayesian model [19] and unsupervised clustering and feature discrimination (SCAD) [20] have been adapted to image annotation.

Most of the existing approaches use clustering algorithms to group image regions into prototypical region clusters that summarize the training data and can be used as the basis for annotating new test images. However, the clustering problem in this application is not trivial as it involves high dimensional and possibly multi-modal features. One possible approach that proved to be effective to cluster high dimensional data is to perform clustering and feature discrimination simultaneously [63, 64, 65]. However, learning using clustering and feature discrimination algorithm, like other unsupervised learning methods, may lead to sub-optimal solutions depending on the complexity of the data. To overcome this potential drawback, partial supervision could be used to "guide" the clustering process.

Most of the existing image database categorization methods assume that the data can be modelled by a mixture of Gaussian distributions. However, this assumption rarely holds in a very high-dimensional space and can affect the performance of subsequent annotation steps. Another common drawback associated with most existing image annotation methods is that they assume that region clusters are independent. For instance, many images may include planes in the sky, or animals on grass. Thus, one could not assume that the "planes" and "sky" regions are independent. This independency assumption could lead to inaccurate image annotation and eventually to the retrieval of irrelevant images.

In this thesis, we propose an efficient and effective approach that addresses the above issues. Our approach consists of three main contributions. First, we propose a possibilistic approach to model image regions using a mixture of Generalized Dirichlet (GD) [75, 76] distributions. This approach associates two types of memberships with each image region. The first one is the posterior probability and indicates how well a sample fits each estimated distribution. The second membership represents the degree of typicality and is used to identify noise regions and outliers. We extend this approach to learn relevance weights for each feature subset within each cluster. We also extend the algorithm to find the optimal number of clusters in an unsupervised and efficient way by exploiting some properties of the possibilistic membership functions. We also propose a semi-supervised version of our algorithm that uses partial supervision information in the form of a set of constraints to guide the clustering process. This proposed clustering algorithm are used to categorize image regions into categories of regions that share common attributes. Membership values, assigned by the clustering algorithm to each region in each cluster, are explored and used to estimate the degree of dependency among the region clusters.

The second component of this thesis consists of the development of a semi-naive Bayesian classifier to automatically annotate unlabeled images. This part is accomplished through two main steps. First, an unannotated image is segmented into homogeneous regions. Then, a greedy selection and joining (GSJ) algorithm is used to decompose the set of region clusters present in this unannotated image into independent subsets. Then, the posterior probability of a concept given a set of independent region cluster subsets is computed and used to assign concept labels to the image regions.

The third contribution of this thesis, consists of designing and implementing a complete CBIR system that uses an iterative similarity propagation approach to exploit mutual reinforcement between images and their annotations.

The organization of the rest of this thesis is as follows: Chapter two gives a literature review of related concepts including unsupervised and semi-supervised clustering, and image annotation techniques. In chapter three, we outline the proposed clustering algorithms. In chapter four, we outline the image annotation algorithms based on image region clustering. We also present an empirical comparison of the proposed methods with three state-of-the-art image annotation techniques. Then, chapter five describes the proposed image retrieval approach based on multi-modal similarity propagation, and its experimental results. Finally, chapter six outlines the conclusions and potential future work.

CHAPTER 2

RELATED WORK

Image retrieval has been an active research area since the 1970's [21]. Researchers from the database management and computer vision communities have proposed two different directions for image retrieval. The first one is text-based and the other one is based on the visual content of the image. Text-based image retrieval requires the images to be annotated with keywords prior to retrieval. With the significant advances of database management and textual information retrieval, this retrieval mode has achieved some success. However, two major difficulties have limited the practicality of this approach when large number of images are involved. The first one is simply the vast amount of tedious labor needed to manually annotate all images in the database. The second one is due to the subjectivity of the annotators; different users may perceive images in very different ways, resulting in different labels.

To overcome the above limitations, Content-Based Image Retrieval (CBIR) emerged as a new technique and started to gain more and more attention. CBIR retrieves images based on their visual content, such as color and

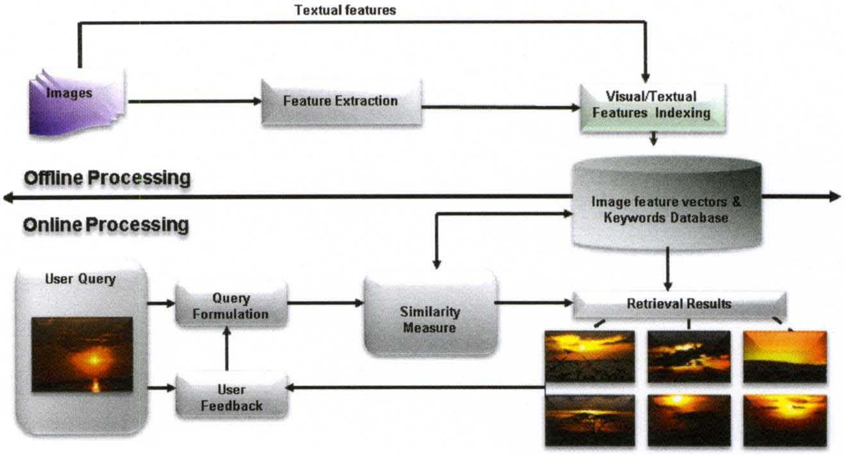


Figure 2.1: Overview of a typical CBIR system

texture, rather than keywords.

The standard CBIR approach is illustrated in figure 2.1. This approach can be conceptually separated in two main components: One is offline and consists of preprocessing, extracting features, and indexing the image database. The second one is online and consists of the user interaction with the system to query and retrieve images.

In the off-line part of the system, visual and textual features (if available) are extracted from the entire image collection. Visual features could be global or local if each image is segmented into homogeneous regions [23]. Textual feature, if available, are encoded into keywords and typically linked to the corresponding images by inverted tables.

The retrieval part of the CBIR system typically starts with a keyword and/or an example image through a user-interface. If the query consists of a set of keywords, the request is then sent to an inverted keyword index. In response, the system retrieves matching images, ranked by a similarity measure with respect to the textual features. In case of query by an example image, a pre-processing step is needed to map the image into a feature vector that describes its visual content. Then, using a similarity measure, the system retrieves images that have similar visual features. Based on the relevancy of the retrieved images and the level of user satisfaction, the user can provide a relevance feedback. The system uses this information to improve the precision in subsequent iterations.

Recently, to take advantages of the text based and content based retrieval modes and overcome their limitations, few approaches that integrate both features have been proposed [5, 6, 7]. Unfortunately, manual image annotation is subjective and labor intensive. Moreover, region labeling may be needed, which makes the process more tedious. Thus, automatic image annotation techniques have attracted a lot of interest in recent years [13, 15, 16, 17]. The aim of automatic annotation techniques is to attach textual labels to un-annotated images in a completly unsupervised manner. These labels could be used as additional descriptors of the content of the image or of particular objects within the image.

Typically, automatic image annotation is based on some machine learning techniques that can learn the correspondence between visual features and the semantics of images. That is, image annotation systems can recognize or classify visual features into some pre-defined classes [25].

Figure 2.2 shows the general architecture of a typical image annotation system. This system uses a set of labeled images for training. First, each training image is segmented into regions and local features are extracted and used to describe each region. There are two main segmentation strategies; The first one partitions the image into a set of fixed sized blocks or grid [18, 27]. The second one partitions the image into a number of homogeneous regions that share common features [2, 3, 4, 5]. Ideally, each region

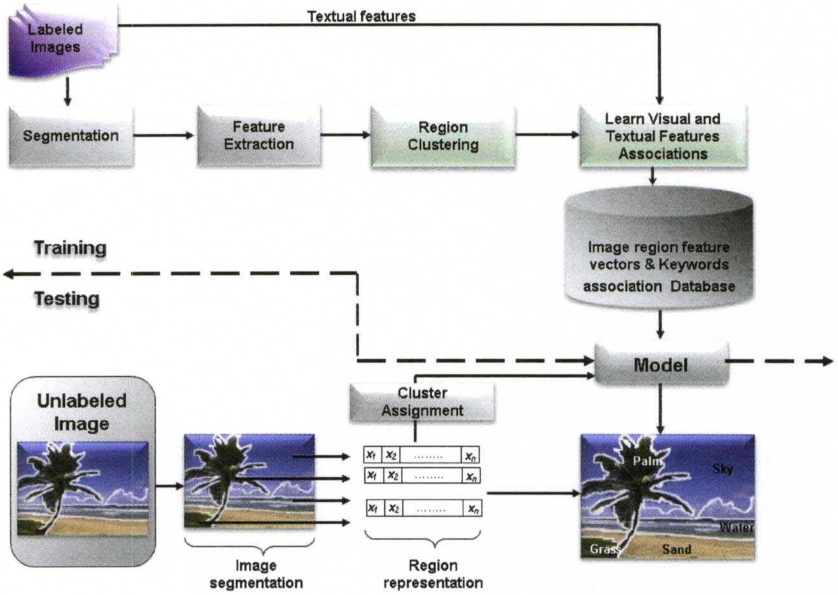


Figure 2.2: Overview of a typical automatic image annotation system

correspond to a different object in the image. After segmentation, each segmented block or region is represented by a feature vector that describe its visual content.

After segmenting all training images and extracting visual features from their regions, a machine learning algorithm is used to learn associations or joint probability distributions between these features and the keywords used to annotate the images.

The testing part of the system takes, as input, an un-annotated image, segments it into homogeneous regions, extracts and encodes the visual content of each region by feature vectors. Then, it uses the learned associations or joint probability distributions to infer the set of keywords that best describe the visual features. These keywords are then used to annotate the image.

In the rest of this chapter, we review the most common learning algorithms used in CBIR systems for image segmentation, region clustering, and association rule mining of visual and textual feature.

2.1 Unsupervised Learning Algorithms

To handle the huge amounts of data available in image data sets, most image annotation systems use clustering algorithms. Clustering consists of partioning the data into homogeneous subsets and summarizing them by few representative samples. There are various clustering approaches that could be used as a component of either CBIR or automatic image annotation systems. Few of these algorithms are outlined in the following subsections.

In the following, let $\mathbf{X} = \{\mathbf{X}_i \in \mathbb{R}^D | i = 1, ..., N\}$ be a set of N feature vectors in a D-dimensional feature space. Let $B = (\beta_1, ..., \beta_M)$ represent a M-tuple of prototypes each of which characterizes one of the M clusters. Each β_j consists of a set of parameters.

2.1.1 The Expectation Maximization (EM) Algorithm

The Expectation Maximization (EM) algorithm [30] is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in case of missing or hidden data. In ML estimation, the goal is to estimate the model parameters for which the observed data are most likely. Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step.

As before, we assume that data $\mathbf{X} = \{\mathbf{X_1}, ..., \mathbf{X_N}\}$ is observed and is generated by some distribution $p(\mathbf{X}/\theta)$. We call $\mathbf{X}$ the incomplete data. We assume that a complete data set exists $\mathbf{Z}$ =$(\mathbf{X}, \mathbf{Y})$ and also assume (or specify) a joint density function:

$$p(\mathbf{Z}_i/\theta) = p(\mathbf{X_i}, /\theta) = p(\mathbf{Y_i}/\mathbf{X_i}, \theta)p(\mathbf{X_i}/\theta)$$

This new likelihood function, $\mathcal{L}(\theta/\mathbf{Z}) = \mathcal{L}(\theta/\mathbf{X}, \mathbf{Y}) = p(\mathbf{X}, \mathbf{Y}/\theta)$, is called the complete-data likelihood.

In the expectation, or E-step, the EM algorithm first finds the expected value of the complete-data log-likelihood $log(\mathbf{X}, \mathbf{Y}/\theta)$ with respect to the unknown data $\mathbf{Y}$ given the observed data $\mathbf{X}$ and the current parameter estimates. That is,

$$Q\left(\theta, \theta^{(i-1)}\right) = E\left[log\left(\mathbf{X},\mathbf{Y}/\theta\right)/\mathbf{X}, \theta^{(i-1)}\right] \tag{2.1}$$

Where $\theta^{(i-1)}$ are the current parameters estimates that are used to evaluate the expectation and θ are the new parameters that are optimized to increase Q.

The second step (M-step) of the EM algorithm is to maximize the expectation computed in the E-step. That is,

$$\theta^{(i)} = \underset{\theta}{argmax} Q\left(\theta, \theta^{(i-1)}\right). \tag{2.2}$$

For instance, for mixture of Gaussian components [31], we assume that $\{\mathbf{Y_i}, 0 \leq i \leq N\}$ are samples drawn from gaussians $\mathbf{X_1}, ..., \mathbf{X_N}$. That is, we assume that $\mathbf{Y_i} \in [1...M]$, where $\mathbf{Y_i} = k$ if the i^{th} sample was generated by the k^{th} mixture component. If the values of $\mathbf{Y}$ are known, the likelihood becomes:

$$log\left(\mathcal{L}(\theta/\mathbf{X},\mathbf{Y})\right) = log\left(p(\mathbf{X},\mathbf{Y}/\theta)\right) = \sum_{i=1}^{N} log\left(P(\mathbf{X}_\mathbf{i}/\mathbf{Y}_\mathbf{i})P(\mathbf{Y})\right)$$

The model to be estimated is then the parameters of the M Gaussian components, that is,

$$\theta = \{c_1, ..., c_M,\ \Sigma_1, ..., \Sigma_M,\ p_1, ..., p_M\}. \tag{2.3}$$

In (2.3), $p_1, ..., p_M$ are the mixture probabilities.

Using the mixture of Gaussian representation, the E-step reduces to computing the conditional probability

$$P(\mathbf{X}_\mathbf{i}|\mathbf{Y}_\mathbf{i},\ \theta_j) = \frac{p(\mathbf{Y}_\mathbf{i}|\mathbf{X}_\mathbf{i}, c_j, \Sigma_j)P(\mathbf{X}_\mathbf{i}|c_j, \Sigma_j)}{\sum_{k=1}^{N} p(\mathbf{X}_\mathbf{k},\ \mathbf{Y}_\mathbf{i}|c_j, \Sigma_j)}. \tag{2.4}$$

and the M-step maximizes the expected log-likelihood

$$Q\left(\theta\right) = \sum_{i=1}^{N}\sum_{j=1}^{M} P(\mathbf{X}_\mathbf{i}|c_j, \Sigma_j)\left(log(p_j) + log\left(p(\mathbf{X}_\mathbf{i}|c_j, \Sigma_j)\right)\right), \tag{2.5}$$

subject to $\sum_{j=1}^{M} P(\mathbf{X}_\mathbf{i}|c_j, \Sigma_j) = 1$.

This optimization leads to the following update equations for the centers and covariances matrices of the Gaussian components:

$$c_j = \frac{\sum_{i=1}^{N} P(\mathbf{X}_\mathbf{i}|\theta_j)\mathbf{X}_\mathbf{i}}{\sum_{i=1}^{N} P(\mathbf{X}_\mathbf{i}|\theta_j)}. \tag{2.6}$$

and

$$\Sigma_j = \frac{\sum_{i=1}^{N} P(\mathbf{X}_\mathbf{i}|\theta_j)(\mathbf{X}_\mathbf{i} - c_j)(\mathbf{X}_\mathbf{i} - c_j)^T}{\sum_{i=1}^{N} P(\mathbf{X}_\mathbf{i}|\theta_j)}. \tag{2.7}$$

Algorithm 1 Expectation Maximization Algorithm

```
Begin
Initialize parameters $\theta^{[0]}$;
Repeat
    Compute $P(\mathbf{X}_i|\mathbf{Y}_i, \theta_j)$ using equation (2.4);
    Compute $Q(\theta)$ using equation (2.5);
    Compute $c_j$ using equation (2.6);
    Compute $\Sigma_j$ using equation (2.7);
Until (point of maximum is reached)
Return $\theta$
End
```

The convergence of the EM algorithm is assured since the algorithm is guaranteed to increase the likelihood in each iteration. The EM algorithm for mixture of Gaussians is summarized below:

2.1.2 The K-means Algorithm

The K-means algorithm [28] formulates the problem of partioning the N feature vectors into M clusters as minimization of the sum of squared error objective function:

$$J = \sum_{j=1}^{M} \sum_{\mathbf{X_i} \in C_j}^{N} ||\mathbf{X_i} - c_j||^2, \tag{2.8}$$

where $||\mathbf{X_i} - c_j||^2$ is the Euclidean distance between a feature point $\mathbf{X}_i$ and the center of the j^{th} cluster c_j.

Minimization of (2.8) with respect to the cluster centers yields:

$$c_j = \frac{\sum_{\mathbf{X_i} \in C_j}^{N} \mathbf{X_i}}{N}. \tag{2.9}$$

Initially, the data points are assigned randomly to clusters. Then, the K-means algorithm iteratively alternates between computing the cluster

Algorithm 2 K-means algorithm

```
Begin
Initialize cluster centers $c_1 ... c_M$.
Repeat
    Assign each point $\mathbf{X_i}$ to the closest cluster $\beta_j$
    For each $\beta_j$, update its center using equation (2.9).
Until (The centroids do not change)
Return $c_1 ... c_M$
End
```

centers and assigning each point to the closest cluster based on its distance to the corresponding center.

The K-means algorithm is summarized below:

2.1.3 The Fuzzy C-means (FCM) Algorithm

The Fuzzy C-Means (FCM) algorithm [67] is an extension of the K-means algorithm that distinguishes between objects strongly associated with a particular cluster from those that have only a marginal association with multiple clusters. The FCM algorithm attempts to partition the N feature vectors into a collection of M fuzzy clusters. It formulates the problem as a minimization of the following objective function

$$J = \sum_{j=1}^{M} \sum_{i=1}^{N} (u_{ji})^m d^2(\mathbf{X_i}, \beta_j) \tag{2.10}$$

where $d^2(\mathbf{X_i}, \beta_j)$ represents the distance from feature vector $\mathbf{X_i}$ to cluster β_j. In (2.10), u_{ji} represents the fuzzy membership of $\mathbf{X_i}$ in cluster β_j and satisfies the following constraints:

$$\begin{cases} u_{ji} \in [0,\, 1], & \forall j \\ 0 < \sum_{i=1}^{N} u_{ji} < N & \forall i,\, j \\ \sum_{j=1}^{M} u_{ji} = 1 & \forall i \end{cases} \tag{2.11}$$

In (2.10), $m \in (1,\, \infty)$ is a weighting exponent. Minimization of (2.10) with respect to $U = [u_{ji}]$, subject to the constraints in (2.11), gives [67]

$$\begin{cases} u_{ji} = \frac{1}{\sum_{k=1}^{M} \left(\frac{d^2(\mathbf{X_i}, \beta_j)}{d^2(\mathbf{X_i}, \beta_k)}\right)^{\frac{1}{m-1}}}, & if\, I_i = 0 \\ u_{ji} = 0 \;\; if\; j \notin I_i & if\, I_i \neq 0 \\ \sum_{j \in I_i} u_{ji} = 1 \;\; if\; j \in I_i & if\, I_i \neq 0 \end{cases} \tag{2.12}$$

where $I_i = \{j \,|\, 1 \leq j \leq C,\, d^2(\mathbf{X_i}, \beta_j) = 0\}$.

If the Euclidean distance

$$d^2(\mathbf{X_i}, \beta_j) = ||\mathbf{X_i} - c_j||^2, \tag{2.13}$$

is used, the FCM will seek spherical clusters. In this case, the update equation for the centroids is obtained by fixing the membership values and minimizing (2.10) with respect to c_j. This minimization yields

$$c_j = \frac{\sum_{i=1}^{N} (u_{ji})^m \mathbf{X_i}}{\sum_{i=1}^{N} (u_{ji})^m}. \tag{2.14}$$

The FCM algorithm is summarized below:

Algorithm 3 FCM Algorithm

```
Begin
  Fix the maximum number of clusters $M$;
  Fix $m \in (1, \infty)$;
  Repeat
    Compute $d^2(\mathbf{X}_i, c_j)$, for $1 \leq j \leq M$ and $1 \leq i \leq N$ ;
    Update the partition matrix $U^{(k)}$ using equation (2.12);
    Update the centers using (2.14);
  Until ($||\triangle U|| < \varepsilon$)
End
```

2.1.4 The Possibilistic C-means (PCM) Algorithm

An alternative approach to make the FCM (2.1.3) robust to noise and outliers is to relax the constraint that the membership degree of a point in all clusters must sum to 1. This is achieved by changing the objective function in (2.10) to

$$J = \sum_{j=1}^{M} \sum_{i=1}^{N} (u_{ji})^m d^2(\mathbf{X}_\mathrm{i}, \beta_j) + \sum_{j=1}^{M} \eta_j \sum_{i=1}^{N} (1 - u_{ji}) \tag{2.15}$$

and the membership cinstraints in (2.11) to

$$\begin{cases} u_{ji} \in [0,\, 1], & \forall i, j, \\ 0 < \sum_{i=1}^{N} u_{ji} < N & \forall j, \end{cases} \tag{2.16}$$

In (2.15), η_j are suitable positive numbers that typically relate to the overall size and shape of the cluster [82]. The first term in (2.15) minimizes the sum of intra-cluster distances, whereas the second term forces the u_{ji} to be as large as possible, thus avoiding the trivial solution where all u_{ji} are zero.

Minimizing (2.15) with respect to $U = [u_{ji}]$, subject to the constraints in

Algorithm 4 PCM Algorithm

```
Begin
  Fix the maximum number of clusters $M$;
  Initialize $\eta_j$ and $m \in (1, \infty)$;
  Repeat
    Compute $d^2(\mathbf{X}_\mathbf{i}, c_j)$, for $1 \leq j \leq M$ and $1 \leq i \leq N$ ;
    Update the partition matrix $U^{(k)}$ using equation (2.17);
    Update the centers using (2.18);
    Update $\eta_j$ as suggested in [82];
  Until $(||\triangle U|| < \varepsilon)$
End
```

(2.16), gives [82]

$$u_{ji} = \frac{1}{1 + (\frac{d^2(\mathbf{X}_\mathbf{i}, \beta_j)}{\mu_j})^{\frac{1}{m-1}}}, \tag{2.17}$$

If the Euclidean distance is used, the PCM will seek spherical clusters. In this case, the update equation for the centroids is obtained by fixing the membership values and minimizing (2.10) with respect to c_j. This minimization yields

$$c_j = \frac{\sum_{i=1}^{N} (u_{ji})^m \mathbf{X}_\mathbf{i}}{\sum_{i=1}^{N} (u_{ji})^m}. \tag{2.18}$$

The PCM algorithm is summarized below:

The possibilistic C-means (PCM) algorithm [82] can identify noise points as those points with low possibilistic membership in all clusters.

More recently, few algorithms that combine features from the PCM and FCM algorithms have been proposed. These methods assign the two types of membership degrees to each point. Examples of these methods include the Robust Competitive Agglomeration (RCA) [83] and the Possibilistic-Fuzzy Clustering Model (PFCM) [84] algorithms.

2.1.5 The Competitive Agglomeration (CA) Algorithm

The Competitive Agglomeration (CA) algorithm [32] is an efficient clustering algorithm that has the advantage of automatically determining the optimal number of clusters M. It minimizes

$$J(B, U, X) = \sum_{j=1}^{M} \sum_{i=1}^{N} (u_{ji})^2 d^2(\mathbf{X}_\mathrm{i}, \beta_j) - \alpha \sum_{j=1}^{M} \left[\sum_{i=1}^{N} u_{ji} \right]^2 . \tag{2.19}$$

In (2.19), M is the initial number of clusters. It is larger than the expected number, and it is dynamically updated during the optimization process. Optimization of J with respect to U yields:

$$u_{ji} = u_{ji}^{FCM} + u_{ji}^{BIAS} , \tag{2.20}$$

where

$$u_{ji}^{FCM} = \frac{[1/d^2(\mathbf{X}_\mathrm{i}, \beta_j)]}{\sum_{k=1}^{M} [1/d^2(\mathbf{X}_\mathrm{i}, \beta_k)]} , \tag{2.21}$$

and

$$u_{ji}^{BIAS} = \frac{\alpha}{d^2(\mathbf{X}_\mathrm{i}, \beta_j)} \left(\sum_{l=1}^{N} u_{jl} - \frac{\sum_{k=1}^{M} [1/d^2(\mathbf{X}_\mathrm{i}, \beta_k)] \sum_{l=1}^{N} u_{kl}}{\sum_{k=1}^{M} [1/d^2(\mathbf{X}_\mathrm{i}, \beta_k)]} \right) . \tag{2.22}$$

The update equation for the centroids are obtained by optimizing (2.19) with respect to β_j. This optimization yields the same equation as the FCM (i.e eq (2.14)).

The choice of α in (2.19) reflects the importance of the second term relative to the first term. In [32], the authors recommend using

$$\alpha(k) = \eta_0 exp(-k/\tau) \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} (u_{ji})^2 d^2(\mathbf{X}_\mathrm{i}, \beta_j)}{\sum_{j=1}^{M} [\sum_{i=1}^{N} u_{ji}]^2} \tag{2.23}$$

Algorithm 5 CA Algorithm

```
Begin
  Initialize the maximum number of clusters $M = M_{max}$;
  Initialize iteration counter $k = 0$ and the fuzzy $M$ partition $U^{(0)}$;
  Compute initial cardinalities $N_j$ for $1 \leq j \leq M$ using $N_j = \sum_{l=1}^{N} u_{jl}$;
  Repeat
    Compute $d^2(\mathbf{X}_i, \beta_j)$, for $1 \leq j \leq M$ and $1 \leq i \leq N$ ;
    Update $\alpha(k)$ using equation (2.23);
    Update the partition matrix $U^{(k)}$ using equation (2.20);
    Compute the cardinality $N_j$ for $1 \leq j \leq M$ ;
    If ( $N_j < \varepsilon$) discard cluster $c_i$ ;
    Update the number of clusters $M$;
    Update the centers using (2.14);
    k++;
  Until (Prototype parameters stabilize)
End
```

where η_0 is the initial value, τ the time constant, and k is the iteration number. The CA algorithm is summarized below:

2.1.6 Simultaneous Clustering and Attribute Discrimination

The challenge of selecting the best subset of features or attributes constitutes an important part of the design of good learning algorithms for real world tasks. Irrelevant features can degrade the generalization performance of these algorithms significantly. This selection is even more critical and challenging in applications involving high dimensional data. This is because clusters tend to form in different subspaces of the original feature space.

Several techniques have been proposed for feature selection and weighting [33, 34, 35]. In particular, Frigui and Nasraoui [36, 37] proposed an algorithm that performs Simultaneous Clustering and attribute Discrimination

(SCAD). The SCAD algorithm is designed to search for the optimal clusters' prototypes and the optimal relevance weights for each feature within each cluster. However, for high dimensional data, learning a relevance weight for each feature may lead to overfitting. To avoid this case, a coarse approach to feature weighting called SCADc was proposed in [38]. SCADc is an extension of SCAD where instead of learning a weight for each feature, the set of features is divided into logical subsets, and a weight is learned for each feature subset.

In [38], the authors assume that the D features have beem partitioned into d subsets: FS^1, $FS^2, \ldots,$ FS^d and each subset, FS^s, includes d^s features. Let d_{ji}^s be the partial distance between $\mathbf{X_i}$ and cluster j using the s^{th} feature subset. Let $V = [v_{js}]$ be the relevance weight for FS^s with respect to cluster j. The total distance, D_{ji}, between $\mathbf{X_i}$ and cluster j is then computed by aggregating the partial distances and their weights, i.e.,

$$D_{ji}^2 = \sum_{s=1}^{d} v_{js} (d_{ji}^s)^2. \tag{2.24}$$

SCADc minimizes

$$J(B, U, V; \mathbf{X}) = \sum_{j=1}^{M} \sum_{i=1}^{N} u_{ji}^m \sum_{s=1}^{d} v_{js} \, (d_{ji}^s)^2 + \sum_{j=1}^{M} \delta_j \sum_{s=1}^{d} v_{js}^2 \,, \tag{2.25}$$

subject to the constraints in (2.11) and

$$v_{js} \in [0, 1] \, \forall j, \, s; \;\; and \;\; \sum_{s=1}^{d} v_{js} = 1, \, \forall j. \tag{2.26}$$

Optimization of J with respect to V yields

$$v_{js} = \frac{1}{d} + \frac{1}{2\delta_j} \sum_{i=1}^{N} (u_{ji})^m \left[D_{ji}^2/d - (d_{ji}^2)^2 \right]. \tag{2.27}$$

The first term in (2.27), $(1/d)$, is the default value if all d feature subsets are treated equally, and no discrimination is performed. the second term is a bias that can be either positive or negative. it is positive for compact feature sets where the partial distance is, on average, less than the total distance (normalized by the number of features). If a feature set is compact, compared to the other features, for most of the points that belong to a given cluster (high u_{ji}), then it is relevant for that cluster.

minimization of J with respect to U, subject to the constraints in (2.11), yields

$$u_{ji} = \frac{1}{\sum_{k=1}^{M} (D_{ji}^2/D_{ki}^2)^{\frac{1}{m-1}}} \tag{2.28}$$

Minimization of J with respect to the prototype parameters depends on the choice of d_{ji}^s. Since the partial distances are treated independent of each other (i.e., disjoint feature subsets), and since the second term in (2.25) does not depend on prototype parameters explicitly, the objective function in (2.25) can be decomposed into d independent problems:

$$J_s = \sum_{j=1}^{M} \sum_{i=1}^{N} u_{ji}^m v_{js} (d_{ji}^s)^2, \quad for\, s = 1, ..., d. \tag{2.29}$$

Each J_s could be optimized with respect to a different set of prototype parameters. For instance, if d_{ji}^s is the Euclidean distance, minimization of J_s would yield the following update equation for the centers of subset s,

$$c_j^s = \frac{\sum_{i=1}^{N} u_{ji}^m \mathbf{X_i^s}}{\sum_{i=1}^{N} u_{ji}^m}. \tag{2.30}$$

Algorithm 6 Coarse SCAD Algorithm

```
Begin
  Fix the maximum number of clusters $C = C_{max}$;
  Fix m, $m \in (1, \infty)$;
  Initialize the centers and the fuzzy M partition matrix U;
  Initialize the relevance weights to 1/d;
  Repeat
    Compute $(d^s_{ji})^2$, for $1 \leq j \leq M$ and $1 \leq i \leq N$ and $1 \leq s \leq d$ ;
    Update the relevance weights $v_{js}$ using equation (2.27);
    Compute $D^2_{ji}$ using equation (2.24);
    Update the partition matrix $U^{(k)}$ using equation (2.28);
    Update the centers using equation (2.30);
  Until ( centers stabilize)
End
```

SCADc is an iterative algorithm that starts with an initial partition and alternates between the update equations of u_{ji}, v_{js}, and c^s_j. It is summarized below:

2.1.7 Dirichlet Mixture Models

Another alternative approach to unsupervised or supervised learning is based on probabilistic modeling. The probabilistic approach assumes that data objects in different clusters are generated by different probability distributions. They can be generated from different types of density functions (e.g., multivariate Gaussian or t-distribution), or the same families, but with different parameters. If the distributions are known, finding the clusters is equivalent to estimating the parameters of the underlying models. The mixture solving approach [85] is a widely used partitional clustering technique based on probabilistic models. It assumes that samples in a cluster are drawn from one of several distributions (usually Gaussian) and attempts to estimate the parameters of the distributions. Despite all recent progress, probabilistic modeling remains a challenging research problem. In high dimensional space, Gaussian mixtures with diagonal covariance matrices have been used frequently. However, Gaussian functions cannot provide reasonable approximation for asymmetric distributions. The problem is more acute when the data are high dimensional and some features may be irrelevant and/or correlated.

Introduced as a good alternative, Dirichlet distribution is a multivariate generalization of the Beta distribution, which offers considerable flexibility and ease of use. In contrast with other distributions, the Dirichlet distribution permits multiple symmetric and asymmetric modes [95].

Let a set of N independent vectors $\mathbf{X} = (\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_N})$, and let the random vector $\mathbf{X_i} = (X_{i1}, X_{i2}, ..., X_{iD})$ follows a Dirichlet distribution [100, 101]. The joint density function is given by

$$p(X_{i1}, X_{i2}, ..., X_{iD}) = \frac{\Gamma(|\alpha|)}{\prod_{l=1}^{D+1} \Gamma(\alpha_l)} \prod_{l=1}^{D+1} X_{il}^{\alpha_l - 1} \tag{2.31}$$

where

$$\begin{aligned}
\sum_{l=1}^{D} X_{il} &< 1 \\
0 < X_{il} &< 1 \quad \forall l = 1..D \\
X_{D+1} &= 1 - \sum_{l=1}^{D} X_{il} \\
|\alpha| &= \sum_{l=1}^{D+1} \alpha_l \\
\alpha_l &> 0 \; \forall l = 1..D+1
\end{aligned}$$

This distribution is the multivariate extension of the two-parameter Beta distribution. The mean and the variance of the Dirichlet distribution are given by

$$E(X_{il}) = \frac{\alpha_l}{|\alpha|} \tag{2.32}$$

$$Var(X_{il}) = \frac{\alpha_l(|\alpha| - \alpha_l)}{|\alpha|^2(|\alpha| + 1)} \tag{2.33}$$

and the variance between X_{il} and X_{ik} is

$$Cov(X_{ik}, X_{il}) = \frac{-\alpha_k \alpha_l}{|\alpha|^2(|\alpha| + 1)}. \tag{2.34}$$

The Dirichlet mixture with M components is defined as

$$p(\mathbf{X}|\theta) = \sum_{j=1}^{M} P(j)\, p(\mathbf{X}|j, \theta_j), \tag{2.35}$$

where $P(j)\,(0 < P(j) < 1$ and $\sum_{j=1}^{M} P(j) = 1)$ are the mixing proportions and $p(\mathbf{X}|j, \theta_j)$ is the Dirichlet distribution. The symbol θ refers to the entire set of parameters to be estimated $\theta = (\alpha_1, ..., \alpha_M, P(1), ..., P(M))$, where α_j is the parameter vector for the j^{th} population. In the rest of this section, we use the notation $\theta_j = (\alpha_j)$ for $j = 1...M$.

The problem of estimating the parameters which determine a mixture has been the subject of diverse studies [102]. During the last two decades, the method of maximum likelihood (ML) [103] has become the most common approach to this problem. Of the variety of iterative methods which have been suggested as alternatives to optimize the parameters of a mixture, the one most widely used is expectation maximization (EM) (2.1.1). However, this algorithm suffers from the need to specify the number of components each time. In order to overcome this problem, criterion functions have been proposed, such as the Akaike information criterion (AIC) [104], minimum description length (MDL) [105], and Schwartz's Bayesian inference criterion (BIC) [106]. A maximum likelihood estimate associated with a sample of observations is a choice of parameters which maximizes the probability density function of the sample. Thus, with ML estimation, the problem of determining θ becomes

$$max_\theta p(\mathbf{X}|\theta) \tag{2.36}$$

with the constraints $\sum_{j=1}^{M} P(j) = 1$ and $P(j) > 0 \forall j \in [1, M]$. These constraints permit to take into consideration a priori probabilities $P(j)$. Using Lagrange multipliers, the following function is maximized

$$\Phi(\mathbf{X}, \theta, \Lambda) = log\left(p(\mathbf{X}|\theta)\right) + \Lambda\left(1 - \sum_{j=1}^{M} P(j)\right) + \mu \sum_{j=1}^{M} P(j) log(P(j)) \tag{2.37}$$

where Λ is the Lagrange multiplier. For convenience, we have replaced the function in (2.36) by the function $log\left(p(\mathbf{X}|\theta)\right)$. If we assume that we have N random vectors $\mathbf{X_i}$ which are independent, we can write

$$\begin{aligned} p(\mathbf{X}|\theta) &= \prod_{i=1}^{N} p(\mathbf{X_i}|\theta) & (2.38) \\ p(\mathbf{X_i}|\theta) &= \sum_{j=1}^{M} p(\mathbf{X_i}, j, \theta_j) P(j). & (2.39) \end{aligned}$$

Replacing (2.38) and (2.39), we obtain

$$\Phi(\mathbf{X}, \theta, \Lambda) = \sum_{i}^{N} log\left(\sum_{j=1}^{M} p(\mathbf{X_i}, j, \theta_j) P(j)\right) + \Lambda\left(1 - \sum_{j=1}^{M} P(j)\right) + \mu \sum_{j=1}^{M} P(j) log(P(j)). \tag{2.40}$$

The maximum-likelihood estimate of these distributions is not available in closed-form. In [108], the author proposed an iterative algorithm based on a fixed-point and Newton-Raphson iterations. The authors in [97], solved this optimization problem and estimated the parameters of this mixture using the maximum likelihood and Fisher scoring methods [107].

