

INTER-QUERY LEARNING IN CONTENT-BASED
IMAGE RETRIEVAL

By

IKER GONDRA

Bachelor of Science
Oklahoma State University
Stillwater, Oklahoma
1998

Master of Science
Oklahoma State University
Stillwater, Oklahoma
2002

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2005

COPYRIGHT

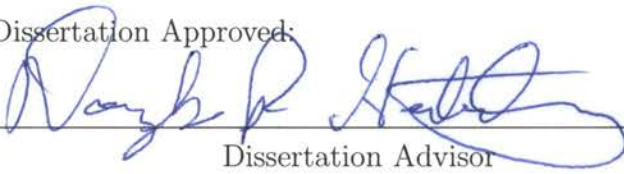
By

Iker Gondra

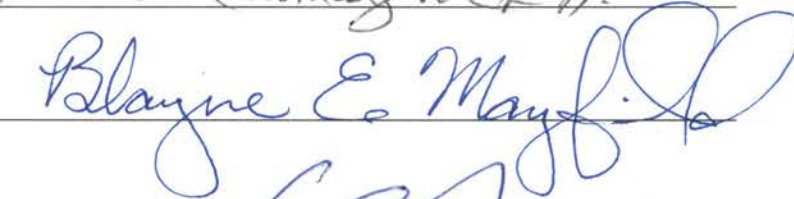
July, 2005

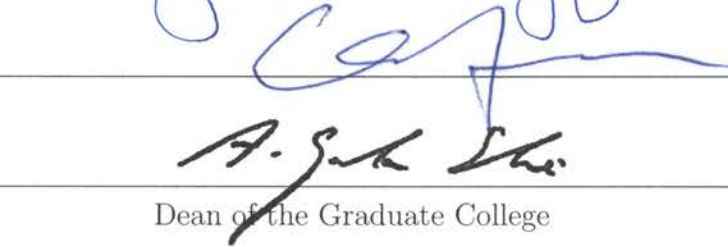
INTER-QUERY LEARNING IN CONTENT-BASED
IMAGE RETRIEVAL

Dissertation Approved:


Dissertation Advisor


Mansur Samalzadeh


Blayne E. Mayfield


Dean of the Graduate College

PREFACE

The rapid development of information technologies and the advent of the World-Wide Web have resulted in a tremendous increase in the amount of available multimedia information. As a result, there is a need for effective mechanisms to search large collections of multimedia data, especially images.

In order to alleviate some of the problems associated with text-based approaches to image retrieval, content-based image retrieval (CBIR) was proposed. The idea is to search on images directly. A set of low-level features, which can be either global or region-based, are extracted from an image to represent its visual content. Retrieval of images is then done by image example where a query image is given as input by the user [130]. The relevance of a database image to the query image is proportional to their feature-based similarity. Those feature representations deemed the most "similar" are returned to the user as the retrieval set. Unfortunately, human notion of similarity is usually based on high-level abstractions, such as activities, events, or emotions displayed in an image. As a result, images with high feature-based similarity may be completely different in terms of user-defined semantics. This discrepancy between low-level features and high-level concepts is known as the *semantic gap* [114].

Relevance feedback (RF) [114] is a supervised learning technique that, by gathering semantic information from user interaction, can reduce the semantic gap and improve retrieval performance. We can distinguish two different types of information provided by RF. The short-term learning obtained within a single query session is intra-query learning. The long-term learning accumulated over the course of many query sessions is inter-query learning. While intra-query learning has been widely used in the literature, less research has been focused on exploiting inter-query learn-

ing.

In this dissertation, the problem of mapping the low-level physical characterization of images to high-level semantic concepts is addressed by focusing on inter-query learning in CBIR with both global and region-based image representations. While the focus is on inter-query learning, novel intra-query learning approaches and image representations are also presented.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my graduate advisor, Dr. Douglas R. Heisterkamp, the best advisor I could have wished for, for his supervision and guidance through the duration of this research, and for his keen interest in my progress. I thank him for introducing me to the topic of Content-Based Image Retrieval and for his invaluable advice along the way.

I would also like to thank Drs. Mansur H. Samadzadeh, Guoliang Fan, and Blayne E. Mayfield, for serving on my dissertation committee and for their valuable input.

I am forever indebted to my parents, family, and friends for their endless encouragement and support.

Table of Contents

1	Introduction	1
1.1	Image Retrieval	1
1.2	Problem Statement	7
1.3	Organization of Dissertation	9
1.4	Notation	14
2	Content-Based Image Retrieval	15
2.1	Introduction	15
2.2	Feature Extraction	17
2.2.1	Image Segmentation	20
2.3	Similarity Measure	26
2.4	Indexing Structure	33
2.5	Relevance Feedback	37
3	Related Work in Machine Learning	40
3.1	Support Vector Machines	40
3.1.1	Risk Minimization	41
3.1.2	Maximal Margin Hyperplanes	45
3.1.3	Non-Linear Classifiers	50
3.1.4	One-Class Support Vector Machines	53
3.1.5	Generalized Support Vector Machines	57

3.2	Multiple-Instance Learning	59
3.2.1	Diverse Density	60
4	Learning with Global Image Representations	64
4.1	Related Work in Intra-Query Learning	65
4.2	Related Work in Inter-Query Learning	69
4.3	Inter-Query Learning with One-Class Support Vector Machines . . .	72
4.3.1	Overview of First Approach	74
4.3.1.1	Summarizing Inter-Query Learning	77
4.3.2	Overview of Second Approach	85
4.3.2.1	Semantic Similarity	89
4.3.2.2	Query Modification and Distance Reweighting	92
4.3.3	Experimental Results	96
5	Learning with Region-Based Image Representations	108
5.1	Related Work in Intra-Query Learning	109
5.2	Probabilistic Region Relevance Learning	112
5.2.1	Region Relevance Measure	113
5.2.2	Estimation of Region Relevance	115
5.2.3	Experimental Results	117
5.3	Intra-Query Learning with Generalized Support Vector Machines . .	121
5.3.1	Experimental Results	125
5.4	Improving Image Segmentation	129
6	Other Image Representations	143
6.1	Image Similarity with Normalized Information Distance	144
6.1.1	The Normalized Information Distance	145
6.1.2	Image Similarity Measure	146
6.1.3	Experimental Results	147

7	Conclusions and Future Work	153
	Bibliography	158

List of Tables

3.1	Common Kernels	52
6.1	<i>Texture</i> Data Set Performance	148
6.2	<i>GroundTruth</i> Data Set Performance	150
6.3	<i>Corel</i> Data Set Performance	151

List of Figures

1.1	Sample Image	2
1.2	Image Representations	3
1.3	General CBIR Computational Framework	4
1.4	A Typical RF Process	5
1.5	Image Retrieval Performance Measures	6
1.6	A Typical Precision-Recall Graph	8
2.1	The RGB Color Model	18
2.2	The HSV Color Model	19
2.3	Sample Human Segmentations	22
2.4	Samples of Segmentation Refinement	24
2.5	Samples of Inconsistent Segmentations	25
2.6	Sample of a Typical Image Segmentation	29
2.7	Integrated Region Matching	31
2.8	Principal Component Analysis of Two-Dimensional Data	35
2.9	M-tree Structure	36
2.10	M-trees with Different Region Volumes and Overlap	37
2.11	Query Shifting	39
3.1	A Simple Binary Classifier	42
3.2	Generalization Performance	43
3.3	VC Dimension	44

3.4	The Perceptron Learning Algorithm	46
3.5	A Simple Linear SVM	48
3.6	Maximum Margin Hyperplane	49
3.7	Support Vectors	50
3.8	Kernel Trick	52
3.9	Sample Hypersphere	55
3.10	Decision Boundaries for 1SVM Methods	56
3.11	Decision Boundaries for 1SVM Methods with Normalized Data	56
3.12	Diverse Density	61
3.13	Sample Diverse Density Space	63
4.1	Sample Feature Relevance	67
4.2	LSI Approach for Inter-Query Learning	70
4.3	Basic Idea of First Approach	74
4.4	Diagram of First Approach	75
4.5	Algorithm of First Approach	78
4.6	Diagram of Modified First Approach	79
4.7	Pre-Image Problem	82
4.8	Method for Estimating Location of Pre-Image	83
4.9	Algorithm of Modified First Approach	85
4.10	Query Modification and Distance Reweighting Framework	86
4.11	PFRL with Query Shifting	87
4.12	Basic Idea of Second Approach	88
4.13	Hypersphere Overlapping	90
4.14	PFRL with Query Shifting and Inter-Query Learning	95
4.15	Sample Images from <i>Texture</i> Data Set	96
4.16	Sample Images from <i>Letter</i> Data Set	97
4.17	Retrieval Performance: Initial Set, First Approach, <i>Texture</i>	99

4.18	Retrieval Performance: Initial Set, First Approach, <i>Letter</i>	100
4.19	Retrieval Performance: 1 RF Iteration, First Approach, <i>Texture</i>	100
4.20	Retrieval Performance: 1 RF Iteration, First Approach, <i>Letter</i>	101
4.21	Retrieval Performance: Data Levels, First Approach, <i>Texture</i>	102
4.22	Sample Retrieval Set with NN Search on <i>Texture</i> Data	102
4.23	Sample Retrieval Set with First Approach on <i>Texture</i> Data	103
4.24	Retrieval Performance: Initial Set, Modified First Approach, <i>Texture</i>	104
4.25	Retrieval Performance: Initial Set, Modified First Approach, <i>Letter</i>	104
4.26	Retrieval Performance: Initial Set, PFRL+ μ_r +1SVM, <i>Texture</i>	106
4.27	Retrieval Performance: 1 RF Iteration, PFRL+ μ_r +1SVM, <i>Texture</i>	106
4.28	Retrieval Performance: Initial Set, PFRL+ μ_r +1SVM, <i>Letter</i>	107
4.29	Retrieval Performance: 1 RF Iteration, PFRL+ μ_r +1SVM, <i>Letter</i>	107
5.1	Region Relevance	114
5.2	PRRL Algorithm	117
5.3	Sample Images from <i>Corel</i> Data Set	118
5.4	Retrieval Performance: RF Iterations, PRRL, <i>Corel</i>	119
5.5	Retrieval Performance: Initial Set, PRRL, <i>Corel</i>	119
5.6	Retrieval Performance: 1 RF Iteration, PRRL, <i>Corel</i>	120
5.7	Retrieval Performance: 2 RF Iterations, PRRL, <i>Corel</i>	120
5.8	Retrieval Performance: 3 RF Iterations, PRRL, <i>Corel</i>	121
5.9	Sample Retrieval Set with PRRL on <i>Corel</i> Data	122
5.10	Decision Boundaries with Valid and Invalid Kernels	123
5.11	GSVM-based RF Learning Algorithm	125
5.12	Retrieval Performance: RF Iterations, GSVM, <i>Corel</i>	127
5.13	Retrieval Performance: Initial Set, GSVM, <i>Corel</i>	127
5.14	Retrieval Performance: 1 RF Iteration, GSVM, <i>Corel</i>	128
5.15	Retrieval Performance: 2 RF Iterations, GSVM, <i>Corel</i>	128

5.16	Retrieval Performance: 3 RF Iterations, GSVM, <i>Corel</i>	129
5.17	Simple Segmentation Algorithm	131
5.18	Basic Idea of MIL-based Approach	133
5.19	Multiple DD Maximizers	134
5.20	Threshold for DD Maximizers	134
5.21	Sample Image with Multiple Semantics	135
5.22	Association of Regions with DD Maximizers	136
5.23	Sample of Under-Segmentation	138
5.24	One-To-One Mapping Between Region and DD Maximizer	138
5.25	Sample of Poor Segmentation	139
5.26	One-To-Many Mapping Between Region and DD Maximizer	139
5.27	Sample of Over-Segmentation	140
5.28	Many-To-Many Mapping Between Region and DD Maximizer	141
5.29	Sample of Under-Segmentation	141
5.30	Algorithm for MIL-based Segmentation	142
6.1	Sample Images from <i>GroundTruth</i> Data Set	148
6.2	Sample Query Images from <i>IAPR-12</i> Data Set	149
6.3	JPEG Compression	151

Chapter 1

Introduction

A picture is worth a thousand words

1.1 Image Retrieval

The rapid development of information technologies and the advent of the World-Wide Web have resulted in a tremendous increase in the amount of available multimedia information. As a result, there is a need for effective mechanisms to search large collections of multimedia data (e.g., image, audio, video). The management of text information has been studied thoroughly and there have been many successful approaches for handling text databases (see [121]). However, the progress in research and development of multimedia database systems has been slow due to the difficulties and challenges of the problem. Of particular interest to us are images.

The development of concise representations of images that can capture the essence of their visual content is an important task. However, as the above saying suggests, representing visual content is a very difficult task. The human ability to extract semantics from an image by using knowledge of the world is remarkable, though probably difficult to emulate.

At present, the most common way to represent the visual content of an image

is to assign a set of descriptive keywords to it. Then, image retrieval is performed by matching the query text with the stored keywords [117]. However, there are many problems associated with this simple keyword matching approach. First, it is usually the case that all the information contained in an image cannot be captured by a few keywords. Furthermore, a large amount of effort is needed to do keyword assignments in a large image database. Also, because different people may have different interpretations of an image's content, there will be inconsistencies [117]. Consider the image in Figure 1.1. One might describe it as “mountains”, “trees”, and “lake”. However, that particular description would not be able to respond to user queries for “water”, “landscape”, “peaceful”, or “water reflection”.



Figure 1.1: Sample image.

In order to alleviate some of the problems associated with text-based approaches, content-based image retrieval (CBIR) was proposed (see [27, 28] for examples of early approaches). The idea is to search on the images directly. A set of low-level features (such as color, texture, and shape) are extracted from the image to characterize its visual content. In traditional approaches [27, 28, 45, 49, 67, 91, 105, 123, 126, 132, 133, 136, 147], a set of global features are extracted from the image. The features are then the components of a feature vector which makes the image correspond to a point in a feature space (See Figure 1.2). In contrast to traditional methods, which extract global image features, region-based approaches [15, 17, 73, 75, 81, 134, 146]

extract features from segmented regions of an image. The main objective of using regions is to do a more meaningful retrieval that is closer to a user's perception of an image's content. That is, instead of looking at the image as a whole, we look at its objects and their relationships (See Figure 1.2).

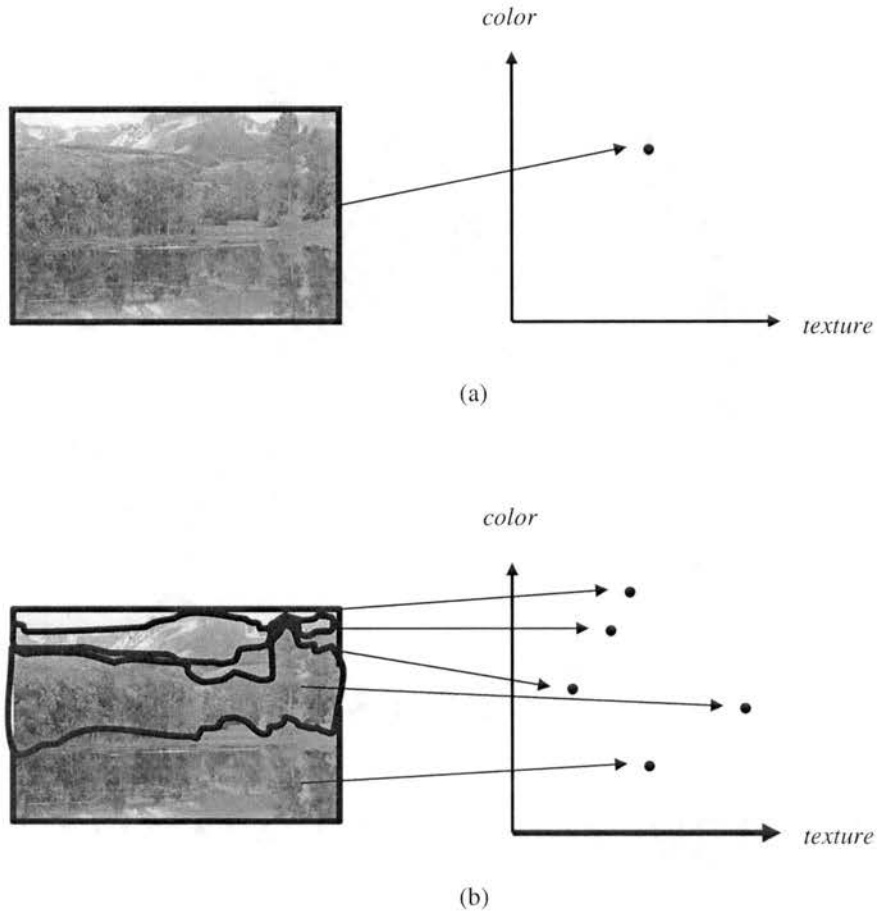


Figure 1.2: Image representations: a) global: a set of global features is extracted and the image is represented by a single point in feature space; b) region-based: a set of local features is extracted from each segmented region and the image is represented by a (variable) number of points in feature space.

Retrieval of images in CBIR is done by image example where a query image is given as input by the user [130]. Thus, the system views the query and database images as a collection of features. The relevance of a database image to the query image is then proportional to their feature-based similarity. The general computa-

tional framework of a CBIR system is depicted in Figure 1.3. In order to create the image database, images are processed by a feature extraction algorithm and their feature representations are stored in the database. The same feature extraction algorithm is used to obtain the features that represent the query image. The similarity measure compares the representation of the query image with the representation of each database image. Those feature representations deemed the most “similar” are returned to the user as the retrieval set. The selection of an appropriate similarity measure is also an important problem. Different similarity measures will affect retrieval performance significantly. Since visual content can be represented by different attributes, the combination of and importance of each set of features has to be considered. In addition, the similarity measure should be adaptive so that it can accommodate the preferences of different users.

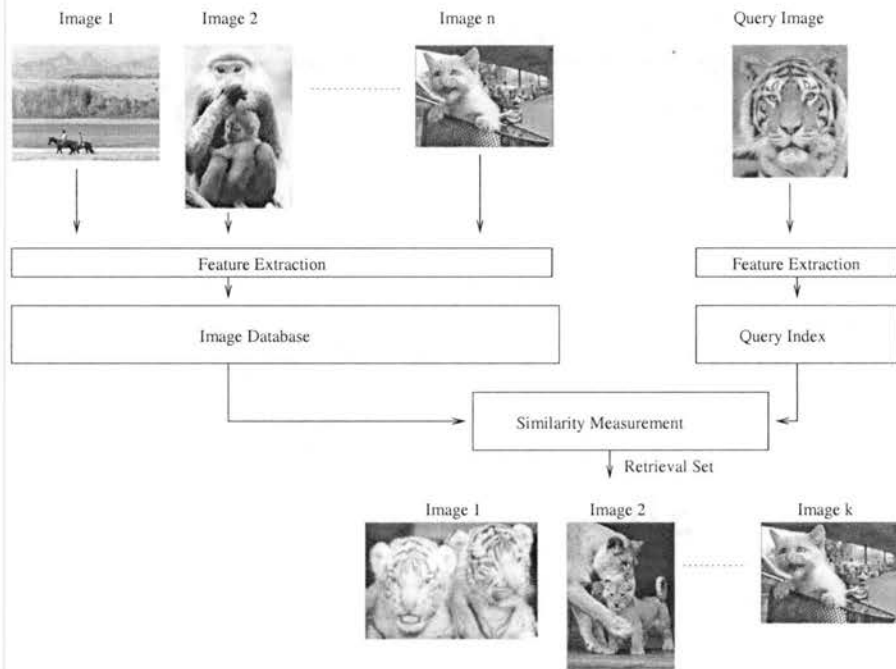


Figure 1.3: General CBIR computational framework.

There are also problems with this general CBIR computational framework. The human notion of similarity is usually based on high-level abstractions such as activities, events, or emotions displayed in an image. Therefore, a database image with a

high feature similarity to the query image may be completely different from the query in terms of user-defined semantics. This discrepancy between low-level features and high-level concepts is known as the *semantic gap* [130].