2.1.8 Generalized Dirichlet Mixture Models

Despite its flexibility, the Dirichlet distribution has a very restrictive negative covariance structure. In this section, we introduce the generalization of the Dirichlet distribution which has a more general covariance structure than the Dirichlet distribution. Let the random vector $\mathbf{X_i} = (X_{i1}, X_{i2}, ..., X_{iD})$ follows a Generalized Dirichlet distribution [96] as follow

$$p(X_{i1}, X_{i2}, ..., X_{iD}) \quad = \quad = \prod_{l=1}^{D} \frac{\Gamma(\alpha_l + \beta_l)}{\Gamma(\alpha_l)\Gamma(\beta_l)} Y_l^{\alpha_l - 1} \left(1 - \sum_{k=1}^{l} X_{ik}\right)^{\gamma_l}$$

where $\sum_{l=1}^{D} X_{il} < 1$; $0 < X_{il} < 1$, for $l = 1, ..., D$; $\gamma_l = \beta_l - \alpha_l - \beta_{l+1}$, for $l = 1, ..., D-1$; and $\gamma_D = \beta_D - 1$. Note that the Generalized Dirichlet distribution is reduced to a Dirichlet distribution when $\beta_l = \alpha_{l+1} + \beta_{l+1}$. The mean of the Generalized Dirichlet distribution satisfy the following conditions:

$$E(X_{il}) \quad = \quad \frac{\alpha_l}{\alpha_l + \beta_l} \tag{2.41}$$

and the covariance between X_{is} and X_{it} is

$$Cov(X_{is}, X_{it}) = E(X_{it}) \left(\frac{\alpha_i}{\alpha_i + \beta_i + 1} \prod_{k=1}^{s-1} \frac{\beta_{k+1}}{\alpha_k + \beta_k} + 1 - E(X_{is}) \right). \tag{2.42}$$

Numerous other properties of this distribution are given in [87].

The Generalized Dirichlet distribution has the advantage that by varying its parameters, it permits multiple modes and asymmetry and can, thus, approximate a wide variety of shapes. Besides, it has a more general covariance structure than the Dirichlet. This generalization has made Dirichlet distribution more practical and useful in Bayesian learning scenarios in general and finite mixture modeling in particular. For instance, in [88] the Generalized Dirichlet was used as the component distribution in finite mixtures to model continuous data. The Generalized Dirichlet was also used as a prior to the multinomial distribution, which is then integrated out to model count data [89]. In [76], the authors proposed using the Dirichlet distribution as a prior to perform multinomial and mixture model estimation. These models have proven to be effective in many applications such as language modeling, and content-based image summarization and retrieval [86].

Given a set of N independent vectors $\mathbf{X} = (\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_N})$. A Generalized Finite Dirichlet mixture with M components is defined as

$$p(\mathbf{X}|\theta) = \sum_{j=1}^{M} P(j) p(\mathbf{X}|\theta_j). \tag{2.43}$$

where $P(j)$ are the mixing probabilities and $p(\mathbf{X}|\theta_j)$ is the Generalized Dirichlet distribution.

Each $\theta_j = (\alpha_{j1}, \beta_{j1}, ..., \alpha_{jD}, \beta_{jD})$ is the set of parameters defining the th component, and θ is the complete set of parameters, $\theta = (\theta_1, ..., \theta_M, P(1), ..., P(M))$, needed to specify the mixture. Of course, being probabilities, $P(j)$ must satisfy

$$0 \quad < P(j) \leq 1 \qquad j = 1...M \tag{2.44}$$

$$\textstyle\sum_{j=1}^{M} P(j) = 1. \tag{2.45}$$

The log-likelihood becomes

$$L(\theta, \mathbf{X}) = log \prod_{i=1}^{N} p(\mathbf{X_i}|\theta) = \sum_{i=1}^{N} log \sum_{j=1}^{M} p(\mathbf{X_i}|\theta_j)P(j). \tag{2.46}$$

The problem of estimating the parameters of Generalized Dirichlet finite mixtures has been the subject of diverse studies. The most common approach is the Maximum likelihood (ML) [90]. This approach seeks the parameters that maximize the probability of generating all of the observed data. The maximum likelihood (ML) estimates

$$\hat{\theta}_{ML} = arg \max_{\theta} \{L(\theta, \mathbf{X})\} \tag{2.47}$$

The maximization defining the ML estimates is subject to the constraints in (2.44) and (2.45). However, the ML solution cannot be obtained analytically. Thus, iterative approaches, such as the expectation-maximization (EM) algorithm (2.1.1), have been proposed to approximate the ML estimates. The majority of the studies either consider a single distribution [91] or are restricted to the two-parameter Beta distribution [92]. In [76], the authors proposed an hybrid stochastic expectation maximization algorithm to estimate the parameters of the Generalized Dirichlet mixture. The algorithm was called stochastic because it contains a step in which the data elements are assigned randomly to components in order to avoid convergence to a saddle point. The adjective "hybrid" is justified by the introduction of a Newton–Raphson step. Moreover, this algorithm autonomously selects the number of components by the introduction of an agglomerative term.

In order to use the Generalized Dirichlet mixture model to get overlapping clustering, where a point can deterministically belong to multiple clusters, most of the existing methods choose a threshold value such that point

$\mathbf{X_i}$ belongs to the j^{th} partition if $p(z_i = j|\mathbf{X_i}, \theta) > \lambda$ where Z_i is the label of $\mathbf{X_i}$ such that $Z_i \in \{0, 1\}$, $\sum_{j=1}^{M} Z_i = 1$, and $Z_i = 1$ if $\mathbf{X_i}$ comes from the j^{th} component. This thresholding technique can enable $\mathbf{X_i}$ to belong to multiple clusters. However, this is not a natural generative model for overlapping clustering. In the mixture model, the underlying model assumption is that a point is generated from only one mixture component, and $p(z_i = j|\mathbf{X_i}, \theta) > \lambda$ simply gives the probability of $\mathbf{X_i}$ being generated from the j^{th} mixture component. Moreover, these methods do not perform well when the data is noisy. In fact, noise points and outliers can drastically affect the model parameters estimation.

2.2 Semi-supervised Clustering

Clustering is a hard optimization problem with many local minima. One possible approach to simplify this problem is to use partial supervision to guide the clustering process and narrow the space of possible solutions. This additional information is usually available under the form of hints [70], constraints [71], or labels [72]. Supervision in the form of constraints is more practical, because it is much easier to specify whether pairs of points should belong to the same cluster or to different clusters. In the following we provide an overview of the semi-supervised mixture modeling, the Semi-supervised K-means [29], and the Semi-supervised Simultaneous Clustering and Attribute Discrimination (sSCAD) [73] algorithms. These algorithms have been applied successfully to categorize large collections of images or image regions.

2.2.1 Semi-supervised Mixture modeling

Recently, researches on semi-supervised learning based on mixture models have been published. Wu and Huang [109] integrate multiple discriminant analysis (MDA) with EM framework so that learners are boosted by exploring discriminant features in a self-supervised fashion. Another approach dealing with labeled and unlabeled data for Gaussian mixture models [110] is to modify the mixture log-likelihood function as the combination of two terms: the one for unlabeled data and the other for labeled data. Recently, in [111], the authors presented a semi-supervised EM algorithm. The supervison information is integrated using concept learning with multiple users' relevance feedbacks.

These algorithms contribute to a general improvement of the learning performance, when few labelled samples are available, with respect to other well-known unsupervised algorithms. However, they assume that the data follow a Gaussian distribution. Moreover, they have not been used with high dimensional datasets, and assume that the data is noise free.

2.2.2 The Semi-supervised K-means Algorithm

The traditional K-means clustering algorithm has been modified to make use of instance-level constraints [29]. Two types of pairwise constraints have been considered. The first one is *Must-link* constraints and specifies that two data points must be assigned to the same cluster. The second type of constraints is *MustNot-link* and specifies that two data points must not be assigned to the same cluster.

Let ML be the set of *Must-link* pairs such as $(\mathbf{X}, \mathbf{Y}) \in ML$ implies that x

and y must be assigned to the same cluster. Similarly, we let NL be the set of *MustNot-link* pairs such as $(\mathbf{X}, \mathbf{Y}) \in NL$ means that $\mathbf{X}$ and $\mathbf{Y}$ must not be assigned to the same cluster. The constrained K-means algorithm minimizes

$$J_{ConstrK-means} = \sum_{j=1}^{C} \sum_{i=1}^{N} ||\mathbf{X}_i^{(j)} - \mathbf{c_j}||^2 + \alpha \Big[\sum_{(\mathbf{X}_m \times \mathbf{X}_n \in \mathbb{NL})} \sum_{j=1}^{C} C_{link}(\mathbf{X}_m, \mathbf{X}_n) \tag{2.48}$$

$$+ \sum_{(\mathbf{X}_m \times \mathbf{X}_n \in \mathbb{ML})} \sum_{j=1}^{C} \sum_{l=1, l \neq j}^{C} M_{link}(\mathbf{X}_m, \mathbf{X}_n) \Big]$$

where

$$C_{link}(\mathbf{X}_m, \mathbf{X}_n) = \begin{cases} 1 & if \, \{\mathbf{X}_m, \mathbf{X}_n\} \in cluster \, j \\ 0 & Otherwise \end{cases}$$

$$M_{link}(\mathbf{X}_m, \mathbf{X}_n) = \begin{cases} 1 & if \, \mathbf{X}_m, \in cluster \, j, \, and \, \mathbf{X}_n \in cluster \, l \\ 0 & Otherwise \end{cases}$$

The first term in (2.48) is the objective function of K-means (2.1.2). The second term consists of the cost of violating the pairwise *Must-link* and *MustNot-link* constraints. The value of α in (2.48) controls the importance of the supervision information compared to the sum of intra-cluster distances.

The constrained K-means algorithm is outlined below:

2.2.3 The Semi-supervised Simultaneous Clustering and Attribute Discrimination (sSCAD) algorithm

In [36], the authors proposed a semi-supervised version of SCADc (2.1.6). As in the constrained K-means, the supervision information consists of

Algorithm 7 Constrained K-means as EM algorithm

```
Begin
Initialize the cluster centers $\mathbf{c}_1 ... \mathbf{c}_M$.
Set the Must-link and MustNot-link constraints.
Repeat
    Assign each point $\mathbf{X}_i$ to the closest cluster $j$ given ML and NL constaints.
    Update the center of each cluster by averaging all points assigned to it.
Until (The centroids do not change)
Return $\mathbf{c}_1 ... \mathbf{c}_M$
End
```

pairs of points that should be assigned to the same cluster and pairs of points that should not be assigned to the same cluster.

The constrained sSCAD [73] algorithm minimizes

$$J = \sum_{j=1}^{M} \sum_{i=1}^{N} u_{ji}^m \sum_{s=1}^{d} (v_{ji})^2 (d_{ji}^s)^2 + \alpha \left[\sum_{(\mathbf{X}_i, \mathbf{X}_k \in ML)} \sum_{j=1}^{M} \sum_{l=1, l \neq j}^{M} u_{ji}^m u_{lk}^m + \sum_{(\mathbf{X}_i, \mathbf{X}_k \in \mathbb{NL})} \sum_{j=1}^{M} u_{ji}^m u_{jk}^m \right] \tag{2.49}$$

subject to (2.11) and (2.26). The first term is the objective function of SCADc (2.1.6) and is used to seek compact clusters and their partial feature relevance weights. The second term consists of the cost of violating the constraints. The value of α controls the importance of the supervision information.

Minimization of J with respect to v_{js} yields

$$v_{js} = \frac{1}{\sum_{t=1}^{d} (d_j^s / d_j^t)}, \tag{2.50}$$

where $d_j^s = \sum_{i=1}^{N} (u_{ji})^m (d_{ji}^s)^2$.

Algorithm 8 Semi-Supervised SCAD Algorithm

```
Begin
  Fix the number of clusters $M$;
  Fix the fuzzifier $m$, $m \in (1, \infty)$;
  Fix the set of Should-Link (ML) and ShouldNot-Link (NL) constraints;
  Initialize the centers;
  Initialize the relevance weights to 1/d;
  Repeat Compute $(d_{ji}^s)^2$, for $1 \leq j \leq M$ and $1 \leq i \leq N$ and $1 \leq s \leq d$ ;
    Update the relevance weights $v_{js}$ using equation (2.50);
    Compute $\tilde{D}_{ji}$ using equation (2.52);
    Update the partition matrix $U^{(k)}$ using equation (2.51);
    Update the centers using equation (2.30);
  Until (centers stabilize)
End
```

Minimization of J with respect to the memberships yields

$$u_{ji} = \frac{(1/\tilde{D}_{ji})^{1/(m-1)}}{\sum_{k=1}^{C}(1/\tilde{D}_{ki})^{1/(m-1)}}, \tag{2.51}$$

where

$$\tilde{D}_{ji} = m[D_{ji}^2 + \alpha(\sum_{(\mathbf{X}_i,\mathbf{X}_k \in ML)} \sum_{l=1,l\neq j} u_{lk}^m + \sum_{(\mathbf{X}_i,\mathbf{X}_k \in NL)} u_{jk}^m)] \tag{2.52}$$

In (2.52), D_{ji} can be viewed as the total cost when considering point $\mathbf{X}_i$ in cluster β_j. This cost depends on the distances of $\mathbf{X}_i$ to cluster β_j and the cost of the violated constraints caused by $\mathbf{X}_i$ and $\mathbf{X}_k$.

Since the second term in (2.49) does not depend on the prototype parameters explicitly, minimizing (2.49) with respect to prototypes yields the same update equations as the SCAD algorithm.

The Semi-Supervised SCAD algorithm is summarized below:

2.3 Unsupervised Image Annotation

Image annotation systems aim at automatically annotating an image with some controlled keywords. They have been proposed as a solution to reduce the semantic gap in CBIR. In these systems, machine learning techniques are used to build a model that maps the image low-level (visual) features to high-level concepts or semantics. After the annotation model is learned, an image is annotated by finding the most likely keywords, describing the high-level concepts, given the visual features of the image. In the following we outline the main techniques that have been used for this task.

2.3.1 Statistics-based Models

Co-occurence Model: The co-occurrence model proposed by Mori et al. [18] is one of the first attempts at image auto-annotation. First, they divide the images into a regular grid, and compute a feature vector of colour and texture for each block. Feature vectors extracted from blocks of a set of training images are then summarized by few clusters. Each cluster is represented by its centroid. Each tile on the grid inherits the whole set of labels from the original image. Then, the probability of a label w related to a cluster c is estimated by the co-occurrence of the label and the image tiles within the cluster using

$$p(w|c) = \frac{m_{c,w}}{\sum_w m_{c,w}}, \tag{2.53}$$

where $m_{c,w}$ is the number of times word w occurs with an image tile from cluster c.

For testing, given an un-annotated image, they divide it into rectangular grid and extract feature vectors as it was done in the training phase. Next, the closest cluster centroid to each tile is identified. Then, the probability of each label in each of the tiles of the test image is computed using

$$p(w|I) = \frac{1}{|I|} \sum_{t \in I} p(w|c_t)\,. \tag{2.54}$$

In (2.54), $p(w|I)$ represents the average probability of word w given image I, c_t is the closest cluster to the region/tile t extracted from image I, and $|I|$ is the number of tiles.

The labels w_i having the highest probabilities $p(w_i|I)$ are chosen as the keywords to labels the test image.

The co-occurence approach is limited because the average probability estimation can be affected by the noisy clusters obtained after categorizing the heterogeneous image tiles. Moreover, the fixed grid approach used to partition the images has its own limitations. For instance, a large number of blocks may result in an over-segmented regions. This may lead to additional computations and irrelevant labeling. On the other hand, a small number of blocks may result in non-homogeneous tiles.

Machine Translation Model Duygulu et al. [17] proposed a machine translation model for automatic image annotation. They argued that region based image annotation is more interesting because global annotation does not give information on which part of the image is related to which label. In their approach, they first use a segmentation algrithm to segment images into object-shaped regions. Then, feature quantization is applied to the feature vectors that are extracted from all the regions, to build a visual vocabulary called 'blobs'. A 'blob' is a representitive of a cluster of visually similar image regions. Finally, a mapping between blobs and keywords, supplied with the images, is learned using a method based on the EM algorithm (described in 2.1.1). A test image is annoated by choosing the most likely words for each of its regions.

The difficulties with this machine translation model arises from the unbalanced distribution of the word frequencies in the training dataset. Moreover, the co-occurence statistics can be affected significantly by the noise in quantizing the huge number of regions into a small number of blobs.

Cross Media Relevance Model Jeon et al. [21] improved the model of Duygulu et al. [17] by introducing a generative language model to image annotation, referred to as the cross-media relevance model (CMRM). They use the same process to extract and represent image blobs. However, instead of assuming one-to-one correspondence between the blobs and words, they assume that a set of blobs is related to a set of words. Thus, instead of seeking a probabilistic translation table, CMRM simply approximates the probability of observing a set of blobs and words in a given image.

In the CMRM model, it is assumed that, for a given un-annotated image I, there exists an underlying probability distribution (denoted as $P(.|I)$) of all possible blobs and words that could appear in image I. If the blob representation of I is $I = \{b_1, ..., b_m\}$, where m is the number of blobs in I, the probability of observing word w is approximated as

$$P(w|I) \simeq P(w|b_1, ...b_m) \tag{2.55}$$

For a given image, calculating $P(w|b_1, ...b_m)$ is equivalent to calculating the joint probability $P(w, b, ...b_m)$, which is approximated as the expectation over the entire training set. Using the assumption that words and blobs are generated independently given a training image J, $P(w, b, ...b_m)$ can then be computed using

$$P(w, b_1, ..., b_m) = \sum_{j \in \tau} P(J)P(w|J) \prod_{i=1}^{m} P(b_i|J) \tag{2.56}$$

where τ is the training set. The prior probabilities, $P(J)$, are kept uniform over all training images, while $P(w|J)$ and $P(b_i|J)$ are estimated by smoothed maximum likelihood.

The actual CMRM approach uses the K-means algorithm [28], and simple inverted lists of the obtained clusters to estimate $P(w|J)$ and $P(b_i|J)$. It also assumes that the events of observing a keyword w and blobs $b_1, ..., b_m$ are mutually independent once image J is selected. This assumption may result in many incorrect annotations and makes the CMRM very sensitive to the training images used to learn the model.

Semi-naive Bayesian Model More recently, Rui et al [9] proposed an approach based on the constrained K-means [19] to cluster image regions using partial supervision information. Then, they build a semi-naive Bayesian model for image annotation. In the learning stage of this approach, image segments are grouped into region clusters using the K-means algorithm with pair-wise constraints [29]. The set of *MustNot-link* relations are deduced from the irrelevance of all concepts annotating the images. In particular, if two images show little correlation in their annotation, then it is assumed that pairs of regions within these two images are semantically irrelevant. Under this assumption, Rui et al assert that for every image pair I_p and I_q, if their annotations C_p and C_q are irrelevant, then all relationships across their regions are marked as *MustNot-link*.

Once the pair-wise constraints between regions from different images is computed, the Pair-wise Constrained K-means (PCK-means) [29] is used to perform the clustering.

After clustering and identifying image region clusters, the dependency between two clusters is computed using

$$R_{co}(R_i, R_j) = \frac{|II(R_i) \cap II(R_j)|}{|II(R_i) \cup II(R_j)|}, \tag{2.57}$$

where

$$II(R_i) = \{I_j \,|\, \exists r \in I_j,\, l(r) = R_i\}. \tag{2.58}$$

Then, a greedy selection and joining (GSJ) algorithm is applied to find independent subsets of region clusters to be used in a semi-naive Bayesian (SNB) classifier .

The annotation algorithm described above has several limitations. First, it is based on a simple K-means clustering algorithm (section 2.1.2) to partition image regions into region categories. Since each region is usually represented by a high-dimensional feature vector that encodes its color, texture and structure information, a simple algorithm that uses the basic Euclidean distance and treats all features equally important may not be appropriate. Second, the set of constraints are extracted based on assumptions and are not necessarily valid. Another limitation is that the boundaries between region clusters is not well defined and using a simple inverted list to compute the dependency between region clusters (see eq.(2.57)) may not be effective.

Other Probabilistic Approaches Another annotation model that has shown promising performance is based on Latent Semantic Analysis (LSA) [74]. In this case, annotation is accomplished by finding the underlying semantic structure of words and image features in a linear latent space. For instance, in [40], Liu et al. reveal these latent variables of words and visual features using Probabilistic LSA (PLSA). The authors extend this approach to use a Nonlinear Latent Space and captures the dependency of images and words using Image-Word Embedding (IWE).

Another probabilistic approach was proposed by Blei and Jordan [41]. They describe three models which are built upon the assumption that images and words are generated by a mixture of latent factors, each model corresponding to the way images and words are generated. The Gaussian-multinomial mixture model assumes that the entire image and captions are conditional on the same factor, while the Gaussian-multinomial LDA model assumes that the image regions and captions are conditional on two disparate sets of factors. Both models are claimed to have some limitations. The third model, correspondence LDA, is a compromise of the former two. It assumes that the image regions can be conditional on any factors, but captions can only be conditional on factors that already exist in the images. Experiments showed that the third model outperforms the other two.

Carneiro et al. [42] proposed to estimate the semantic class distributions through a "pooling" process that is justified by Multiple Instance Learning (MIL) [43], without the need to segment the images.

2.3.2 Vector Space-Based Approaches

The vector space model framework is another common technique in information retrieval, especially text retrieval. Generally, documents are represented as vectors, each of which contains the occurrences of words within the document in question. The length of the vectors is equal to the vocabulary size. In this section, several automatic image annotation approaches that utilize the vector space model are outlined. These approaches treat images as documents, and build visual terms which are analogous to words, from the image feature descriptors.

The SvdCos Approach Pan et al. [44] proposed a series of auto-annotation methods which capture the association between words and blobs [17] through their pattern of occurrence over the entire training set. According to their reported results, the SvdCos method achieved the best performance. In this method, first, they construct a data matrix $D_{N,W+B} = [D_W|D_B]$, where $D_W(i,j)$ is the count of word w_j in image I_i, and $D_B(i,j)$ is the count of blob b_j in image I_i. After weighting the matrix D according to the uniqueness of every kind of blobs and words, they applied singular value decomposition (SVD) in order to "clean up noise and reveal informative structure". The largest singular values that preserve 90% of the variance were kept and the remaining were set to zero. Let $D_{SVD} = [D_{W,SVD}|D_{B,SVD}]$ denote the matrix after SVD. Then, they calculated a translation table T, where T_{ij} is the cosine value of the angle between the i^{th} column vector of D_W and the j^{th} column vector of D_B, i.e. $T_{i,j} = cos(D_W(i),\ D_B(j))$. Given a query image with a blob representation $q = [q_1, ..., q_B]$, the words to be predicted can be chosen from the term-likelihood vector $p = Tq$, where $p = [p_1, ..., p_W]^T$, and p_i is the likelihood of word w_i. This approach requires, the specification of the optimal number of blobs, which is not trivial when dealing with huge dataset.

Cross-Language Latent Semantic Indexing based Approach Dumais et al. [45] have demonstrated that Latent Semantic Indexing (LSI) can be used for cross-language information retrieval. Their system can perform text searching on a collection of French and English documents where queries could be in either language. This was realized by applying SVD to the term-by-document matrix in which the term vectors contain both French and English terms. As a result, the documents are projected into a low dimensional sub-space where co-occurrences of words from different languages were captured. Documents that are only in one language can then be mapped into the space and queried by keywords from the other language. Hare et al. [46] extended this approach to image retrieval of unannotated images through keyword queries. In terms of auto-annotation, the retrieval results indicate the likelihood of a label related to an image. This technique, called Cross-Language Latent Semantic Indexing (CL-LSI), is more suitable in bridging the semantic gap in image retrieval than in annotating image.

2.3.3 Classification-Based Approaches

Classification approaches for automatic image annotation view the process of attaching words to images as that of classifying images to a number of pre-defined groups, each of which is characterised by a concept or word.

Non-negative Matrix Factorization Approaches Non-negative matrix factorization (NMF) [48] is a matrix factorization technique that has become popular recently. Because of its non-negative constraints, many researchers [49] [50] from the information retrieval community regard it as more suitable for partial representation of data, such as text documents and images, and for further applications such as classification or retrieval. In [50], Xu et al. adopted the NMF approach to document classification. They factor the term-by-document matrix X into a basis matrix U and coefficient matrix V. The class label of a document is chosen as the one with the maximum value in the corresponding column of V. In [49], Guillamet et al. used NMF for image classification. They build a collection of image patches which were categorized into 10 classes. Both the training set and test set are built by randomly choosing 1000 patches respectively. For the training patches, they apply NMF in order to map them into a sub-space in which a classifier is learned. Given a test image to classify, they project it to all the 10 sub-spaces built from the training set and choose the one which achieves the high value based on the classifiers. This method is highly sensitive to the distance metric, and the optimal distance metric should be determined empirically which could be tedious and time consuming when the concerned dataset is huge. Moreover, it is practical only for a small number of classes.

2.3.3.1 Thesaurus Based Image Annotation

The thesaurus based image annotation approach (TBIA) [22] is based on image segmentation and clustering the visual features of all image regions. The cluster representatives are then used to create a visual thesaurus capable of translating region features into semantic labels. To address the high dimensionality of the feature space, the authors make use of an unsupervised learning algorithm that performs simultaneous clustering and attribute discrimination (SCAD) [36].

For each identified cluster, its visual prototype (closest image to centroid), the features of its centroid, the relevance weights for each feature subset, and the dominant keywords from the textual feature are used to form one visual profile. The visual profiles of all clusters constitute the multi-modal thesaurus. This thesaurus is then used to translate from one modality to another.

2.4 Major Contributions and Relation to Existing Work

This thesis has three mains components. The first one consists of a novel possibilistic clustering and feature weighting algorithm based on robust modeling of the Generalized Dirichlet (GD) finite mixture. Unlike the FCM and Gaussian distribution based algorithms, which seek symetric and spherical clusters, our approach exploits the property of the GD and can model clusters with different and asymetric shapes. Moreover, to overcome the sensitivity to noise and outliers of the existing FCM and GD based algorithms, our approach can handle noise points and outliers and limit their influence on the learned models by using possibilistic membership functions. We also address the problems associated with high-dimensional feature spaces of existing clustering methods by transforming the data to make the features independent and follow Beta distribution, and by learning an optimal relevance weight for each feature subset within each cluster.

Our second contribution consists of a novel approach to unsupervised image annotation. Our approach is based on : (i) the proposed semi-supervised possibilistic clustering; (ii) a greedy selection and joining algorithm (GSJ) to avoid the independency assumption used by most of the existing methods; (iii) Bayes rule; and (iv) a probabilistic model that is based on possibilistic memebership degrees generated by the clustering algorithm to annotate an image. We explore four variations and compare them to existing methods.

The third contribution consists of an image retrieval framework based on multi-modal similarity propagation. The proposed framework is designed to take advantages of the two data modalities: low-level visual features and high-level textual keywords generated by our proposed image annotation algorithm. The multi-modal similarity propagation system exploits the mutual reinforcement of relational data and results in a nonlinear combination of the different modalities to overcome the semantic gap problem. Specically, It is used to learn the semantic similarities between images by leveraging the relationships between features from the different modalities.

CHAPTER 3

DATA CLUSTERING BASED ON GENERALIZED DIRICHLET MIXTURE MODELS

The first step of our proposed image annotation process is to summarize the collection of image regions by few clusters of regions that share common attributes. Then, instead of analysing each individual region, we analyse the clusers' representatives to identify correlations among the different modalities. Summarizing the image region collection involves clustering sparse and high dimensional data. The problem is more acute when this high dimensional data are corrupted by noise and outliers. Generalized Dirichlet (GD) proved to be more appropriate for modeling data that are compactly supported, such as data originating from videos, images, or text. Our approach relies Generalized Dirichlet mixture to solve this challenge.

In this chapter, we first propose a novel possibilistic clustering approach based on robust modeling of Generalized Dirichlet finite mixture. This approach exploits a property of the Generalized Dirichlet distribution that transforms the data to make the features independent and follow a Beta distribution. Second, we extend our approach to learn feature relevance weights for each cluster. Third, we propose a semi-supervised version of this clustering. The supervision information consists of pairs of data points that should or should not be included in the same cluster. This partial supervision is used to guide the clustering process to avoid local minima and obtain more meaningfull clusters. Finally, we extend our approach to find the optimal number of clusters in an unsupervised and efficient way by exploiting some properties of the possibilistic membership function.

3.1 Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models

In this section, we propose a possibilistic approach for Generalized Dirichlet (GD) mixture parameter estimation and data clustering. This approach associates two types of memberships with each data sample. The first one is a posterior probability and indicates how well a sample fits each estimated distribution. The second membership represents the degree of typicality and is used to identify noise points and outliers. The proposed algorithm, called Robust and Unsupervised Learning of Finite Generalized Dirichlet Mixture Models (RULe_GDM), minimizes one objective function to optimize GD mixture parameters and possibilistic membership values. This optimization is done iteratively by dynamically updating the Dirichlet mixture parameters and the membership values in each iteration.

Let $Y = (\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_N)$ be a set of N points where $\mathbf{Y}_i \in \mathbb{R}^D$. We assume that $\mathbf{Y}$ is generated by a mixture of GD distributions with parameters $\theta = (\theta_1, \theta_2, ..., \theta_M, \mathbf{p}_1, ..., \mathbf{p}_M)$, where θ_j, is the parameter vector of the j^{th} GD component and $\mathbf{p}_j$ are the mixing weights. The finite GD mixture models the data using

$$p(\mathbf{Y}|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \, p(\mathbf{Y}|\theta_j), \tag{3.1}$$

where $p(\mathbf{Y}|\theta_j)$ is the GD distribution. Each $\theta_j = (\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, ..., \alpha_{jD}, \beta_{jD})$ is the parameter vector of the j^{th} GD component and $\mathbf{p}_j$ are the mixing weights where

$$\sum_{j} \mathbf{p}_j = 1 \quad for \; j = 1..M \tag{3.2}$$

Each GD distribution, $p(\mathbf{Y}|\theta_j)$, is defined as

$$\begin{aligned} p(\mathbf{Y}|\theta_j) &= p(\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_D|\theta_j), \\ &= \prod_{l=1}^{D} \frac{\Gamma(\alpha_{jl} + \beta_{jl})}{\Gamma(\alpha_{jl})\Gamma(\beta_{jl})} Y_l^{\alpha_{jl}-1} \left(1 - \sum_{k=1}^{l} Y_k\right)^{\gamma_{jl}} \end{aligned} \tag{3.3}$$

where $\sum_{l=1}^{D} \mathbf{Y}_l < 1$; $0 < \mathbf{Y}_l < 1$, for $l = 1, ..., D$; $\gamma_{jl} = \beta_{jl} - \alpha_{jl} - \beta_{jl+1}$, for $l = 1, ..., D-1$; and $\gamma_{jD} = \beta_{jD} - 1$.

In the mixture-based clustering, each $\mathbf{Y}_i$ is assigned to each component, j, with a posterior probability $p(j|\mathbf{Y}_i) \propto \mathbf{p}_j p(\mathbf{Y}_i|\theta_j)$. The GD distribution has a desirable property that allows the factorization of the posterior probability as

$$p(j|\mathbf{Y}_i) \propto \mathbf{p}_j \prod_{l=1}^{D} p_b(\mathbf{X}_{il}|\theta_{jl}), \tag{3.4}$$

where $\mathbf{X}_{i1} = \mathbf{Y}_{i1}$, and $\mathbf{X}_{il} = \frac{\mathbf{Y}_{il}}{1 - \sum_{k=1}^{l-1} \mathbf{Y}_{ik}}$ for $l > 1$. In (3.4), $p_b(\mathbf{X}_{il}|\theta_{jl})$

is a Beta distribution with $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$, $l = 1, ..., D$. In other words, the clustering structure underlying $\mathbf{Y}$ is the same as that underlying $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N)$ governed by

$$p(\mathbf{X}_i|\theta^*) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{l=1}^{D} p_b(\mathbf{X}_{il}|\theta_{jl}). \tag{3.5}$$

with conditionally independent features $\mathbf{X}$. Thus, the problem of estimating the parameters of the Generalized Dirichlet mixture of $\mathbf{Y}$ is reduced to the estimation of the Beta mixture of $\mathbf{X}$.

In the following, we formulate the identification of the M mixture components as an optimization problem. In particular, we define the following objective function

$$J = -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \left(log(p_j) + log\left[p(\mathbf{X}_i|\theta)\right] \right) + \sum_{j=1}^{M} \eta_j \sum_{i=1}^{N} (1 - \mathbf{u}_{ji})^m. \tag{3.6}$$

The first term in (3.6) is related to the log likelohood of all N points being fitted by M components. In this term, $\mathbf{u}_{ji}$ represents the possibilistic membership of point X_i in component j. We use a possibilistic membership [82] function that satisfies the constraints

$$\mathbf{u}_{ji} \in [0,\ 1],\ \ and\ 0 < \sum_{i=1}^{N} \mathbf{u}_{ji} < N \tag{3.7}$$

The membership value $\mathbf{u}_{ji}$ is high (close to 1) for point $\mathbf{X}_i$ that is typical of distribution j and low (close to 0) for points that do not fit the distribution. Points that do not fit any of the M distributions will have low membership values in all components (i.e low $\sum_{i=1}^{M} \mathbf{u}_{ji}$) and can be considered as noise.

The second term in (3.6) forces $\mathbf{u}_{ji}$ to be as large as possible to avoid the trivial solution of the first term where all $\mathbf{u}_{ji}$ are zero. The parameter η_j is a positive constants that control the importance of the second term with respect to the first one. It is related to the resolution parameter in the potential function and the deterministic annealing approaches [82]. It is also related to the idea of "scale" in robust statistics. In (3.6), $m \in [1, \infty)$ is called the fuzzifier.

Using (3.5), the objective function ((3.6) can be written as

$$J == -\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ji}^{m}\left(log(\mathbf{p}_j) + \sum_{l=1}^{D} log\left[p_b(\mathbf{X}_{il}|\theta_{jl})\right] \right) + \sum_{j=1}^{M}\eta_j\sum_{i=1}^{N}(1-u_{ji})^m (3.8)$$

subject to the membership constraint in (3.7). Since the columns of $\mathbf{U}$ are independent of each other, Minimizing J with respect to $\mathbf{U}$ is equivalent to minimizing the following individual objective functions with respect to each column j of $\mathbf{U}$:

$$J^{(j)}(U_j) = -\sum_{i=1}^{N} u_{ji}^{m}\left(log(\mathbf{p}_j) + \sum_{l=1}^{D} log\left[p_b(\mathbf{X}_{il}|\theta_{jl})\right] \right) + \sum_{i=1}^{N}(1-u_{ji})^m, (3.9)$$

for $j = 1, ..., M$. By setting the gradient of $J^{(j)}$ with respect to u_{ji} to zero, we obtain

$$\begin{aligned}
\frac{\partial J^{(j)}(\mathbf{U}_j)}{\partial u_{ji}} &= -\sum_{i=1}^{N} \frac{\partial u_{ji}^{m}\left(log(\mathbf{p}_j) + \sum_{l=1}^{D} log\left[p_b(\mathbf{X}_{il}|\theta_{jl})\right]\right)}{\partial u_{ji}} \\
&\quad +\eta_j \sum_{i=1}^{N} \frac{\partial(1-u_{ji})^m}{\partial u_{ji}} = 0 \\
&= -m(\mathbf{u}_{ji})^{m-1}\left(log(\mathbf{p}_j) + \sum_{l=1}^{D} log\Big(p_b(\mathbf{X}_{il}|\theta_{jl})\big)\right) \\
&\quad +m\eta_j(1-u_{ji})^{m-1} = 0 \\
&= -m(\mathbf{u}_{ji})^{m-1}\left(log\left[\mathbf{p}_j \prod_{l=1}^{D} p_b(\mathbf{X}_{il}|\theta_{jl})\right]\right) + m\eta_j(1-u_{ji})^{m-1} = 0 \\
= m \quad & (1-u_{ji})^{m-1} - m(u_{ji})^{m-1} - m\frac{\left(log\left[\mathbf{p}_j \prod_{l=1}^{D} p_b(\mathbf{X}_{il}|\theta_{jl})\right]\right)}{\eta_j}
\end{aligned} \tag{3.10}$$

This yields the following necessary condition to update the possibilistic membership degrees:

$$u_{ji} = \left[1 + \left(\frac{log\left[\mathbf{p}_j \prod_{l=1}^{D} p_b(\mathbf{X}_{il}|\theta_{jl})\right]}{\eta_j} \right)^{\frac{1}{m-1}} \right]^{-1}, \tag{3.11}$$

To minimize J with respect to $\mathbf{p}_j$ subject to (3.7), we use the Lagrange multiplier technique, and obtain

$$J = -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m}\Big(log(p_j) + log\left[\prod_{l=1}^{D}(p_b(\mathbf{X}_{il}|\theta_{jl})\right]\Big) - \lambda(\sum_{j=1}^{M}\mathbf{p}_j - 1). \tag{3.12}$$

By setting the gradient of J with respect to λ and $\mathbf{p}_j$ to zero, we obtain

$$\frac{\partial J}{\partial \lambda} = (\sum_{j=1}^{M}\mathbf{p}_j - 1) = 0, \tag{3.13}$$

and

$$\frac{\partial J}{\partial \mathbf{p}_j} = -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \frac{\partial log(\mathbf{p}_j)}{\partial \mathbf{p}_j} - \lambda = 0, \tag{3.14}$$

Solving equations (3.13) and (3.14) for $\mathbf{p}_j$ yields the following update equation for the GD mixture weights:

$$\mathbf{p}_j = \frac{\sum_{i=1}^{N} u_{ji}^{m}}{\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ji}^{m}}. \tag{3.15}$$

The presence of Gamma functions in the Beta distribution prevents obtaining a closed-form solution for θ_{jl} that minimizes J. Thus, to minimize J with respect to θ, we use the gradient descent method and estimate θ iteratively using

$$\theta_{jl}^{(t+1)} = \theta_{jl}^{(t)} - \lambda \frac{\partial J}{\partial \theta_{jl}} \tag{3.16}$$

where

$$\frac{\partial J}{\partial \theta_{jl}} = -\sum_{j=1}^{M}\sum_{i=1}^{N} u_{ji}^{m} \frac{\partial log\left[p_b(\mathbf{X}_{il}|\theta_{jl})\right]}{\partial \theta_{jl}} \tag{3.17}$$

It can be shown [75] that

$$\frac{\partial log\left[p_b(\mathbf{X}_{il}|\alpha_{jl})\right]}{\partial \alpha_{jl}} = \Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl}) + log(\mathbf{X}_{il}) \tag{3.18}$$

and

$$\frac{\partial log\left[p_b(\mathbf{X}_{il}|\beta_{jl})\right]}{\partial \beta_{jl}} = \Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\beta_{jl}) + log(1 - \mathbf{X}_{il}). \tag{3.19}$$

Algorithm 9 Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models (RULe_GDM)

```
Begin
 Fix the number of clusters $M$;
 Fix $m$, $m \in (1, \infty)$;
 Initialize $\mathbf{U}$ ,$\theta$, and $\eta$,
 Repeat
    Compute $log\left[p_b(\mathbf{X}_{il}|\theta_{jl})\right]$
    Update $\theta$ for few iterations using (3.16);
    Update the partition matrix $\mathbf{U}$ using (3.11);
    Update the mixture weights $\mathbf{p}$ using (3.15);
 Until ($\mathbf{U}$ stabilize)
End
```

In (3.18) and (3.19), $\Psi(.)$ is the gamma function. Thus,

$$\frac{\partial J}{\partial \alpha_{jl}} = -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \left(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\alpha_{jl}) + log(\mathbf{X}_{il})\right)$$

and

$$\frac{\partial J}{\partial \beta_{jl}} = -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \left(\Psi(\alpha_{jl} + \beta_{jl}) - \Psi(\beta_{jl}) + log(\mathbf{1} - \mathbf{X}_{il})\right).$$

The RULe_GDM algorithm is summarized below:

3.2 Robust Unsupervised Learning of Finite GD Mixture Models with Feature Discrimination

The objective function in (3.8) can be optimized to yield the parameters of the M distributions that best fit the data. However, in high dimensional feature space, as in Image database categorization we do not expect all D features to be equally relevant for all M components. To address this issue, we propose a modification to (3.8) to learn the relevant features for each component. We consider the l^{th} feature as irrelevant to cluster j if its distribution is independent of the corresponding component, i.e., if it follows density, denoted by $q(\mathbf{X}_l/\lambda_l)$, that is common to all components.

Let $\phi_j = (\phi_{j1}, ..., \phi_{jD})$ be a set of binary parameters, such that $\phi_{jl} = 1$ if feature l is relevant to cluster j and $\phi_{jl} = 0$ otherwise. The likelihood function in (3.5) can be rewritten as

$$p(\mathbf{X}_i|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{l=1}^{D} [p_b(\mathbf{X}_{il}|\theta_{jl})]^{\phi_{jl}} [q(\mathbf{X}_{il}|\lambda_l)]^{(1-\phi_{jl})}. \qquad (3.20)$$

Using an approach similar to the one in [93], we treat ϕ_{jl} as a missing variable and define the probability that the l^{th} feature is relevant to cluster j as the feature saliency $\rho_{jl} = P(\phi_{jl} = 1)$. Thus, equation (3.20) becomes

$$p(\mathbf{X}_i|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{l=1}^{D} \left(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1 - \rho_{jl}) q(\mathbf{X}_{il}|\lambda_l)\right). \qquad (3.21)$$

where $\theta = \{\{\mathbf{p}_j\}, \{\theta_{jl}\}, \{\lambda_l\}, \{\rho_{jl}\}\}$ includes all model parameters. An intuitive way to see how (3.21) is related to (3.20) is to notice that, because ϕ_{jl} is binary, $[p_b(\mathbf{X}_{il}|\theta_{jl})]^{\phi_{jl}}[q(\mathbf{X}_{il}|\lambda_l)]^{(1-\phi_{jl})}$ can be written as $\phi_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\phi_{jl})q(\mathbf{X}_{il}|\lambda_l)$.

We approximate irrelevant features by one distribution, q, that is common to all clusters and that reflects our prior knowledge about the distribution of irrelevant features. In particular, we consider the distribution of an irrelevant feature as a Beta distribution that is independent of the clusters.

By integrating the feature selection model in (3.21) into the objective function in (3.8), we minimize the following objective function

$$\begin{aligned} J &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\left(log(\mathbf{p}_j)+log\prod_{l=1}^{D}\left[\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)\right]\right) \\ &\quad +\sum_{j=1}^{M}\eta_j\sum_{i=1}^{N}(1-u_{ji})^{m}, \qquad (3.22) \\ &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\left(\mathbf{u}_{ji}^{m}log(\mathbf{p}_j)+\mathbf{u}_{ji}^{m}\sum_{l=1}^{D}log\left[\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)\right]\right) \\ &\quad +\sum_{j=1}^{M}\eta_j\sum_{i=1}^{N}(1-\mathbf{u}_{ji})^{m}, \qquad (3.23) \end{aligned}$$

subject to the membership constraint in (3.7). Since the coloumns of $\mathbf{U}$ are independent of each other, minimizing J with respect to $\mathbf{U}$ is equivalent to minimizing the following individual objective functions with respect to each column of $\mathbf{U}$:

$$\begin{aligned} J^{(j)}(\mathbf{U}_j) &= -\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\left(log(\mathbf{p}_j)+\sum_{l=1}^{D}\left(log(\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)\right)\right) \\ &\quad +\sum_{i=1}^{N}(1-\mathbf{u}_{ji})^{m}, \qquad (3.24) \end{aligned}$$

for $j = 1, ..., M$. By setting the gradient of $J^{(j)}$ with respect to u_{ji} to zero, we obtain

$$\begin{aligned}
\frac{\partial J^{(j)}(\mathbf{U}_j)}{\partial u_{ji}} &= -\sum_{i=1}^{N} \frac{\partial \mathbf{u}_{ji}^{m} \left(log(\mathbf{p}_j) + \sum_{l=1}^{D} \left(log(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l) \right) \right)}{\partial u_{ji}} \\
&\quad + \sum_{i=1}^{N} \frac{\partial (1-\mathbf{u}_{ji})^m}{\partial u_{ji}} = 0 \\
&= -m(\mathbf{u}_{ji})^{m-1} \Bigg(log(\mathbf{p}_j) + \sum_{l=1}^{D} log\Big(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) \\
&\quad + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l) \Big) \Bigg) + m\eta_j (1-u_{ji})^{m-1} = 0 \\
&= m(1-u_{ji})^{m-1} - m(u_{ji})^{m-1} \\
&\quad - m \frac{\left(log \left[\mathbf{p}_j \prod_{l=1}^{D} \left(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l) \right) \right] \right)}{\eta_j} = 0 \qquad (3.25)
\end{aligned}$$

This yields the following necessary condition to update the possibilistic membership degrees:

$$\mathbf{u}_{ji} = \left[1 - \left(\frac{log \left[\mathbf{p}_j \prod_{l=1}^{D} (\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l)) \right]}{\eta_j} \right)^{\frac{1}{m-1}} \right]^{-1} . \qquad (3.26)$$

Minimizing J with respect to the feature weights yields

$$\begin{aligned}
\frac{\partial J}{\partial \rho_{jl}} &= -\sum_{j=1}^{M} \sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \frac{\partial log \left[(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l) \right]}{\partial \rho_{jl}} \\
&= -\sum_{j=1}^{M} \sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \frac{\left[(p_b(\mathbf{X}_{il}|\theta_{jl}) - q(\mathbf{X}_{il}|\lambda_l) \right]}{\left[(\rho_{jl} p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl}) q(\mathbf{X}_{il}|\lambda_l) \right]} .
\end{aligned}$$

Setting $\frac{\partial J}{\partial \rho_{jl}}$ to zero, and assuming that ρ_{js} does not change significantly from iteration (t) to iteration $(t+1)$ we obtain the following update equation for ρ_{js} :

$$\rho_{jl}^{(t+1)} = \frac{p_b(\mathbf{X}_l|\theta_{jl}) - q(\mathbf{X}_l|\lambda_l)}{p_b(\mathbf{X}_l|\theta_{jl}) - q(\mathbf{X}_l|\lambda_l) + \frac{q(\mathbf{X}_l|\lambda_l)}{\rho_{jl}^{(t)}}}. \tag{3.27}$$

To minimize J with respect to $\mathbf{p}_j$ subject to (3.7), we use the Lagrange multiplier technique, and obtain

$$\begin{aligned} J &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\left(log(p_j) + log\left[\prod_{l=1}^{D}(\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl}) + (1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l))\right]\right) \\ &\quad -\lambda(\sum_{j=1}^{M}\mathbf{p}_j - 1). \end{aligned} \tag{3.28}$$

By setting the gradient of J with respect λ and $\mathbf{p}_j$ to zero, we obtain

$$\frac{\partial J}{\partial \lambda} = (\sum_{j=1}^{M}\mathbf{p}_j - 1) = 0, \tag{3.29}$$

and

$$\frac{\partial J}{\partial \mathbf{p}_j} = -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\partial log(\mathbf{p}_j)}{\partial \mathbf{p}_j} - \lambda = 0, \tag{3.30}$$

Solving equations (3.29) and (3.30) for $\mathbf{p}_j$ yields the following update equation for the GD mixture weights:

$$\mathbf{p}_j = \frac{\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}}{\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}}. \tag{3.31}$$

As in RULE_GDM, to minimize J with respect to θ and λ, we use the gradient descent method and estimate θ and λ, iteratively using

$$\theta_{jl}^{(t+1)} = \theta_{jl}^{(t)} - \xi_1\frac{\partial J}{\partial \theta_{jl}} \tag{3.32}$$

$$\lambda_l^{(t+1)} = \lambda_l^{(t)} - \xi_2\frac{\partial J}{\partial \lambda_l} \tag{3.33}$$

where

$$\begin{aligned}\frac{\partial J}{\partial \theta_{jl}} &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\partial log\left[\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)\right]}{\partial \theta_{jl}}\\ &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\rho_{jl}\frac{\partial p_b(\mathbf{X}_{il}|\theta_{jl})}{\partial \theta_{jl}}}{\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)},\end{aligned}$$

and

$$\begin{aligned}\frac{\partial J}{\partial \lambda_l} &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\partial log\left[\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)\right]}{\partial \lambda_l}\\ &= \sum_{j=1}^{M}\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\rho_{jl}\frac{\partial p_b(\mathbf{X}_{il}|\lambda_l)}{\partial \lambda_l}}{\rho_{jl}p_b(\mathbf{X}_{il}|\theta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)}.\end{aligned}$$

Thus,

$$\frac{\partial J}{\partial \alpha_{jl}} = -\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\rho_{jl}p_b(\mathbf{X}_{il}|\alpha_{jl})\left(\Psi(\alpha_{jl}+\beta_{jl})-\Psi(\alpha_{jl})+log(\mathbf{X}_{il})\right)}{\rho_{jl}p_b(\mathbf{X}_{il}|\alpha_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)} \tag{3.34}$$

and

$$\frac{\partial J}{\partial \beta_{jl}} = -\sum_{i=1}^{N}\mathbf{u}_{ji}^{m}\frac{\rho_{jl}p_b(\mathbf{X}_{il}|\beta_{jl})\left(\Psi(\alpha_{jl}+\beta_{jl})-\Psi(\beta_{jl})+log(\mathbf{1}-\mathbf{X}_{il})\right)}{\rho_{jl}p_b(\mathbf{X}_{il}|\beta_{jl})+(1-\rho_{jl})q(\mathbf{X}_{il}|\lambda_l)}. \tag{3.35}$$

The resulting algorithm, called Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models and Feature Selection (RULe_GDM_FS) is summarized below:

Algorithm 10 Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models and Feature Selection (RULe_GDM_FS)

```
Begin
Fix the number of clusters $M$;
 Fix $m$, $m \in (1, \infty)$;
 Initialize $\mathbf{U}$ ,$\theta$, $\lambda$, $\rho$, and $\eta$,
    Repeat
      Compute log $[p_b(\mathbf{X}_{il}|\theta_{jl})]$
      Update $\theta$ for few iterations using (3.32);
      Update $\lambda$ for few iterations using (3.33);
      Update the partition matrix $\mathbf{U}$ using (3.26);
      Update the mixture weights $\mathbf{p}$ using (3.31);
    Until ($\mathbf{U}$ stabilize)
End
```

3.3 Robust Unsupervised Learning of Finite GD Mixture Models with Feature Subset Selection

RULe_GDM_FS algorithm proposed in section 3.3 is designed to search for the optimal relevance weights for each feature within each cluster. However, for high-dimensional data learning relevance weights for each feature may lead to overfitting. To avoid this case, we propose a coarse approach to feature weighting. We assume that the D features have been partitioned into d subsets and that each subset s has k_s features, that is, $D = \sum_{s=1}^{d} k_s$. For instance, in the considered image region collection clustering, we may have one subset for color features, another one for texture features, and a third subset for structure features. We use $\mathbf{Y}_i^s$ to denote the components of $\mathbf{Y}_i$ that include only features from subset s.

The mixture of M GD distributions in (3.1) can be re-written as,

$$p(\mathbf{Y}|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \, p(\mathbf{Y}|\theta_j) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{s=1}^{d} p_b(\mathbf{Y}^s|\theta_j^s). \tag{3.36}$$

In (4.8), $\theta_j = (\alpha_j^1, \beta_j^1, \alpha_j^2, \beta_j^2, ..., \alpha_j^d, \beta_j^d)$ is the parameter vector of the j^{th} GD component and $\mathbf{p}_j$ are the mixing weights where $\sum_j p_j = 1$, for $j = 1..M$.

The factorization of the posterior probability in (3.4) becomes

$$\begin{aligned} p(j|\mathbf{Y}_i) &\propto \mathbf{p}_j \prod_{s=1}^{d} p_b(\mathbf{X}_i^s|\theta_j^s), \\ &\propto \mathbf{p}_j \prod_{s=1}^{d} \prod_{l=1}^{k^s} p_b(\mathbf{X}_{il}^s|\theta_{jl}^s) \end{aligned} \tag{3.37}$$

Where $\mathbf{X}$ is the data representation in the new feature space as outlined in section (3.3). In (3.37), $p_b(\mathbf{X^s}_{il}|\theta_{jl}^s)$ is a Beta distribution with $\theta_{jl}^s = (\alpha_{jl}^s, \beta_{jl}^s)$, $l = 1, ..., k^s$. That is, the clustering structure underlying $\mathbf{Y}$ is the same as that underlying $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N)$ governed by

$$p(\mathbf{X}_i|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{s=1}^{d} p_b(\mathbf{X}_i^s|\theta). \tag{3.38}$$

Instead of assuming a set a binary parameters for each feature, let $\phi_j = (\phi_{j1}, ..., \phi_{jd})$ be a set of binary parameters, such that $\phi_{js} = 1$ if feature subset s is relevant to cluster j and $\phi_{js} = 0$ otherwise. We treat ϕ_{js} as a missing variable and define the probability that the s^{th} feature subset is relevant to cluster j as the feature saliency $\rho_{js} = P(\phi_{js} = 1)$. Thus, the likelihood function in (3.38) can be rewritten as

$$p(\mathbf{X}_i|\theta) = \sum_{j=1}^{M} \mathbf{p}_j \prod_{s=1}^{d} \left(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)\right). \qquad (3.39)$$

where $\theta = \{\{\mathbf{p}_j\}, \{\theta_j^s\}, \{\lambda_s\}, \{\rho_{js}\}\}$ includes all model parameters. We approximate irrelevant feature subsets by one distribution, q, that is common to all clusters. In particular, we consider the distribution of an irrelevant feature subset as a Beta distribution that is independent of the clusters.

By integrating the feature selection model in (3.39) into the objective function in (3.23), we minimize

$$\begin{aligned} J &= -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \Big(log(\mathbf{p}_j) + \sum_{s=1}^{d} log\left[\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s)\right. \\ &\quad \left. + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)\right]\Big) + \sum_{j=1}^{M} \eta_j \sum_{i=1}^{N} (1-\mathbf{u}_{ji})^m, \qquad (3.40) \end{aligned}$$

subject to the membership constraint in (3.7).

Minimizing J with respect to $\mathbf{U}$ is equivalent to minimizing the following individual objective functions with respect to each column of $\mathbf{U}$:

$$\begin{aligned} J^{(j)}(\mathbf{U}_j) &= -\sum_{i=1}^{N} \mathbf{u}_{ji}^{m} \Big(log(\mathbf{p}_j) + \sum_{s=1}^{d} log[\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) \\ &\quad + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)]\Big) + \eta_j \sum_{i=1}^{N} (1-\mathbf{u}_{ji})^m, \qquad (3.41) \end{aligned}$$

for $j = 1, ..., M$. By setting the gradient of $J^{(j)}$ with respect to u_{ji} to zero, we obtain

$$\begin{aligned}
\frac{\partial J^{(j)}(\mathbf{U}_j)}{\partial u_{ji}} &= -\sum_{i=1}^{N} \frac{\partial \mathbf{u}_{ji}^{m} \left(log(\mathbf{p}_j)+ \sum_{s=1}^{d} log\left(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s)\right)\right)}{\partial u_{ji}} \\
&\quad +\eta_j \sum_{i=1}^{N} \frac{\partial(1-\mathbf{u}_{ji})^m}{\partial u_{ji}} = 0 \\
&= -m(\mathbf{u}_{ji})^{m-1} \Biggl(log(\mathbf{p}_j) + \sum_{s=1}^{d} log[\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) \\
&\quad +(1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s)] \Biggr) + m\eta_j(1-u_{ji})^{m-1} = 0 \\
&= m(1-u_{ji})^{m-1} - m(u_{ji})^{m-1} \\
&\quad -m\frac{\Bigl(log\Bigl[\mathbf{p}_j \prod_{s=1}^{d}(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s)\Bigr]\Bigr)}{\eta_j} = 0 \qquad (3.42)
\end{aligned}$$

This yields the following necessary condition to update the possibilistic membership degrees:

$$\mathbf{u}_{ji} = \left[1 - \left(\frac{log\left[\mathbf{p}_j \prod_{s=1}^{d}(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s))\right]}{\eta_j}\right)^{\frac{1}{m-1}}\right]^{-1}. \tag{3.43}$$

Setting $\frac{\partial J}{\partial \rho_{js}}$ to zero, and assuming that ρ_{js} does not change significantly from iteration (t) to iteration $(t+1)$ we obtain the following update equation for ρ_{js} :

$$\rho_{js}^{(t+1)} = \frac{p_b(\mathbf{X}^s|\theta_j^{s*}) - q(\mathbf{X}^s|\lambda_s)}{p_b(\mathbf{X}^s|\theta_j^{s*}) - q(\mathbf{X}^s|\lambda_s) + \frac{q(\mathbf{X}^s|\lambda_s)}{\rho_{js}^{(t)}}}. \tag{3.44}$$

Minimizing (3.40) with respect to GD mixture weights yields the same update equation as in section 3.3.

As outlined in section , to minimize J with respect to θ and λ, we use the gradient descent method and estimate $\hat{\theta}$ and $\hat{\lambda}$ iteratively using

$$\begin{aligned} \theta_j^{s(t+1)} &= \theta_j^{s(t)} - \xi_1 \frac{\partial J}{\partial \theta_j^s}, & (3.45) \\ \lambda_s^{(t+1)} &= \lambda_s^{(t)} - \xi_2 \frac{\partial J}{\partial \lambda_s} & (3.46) \end{aligned}$$

where

$$\begin{aligned} \frac{\partial J}{\partial \theta_j^s} &= -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\partial log\left[(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)\right]}{\partial \theta_j^s} \\ &= -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\rho_{js} \frac{\partial p_b(\mathbf{X}_i^s|\theta_j^s)}{\partial \theta_j^s}}{(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial J}{\partial \lambda_s} &= -\sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\partial log\left[(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)\right]}{\partial \lambda_s} \\ &= \sum_{j=1}^{M}\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\rho_{jl} \frac{\partial p_b(\mathbf{X}_i^s|\lambda_s)}{\partial \lambda_s}}{(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)}. \end{aligned}$$

Thus,

$$\frac{\partial J}{\partial \alpha_j^s} = -\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\rho_{js} p_b(\mathbf{X}_i^s|\alpha_j^s)(\Psi(\alpha_j^s + \beta_j^s) - \Psi(\alpha_j^s) + log(\mathbf{X}_i^s))}{\rho_{js} p_b(\mathbf{X}_i^s|\alpha_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)} \tag{3.47}$$

and

$$\frac{\partial J}{\partial \beta_j^s} = -\sum_{i=1}^{N} \mathbf{u}_{ji}^m \frac{\rho_{js} p_b(\mathbf{X}_i^s|\beta_j^s)(\Psi(\alpha_j^s + \beta_j^s) - \Psi(\beta_j^s) + log(\mathbf{1} - \mathbf{X}_\mathbf{i}^\mathbf{s}))}{\rho_{js} p_b(\mathbf{X}_{il}|\beta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)}. \tag{3.48}$$

The resulting algorithm, called Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models and Feature Subset Selection (RULe_GDM_FSS) is summarized below:

Algorithm 11 Robust Unsupervised Learning of Finite Generalized Dirichlet Mixture Models and Feature Subset Selection (RULe_GDM_FSS)

```
Begin
 Fix Thre, $m \in [1, \infty)$;
 Fix the number of clusters $M$.
 Initialize $\mathbf{U}$ ,$\theta$, $\lambda$, $\rho$, and $\eta$,
    Repeat
       Compute log $[p_b(\mathbf{X}_i^s|\theta_{js})]$
       Update $\theta$ and $\lambda$ for few iterations using (3.45) and (3.46);
       Update $\mathbf{U}$ and $\mathbf{p}$ using (3.43) and (3.31);
    Until ($\mathbf{U}$ stabilize)
End
```

3.4 Semi-supervised Possibilistic Clustering and Feature Subset Weighting based on Robust GD Mixture Modeling

The unsupervised learning approaches proposed in this chapter require estimating several parameters using complex optimization and is prone to several local minima. Moreover. a large amount of data is required to obtain accurate estimates of the parameters of the Generalized Dirichlet mixture. To overcome this potential drawback, we propose a semi-supervised version of those algorithms. The supervision information consists of two types of pairwise constraints. The first one is *Should-link* constraints and specifies that two data points should be assigned to the same cluster. The second type of constraint is *ShouldNot-link* and specifies that two data points should not be assigned to the same cluster.

Let SL be the set of *Should-link* pairs such as $(X_i, X_j) \in SL$ implies that X_i and X_j should be assigned to the same cluster. Similarly, we let NL be the set of *ShouldNot-link* pairs such as $(X_i, X_j) \in NL$ means that X_i and X_j should not be assigned to the same cluster. We reformulate the problem of identifying the M mixture components in section 3.3 as a constrained optimization problem. In particular, we modify the objective function in (3.40) as follow

$$\begin{aligned} J &= -\sum_{j=1}^{M}\sum_{i=1}^{N}\Bigg(\mathbf{u}_{ji}^{m} log(\mathbf{p}_j) + \mathbf{u}_{ji}^{m}\sum_{s=1}^{d} log[\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) \\ & \quad +(1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s))]\Bigg) + \sum_{j=1}^{M}\eta_j\sum_{i=1}^{N}(1-\mathbf{u}_{ji})^m \\ & \quad +\mu\left[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)}\sum_{j=1}^{M}\mathbf{u}_{jt}^{m}\mathbf{u}_{jk}^{m} + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)}\sum_{j=1}^{M}\sum_{p=1,p\neq j}^{M}\mathbf{u}_{jt}^{m}\mathbf{u}_{pk}^{m}\right] \end{aligned} \quad (3.49)$$

subject to the membership constraint in (3.7).

The last term in (3.49) consists of the cost of violating the pairwise *Should-link,* and *ShouldNot-link* constraints. The value of μ controls the importance of the supervision information compared to the first term which is related to the log likelihood of all N points being fitted by M components. The second term as in RULe_GDM forces the $\mathbf{u}_{ji}$ to be as large as possible to avoid the trivial solution of the first term where all $\mathbf{u}_{ji}$ are zero.

Minimizing J with respect to $\mathbf{U}$ is equivalent to minimizing the following individual objective functions with respect to each column of $\mathbf{U}$:

$$\begin{aligned} J^{(j)}(\mathbf{U}_j) &= -\sum_{i=1}^{N} \mathbf{u}_{ji}^m \Bigg(log(\mathbf{p}_j) + \sum_{s=1}^{d} log[\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) \\ &\quad +(1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)]\Big) + \eta_j \sum_{i=1}^{N} (1-\mathbf{u}_{ji})^m \\ &\quad +\mu \left[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jt}^m \mathbf{u}_{jk}^m + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{jt}^m \mathbf{u}_{pk}^m \right], \end{aligned} \tag{3.50}$$

for $j = 1, ..., M$. By setting the gradient of $J^{(j)}$ with respect to $\mathbf{u}_{ji}$ to zero, we obtain

$$\begin{aligned} \frac{\partial J^{(j)}(\mathbf{U}_j)}{\partial u_{ji}} &= -\sum_{i=1}^{N} \frac{\partial \mathbf{u}_{ji}^m \left(log(\mathbf{p}_j) + \sum_{s=1}^{d} log\left(\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)\right)\right)}{\partial u_{ji}} \\ &\quad +\eta_j \sum_{i=1}^{N} \frac{\partial (1-\mathbf{u}_{ji})^m}{\partial u_{ji}} + \mu \frac{\partial \left[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jt}^m \mathbf{u}_{jk}^m \right]}{\partial u_{ji}} \\ &\quad +\mu \frac{\partial \left[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{jt}^m \mathbf{u}_{pk}^m \right]}{\partial u_{ji}} = 0 \\ &= -m(\mathbf{u}_{ji})^{m-1} \Bigg(log(\mathbf{p}_j) + \sum_{s=1}^{d} log[\rho_{js} p_b(\mathbf{X}_i^s|\theta_j^s) \\ &\quad +(1-\rho_{js}) q(\mathbf{X}_i^s|\lambda_s)] + \mu \left[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jk}^m + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{pk}^m \right] \Bigg) \\ &\quad +m\eta_j (1-u_{ji})^{m-1} = 0 \end{aligned}$$

$$
\begin{aligned}
&= m(1-u_{ji})^{m-1} - m(u_{ji})^{m-1} \\
&\quad - m\frac{\Big(log\Big[\mathbf{p}_j \prod_{s=1}^{d}(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s)\Big]\Big)}{\eta_j} \\
&\quad - m\frac{\Bigg(\mu\Bigg[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jk}^m + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{pk}^m\Bigg]\Bigg)}{\eta_j} = 0
\end{aligned}
$$

This yields the following necessary condition to update the possibilistic membership degrees:

$$
\mathbf{u}_{ji} = \left[1 - \Bigg(\frac{m\Big(log\Big[\mathbf{p}_j \prod_{s=1}^{d}(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s))\Big]}{\eta_j} \frac{+\mu\Bigg[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jk}^m + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{pk}^m\Bigg]\Big)}{\eta_j}\Bigg)^{\frac{1}{m-1}}\right]^{-1} \tag{3.51}
$$

The term $m\Big(log\Big[\mathbf{p}_j \prod_{s=1}^{d}(\rho_{js}p_b(\mathbf{X}_i^s|\theta_j^s) + (1-\rho_{js})q(\mathbf{X}_i^s|\lambda_s)\Big] + \mu\Bigg[\sum_{(\mathbf{X}_t,\mathbf{X}_k \in NL)} \mathbf{u}_{jk}^m + \sum_{(\mathbf{X}_t,\mathbf{X}_k \in SL)} \sum_{p=1,p\neq j}^{M} \mathbf{u}_{pk}^m\Bigg]\Big)$ can be viewed as the total cost when considering point $\mathbf{X}_i$ in cluster j. This cost depends on the posterior probabilties, and the cost of the violated constraints due to cluster j.

Since the third term in (3.49) does not depend on the distribution parameters, the GD mixing weights, and the feature subset weights, minimizing

Algorithm 12 Semi-supervised Robust Learning of Finite Generalized Dirichlet Mixture Models and Feature Subset Selection (SRLe_GDM_FSS)

```
Begin
 Fix Thre, $m \in [1, \infty)$;
 Let $M$ be an overspecified number of clusters.
 Fix the set of ShouldLink (SL) and ShouldNotLink (CL) constraints.
 Initialize $\mathbf{U}$ ,$\theta$, $\lambda$, $\rho$, and $\eta$,
    Repeat
       Compute log $[p_b(\mathbf{X}_i^s|\theta_{js})]$
       Update $\theta$ and $\lambda$ for few iterations using (3.45) and (3.46);
       Update $\mathbf{U}$ and $\mathbf{p}$ using (3.48) and (3.31);
    Until ($\mathbf{U}$ stabilize)
 Merge similar clusters.
End
```

(3.49) with respect to θ_{js}, p_j, and ρ_{js} yields the same update equations as in section 3.4.

The resulting algorithm, called Semi-supervised Robust Learning of Finite Generalized Dirichlet Mixture Models and Feature Subset Selection (SRLe_GDM_FSS) is summarized below:

3.5 Finding the Optimal Number of Clusters

A nice property of the proposed Generalized Dirichlet based algorithms is that they associate a possibilistic membership degree with every sample in every cluster. Moreover, the memberships of a given point in all clusters are independent of each other and are not constrained to sum to 1. Thus, if we start with an initial partition that has an overspecified number of clusters/models M, clusters will be created independently of each other and many of them will converge to the same dense regions in the feature space. This observation is illustrated in Figure 3.5(a) with a simple synthetic data set consisting of 2 clusters. We do not assume that we know the number of components, and we overspecify this value to 5. For each component, the proposed algorithms learn the GD model parameters and the parameters of its possibilistic membership function. Then, for each point in the feature space, we compute its possibilistic membership in all 5 clusters. These membership functions are displayed in Figure 3.5(b)-(f). As it can be seen, clusters 2, 3, 4 and 5 have very similar distributions. That is, these 4 clusters are very similar and are modeled by 4 similar distributions.

To detect similar clusters, we use the cluster similarity measure proposed in [94]. Given two clusters j_1 and j_2 , we compute their fuzzy similarity using the membership values of all points in the two clusters.

$$S(j_1, j_2) = 1 - \frac{\sum_{i=1}^{N} abs(u_{j_1 i} - u_{j_2 i})}{\sum_{i=1}^{N} u_{j_1 i} - \sum_{i=1}^{N} u_{j_2 i}} \tag{3.52}$$

Clusters that have a similarity values larger than a certain threshold get merged, and the number of clusters is updated accordingly.

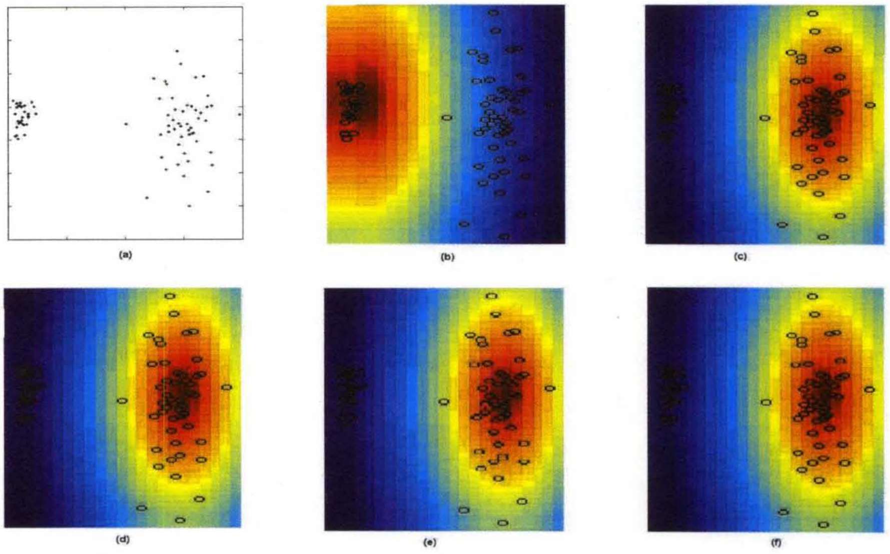


Figure 3.1: Finding the optimal number of clusters. (a) data set containing two Beta distributions, (b)-(f) Possibilistic membership of every point in the feature space in the 5 identified clusters.