Relevance Feedback (RF), originally developed for information retrieval [114], has been proposed as a learning technique aimed at reducing the semantic gap. It works by gathering semantic information from user interaction. Based on the user’s feedback on the retrieval results, the retrieval scheme is adjusted. Thus, by providing an image similarity measure under human perception, RF can be seen as a form of supervised learning. In order to learn a user’s query concept, the user labels each image returned in the previous query round as relevant or non-relevant. Based on the feedback, the retrieval scheme is adjusted and the next set of images is presented to the user for labelling. This process iterates until the user is satisfied with the retrieved images or stops searching. Figure 1.4 shows a typical RF process.

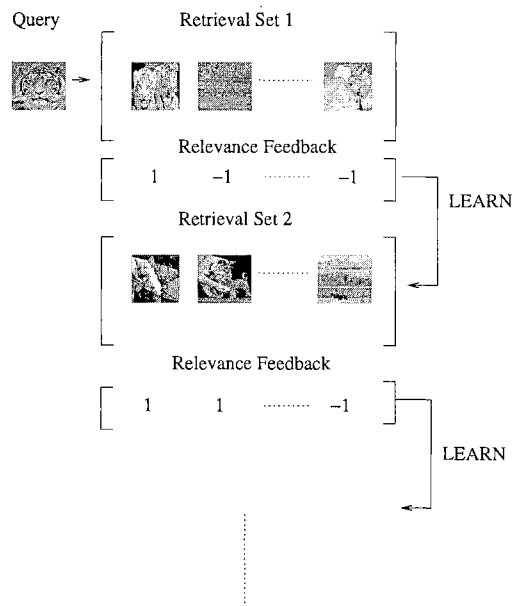


Figure 1.4: A typical RF process.

Precision and *recall* are common measures that are used to evaluate the performance of an image retrieval system. Consider an image database consisting of a set of images \mathcal{D} . Let \mathbf{q} be a query image and $\mathcal{A} \subset \mathcal{D}$ be the subset of images in \mathcal{D} that are

relevant to \mathbf{q} . Assume that a given image retrieval strategy processes \mathbf{q} and generates $\mathcal{R} \subset \mathcal{D}$ as the retrieval set. Then, $\mathcal{R}^+ = \mathcal{R} \cap \mathcal{A}$ is the set of relevant images to \mathbf{q} that appear in \mathcal{R} . Similarly, $\mathcal{R}^- = \mathcal{R} - \mathcal{A}$ is the set of non-relevant images to \mathbf{q} that appear in \mathcal{R} . Figure 1.5 illustrates these sets. The precision and recall measures are as follows

1. Precision measures the ability to retrieve only relevant images. It is defined as

$$\text{Precision} := \frac{|\mathcal{R}^+|}{|\mathcal{R}|} \quad (1.1)$$

2. Recall measures the ability to retrieve all relevant images. It is defined as

$$\text{Recall} := \frac{|\mathcal{R}^+|}{|\mathcal{A}|} \quad (1.2)$$

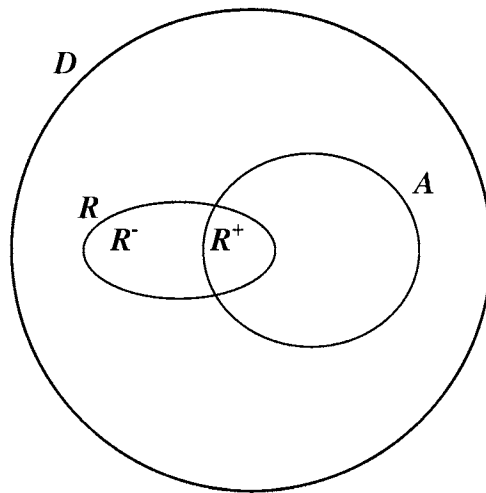


Figure 1.5: Image retrieval performance measures: \mathcal{D} is the set of all database images; \mathcal{A} is the set of all images relevant to a query; \mathcal{R} is the retrieval set in response to the query; precision is $|\mathcal{R}^+|/|\mathcal{R}|$; recall is $|\mathcal{R}^+|/|\mathcal{A}|$.

Both high recall and high precision are desirable, though not often obtainable. That is, in many cases, improvement of one leads to the deterioration of the other.

Note that perfect recall could be achieved simply by letting $\mathcal{R} = \mathcal{D}$ (i.e., by retrieving all images in the database in response to \mathbf{q}). However, obviously, users would probably not be happy with this approach. Thus, recall by itself is not a good measure of the performance of an image retrieval system. Instead, users want the database images to be ranked according to their relevance to \mathbf{q} and then be presented with only the k most relevant images so that $|\mathcal{R}| = k < |\mathcal{D}|$. Therefore, in order to account for the quality of image rankings, precision at a cut-off point (e.g., k) is commonly used. For example, if $k = 20$ and the top 20 ranked images are all relevant to \mathbf{q} , then \mathcal{R} contains only relevant images and thus precision is 1. On the other hand, if $k = 40$ and only the first top 20 images are all relevant to \mathbf{q} , then half of the images in \mathcal{R} are non-relevant to \mathbf{q} and thus precision is only 0.5. A common way to depict the degradation of precision as k increases is to plot a precision-recall graph. Figure 1.6 shows a typical precision-recall graph. This graph shows the tradeoff between precision and recall. That is, attempting to increase recall results in the introduction of more non-relevant images into \mathcal{R} , thus decreasing precision. Ideally, we would like improvements in the image retrieval system to result in the precision-recall curve moving upwards and towards the right (i.e., both high precision and high recall).

1.2 Problem Statement

Since its introduction to CBIR by Minka [96], RF has been incorporated into a variety of systems. However, most do not implement one of the main goals set forth by Minka - the ability to apply what is learned from past RF interactions to the current task. In most current systems, all prior experience is lost. The retrieval strategy is refined by using only RF supplied by the current user and the learning process starts from ground up for each new query. That is, the system only takes into account the current query session without using any long-term learning. Thus, these systems are based on the assumption that users are willing to patiently perform several iterations of RF

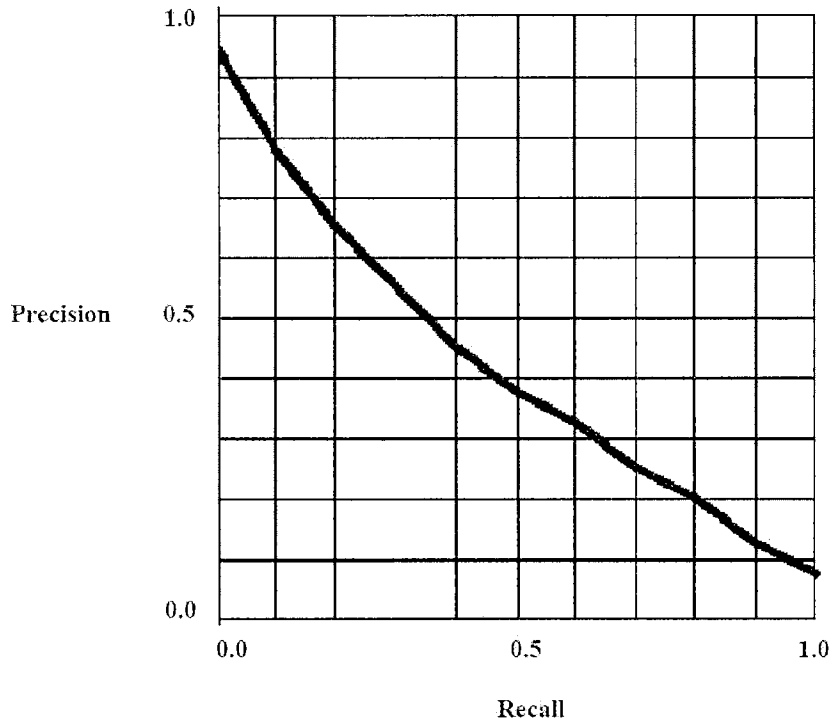


Figure 1.6: A typical precision-recall graph.

for each query.

We can distinguish two different types of information provided by RF. The short-term learning obtained within a single query session is *intra-query learning*. The long-term learning accumulated over the course of many query sessions is *inter-query learning*. By accumulating knowledge from users, long-term learning can be used to enhance future retrieval performance. The fact that two images were regarded as similar by a previous user is a cue for similarities in their semantic content. This is because, although different people may associate the same image into different concepts, there is some common semantic agreement. While short-term learning has been widely used in the literature, less research has been focused on exploiting long-term learning.

In this dissertation, the problem of mapping the low-level physical characterization of images to high-level semantic concepts is addressed by focusing on inter-query

learning in CBIR with both global and region-based image representations. While the focus is on inter-query learning, we also present some novel intra-query learning approaches and image representations. The following are some of the key issues that are addressed:

- What learning approaches can be used to exploit the information that is obtained during the RF process? What long-term learning structures can be used to represent and memorize this knowledge?
- How to handle different interpretations of the semantics of an image?
- How can we combine intra and inter-query learning in an adaptive manner? There may be situations in which it may be advantageous to rely more heavily on one type of learning. For instance, at the beginning, when only a few queries have been processed, inter-query learning can be unreliable and we may want to depend more on intra-query learning. Similarly, as more queries are processed and experience accumulates, it may be advantageous to rely more on inter-query learning. Thus, it is desirable to have a principled way for exploiting intra and inter-query learning that adapts to the current situation.
- How can we exploit inter-query learning in an efficient manner? If the memorization and exploitation of learned knowledge results in a large increase in space and/or time requirements, we may have to question whether the advantages of using inter-query learning justify this. Thus, we must ensure that inter-query learning does not result in large overhead. A compact representation with good generalization performance is desirable.

1.3 Organization of Dissertation

The remainder of this dissertation is organized as follows.

- **Chapter 2. Content-Based Image Retrieval**

In this chapter, we give an overview of CBIR and review the following important issues: feature extraction (i.e., how to represent the visual content of an image), similarity measure (i.e., how to decide the similarity of two images), indexing techniques (i.e., how to search images efficiently), and RF (i.e., how to reduce the semantic gap).

- **Chapter 3. Related Work in Machine Learning**

The field of machine learning focuses on the study of algorithms that improve their performance at some task automatically through experience [97]. In this chapter, we summarize two machine learning techniques, support vector machine, and multiple instance learning, which will be used in this dissertation.

- **Chapter 4. Learning with Global Image Representations**

In this chapter, we first summarize related work on intra and inter-query learning with global image representations. Next, we present two novel techniques for performing inter-query learning with global image representations. Both techniques use support vector machines for learning the class distributions of users' high-level query concepts from retrieval experience. They are based on a RF framework that learns one-class support vector machines from retrieval experience to represent the set memberships of users' high-level query concepts and stores them in a "concept database". The "concept database" provides a mechanism for accumulating inter-query learning obtained from previous queries. The geometric view of one-class support vector machines allows a straightforward interpretation of the density of past interaction in a local area of the feature space and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

The first approach, presented in [35, 36, 40, 42], does a fuzzy classification of

a new query into the regions of support represented by the one-class support vector machines in the “concept database”. In this way, past experience is merged with current intra-query learning. The second approach, presented in [39], incorporates inter-query learning into the query modification and distance reweighing framework. One of the main advantages of these approaches is the capability of making an intelligent initial guess on a new query when the query is first presented to the system.

- **Chapter 5. Learning with Region-Based Image Representations**

In this chapter, we first summarize related work on intra-query learning with region-based image representations. Next, we present two novel intra-query learning approaches for CBIR with region-based image representations. The first approach, probabilistic region relevance learning [38], is based on the observation that regions in an image have unequal importance for computing image similarity. It automatically estimates region relevance based on user’s feedback. It can be used to set region weights in region-based image retrieval frameworks that use an overall image-to-image similarity measure.

The second approach, presented in [37], is based on support vector machine learning. Traditional approaches based on support vector machine learning require the use of fixed-length image representations (i.e., global representations) because support vector machine kernels represent an inner product in a feature space that is a non-linear transformation of the input space. However, many CBIR methods that use region-based image representations create a variable-length image representation and define a similarity measure between two variable-length representations. Thus, the standard support vector machine approach cannot be applied because it violates the requirements that a support vector machine places on the kernel. Fortunately, a generalized support vector machine [84] has been developed that allows the use of an arbitrary kernel.

We present a learning algorithm based on generalized support vector machines. Since a generalized support vector machine does not place restrictions on the kernel, any image similarity measure can be used.

Next, we present an intra/inter-query learning approach that addresses the problem of semantically-meaningful image segmentation. A large number of image segmentation techniques have been proposed in the literature. However, most image segmentation algorithms create regions that are homogeneous with respect to one or more low-level features according to some similarity measure. Unfortunately, homogeneous regions based on low-level features usually do not correspond to meaningful objects. To the best of our knowledge, no approach has been proposed that exploits intra/inter-query learning for automatically improving image segmentation. We propose an algorithm based on multiple-instance learning [25, 85, 87] that exploits both intra and inter-query learning for automatically improving the segmentation of images in a database. The main advantage of this approach is that it can automatically refine the segmentation of images into semantically-meaningful objects.

- **Chapter 6. Other Image Representations**

The main idea of CBIR is to search on images directly. That is, instead of searching based on assigned keywords, it is preferable to search visual content directly. However, we still need to use a set of features to represent visual content. In this chapter, we present our initial investigation into what we believe is the logical continuation of the CBIR idea of searching visual content directly. It is based on the observation that, since ultimately, the entire visual content of an image is encoded into its raw data (i.e., the raw pixel values), in theory, it should be possible to determine image similarity based on the raw data alone. That is, everything that we need to know regarding the visual content of the image is in the raw data itself. Humans are very good at looking at an

image (i.e., the raw data) and extracting all the important features. Thus, all the important features are “hidden” in the raw data. The problem of feature extraction is just that we do not entirely know yet how (we, humans) “find” them. Thus, instead of attempting to determine image similarity based on a (probably incomplete) set of features, why not have a similarity measure that is based on the raw data itself (since everything is in the raw data). We present an initial investigation, conducted in [41], into an image dissimilarity measure following from the theoretical foundation of the recently proposed normalized information distance [74]. A very crude approximation of the Kolmogorov complexity of an image is created by compression. Using this approximation, we can calculate the normalized information distance between images and use it as a metric for CBIR.

- **Chapter 7. Conclusions and Future Work**

In this chapter, we summarize the contributions of this dissertation on exploiting both intra-query and inter-query learning to improve the performance of CBIR. We also examine the limitations of the proposed approaches and suggest some directions for future research.

1.4 Notation

Throughout this dissertation, the following notational conventions will be used. A lowercase italic roman or greek letter will refer to a scalar, for example, a , or α . In Chapter 6, a lowercase italic roman letter will also refer to a string. A boldface lowercase letter will refer to a vector, for example \mathbf{x} . For a vector \mathbf{x} , $\|\mathbf{x}\|$ denotes its 2-norm (i.e., Euclidean norm). An uppercase boldface letter will refer to a matrix, for example, \mathbf{M} . For a matrix \mathbf{M} , \mathbf{M}^{-1} denotes its inverse. The superscript T in for example \mathbf{M}^T , stands for the transpose of matrix \mathbf{M} . The dot product of two vectors \mathbf{a} and \mathbf{b} will be denoted by $\mathbf{a} \cdot \mathbf{b}$, or $\mathbf{a}^T \mathbf{b}$. Functions will be distinguished by always taking in parameters, for example $f(x)$, $K(\mathbf{x}, \mathbf{y})$, or $\Phi(\mathbf{x})$. A calligraphic uppercase letter will refer to a set, for example \mathcal{S} . For a set \mathcal{S} , $|\mathcal{S}|$ refers its cardinality. The subscript i denotes the i -th component of a vector or the i -th element of a set, for example x_i . For a set \mathcal{S} , the notation $\mathcal{S} = \{x_i\}_a^b$ is shorthand for $\mathcal{S} = \{x_a, x_{a+1}, \dots, x_{b-1}, x_b\}$.

Chapter 2

Content-Based Image Retrieval

In this chapter, we give an overview of content-based image retrieval (CBIR) and review the following important issues: feature extraction (i.e., how to represent the visual content of an image), similarity measure (i.e., how to decide the similarity of two images), indexing techniques (i.e., how to search images efficiently), and relevance feedback (RF) (i.e., how to reduce the semantic gap).

2.1 Introduction

As described in Chapter 1, early approaches to image retrieval were mainly text-based techniques consisting on the manual annotation of images with descriptive keywords. This manual annotation is very time consuming and cumbersome for large image databases. Furthermore, it is very subjective and error-prone. Recently, some approaches for automatic image labelling [100, 128, 135] have been proposed as an attempt to improve this manual annotation process. In [100], image recognition techniques are used for automatically assigning descriptive keywords to images. Their approach uses only a limited number of keywords. Furthermore, because image recognition techniques are not completely reliable, automatically assigned keywords still have to be verified by a human. In [128], the textual context of images in a web

page is used to automatically extract descriptive keywords. The collateral text that usually accompanies an image (e.g., captions) is exploited in [135]. The performance of those approaches is not as high as that obtained with manual annotation and their applicability is limited in situations where there is no textual context (e.g., a photo album). In [149] a semi-automatic annotation that assigns images to keywords based on users' RF is proposed. Their approach uses both keyword and content-based retrieval strategies. A weighted sum of the keyword-based and visual feature-based similarity measures is used to calculate the overall similarity of an image. Based on the user's RF, the annotation of each image in the retrieval set is updated. The experiments conducted in [149] indicate that this strategy of semi-automatic annotation outperforms manual annotation in terms of efficiency and automatic annotation in terms of accuracy. However, the performance of this approach depends heavily on the performance of the particular CBIR and RF algorithms used, specially when there is no initial annotation in the database at all [149].

In order to overcome the above-mentioned drawbacks associated with text-based approaches, it would be more suitable to search on the images directly based on their visual content (in Chapter 6 we present our initial investigation on what we believe is the logical continuation of the idea of searching on images directly). In the early 1990's, CBIR was proposed as a way of allowing a user to search target images in terms of their content represented by visual features. Since then, many CBIR systems have been developed including Blobworld[15], QBIC[27], IRM[73], NeTra[81], MARS[91], Photobook[105], WebSEEK[133], and SIMPLiCity[146], just to name a few.

Retrieval of images in CBIR is done by image example where a query image is given as input by the user [130]. Thus, the system views the query and database images as a collection of features. The relevance of a database image to the query image is then proportional to their feature-based similarity. In general, a CBIR system involves three major issues: feature extraction, similarity measure, and indexing structure

(See Figure 1.3).

2.2 Feature Extraction

Feature (content) extraction is the basis of CBIR. In traditional approaches [27, 28, 45, 49, 67, 91, 105, 123, 126, 132, 133, 136, 147], a CBIR system extracts a single set of global features (such as color, texture, and shape) from an image. The features are then the components of a feature vector which makes the image correspond to a single point in a feature space (See Figure 1.2).

Color is one of the most widely used visual features. The color histogram is a popular image feature that has been exploited by many CBIR systems [43]. It is a very simple description of the distribution of colors in an image. It is also usually invariant to translation and rotation of an image. However, histograms do not include any spatial information so images with different layouts may have the same histogram. Early work on color includes color indexing using histogram intersection [80]. The representation of color is an important issue. In the RGB (Red-Green-Blue) color space, color is labelled as relative weights of the three primary colors. In this system (0,0,0) is black, (1,1,1) is white, and the space of all available colors is represented by a cube (See Figure 2.1). While this color space is the most commonly used, it does not model human perception of color. For example, what is the RGB value of “medium green”? Once a color has been chosen (e.g., “green”), how to make subtle changes to it is not obvious. The HSV (Hue-Saturation-Value) color space [131] provides a better model of human perception of color. It is usually represented as a double cone (See Figure 2.2) which is a non-linear transformation of the RGB cube. In order to define a color, the perceptually based variables Hue, Saturation, and Value are used. The axis of the cone represents the intensity/value. Hue is represented by the angle around the vertical axis and saturation is given by the distance to the central axis. In this model, varying Hue corresponds to selecting a color, decreasing Value corresponds to adding

black, and decreasing Saturation corresponds to adding white. Making subtle changes to a color is much easier when these perceptual variables are used. It is important to note that the set of all colors in both the RGB and HSV color space is a subset of the colors that can be perceived by humans. The CIE (Commission Internationale de l'Eclairage), which stands for International Commission on Illumination, defined the XYZ color space in 1931 [29]. This space embraces all colors that can be perceived by humans. Every color in this space is defined by three standard primaries (X, Y, and Z) that replace red, green, and blue. The primary Y closely matches the quality of *luminance* of a color [29]. The CIE LUV color space is a derivation of this color space in which two colors are equally distant in color space whenever they are equally distant perceptually [29].