Thus, RULe_GDM is extended to the following algorithm

Algorithm 13 Extension of RULe_GDM Algorithm

```
Begin
 Fix Thre, $m \in (1, \infty)$;
 Let $M$ be an overspecified number of clusters.
 $merge = 1$;
 While ($merge$)
 $merge = 0$;
 Initialize $\mathbf{U}$ ,$\theta$, $\rho$, and $\eta$,
    Repeat
       Compute $log\,[p_b(\mathbf{X}_{il}|\theta_{jl})]$
       Update $\theta$ for few iterations using (3.16);
       Update the partition matrix $\mathbf{U}$ using (3.11);
       Update the mixture weights $\mathbf{p}$ using (3.15);
    Until ($\mathbf{U}$ stabilize)
 For each pair of clusters $i$ and $j$ compute $S(i,j)$ using (3.52)
          If ($S(i,j) \geq Thre$)
            Merge cluster $i$ and cluster $j$.
           Update the number of clusters $M$
           Set $merge = 1$;
          End
 End
End
End
```

3.6 Experimental results

We first illustrate the performance of the proposed Algorithms using synthetic data sets. For all results reported using the Generalized Dirichlet mixture based algorithms, we use the following initialization scheme. First, we partition the data into M clusters using the fuzzy C-means (2.1.3). Then, we use the method of moments (MM) [112] to obtain initial beta distribution parameters for each cluster. For each iteration, we update θ and λ using (3.32) and (3.33) for 3 iterations.

	cluster #1		**cluster #2**	
	x1	**x2**	**x1**	**x2**
relevance weights	0.4973	0.5027	0.4819	0.5180
Alpha	6.3597	4.8794	91.3969	109.0321
Beta	70.6678	133.0715	48.0923	43.7654

Table 3.1: Parameters learned by RULe_GDM_FS for two Beta distributed clusters

	cluster #1				**cluster #2**			
Features	**x1**	**x2**	**x3**	**x4**	**x1**	**x2**	**x3**	**x4**
Relevance weights	0.39	0.09	0.40	0.12	0.06	0.45	0.11	0.38
Alpha	6.35	1.16	4.87	1.12	2.30	91.39	1.53	109.03
Beta	70.68	1.23	133.8	1.28	3.01	47.75	1.95	43.13

Table 3.2: Parameters learned by RULe_GDM_FS for a 4-dimensional data containing irrelevant features

We generate two Beta distributed clusters. Each cluster contains 200 points. We fix the fuzzyfier m to 2, and the resolution parameter for the possibilistic membership function, η_j, to 0.7 for all clusters. RULe_GDM_FS converged after 3 iterations, and the estimated parameters of the two distributions are displayed in Table 3.1. These parameters are very close to those used to generate the data. Also, since both features are equally important for both distributions, RULe_GDM_FS assigns similar relevance weights (close to 0.5) to each feature.

To demonstrate the ability of RULe_GDM_FS to cluster and identify relevant features, we increase the number of features in the previous data to four by adding two irrelevant features uniformely distributed in the interval [0 1]. We reorganize the features so that different features are relevant for different clusters. In particular, for cluster 1, features 2 and 4 are irrelevant and for cluster 2, features 1 and 3 are irrelevant.

The RULe_GDM_FS algorithm converged after 5 iterations, and the results are displayed in Table 3.2. As it can be seen, for cluster 1, the first and third features were correctly identified as the most relevant features. On the other hand, features two and four were identified as less relevant ones. Similarly, for cluster 2, the second and fourth features were correctly identified as relevant features, and features one and three were detected as irrelevant and assigned lower weights. In table 3.2, we also show the estimated Beta distribution parameters of the two clusters. As it can be seen, the obtained values are similar to those reported in Table 3.1 obtained before adding the irrelevant clusters. Thus, by detecting the irrelevant features and assigning low weights to them, the distribution of the relevant features can be estimated robustly.

To assess the robustness of RULe_GDM_FS with respect to noise and outliers, we generate a synthetic data set from two 2D Beta distributions with different parameters. 200 points were generated from each distribution. In addition, we generate 200 noise points (uniformly distributed in $[0,1]$). This dataset is shown in Figure 3.2(a). In Figure 3.2(b) and (c), we display the partitions obtained with the method proposed in [38], and using Gaussian mixture model as described in [1] respectively. Each data point is assigned to the component that has the highest posterior probability. We should emphasize here that since the sum of the posterior probabilities is 1, noise points cannot be identified and get assigned to the closest component. Moreover, since their posterior probability can be high (close to 1), they can affect the estimated parameters significantly. In Figure 3.2(d), we display the partition obtained using RULe_GDM_FS. Points that are assigned low possibilistic memberships (<0.1) in both clusters (i.e. noise points) are displayed using the '+' symbol. As it can be seen, the obtained partition reflects the true structure of the data and the identified noise points would have a minimal effect on the estimated parameters.

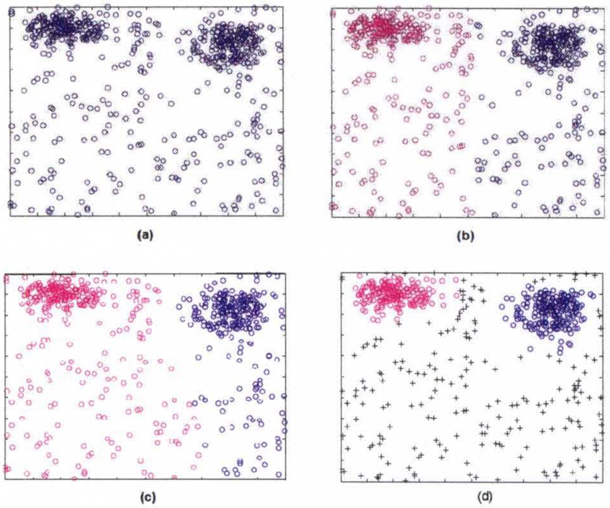


Figure 3.2: Clustering 2 Beta distributions corrupted with uniform noise. (a) data set, (b) partition obtained with the method in [113], (c) partition obtained using Gaussian mixture model as described in [85], and (d) partition obtained using RULe_GDM_FS. Identified noise points are displayed with a '+' sign

In Table 3.3, we display the true model parameters used to generate the clusters in Figure 3.2(a) and the estimated parameters obtained with RULe_GDM_FS and the method in [113]. As expected, noise affects the parameters estimated with the EM method. On the other hand, RULe_GDM_FS can identify noise points and assigns low possibilistic memberships to them, and thus, limiting their influence on the estimated model parameters.

To assess the robustness of RULe_GDM_FS in high dimensional spaces, we generate a data set with two Beta distributed clusters in a 30-dimensional feature space. Each cluster contains 3000 points. We increase the noise rate

	cluster 1	**cluster 2**
Model parameters $\begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix}$	[19.05 30.16] [6.07 7.10]	[4.95 53.17] [20.0 5.52]
Estimated with the method in [113]	[17.39 25.22] [3.11 5.74]	[8.11 53] [12.06 4.24]
Estimated with RULe_GDM_FS	[18.91 29.11] [6.13 6.94]	[4.99 53.21] [20.01 5.51]

Table 3.3: Comparaison of the parameters used to generate the data to the parameters estimated using the method in [113] and RULe_GDM_FS.

progressively from 10 to 50%. For each run, we compare the obtained partition to the ground truth and compute the relative accuracy. As it can be seen in Figure 3.3, the performance of RULe_GDM_FS degraded at a much lower rate than the performance of the method in [38].

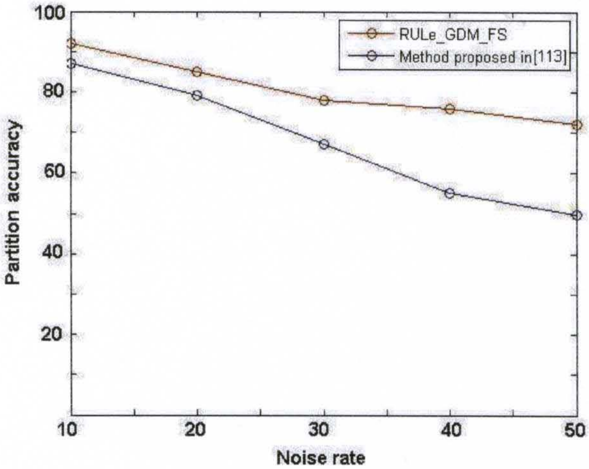


Figure 3.3: Comparison of the accuracy of the data clustered with the method in [113] and RULe_GDM_FS when the dimensionality of the feature space is fixed to 30 and the noise rate is varied from 10 to 50 %

In the next experiment, we assess the robustness of RULe_GDM_FS as we vary the dimensionality of the feature space. We generate two Beta distributed clusters. Each cluster contains 3000 points. We fix the noise rate to 30% (2000 points) and we increase the dimensionality of the feature space progressively from 2 to 40. For each run, we compare the accuracy of each algorithm. As it can be seen in Figure 3.4, using the method in [113] the accuracy decreases from 69 to 50%. On the other hand, the accuracy of RULe_GDM_FS remained above 70%.

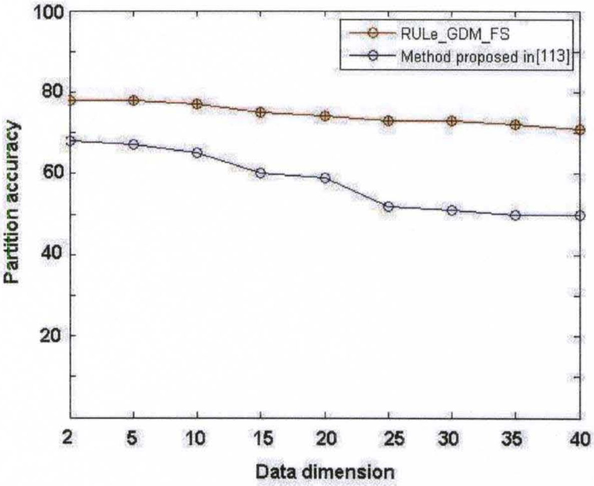


Figure 3.4: Comparison of the accuracy of the data clustered with the method in [113] and RULe_GDM_FS when the noise rate is fixed at 30% and the dimensionality of the feature space is varied from 2 to 40.

CHAPTER 4

IMAGE ANNOTATION BASED ON CONSTRAINED REGION CLUSTERING

In this chapter, we describe our image annotation approach that relies on: (*i*) semi-supervised clustering and feature weighting; (*ii*) a greedy selection and joining algorithm (GSJ); (*iii*) Bayes rule; and (*iv*) membership based Cross Media Relevance Model (CMRM). Clustering is used to group image regions into region clusters and provide a summary of the training data. These summaries will be used as the basis for annotating new test images. Since this learning task involves clustering sparse and high dimensional data that are corrupted by noise and outliers, we use a semi-supervised constrained learning apprach that performs simultaneous clustering and feature weighting. The constraints consist of pairs of image regions that should not be included in the same cluster. These constraints are deduced from the irrelevance of all concepts annotating the training images, and are

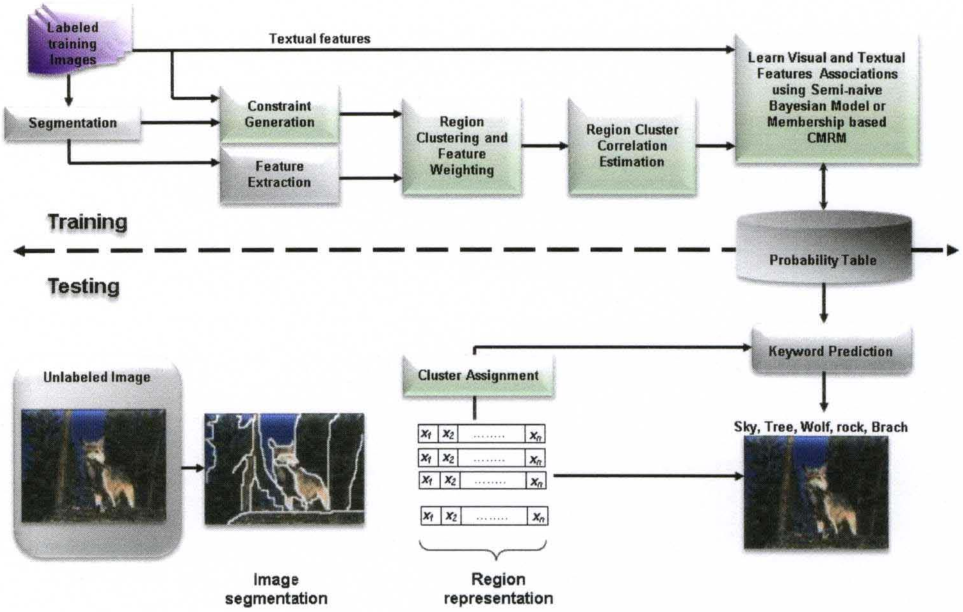

Labeled training images
Textual features
Segmentation
Constraint Generation
Feature Extraction
Region Clustering and Feature Weighting
Region Cluster Correlation Estimation
Learn Visual and Textual Features Associations using Semi-naive Bayesian Model or Membership based CMRM
Training
Testing
Probability Table
Keyword Prediction
Cluster Assignment
Unlabeled Image
x_1 x_2 .... x_n
Sky, Tree, Wolf, rock, Brach
Image segmentation
Region representation

Figure 4.1: Overview of the proposed image annotation system

used to guide the clustering process.

The GSJ algorithm uses the fuzzy membership values generated by the clustering algorithm to compute a degree of mutual dependency among the clusters. Finally, Bayes rule and a membership based CMRM are used to label images based on their posterior probability in each concept.

Figure 4.1 gives an overview of the proposed image annotation system. For the training phase, the labeled training images are segmented into homogeneous regions and each region inherits the annotating keywords of its image. We extract multiple visual features from each image region and combine them to form one feature descriptor for the region. This high dimensional feature representation is needed to represent the diverse image regions. However, it results in a very sparse feature space where features are not equally relevant to all categories. Consequently, standard unsupervised clustering algorithm may not perform well for this application. To overcome this problem, we derive a set of constraints from the co-occurence of the annotating keywords. These constraints are then used within our proposed semi-supervised clustering and feature weighting algorithm to guide the clustering process.

After region clustering, we propose two different approaches to learn associations or joint probability distributions of region clusters and textual vocabulary. The first one uses a semi-naive Bayesian model to estimate the posterior probability of each keyword given a set of image region clusters. The second one consists of a membership based Cross Media Relevance Model. Both of these approaches use a greedy Selection and Joining algorithm to avoid making the assumption that region clusters are independent.

The testing part of the proposed system takes, as input, an unlabeled image, segments it into homogeneous regions, extracts and encodes the visual content of each region by a feature vector, and assigns each image region to one of the predefined region categories. Then, it uses the learned models to infer a set of keywords that best describe the image. These keywords are then used to annotate the image.

The rest of this chapter is organized as follows. In section 4.1, we describe the format of the training data and its feature representation. We also outline the constraints fomulation, and the semi-supervised clustering and feature weighting algorithm used to summarize the training image regions. Then, in section 4.2, we outline the proposed image annotation approaches. The experiments used to evaluate the performance of the proposed methods are described in section 4.3.

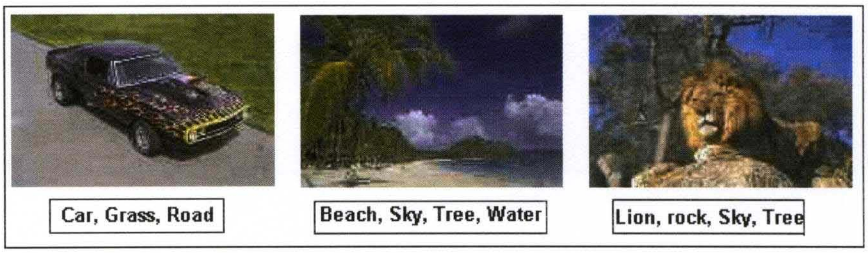


Figure 4.2: Examples of globally annotated images

4.1 Image Database Organization

We assume that we have a training image collection, τ, that contains a total of N images, and that each image is labeled by 1 to m keywords. The keywords provide global description of the images and are not explicitly associated with specific regions. Figure 4.2 provides a sample of three annotated images. This type of annotation does not require image segmentation and could be easily generated.

4.1.1 Image Segmentation

Each training image is segmented into a small number of homogeneous regions. Segmentation is achieved by clustering the pixels' color information. We use the Competitive agglomeration (CA) (detailed in section 2.1.5). Our choice is based on the computational efficiency of this algorithm and its ability to cluster each image into an optimum number of regions. The initial segmentation of the database is carried out offline and computational efficiency is not a major issue. However, for test images, segmentation must be carried out online, and this process should be efficient.

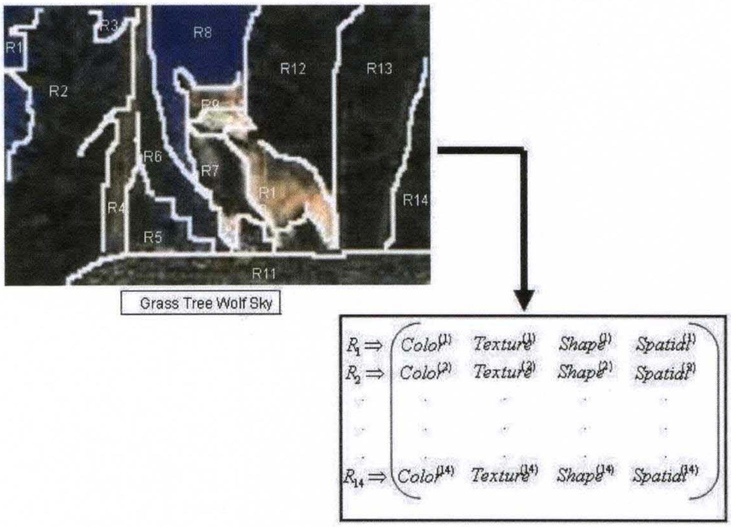

R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
Grass Tree Wolf Sky
R1 ⇒ Color(1) Texture(1) Shape(1) Spatial(1)
R2 ⇒ Color(2) Texture(2) Shape(2) Spatial(2)
R14 ⇒ Color(14) Texture(14) Shape(14) Spatial(14)

Figure 4.3: Visual feature representation

4.1.2 Feature Extraction and Representation

After segmenting the training images, all image regions are represented by various features that represent color, texture and structure information.

Formally, each region r_j is represented by q feature subsets. Let r_j^s be the representation of region r_j by the s^{th} featue subset. Each r_j^s is represented by a d_s-dimensional vector, $\{f_{s1}^{(j)}, ..., f_{sd_s}^{(j)}\}$. Thus, an image that includes k regions $r_1, ..., r_k$ would be represented by k vectors of the form:

$$f_{11}^{(j)}, ..., f_{1d_1}^{(j)}, ..., f_{q1}^{(j)}, ..., f_{qd_q}^{(j)}, \text{ for } j = 1, ...k,$$

where $f_{i1}^{(j)}, ..., f_{id_i}^{(j)}$ is the representation of the i^{th} visual feature subsets of region r_j.

Each region inherits the annotating keywords of its image. The assumption is that, if word w describes a given region r_j, then a subset of its visual features will be present in many annotated images. Thus, an association rule among them could be mined. Figure 4.3 illustrates our image representation approach.

4.1.3 Constraints Formulation

Clustering image regions in a high dimensional and sparse feature space is a hard optimization problem that is prone to many local minima. One possible approach to achieve robust results is to use partial supervision to guide the clustering process and narrow the space of possible solutions. This additional information can be under the form of labels, hints, or constraints. Supervision in the form of constraints is more practical. Typically, it consists of a set of pairs of points that must belong to the same cluster and another set of points that must belong to different clusters [71]. Unfortunately, for large datasets, this approach is not practical because the constraint generation task could be tedious. To overcome this problem, we propose a method to extract these constraints in an unsupervised way based on the relevance of the concepts annotating the training image regions. In particular, we first extract concept relevancy information based on the annotating keywords. Then, we use this information to infer a set of *ShouldNot-link* constraints.

Let r_j denote an individual region j. Every segmented region r_j inherits its image level annotation. First, we build a weighted data matrix $D_{W \times Q}$ where Q is the total number of regions extracted from all training images, and W is the size of the vocabulary (i.e number of keywords). The idea is to assign higher weights to keywords which are more "unique" in the training set, and assign lower weights to common keywords. Thus, the

(w_i, r_j) element of matrix D is defined as

$$D_{w_i,r_j} = \begin{cases} log(\frac{Q}{z_i}) & \text{if } w_i \text{ is one of the keywords annotating } r_j \\ 0 & \text{otherwise} \end{cases} .$$

where z_i is the number of image regions annotated with keyword w_i. If we define a feature space where each dimension is an image region. Then, matrix D can be viewed as a mapping of the vocabulary into the training regions feature space.

Let concept C_p be the set of keywords annotating image I_p. We define the relevance of two sets of concepts C_p and C_q, annotating images I_p and I_q, as:

$$Rel(C_p, C_q) = arg\, max_{w_{i_p} \in C_p, w_{i_q} \in C_q}(R_c(w_{i_p}, w_{i_q})) , \tag{4.1}$$

where

$$R_c(w_{i_p}, w_{i_q}) = cos(D_{w_{i_p}, r_1:r_Q}, D_{w_{i_q}, r_1:r_Q}) \tag{4.2}$$

is the cosine similarity in the regions feature space.

In (4.2), $D_{w_{i_p}, r_1:r_Q}$ is the i_p^{th} row of D and $D_{w_{i_q}, r_1:r_Q}$ is the i_q^{th} row of D.

If the relevance of two image annotations, $Rel(C_p, C_q)$, is smaller than a predefined threshold, θ, then C_p and C_q are regarded as "irrelevant" to each other and all of their corresponding image regions are considered as *ShouldNot-link*. Intuitively, this means that if two images show little concept relevancy, then it is assumed that pairs of regions within these two images are semantically different. Thus, we define the set of image region pairs that should not be assigned to the same cluster, NL, as

$$NL = \{(r_p, r_q)/r_p \in I_p \, , \, r_q \in I_q, \text{ and } Rel(C_p, C_q) < \theta\} \tag{4.3}$$

Figure 4.4 shows an example of two correlated images. The relevancy of the set of keywords annotating these two images, computed using (4.2), is shown in Table 4.1. The total relevancy of these two set of keywords is computed using (4.1) and it is equal to 1. Since this correlation value is high, the two images are considered relevant to each other. Thus, we cannot infer *ShouldNot-link* constraints between any pair of regions from these two images.

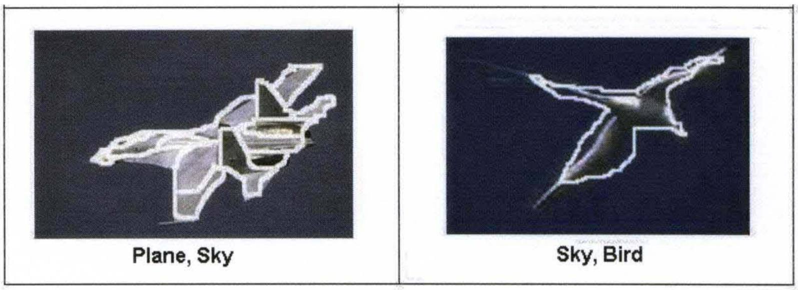
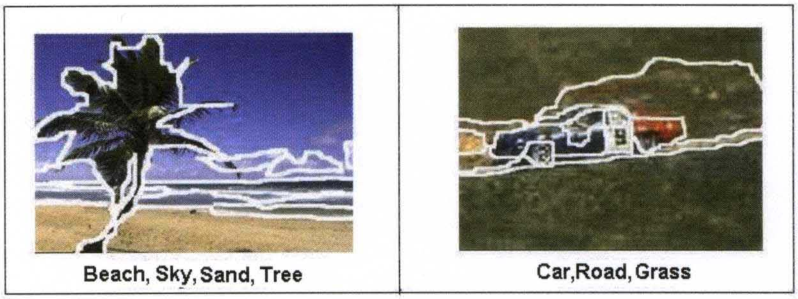

Plane, Sky
Sky, Bird

Figure 4.4: Example of two correlated images. The first image is annotated by keywords "Plane" and "Sky", while the second image is annotated by keywords "Sky" and "Bird".

Rc	Sky	Bird
Plane	0.23	0
Sky	1	0.04

Table 4.1: Relevancy between pairs of keywords annotating the images in Fig 4.4

Figure 4.5 displays two images that have weak concept relevancy. For instance, the keywords "beach", "Sky", "Sand", and "Tree" do not co-occur often with keywords like "Car", "Road" and "Grass" across the training data set. The relevancy of the set of keywords annotating these two images, computed using (4.2), is shown in Table 4.2. The total relevancy of these two sets of concepts, computed using (4.1), is 0.14. Since this concept


Beach, Sky, Sand, Tree

Car,Road, Grass

Figure 4.5: Example of two uncorrelated images. The first image is annotated by keywords “Beach”, “Sky”, “Sand” and “Tree” while the second image is annotated by keywords “car”, “road ”and “grass”.

Rc	Beach	Sky	Sand	Tree
Car	0	0.01	0	0.01
Road	0	0.013	0.003	0.02
Grass	0.004	0.07	0.003	0.14

Table 4.2: Relevancy between pairs of keywords annotating the images in Figure 4.5

relevancy value is low, these two images are considered irrelevant to each other. Thus, a set of *ShouldNot-link* constraint is created between all pairs of regions from these two images.

4.1.4 Semi-supervised Clustering and Cluster Correlation Estimation

Most existing image annotation approaches [13, 15, 18, 21] assume that clusters of image regions are independent. However, images contain multiple objects and some of them can be correlated to a certain degree. For instance, many images, would include planes in the sky. Thus, one could not assume that “Plane” and “Sky” regions are independent.

A natural solution to avoid making this independence assumption is to estimate the correlation among the regions and make use of it in the annotation process. In [19], the authors used simple inverted lists of each region cluster to estimate this correlation. Unfortunately, the boundaries between the region clusters are not well defined and using a simple inverted list to compute the dependency between them is not effective. Moreover, image region collections may contain noise and outliers since the image segmentation process cannot be accurate. To overcome these limitations, we first summarize the image region collection using clustering. Then, we use the generated membership degrees of all regions in all clusters to estimate the inter-cluster correlation.

To achieve good clustering peformance, we use two semi-supevised clustering algorithms that peform simultaneous clustering and feature weighting. The supervision information consists of a set of *ShouldNot-link* constraints and specifies that two image regions should not be assigned to the same cluster. This set of constraints is extracted in an unsupevised way as described in section 4.1.3. The first clusteing algorithm is the Semi-supervised simultaneous Clustering and Attribute Discrminiation algorithm (sSCAD) (outlined in section 2.2.3). sSCAD is a distance based algorithm that partitions the data into C clusters. It leans the center of each cluster and assigns a relevance weight to each feature subset in each cluster R_j. Let profile P_{R_j} consists of the visual features of the center, c_{R_j}, and the relevance weights for each feature subset, $v^s_{R_j}$. In addition, sSCAD assigns a fuzzy membership degree $u_{r_e R_j}$ to each region r_e in each cluster R_j.

The second algorithm we use to partition the image regions is the semi-supervised possibilistic clustering and feature subset weighting based on robust GD mixture modeling (sRULe_GDM_FSS) that we proposed in section 3.4. sRULe_GDM_FSS is a probabilistic approach that learns C

Generalized Dirichlet models that best fit the training image regions. Fo each learned model, it identifies the relevant feature subsets. In addition, this clustering algorithm generates possibilistic membership degrees $u_{r_eR_j}$ to each image region r_e in each model R_j.

After clustering the image regions (using sSCAD or sRULe_GDM_FSS), we obtain a set of region clusters, R_j , $j = 1, .., C$. Each cluster R_j includes a set of regions that share similar visual features and common keywords. Then, we use the fuzzy or possiblistic membership values to define the correlation between region clusters R_j and R_k as

$$R_{co}(R_j, R_k) = \frac{\sum_{I=1}^{N}\sum_{e=1}^{k_I}\sum_{f=1}^{k_I} min(u_{r_eR_j}, u_{r_fR_k})}{\sum_{I=1}^{N}\sum_{e=1}^{k_I}\sum_{f=1}^{k_I} max(u_{r_eR_j}, u_{r_fR_k})}. \tag{4.4}$$

In (4.4), N is the total number of images in the training set, k_I is the number of regions in image I, $u_{r_eR_j}$ is the membership degree of region r_e in cluster R_j. This could be either the fuzzy membership generated by sSCAD or the possibilistic membership generated by sRULe_GDM_FSS. In other words, region clusters R_j and R_k are highly correlated if most image regions in these clusters share similar membership values.

The poposed approach to estimate cluster correlation is illustrated by the block diagram in Figure 4.1.4.

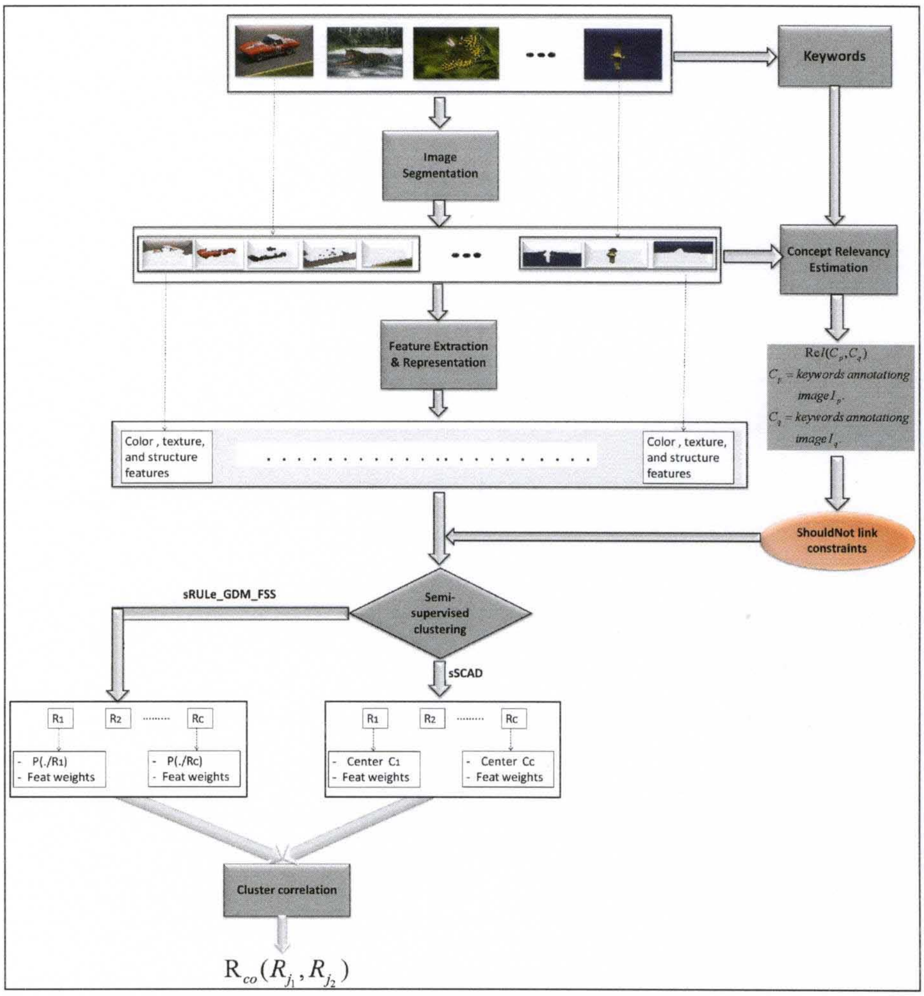

Keywords
Image Segmentation
Concept Relevancy Estimation
$Rel(C_p, C_q)$
C_p = keywords annotationg image I_p.
C_q = keywords annotationg image I_q.
Feature Extraction & Representation
Color , texture, and structure features
Color , texture, and structure features
ShouldNot link constraints
sRULe_GDM_FSS
Semi-supervised clustering
sSCAD
R1
R2
Rc
- P(./R1)
- Feat weights
- P(./Rc)
- Feat weights
R1
R2
Rc
- Center C1
- Feat weights
- Center Cc
- Feat weights
Cluster correlation
$R_{co}(R_{j_1}, R_{j_2})$

Figure 4.6: Block diagram of the proposed approach to estimate correlation between clusters of image regions

4.2 Image Annotation

In this section, we describe our approach that uses a set of training images to build a model that learns the correspondence between region clusters and keywords that annotate the training images. This correspondence would be used as the foundation to translate from one modality to another. In particular, translating visual features into keywords, i.e., image annotation. We propose two different approaches. The first one is based on a semi-naive Bayesian model. The second approach is a membership based Cross Media Relevance Model.

4.2.1 Image Regions Assignment

Given an unlabeled test image I^*, we first segment it using the same method used to segment all training images (i.e CA with color distribution). Let $\{r_1, r_2, ..., r_k\}$ be the set of regions of image I^*. For each region r_j, we extract its visual feature subsets, r_j^s, $s = 1, .., q$. Then, we assign each region to the closest region cluster. The cluster assignment depends on the clusteing algorithm used to categorize the image regions and is outlined in the following subsections.

4.2.1.1 Minimum Distance Image Region Assignment:

If the clustering algorithm used to summarize the regions of the training images is sSCAD, then, the algorithm summarizes each cluster R_l and represent it by a center c_{R_l}. First, we compare the visual features of each region r_j to the center of each region cluster R_l using

$$D(r_j, R_l) = \sum_{s=1}^{q} \frac{v^s_{R_l} \times dist(r^s_j, c^s_{R_l})}{D^s_{avg}}, \text{for } j = 1...Q, \text{ and } l = 1...C \tag{4.5}$$

In (4.5), $s = 1...q$ are the q feature subsets, $v^s_{R_l}$ is the relevance weight learned by sSCAD for feature subset s (computed using (2.50)), and $c^s_{R_l}$ is the the center of cluster R_l that takes into account only feature subset s. In (4.5), $dist(\,)$ is the partial distance between visual features of image region r_j and center of cluster R_l taking into account only feature subset s. In (4.5), D^s_{avg} is the average intra-cluster distance computed over the training data using subset s. It is used to normalize each partial distance to make all partial distances within a comparable range. This distance is computed using

$$D^s_{avg} = \frac{\sum_{j=1}^{Q} \sum_{l=1}^{C} u_{r_j R_l} \times dist(r^s_j, c^s_{R_l})}{\sum_{j=1}^{Q} \sum_{l=1}^{C} u_{r_j R_l}}, \tag{4.6}$$

Then, we assign region r_j to cluster R^* such that

$$R^* = \underset{R_l \in \{R_1...R_C\}}{arg\,min}\ D(r_j, R_l) \tag{4.7}$$

4.2.1.2 Probabilistic Image Region Assignment:

If the Generalized Dirichlet mixture modeling algorithm is used to summarize the image regions, the assignment of a new image regions will be based on the distribution of the learned models. In paticular, for each region r_j, we compute its posterior probability with respect to all models and select the one with the highest probability. In other words, we assign each image region r_j to region cluster R^* such that

$$R^* = \underset{R_l \in \{R_1,..,R_C\}}{arg\,max} \; (p(R_l/r_j))$$

where $p(R_l/r_j)$ is the posterior probability of assigning region $r_j$to cluster R_l.

4.2.2 Identifying Independent Subsets of Image Regions

Most existing image annotation approaches assume that the events of observing region clusters within an image are mutually independent once an image is selected. However, this assumption does not often hold. For instance, within the same image, the cluster of "Sky" regions can be highly correlated to the cluster of "Plane" regions. To overcome the restrictions of this "naive" assumption, our proposed annotation appoach takes into account the correlation among the region clusters of the test image. This correlation is estimated using the cluster correlation matrix R_{co} (computed using (4.4)), the membership degrees of all regions of the test image, and a greedy selection and joining (GSJ) algorithm for finding the independent subsets of region clusters.