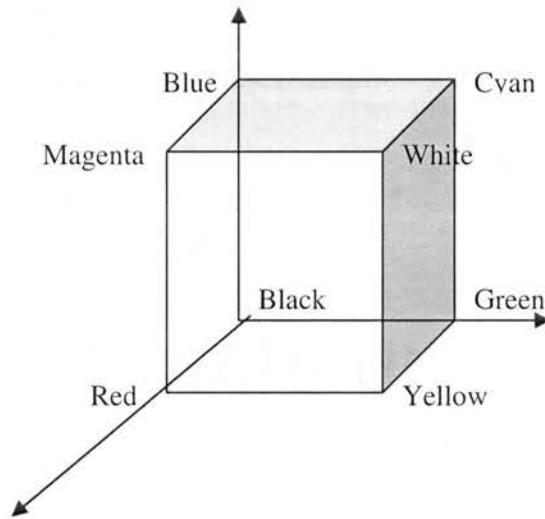


Figure 2.1: The RGB color model.

Texture is another image feature that has been intensively explored [43]. It refers to the patterns in an image representing the homogeneity properties that do not result from the presence of a single color. The well-known Tamura features [138] include *coarseness*, *contrast*, *directionality*, *line-likeness*, *regularity*, and *roughness*. These visual texture properties were found to be important in psychological studies

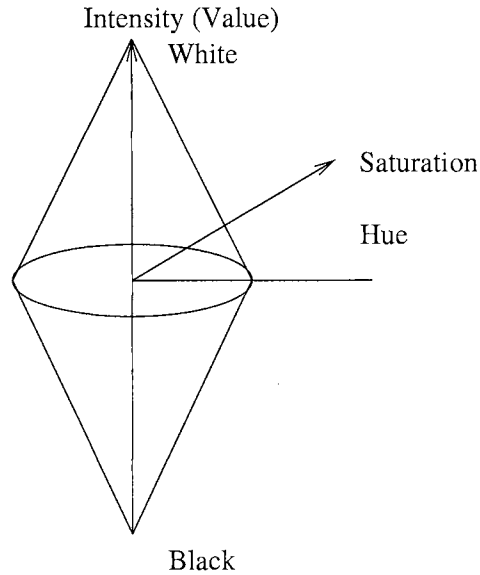


Figure 2.2: The HSV color model.

[138]. A co-occurrence matrix representation was proposed in [50]. It is based on the construction of a co-occurrence matrix based on the orientation and distance between image pixels and on the extraction of meaningful statistics from this matrix as a representation of texture. VisualSeek [132] and WebSeek [133] were both developed at Columbia university. They are web-based text/image search engines that use color and texture features.

While color and texture are global attributes of an image, shape requires some kind of image segmentation and region identification process. In contrast to traditional methods, which extract global image features, region-based approaches [15, 17, 73, 75, 81, 134, 146] extract features from segmented regions of an image. The image is then represented by a (variable) number of points in feature space (See Figure 1.2). The main objective of using regions is to do a more meaningful retrieval that is closer to a user's perception of an image's content by looking at its objects and relationships. Fourier descriptors [106] and moment invariants [110] are well known shape representations [118]. Shape representations can be either boundary-based (e.g., Fourier descriptors) or region-based (e.g., moment invariants) [43]. Boundary-based represen-

tations use the outer boundary of the shape and region-based representations use the entire shape of a region. It is important that the shape representation be invariant to translation, scaling, and rotation. A modified Fourier descriptor that is translation, scaling, and rotation invariant was proposed in [120]. Note that for region-based approaches, it is very important to be able to properly identify the objects in an image by performing a good segmentation.

2.2.1 Image Segmentation

Many algorithms have been proposed for image segmentation. However, robust and accurate image segmentation remains a difficult problem. A review of many early image segmentation techniques can be found in [51, 101] and a review of more recent ones in [78, 79]. In edge-based approaches [47, 150], segmentation is based on spatial discontinuities. That is, by detecting sudden changes in local features, region boundaries can be obtained. On the other hand, segmentation in region-based approaches [129] is based on spatial similarity among pixels. Thus, a measure of region homogeneity has to be defined in advance. There are two main region-based approaches: region-growing and split-and-merge. In region-growing approaches, a number of uniform regions is defined in advance and surrounding pixels are merged into one of the regions according to the homogeneity criteria. On the other hand, in split-and-merge approaches, regions that are non-uniform according to the homogeneity criteria are broken down into smaller regions until all regions are uniform. Then, neighboring regions that are close in feature space are merged. Clustering-based approaches classify pixels into one of several clusters. The classical k -means [90] algorithm is probably one of the best known and most commonly used methods for clustering data. Recently, modified versions of this algorithm (e.g., fuzzy k -means [107]) have been proposed to improve its robustness and efficiency. Among the many segmentation algorithms, a Normalized Cuts framework is introduced in [129]. This framework is capable of

detecting clusters of various shapes and is an example of a clustering-based approach derived from graph theory. Hopfield artificial neural networks are used in [14, 58, 62] to solve the image segmentation problem.

A large number of image segmentation techniques have been proposed in the literature. However, there is a lack of work on evaluating and comparing the performance of the various techniques. The first extensive survey on image segmentation evaluation methods was presented in [159]. A more up-to-date review of recent progress on this area was given in by the same author in [160]. In [160], a scheme for classifying evaluation methods for image segmentation is proposed. According to this scheme, evaluation methods can be classified into three distinct groups: *analytical*, *empirical goodness*, and *empirical discrepancy* methods. Analytical methods consider characteristics (e.g., complexity, requirements, etc...) of segmentation algorithms. These methods can contribute only some additional information to that obtained by other methods and thus, are seldom used in isolation [160]. The empirical goodness methods evaluate a segmentation based on some intuitive measure of goodness (e.g., uniformity within regions, contrast between regions). Finally, the discrepancy methods make use of “ground truth” (i.e., ideal) segmentations to assess the performance of a segmentation algorithm based on how different the segmentation that it generates is from a “ground truth” segmentation of the same image. Comparative experiments indicate that these methods are better than the goodness methods [160]. Many researchers believe that human assessment of segmentation results is best. In fact, [101] indicates that a person is the best judge for evaluating an image segmentation.

As indicated in [88, 89], the major challenge in using “ground truth” segmentations is that the question “What is a correct segmentation?” is very subtle. That is, segmentations of an image produced by different people may not be identical (See Figure 2.3). Therefore, how can we make a reliable evaluation of a segmentation algorithm when there is not a single “ground truth” set of segmentations that we can

use to compare against?

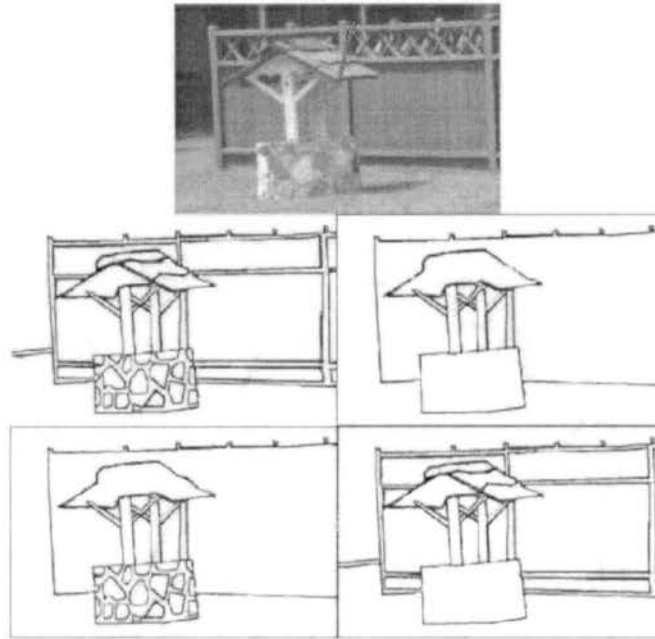


Figure 2.3: Segmentations of an image produced by different people may not be identical.

The thesis of [88, 89] is that, even though segmentations of an image by different people are not identical, there is considerable consistency among them. In [88, 89], it is demonstrated empirically that differences in segmentations are due to the fact that, even though two observers have exactly the same perceptual organization of an image, they may choose to segment at varying levels of granularity. This suggests that a good segmentation error measure should penalize differences that arise from different perceptual organizations of the image. However, if one segmentation is simply a refinement of the other, then the error should be small [88, 89]. In [88, 89], a “ground truth” database containing “ground truth” segmentations generated by humans for images of a wide variety of natural scenes is obtained. Then, an error measure which quantifies consistency (in terms of similar perceptual organization) between segmentations of differing granularity is proposed. It is found that different human segmentations of the same image are highly consistent (See Figure 2.4). As

a result, the potential problem of not having a unique segmentation of an image is eliminated.

Based on the assumption that all people share the same perceptual organization of an image, we can model any perception of a scene as a tree, which is called the *percept tree* in [88, 89] (See Figure 2.4). Thus, any two (consistent) segmentations of an image must represent a cut through some percept tree. Therefore, for any particular pixel in the image, the regions in the two segmentations that contain the pixel must have a subset relationship. Otherwise, if one region does not contain the other, they cannot share a common percept tree and the segmentations are inconsistent [88, 89] (See Figure 2.5). The *Local Refinement Error* $E(S_1, S_2, p_i)$ [88, 89], which tolerates refinement but not overlapping, measures the degree to which two segmentations S_1 and S_2 agree at pixel p_i

$$E(S_1, S_2, p_i) = \frac{|\mathcal{P}_{S_1, p_i} - \mathcal{P}_{S_2, p_i}|}{|\mathcal{P}_{S_1, p_i}|}$$

where \mathcal{P}_{S, p_i} is the set of pixels in segmentation S which are in the same region as pixel p_i , and $-$ denotes set difference. Note that this quantity is not symmetric. The *Local Consistency Error* $LCE(S_1, S_2)$ [88, 89] allows refinement in both directions

$$LCE(S_1, S_2) = \frac{1}{n} \sum_{i=1}^n \min(E(S_1, S_2, p_i), E(S_2, S_1, p_i))$$

where n is the number of pixels in the image. Note that, for different parts of the image, this measure allows refinement in different directions. The *Global Consistency Error* (GCE) [88, 89] forces all refinements to be in the same direction

$$GCE(S_1, S_2) = \frac{1}{n} \min \left(\sum_{i=1}^n E(S_1, S_2, p_i), \sum_{i=1}^n E(S_2, S_1, p_i) \right)$$

Note that $GCE > LCE$. Because mutual refinements are common, LCE is

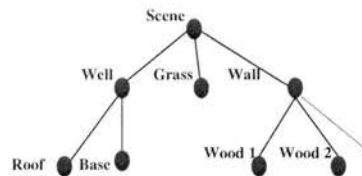
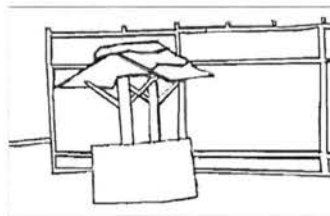
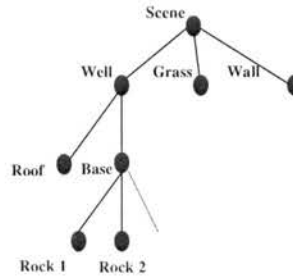
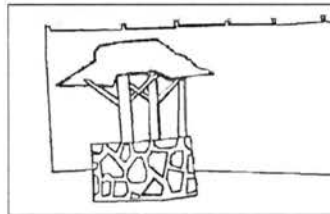
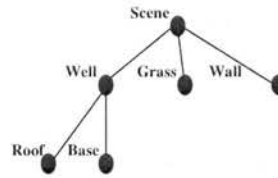
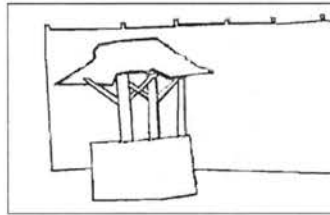
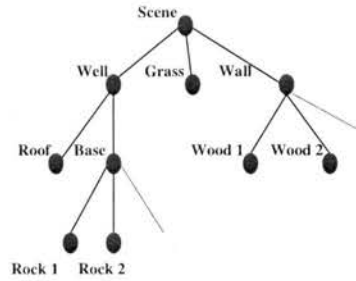


Figure 2.4: Although different, human segmentations of an image are not inconsistent because (it is presumed that) they share the same *percept tree* (to the right of each image). Variation is just due to different amounts of refinement in the segmentation of each object in an image.

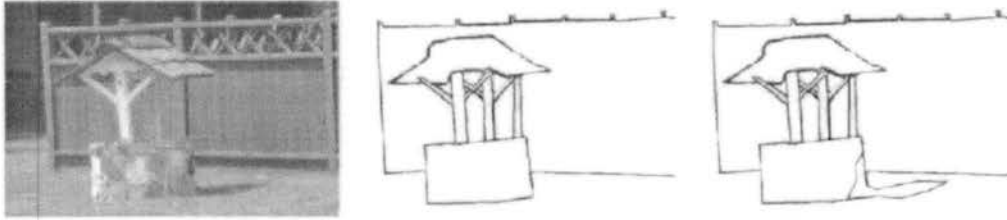


Figure 2.5: Samples of inconsistent segmentations. In this case, there is not a common *percept tree* that can explain the two different segmentations of the image on the left. The segmentation of the shadow results in overlapping, rather than nested, regions.

preferred over *GCE* in [88, 89]. However, a degenerate segmentation that has either one region for the entire image or one region for each pixel will have a zero *LCE* when compared to any other segmentation. In [88, 89], it is found that the distribution of *LCE* over the dataset for same-image pairs is unimodal, peaked at zero, and separable from the distribution of different-image pairs thus providing evidence that human segmentations of an image are consistent.

The *LCE* measure allows refinement in both directions. Therefore, it is too lenient for evaluating the output of a segmentation algorithm. By simply replacing the pixelwise minimum with a maximum the *Bidirectional Consistency Error* (*BCE*) [88], which does not tolerate refinement at all, is obtained

$$BCE(S_1, S_2) = \frac{1}{n} \sum_{i=1}^n \max(E(S_1, S_2, p_i), E(S_2, S_1, p_i))$$

In order to measure the consistency of a segmentation S produced by an algorithm with all human segmentations S_j of the image, the *BCE* measure can be extended as follows [88]

$$BCE^*(S) = \frac{1}{n} \sum_{i=1}^n \min_j \max(E(S, S_j, p_i), E(S_j, S, p_i))$$

It is important to point out that another way of evaluating a segmentation algorithm, which is not mentioned in [160], would be based on the performance of the

application that uses the particular segmentation algorithm. For example, in image retrieval, performance measures such as precision (1.1) and recall (1.2) can be used to evaluate the goodness of a segmentation algorithm. That is, if image retrieval performance improves using a particular segmentation algorithm then, for that particular application, the algorithm is better regardless of whether or not the segmentations it produces are good under human evaluation. To the best of our knowledge, no approach has been proposed that exploits RF for automatically improving image segmentation. In Chapter 5 we propose an intra/inter-query learning method for automatically improving image segmentation.

The selection of an appropriate similarity measure is also an important problem. Different similarity measures will affect retrieval performance significantly. Since visual content can be represented by different attributes, the combination of and importance of each set of features has to be considered. In addition, the similarity measure should be adaptive so that it can accommodate the preferences of different users.

2.3 Similarity Measure

In order to form the retrieval set in response to a query, we need to measure similarity between images. The similarity measure compares the feature representation of the query image with that of each database image. Then, images whose feature representations are deemed the most similar are returned to the user as the retrieval set. When retrieving similar images based on color, most existing techniques use a color histogram generated from the entire image [63]. In [80], image similarity was based solely on color. The distribution of color was represented by color histograms. The similarity between two images was then based on a similarity measure between their corresponding histograms called the “normalized histogram intersection”.

Conversely, we can measure distance between images. In this case, small distances

between feature representations correspond to large similarities and large distances correspond to small similarities. Thus, distance is a measure of dissimilarity. One way to transform between a distance measure and a similarity measure is to take the reciprocal. Some commonly used distance measures are the Euclidean (also known as the L2-distance) and city-block distances (also known as the Manhattan distance or L1-distance) [9]. For example, Netra [81] uses Euclidean distance on color and shape features; MARS [91] uses Euclidean distance on texture features; Blobworld[15] uses Euclidean distance on texture and shape features. IBM’s QBIC [27] was the first commercial system that implemented CBIR. It addressed the problems of non-Euclidean distance measuring and high-dimensionality of feature vectors. MIT’s Photobook [105] implements a set of interactive tools for browsing and searching images. It consists of three subsystems: one that allows the user to search based on appearance, one that uses 2D shape, and one that allows search based on textural properties. While searching, these image features can be combined with each other and with keywords to improve retrieval performance.

Note that with (uniformly-weighted) Euclidean distance, every feature is treated equally. However, some features may be more important than others. Similarly, in region-based approaches (where similarity between regions of two images has to be computed), some regions may be more important than others in determining overall image-to-image similarity. Thus, the weight of each feature (or region) should be based on its discriminative power between the relevant and non-relevant images for the current query (See Figure 4.1). Then, the similarity measure of images can be based on a weighted distance in the feature space. For example, the (weighted) Euclidean distance between two n -dimensional vectors \mathbf{x} and \mathbf{y} is defined as

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2} \quad (2.1)$$

where w_i is the weight of the i -th dimension.

The querying system developed in [134] decomposes an image into regions with characterizations pre-defined in a finite pattern library. In Blobworld [15], images are partitioned into regions that have similar color and texture. Each pixel is then associated with a set of color, texture, and spatial features. The distribution of pixels for each region is calculated and the distance between two images is equal to the distance between their regions in terms of color and texture. In NeTra[81], regions are segmented based on color. Then, texture, shape, color, and spatial properties are used to determine similarity. Both Blobworld[15] and NeTra[81] require the user to select the region(s) of interest from the segmented query image. This information is then used for determining similarity with database images. In [111], a system that uses a measure of correlation to indicate similarity is used. This system works for a variety of images but it requires the user to select the region(s) of interest from the images.

A major problem with these systems is that the segmented regions they produce usually do not correspond to actual objects in the image. For instance, an object may be partitioned into several regions, with none of them being representative of the object (See Figure 2.6). Thus, due to the great difficulty of accurately segmenting an image into regions that correspond to a human's perception of an object, several approaches have been proposed [17, 75, 134, 146] that consider all regions in an image for determining similarity. As a result, the problems of inaccurate segmentation are reduced.

Integrated region matching (IRM) [75] is proposed as a measure that allows a many-to-many region mapping relationship between two images by matching a region of one image to several regions of another image. Thus, by having a similarity measure that is a weighted sum of distances between all regions from different images, IRM is more robust to inaccurate segmentation. The image segmentation algorithm that is used in IRM first partitions an image into blocks of 4x4 pixels. Then, a feature

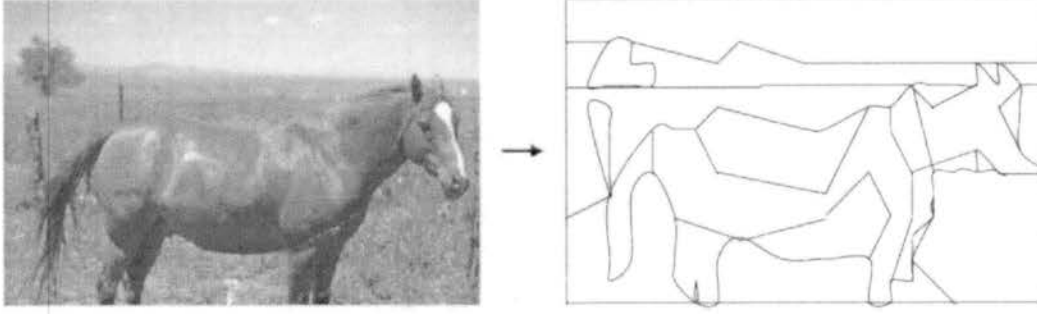


Figure 2.6: Sample of a typical image segmentation in which segmented regions do not correspond to semantically meaningful objects.

vector $\mathbf{f} = [f_1, f_2, f_3, f_4, f_5, f_6]^T$ representing color and texture properties is extracted for each block. The first three features are the average color components and the other three represent energy in high frequency bands of the wavelet transforms [22, 94]. The k -means algorithm is then used to cluster the feature vectors into several regions. The number of regions is adaptively chosen according to a stopping criteria. A feature vector $\mathbf{h} = [h_1, h_2, h_3]^T$ is then extracted for each region to describe its shape characteristics. The shape features are normalized inertia [34] of order 1 to 3. A region is described by $\mathcal{R} = \{\hat{\mathbf{f}}, \mathbf{h}\}$, where $\hat{\mathbf{f}}$ is the average of the feature vectors of all blocks assigned to the region.

Let $\{\mathcal{R}_i\}_1^m$ and $\{\mathcal{R}'_i\}_1^n$ be the region descriptors of two images, where $\mathcal{R}_i = \{\hat{\mathbf{f}}_i, \mathbf{h}_i\}$ and $\mathcal{R}'_i = \{\hat{\mathbf{f}}'_i, \mathbf{h}'_i\}$. For non-textured images, the distance between two regions $d(\mathcal{R}, \mathcal{R}')$ is defined as

$$d(\mathcal{R}, \mathcal{R}') = g(d_s(\mathcal{R}, \mathcal{R}'))d_t(\mathcal{R}, \mathcal{R}')$$

where $d_s(\mathcal{R}, \mathcal{R}')$ is the shape distance computed by

$$d_s(\mathcal{R}, \mathcal{R}') = \sum_{i=1}^3 w_i (h_i - h'_i)^2$$

where the parameter w_i is chosen to adjust the effect of the i -th feature dimension

and $d_t(\mathcal{R}, \mathcal{R}')$ is the color and texture distance computed by

$$d_t(\mathcal{R}, \mathcal{R}') = \sum_{i=1}^6 w_i (\hat{f}_i - \hat{f}'_i)^2$$

The function $g(d_s(\mathcal{R}, \mathcal{R}'))$ is used to ensure a proper influence of the shape distance on the total distance and is defined as

$$g(d_s(\mathcal{R}, \mathcal{R}')) = \begin{cases} 1 & : d_s(\mathcal{R}, \mathcal{R}') \geq 0.5 \\ 0.85 & : 0.2 < d_s(\mathcal{R}, \mathcal{R}') \leq 0.5 \\ 0.5 & : d_s(\mathcal{R}, \mathcal{R}') < 0.2 \end{cases}$$

For textured images, $d(\mathcal{R}, \mathcal{R}') = d_t(\mathcal{R}, \mathcal{R}')$. The IRM distance between the two region sets is then

$$d_{IRM}(\{\mathcal{R}_i\}_1^m, \{\mathcal{R}'_i\}_1^n) = \sum_{i=1}^m \sum_{j=1}^n s_{i,j} d(\mathcal{R}_i, \mathcal{R}'_j)$$

where $s_{i,j}$ is a significance credit indicating the importance of the matching between regions $\mathcal{R}_i, \mathcal{R}'_j$ in determining similarity between the images. Thus, to ensure robustness against segmentation errors, each region is matched to several regions in another image and the matching is assigned with a significance credit (See Figure 2.7). Basically, the “most similar highest priority principle” is used and the smaller the distance between two regions is, the larger their significance credit.

Recently, a fuzzy logic approach, unified feature matching (UFM) [17] was proposed as an improved alternative to IRM. UFM uses the same segmentation algorithm as IRM. In UFM, an image is characterized by a fuzzy feature denoting color, texture, and shape characteristics. Because fuzzy features can characterize the gradual transition between regions in an image, segmentation-related inaccuracies are implicitly considered by viewing them as blurring boundaries between segmented regions. As a result, a feature vector can belong to multiple regions with different degrees of

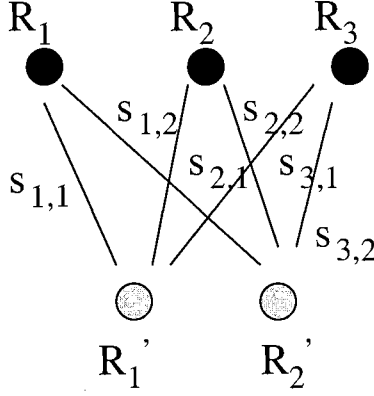


Figure 2.7: Integrated region matching.

membership as opposed to classical region representations in which a feature vector belongs to only one region. The similarity between two images is then defined as the overall similarity between two sets of fuzzy features. A fuzzy feature is defined by a membership function that measures the degree of membership of a feature vector \mathbf{x} to the fuzzy feature. A Cauchy function [57], $\mathcal{C} : \mathfrak{R}^n \rightarrow [0, 1]$, is defined as

$$\mathcal{C}(\mathbf{x}) = \frac{1}{1 + \left(\frac{\|\mathbf{x} - \mathbf{c}\|}{d}\right)^\alpha}$$

where $\mathbf{c} \in \mathfrak{R}^n$ is the center point of the function, d is its width, and α determines its shape. Accordingly, in [17], the color and texture properties of each region \mathcal{R}_i are represented by a fuzzy feature with a Cauchy membership function $\mu_{\mathcal{R}_i, f} : \mathfrak{R}^6 \rightarrow [0, 1]$ defined as

$$\mu_{\mathcal{R}_i, f}(\mathbf{f}) = \frac{1}{1 + \left(\frac{\|\mathbf{f} - \hat{\mathbf{f}}_i\|}{d_f}\right)^\alpha}$$

where

$$d_f = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \|\hat{\mathbf{f}}_i - \hat{\mathbf{f}}_j\|$$

is the average distance between region centers. The shape characteristics of each

region \mathcal{R}_i are also represented by a fuzzy feature with a Cauchy membership function $\mu_{\mathcal{R}_i, h} : \mathbb{R}^3 \rightarrow [0, 1]$ defined as

$$\mu_{\mathcal{R}_i, h}(\mathbf{h}) = \frac{1}{1 + \left(\frac{\|\mathbf{h} - \mathbf{h}_i\|}{d_h}\right)^\alpha}$$

where

$$d_h = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \|\mathbf{h}_i - \mathbf{h}_j\|$$

is the average distance between shape features.

Let $\{(\mu_{\mathcal{R}_i, f}, \mu_{\mathcal{R}_i, h})\}_1^m$ and $\{(\mu_{\mathcal{R}'_i, f}, \mu_{\mathcal{R}'_i, h})\}_1^n$ be the fuzzy feature representations for two images. The color and texture similarity between the two images is captured by the similarity vector

$$\mathbf{c} = [l_1, l_2, \dots, l_m, l'_1, l'_2, \dots, l'_n]^T$$

where

$$\begin{aligned} l_i &= S\left(\mu_{\mathcal{R}_i, f}, \bigcup_{j=1}^n \mu_{\mathcal{R}'_j, f}\right) \\ &= \frac{d_f + d'_f}{d_f + d'_f + \min_{j=1, \dots, n} \|\hat{\mathbf{f}}_i - \hat{\mathbf{f}}'_j\|} \\ l'_i &= S\left(\mu_{\mathcal{R}'_i, f}, \bigcup_{j=1}^m \mu_{\mathcal{R}_j, f}\right) \\ &= \frac{d_f + d'_f}{d_f + d'_f + \min_{j=1, \dots, m} \|\hat{\mathbf{f}}'_i - \hat{\mathbf{f}}_j\|} \end{aligned}$$

and similarly for the shape similarity, captured by similarity vector \mathbf{s} . The UFM

measure for the two images is then defined as

$$d_{UFM}(\{(\mu_{\mathcal{R}_{i,f}}, \mu_{\mathcal{R}_{i,h}})\}_1^m, \{(\mu_{\mathcal{R}'_{i,f}}, \mu_{\mathcal{R}'_{i,h}})\}_1^n) = (1 - \rho)[(1 - \lambda)\mathbf{w}_a + \lambda\mathbf{w}_b]^T \mathbf{c} + \rho\mathbf{w}_a^T \mathbf{s}$$

where the normalized weight vectors \mathbf{w}_a and \mathbf{w}_b can be set according to some region weighting heuristic, $0 \leq \lambda \leq 1$ adjusts the importance of \mathbf{w}_a and \mathbf{w}_b , and $0 \leq \rho \leq 1$ determines the significance of \mathbf{c} (i.e., color and texture similarity) and \mathbf{s} (i.e., shape similarity).

A key factor in these types of systems that consider all the regions to perform an overall image-to-image similarity is the weighting of regions. The weight that is assigned to each region for determining similarity is usually based on prior assumptions such as that larger regions, or regions that are close to the center of the image, should have larger weights. For example, in IRM, an *area percentage scheme*, which is based on the assumption that important objects in an image tend to occupy larger areas, is used to assign weights to regions. The location of a region is also taken into consideration. For example, higher weights are assigned to regions in the center of an image than to those around boundaries. These region weighting heuristics are often inconsistent with human perception. For instance, a facial region may be the most important when the user is looking for images of people while other larger regions such as the background may be much less relevant. Some RF approaches are motivated by the need to have a similarity measure that is flexible to user preferences. In Chapter 5 we present our work on a learning algorithm that can be used in region-based CBIR systems for estimating region weights in an image.

2.4 Indexing Structure

Many data structures (e.g., *B-tree* [4]) have been proposed for the efficient managing of one-dimensional data in traditional database systems. However, because of the rapid

development of multimedia database systems during the past decade, the efficient manipulation of multi-dimensional data is vital [19]. In particular, there is an urgent need for indexing techniques that support the efficient execution of similarity queries. Therefore, a number of data storage and indexing techniques (such as the *R-tree* [46]) have been proposed. However, most of those techniques suffer from the *curse of dimensionality* [5], a phenomenon in which performance degrades as the number of dimensions increases.

A dimensionality reduction technique can be used to reduce the number of features by keeping only the most important ones (i.e., the ones that allow us to retain as much discriminatory information as possible). That is, we should aim at keeping features that result in large interclass distance and small intraclass variance in the feature space. It is also desirable to remove the correlation between features so that any redundant information can be removed. This can be achieved through principal component analysis (PCA) (or discrete Karhunen-Loeve transform) [66]. Suppose we want to reduce our n dimensional data to $m \ll n$ dimensions. The basic idea in PCA is to find the m components that can explain the maximum possible amount of variance by m linearly transformed components. It can be proven that the representation given by PCA is an optimal linear reduction in the mean-square sense [66]. The basic procedure consists of computing m orthonormal vectors (i.e., eigenvectors) that form a basis for the data. Those vectors are the “principal components” and the data are linear combinations of them. The principal components provide important information about the variance in the data. It turns out that the projected data shows the most variance on the first principal component, the next highest variance on the second principal component, and so on. Thus, the dimensionality of the data can be reduced simply by eliminating the last principal components (i.e., the ones with smallest eigenvalues that do not account for much of the variance in the data). Therefore, by keeping only the first principal components (i.e., the ones with largest

eigenvalues that account for most of the variance in the data), it is possible to reconstruct a good approximation of the original data while, at the same time, achieving a reduction in dimensionality (See Figure 2.8).

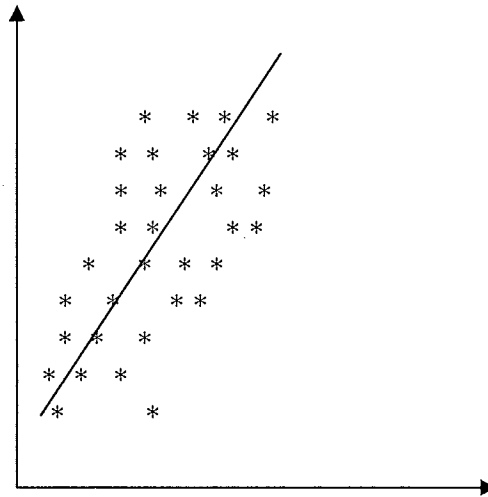


Figure 2.8: Principal component analysis of two-dimensional data. The line shown is the direction of the first principal component (i.e., the one that accounts for most of the variance in the data). By keeping only this principal component, an optimal linear reduction in the number of dimensions from two to one is obtained.

Metric trees are a general approach to the similarity indexing problem. In order to organize and partition the search space, they only consider relative distances between objects. They just require that the distance function is a *metric* (i.e., that it satisfies the symmetry, non negativity, and triangle inequality properties) [19]. An M-tree [19] is an example of a metric tree. It is a paged, balanced, and dynamic tree. It provides an efficient platform for the execution of multi-dimensional similarity queries using an arbitrary metric. The M-tree partitions objects on the basis of their relative distances, as measured by a particular distance function, and stores those objects into nodes of fixed capacity, which correspond to constrained regions of the metric space. The leaf nodes contain the indexed (database) objects themselves while the *routing objects* (stored in the inner nodes) represent the metric regions of the space.

An entry in a leaf node contains the feature vector \mathbf{o}_i of a database object, an object identifier $oid(\mathbf{o}_i)$, and the distance $d(\mathbf{o}_i, P(\mathbf{o}_i))$ between the object and its parent routing object. A routing object contains the feature vector \mathbf{o}_j of the routing object, a pointer $ptr(T(\mathbf{o}_j))$ to a covering subtree, its covering radius $r(\mathbf{o}_j)$, and the distance $d(\mathbf{o}_j, P(\mathbf{o}_j))$ between the object and its parent routing object (this value is zero for the routing objects stored in the root). A routing object \mathbf{o}_j determines a hyper-spherical region in the metric space where \mathbf{o}_j is the center of that region and the radius $r(\mathbf{o}_j)$ specifies its boundary. All objects stored in leafs of the covering subtree of \mathbf{o}_j must be spatially located inside this region (See Figure 2.9). In order to process a similarity query, the M-tree hierarchy is traversed down. The covering subtree of \mathbf{o}_j is relevant to the query (and is thus further processed) only if the region corresponding to \mathbf{o}_j intersects the query region.

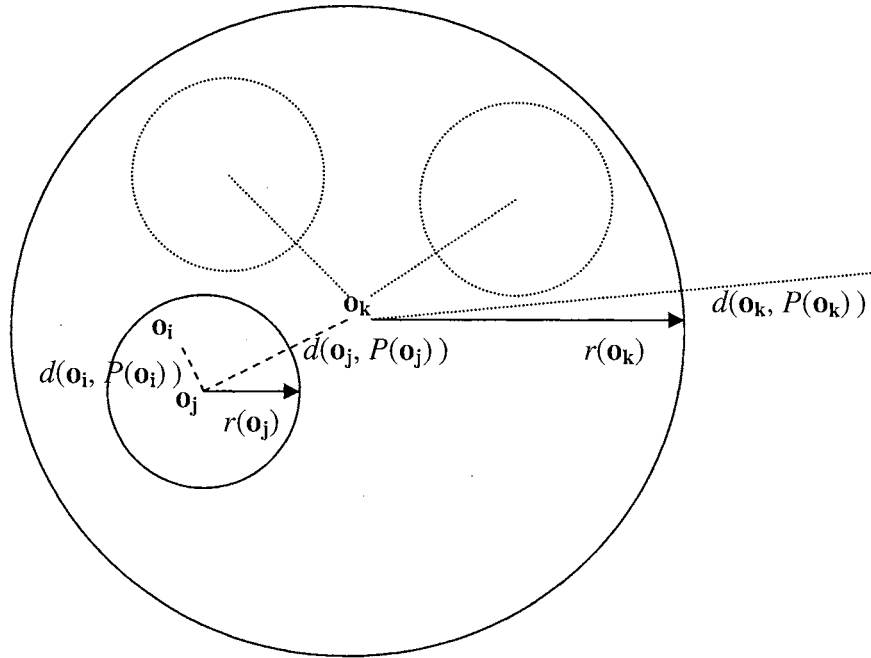


Figure 2.9: M-tree structure.

The retrieval efficiency of the M-tree is highly dependent on the overall "volume" of the regions covered by routing objects and their corresponding region overlap. That

is, the larger the volume of a region is, the larger the amount of indexed "dead space" (i.e., space where no object is present). Also, the smaller the overlap between regions, the fewer the number of paths that have to be traversed for answering a query (See Figure 2.10). These criteria lead to the development of algorithms for building the M-tree that specify how objects are inserted and deleted, and how node overflows and underflows are managed. For more details, refer to [19].

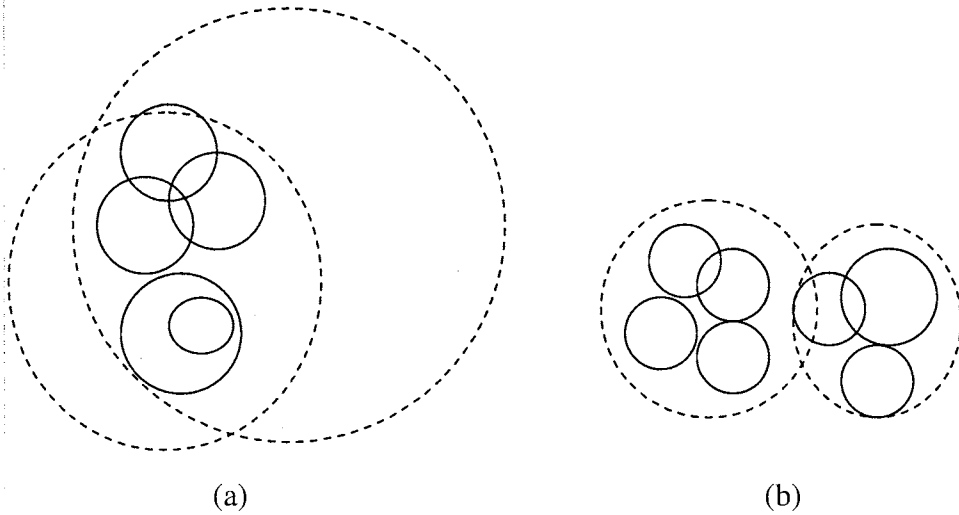


Figure 2.10: Examples of M-trees with: a) large region volumes and overlap; b) small region volumes and overlap.

2.5 Relevance Feedback

The human notion of similarity is usually based on high-level abstractions such as activities, events, or emotions displayed in an image. Therefore, a database image with a high feature similarity to the query image may be completely different from the query in terms of semantics. With the exception of some constrained applications such as face and fingerprint recognition, low-level features do not capture the high-level semantics of images [118]. This discrepancy between low-level features and high-level

concepts is known as the *semantic gap* [130].

Relevance feedback (RF), originally developed for information retrieval [114], has been proposed as a learning technique aimed at reducing the semantic gap. It works by gathering semantic information from user interaction. Based on the user’s feedback on the retrieval results, the retrieval scheme is adjusted. Thus, by providing an image similarity measure under human perception, RF can be seen as a form of supervised learning. In order to learn a user’s query concept, the user labels each image returned in the previous query round as relevant or non-relevant. Based on the feedback, the retrieval scheme is adjusted and the next set of images is presented to the user for labelling. This process iterates until the user is satisfied with the retrieved images or stops searching (See Figure 1.4).

The key issue in RF is how to use the positive and negative examples to adjust the retrieval scheme so that the number of relevant images in the next retrieval set will increase. Two main RF strategies have been proposed in CBIR: query modification [119], and distance reweighing [11, 61, 103, 117, 127]. Query modification changes the representation of the user’s query in a form that is closer (hopefully) to the semantic intent of the user. In particular, query shifting involves moving the query towards the region of the feature space containing relevant images and away from the region containing non-relevant images (See Figure 2.11). Based on RF, the next query location can be determined with the standard Rocchio formula [122]

$$\mathbf{q}' \leftarrow \alpha \mathbf{q} + \beta \left(\frac{1}{|\mathcal{R}^+|} \sum_{\mathbf{x} \in \mathcal{R}^+} \mathbf{x} \right) - \gamma \left(\frac{1}{|\mathcal{R}^-|} \sum_{\mathbf{x} \in \mathcal{R}^-} \mathbf{x} \right)$$

where \mathbf{q} is the initial query, \mathbf{q}' is the new query location, \mathcal{R}^+ is the set of relevant retrievals, and \mathcal{R}^- is the set of non-relevant retrievals. Thus, the new query location \mathbf{q}' is a linear combination of the mean feature vectors of the relevant and non-relevant retrieved images so that \mathbf{q}' is close to relevant mean and far from the non-relevant

mean. The values for the parameters α , β , and γ are usually chosen by experimental runs. Note that the refined query vector represents an ideal query point and does not longer correspond to any actual image.