Assume that the regions of test image I^*,$\{r_i,\, i = 1...k\}$, belong to the clusters $H = \{R_{j'},\, j' = 1...k'\}$, where $k' < k$. We should note here that subscript i refers to the i^{th} region of test image I^*, while subscript j' refers to the j'^{th} region cluster. The GSJ algorithm is described below.

Algorithm 14 The Greedy Selection and Joining algorithm (GSJ)

```
1 Initialization
  $B = \emptyset$, $S = 1$, choose $R_{j'} \in H$ randomly,
  $1 \leq j' \leq k', B_h = \{R_{j'}\}, H = H / \{R_{j'}\}$,
2 Selection step:
  Select $R_{j'} = argmax_{R_{j_{2'}} \in B_h} \sum_{R_{j_{1'}} \in B_h} |R_{co}(R_{j_{1'}}, R_{j_{2'}})|$,
  and for any $R_{j_{1'}} \in B_h$,$|R_{co}(R_{j_{1'}}, R_{j'})| > \varepsilon$,
  $\varepsilon$ is a pre-defined threshold and $R_{co}(R_{j_{1'}}, R_{j'})$ is defined in 4.4;
3 Joining step:
  If $R_{j'}$ exists and $|B_h| < t$
    $B_h = B_h \cup \{R_{j'}\}$ ,$H = H / \{R_{j'}\}$
    Go to 2;
  Else If $H \neq \phi$
    $h = h + 1, B = B \cup \{B_h\}$
    Go to 1;
  Else
    Exit
End
```

In the GSJ algorithm, t is a threshold and it is used to control the number of region clusters to be included in each independent region cluster subset.

The greedy selection and joining algorithm is thus used to decompose the clusters $H = \{R_1, R_2, ..., R_{k'}\}$, occuring in a given test image, into l independent subsets $B = \{B_1, B_2, ..., B_l\}$, where

$$\cup_{h=1}^{l} B_h = H \ , \ B_{h_1} \cap B_{h_2} = \oslash \ \forall B_{h_1}, \ B_{h_2} \in B.$$

4.2.3 Image Annotation using a Semi-naive Bayesian Approach

To annotate a test image using the maximum a posteriori (MAP) criterion, we first compute the posterior probability, $P(w_i / R_1, R_2, ..., R_{k'})$, for all keywords w_i in the dictionary. Then, we select a subset of few keywords that have the highest posterior probability.

Using Bayes rule, the posterior probability can be computed using

$$P(w_i/R_1, R_2, ..., R_{k'}) = \frac{P(R_1, R_2, ..., R_{k'}|w_i)P(w_i)}{P(R_1, R_2, ..., R_{k'})}, \tag{4.8}$$

In (4.8), $\{R_1, R_2, ..., R_{k'}\}$ are the k' region clusters to which the regions of the test image are assigned, and $P(R_1, R_2, ..., R_{k'})$ is the evidence of the observed region clusters, which serves simply as a normalizing constant.

If we assume that all regions $\{R_1, R_2, ..., R_{k'}\}$ are independent, then

$$P(R_1, R_2, ..., R_{k'}/w_i) = \prod_{j'=1}^{k'} P(R_{j'}/w_i). \tag{4.9}$$

where

$$P(R_{j'}|w_i) = \frac{vol(R_{j'}, w_i)}{vol(w_i)}. \tag{4.10}$$

In (4.10), $vol(w_i)$ is the number of images annotated with word w_i, and $vol(R_{j'}, w_i)$ is the number of images that include a region assigned to region cluster $R_{j'}$ and labeled with word w_i.

As mentioned earlier, typically the assumption that region clusters $\{R_1, R_2, ..., R_{k'}\}$ are independent may not be valid. For instance, many images, would include planes in the sky, or animals on grass. Thus, one could not assume that the "planes" and "sky" regions are independent. To overcome this limitation, we propose an alternative labeling method that does not rely on the independence assumption. First, we estimate the degree of dependency among all region clusters of the database as outlined in in section 4.1.4. Second, we use the greedy selection and joining (GSJ) algorithm, outlined in section 4.2.2, to decompose the set of region clusters, $\{R_1, R_2, ..., R_{k'}\}$, of the test image I^*, into l' independent subsets $\{B_1, B_2, ..., B_{l'}\}$. Finally, we compute the class conditional density using

$$P(R_1, R_2, ..., R_{k'}|w_i) = P(B_1, B_2, ..., B_{l'}|w_i) = \prod_{h'=1}^{l'} P(B_{h'}|w_i). \qquad (4.11)$$

In (4.11), $P(B_{h'}|w_i)$ is the probability of observing a region from subset $B_{h'}$, given a word w_i. It can be estimated using

$$P(B_{h'}|w_i) = \frac{vol(B_{h'}, w_i)}{vol(w_i)}. \qquad (4.12)$$

where $vol(w_i)$ is the number of images annotated with word w_i, and $vol(B_{h'}, w_i)$ is the number of images that include a region assigned to region clusters from subset $B_{h'}$ and labeled with keyword w_i.

The semi-naive Bayesian image annotation algorithm is summarized in Algorithm 15 and is illustrated by the block diagram in Figue 4.7.

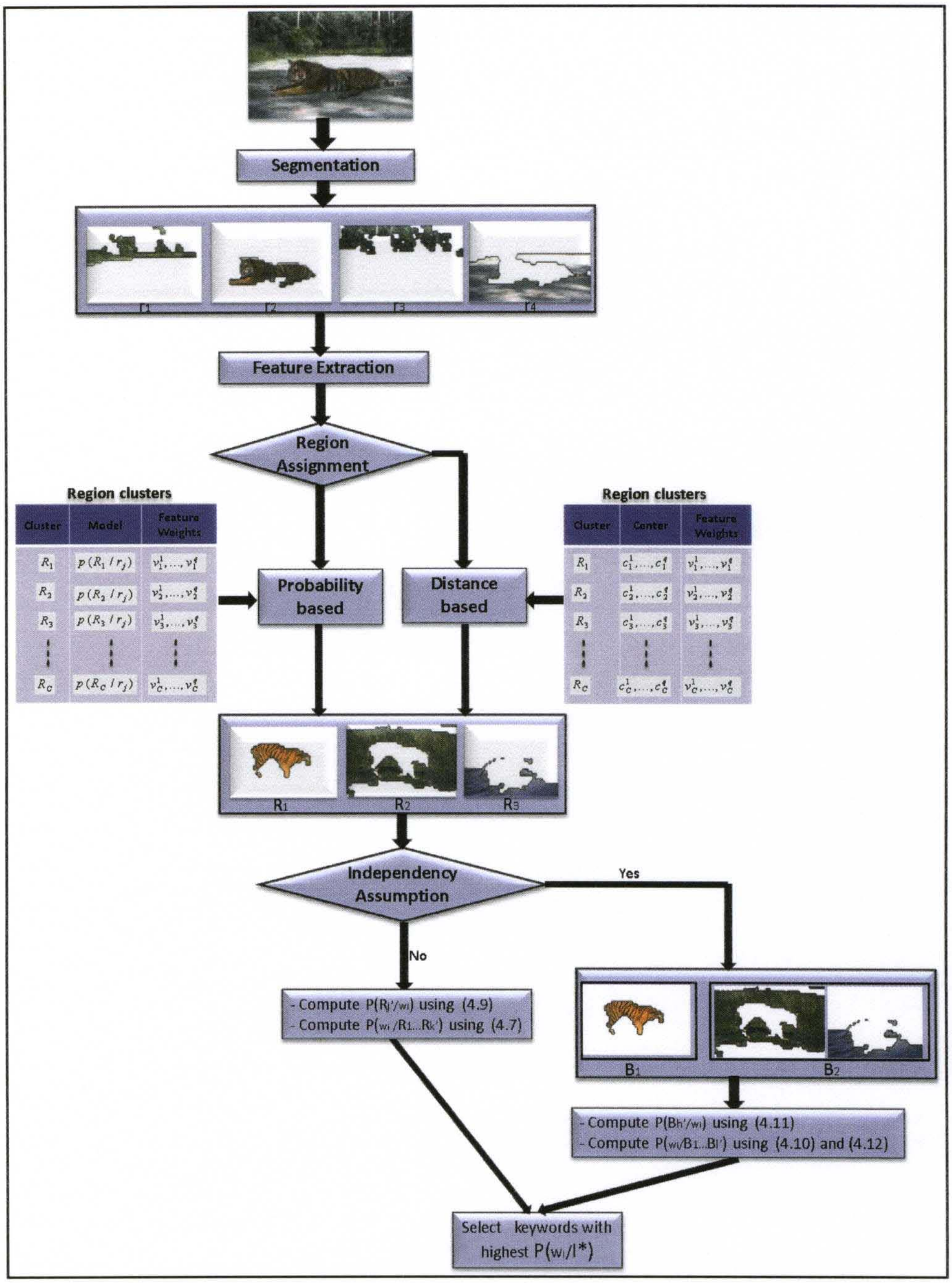

Segmentation
r1
r2
r3
r4
Feature Extraction
Region Assignment
Region clusters
Cluster
Model
Feature Weights
R_1 $p(R_1/r_j)$ $v_1^1,\ldots,v_1^q$
R_2 $p(R_2/r_j)$ $v_2^1,\ldots,v_2^q$
R_3 $p(R_3/r_j)$ $v_3^1,\ldots,v_3^q$
R_C $p(R_C/r_j)$ $v_C^1,\ldots,v_C^q$
Probability based
Distance based
Region clusters
Cluster
Center
Feature Weights
R_1 $c_1^1,\ldots,c_1^q$ $v_1^1,\ldots,v_1^q$
R_2 $c_2^1,\ldots,c_2^q$ $v_2^1,\ldots,v_2^q$
R_3 $c_3^1,\ldots,c_3^q$ $v_3^1,\ldots,v_3^q$
R_C $c_C^1,\ldots,c_C^q$ $v_C^1,\ldots,v_C^q$
R_1
R_2
R_3
Independency Assumption
Yes
No
- Compute $P(R_j'/w_i)$ using (4.9)
- Compute $P(w_i/R_1 \ldots R_k')$ using (4.7)
B_1
B_2
- Compute $P(B_h'/w_i)$ using (4.11)
- Compute $P(w_i/B_1 \ldots B_l')$ using (4.10) and (4.12)
Select keywords with highest $P(w_i/I^*)$

Figure 4.7: Block diagram of the proposed image annotation approach based on Semi-naive Bayesian Model

Algorithm 15 Image Annotation using a Semi-naive Bayesian Approach

```
For each test image $I^*$;
  Segment $I$ using (CA) algorithm (detailed in section 2.1.5);
  Assign each region of $I^*$ to a cluster. Let $H = \{R_1, R_2, ..., R_{k'}\}$ be the set of region clusters ;
  Apply GSJ to decompose $H$ into $l$ independent subsets $\{B_1, ..., B_{l'}\}$;
  For each subset $B_{h'}$and keyword $w_i$
    Compute $P(B_{h'}|w_i)$ using (4.12);
  end
  For each keyword $w_i$
    Compute $P(R_1, R_1, ..., R_{k'}|w_i)$ using (4.11);
    Compute $P(w_i|I^*)$ using (4.8);
  end
  Label $I^*$ with few keywords that have the highest $P(w_i|I^*)$.
End
```

4.2.4 Image Annotation using Membership based Cross Media Relevance Model

In the membership based CMRM model, we assume that for a given un-annotated image I^*, there exists an underlying probability distribution (denoted as $P(.|I^*)$) of all possible region clusters and keywords that could appear in image I^*. As in the Bayesian approach, we start by segmenting the image I^* into k regions $\{r_1, ..., r_k\}$, and assigning each region to one of the region clusters. Let $\{R_1, R_2, ..., R_{k'}\}$ be the region clusters to which regions $\{r_1, ..., r_k\}$ are assigned. The image annotation goal is to estimate the probability of observing keyword w_i given the test image I^*, i.e.,

$$P(w_i|I^*) \simeq P(w_i R_1, R_2, ..., R_{k'}) \tag{4.13}$$

Since $P(R_1, R_2, ..., R_{k'})$, the evidence of the observed region clusters, serves simply as a normalizing constant, calculating $P(w_i|R_1, R_2, ..., R_{k'})$ is equivalent to calculating the joint probability $P(w_i, R_1, R_2, ..., R_{k'})$. Since the test image representation $\{R_1, R_2, ..., R_{k'}\}$ does not contain any keyword, it is not possible to use the maximum-likelihood estimator. Instead, we use the training set of annotated images, τ, to estimate the joint probability of observing the keyword w_i and the region clusters $\{R_1, R_2, ..., R_{k'}\}$ in I^*. That is,

$$P(w_i|I^*) \simeq P(w_i, R_1, R_2, ..., R_{k'}) = \sum_{I \in \tau} P(I)P(w_i, R_1, R_2, ..., R_{k'}|I)\,. \tag{4.14}$$

The prior probability $P(I)$ is kept uniform over all images in τ .

Using the assumption that words and region clusters are generated independently given a training image I, $P(w_i, R_1, R_2, ..., R_{k'})$ can then be computed using

$$P(w_i, R_1, R_2, ..., R_{k'}) = \sum_{I \in \tau} P(I)P(w_i|I) \prod_{j'=1}^{k'} P(R_{j'}|I) \tag{4.15}$$

The posterior probabilities $P(w_i|I)$ and $P(R_{j'}|I)$ are estimated by smoothed maximum likelihood. In particular, the probability of drawing word w_i from image I is given by:

$$P(w_i|I) = \alpha \frac{vol(I, w_i)}{|I|} + (1-\alpha) \frac{\sum_{I=1}^{N} vol(I, w_i)}{|\tau|}, \tag{4.16}$$

where $vol(I, w_i)$ denotes the actual number of times the keyword w is used to annotate image I (usually 0 or 1, since the same word is rarely used multiple times for the same image), $|I|$ stands for the aggregate count of all words occurring in image I, and $|\tau|$ denotes the total size of the training set. In (4.16), the term $\sum_{I=1}^{N} vol(I, w_i)$ represents the total number of times

w is used to annotate images in the training set τ.

The computation of the probability of drawing a region cluster $R_{j'}$ from image I depends on wether we assume that the clusters of regions are independent or not. For independent clusters, this pobability can be computed using

$$P(R_{j'}|I) = \beta \frac{\sum_{j=1}^{k_I} u_{r_j R_{j'}}}{|I|} + (1-\beta) \frac{\sum_{I=1}^{N} \sum_{j=1}^{k_I} u_{r_j R_{j'}}}{|\tau|}, \tag{4.17}$$

where K_I is the number of image regions in image I, and the term $\sum_{j=1}^{k_I} u_{r_j R_{j'}}$ represents the sum of the membership degrees of all regions of image I in cluster $R_{j'}$. These memberships could be the fuzzy membership produced by the sSCAD algorithm or the possibilistic membership produced by sRULe_GDM_FSS. Similarly, $\sum_{I=1}^{N} \sum_{j=1}^{k_I} u_{r_j R_{j'}}$ is the cumulative sum of the membership degrees of all regions in cluster $R_{j'}$. In (4.16) and (4.17), the smoothing parameters α and β determine the degree of interpolation between the maximum likelihood estimates and the background probabilities for the words and the regions respectively .

Without the independence assumption, we first use the GSJ algorithm to map the $\{R_1, R_2, ..., R_{k'}\}$ region clusters to l' independent subset of clusters $\{B_1, B_2, ..., B_{l'}\}$. Then, we rewrite (4.15) as

$$P(w_i, R_1, R_2, ..., R_{k'}) = P(w_i, B_1, B_2, ..., B_{l'}) = \sum_{I \in \tau} P(I) P(w|I) \prod_{h'=1}^{l'} P(B_{h'}|I) \tag{4.18}$$

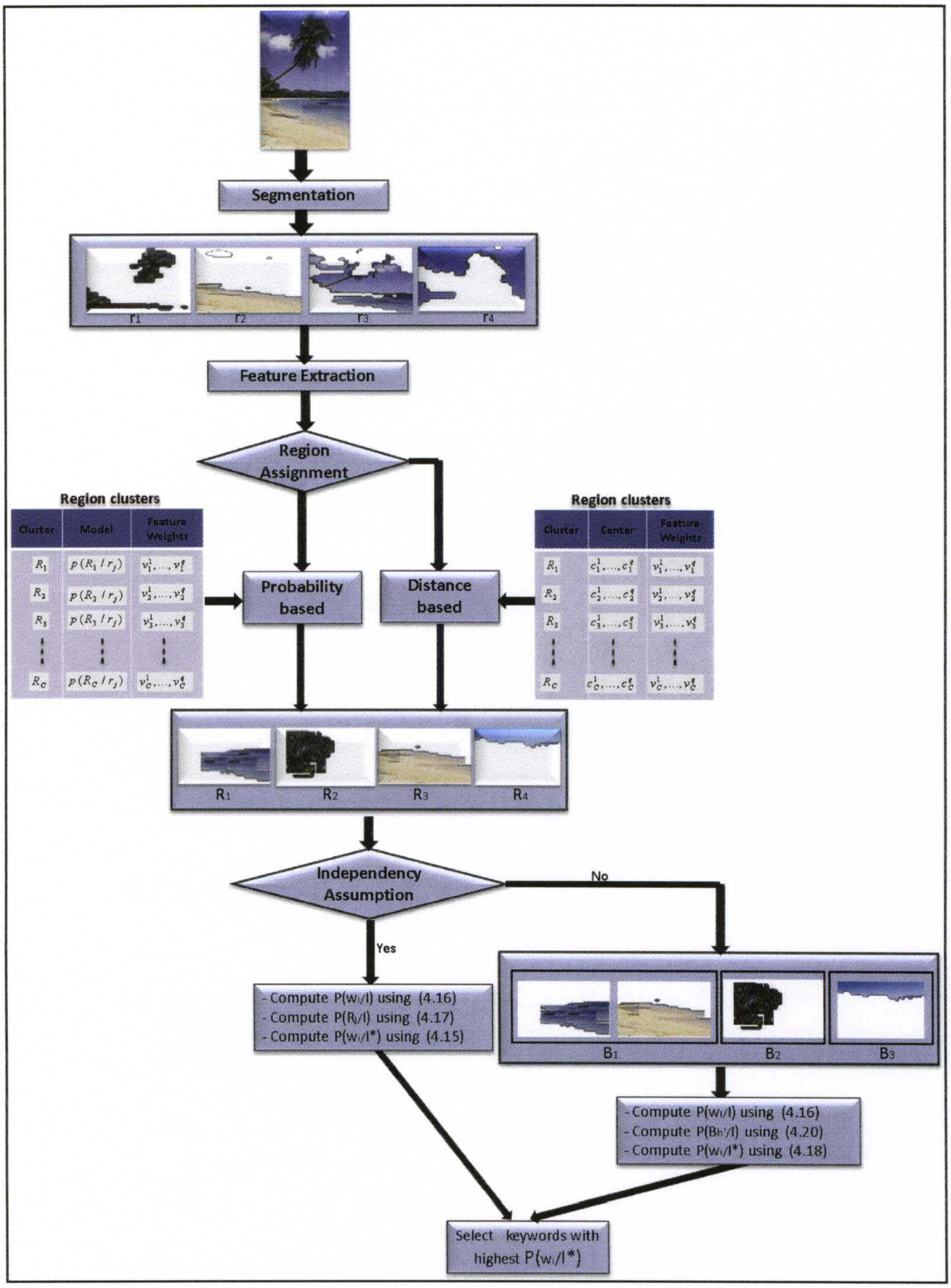
The probability of drawing word w_i from image I, i.e. $P(w_i|I)$, is still computed using (4.16), and $P(B_{h'}|I)$ is computed using

$$P(B_{h'}|I) = \beta \frac{\sum_{j=1}^{k_I} max_{R_{j'} \in B_{h'}}(u_{r_j R_{j'}})}{|I|} + (1-\beta) \frac{\sum_{I=1}^{N} \sum_{j=1}^{k_I} max_{R_j \in B_{h'}}(u_{r_j R_{j'}})}{|\tau|}. \tag{4.19}$$

The term $\sum_{j=1}^{k_I} max_{R_{j'} \in B_{h'}}(u_{r_j R_{j'}})$ represents the sum of the maximum membership degrees of image I regions to the elements of subset $B_{h'}$. As in (4.19), these memberships can be either fuzzy or possibilistic depending on the clustering algorithm used to group the image regions. The term $\sum_{I=1}^{N} \sum_{j=1}^{k_I} max_{R_j \in B_{h'}}(u_{r_j R_{j'}})$ is the cumulative sum of the maximum membership degrees to the subset B_h elements of all region in the training set.

Equations (4.16) - (4.19) provide a process for approximating the probability distribution $P(w_i|I)$ underlying a given training image I. We generate automatic annotations for unlabeled test images by first estimating $P(w_i|I^*)$ and then selecting few keywords that have the highest probability.

The membership based Cross Media Relevance Model based image annotation algorithm is outlined in Algorithm 16 is illustrated in the block diagram in Figure 4.8.


Segmentation
r1
r2
r3
r4
Feature Extraction
Region Assignment
Region clusters
Probability based
Distance based
Region clusters
R1
R2
R3
R4
Independency Assumption
No
Yes
- Compute P(wi/I) using (4.16)
- Compute P(Rj/I) using (4.17)
- Compute P(wi/I*) using (4.15)
B1
B2
B3
- Compute P(wi/I) using (4.16)
- Compute P(Bh'/I) using (4.20)
- Compute P(wi/I*) using (4.18)
Select keywords with highest P(wi/I*)

Region clusters

Cluster	Model	Feature Weights
R_1	$p(R_1 / r_j)$	$v_1^1,\dots,v_1^q$
R_2	$p(R_2 / r_j)$	$v_2^1,\dots,v_2^q$
R_3	$p(R_3 / r_j)$	$v_3^1,\dots,v_3^q$
$\vdots$	$\vdots$	$\vdots$
R_C	$p(R_C / r_j)$	$v_C^1,\dots,v_C^q$

Region clusters

Cluster	Center	Feature Weights
R_1	$c_1^1,\dots,c_1^q$	$v_1^1,\dots,v_1^q$
R_2	$c_2^1,\dots,c_2^q$	$v_2^1,\dots,v_2^q$
R_3	$c_3^1,\dots,c_3^q$	$v_3^1,\dots,v_3^q$
$\vdots$	$\vdots$	$\vdots$
R_C	$c_C^1,\dots,c_C^q$	$v_C^1,\dots,v_C^q$

Figure 4.8: Block diagram of the proposed image annotation approach using Membership based Cross Media Relevance Model

Algorithm 16 Image annotation using Membership based Cross Media Relevance Model

```
For each test image $I^*$;
  Segment $I^*$ using (CA) algorithm (detailed in section 2.1.5);
  Assign each region of $I$ to a cluster. Let $H = \{R_1, R_2, ..., R_{k'}\}$ be the set of region clusters ;
  Apply GSJ algorithm to decompose $H$ into $l$ independent subsets $\{B_1, ..., B_{l'}\}$;
  For each subset $B_{h'}$, keyword $w_i$
    Compute $P(B_{h'}|I)$ using (4.19);
  end
  For each keyword $w_i$
    Compute $P(w_i|I^*)$ using (4.15);
  end
  Label $I^*$ with few keywords that have the highest $P(w_i|I^*)$.
End
```

4.3 Experimental Results

A range of experiments were performed to asses the strengths and weaknesses of the proposed approaches. We use a subset of the Corel Stock Photo library [69]. This is a collection of high-resolution color photographs grouped according to specific themes into CDs of 100 images each. The Corel subset used for this experiment consists of 9,264 images. Each image in the training set is manually labeled by 1 to 7 keywords. A total of 97 keywords were used which provide a global description of the images and are not explicitly associated with specific regions. A list of these keywords is provided in Table 4.3.

Figure 4.9 plots the occurrence frequencies of each keyword. The frequencies are sorted in decreasing order. The plot shows that some common words, such as "sky", "grass", and "tree" have a high occurrence rate, whereas more specific words, such as "whale", "giraffe", and "raccoon" appear seldom.

antelope
ape
badger
balloon
beach
bear
bird
bison
boat
branch
bridge
building
bus
bush
butterfly
cactus
car
castle
cat
cheetah
cherrytree
chicken
chipmunk
city
cliff
cloud
column
cow
crocodile
deer
desert
dirt
dog
donkey
elephant
fence
field
fire
fish
flower
footballfield
forest
fox
frog
giraffe
goat
grapes
grass
ground
groundhog
helicopter
hippo
horse
leaves
leopard
lion
lizard
llama
manatee
mane
miscellaneous
monkey
mountain
mushroom
night
opossum
owl
people
person
pig
plane
porcupine
rabbit
raccoon
rhino
road
rock
sand
sculpture
seal
sheep
skunk
sky
smoke
snake
snow
squirrel
stone
sun
tiger
train
tree
turtle
wall
water
whale
wolf
zebra

Table 4.3: List of words used to label the training images

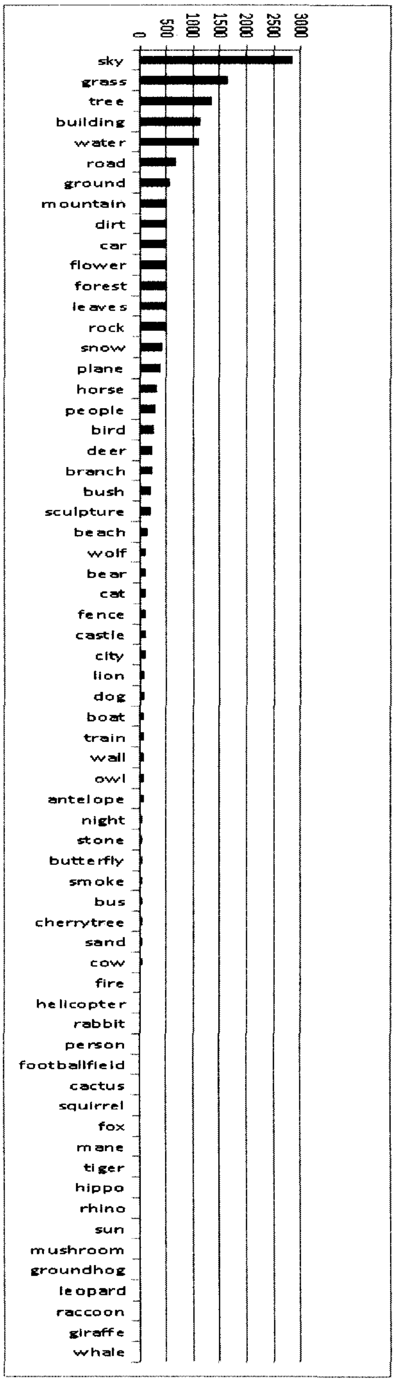


Figure 4.9: frequency of the keywords used to label the set of training images

4.3.1 Image Segmentation

The images have been coarsely segmented by clustering the color distributions. The Competitive Agglomeration (CA) algorithm (described in section 2.1.5) was used to cluster each image into an optimum number of regions. We fixed the initial number of clusters to 10, and the parameters η and τ in (2.23) to 0.01 and 10, respectively. Segmentation of all images resulted in a total of 40,051 regions. Examples of segmented images are provided in Figure 4.10 where each region is represented by the average cluster color.

4.3.2 Feature Representation

All extracted regions are represented by various features that represent color, texture, structure, and shape information. In our experiment, we use mainly standard MPEG-7 features [58] as they are commonly used in CBIR platforms [56, 57]. Each image region is characterized by the following set of features:

4.3.2.1 RGB Color Histogram:

The R, G and B color channels in each region are quantized into 64 bins, and represented by a 64−dimensional histogram. Each color histogram feature is normalized such that its components sum to 1.

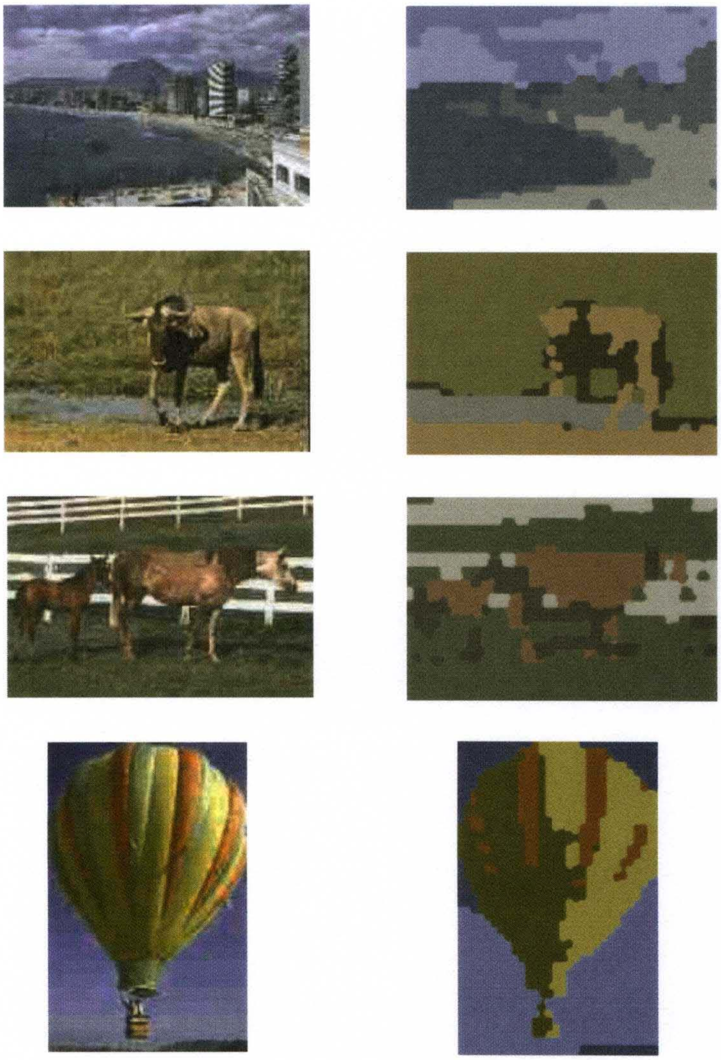


Figure 4.10: Example of images from the training set segmented using the CA clustering algorithm

4.3.2.2 HSV Color Moments:

Each region is mapped to the HSV color space. Then, the mean, standard deviation and skewness of the H, S, and V components are computed and used as features. This feature subset is represented by a 9-dimensional vector.

4.3.2.3 LUV Color Moments:

Each region is mapped to the LUV color space. Then, the mean, standard deviation and skewness of the L, U, and V components are computed. This feature subset is represented by a 9-dimensional vector.

Both the HSV and LUV color Moments feature subsets are normalized to have zero mean and unit standard deviation.

4.3.2.4 Edge Histogram:

A variant of the MPEG-7 edge histogram descriptor (EHD) [58] is used to represent the frequency and directionality of edges within each image region. Simple edge detector operator are used to detect edges and group them into five categories: vertical, horizontal, diagonal, anti-diagonal and non-edge. The EHD includes five bins corresponding to the frequencies of the five categories.

4.3.2.5 Wavelet Texture Features:

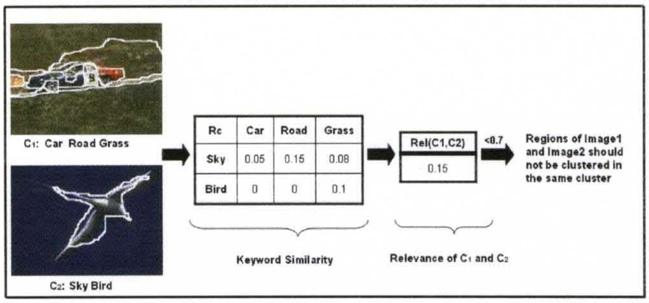
Each region is analyzed at different frequencies with different resolutions. The Haar filter bank is used to decompose the image into three scales, resulting in a total of ten components that include the approximation at scale three, and horizontal, vertical, and diagonal components at the three scales. Then, the mean and standard deviation are computed for each component. This makes the features vector 20-dimensional. This feature subset is normalized to have zero mean and unit standard deviation.

4.3.2.6 Shape Feature:

For each region, the eccentricity, orientation, area, solidity, and extent are computed. Eccentricity is computed by first finding an ellipse with the same second-moments as the region and then computing the ratio of the distance between the foci of the ellipse and its major axis length. The orientation is defined as the angle in degrees between the x-axis and the major axis of the ellipse containing the same second-moments as the region. The area is defined as the actual number of pixels within the region. The solidity is defined as the proportion of pixels in the convex hull that are also in the region. The extent is defined as the proportion of the pixels in the bounding box of the regions that are also in the region. It is computed as the area divided by the area of the bounding box.

4.3.3 Constraint Formulation

As detailed in section 4.1.3, we infer partial supervision information for the clustering algorithm from the training data itself. In our experiment, we set the threshold used to decide wether two annotations are relevant to each other or not (θ in (4.3)) to 0.7. Thus, if the relevance of two annotations $Rel(C_1, C_2)$ is smaller than 0.7, then, their corresponding image regions



Rc	Car	Road	Grass
Sky	0.05	0.15	0.08
Bird	0	0	0.1

Figure 4.11: Constraint Formulation Example

are regarded as "irrelevant" to each other and should not be grouped in the same cluster.

Figure 4.11 illustrates an example of constraint formulation. In this figure, one image was labeled by three words: "Car", "Road", and "Grass". The second image was labeled by two words: "Sky" and "Bird". The relevance between pairs of these keywords $R_c(w_i, w_j)$ is shown in Figure 4.11. The total correlation computed using equation (4.1) is 0.15 . Since this is below the threshold, these two images are considered irrelevant and *ShouldNot-link* constraints are created between all inter-image region pairs.

Using this approach we considered 1931 image pairs to be "irrelevant" to each other. Thus, we obtained a total of 11,702 *ShouldNot-link* relations between inter-image regions that were used to guide the clustering process.

We should note here that our unsupervised approach to construct the set of *ShouldNot-link* constraints is not accurate. There will be cases where similar and relevant image regions would be included in the set of *ShouldNot-link* constraints. However, this should not be a problem. In fact, our semi-supervised clustering algorithm takes these as suggestions and will not necessarly enforce them.

4.3.4 Image Region Clustering

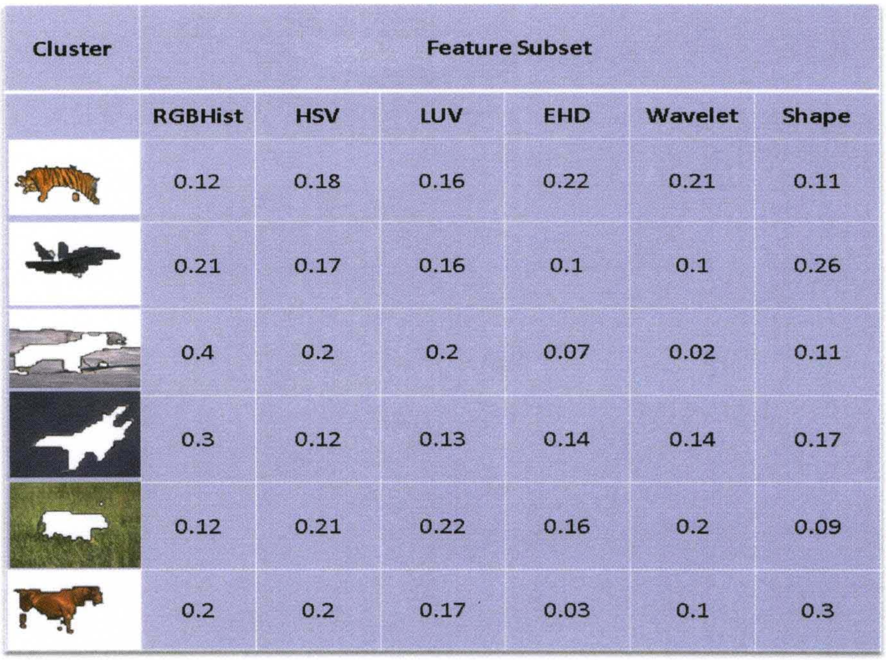
4.3.4.1 Minimum-distance based Clustering

The 40,051 image regions encoded by the 6 feature subsets were clustered using sSCAD (detailed in section 2.2.3). Since this algorithm requires the specification of the number of clusters, we fix C to 380 (value found by sRULe_GDM_FSS). The experimental parameters of this step are reported in Table 4.4.

Constant	**Constant Name**	**Constant Value**
Number of feature subset	K	7
Maximum cluster number	C	380
Constraint term scaling	a	5
Fuzzifier	m	1.1

Table 4.4: Values of the constants used in the clustering process using sSCAD

The clustering algorithm was relatively successfull in partitioning the data into homogeneous categories. Figure 4.12 displays representative regions (closest region to the cluster center) for six sample clusters. In addition, to partitionning the data into homogeneous clusters, sSCAD identified relevance weights for each feature subset in each cluster. The feature relevance weights for the 6 clusters shown in Figure 4.12 are shown in Table 4.5. For instance, for the "horse" and "tiger" clusters, the shape and color features are more relevant than the other visual features. For the "grass" cluster, the texture and color are the most relevant features. For the "sky" cluster, the regions are blueish and consistently smooth, but the shape is less consistent. For this cluster, the color and texture features are more relevant



Cluster	Feature Subset					
	RGBHist	HSV	LUV	EHD	Wavelet	Shape
	0.12	0.18	0.16	0.22	0.21	0.11
	0.21	0.17	0.16	0.1	0.1	0.26
	0.4	0.2	0.2	0.07	0.02	0.11
	0.3	0.12	0.13	0.14	0.14	0.17
	0.12	0.21	0.22	0.16	0.2	0.09
	0.2	0.2	0.17	0.03	0.1	0.3

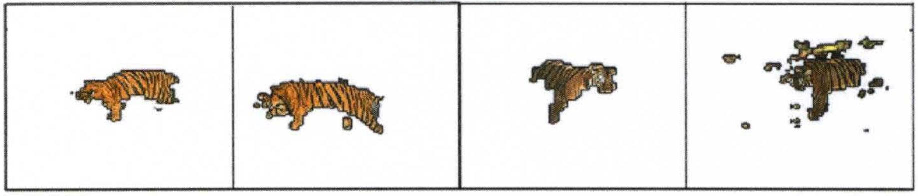
Table 4.5: Feature relevance weights for the 6 clusters displayed in Fig. 4.12

than shape. For the "plane" cluster, the shape of the regions is the most consistant and the corresponding feature is relatively relevant.

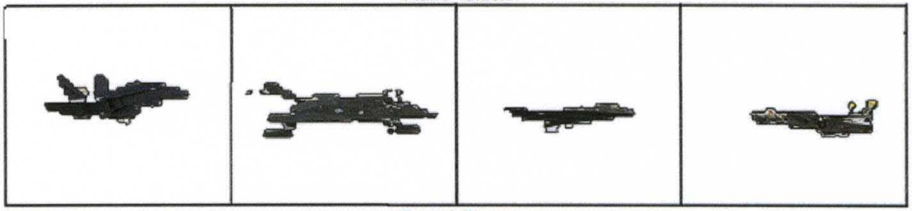
4.3.4.2 Probabilistic Clustering

An alternative to using sSCAD to cluster the image regions is to use the semi-supervised possibilistic clustering and feature subset weighting algorithm based on robust Geneneralized Dirichlet mixture Model (sRULe_GDM_FSS) that we proposed in section 3.4. For this algorithm, we set the fuzzyfier m to 1.2 and estimate the scale parameter η_j for each cluster j as suggested in [82]. We use the following initialization scheme. First, we partition the image region collection using the fuzzy C-means [67]. Then, we use the method of moments (MM) [112] to obtain initial beta distribution pa-

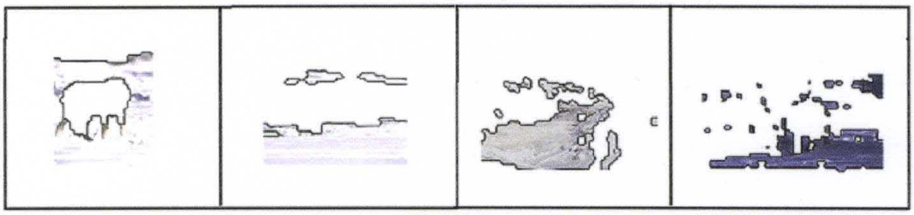
"Tiger" Cluster



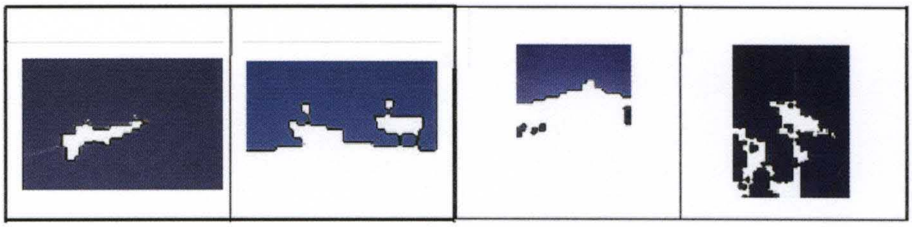
"Plane" Cluster



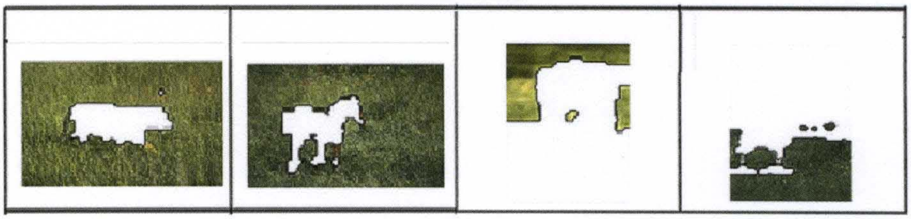
"Snow" Cluster



"Sky" Cluster



"Grass" Cluster



"Horse" Cluster

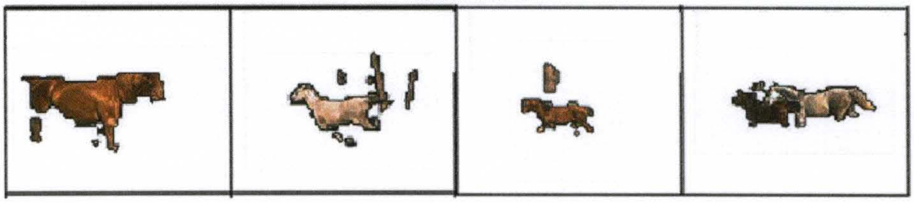


Figure 4.12: Representative regions of six sample clusters obtained by sS-CAD

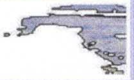
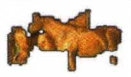
Cluster	Feature Subset					
	RGBHist	HSV	LUV	EHD	Wavelet	Shape
	0.12	0.18	0.16	0.22	0.21	0.11
	0.09	0.15	0.15	0.41	0.14	0.06
	0.5	0.2	0.12	0.07	0.02	0.09
	0.33	0.14	0.14	0.1	0.12	0.17
	0.17	0.17	0.22	0.15	0.2	0.09
	0.13	0.2	0.17	0.1	0.1	0.3

Table 4.6: Feature relevance weights for the 6 clusters displayed in Fig. 4.13

rameters for each cluster. We overspecify the number of clusters to 450. sRULe_GDM_FSS converged after 210 iterations and the number of clusters reduced to 380. For each iteration of sRULe_GDM_FSS, we update θ and λ using (*3.45*) and (*3.46*) for 2 iterations.

Figure 4.13 displays representative regions for six sample clusters. Similarly to sSCAD, sRULe_GDM_FSS identified relevance weights for each feature subset in each cluster. The feature relevance weights for the 6 clusters in Figure 4.13 are shown in Table 4.13. For instance, for the "plane" and "horse" clusters, the shape and color features are more relevant than the other visual features. For the "grass" cluster, texture and color are the most relevant features. For the "tiger" cluster, the texture of the regions is the most consistant and the corresponding feature is relatively relevant.

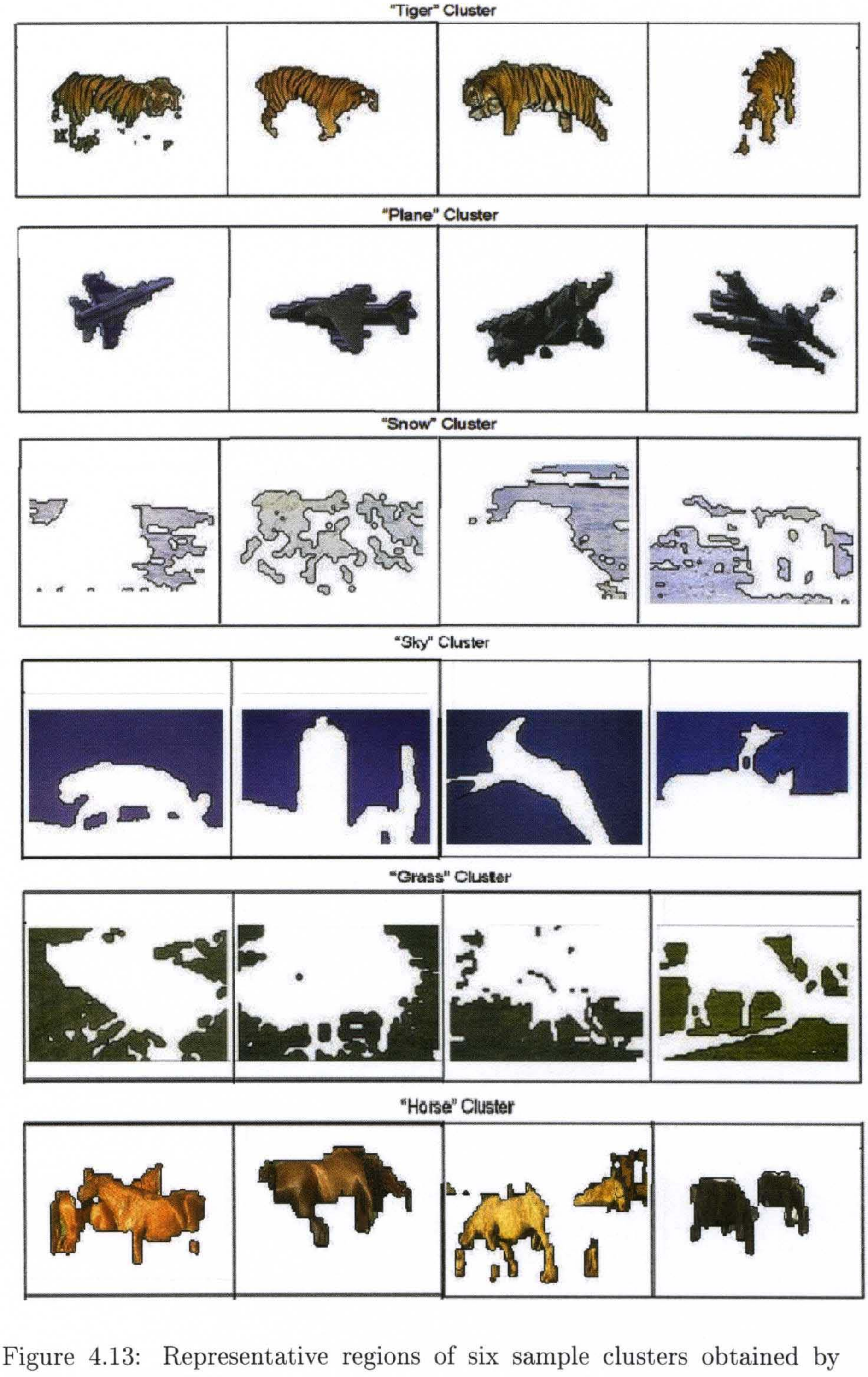


Figure 4.13: Representative regions of six sample clusters obtained by sRULe_GDM_FSS

Both sSCAD and sRULe_GDM_FSS algorithms achieve reasonable image region clustering. In particular, both algorithms performed well for hard cases where regions with similar visual colors such as “deer” and “horse” or “sky” and “water”, but different semantics were assigned to different clusters. This was possible due to the extracted constraints. For instance, deer” and “horse” annotations are irrelevants to each other based on their correlation ocross the training set. This irrelevancy yields *ShouldNot-link* constraints between regions annotated by “horse” and regions annotated by “deer”.

By analyzing and comparing the content of the different clusters generated by the 2 clustering approaches, we observed that sSCAD splits many categories over several clusters. For instance, several clusters were used for the “flower” and “butterfly” categories. This is because these categories have large intra-cluster color variations and do not necessarly have spherical shapes in the high dimensional feature space. Moreover, the image region collection includes regions that are unique and have different visual appearance than the majority of the regions in all categories. sSCAD does not detect these noise regions. This affects the clustering accuracy and yields non-homogeneous clusters. On the other hand, sRULe_GDM_FSS uses possibilistic memberships and can detect these noise regions and limit their influence on the estimated Generalized Dirichlet distributions.

4.3.5 Image Annotation

To validate the proposed annotation methods and compare them to existing methods, we use the following performance measures.

Precision and Recall: Precision and recall, which are the most common metrics for evaluating different information retrieval systems, are also widely adopted for evaluating the effectiveness of automatic annotation approaches. For our application, we use a per-image precision and recall. In particular, for each test image, precision is defined as the ratio of the number of words that are correctly predicted to the total number of words used for annotation. Similarly, recall is defined as the ratio of the number of words that are correctly predicted to the number of words in the ground-truth or manual annotations. Formally, these measures are computed using

$$R(I) = \frac{m_c(I)}{m_T(I)}, \tag{4.20}$$

and

$$P(I) = \frac{m_c(I)}{m_c(I) + m_w(I)} \tag{4.21}$$

where $m_c(I)$ is the number of keywords predicted correctly in annotating image I, $m_T(I)$ is the total number of words used to label image I, and $m_w(I)$ is the number of irrelevant keywords predicted for image I. The per-image precision and recall values are averaged over the whole set of test images to generate the mean precision and recall values.

F-measure: In general, probabilistic models involve smoothing maximum likelihood probabilities, and include smoothing parameters. Thus, there is an implicit tradeoff between recall and precision, and both of these measures should be used simultaneously for setting the model parameters. A single comprehensive measure that combines both terms is the F-measure. The F-measure is defined as the harmonic mean of precision and recall, i.e.,

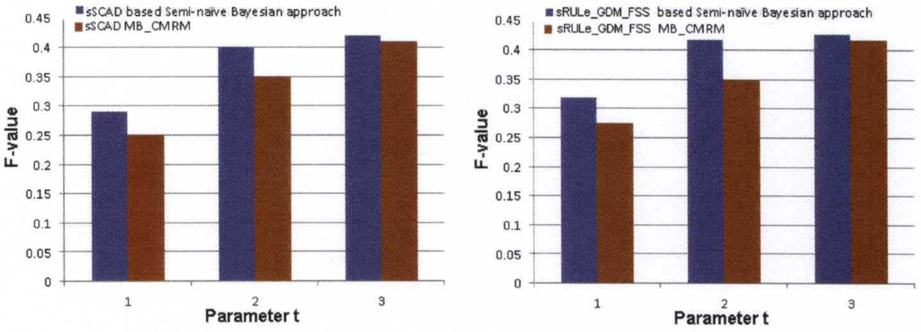
$$F(I) = \frac{2\,.\,R(I)\,.\,P(I)}{R(I)\ +\ P(I)} \tag{4.22}$$

We use the F-measure only during training to select the optimal parameters. As a single quantity, it cannot illustrate how recall and precision change with respect to each other. Thus, to evaluate and compare the performance during the testing phase, we use both recall and precision.

Word accuracy: Word accuracy is defined as the ratio of the number of times a given word w_i is used in correct annotation to the total number of times w_i is used in annotating all images.

We use a 4-fold cross validation approach where we divide the 9,267 images into 4 subsets of equal sizes. For each fold, we use 75% of the data for training and learning the model parameters and the remining 25% for testing. The final results are reported as the average of the 4 folds.

To limit the level of dependency between region clusters to which the regions of a test image are assigned, we carry out experiments by setting the parameter t used in the GSJ algorithm (described in section 4.2.2) to 1, 2 and 3. The F-values obtained by varying parameter t from 1 to 3 for both clustering algorithms are reported in Figures 4.14(a)-(b).



(a) (b)

Figure 4.14: Effect of the parameter t used in the GSJ on the annotation results using (a) sSCAD algorithm, and (b) sRULe_GDM_FSS algorithm.

As it can be seen, proposed image annotation approaches are more effective as we increase t. This is because a larger value of t can capture the co-occurrence information of region clusters better. However, a larger value of t requires a considerably more computational time and a larger training set to evaluate the dependencies between the larger set of region clusters. As the increase of the performance measure from $t = 2$ to $t = 3$ is quite small, we use $t = 2$ for the rest of experiments. We also set the value of the threshold ε used in the GSJ algorithm to 0.1.

The smoothing parameters α and β in equations (4.16) and (4.17) can influence the performance of the fuzzy membership based Cross Media Relevance Model. In particular, α determines how much we rely on word frequency in an individual annotation to approximate the underlying model of an image. A larger α causes the probability distributions of the models to move closer to the distribution of the background. As a result of smoothing out the individual frequencies, the model becomes strongly biased by the most frequent words. In annotation, this would have the effect of annotating most images with the same frequent words.

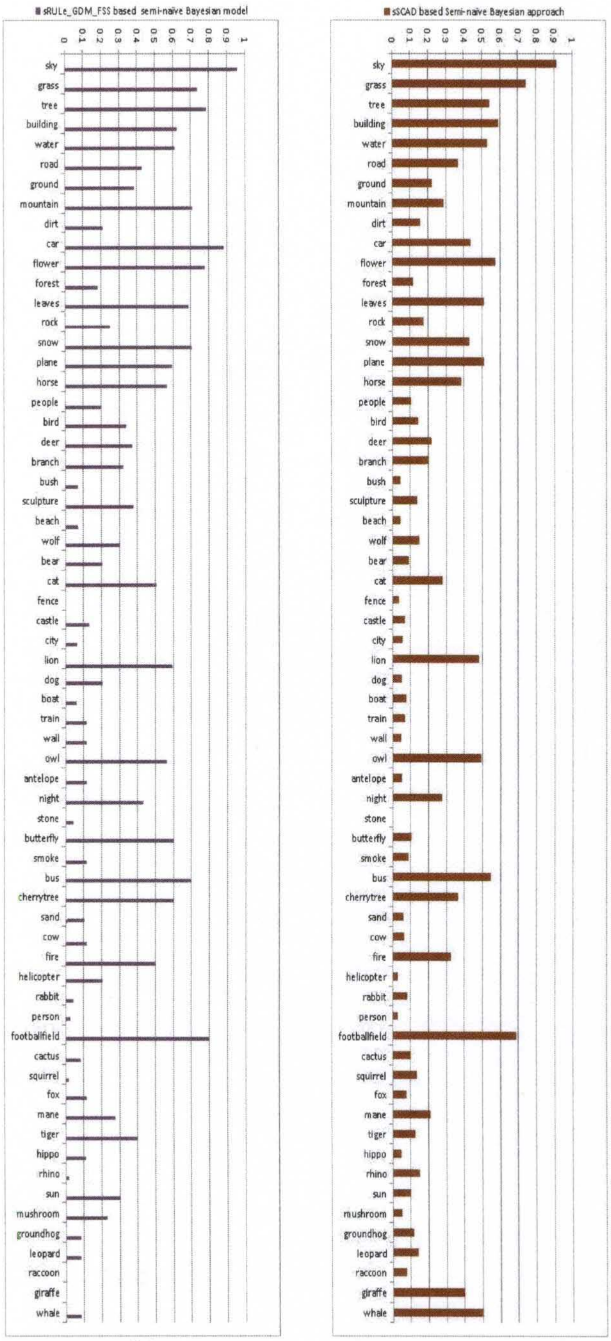
During training we do not examine all possible combinations of the α and β parameter values exhaustively in order to find the optimum values. We simply set the first one to a certain value and then vary the second one to find a local maximum. The optimal values for the smoothing parameter β was found to be 0.8, and the optimal value for the smoothing parameter α was found to be 0.1.

Figure 4.15 displays the individual keyword accuracy when five words are used to label an image using Semi-naive Bayesian Model. As expected, the accuracy is higher for most frequent keywords. For instance, frequent keywords such “sky”, “grass” and “tree” have the highest accuracy values. Reasonable accuracy is also obtained for less frequent keywords. For instance, keywords such “footballfield” and “Bus” are most of the time correctly predicted although they are relatively rare in the data set. This could be explained by the fact that images originally labeled by these keywords are easy to segment and the low-level features extracted from the resulting regions are very discriminative. This helps to learn the correspondence between visual features and textual keywords.

Figure 4.16 shows the individual keyword accuracy when five words are used to annotate an image using the membership based Cross Media Relevance Model. As it can be seen, the membership based CMRM approach, when used with sSCAD or sRULe_GDM_FSS algorithms, yields reasonable accuracy values for most frequent keywords. The CMRM approach with fuzzy membership learned by sSCAD cannot learn efficiently the association between clusters and less frequent keywords such “bear”, bird”, and “train”. On the other hand, the CMRM with possibilistic membership learned by sRULe_GDM_FSS performs slightly better and less frequent words such “mushroom” and “butterfly” have higher accuracy.

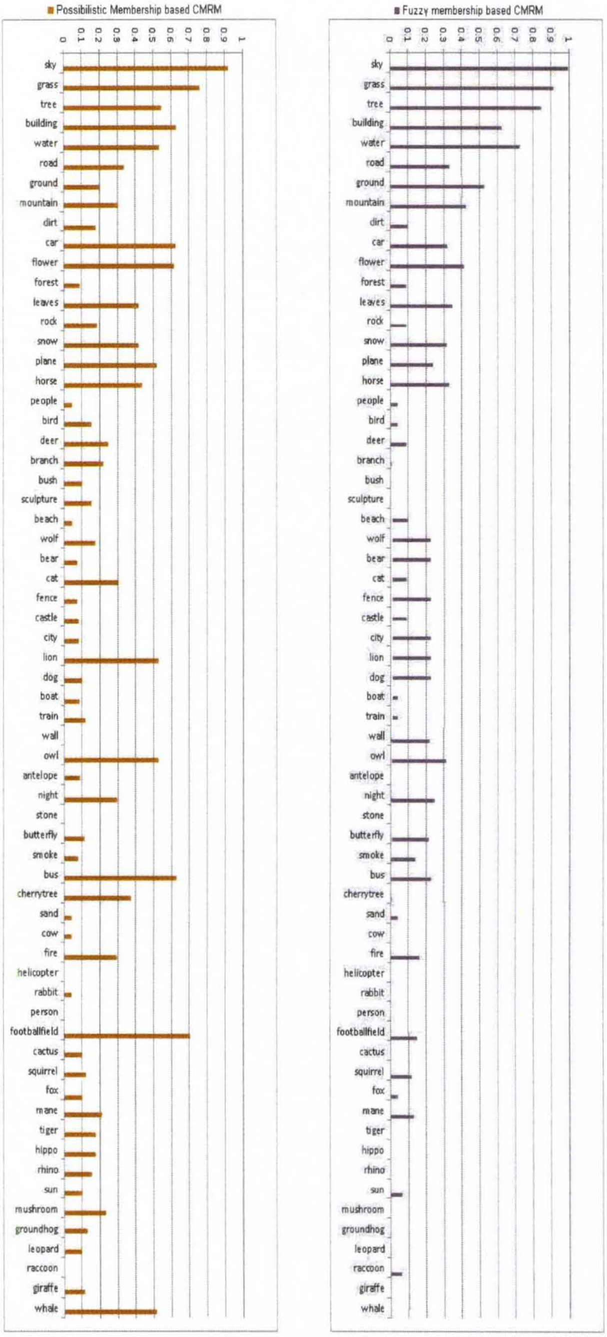
The goal of image annotation is to obtain high per-image precision and recall values, and high accuracy values for all words. For most frequent words, the accuracy values are reasonable, especially when we use five words to label the test images. This means that these words used correctly most of the time. However, there are some words with low accuracy which means that although they are not predicted often, the predictions are usually correct.

In Table 4.7, 4.8, and 4.9 we report the average accuracy, precision and recall of the proposed image annotation approaches. From Table 4.7, one



(b) (a)

Figure 4.15: Word accuracy obtained using semi-naive Bayesian Model with (a) sSCAD algorithm, (b) sRULe_GDM_FSS algorithm.



(b) **(a)**

Figure 4.16: Word accuracy obtained using membership based CMRM with (a) sSCAD algorithm, (b) sRULe_GDM_FSS algorithm.

	1 word	3 words	5 words	7 words
sSCAD based semi-naive Bayesian model	12%	20%	26%	29%
Fuzzy membership based CMRM	11%	16%	24%	26%
sRULe_GDM_FSS based semi-naive Bayesian model	15%	28%	34%	37%
Possibilistic membership based CMRM	14%	19%	24%	29%

Table 4.7: Average accuracy of the proposed images annotation approaches when 1, 3, 5, and 7 words are used to annotate each image

can notice that while there is a drastic increase in accuracy when one, three, or five words are used to label each image, there is only a slight increase when the number of words is increased to seven. Similarly, Tables 4.8 and 4.9 indicate that 5 words provides a reasonable compromise between precision and recall. Thus, we use five annotating keywords to validate the proposed image annotation approaches.

Figures 4.17 and 4.18 present samples of image annotation obtained using the 4 proposed image annotation methods. As it can be seen, all proposed image annotation approaches achieved good performance. However, methods based on sRULe_GDM_FSS clustering slightly outerform the approaches based on sSCAD clustering.

	1 word	3 words	5 words
sSCAD based semi-naive Bayesian model	68%	46%	37%
Fuzzy membership based CMRM	67%	42%	35%
sRULe_GDM_FSS based semi-naive Bayesian model	73%	50%	42%
Possibilistic membership based CMRM	72%	45%	37%

Table 4.8: Average per-image precision of the proposed image annotation methods when 1, 3, and 5 words are used to annotate each image

	1 word	3 words	5 words
sSCAD based semi-naive Bayesian Model	18%	38%	55%
Fuzzy membership based CMRM	19%	40%	51%
sRULe_GDM_FSS based semi-naive Bayesian Model	24%	39%	57%
Possibilistic membership based CMRM	22%	41%	53%

Table 4.9: Average per-image recall of the proposed image annotation methods when 1, 3, and 5 words are used to annotate each image

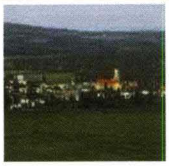
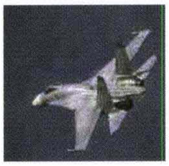
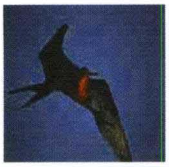
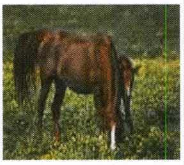
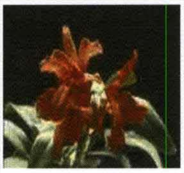
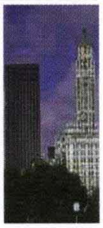
Images	Original Annotation	sSCAD and semi-naive Bayesian model	Fuzzy membership based CMRM	sRULe_GDM_FSS and semi-naive Bayesian model	Possibilistic membership based CMRM
	Sky, Mountain, Building, Grass	Sky, Mountain, Tree, Grass	Sky, Grass, Tree, Building	Sky, Mountain, building, Grass	Sky, Mountain, Grass, Building
	Sky, Plane	Sky, Smoke, Plane	Sky, Plane, Bird	Sky, Smoke, Plane	Sky, Plane, Bird
	Sky, Bird	Sky, Bird, Plane	Sky, Bird, Plane	Sky, Bird, Plane	Sky, Bird, Plane
	Horse, Grass, Flower	Horse, Grass, Flower	Horse, Grass, Flower	Horse, Grass, Flower	Horse, Grass, Flower
	Flower, Leaves	Flower, Leaves, Grass	Flower, Leaves, Sky	Flower, Leaves, Sky	Flower, Leaves, Sky
	Sky, Building	Sky, Building, Tree	Sky, Building, Tree	Sky, Building, City	Sky, Building, Tree

Figure 4.17: Image Annotation Samples (1)

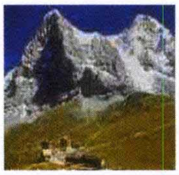
Images	Original Annotation	sSCAD and semi-naive Bayesian model	Fuzzy membership based CMRM	sRULe_GDM_FSS and semi-naive Bayesian model	Possibilistic membership based CMRM
	Beach, People, Sky, Water	Sky, Beach, People, Cliff	Sky, Water, Grass, Building	Beach, City, Sky, Water	Beach, Building, Sky, Water
	Wolf, Tree, Snow	Wolf, Snow, Forest	Snow, Tree, Rock	Wolf, Tree, Snow	Wolf, Tree, Snow
	Car, Fence, Road	Car, Road, dirt	Road, Building, Water	Car, Fence, Road	Car, Fence, City
	Boat, Cliff, Sky, Water	Sky, Water, Boat, Tree	Sky, Water, Grass, Tree	Boat, People , Sky, Water	Boat, Tree, Sky, Water
	Plane, Sky, Road	Plane, Sky, Road	Plane, Water, Building	Plane, Sky, Road	Plane, Sky, Road
	Sky, Mountain, Grass	Sky, Mountain, Rock	Sky, Mountain, Water	Sky, Mountain, Grass, Snow	Sky, Mountain, Grass, Snow

Figure 4.18: Image Annotation Samples (2)

A further analysis of the results and a comparison between the annotation based on sSCAD and sRULe_GDM_FSS revealed that there are two main reasons behind incorrect annotation:

Bad segmentation: Some segmented regions are not homogeneous and may include parts of different objects. These regions represent noise points and outliers in the image region collection. The sSCAD algorithm is more sensitive to this issue as it does not indentify and discard noise points. These regions can affect the clustering partition and the overall annotation performance.

Model assumptions: In our experiment, we used the Euclidean distance with sSCAD to cluster the image region collection. That is, sSCAD seeks spherical clusters. For the sRULe_GDM_FSS, we assume that the region clusters fit a Beta distribution. However, many image region categories have large intra-cluster color, texture, and structural variations and do not fit any specific model. The sSCAD based approach is more sensitive to this limitation as the spherical assumption is more restrictive.

4.3.6 Empirical Comparison with State-of-the-art Methods

The performance of the proposed sRULe_GDM_FSS methods is assessed against three other methods: Two of them, the CMRM (described in section 2.3.1) and the constrained K-means based (described in section 2.2.2) image annotation are global and assign labels to the entire image. The third one, Image to Word Transformation based image annotation (described in 2.3.1) is local and assigns labels to image regions.

The K-means (described in section 2.1.2) and the pair-wise constrained K-means (described in section 2.2.2) algorithms are used as clustering algorithm for the CMRM and the semi-naive Bayesian model based annotation methods, respectively, to summarize the image regions extracted from the training set. On the other hand, our approach, sRULe_GDM_FSS is based on simultaneous clustering and feature weighting. Moreover, it relies on the generated possibilistic membership to compute the cross media relevancy.

Figure 4.19 compares the precision/recall curves of the different algorithms averaged over the four cross validation sets. As it can be seen, the two proposed annotation methods outperform the three other method significantly.

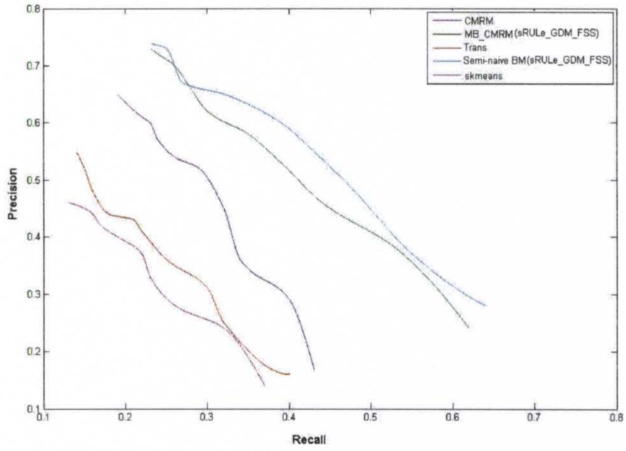


Figure 4.19: Comparison of the Precision vs. Recall curves of the two proposed sRULe_GDM_FSS based methods and three other existing methods

In figure 4.20, we compare the average accuracy for each word individually. This is basically the number of times this word appears in the top five annotation labels. For the most frequent words, like 'building', 'grass', 'sky', 'tree' and 'water' (see Figure 4.9) , the five methods have satisfactory and comparable accuracy. However, for less frequent words, the sRULe_GDM_FSS based semi-naive Bayesian model based annotation

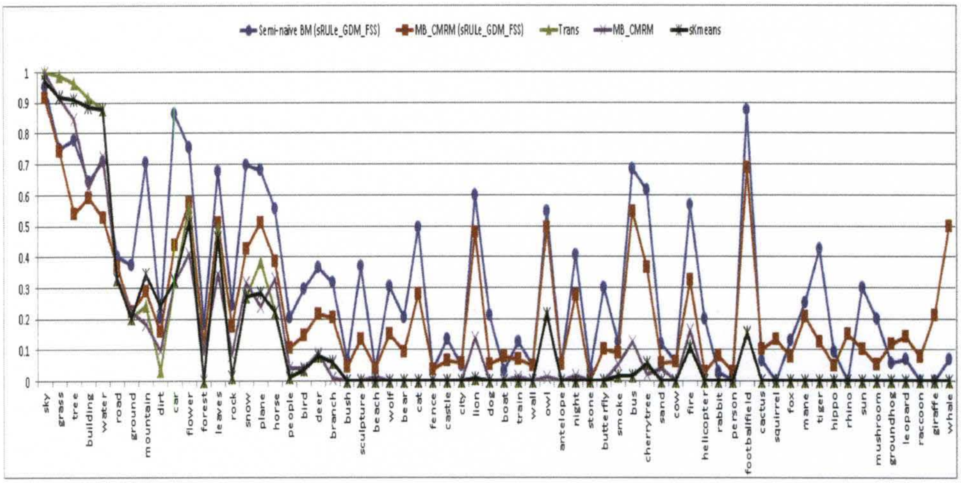


Figure 4.20: Comparison of the accuracy of the most frequent words using 5 different annotation methods

	Trans	CMRM	sKmeans	Semi-naive BM (sRULe_ GDM_ FSS)	MB_CMRM (sRULe_ GDM_ FSS)
Average accuracy	0.11	0.16	0.18	0.37	0.32

Table 4.10: Average accuracy of 5 image annotation methods

(light blue curve)) and the possibilistic membership based Cross Media Relevance Model (red curve) outperform the other methods.. This confirms the precison/recall analysis of Figure 4.19. The same conclusion can also be reached by comparing the results in Table 4.10 which displays the overall accuracy when averaged over all keywords.