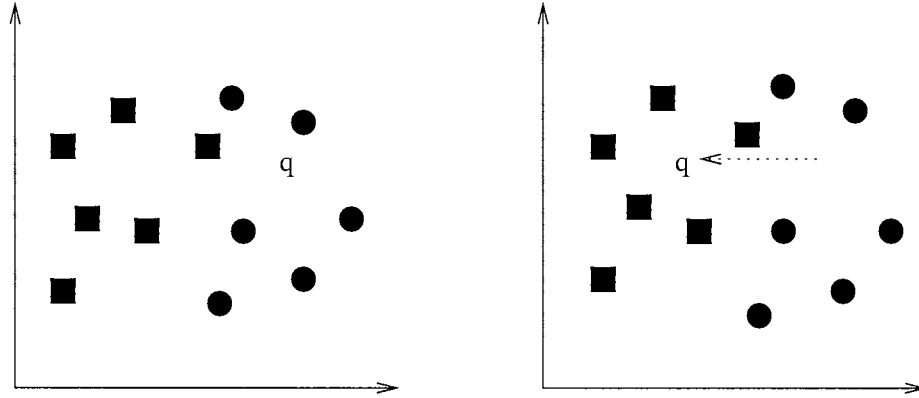


Figure 2.11: Query shifting. The query is moved towards the region of the feature space containing user-labelled relevant images (squares) and away from the region containing user-labelled non-relevant images (circles).

Distance reweighing changes the calculation of image to image similarity to strengthen the contribution of relevant image components in regard to the current query. Thus, the task is to determine the features that help the most in retrieving relevant images and increase their importance in determining similarity.

We can distinguish two different types of information provided by RF. The short-term learning obtained within a single query session is intra-query learning. The long-term learning obtained accumulated over the course of many query sessions is inter-query learning. Previous work on intra and inter-query learning with global and region-based image representations is reviewed in Chapters 4 and 5 respectively.

Chapter 3

Related Work in Machine Learning

The field of machine learning focuses on the study of algorithms that improve their performance at some task automatically through experience [97]. In this chapter, we present two machine learning techniques, support vector machines (SVM), and multiple instance learning (MIL), which will be applied in subsequent chapters.

3.1 Support Vector Machines

This section presents the basic concepts of support vector machines (SVM). For more detailed gentle introductions, refer to [13, 21, 139]. A SVM is a system for training linear learning machines in a kernel-induced feature space efficiently while at the same time, respecting the insights provided by generalization theory and exploiting optimization theory [21]. The objective of support vector classification is to create a computationally efficient method of learning “good” separating hyperplanes in a high dimensional feature space, where “good” corresponds to optimizing the generalization bounds given by generalization theory [21].

3.1.1 Risk Minimization

Suppose we are given training data for a classification problem as a set of n observations. Each observation is a pair (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in \mathfrak{R}^d$ and $y_i \in \mathfrak{R}$ is the corresponding class label. We assume that the training data has been drawn independently from some unknown cumulative probability distribution $P(\mathbf{x}, y)$. The goal is to find a machine (i.e., a function $f : \mathfrak{R}^d \mapsto \mathfrak{R}$) that implements the optimal mapping. In order to make learning feasible, we have to specify a function space \mathcal{F} from which a machine is chosen. For example, \mathcal{F} can be the set of hyperplanes in \mathfrak{R}^d , artificial neural networks with a certain structure, or any other set of parameterized functions. The functions are labelled by a set \mathcal{P} of adjustable parameters. Thus, a learning machine is a family of functions \mathcal{F} and a particular choice of \mathcal{P} results in a “trained machine” [13]. The task is to choose a function from a set of functions defined by the construction of the particular learning machine. For instance, in an artificial neural network, the problem reduces to finding the optimal set of weights for a particular network architecture.

In particular, consider a binary classification task with training data $\{(\mathbf{x}_i, y_i)\}_1^n$ where $\mathbf{x}_i \in \mathfrak{R}^d$ and $y_i \in \{1, -1\}$ is the class label. If the training data is linearly separable, we can let \mathcal{F} be the set of linear decision boundaries of the form

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

where $\mathbf{w} \in \mathfrak{R}^d$ and $b \in \mathfrak{R}$ are the adjustable parameters (i.e., $\mathcal{P} = \{\mathbf{w}, b\}$). Thus, choosing particular values for \mathcal{P} results in a trained classifier (See Figure 3.1).

One way to measure the performance of a trained classifier $f \in \mathcal{F}$ is to look at the mean error computed from the training data. This is known as the empirical risk (or training error) and is defined as

$$R_{emp}(\mathcal{P}) = \frac{1}{n} \sum_{i=1}^n Q(\mathbf{x}_i, \mathcal{P})$$

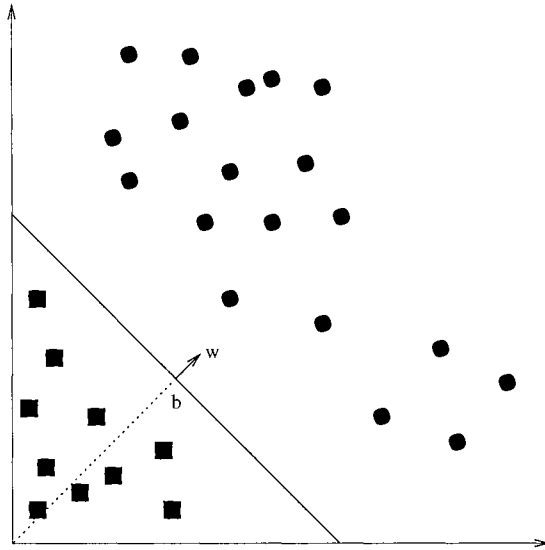


Figure 3.1: A simple binary classifier.

where $Q(\mathbf{x}_i, \mathcal{P}) = 1$ if $f(\mathbf{x}_i, \mathcal{P}) \neq y_i$ and $Q(\mathbf{x}_i, \mathcal{P}) = 0$ if $f(\mathbf{x}_i, \mathcal{P}) = y_i$. Minimizing the empirical risk is one of the most commonly used optimization procedures. However, even when there is no error on the training data, the classifier may not generate correct classifications on unseen data (See Figure 3.2). This problem is known as overfitting and it drove the initial development of SVMs [13]. The ability of a machine to correctly classify new data that is not in the training set is known as generalization. Having a machine with good generalization is, of course, a much harder problem. The generalization performance of a particular trained machine f can be measured by the expected risk (or just the risk) defined as

$$R(\mathcal{P}) = \int Q(\mathbf{x}, \mathcal{P}) \, dP(\mathbf{x}, y)$$

Choosing optimal values for \mathcal{P} that minimize the expected risk is known as risk minimization. However, this is not a trivial problem because $P(\mathbf{x}, y)$ is usually unknown. There is a competition of terms. As the complexity of the classifier increases, the empirical risk tends to decrease. However, the generalization error usually increases with increasing complexity (See Figure 3.2). Therefore, in order to control the ex-

pected risk, we have to control both the empirical risk and the complexity of the classifier. Note that these two tasks are in conflict with one another. For example, an artificial neural network with a very simple structure may not be capable of correctly classifying most of the training data. That is, it may have high empirical error. On the other hand, an artificial neural network with a very complex structure may correctly classify all the training data but may not generalize well on unseen data. In order to choose from among multiple classifiers, we can follow Ockham's razor: prefer the simplest classifier that is consistent with the training data. The best generalization performance can be obtained when the complexity of the learning machine is restricted to one that is suitable to the amount of available training data [13]. The principle of structural risk minimization is an attempt to identify the optimal balance between the quality of the approximation of the training data and the complexity of the approximating function (See Figure 3.2).

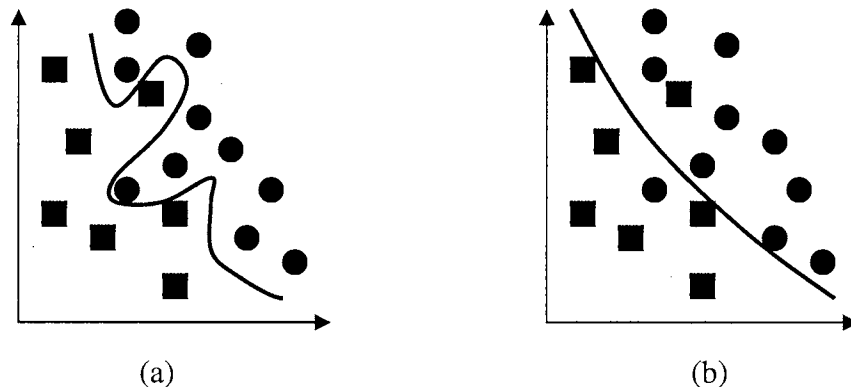


Figure 3.2: Generalization performance: a) an overly complex classifier that results in zero error on the training data, but may not generalize well to unseen data; b) a classifier that might represent the optimal tradeoff between error in the classification of training data and complexity of the classifier, thus capable of generalizing well on unseen data.

The Vapnik Chervonenkis (VC) dimension [143] is a measure of the complexity of a set of classifiers \mathcal{F} . It is defined as the size of the largest subset of points that can be shattered (or arbitrarily labelled) by choosing classifiers from \mathcal{F} with different

values of \mathcal{P} (See Figure 3.3). Any given set of classifiers \mathcal{F} has a fixed VC dimension. For example, an artificial neural network with a fixed structure represents a set of classifiers (obtained by all possible values for the weights) with a fixed complexity (i.e., fixed VC dimension).

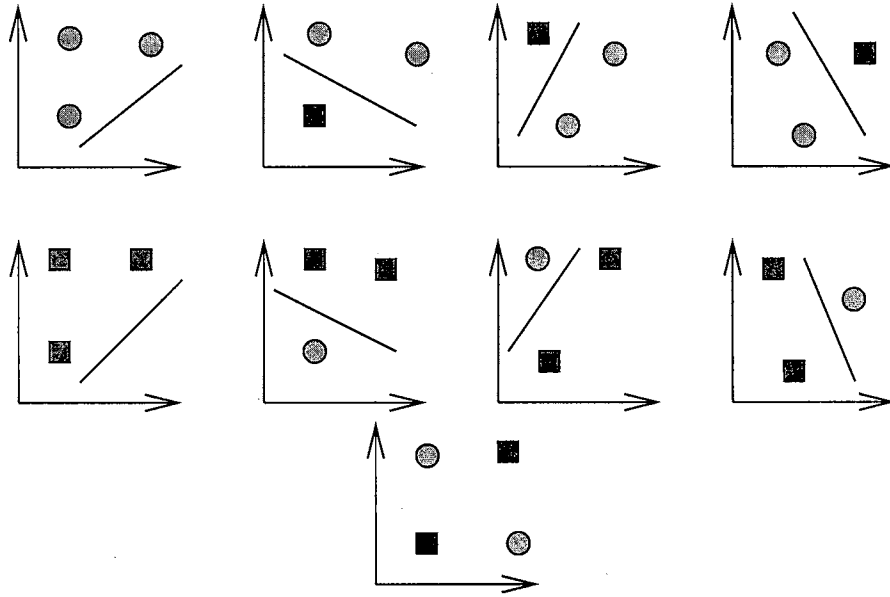


Figure 3.3: The VC dimension of linear decision boundaries is 3 because they can shatter (any) 3 points in a 2-dimensional space but not (any) 4 points.

There is a number of bounds on the expected risk. Vapnik and Chervonenkis [143] proved that, given a set of n training examples and a set of classifiers \mathcal{F} , with probability $1 - \eta$ over the choice of training set, the expected risk of a trained classifier $f \in \mathcal{F}$ is bounded by

$$R(\mathcal{P}) \leq R_{emp}(\mathcal{P}) + \sqrt{\frac{h(1 + \ln \frac{2n}{h}) - \ln \frac{\eta}{4}}{n}}$$

where h is the VC dimension of \mathcal{F} [143]. Therefore, in order to control the expected risk, by the principle of structural risk minimization, we have to control both the empirical risk (i.e., we have to minimize the error on the training data) and the VC dimension (i.e., we have to minimize the complexity of the classifier).

3.1.2 Maximal Margin Hyperplanes

In supervised learning, the learning machine is given a set of labelled examples. That is, each observation is a pair (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in \mathfrak{R}^d$ and $y_i \in \mathfrak{R}$ is the corresponding class label. Once this training data is available, a number of functions spaces could be chosen for the problem. Among these, linear functions are the best understood and simplest to apply [21]. In particular, given training data $\{(\mathbf{x}_i, y_i)\}_1^n$ for a binary classification task where $\mathbf{x}_i \in \mathfrak{R}^d$ and $y_i \in \{1, -1\}$ is the class label. Assume that the data is linearly separable and let \mathcal{F} be the set of linear decision boundaries of the form

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

where $\mathbf{w} \in \mathfrak{R}^d$ and $b \in \mathfrak{R}$ are the adjustable parameters (i.e., $\mathcal{P} = (\mathbf{w}, b)$). Thus, choosing particular values for \mathcal{P} results in a trained classifier (See Figure 3.1). For any trained classifier, the hyperplane corresponding to $\mathbf{w}^T \mathbf{x} + b = 0$ is the decision boundary (See Figure 3.5).

In the late 1950s, Rosenblatt [115] introduced the first iterative algorithm for learning linear classifiers, the perceptron learning rule. After initializing \mathbf{w} and b randomly, each training point \mathbf{x}_i is presented and the value of $f(\mathbf{x}_i)$ is compared against y_i . If $f(\mathbf{x}_i)$ and y_i are different (i.e., \mathbf{x}_i is misclassified) the values of \mathbf{w} and b are adapted by moving them either towards or away from \mathbf{x}_i . Rosenblatt proved that, assuming the classes are linearly separable, the algorithm will always converge and find values for \mathbf{w} and b that solve the classification problem. The algorithm is shown in Figure 3.4.

It is important to observe that the perceptron learning algorithm works by adding or subtracting misclassified training points to a randomly initialized \mathbf{w} . Without any loss of generality we can assume that \mathbf{w} is initialized to the zero vector, and thus its

1. Given training set $\{(\mathbf{x}_i, y_i)\}_1^n$ and learning rate $\eta \in \mathfrak{R}$
2. Initialize \mathbf{w} and b to small random values
3. Repeat
4. For $i = 1$ to n
 - If $y_i f(\mathbf{x}_i) \leq 0$ (if misclassification)
 - $\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i$
 - $b \leftarrow b + \eta y_i$
5. End for
6. Until no misclassifications made within the for loop
7. Return \mathbf{w}, b

Figure 3.4: The Perceptron Learning Algorithm.

final value will be a linear combination of the training points [21]

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

where α_i is a positive value proportional to the number of times misclassification of \mathbf{x}_i has caused \mathbf{w} to be updated. Intuitively, α_i can also be regarded as a measure of the information content of \mathbf{x}_i . The decision function can then be rewritten in dual coordinates as follows [21]

$$\begin{aligned} f(\mathbf{x}) &= \text{sign}(\mathbf{w}^T \mathbf{x} + b) \\ &= \text{sign} \left(\left\langle \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^T \mathbf{x} \right\rangle + b \right) \\ &= \text{sign} \left(\sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i^T \mathbf{x} \rangle + b \right) \end{aligned}$$

An important property of this dual representation of the decision function is that only the inner products of the training data with the new test point are needed.

In Figure 3.5, the hyperplanes corresponding to $\mathbf{w}^T \mathbf{x} + b = -1$ and $\mathbf{w}^T \mathbf{x} + b = 1$

are the bounding hyperplanes. The distance between the two bounding hyperplanes is the margin and it is equal to $\frac{2}{\|\mathbf{w}\|}$. It can be shown that, with a large margin, the number of possible labellings of points can be dramatically less than the (basic) VC dimension. The set of separating hyperplanes which attain margin γ or better for training data within a hypersphere of radius r has VC dimension bounded by [142]

$$h \leq \frac{r^2}{\gamma^2} \quad (3.1)$$

Thus, for given training data, maximizing the margin of separation between the two classes has the effect of minimizing h and thus optimizing generalization performance. It can be shown that the optimal hyperplane (i.e., the one that minimizes the generalization error or the bound on the expected risk) corresponds to the one that minimizes the empirical risk and, at the same time, has the maximal margin of separation between the two classes [13]. The optimal hyperplane has the smallest complexity (i.e., the lowest VC dimension). Figure 3.6 shows three hyperplanes that achieve a perfect classification. That is, all of them have zero empirical risk. However, only the hyperplane with maximum margin of separation between the two classes achieves optimal generalization.

In order to find the optimal separating hyperplane, the following convex optimization problem is solved

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

with the constraints that

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, n$$

Thus, the task is to maximize the margin while achieving the correct classification

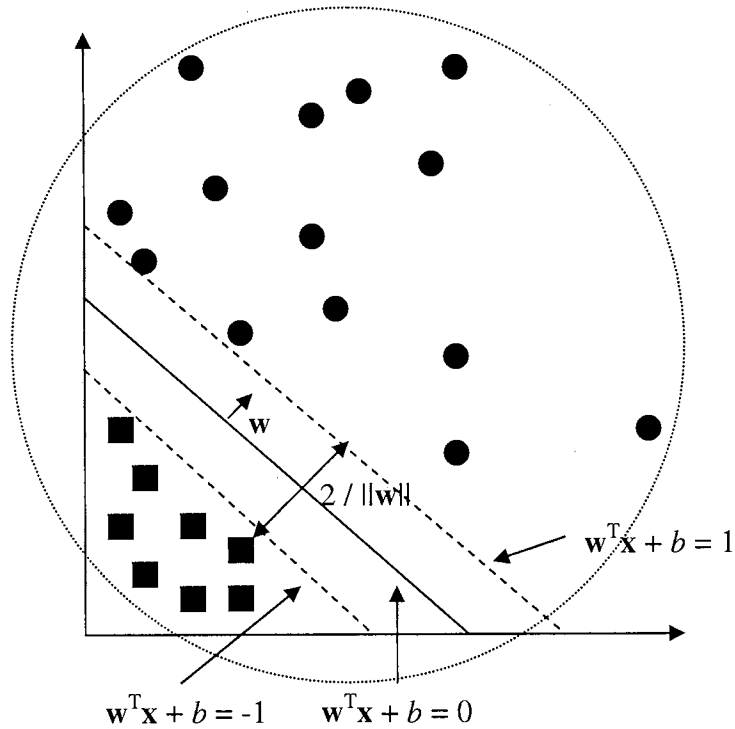


Figure 3.5: A simple linear SVM. The optimal separating hyperplane has the maximal margin of separation between the two classes.

of all the training data. In order to allow for the possibility that the two classes are not linearly separable, slack variables are introduced that allow for misclassifications. The optimization problem then becomes

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^n \zeta_i$$

with the constraints that

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad i = 1, 2, \dots, n$$

where $\zeta_i \geq 0$ is a slack variable. The parameter c is the soft-hard margin penalty and it gives the tradeoff between the size of the margin (i.e., the VC dimension) and the number of misclassifications (i.e., the empirical risk).

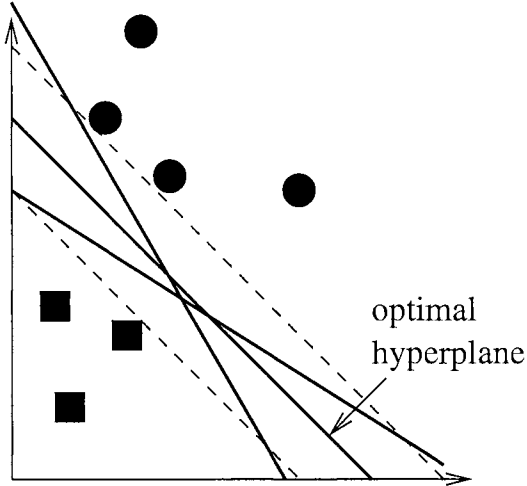


Figure 3.6: The optimal hyperplane is the one that minimizes the empirical risk and, by maximizing the margin of separation between the two classes, results in the best generalization performance.

Applying the Karush-Kuhn-Tucker conditions [21], any \mathbf{w} in a solution to the above optimization problem can be written as a linear combination of the training data

$$\mathbf{w} = \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i$$

where $\alpha_i \in \mathfrak{R}$ are the weights associated with each data point. Those points for which $\alpha_i > 0$ are called support vectors and lie closest to the hyperplane (See Figure 3.7). All other points have $\alpha_i = 0$ thus the support vectors are the critical elements of the training set [13]. The number of support vectors is usually much smaller than n . The final decision function is of the form

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b \right) \quad (3.2)$$

where the α_i 's can be found by solving the following dual optimization problem

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (3.3)$$

with the constraints that

$$c \geq \alpha_i \geq 0, i = 1, 2, \dots, n$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

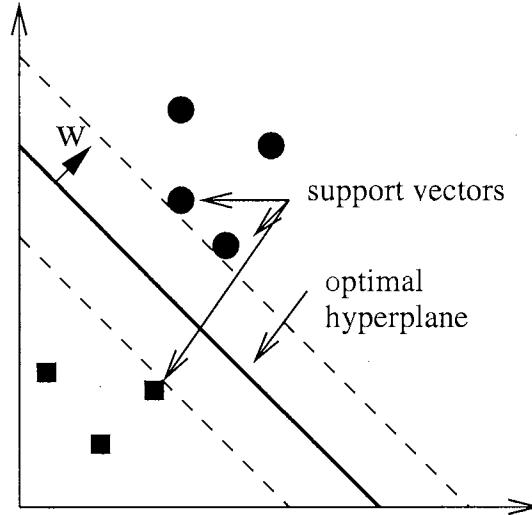


Figure 3.7: The points that lie closest to the separating hyperplane are known as support vectors and are the critical elements of the training set.

3.1.3 Non-Linear Classifiers

A linear decision boundary is a simple classifier that can be learned very efficiently. However, due to its small complexity it can correctly classify data that is linearly separable only. On the other hand, a more complex decision boundary can correctly classify general data that may not be linearly separable. However, such a classifier may be much harder to train. A SVM combines the best of both worlds. That is, it uses an efficient training algorithm while at the same time being capable of representing complex decision boundaries.

In order to generalize to the case where the decision function is not linearly separable, SVMs first map the data into some other (possibly infinite dimensional) feature space using a mapping $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, with $d' \geq d$ (see Figure 3.8). Clearly, there is

no linear separator for the data in the Figure. However, the data is linearly separable in the new feature space. This is because data that is mapped into a sufficiently high dimensional space will always be linearly separable. In order to avoid confusion, from now on when in the context of SVMs, we will refer to the original lower dimensional feature space (i.e., \mathbb{R}^d) as the “input space” and to the higher dimensional feature space (i.e., $\mathbb{R}^{d'}$) as the “feature space”. Note that both the optimization problem (3.3) and the final decision function (3.2) depend on the data through dot products in the input space (i.e., $\mathbf{x}_i^T \mathbf{x}_j$). This implies that there is no need to evaluate $\Phi(\mathbf{x}_i)$ or $\Phi(\mathbf{x}_j)$ as long as we know what the value of $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ is. We can use a kernel function to avoid having to perform an explicit mapping into the feature space. A kernel function K calculates the dot product in the feature space of the image of 2 points from input space, $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. Table 3.1 shows some commonly used kernel functions. Thus, we can find a linear separator in the feature space simply by replacing $\mathbf{x}_i^T \mathbf{x}_j$ in (3.3) with $K(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{x}_i^T \mathbf{x}$ in (3.2) with $K(\mathbf{x}_i, \mathbf{x})$. The importance of this is that we can learn complex decision boundaries in feature space efficiently (i.e., without having to work with the feature space representation of each data point). When mapped back to the original input space, the resulting linear separators can correspond to arbitrary nonlinear decision boundaries between the two classes. Mercer’s theorem [92] indicates that any kernel whose matrix $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite corresponds to some feature space and is thus a valid kernel. Distance in the feature space can be calculated by means of the kernel function [21]. Given \mathbf{x}_i and \mathbf{x}_j in input space, the corresponding distance in feature space is

$$\begin{aligned} dist_F(\mathbf{x}_i, \mathbf{x}_j)^2 &= \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 \\ &= K(\mathbf{x}_i, \mathbf{x}_i) - 2K(\mathbf{x}_i, \mathbf{x}_j) + K(\mathbf{x}_j, \mathbf{x}_j) \end{aligned}$$

This is known as the kernel trick and it allows SVMs to implicitly project the original

Table 3.1: Common Kernels.

Kernel	Formula
Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^n$
Gaussian	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\ \mathbf{x}_i - \mathbf{x}_j\ ^2 / \sigma^2}$

training data to the feature space.

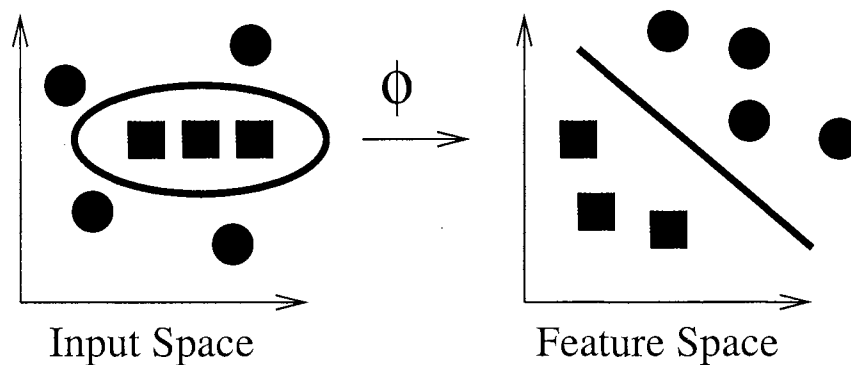


Figure 3.8: A SVM maps the training data nonlinearly into a higher dimensional feature space via Φ . By the use of a kernel function, the optimal separating hyperplane can be computed without explicitly carrying out the map into the feature space.

Substituting $K(\mathbf{x}_i, \mathbf{x}_j)$ for $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ gives the following optimization problem

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

with the constraints that

$$c \geq \alpha_i \geq 0, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

Solving for the α 's in the above optimization problem results in the following final decision function

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \right)$$

which corresponds to a linear hyperplane in the feature space and an arbitrarily complex decision boundary in the input space.

3.1.4 One-Class Support Vector Machines

In a one-class classification problem, data from only one of the classes (i.e., the target class) is available. For instance, user-labelled relevant images give us information about the user’s high level concept. Many terms (e.g., concept learning [64], outlier detection [113], novelty detection [10]) have been used according to the different applications to which one-class classification can be applied. One approach to this problem is to model the support of the target data distribution (i.e., to create a function which is positive in those regions of input space where most of the target data is located and negative elsewhere).

The approach taken in [139] consists of mapping the training data to a feature space and then attempting to include most of it into a hypersphere of minimum size. Thus, the task is to create a boundary around the target class such that most of the target data is included while, at the same time, minimizing the risk of accepting outliers (i.e., data that does not belong to the target class). This model can be rewritten in a form comparable to the support vector classifier [142] and it is therefore called the support vector data description (SVDD) [139]. Consider training data as a set of n observations $\{\mathbf{x}_i\}_1^n$ where $\mathbf{x}_i \in \mathfrak{R}^d$. If the hypersphere contains all the training data, the empirical error is equal to zero. This is analogous to a maximum margin hyperplane that correctly classified all of its training data. Similarly, from (3.1), minimizing the radius of the hypersphere that encloses the training data results in an optimization of generalization performance. Thus, the task is to solve the following optimization problem (See Figure 3.9)

$$\min_{r, \zeta, \mathbf{a}} r^2 + c \sum_{i=1}^n \zeta_i$$

where $r \in \mathfrak{R}$ and $\mathbf{a} \in \mathfrak{R}^{d'}$ are the radius and center respectively of the hypersphere, with constraints that (almost) all the training data are within the hypersphere

$$\|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2 \leq r^2 + \zeta_i, \zeta_i \geq 0, i = 1, 2, \dots, n$$

The parameter $0 \leq c \leq 1$ is the soft-hard margin penalty and it gives the tradeoff between the size of the hypersphere and the number of training data that can be included. By setting partial derivatives to 0 in the corresponding Lagrangian the following expression for \mathbf{a} is obtained

$$\mathbf{a} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}_i)$$

Replacing partial derivatives into the Lagrangian and noticing that \mathbf{a} is a linear combination of the training data, which allows us to use a kernel function, the following objective function (in dual form) is obtained

$$\min_{\boldsymbol{\alpha}} \sum_{i=j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=j=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}_i)$$

with constraints

$$0 \leq \alpha_i \leq c, \sum_{i=1}^n \alpha_i = 1$$

where K is an appropriate Mercer kernel. A quadratic programming method is used to find the optimal α values in the objective function [139]. Given \mathbf{x} in input space and hypersphere center \mathbf{a} , their corresponding distance in feature space is

$$\begin{aligned} dist_F(\mathbf{x}, \mathbf{a})^2 &= \|\Phi(\mathbf{x}) - \mathbf{a}\|^2 \\ &= K(\mathbf{x}, \mathbf{x}) - 2 \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i=j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

Also, \mathbf{x} falls inside the hypersphere when this distance is smaller than or equal to the radius (i.e., $dist_F(\mathbf{x}, \mathbf{a})^2 \leq r^2$)

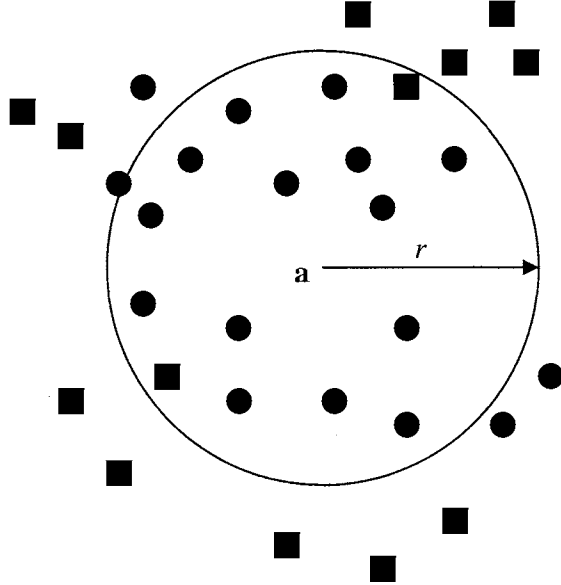


Figure 3.9: A hypersphere containing most of the training data.

A method for adapting the standard two-class SVM techniques to the one-class classification problem was proposed by Schölkopf in [125]. The basic idea of their approach is to treat the origin as the only member of the second class. That is, via the use of a kernel function, the training data is first mapped into a feature space and then separated from the origin with maximum margin (See Figure 3.10). Although this is not a closed boundary around the data, it gives equivalent solutions to Tax's hypersphere approach [139] when the data is preprocessed to have unit norm [139] (See Figure 3.11). In the case of a Gaussian kernel, the data is implicitly rescaled to unit norm since $K(\mathbf{x}, \mathbf{x}) = \Phi(\mathbf{x})^T \Phi(\mathbf{x}) = 1$ and thus all vectors in the feature space lie in a unit hypersphere. Indeed the angles between all vectors are smaller than $\pi/2$. Therefore, the data points are placed on a portion of the same octant on the unit hypersphere in the feature space and thus can be more easily separated from the origin by the hyperplane [44]. In their practical implementation, this approach and

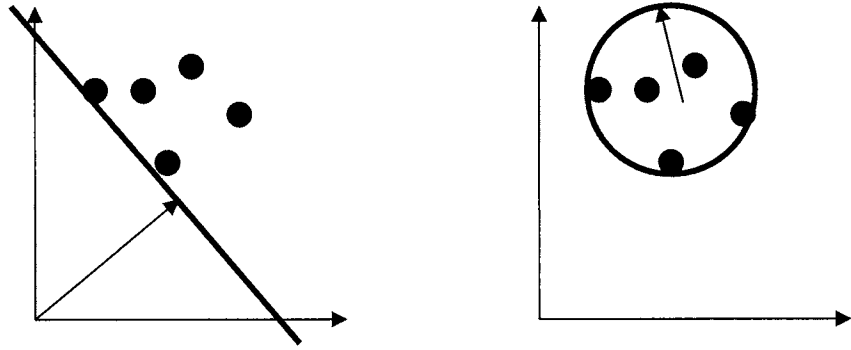


Figure 3.10: The decision boundary on the left is generated by Schölkopf's hyperplane approach; the one on the right corresponds to Tax's hypersphere method.

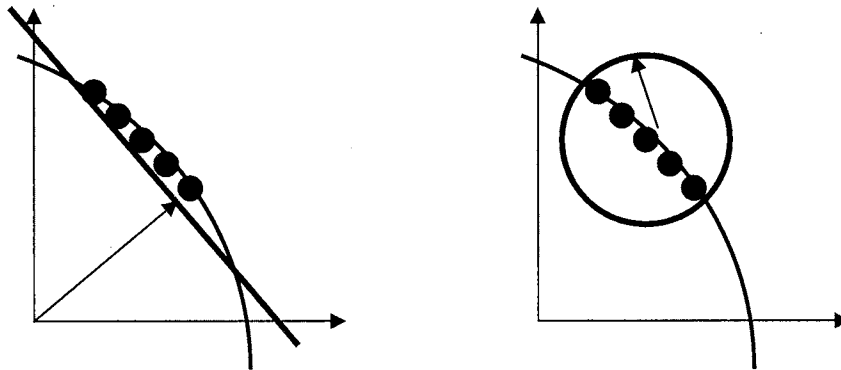


Figure 3.11: When the data is normalized to unit norm, it lies on a unit hypersphere. The decision boundary on the left is generated by Schölkopf's hyperplane approach; the one on the right corresponds to Tax's hypersphere method.

Tax's hypersphere method operate comparably and perform best when the Gaussian kernel is used [139]. In this dissertation, we use Tax's hypersphere approach [139] which, we believe, has a more intuitive description. In order to emphasize the one-class classification task, from now on we will refer to this approach as the one-class support vector machine (1SVM).

3.1.5 Generalized Support Vector Machines

A conventional SVM requires symmetry and positive definiteness of the kernel. A generalized support vector machine (GSVM) [84] has been developed that allows the use of an arbitrary kernel and it can lead to a decision function that is as satisfactory as that of a conventional SVM. Even for negative definite kernels, a GSVM can generate a decision function that can correctly classify the training data whereas the conventional SVM does not. A GSVM can be very useful in the case of variable-length training data. Traditional classification approaches based on SVM learning require the use of fixed-length representations for the training data because SVM kernels represent an inner product in a feature space that is a non-linear transformation of the input space. However, many classification problems create variable-length representations of the data and define a similarity measure between two variable-length representations. Thus, the standard SVM approach cannot be applied because it violates the requirements that SVM places on the kernel. Since GSVM does not place restrictions on the kernel, any similarity measure (i.e., not necessarily an inner product one) can be used.

We follow the matrix notation of [84]. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times l}$. The kernel $K(\mathbf{X}, \mathbf{B})$ implements an arbitrary function mapping $\mathbb{R}^{m \times n} \times \mathbb{R}^{n \times l}$ into $\mathbb{R}^{m \times l}$. In particular, given two column vectors $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, $K(\mathbf{x}^T, \mathbf{X}^T)$ is a row vector in \mathbb{R}^m , $K(\mathbf{x}^T, \mathbf{b}) \in \mathbb{R}$, and $K(\mathbf{X}, \mathbf{B}^T)$ is an $m \times m$ matrix [84].

Given training data $\{(\mathbf{x}_i, y_i)\}_1^n$ for a binary classification task, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, -1\}$ is the class label, represent it by matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and diagonal matrix of plus or minus ones $\mathbf{Y} \in \mathbb{R}^{n \times n}$. Suppose we have a separating hyperplane induced by $K(\mathbf{X}, \mathbf{X}^T)$ defined as follows

$$K(\mathbf{x}^T, \mathbf{X}^T)\mathbf{Y} \cdot \mathbf{u} = b \tag{3.4}$$

where $\mathbf{u} \in \mathfrak{R}^n$ and $b \in \mathfrak{R}$. In the particular case that K is an inner product kernel under Mercer's condition, the separating surface becomes

$$\Phi(\mathbf{x})^T \Phi(\mathbf{X})^T \mathbf{Y} \cdot \mathbf{u} = b$$

where $\Phi : \mathfrak{R}^d \rightarrow \mathfrak{R}^{d'}$ with $d' \geq d$. The parameters \mathbf{u} and b in (3.4) can be obtained by solving the following optimization problem

$$\min_{\mathbf{u}, b, \zeta} c\mathbf{e} \cdot \zeta + \theta(\mathbf{u}) \tag{3.5}$$

$$\begin{aligned} s.t. \quad \mathbf{Y}(K(\mathbf{X}, \mathbf{X}^T))\mathbf{Y}\mathbf{u} - \mathbf{e}b + \zeta &\geq \mathbf{e} \\ \zeta &\geq 0. \end{aligned}$$

where $\mathbf{e} \in \mathfrak{R}^n$ is a column vector of ones, θ is some convex function, c is a positive parameter that weights the separation error $\mathbf{e} \cdot \zeta$ versus suppression of the separating surface parameter \mathbf{u} . Suppression of \mathbf{u} can be interpreted as minimizing the number of constraints of (3.5) with positive multipliers (i.e., number of support vectors). In the particular case that θ is a quadratic function induced by a positive definite kernel, we have the standard interpretation of a maximal margin hyperplane [84]. A solution to (3.5) with corresponding decision function is referred to as a GSVM in [84].

In the particular case that θ in (3.5) is a convex quadratic function (i.e., $\theta(\mathbf{u}) = \frac{1}{2}\mathbf{u} \cdot \mathbf{H}\mathbf{u}$, where $\mathbf{H} \in \mathfrak{R}^{n \times n}$ is a symmetric positive definite matrix), the Wolfe dual [83, 151] of (3.5) is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \cdot \mathbf{Y}K(\mathbf{X}, \mathbf{X}^T)\mathbf{Y}\mathbf{H}^{-1}\mathbf{Y}K(\mathbf{X}, \mathbf{X}^T)^T\mathbf{Y}\boldsymbol{\alpha} - \mathbf{e} \cdot \boldsymbol{\alpha}$$

$$s.t. \quad \mathbf{e} \cdot \mathbf{Y}\boldsymbol{\alpha} = 0$$

$$0 \leq \alpha \leq ce.$$

where $\alpha \in \mathfrak{R}^n$ and $\mathbf{u} = \mathbf{H}^{-1}\mathbf{Y}K(\mathbf{X}, \mathbf{X}^T)^T\mathbf{Y}\alpha$. If $K(\mathbf{X}, \mathbf{X}^T)$ is assumed to be symmetric positive definite and $\mathbf{H} = \mathbf{Y}K(\mathbf{X}, \mathbf{X}^T)\mathbf{Y}$, then we obtain the dual problem for a standard SVM with $\mathbf{u} = \alpha$ [84]. The basic idea in [84] is to choose other values for the matrix \mathbf{H} that will also suppress \mathbf{u} . In the simplest case, choosing $\mathbf{H} = \mathbf{I}$ (i.e., the identity matrix) with $\mathbf{u} = \mathbf{Y}K(\mathbf{X}, \mathbf{X}^T)^T\alpha$ results in the following dual problem

$$\min_{\alpha} \frac{1}{2} \cdot \mathbf{Y}\mathbf{A}\mathbf{Y}\alpha - \mathbf{e} \cdot \alpha \tag{3.6}$$

$$s.t. \mathbf{e} \cdot \mathbf{Y}\alpha = 0$$

$$0 \leq \alpha \leq ce.$$

where $\mathbf{A} = K(\mathbf{X}, \mathbf{X}^T)K(\mathbf{X}, \mathbf{X}^T)^T$ is a positive semidefinite matrix. Thus, this is an always solvable convex quadratic problem for any kernel K [84].