In Figure 4.21, we display the plot of the keyword average accuracy versus the number of labeling keywords for each method. For all methods, the average accuracy increases linearly as we increase the number of labeling

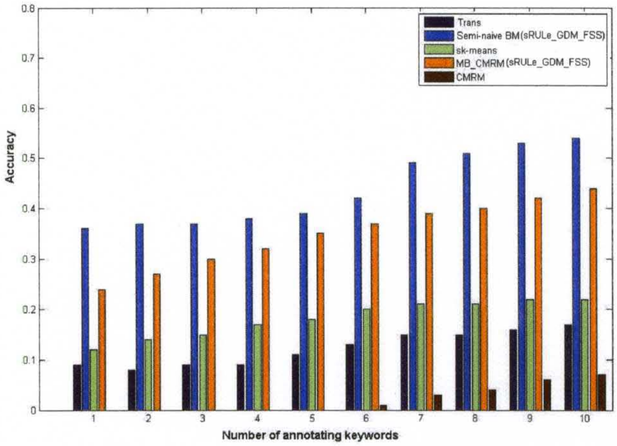


Figure 4.21: Word average accuracy vs. number of labeling keywords

keywords which makes the empirical comparison of these methods with respect to their average accuracy independent from the final number of labeling keywords.

The empirical comparison indicates that automatic labeling can be reliable for keywords that are frequent in the training dataset. However, for infrequent words, the precision of all methods is usually low. The results also show that overall, the sRULe_GDM_FSS based methods outperform the other three methods. This is particularly true for keywords that are not frequent across the entire database.

CHAPTER 5

IMAGE RETRIEVAL BASED ON MULTI-MODAL SIMILARITY PROPAGATION

In this chapter, we propose an image retrieval framework based on multi-modal similarity propagation. We use the proposed image annotation, outlined in the previous chapter, to augment the standard content based image retrieal approach in an attempt to improve the retrieval performance. In particular, we explore the correlation between visual and textual features to capture their semantics and discover the intrinsic similarity of images. First, we use our image annotation approaches to generate labeling keywords. Then, these keywords are used as additional features in a content-based image retrieval system.

The proposed CBIR framework is outlined in figure 5.1. This system can be conceptually separated into two main components: One is offline and consists of preprocessing, segmenting images, extracting features, annotation,

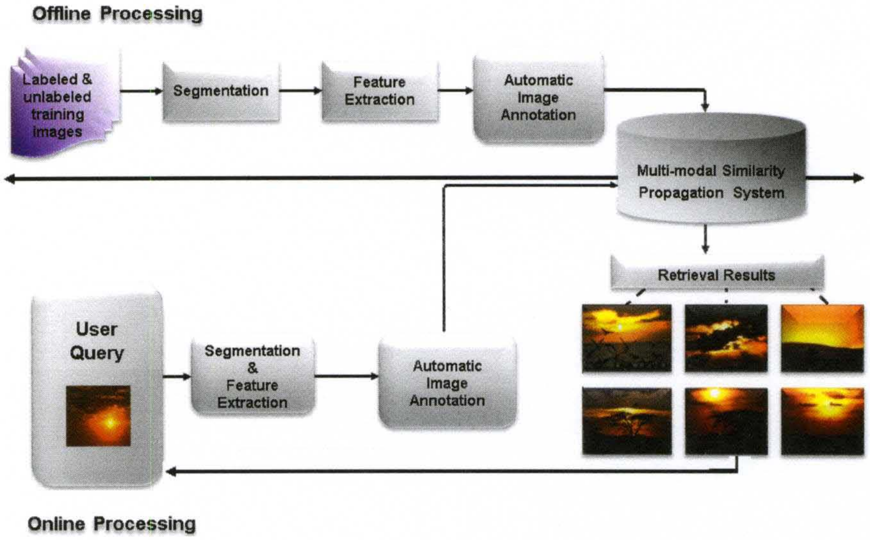

Offline Processing
Labeled & unlabeled training images
Segmentation
Feature Extraction
Automatic Image Annotation
Multi-modal Similarity Propagation System
Retrieval Results
User Query
Segmentation & Feature Extraction
Automatic Image Annotation
Online Processing

Figure 5.1: Block Diagram of the Proposed Image Retrieval System

and indexing the image database. The second one is online and consists of the user's interaction with the system to query and retrieve images.

In the off-line step, first, color, texture, and shape features are extracted from each image to represent its visual content. Second, each image is segmented into homogeneous regions and annotated by few keywords using our image annotation algorithms. These keywords are then encoded into a textual feature vector linked to the corresponding images by inverted tables.

The retrieval part starts with the user providing an example image through a graphical user-interface. First, the query image is segmented into homogeneous regions based on color feature. Then, low-level features are extracted from each region, and the image is annotated by few keywords. Finally, using a multi-modal similarity propagation algorithm, the system retrieves images that are semantically similar to the query image.

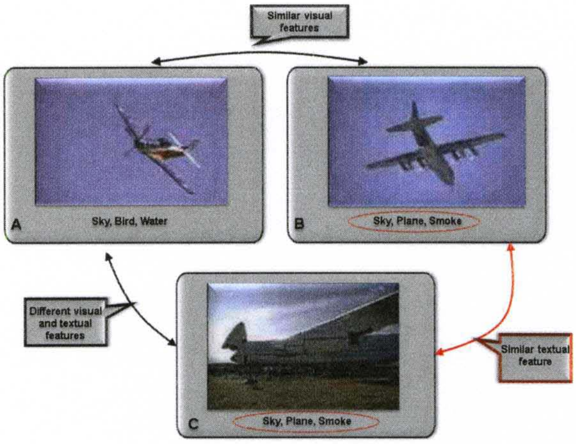


Figure 5.2: Illustration of the Multi-modal similarity propagation. Images A and B have similar visual features. Images B and C have similar textual features. Images B would be used as a bridge to enhance the similarity between images A and C.

The proposed image retrieval approach, based on multi-modal similarity propagation, relies on the assumption that two images are similar if they are annotated with some common keywords. Similarly, if two images are labeled with two different, but similar annotations, then they are similar to a certain degree. The goal of the similarity propagation is to enhance or reduce the similarity between two objects (image or text). In other words, the similarity of two images will be increased (or decreased) if they are annotated by similar (or dissimilar) keywords. Figure 5.2 illustrates a case of similarity enhancement. This figure displays three images and their annotating keywords. As it can be seen, image B has similar visual features to image A and has similar textual features to image C. Our proposed approach uses image B as a bridge to enhance the similarity between image A and image C.

The key features of the proposed image retrieval include:

1. Instead of treating the image annotation as an additional feature, we use

an iterative approach to explore the mutual reinforcement between visual and textual features. This approach avoids any bias that may be introduced by the feature encoding, and provides a better combination of the visual and textual modalities.

2. Since similarity is the variable that is propagated between different modalities, our approach can handle the sparse and high dimensional feature space quite effectively. The intra- and inter-object similarities are refined during the process. This in turn can reduce both false positives and false negatives and can reveal intrinsic similarities at the semantic level.

3. Our approach is an iterative process. The effect of each retrieval modality is propagated to its related modalities in each iteration, and the interactions inside and across the sets of relational data are explored during the mutual reinforcement.

4. Fundamentally, this approach can be seen as a non-linear combination of different retrieval modalities that can exploit the relationships among different data types more effectively, and use these relationships to discover implicit semantic object similarities.

5.1 Inter-modality Similarity Propagation

The basic idea of iterative similarity propagation is that object similarities, with respect to different modalities, can mutually influence each other. Figure 5.3 illustrates this process. In this figure, V and T denote two heterogeneous object spaces of visual (v) and textual (t) features, and v_i and t_j represent instances of these features. In particular, v_i is the visual features of one of the images in the database and t_j is one of the keywords used

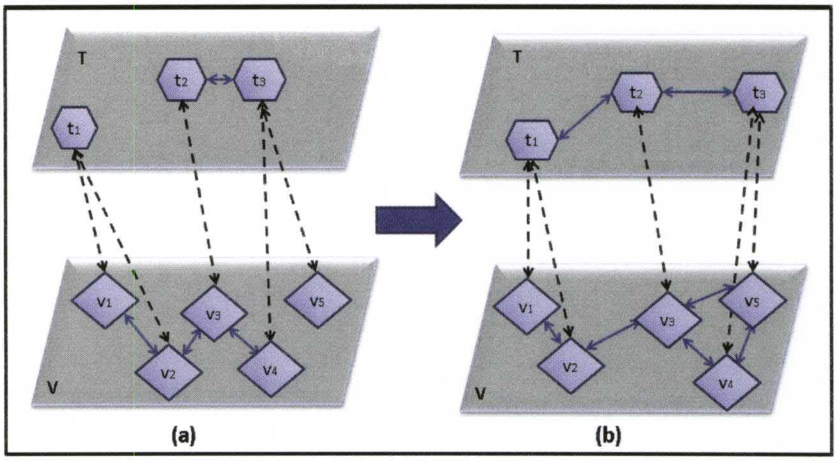


Figure 5.3: Illustration of the Similarity propagation process

to annotate the database. The dotted lines represent links among modalities (i.e. inter-object relation). The solid lines represent intra-modality similarities. The length of these lines is proportional to the degree of similarity. Figure-5.3(a) displays the original object relationships. As it can be seen, in the visual space V , images I_1, I_2, I_3 and I_4 are similar to each other, but are dissimilar to image I_5. However, using the textual space, one can deduce that images I_4 and I_5 are semantically similar since they are annotated by the same keyword t_3. Moreover, images I_4 and I_5 may be semantically similar to image I_3. This is because I_3 is annotated by keyword t_2 which is similar to keyword t_3. On the other hand, images I_2 and I_3 which appear to be visually similar are not semantically similar because their annotating keywords are not related.

Figure-5.3(b) displays the relationship among the different objects after propagating the inter-modal similarity. As it can be seen, in the visual space V, image I_5 became similar to images I_3 and I_4. On the other hand, the similarity between images I_2 and I_3 was reduced. Similarly, in the textual space, a weak similarity between t_1 and t_2 has been established because these keywords annotate images that are visually similar.

Formally, let $K_{M\times M}$ denote the similarity matrix between pairs of images in the database based on the visual content of the images. Let $G1_{N_1\times N_1}$ and $G2_{N_2\times N_2}$ denote the intra-object similariy matrices of two set of image annotations in the textual feature space. In our system, these two sets of annotations are provided by the two annotation approaches proposed in chapter 4. Let $\hat{K}_{M\times M}$, $\hat{G}1_{N_1\times N_1}$ and $\hat{G}2_{N_2\times N_2}$ denote the intra-object similarity matrices after similarity propagation. Let $Z1_{M\times N_1}$be the link matrix between images and text annotations obtained using the first image annotation approach and $Z2_{M\times N_2}$be the link matrix between image space and the second set of text annotations obtained using the second image annotation approach. Note that the transpose of the matrices, i.e. Z_1' and Z_2', are the link matrices from the textual space to the visual space.

The Z_1 and Z_2 matrices are constructed using

$$Z_{ij} = \begin{cases} 1/\theta_i, & if\, image\, I_i\, is\, annotated\, with\, keyword\, a_j \\ 0 & otherwise \end{cases} \tag{5.1}$$

In (5.1), θ_i is the number of non zero elements in the i^{th} row of Z.

The similarity propagation is an iterative process that updates the matrices $\hat{K}$, $\hat{G}_1$ and $\hat{G}_2$ in each iteration using

$$\begin{aligned} \hat{K} &= \alpha K + (1-\alpha)\lambda\left[Z_1\hat{G}_1Z_1' + Z_2\hat{G}_2Z_2'\right] \\ \hat{G}_1 &= \beta_1 K + (1-\beta_1)\lambda Z_1'\hat{K}Z_1 \\ \hat{G}_2 &= \beta_2 K + (1-\beta_2)\lambda Z_2'\hat{K}Z_2 \end{aligned} \tag{5.2}$$

where α, β_1 and β_2 are constants, and λ is a decay factor used to ensure that the propagated similarities are weaker than the original similarities.

In (5.2), $Z_1\hat{G}_1Z_1'$ and $Z_2\hat{G}_2Z_2'$ are the inter-object similarity matrices, i.e. the part of the intra-object similarities G_1 and G_1 that are propagated from the textual space T to the visual space V through the links Z_1 and Z_2. Similarly, $Z_1'\hat{K}Z_1$ and $Z_2'\hat{K}Z_2$ are the inter-object similarity matrices, i.e. the parts of intra-object similarities K which are propagated from space V to space T through the links Z_1 and Z_2.

The equations in (5.2) combine both the intra- and inter-object similarities and address the mutual reinforcement in an iterative way. It is based on the idea that similarities based on one modality should be affected by the similarity with respect to other modalities. It is basically a non-linear combination method that takes into account the different degrees of similarity from different modalities. This non-linear method is needed because the interactions among the objects are most probably non-linear and cannot be achieved by a simple linear combination method.

5.1.1 Convergence of the Algorithm

In this section, we prove that the system of equations in (5.2) converges. Let $\hat{K}^{(n)}$, ${\hat{G}_1}^{(n)}$ and ${\hat{G}_2}^{(n)}$ denote the matrices $\hat{K}$, $\hat{G}_1$ and $\hat{G}_2$ at the n^{th} iteration. Assume that the process starts with propagation from space V

to space T. Then,

$$\begin{aligned}\hat{K}^{(n)} - \hat{K}^{(n-1)} &= \left(\alpha K + (1-\alpha)\lambda\left[Z_1 \hat{G_1}^{(n)} Z_1' + Z_2 \hat{G_2}^{(n)} Z_2'\right]\right) \\ &- \left(\alpha K + (1-\alpha)\lambda\left[Z_1 \hat{G_1}^{(n-1)} Z_1' + Z_2 \hat{G_2}^{(n-1)} Z_2'\right]\right) \qquad (5.3) \\ &= (1-\alpha)\lambda\left[Z_1(\hat{G_1}^{(n)} - \hat{G_1}^{(n-1)})Z_1' + Z_2(\hat{G_2}^{(n)} - \hat{G_2}^{(n-1)})Z_2'\right],\end{aligned}$$

$$\begin{aligned}\hat{G_1}^{(n)} - \hat{G_1}^{(n-1)} &= \left(\beta_1 K + (1-\beta_1)\lambda Z_1' \hat{K}^{(n-1)} Z_1\right) - \left(\beta_1 K + (1-\beta_1)\lambda Z_1' \hat{K}^{(n-2)} Z_1\right) \\ &= (1-\beta_1)\lambda Z_1' \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) Z_1, \qquad (5.4)\end{aligned}$$

and,

$$\begin{aligned}\hat{G_2}^{(n)} - \hat{G_2}^{(n-1)} &= \left(\beta_2 K + (1-\beta_2)\lambda Z_2' \hat{K}^{(n-1)} Z_2\right) - \left(\beta_2 K + (1-\beta_2)\lambda Z_2' \hat{K}^{(n-2)} Z_2\right) \\ &= (1-\beta_2)\lambda Z_2' \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) Z_2. \qquad (5.5)\end{aligned}$$

Then, if we substitute $\hat{G_1}^{(n)} - \hat{G_1}^{(n-1)}$ and $\hat{G_2}^{(n)} - \hat{G_2}^{(n-1)}$ in (5.3) by their expressions in (5.4) and (5.5), we get

$$\begin{aligned}\hat{K}^{(n)} - \hat{K}^{(n-1)} &= (1-\alpha)\lambda^2\left[(1-\beta_1)Z_1 Z_1' \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) Z_1 Z_1'\right. \\ &+ \left.(1-\beta_2)Z_2 Z_2' \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) Z_2 Z_2'\right]. \qquad (5.6)\end{aligned}$$

Let $\Phi_1 = (1-\alpha)(1-\beta_1)\lambda^2$, $\Phi_2 = (1-\alpha)(1-\beta_2)\lambda^2$, $\Lambda_1 = Z_1 Z_1'$ and $\Lambda_2 = Z_2 Z_2'$. Then, eq(5.6) can be rewritten as

$$\begin{aligned}\hat{K}^{(n)} - \hat{K}^{(n-1)} &= \Phi_1 \Lambda_1 \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) \Lambda_1 + \Phi_2 \Lambda_2 \left(\hat{K}^{(n-1)} - \hat{K}^{(n-2)}\right) \Lambda_2 \\ &= \Phi_1^{n-1} \Lambda_1^{n-1} \left(\hat{K}^{(1)} - \hat{K}^{(0)}\right) \Lambda_1^{n-1} + \Phi_2^{n-1} \Lambda_2^{n-1} \left(\hat{K}^{(1)} - \hat{K}^{(0)}\right) \Lambda_2^{n-1} \\ &= \Phi_1^{n-1} \Lambda_1^{n-1} \left(\hat{K}^{(1)} - K\right) \Lambda_1^{n-1} + \Phi_2^{n-1} \Lambda_2^{n-1} \left(\hat{K}^{(1)} - K\right) \Lambda_2^{n-1} (5.7)\end{aligned}$$

According to the definitions of Z_1 and Z_2 given in (5.1) , $\Lambda_{1_{ij}}$,$\Lambda_{2_{ij}} \leq 1 \, \forall i, j$. Hence, we have $\underset{n\to\infty}{lim} \Lambda_1^{n-1} = 0$ and $\underset{n\to\infty}{lim} \Lambda_2^{n-1} = 0$. Also, $\left(\hat{K}^{(1)} - K\right)$ is constant, $\Phi_1 < 1$ and $\Phi_2 < 1$. Thus $\hat{K}^{(n)} - \hat{K}^{(n-1)} \to 0$ which proves the convergence of the system of equations (5.2).

5.2 Image Retrieval Using Iterative Similarity Propagation

The learned similarity matrices could be used to improve the acuracy of the image retrieval system. We use visual features, extracted from the image content for one modality, and the textual annotation of each image for the second modality. In particular, we construct the K similarity matrix using the Euclidean distance between MPEG-7 visual features. The Z_1 and Z_2 matrices are constructed by linking the images to the keywords used to annotate them. The G_1 and G_2 matrices are constructed based on the correlation between the annotating keywords.

Let X be the visual feature matrix with rows as image regions and columns as their visual features. Let Y_1 and Y_2 be the document term matrices with the images as rows and the terms (weighted by TF*IDF) as columns provided by the two different image annotation approaches outlined in chapter 4.

The initial image similarity matrix $K = [K_{ij}]_{M \times M}$, which is based on the low-level visual features, is given by

$$K_{ij} = \frac{\sum_{x_i=1}^{n_i} \underset{x_j}{argmax}\{S(x_i, x_j)\}}{max(n_i, n_j)}, \tag{5.8}$$

where $x_1, .., x_{n_i}$ are the set of n_i regions forming image I_i and $x_1, .., x_{n_j}$ are the set of n_j regions forming image I_j. Let $S(x_i, x_j)$ be the similarity between regions x_i and x_j of images I_i and I_j. The similarity is computed by converting the Euclidean distance between image regions into similarities using

$$S(x_i, x_j) = 1 - \frac{Eud(X_i, X_j)}{\underset{i,j}{max} Eud(X_i, X_j)} = 1 - \frac{\sqrt{(X_i - X_j)(X_i - X_j)^T}}{\underset{i,j}{max} \sqrt{(X_i - X_j)(X_i - X_j)^T}} \quad (5.9)$$

where X_i and X_j denote the i^{th} and j^{th} rows of matrix X respectively.

For the textual features, the initial similarity matrices $G_1 = [G_{1_{ij}}]$ and $G_2 = [G_{2_{ij}}]$ are calculated based on the cosine similarity using

$$G_{1_{ij}} = \frac{Y_{1_i} \bullet Y_{1_j}}{||Y_{1_i}||.||Y_{1_j}||}, \quad (5.10)$$

and

$$G_{2_{ij}} = \frac{Y_{2_i} \bullet Y_{2_j}}{||Y_{2_i}||.||Y_{2_j}||}. \quad (5.11)$$

where Y_{1_i} and Y_{1_j} denote the i^{th} and j^{th} colums of matrix Y_1 respectively, and Y_{2_i} and Y_{2_j} denote the i^{th} and j^{th} colums of matrix Y_2 respectively.

Initially, we set the initial intra-object similarities to be their content similarities, i.e. $\hat{K}^{(0)} = K$, $\hat{G_1}^{(0)} = G_1$ and $\hat{G_2}^{(0)} = G_2$. Then, we perform few iterations of the similarity propagation using the system of equations in (5.2).

Figure 5.4 shows the block diagram of the proposed image retrieval system based on multi-modal similarity propagation.

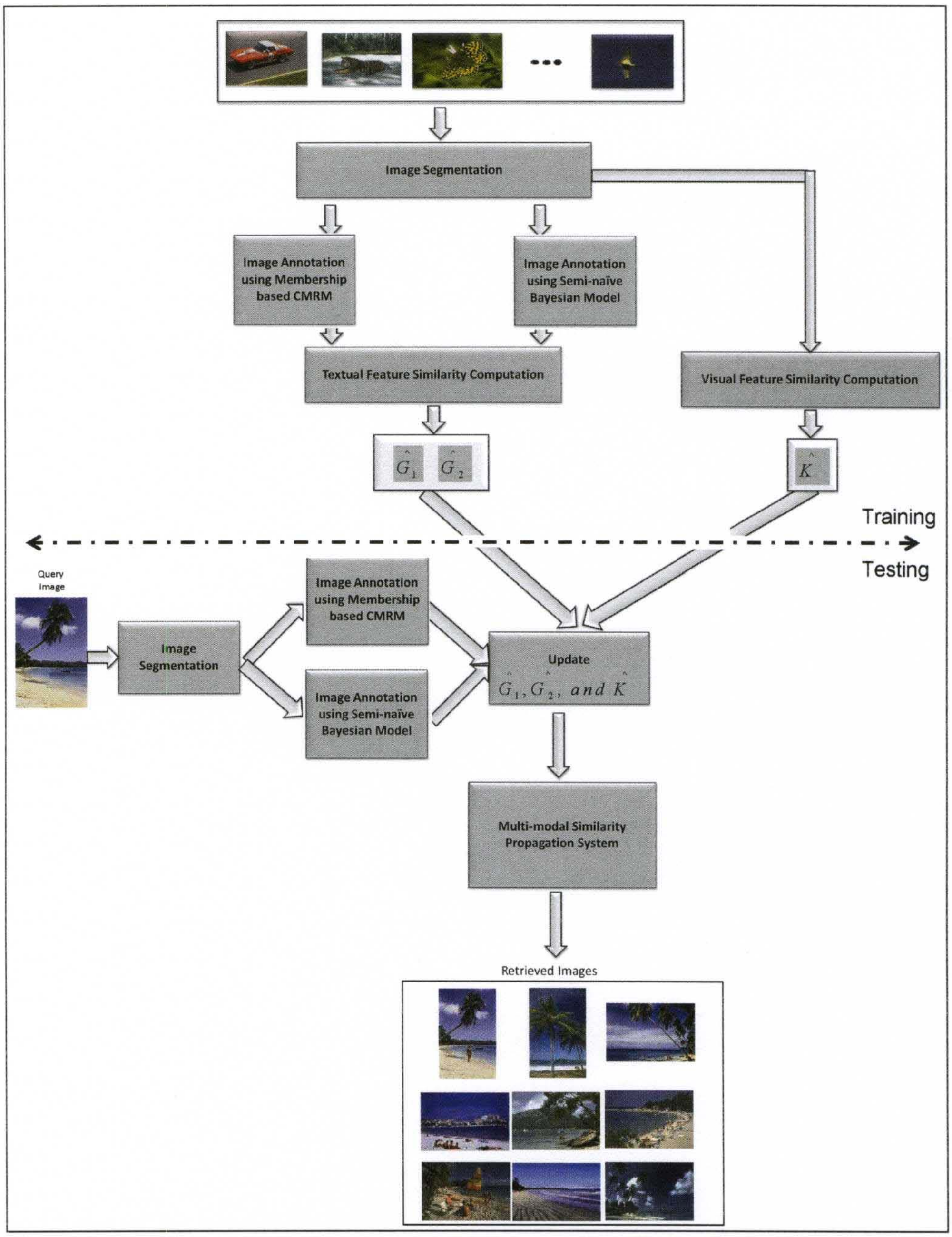


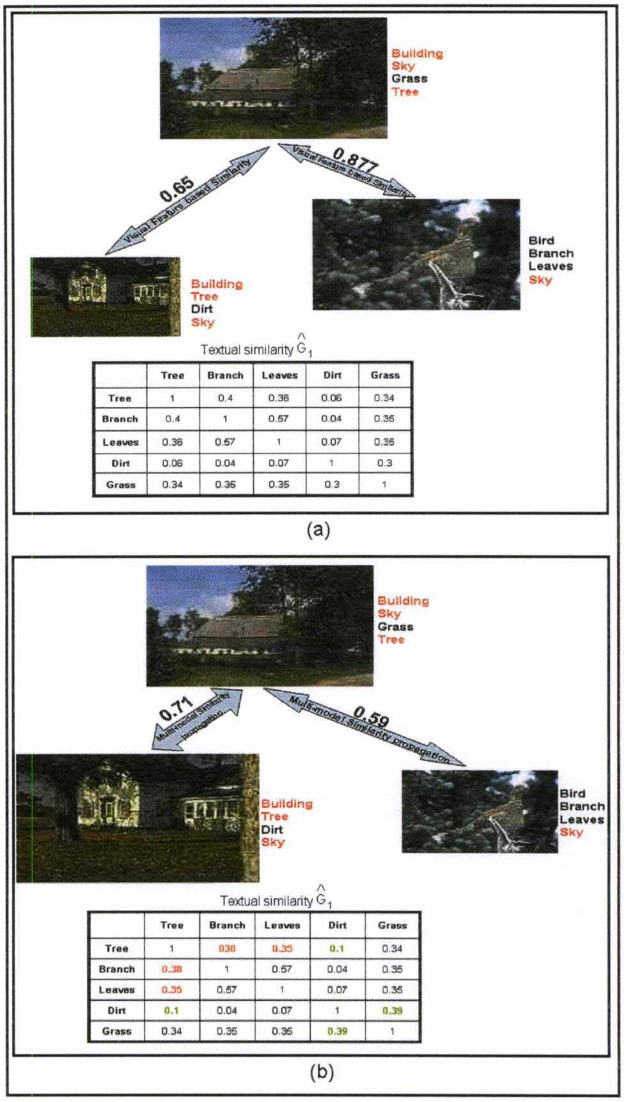
Image Segmentation
Image Annotation using Membership based CMRM
Image Annotation using Semi-naïve Bayesian Model
Textual Feature Similarity Computation
Visual Feature Similarity Computation
$\hat{G}_1$ $\hat{G}_2$
$\hat{K}$
Training
Testing
Query Image
Image Segmentation
Image Annotation using Membership based CMRM
Image Annotation using Semi-naïve Bayesian Model
Update $\hat{G}_1, \hat{G}_2,$ and $\hat{K}$
Multi-modal Similarity Propagation System
Retrieved Images

Figure 5.4: Block diagram of the proposed image retrieval using multi-modal similarity propagation

5.3 Experimental Results

In these experiments, we use the dataset described in section 4.3. First, all training images are coarsely segmented by clustering the color distribution. The Competitive Agglomeration (detailed in 2.1.5) is used to cluster each image into an optimum number of regions. Then, each region is characterized by two color, two texture, one shape and one textual feature set. The color features consist of HSV and LUV color moment of 9-Dim each. The texture feature consists of one global 5-Dim edge histogram, and 20-Dim wavelet coefficients. The shape feature consists of the eccentricity, orientation, area, solidity and extent of each region. Each low-level feature set is normalized such that its components sum to 1.

In Figure 5.5, we present an illustrative example of the multi-modal similarity propagation. In Figure 5.5(a), we show the similarity of two images to a given query image (top image) based on visual features. The closer and the bigger the image is, the more similar it is to the query image. For instance, the "Bird" image is more similar to the query image based on color and texture features. However, semantically this image is not relevant to the query image. On the other hand, the third image is semantically very relevant but it is less similar to the query based on the visual feature. In Figure 5.5(b), we show the same image similarity values after the multi-modal similarity propagation. We notice that the new similarity values reflect the semantic relevance of the images. For instance, the "Bird" image similarity decreased because its annotation is irrelevant to the query image. On the other hand, the other "building" image gains in relevance because it is annotated with keywords that are common with the query image.



Textual similarity $\hat{G}_1$

	Tree	Branch	Leaves	Dirt	Grass
Tree	1	0.4	0.36	0.06	0.34
Branch	0.4	1	0.57	0.04	0.35
Leaves	0.36	0.57	1	0.07	0.35
Dirt	0.06	0.04	0.07	1	0.3
Grass	0.34	0.36	0.35	0.3	1

Textual similarity $\hat{G}_1$

	Tree	Branch	Leaves	Dirt	Grass
Tree	1	038	0.35	0.1	0.34
Branch	0.38	1	0.57	0.04	0.35
Leaves	0.35	0.57	1	0.07	0.35
Dirt	0.1	0.04	0.07	1	0.39
Grass	0.34	0.35	0.36	0.39	1

Figure 5.5: Illustration of the Multi-modal Similarity Propagation. (a) Similarity before propagation, and (b) Similarity after propagation.

In Figure 5.5, we also display two tables that show similarities between pairs of keywords ($\hat{G}_1$), annotating the images, before and after the multi-modal similarity propagation. Keyword similarity values which decreased (increased) after the multi-modal similarity propagation are marked in

Parameter	α	β_1	β_2	λ
Value	0.4	0.7	0.7	0.6

Table 5.1: Optimal values of the similarity propagation parameters

Red (Green). For instance, keywords such "Dirt" and "Grass", or "Dirt" and "Tree" have their similarity values increased. Similarly, keywords such "Leaves" and "Tree", or "Branch" and "Tree" have their similarity decreased.

The proposed similarity propagation is evaluated by using it to retrieve images and compare the precision and recall values with those obtained using standard CBIR system that uses visual features only. Precision is defined as the number of retrieved relevant images over the number of retrieved images. Recall is defined as the number of retrieved relevant images over the total number of relevant images.

The weights used for similarity matrices (α, β_1 and β_2) and the decay factor λ are determined experimentally as the set of parameters that yield the best precision. The optimal values of these parameters are shown in Table 5.1.

5.3.1 Comparison with Standard CBIR

In Figure 5.6, we plot the precision values versus the number of iterations of the multi-modal similarity propagation when the top 20 images are considered. As expected, the precision obtained when we retrieve images using visual features is constant. On the other hand, the precision value of the proposed image retrieval method doubled after only three iterations and reached its maximum after five iterations. The precision value remains constant and does not improve beyond five iterations.

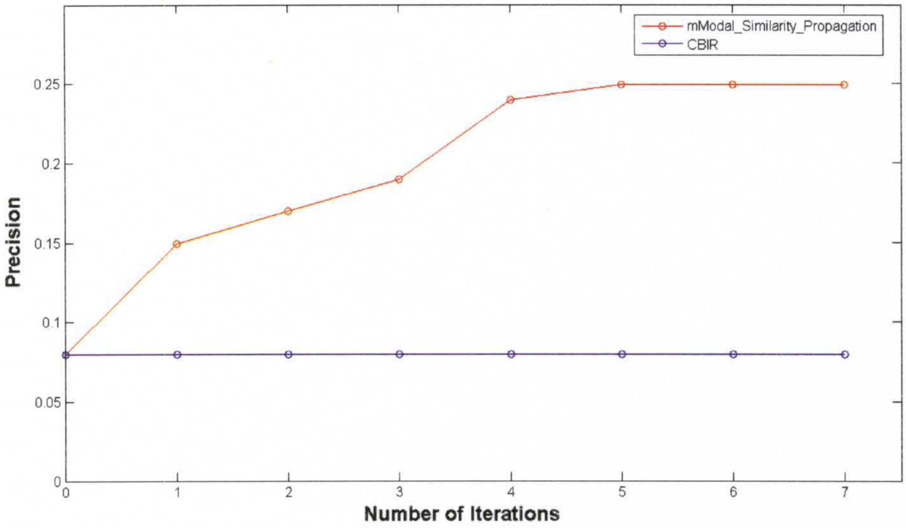


Figure 5.6: Precision Vs Number of Iterations when 20 images are retrieved.

In Figure 5.7, we plot the recall values versus the number of iterations used for the multi-modal similarity propagation and compare the results with standard CBIR. As expected, the recall obtained when images are retrieved using visual features only is constant. On the other hand, the recall value of the proposed method doubled when the number of iterations is two and reaches its maximum within five iterations. after that, the recall value remains constant. Thus, in the following experiments, we set the number of iterations to five.

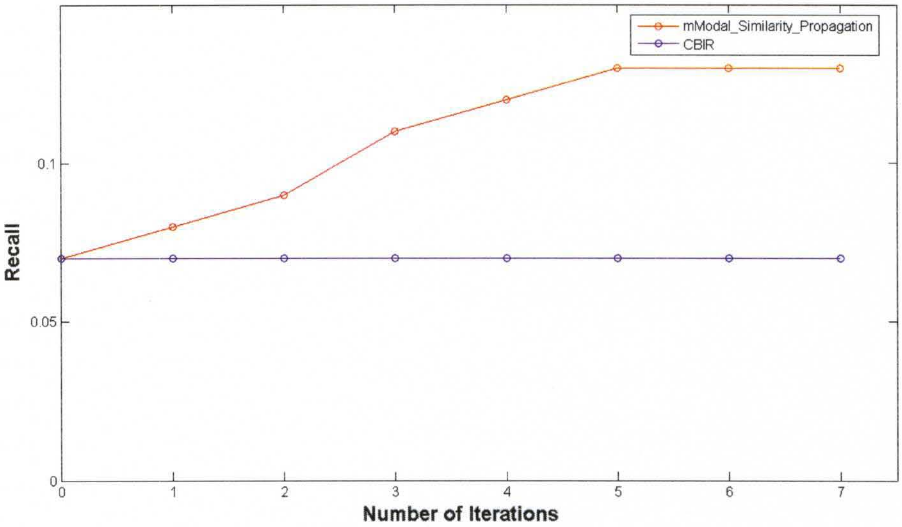


Figure 5.7: Recall Vs Number of Iterations when 20 images are retrieved.

5.3.2 Comparison with Hybrid Method

In this section, we compare our approach to a similar method that uses both visual and textual features to retrieve similar images. We use the method proposed in [74]. We implemented this baseline method as outlined in [74]. We tune the optimal weight parameter to the data set that we are using. The optimal value of this weight of similarity matrix was found to be 0.5. Although the low-level visual features we used are different from [74], these differences will not bias the final evaluation since it is the method itself rather than the features that are being evaluated. The reason that we choose the method proposed in [74] as baseline method is as follows. First, this approach represents a traditional way of combining multi-modalities for image retrieval. Second, we do not involve a training phase to select representative query set and learn the optimal weight set for them.

Figures 5.8, 5.9 and 5.10 compare the precision and recall values of the baseline system with those obtained using the proposed system. In these experiments, we vary the number of retrieved images from 1 to 50. For each value, we process all query images and average the results.

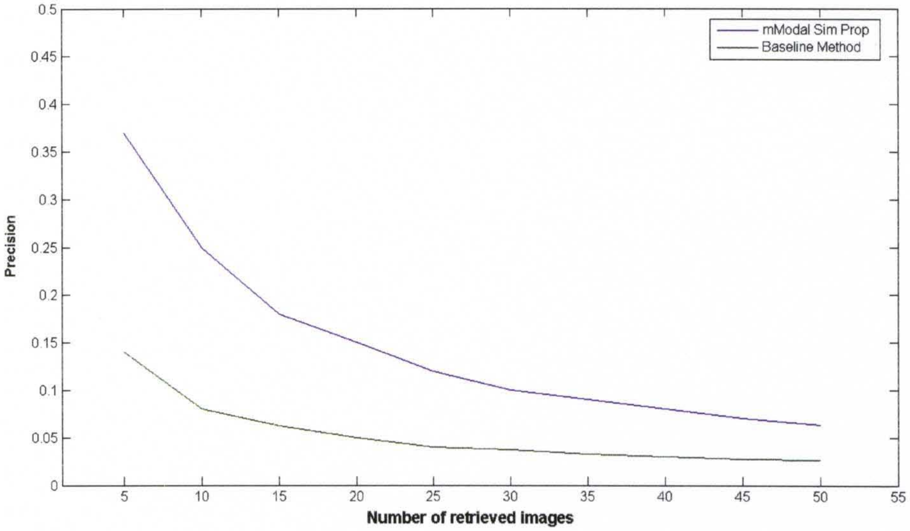


Figure 5.8: Comparison of the precision values for the proposed system and the baseline system versus the number of retrieved images.

In Figure 5.8, we display the retrieval precision values versus the number of retrieved images for both methods. As it can be seen, the proposed similarity propagation method yields higher precision values. The difference between the two systems is more pronounced when fewer images are retrieved . This indicates that the proposed similarity propagation does a better job at ranking the similar images.

In Figure 5.9, we compare the recall values versus the number of retrieved images of the proposed and the baseline methods. As it can be seen, the similarity propagation method has a much higher recall.

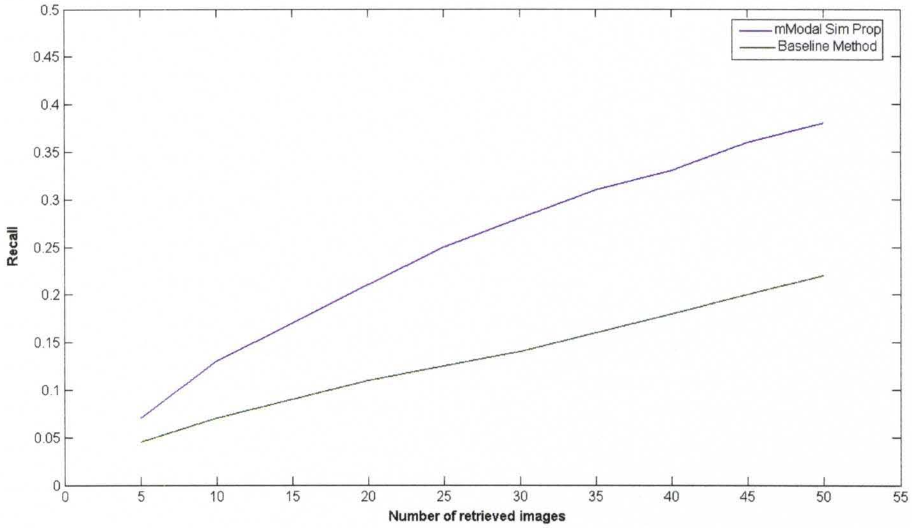


Figure 5.9: Comparison of the recall values for the proposed system and the baseline system versus the number of retrieved images.

In Figure 5.10, we plot the recall vs. precision graph, and compare the two curves. This figure confirms that our method significantly outperforms the baseline method. This greater performance is due to two main factors. First, the effectiveness of our image annotation approaches presented in chapter 4. Second, our similarity propagation method can reduce the effect of the semantic gap. Typically, textual features are more effective than image content features. However, when the annotation is automatic, it is prone to several mislabeling errors. Hence, combining the two kinds of features will do a better job, just as indicated in [75] that "while text and images are separately ambiguous, jointly they tend not to be". Our experiments confirmed this observation. However, combining different features can also be biased by the features themselves. The iterative propagation approach explores the mutual reinforcement among different data types which in some sense can correct such biases. It can also be regarded as a non-linear combination method of different types of features.

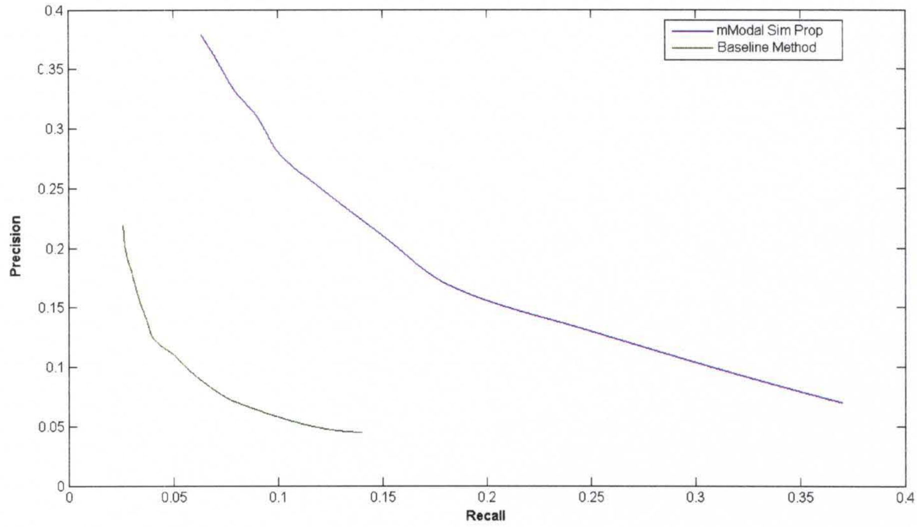
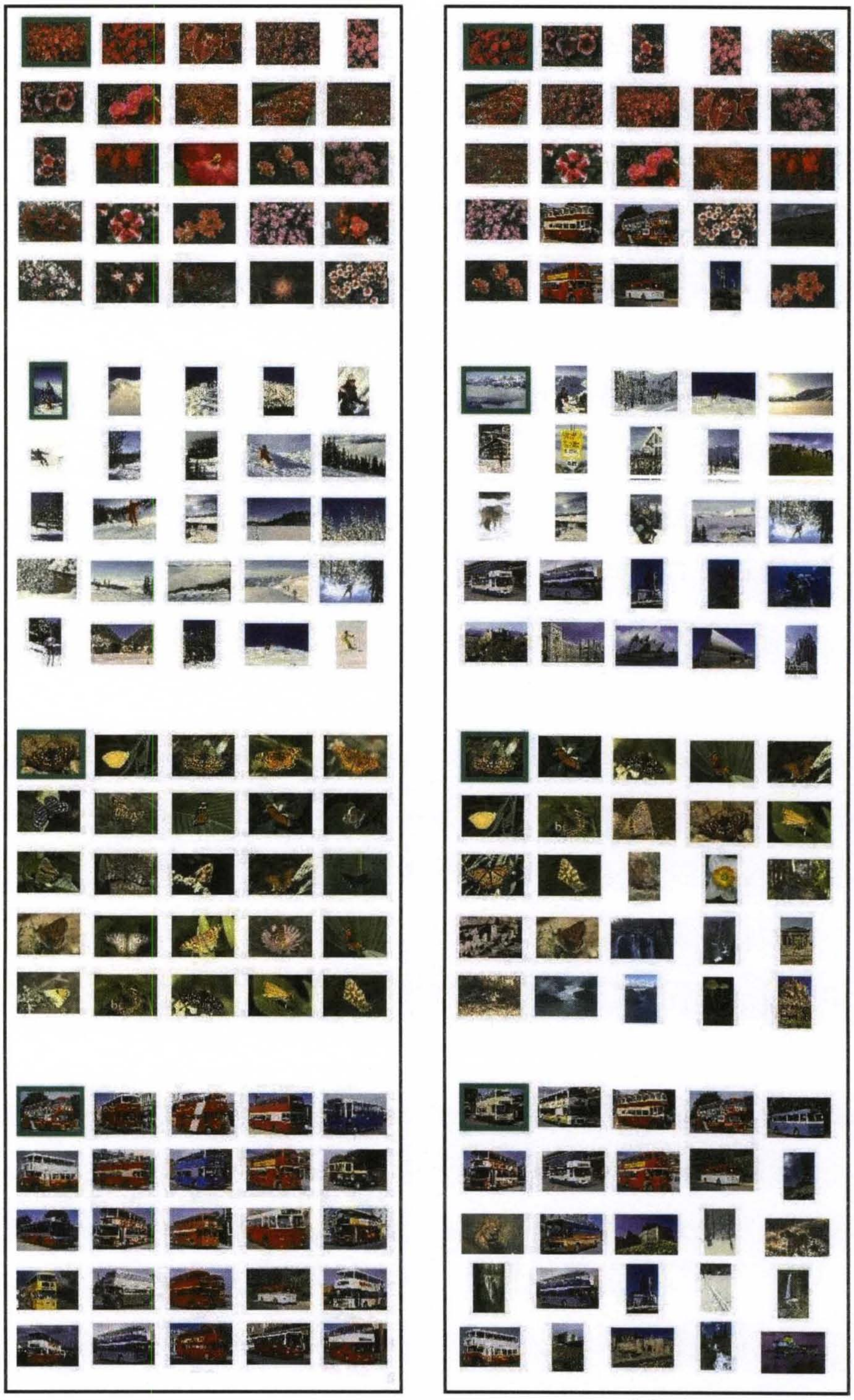


Figure 5.10: Comparison of the precision values for the proposed system and the baseline system versus recall values

In Figure 5.11, we display the 24 most similar images to 4 sample query images retrieved using the proposed and the baseline methods. The first image represents the query provided by the user. As it can be seen, the retrieved images are the closest (share many images) to the ground truth partition. In other words, the retrieved images are compatible with the users' notion of similar images.

One can notice that the images retrieved by the baseline method are less homogeneous. For instance, in Figure 5.11(b), many images from the bus category are retrieved when the query image is a flower or a ski scene. this is because these images share similar colors. Similarly, images from the waterfall category are retrieved in response to a butterfly query image because they have similar visual appearance based on their texture descriptors. On the other hand, the proposed multi-modal similarity propagation approach, combines both visual and textual features, and does not retrieve as many



(a) (b)

Figure 5.11: Top 25 representative from 4 typical clusters generated by (a) the proposed method, (b) the baseline method.

irrelevant images.

CHAPTER 6

CONCLUSIONS AND POTENTIAL FUTURE WORK

6.1 Conclusions

In the first part of this thesis, we proposed a possibilistic approach for Generalized Dirichlet (GD) mixture parameter estimation, data clustering, and feature weighting. The proposed algorithm, called Robust and Unsupervised Learning of Finite Generalized Dirichlet Mixture Models (RULe_GDM) addresses the problems associated with the high dimensionality and sparsity of the feature space. RULe_GDM exploits a property of the Generalized Dirichlet distributions that transforms the data to make the features independents and follow Beta distributions. Then, it searches for the optimal relevance weight for each feature within each cluster. This property makes RULe_GDM suitable for noisy and high-dimensional feature spaces. In addition, RULe_GDM associates two types of memberships with each data sample. The first one is the posterior probability and indicates how well a sample fits each estimated distribution. The second membership represents the degree of typicality and is used to identify and discard noise points and outliers. RULe_GDM minimizes one objective function that combines learning the two membership functions, the distribution parameters, and relevance weights for each feature within each distribution. In addition to the baseline RULe_GDM, we proposed extensions to this approach. The first one adapts the algorithm to learn relevance weights for each feature subset within each cluster. The second extension generalizes RULe_GDM to find the optimal number of clusters in an unsupervised and efficient way by exploiting some properties of the possibilistic membership function. The third extension is a semi-supervised version of RULe_GDM that uses partial supervision information in the form of constraints to guide the clustering process. The performance of our clustering approach is illustrated and compared to similar algorithms. We used synthetic data to illustrate robustness to noisy and high dimensional features. We also integrate it as main component of our image annotation system.

In the second part of this thesis, we proposed two probabilistic image annotation approaches where words are assigned conditionally to images. The first image annotation method relies on a semi-naive Bayesian model. The second one relies on a membership based Cross Media Relevance Model. We used our proposed semi-supervised possibilistic clustering and feature subset weighting based on robust GD mixture modeling (sRULe_GDM_FSS) to summarize the image region collection. We proposed an approach that extracts partial supervision information based on the relevancy of the keywords annotating the images. The possibilistic memberships generated by sRULe_GDM_FSS algorithm are used in subsequent steps to identify dependent region clusters using a greedy selection and joining algorithm.

Finally, Bayes rule and the possibilistic membership based Cross Media Relevance Model are used to label images based on the posterior probability of each concept.

The proposed image annotation approaches were implemented and tested with standard benchmark dataset. We compared our proposed image annotation approaches to three state-of-the-art methods. We showed that our approaches outperform these methods. We argued that the improvement in performance can be accredited to the following factors:

- The use of Generalized Dirichlet (GD) to model the image region collection.
- The use of possibilistic approach to detect noise points and outliers, and find the optimal number of clusters.
- The use of constrained clustering and feature weighting algorithm to group image regions into homogeneous categories.
- The extraction of pairwise constraints in an unsupervised way based on the relevancy of all concepts annotating the training image regions.
- The use of membership degrees instead of simple inverted lists to estimate the correlation among the region clusters.
- Instead of assuming the events of observing region clusters within an image are mutually independent, we use a Greedy Selection and Joining algorithm to find independent subsets of region clusters.

In the third part of the thesis, we presented an image retrieval framework based on multi-modal similarity propagation. The proposed framework is designed to deal with two data modalities: low-level visual features and high-level textual keywords. The iterative similarity propagation model attempts to fully exploit the mutual reinforcement of relational data which results in a non-linear combination of different modalities. It uses the intra-object similarities of textual modality to influence the low-level visual modality. It performs this approach iteratively and attempts to capture the similarities of images at the semantic level. Our experimental results demonstrated the effectiveness of the proposed multi-modal similarity propagation compared to the standard CBIR and hybrid image retrieval systems. We have shown that when low-level features are not sufficient to capture the high-level semantics of the images, the inclusion and propagation of the high-level keywords could improve the performance significantly. Similarly, when the annotating keywords are erroneous, due to the completely unsupervised method, their propagation with the visual features could adjust the correlation of these features and limit their influence on the overall retrieval accuracy.

6.2 Potential Future Work

The obtained experimental results have indicated that our proposed approach is effective and promising. However, they have also identified some limitations that could be addressed. In the following, we list some tasks that could be explored to enhance the performance of the proposed image annotation and retrieval framework.

6.2.1 System Scalability

One extension of our system could be related to the scalability issue. In fact, we used a relatively small vocabulary size (less than 100 keywords and less than 10k images). In a more realistic scenario, a much larger data set may be needed. In this case, the vector space notation may not be appropriate, and thus, integrating the low-level features into the clustering phase is not trivial. Moreover, the sRULe_GDM_FSS algorithm used to categorize the image regions is not scalable. That is it cannot handle a large data set that does not fit into memory. One possible way to develop a scalable version of sRULe_GDM_FSS is to partition the data, cluster each partition, and then combine the clustering results. In this case, each partition could be clustered in parallel on separate machines or in separate threads.

6.2.2 User Relevance Feedback

It is possible to integrate a relevance feedback component into our image retrieval system to further minimize the semantic gap. Relevance feedback has shown great results in focusing on users query. If this feedback could be captured and represented in an efficient way, it could be used to strengthen each component of our proposed system. For instance, relevance feedback could be used to provide more reliable supervision information for our semi-supervised clustering algorithm. Similarly, it could be used to enhance the pefomance of the image annotation component and to adjust the image retrieval results.

REFERENCES

[1] ludicorp, “flickr”, *http://www.flickr.com/.*

[2] ImageShack Corp., “Imageshack,” *http://www.imageshack.us/.*

[3] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, & R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Trans. Patt. Analysis Mach. Intell., vol. 22..*

[4] Ornager, S. “Image retrieval: Theoretical and empirical user studies on accessing information in images”. *In Proceedings of the 60th American Society of Information Retrieval Annual Meeting. Volume 34. (1997) 202–211*

[5] X. S. Zhou & T. S. Huang, “Unifying keywords and visual contents in image retrieval,” *IEEE Multimedia, vol. 9, no. 2, pp. 23–33, 2002.*

[6] A. Del Bimbo, “Visual Information Retrieval book”, *Morgan Kaufmann,1999.*

[7] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: “Content-based image retrieval at the end of the early years”. *IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1349–1380*

[8] M. Szummer and R. Picard, "Indoor-Outdoor Image Classification," *Proc. Workshop Content-Based Access to Image and Video Databases, 1998.*

[9] A. Vailaya, A. Jain, and H. Zhang, "On Image Classification: City vs. Landscape," *Pattern Recognition, vol. 31, pp. 1921-1936, Dec. 1998.*

[10] Hichem Frigui & Olfa Nasraoui, "Unsupervised learning of prototypes and attribute weights," *Pattern Recognition Journal, vol. 37, pp. 567-581, 2004.*

[11] V. Lavrenko, R. Manmatha & J. Jeon, "A model for learning the semantics of pictures," *Neural Information Processing System (NIPS), 2003.*

[12] T.M. Mitchell, "Machine Learning," *McGraw Hill, 1997.*

[13] K. Barnard and D. Forsyth, "Learning the Semantics of Words and Pictures," *Proc. Int'l Conf. Computer Vision, vol. 2, pp. 408-415, 2001.*

[14] D. Blei and M. Jordan, "modeling Annotated Data," *Proc. ACM SIGIR Conf. Research and Development in Information Retrieval, 2003.*

[15] K. Barnard & D. Forsyth. "Learning the semantics of words and pictures". *In International Conference on Computer Vision, Vol.2, pages 408-415, 2001.*

[16] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, & J. Malik. "Blobworld: A system for region-based image indexing and retrieval". *In Third International Conference on Visual Information Systems, Lecture Notes in Computer Science, 1614, pages 509-516, 1999.*

[17] P. Duygulu, K. Barnard, N. de Freitas, & D. Forsyth. "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary". *In Seventh European Conference on Computer Vision, pages 97-112, 2002.*

[18] Y. Mori, H. Takahashi, & R. Oka. "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.*

[19] Shi Rui, Wanjun Jin & Tat-Seng Chua, "A Novel Approach to Auto Image Annotation Based on Pairwise Constrained Clustering and Semi-naïve Bayesian Model", *Proceedings of the 11th International Multimedia modeling Conference (MMM'05).*

[20] H.Frigui, J.Caudill, "Region Based Image Annotation", *Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR'06), 2006.*

[21] Y. Rui, T. Huang, and S. Chang. "Image retrieval: Current techniques, promising directions and open issues". *Journal of Visual Communication and Image Representation, 10(4):39–62, April 1999*

[22] Flickner M, Sawhney H, Niblack W, et al. "Query by image and video content: the QBIC system". *IEEE Computer 1995; 28(9): 23- 32.*

[23] Smith JR, Chang S-F. "VisualSEEk: a fully automated content-based image query system". *Proceedings of the ACM International Conference on Multimedia, Boston, Massachusetts, Nov. 18-22, 1996; 87-98.*

[24] J. S. Hare, P. H. Lewis, P. G. B. Enser, and C. J. Sandom, "Mind the gap: Another look at the problem of the semantic gap in im-

age retrieval". *in Proceedings of SPIE Multimedia Content Analysis, Management and Retrieval 2006.*

[25] Antani S, Kasturi R, Jain R. "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video". *Patt Recog 2002; 35: 945-965.*

[26] Swain MJ, Ballard DH. "Color indexing". *Int J Comp Vision 1999; 7(1): 11-32.*

[27] Jiwoon Jeon and R. Manmatha. "Using Maximum Entropy for Automatic Image Annotation". *International conference on image and video retrieval No3, Dublin , IRLANDE, 2004.*

[28] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", *Berkeley, University of California Press, 1:281-297*

[29] K. Wagstaff, C. Cardie, S. Rogers & S. Schroedl, "Constrained K-means clustering with background knowledge," *Proc. of Int'l Conference on Machine Learning (ICML-2001).*

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from in- complete data via the em algorithm". *Journal of the Royal Statistical Society: Series B, 39(1):1-38, November 1977.*

[31] X. Zhuang, Y. Huang, K. Palaniappan, and J. S. Lee, "Gaussian mixture modeling, decomposition and applications," *IEEE Trans. Image Processing, vol. 5, pp. 1293-1302, 1996*

[32] H. Frigui & R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition, vol. 30, no. 7, pp. 1223-1232, 1997.*

[33] H. Almuallim and T. G. Dietterich, "Leaning wilh many imelevanl features:' *in Ninth Notional Conf on AI, 1991, pp. 547-552.*

[34] K. Kim and L. A. Rendell. "The feature selection problem: Traditional methods and a new algorithm," *in Tenth Notional Conf on AI, 1992, pp. 129-134.*

[35] L. A. Rendell and K. Kira. "A practical approach to feature selection," *in Inter. Conf on machine learning, 1992. pp. 249-256.*

[36] Hichem Frigui & Olfa Nasraoui, "Unsupervised learning of prototypes and attribute weights," *Pattern Recognition Journal, vol. 37, pp. 567–581, 2004.*

[37] Hichem Frigui & Olfa Nasraoui, "Simultaneous clustering and attribute discrimination" *in Proceedings of the IEEE International Conference on Fuzzy systems, 2000, pp. 158-163*

[38] H. Frigui and S. Salem, "Fuzzy clustering and subset feature weighting" *in IEEE International Conference on Fuzzy systems, 2003.*

[21] J. Jeon, V. Lavrenko & R. Manmatha, "Automatic Image Annotation and Retrieval using CrossMedia Relevance Models", *SIGIR'03, July 28–August 1, 2003.*

[40] W. Liu and X. Tang, "learning an image-word embedding for image auto-annotation on the nonlinear latent space," *in Proceedings of ACM multimedia, 2005.*

[41] D. Blei, A. Ng, and M. Jordan. "Latent Dirichlet allocation". *Journal of Machine Learning Research , 3:993–1022, January 2003.*

[42] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. “Supervised Learning of Semantic Classes for Image Annotation and Retrieval”. *IEEE Transactions on pattern analysis and machine intelligence, vol. 29, no. 3, 2007.*

[43] Maron, O., T. Lozano-Perez, “A Framework for Multiple Instance Learning”. *Neural Information Processing Systems 10, 1998.*

[44] Jia-yu Pan, Hyung-jeong Yang, Pinar Duygulu, Christos Faloutsos. “Automatic image captioning”. *International Conference on Multimedia and Expo (ICME). 2004.*

[45] Bob Rehder, Michael L. Littman, Susan Dumais, Thomas K. Landauer. ”Automatic 3-Language Cross-Language Information Retrieval with Latent Semantic Indexing”. *In The Sixth Text Retrieval Conference Notebook Papers (TREC6), 1997.*

[46] J. S. Hare, P. H. Lewis, P. G.B. Enser, and C. J. Sandom. “A Linear-Algebraic Technique with an Application in Semantic Image Retrieval”. 2006

[47] K. Yu, S. Yu, and V. Tresp, “Multi-Label Informed Latent Semantic Indexing”, *in Proceedings of 28th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR) , 2005.*

[48] D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". *Nature 401, 1999.*

[49] David Guillamet and Jordi Vitri‘a. “Determining a Suitable Metric When using Non-negative Matrix Factorization ”. *Proceedings of the*

16 th International Conference on Pattern Recognition (ICPR'02). 2002.

[50] Xu, W., Liu, X., & Gong, Y. "Document-clustering based on non-negative matrix factorization". *In Proceedings of SIGIR03, 2003.*

[51] Chapelle, O. Haffner, P. Vapnik, V.N. "Support vector machines for histogram-based image classification". *Speech & Image Process. Services Res. Lab., AT&T Labs-Res., 1999.*

[52] X. Qi, Y. Han. "Incorporating multiple SVMs for automatic image annotation". *Pattern Recognition. 2007.*

[53] H.Frigui, J.Caudill, "Unsupervised Image Segmentation and Annotation for Content-Based Image Retrieval", *IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006*

[22] H.Frigui, J.Caudill, "Region Based Image Annotation", *Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR'06),2006.*

[23] R. Yan & A. Hauptman, "A discriminative learning framework with pair-wise constraints for video object classification," *CVPR. 2004.*

[56] R.Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age" *in Penn State University Technical Report CSE, 2006.*

[57] Chen, C., Gagaudakis, G., Rosin, P. "Similarity-based image browsing." *Proceedings of the 16th IFIP World Computer Congress. International Conference on Intelligent Information Processing. 2000.*

[58] B. S. Manjunath, T. Sikora, P. Salembier. "Introduction to MPEG-7: Multimedia Content Description Interface ". *John Wiley, 2002.*

[59] S. Basu, A. Banerjee, and R. Mooney, "Active semi supervision for pairwise constrained clustering". *in proc. SIAM DM, 2004.*

[60] N. Grira, M. Crucianu, and N. Boujemaa, "Semisupervised fuzzy clustering with pairwise-constrained competitive agglomeration," *in IEEE Conf. Fuzzy Systems, 2005.*

[61] V. Lavrenko, M. Choquette, & W. Croft. "Cross-lingual relevance models". *Proceedings of the 25th annual international ACM SIGIR conference, 2002.*

[62] S. Basu, A. Banerjee, and R. Mooney, "Active semi supervision for pairwise constrained clustering," *in proc. SIAM DM, 2004.*

[63] H. Almuallim and T. G. Dietterich. "Learning with many irrelevant features". *In Ninth National Conference on artificial intelligence, pages 547–552, 1991.*

[64] L. A. Rendell and K. Kira. "A practical approach to feature selection". *In International Conference on machine learning, pages 249–256, 1992.*

[65] D. Wettschereck, D. W. Aha, and T. Mohri. "A review and empirical evaluation of feature weightin g methods for a class of lazy learning algorithms". *Artificial Intelligence Review, 11:273–314, 1997.*

[66] R. Datta, D. Joshi, J. Li, and J. Z. WANG. "Image Retrieval: Ideas, Influences, and Trends of the New Age". *ACM Comput. Surv. 2008.*

[67] J. Bezdek, "Pattern Recognition with Fuzzy Objective Function algorithms", *Plenum Press, New York, 1981.*

[68] G. Salton and M. McGill. "An Introduction to Modern Information Retrieval". *1983.*

[69] University of California at Berkeley. Corel database website at *http://elib.cs.berkeley.edu/photos/corel/.*

[70] S. Basu, A. Banerjee, and R. Mooney, "Active semi supervision for pairwise constrained clustering," *in proc. the SIAM int. Conf: on Data mining, 2004, pp. 333-344.*

[71] A. M. Bensaid, L. 0. Hall, J. C. Bezdek, and L. P. Clarke, "Partially supervised clustering for image segmentation," *Pattern Recognition, vol. 29, pp. 859-872, 1996.*

[72] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," *in proc. Int. Conf on Machine Learning, 2002, pp. 19-26.*

[73] H. Frigui and R. Mahdi. "Semi-Supervised Clustering and Feature Discrimination with Instance-Level Constraints". *IEEE 2007.*

[74] Scott C. Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. "Indexing by latent semantic analysis". *Journal of the American Society of Information Science, 41(6):391-407, 1990.*

[74] Chen, Z., Liu, W.Y., Zhang, F., Li, M.J. and Zhang, H.J. "Web Mining for Web Image Retrieval", *Journal of the American Society for Information Science and Technology, 52(10), (2001), 831-839.*

[75] N. Bouguila and D. Ziou. "MML-Based Approach for High-Dimensional Learning using the Generalized Dirichlet Mixture". *In Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03, page 53, 2005.*

[76] N. Bouguila and D. Ziou. "A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture". *IEEE Transactions on Image Processing, 15(9):2657–2668, 2006.*

[77] N. Bouguila and D. Ziou, "MML-Based Approach for High-Dimensional Learning Using the Generalized Dirichlet Mixture," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition—Workshops, p. 53, 2005.*

[78] C.S. Wallace and D.L. Dowe, "MML Clustering of Multi-State, Poisson, von Mises Circular and Gaussian Distributions," *Statistics and Computing, vol. 10, no. 1, pp. 73-83, 2000.*

[79] Samad Kazi , Sverker Fridqvist , "Scalable Discriminant Feature Selection for Image Retrieval and Recognition" – *Vasconcelos, Vasconcelos - 2004*

[80] R. O. Stehling, M. A. Nascimento, and A. X. Falcao. "MiCRoM: A Metric Distance to Compare Segmented Images". *VISUAL 2002, LNCS 2314, pp. 12–23, 2002.*

[81] Q. Zhao, S. Brennan and H. Tao. "Differential EMD Tracking". *2007 IEEE*

[82] R. Krishnapuram and J. Keller, "A possihilistic approach to clustering," *IEEE Trans. Fuzzy Syst., vol. 1, pp. 98-110, May 1993.*

[83] Frigui, H., Krishnapuram, R., "A Robust Competitive Clustering Algorithm with Applications in Computer Vision", *PAMI(21), No. 5, May 1999, pp. 450-465.*

[84] Pal, N.R., Pal, K., Keller, J.M., Bezdek, J.C. "A Possibilistic Fuzzy c-Means Clustering Algorithm", *Fuzzy Systems, IEEE Transactions on Volume: 13. 2005.*

[85] G. McLachlan and D. Peel, "Finite Mixture Models". *New York: John Wiley & Sons, 2000.*

[86] N. Bouguila and D. Ziou, "Unsupervised Learning of a Finite Discrete Mixture: Applications to Texture modeling and Image Databases Summarization," *J. Visual Comm. and Image Representation, vol. 18, no. 4, pp. 295-309, 2007.*

[87] T. Wong, "Generalized Dirichlet Distribution in Bayesian Analysis," *Applied Math. and Computation, vol. 97, pp. 165-181, 1998.*

[88] N. Bouguila and D. Ziou, "A Powerful Finite Mixture Model Based on the Generalized Dirichlet Distribution: Unsupervised Learning and Applications," *Proc. 17th Int'l Conf. Pattern Recognition (ICPR '04), pp. 280-283, 2004.*

[89] Bouguila, N. "Clustering of Count Data Using Generalized Dirichlet Multinomial Distributions". *Knowledge and Data Engineering, IEEE Transactions on Volume: 20. 2008.*

[90] R. Duda, P. Hart, and D. Stork, "Pattern Classification". *2nd ed. New York: Wiley, 2001.*

[91] A. Narayanan, "A note on parameter estimation in the multivariate Beta distribution", *Comput. Math. Applicat., vol. 24, no. 10, pp. 1117, 1992.*

[92] G. Ronning, "Maximum likelihood estimation of Dirichlet distributions", *J. Stat. Comput. Simul., 1989.*

[93] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain, "Simultaneous Feature Selection and Clustering Using Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, *vol. 26, no. 9, pp. 1154-1166, Sept. 2004.*

[94] Hichem Frigui and Raghu Krishnapuram, "A robust algorithm for automatic extraction of an unknown number of clusters from noisy data ", *Pattern Recognition Letters , Volume 17 Issue 12 . 1996.*

[95] A. El Zaart and D. Ziou, "Statistical modeling of multimodal SAR images," *Int. J. Remote Sens., to be published.*

[96] R. J. Connor and J. E. Mosimann, "Concepts of independence for proportions with a generalization of the Dirichlet distribution," *J. Amer. Stat. Assoc., vol. 39, pp. 1–38, 1977.*

[97] Nizar Bouguila, Djemel Ziou, and Jean Vaillancourt, "Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and Its Application", *IEEE TRANSACTIONS ON IMAGE PROCESSING*, *VOL. 13, NO. 11, 2004.*

[98] B. D. Fielitz and B. L. Myers, "Estimation of parameters in the Beta distribution," *Decision Sci., vol. 6, pp. 1–13, 1975.*

[99] G. H. Weis and M. Dishon, "Small sample comparison of estimation methods for the Beta distribution," *J. Stat. Comput. Simul., vol. 11, pp. 1–11, 1980.*

[100] K. Samuel, K.W. Ng, and K. Fang, "Symmetric Multivariate and Related Distributions". *London, U.K.: Chapman & Hall, 1990.*

[101] K. Samuel, K. Balakrishman, and J. Norman, "Continous Multivariate Distributions". *New York: Wiley, 2000, vol. 1.*

[102] R. A. Redner and H. F.Walker, "Mixture densities, maximum likelihood and EM algorithm," *SIAM Rev., vol. 26, no. 2, pp. 195–239, Apr. 1984.*

[103] T. J. Santner and D. E. Duffy, "The Statistical Analysis of Discrete Data". *New York: Springer-Verlag, 1989.*

[104] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr., vol. AC-19, pp. 716–723, June 1974.*

[105] J. Rissanen, "Modeling by shortest data description," *Biometrika, vol. 14, pp. 465–471, 1978.*

[106] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat., vol. 6, no. 2, pp. 461–464, 1978.*

[107] S. Ikeda, "Acceleration of the EM algorithm," *Syst. Comput. Jpn., vol. 31, no. 2, pp. 10–18, Feb. 2000.*

[108] T. Minka, "Estimating a Dirichlet Distribution", 2003.

[109] Y. Wu and T. S. Huang, "Towards self-exploring discriminating features for visual learning," *Engineering Applications of AI, vol. 15, pp. 139–150, Sept. 2002.*

[110] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning, vol. 39, pp. 103–134, 1999.*

[111] Selina Chu; Narayanan, S.; Kuo; C.-C.J. "A semi-supervised learning approach to online audio background detection", *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*

[112] P. Rao, "Asymtotic theory of statistical inference", *in Wiley Series in Probability and Mathematical Statistics.* New York: Wiley, 1987.

[113] S.Boutemedjet, N.Bouguila, D.Ziou: "A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering". *IEEE Trans. PAMI. 2009.*

CURRICULUM VITAE

NAME: Mohamed Maher Ben Ismail

ADDRESS: Department of Computer Engineering and Computer Science
University of Louisville Louisville, KY 40292

DOB: Tunis, Tunisia - May 11, 1979

EDUCATION: B.S., Electrical Engineering, 2002 National School of Engineering of Tunisia, Tunis, Tunisia

HONORS AND AWARDS:

1. Invited member – Golden Key International Honor Society.
2. Graduate Research Assistantship – University of Louisville.
3. Promoted to job grade 12 using the HAY evaluation method within STMicroelectronics on March 2004.

AFFILIATIONS:

1. Member of the "Engage 2010" organized by the Office of the Civic Engagement, Leadership and Service. University of Louisville, Spring 2010.
2. Member of the Institute of Electrical and Electronics Engineers (IEEE).
3. Member of the Society of Photographic Instrumentation Engineers (SPIE).

PUBLICATIONS:

JOURNAL PUBLICATIONS:

1. M. Maher Ben Ismail, Hichem Frigui: "Image Database Categorization using Robust Modeling Of Finite Generalized Dirichlet Mixtures". International Journal of Signal and Imaging Systems Engineering (IJSISE). Special Issue on Feature Extraction and Selection for Images Recognition in Large Databases. 2011. (under review)
2. M. Maher Ben Ismail, Hichem Frigui: "Image Database Categorization using Robust Unsupervised Learning Of Finite Generalized Dirichlet Mixture Models". ICIP 2011. (under review)

CONFERENCE PUBLICATIONS:

1. M. Maher Ben Ismail, Hichem Frigui: "Robust and Unsupervised Learning of Finite Generalized Dirichlet Mixture Models". Transactions on Fuzzy Systems journal 2011. (under review)
2. M. Maher Ben Ismail, Hichem Frigui: "Possibilistic Clustering based on Robust Modeling of Finite Generalized Dirichlet Mixture". Twentieth conference of the International Association for Pattern Recognition (IAPR). ICPR 2010.
3. M. Maher Ben Ismail, Hichem Frigui: "Image Database Categorization using Robust Modeling of Finite Generalized Dirichlet Mixture" International Workshop on Image Processing Theory, Tools and Application. IPTA 2010.
4. M. Maher Ben Ismail, Hichem Frigui: "Image Annotation based on Constrained Clustering and Semi-naive Bayesian Model". IEEE symposium on Computers and Communications. ISCC 2009.
5. M. Maher Ben Ismail, Hichem Frigui, Caudill Joshua: "Empirical Comparison of automatic Image Annotation Systems". International Workshop on Image Processing Theory, Tools and Application. IPTA 2008.