3.2 Multiple-Instance Learning

In traditional supervised learning, the training set consists of individually labelled examples. That is, each observation is a pair (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in \mathfrak{R}^d$ and $y_i \in \mathfrak{R}$ is the corresponding class label. Multiple-instance learning (MIL) [25, 85, 87] is a generalization of this in which training class labels are associated with sets (or *bags*) of examples (or *instances*). While every instance may have an associated true label, individual instances are not given a label. Instead, each bag is labelled. More formally, the training data is $\{(\mathcal{B}_i, y_i)\}_1^n$ where \mathcal{B}_i is a bag and $y_i \in \mathfrak{R}$ is its corresponding class label. The label y_i of a bag $\mathcal{B}_i = \{\mathbf{b}_{i1}, \mathbf{b}_{i2}, \dots, \mathbf{b}_{im}\}$, where $\mathbf{b}_{ij} \in \mathfrak{R}^d$ is its j -th instance, is determined by the instance with the highest label. In the binary case,

a bag is labelled positive if it contains at least one instance which is positive. On the other hand, a bag is labelled negative if all the instances in it are negative. In standard supervised learning, we can observe the label of an instance \mathbf{b}_{ij} . In the multiple instance model we can only see the label of its bag \mathcal{B}_i .

The MIL model was only recently formalized by [25]. Their work was motivated by the *drug activity prediction problem* where a bag is a molecule (i.e., a drug) of interest and instances in the bag correspond to possible configurations (i.e., shapes) that the molecule is likely to take. The efficacy of a molecule (i.e., how well it binds to a “binding site”) can be tested experimentally, but there is no way to control for individual configurations. Thus, the objective is to determine those shapes which will bind with a receptor molecule. There has been a significant amount of research directed towards this problem. Several other applications of MIL, including image classification and retrieval [2, 86, 154, 158], have also been studied.

3.2.1 Diverse Density

Maron and Lozano-Pérez [87] devised a framework called diverse density (DD) (see also [85]) to solve the MIL problem. The main idea behind the DD algorithm is to find areas in feature space that are close to at least one instance from every positive bag and far from all instances in negative bags. The DD at a point in the feature space is a measure of how many *different* positive bags have instances near that point, and of how far all instances in negative bags are from that point. Note that this differs from the more regular density concept of finding a point in the feature space with both high density of positive instances and low density of negative instances. The algorithm searches the feature space for points with high DD (See Figures 3.12 and 3.13).

Next, we introduce a derivation of DD from Maron and Lozano-Pérez [87] based on a probabilistic framework. Following the same notation as in [87], denote positive

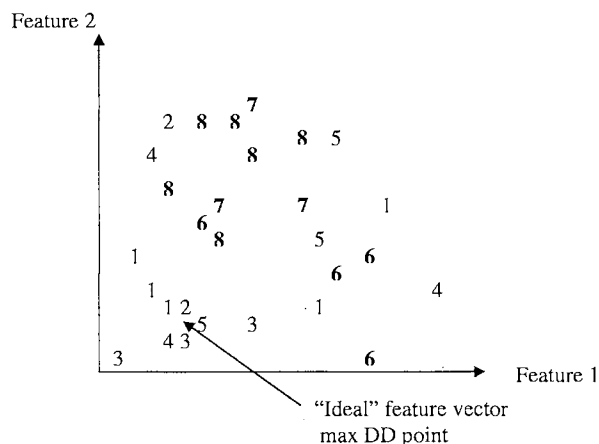


Figure 3.12: The main idea behind diverse density is to find areas in feature space that are close to at least one instance from every positive bag and far from all instances in negative bags. The numbers indicate the location of instances from each of 8 different bags. Instances from negative bags (6 through 8) are in bold.

bags as $\mathcal{B}_1^+, \mathcal{B}_2^+, \dots, \mathcal{B}_n^+$ and the negative bags as $\mathcal{B}_1^-, \mathcal{B}_2^-, \dots, \mathcal{B}_m^-$. Let $\mathbf{b}_{ij}^+ \in \mathcal{R}^d$ be the j -th instance in positive bag \mathcal{B}_i^+ . Likewise, $\mathbf{b}_{ij}^- \in \mathcal{R}^d$ is the j -th instance in negative bag \mathcal{B}_i^- . Because not all d dimensions contribute equally for discriminating between positive and negative instances, we also need to give a weight to each dimension in order to maximize DD. Let $\mathbf{w} \in \mathcal{R}^d$ be a weight vector defining the relevance or importance of each feature. Using Bayes' rule and assuming a uniform prior, we look for the point $\mathbf{t} \in \mathcal{R}^d$ with highest DD value as defined by

$$DD(\mathbf{t}, \mathbf{w}) = \prod_{i=1}^n Pr(\mathbf{t} | \mathcal{B}_i^+) \prod_{i=1}^m Pr(\mathbf{t} | \mathcal{B}_i^-)$$

The noisy-or model (see [85] for details) is used in [87] to define the terms in the products. This model is based on two assumptions. First, for \mathbf{t} to be the target concept it is caused (and thus close to) one of the instances in the bag. Second, the probability of an instance not being the target concept is independent of any other

instance not being the target. This yields

$$Pr(\mathbf{t} \mid \mathcal{B}_i^+) = 1 - \prod_j (1 - Pr(\mathbf{b}_{ij}^+ = \mathbf{t}))$$

$$Pr(\mathbf{t} \mid \mathcal{B}_i^-) = \prod_j (1 - Pr(\mathbf{b}_{ij}^- = \mathbf{t}))$$

Finally, the probability $Pr(\mathbf{b}_{ij} = \mathbf{t})$ of an instance being the target concept is defined as a Gaussian based on the distance from the instance to the target concept

$$Pr(\mathbf{b}_{ij} = \mathbf{t}) = \exp(-\|\mathbf{b}_{ij} - \mathbf{t}\|^2)$$

where $\|\mathbf{b}_{ij} - \mathbf{t}\|^2$ is weighted as follows

$$\|\mathbf{b}_{ij} - \mathbf{t}\|^2 = \sum_{l=1}^d w_l^2 (b_{ijl} - t_l)^2$$

where b_{ijl} , w_l , and t_l are the l -th entries of vectors \mathbf{b}_{ij} , \mathbf{w} , and \mathbf{t} respectively. The problem of finding the global maximum DD point is difficult because the size and number of local maxima in the search space is large. However, according to the definition of the DD function, the global maximum DD point is made of contributions from some set of positive bags. Thus, if we start a gradient ascent from every instance in a positive bag, one of them is likely to be closest to the global maximum DD point, contribute the most to it and have a climb directly to it [87]. Therefore, a simple heuristic is applied in [87] to search for the global maximum DD point: start an optimization of the DD function at each instance from every positive bag with uniform weights and record the resulting maximizer (i.e., \mathbf{t} and \mathbf{w}). Then, from among all the maximizers that were found, select the one that resulted in the largest DD value.

Recently, the EM-DD algorithm [157] was developed which combines the DD algorithm with the expectation-maximization (EM) algorithm [24]. EM-DD views

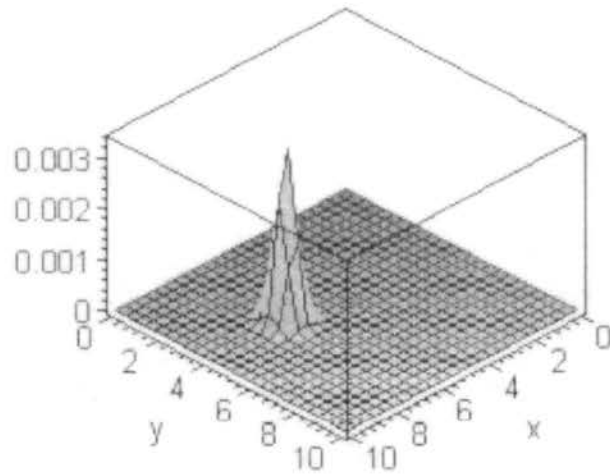


Figure 3.13: The space defined by the Diverse Density function on the plot of Figure 3.12

the knowledge of which instance corresponds to the label of the bag as a missing attribute and applies the EM algorithm to convert the MIL problem to a standard supervised learning problem. In [2], SVMs are used to solve the MIL problem. The proposed extension of the SVM learning approach leads to a mixed integer quadratic program that can be solved heuristically. The mixed integer quadratic program is thus a generalized soft-margin SVM in which the soft-margin criterion is maximized jointly over possible label assignments as well as hyperplanes. Basically, the problem reduces to finding an (optimal) linear separating discriminant such that there is at least one instance from every positive bag in the positive halfspace, while all instances belonging to negative bags are in the negative halfspace.

Chapter 4

Learning with Global Image Representations

In this chapter, we first summarize related work on intra and inter-query learning with global image representations. Next, we present two novel techniques for performing inter-query learning with global image representations. Both techniques use support vector machines (SVM) for learning the class distributions of users' high-level query concepts from retrieval experience. They are based on a relevance feedback (RF) framework that learns one-class support vector machines (1SVM) from retrieval experience to represent the set memberships of users' high-level query concepts and stores them in a "concept database". The "concept database" provides a mechanism for accumulating inter-query learning obtained from previous queries. The geometric view of 1SVMs allows a straightforward interpretation of the density of past interaction in a local area of the feature space and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

The first approach, presented in [42, 36, 35, 40], does a fuzzy classification of a new query into the regions of support represented by the 1SVMs in the "concept database". In this way, past experience is merged with current intra-query learning. The second approach, presented in [39], incorporates inter-query learning into the

query modification and distance reweighing framework. One of the main advantages of these approaches is the capability of making an intelligent initial guess on a new query when the query is first presented to the system.

4.1 Related Work in Intra-Query Learning

Two main RF strategies have been proposed in content-based image retrieval (CBIR): query modification [119], and distance reweighing [11, 61, 103, 117, 127]. Query modification changes the representation of the user's query in a form that is closer (hopefully) to the semantic intent of the user. In particular, query shifting involves moving the query towards the region of the feature space containing relevant images and away from the region containing non-relevant images (See Figure 2.11). Distance reweighing changes the calculation of image to image similarity to strengthen the contribution of relevant image components in regard to the current query. Thus, the task is to determine the features that help the most in retrieving relevant images and increase their importance in determining similarity.

In [117], the weight and representation of each feature is updated according to their ability to discriminate between the set of relevant and non-relevant images in the current query. In [103] a probabilistic feature relevance learning (PFRL) method that automatically captures feature relevance based on RF is presented. It computes flexible retrieval metrics for producing neighborhoods that are elongated along less relevant feature dimensions and constricted along most influential ones (See Figure 4.1). PFRL is an application of the approach described in [31] for learning local feature relevance. In [31], the observation is made that input variables of low relevance can degrade the performance of nearest-neighbor classifiers if they are allowed to be equally influential with those of high relevance in defining the distance from the point to be classified. Thus, if the relative local relevance of each input variable were known, this information would be used to construct a distance metric that provides

an optimal differential weighting for the input variables [31]. In PFRL, retrieved images with RF are used to compute local feature relevance. If we let the class label $y \in \{1, 0\}$ at query $\mathbf{x} \in \mathfrak{R}^d$ be treated as a random variable from a distribution with the probabilities $\{Pr(1 | \mathbf{x}), Pr(0 | \mathbf{x})\}$, we have

$$f(\mathbf{x}) \doteq Pr(y = 1 | \mathbf{x}) = E(y | \mathbf{x})$$

In the absence of any variable assignments, the least-squares estimate for $f(\mathbf{x})$ is

$$E[f] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

where $p(\mathbf{x})$ is the joint density. Now given only that \mathbf{x} is known at dimension $x_i = z_i$.

The least-squares estimate becomes

$$E[f | x_i = z_i] = \int f(\mathbf{x})p(\mathbf{x} | x_i = z_i)d\mathbf{x}$$

where $p(\mathbf{x} | x_i = z_i)$ is the conditional density of the other input variables. In image retrieval, $f(\mathbf{z}) = 1$, where \mathbf{z} is the query. Then

$$[(f(\mathbf{z}) - 0) - (f(\mathbf{z}) - E[f | x_i = z_i])] = E[f | x_i = z_i]$$

represents a reduction in error between the two predictions. Thus, a measure of feature relevance at query \mathbf{z} can be defined as

$$r_i(\mathbf{z}) = E[f | x_i = z_i]$$

The relative relevance can be used as a weighting scheme for a weighted k -nearest

neighbor search where the weight for the i -th dimension is given by

$$w_i(\mathbf{z}) = \frac{e^{vr_i(\mathbf{z})}}{\sum_{j=1}^d e^{vr_j(\mathbf{z})}}$$

where v is a parameter that can be chosen to maximize(minimize) the influence of r_i on w_i . For further details, see [103]. This technique has shown promise in a number of image database applications.

Some methods for incorporating both query shifting and feature relevance weighting have also been proposed [53, 61]. In [53], a retrieval method that combines feature relevance learning and query shifting to achieve the best of both worlds is proposed. This method uses a linear discriminant analysis to compute the new query and exploit the local neighborhood structure centered at the new query by using PFRL.

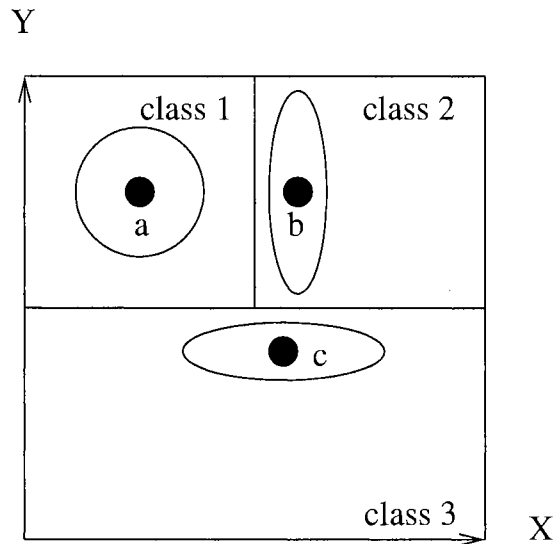


Figure 4.1: Features are unequal in their differential relevance for computing similarity. The neighborhoods of queries **b** and **c** should be elongated along the less relevant Y and X axis respectively. For query **a**, features X and Y have equal discriminating strength.

In [55], distance in the feature space associated with a kernel is used to rank relevant images. An adaptive quasiconformal mapping based on RF is used to generate successive new kernels. The kernel is constructed in such a way that the spatial reso-

lution is contracted around relevant images and dilated around non-relevant images. Then, the distance from the query to new images is measured in this new space. Instead of updating individual feature weights, we could also select from a pre-defined set of similarity measures. For example, in [126], an approach is described that minimizes mean distance between user-labelled relevant images by selecting from a set of pre-defined distance metrics.

In PicHunter [60], a Bayesian framework is used to associate each image with a probability that it corresponds to the user’s query concept. The probability is updated based on the user’s feedback at each iteration. In [140], a “boosting” algorithm is proposed to improve RF learning. Recently, SVM learning has been applied to CBIR systems with RF to significantly improve retrieval performance [18, 56, 141, 156]. Basically, the probability density of relevant images can be estimated by using SVMs. For instance, in [18], a 1SVM is used to include as many relevant images as possible into a hypersphere of minimum size. That is, relevant images are used to estimate the distribution of target images by fitting a tight hypersphere in the non-linearly transformed feature space. In [156], the problem is regarded as a two-class classification problem and a maximum margin hyperplane in the non-linearly transformed feature space is used to separate relevant images from non-relevant images. Many other approaches, such as [54, 102, 161], have provided improved alternatives for utilizing kernel methods in CBIR.

Other classical machine learning approaches, such as decision trees [82], nearest neighbor classifiers [152], and artificial neural networks [71] have also been applied to RF in CBIR. In [82], a decision tree is used to sequentially split the feature space until all points within a partition are of the same class. Then, images that are classified as relevant are returned as the nearest neighbors of the query image.

4.2 Related Work in Inter-Query Learning

Most current RF systems are based on an intra-query-learning-only approach. That is, the system refines the query by using RF supplied by the user and the learning process starts from ground up for each new query. A few approaches [6, 20, 52, 54, 69, 72, 75, 96, 98, 137, 144, 155, 156] attempt inter-query learning (i.e., RF from past queries are used to improve the retrieval performance of the current query). The initial results from those approaches for inter-query learning show a tremendous benefit in the initial and first iteration of retrieval. Inter-query learning thus offers a great potential for reducing the amount of user interaction by reducing the number of iterations needed to satisfy a query.

The approach proposed in [72] was one of the first attempts to explicitly memorize learned knowledge to improve CBIR performance. A correlation network is used to accumulate semantic relevance between image clusters learned from users' RF. In [52, 54] latent semantic analysis (LSI) [23] was used to provide a generalization of past experience. LSI is an important technique in information retrieval. It uses the context of a word's usage (i.e., a document) to uncover the hidden (i.e., latent) meaning of the word. LSI creates a semantic space by applying the singular value decomposition to a term-by-document matrix \mathbf{M} . Each column of \mathbf{M} represents a document. The components of the column represent the relationship of the term to the document (such as a frequency weight of the occurrences of the term in the document). The term-by-document matrix is then approximated by using the k largest singular values and their associated singular vectors:

$$\underbrace{\mathbf{M}}_{t \times d} = \underbrace{\mathbf{U}}_{t \times r} \underbrace{\mathbf{S}}_{r \times r} \underbrace{\mathbf{V}^T}_{r \times d} \approx \underbrace{\hat{\mathbf{M}}}_{t \times d} = \underbrace{\hat{\mathbf{U}}}_{t \times k} \underbrace{\hat{\mathbf{S}}}_{k \times k} \underbrace{\hat{\mathbf{V}}^T}_{k \times d}$$

where t is the number of terms, d is the number of documents, r is the rank of \mathbf{M} , \mathbf{U} and \mathbf{V} are orthonormal, and \mathbf{S} is diagonal. The $\hat{\mathbf{M}}$, $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}$ matrices are

the approximations of the respective matrices when using just the k largest singular values. To process a previously unknown query document, first a *pseudo-document*, $\hat{\mathbf{d}}$, is created as a vector of its component terms. This vector is then projected into the semantic space by $\mathbf{q} = \hat{\mathbf{U}}^T \hat{\mathbf{d}}$. The distance of the query to each of the documents is then the distance of \mathbf{q} to the corresponding column of $\hat{\mathbf{S}}\hat{\mathbf{V}}^T$.

In [54], the images in a database are viewed as the fundamental vocabulary of the system. The RF from each query is considered as a document composed of many terms (images) (See Figure 4.2). Thus, assuming that the terms of a document have a latent semantic relationship, it is possible to use LSI to capture inter-query learning.






	Query1	Query2	Query3
	1	0	0
	1	0	0
	0	1	0
	1	0	0
	0	0	1

Figure 4.2: LSI approach for inter-query learning. Each column in the matrix represents a query and the set of marked relevant (1) and non-relevant(0) retrieved images. LSI can be performed on the matrix to obtain useful inter-query learning.

Both [20], [75], and [155], take the approach of complete memorization of prior history. In PicHunter [20], the entire history of user selections contributes to the system's estimate of the user's concept. To accomplish this, Bayesian learning based on a probabilistic model of the user's behavior is used. The predictions of this model are

combined with the selections made during a query session to estimate the probability associated with each image. These probabilities are then used to retrieve images. In [75] the correlation between past image labelling is merged with low-level features to rank images for retrieval. The model estimates the semantic correlation between two images based on their co-occurrence frequency (i.e., the number of query sessions in which both images were labelled relevant). Intuitively, the larger the co-occurrence frequency of two images is, the more likely that they are semantically similar. Given a query \mathbf{x} , the semantic similarity to each image is initialized to its feature-based similarity. Then, semantic similarities are iteratively updated based on correlation with top-ranked images. Thus, images having strong correlations with the top-ranked images are likely to have a high semantic similarity with \mathbf{x} , even if their feature-based similarity is low [75].

In [155] the extra inter-query information is efficiently encoded by adding a virtual feature (VF) to the feature vector of an image. Initially, the VF of each image is empty. Given a query \mathbf{x} , the k nearest neighbor images to it are retrieved and the user labels each of them as relevant or non-relevant. Then, a number from a system counter is concatenated to the VFs of all user-labelled relevant images to indicate that they deliver the same concept as \mathbf{x} . To determine relevance between \mathbf{x} and database images, the VF of \mathbf{x} is computed as the concatenation of the VFs of all user-labelled relevant images in the previous RF iteration. The VFs of \mathbf{x} and the database images are then used in a probabilistic dissimilarity measure that dynamically adjusts the distance between \mathbf{x} and the database images [155]. One of the shortcomings of this method is that it needs at least one RF iteration and thus inter-query learning cannot be used to improve the performance in the initial retrieval set.

In [98], the log files of the *Viper* system are used to perform feature relevance weighting. In [144], a Bayesian approach is presented for both intra and inter-query learning. Self-Organizing Maps are used for inter-query learning in the PicSOM

system [69]. In [96], a multilayer method for image organization and searching is presented. User interaction is combined with offline image processing and knowledge from previous interactions is remembered. In [137], a framework for accumulating RF and constructing a relevance graph for later usage is presented. A general active learning framework is proposed in [156]. The framework is used to guide hidden annotations in order to improve retrieval performance. In [30], a long-term similarity learning algorithm which uses RF from previous sessions is given. The *MetaSeek* system presented in [6] selects and queries its target image search engines according to their success under similar query conditions in previous searches. For this purpose, the system keeps a performance database in which the performance of each target engine is kept according to the user’s RF.

4.3 Inter-Query Learning with One-Class Support Vector Machines

We present two novel RF approaches for performing inter-query learning in CBIR with global image representations. By accumulating experience in the form of users’ RF, it is possible to learn the class distributions of users’ high-level concepts. Then, this inter-query learning (in the form of high-level concept classification) can be exploited to improve retrieval performance. We require a long-term memory structure for the representation of inter-query learning accumulated from queries over time. Because of their straightforward interpretation as the density of past interaction in a local area of the feature space, we have chosen 1SVMs as this long-term learning structure. Both approaches are based on using 1SVMs for learning the class distributions of users’ query concepts from retrieval experience. They are based on a RF framework that learns 1SVMs from retrieval experience to represent the set memberships of users’ query concepts and stores them in a “concept database”. The “concept database”

provides a mechanism for accumulating inter-query learning obtained from previous queries. The geometric view of 1SVMs allows a straightforward interpretation of the density of past interaction in a local area of the feature space and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

Let $\mathbf{x} \in \mathbb{R}^d$ be the input space representation of the query image (i.e., vector of feature values extracted from the image), and $\mathbf{w} \in \mathbb{R}^d$ be the feature weights for an arbitrary distance/similarity measure. For simplicity, from now on for any image, we will use its input space representation \mathbf{x} to refer also to the image itself. Thus, when using \mathbf{x} , it will be clear from the context whether we are referring to the image itself or to its representation in input space. Let $\mathcal{R} = \{(\mathbf{x}_i, y_i)\}_1^n$ be the set of all cumulative retrievals for \mathbf{x} , where y_i is either 1 (relevant image) or 0 (non-relevant image) marked by the user as the class label associated with \mathbf{x}_i . Let $\mathcal{R}^+ = \{\mathbf{x}_i \mid (\mathbf{x}_i, 1) \in \mathcal{R}\}$ and $\mathcal{R}^- = \{\mathbf{x}_i \mid (\mathbf{x}_i, 0) \in \mathcal{R}\}$ be the set of cumulative relevant and non-relevant retrievals, respectively. Let $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ with $d' \geq d$ be the mapping from input space to feature space. Thus, $\Phi(\mathbf{x})$ refers to the feature space representation of \mathbf{x} .

At the end of the search session for \mathbf{x} , we use \mathcal{R}^+ as training data for a 1SVM. Then, we store the resulting 1SVM in the “concept database”. Let the descriptor of the corresponding hypersphere be $\mathcal{H} = \{\mathcal{R}, \mathbf{a}, r\}$, where \mathbf{a} and r are its center and radius respectively. The basic idea is that a future query image that falls within the same region of support is classified by the 1SVM as having the same semantics. Thus, inter-query learning can provide us with a cue about the semantics of an image (See Figure 4.3).

The M-tree [19] data structure (described in Section 2.4) is used for the efficient search of nearest neighbor images in feature space. We use M-trees for the efficient search of both historical information and images in the database. The image M-tree contains all the images in the database and the history M-tree contains the learned

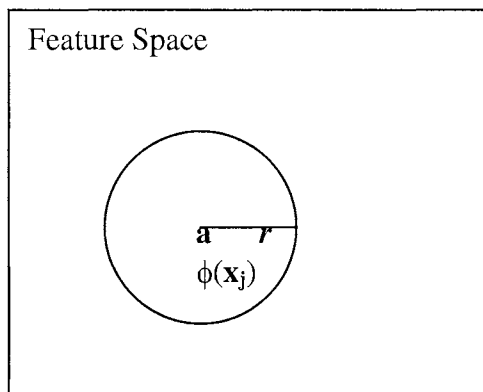


Figure 4.3: Basic idea of first approach. The 1SVM generated with \mathcal{R}^+ as training data at the end of a query session for query \mathbf{x}_i . The (feature space) representation of a future query \mathbf{x}_j falls inside this hypersphere. The 1SVM classifies \mathbf{x}_j into the same concept as \mathbf{x}_i .

1SVMs (i.e., the “concept database”).

4.3.1 Overview of First Approach

By doing a fuzzy classification of a query into the regions of support represented by the 1SVMs in the “concept database”, past experience is merged with current intra-query learning. Figure 4.4 shows a diagram of the proposed method. The approach that is used for selecting the images in the retrieval set is based on exploiting both intra and inter-query learning. After each RF iteration, \mathcal{R}^+ is used as training data for a 1SVM. Then, intra-query learning is exploited by including $(w_{intra})k$ nearest neighbor images to the hypersphere’s center \mathbf{a} into the retrieval set, where $0 \leq w_{intra} \leq 1$ is the intra-query learning weight and k is the number of images in the retrieval set. Initially (i.e., before any RF iterations), $\mathbf{a} = \Phi(\mathbf{x})$. The remaining $(1 - w_{intra})k$ images in the retrieval set are obtained by exploiting the accumulated inter-query learning in the “concept database”. Thus, the ratio of intra to inter-query learning that is used in processing a query is $w_{intra} : (1 - w_{intra})$. We now explain how the remaining $(1 - w_{intra})k$ “inter-query learning” images are selected.

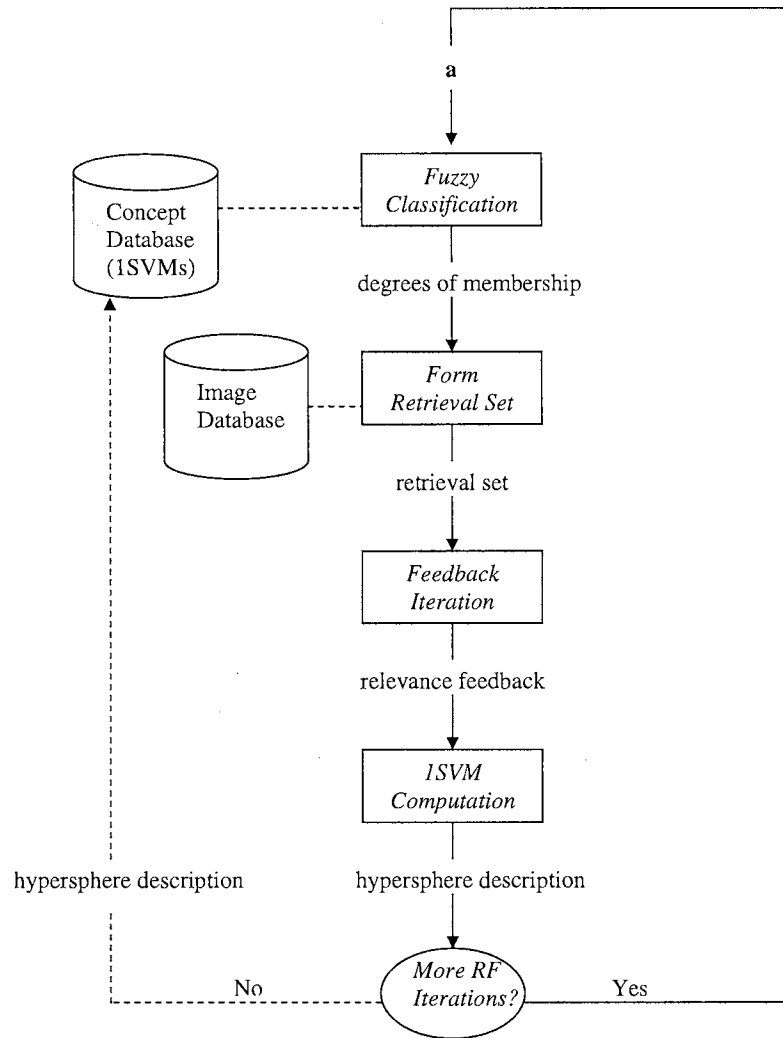


Figure 4.4: Diagram of first approach.

In order to integrate the prior experience in the “concept database” with \mathbf{x} , a fuzzy classification of \mathbf{x} into the existing regions of support (i.e., 1SVMs) is performed. Thus, the “concept database” is searched and it is determined whether \mathbf{a} falls into any of the accumulated 1SVMs. Because it is very common for an image to be ascribed into many different concepts, we expect to have queries that fall into many hyperspheres. One possible way of exploiting inter-query learning would be to perform a hard classification by selecting $(1 - w_{intra})k$ nearest neighbor images to the closest hypersphere’s center (i.e., closest prototype). However, this is not a very good strategy

since a query may be a member of several concept sets (i.e., it may fall into many hyperspheres). Thus, it may as well be ascribed to the concept corresponding to any one of the other 1SVMs. Furthermore, a query may be ascribed to a combination of different concepts.

The results of experiments conducted in [3] for learning users' text preferences suggest that, for simple queries (i.e., queries that can be ascribed to one concept), a purely exploitative strategy delivers good performance. However, for complex queries (i.e., queries that can be ascribed to more than one concept), there is a tradeoff between faster learning of the user's query concept and the delivery of more relevant documents. Therefore, instead, we use the ideas from possibilistic cluster analysis [57] and assign a degree of membership to each one of the 1SVMs (i.e., to each cluster) according to the degree by which \mathbf{x} can be ascribed to its particular concept.

Given a set of points, the fuzzy c -means algorithm [57] searches for an optimal set of clusters. The clusters are represented by their corresponding centers and each point has a degree of membership in each cluster, which models the degree of the point belonging to the cluster [57]. In our case, the set of clusters (in the form of 1SVMs) is formed by the historical interaction of users with the system. Let $\{\mathcal{H}_i\}_1^m$, where $\mathcal{H}_i = \{\mathcal{R}_i, \mathbf{a}_i, r_i\}$, be the set of hyperspheres into which \mathbf{a} falls. We then use the following function to assign a membership of \mathbf{x} into each hypersphere

$$\mu(\mathbf{x}, \mathcal{H}_i) = \frac{1}{\sum_{j=1}^m \frac{\text{dist}_F(\mathbf{a}, \mathbf{a}_i)}{\text{dist}_F(\mathbf{a}, \mathbf{a}_j)}}, \quad i = 1, 2, \dots, m$$

where dist_F refers to the feature space distance. Therefore, the degree of membership of \mathbf{x} into a 1SVM is based on the relative distances between \mathbf{a} and the centers of all hyperspheres into which \mathbf{a} falls. If $\Psi(\mathcal{H}_i)$ denotes the concept that is embodied by hypersphere \mathcal{H}_i then the belief (or our degree of confidence) that \mathbf{x} is delivering concept $\Psi(\mathcal{H}_i)$ is equal to $\mu(\mathbf{x}, \mathcal{H}_i)$.

To form the retrieval set, sample representative images from each hypersphere into which \mathbf{a} falls are included. The number of representatives that a particular concept $\Psi(\mathcal{H}_i)$ has in the retrieval set is proportional to $\mu(\mathbf{x}, \mathcal{H}_i)$. Thus, the number of images of concept $\Psi(\mathcal{H}_i)$ that appear in the retrieval set will be greater than the number of images of concept $\Psi(\mathcal{H}_j)$ whenever $\mu(\mathbf{x}, \mathcal{H}_i) > \mu(\mathbf{x}, \mathcal{H}_j)$. Because \mathbf{a} may fall into many hyperspheres but only $(1 - w_{intra})k$ “inter-query-learning” images are to be included in the retrieval set, priority is given to hyperspheres with higher μ value. Thus, after $(1 - w_{intra})k$ images are selected, the remaining hyperspheres with smaller μ values are ignored.

The retrieval set is thus formed by exploiting both intra and inter-query learning. Then, the user evaluates the relevance of images in the retrieval set and \mathcal{R}^+ is used as training data for a 1SVM. The center \mathbf{a} of the resulting hypersphere becomes the new query location for the second round of RF and this process continues until the user is satisfied with the results or quits. When the session is over, the final 1SVM is stored in the “concept database”. The algorithm for the first approach is summarized in Figure 4.5.

One of the weaknesses of this approach is that inter-query learning is represented by a constantly growing number of (possibly overlapping) 1SVMs (i.e., regions) in the feature space. Thus, as previously mentioned in Chapter 1, summarization may be desirable when the amount of inter-query learning (i.e., the size of the “concept database”) is very large.

4.3.1.1 Summarizing Inter-Query Learning

In the proposed approach, inter-query learning is accumulated in the form of 1SVMs. However, this way of storing inter-query learning results in a constantly increasing number of (possibly overlapping) clusters (i.e., 1SVMs) in the feature space. In this section, we alleviate this problem by incorporating an implicit cluster-merging pro-

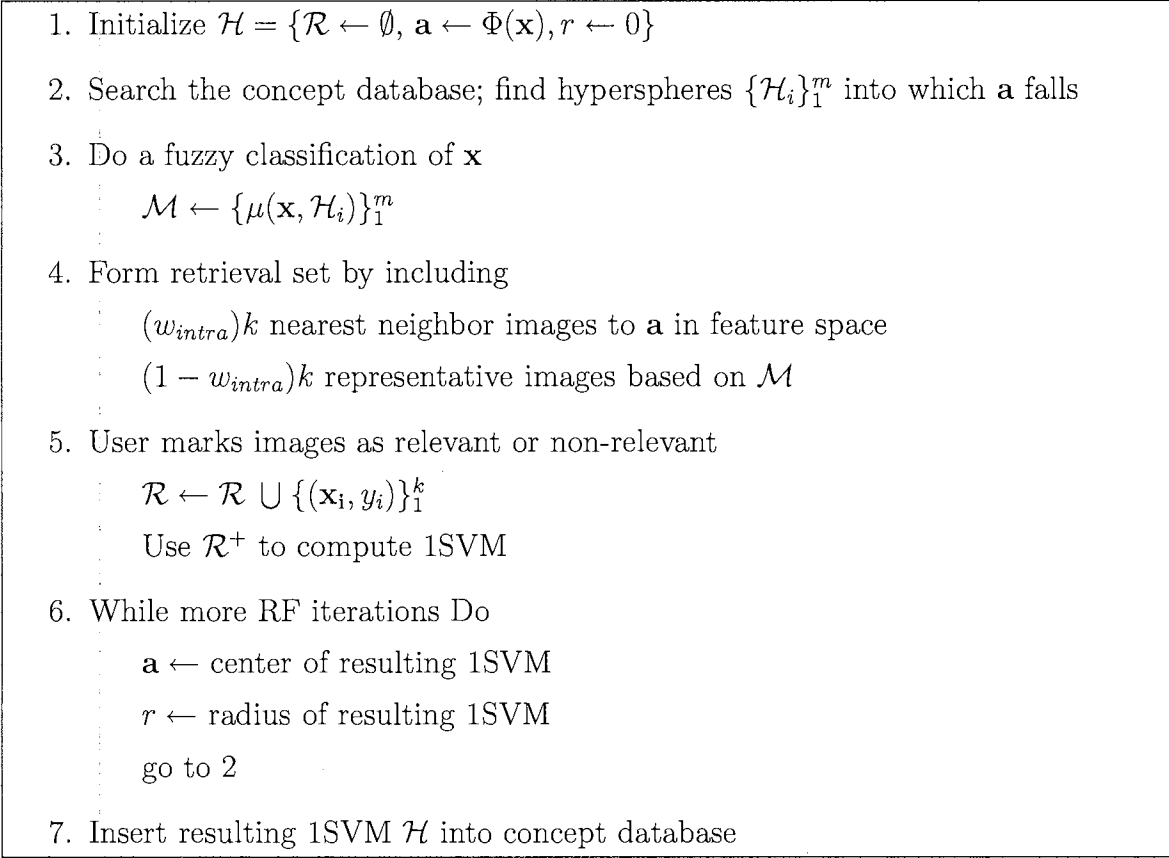


Figure 4.5: Algorithm of First Approach.

cess to incrementally summarize the derived inter-query learning. The similarity measure that is used for clustering 1SVMs and classifying the query takes both distance in feature space and a probabilistic perceptual closeness (based on users' RF) into consideration. The main advantage of doing this is that the system becomes scalable and query processing can be accelerated by considering only a small number of cluster representatives, rather than the entire set of accumulated 1SVMs.

Figure 4.6 shows a diagram of the modified approach. The difference is that the $(1 - w_{intra})k$ "inter-query learning" images in the retrieval set are nearest neighbor images to the cluster representative that is closest to \mathcal{H} . Also, when the query session is over, the resulting 1SVM \mathcal{H} is not directly added to the "concept database". Instead, an implicit cluster-merging process takes place. This process determines, from a fixed number of cluster representatives, the most similar one to \mathcal{H} and com-

bins both. Thus, inter-query learning is summarized by a small number of cluster representatives.

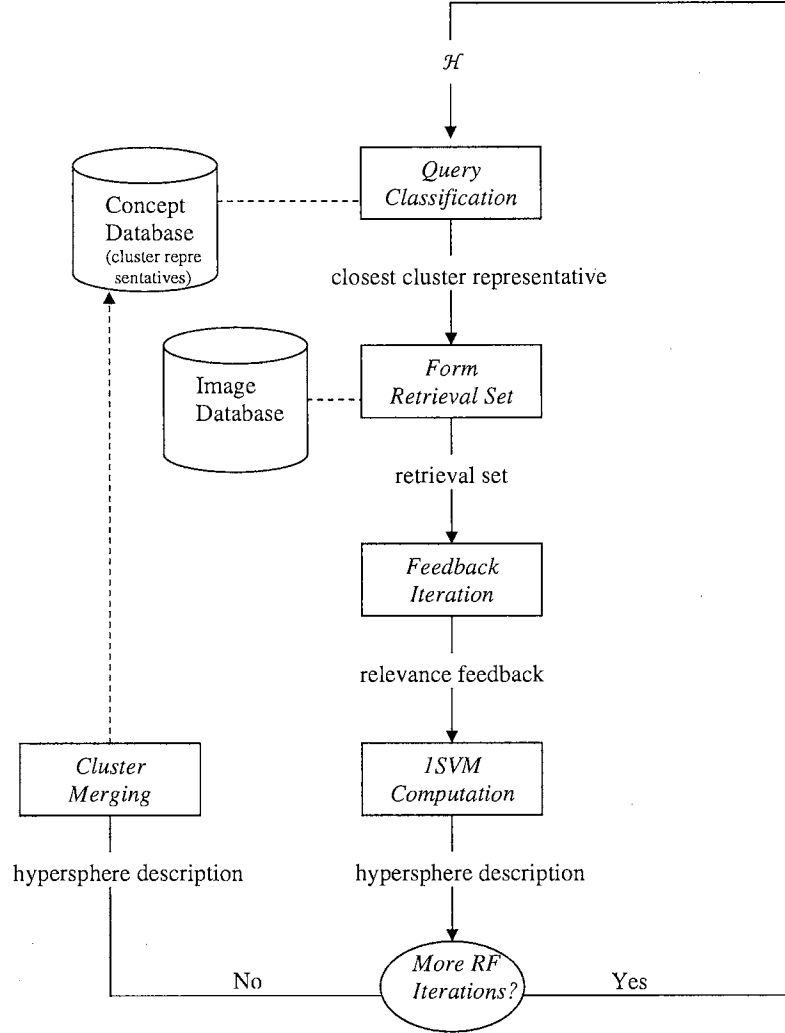


Figure 4.6: Diagram of Modified First Approach.

The accumulated intra-query learning at the end of the RF iterations for \mathbf{x} is given by \mathcal{R} . The center \mathbf{a} of \mathcal{H} is

$$\mathbf{a} = \sum_{\mathbf{x}_i \in \mathcal{R}^+} \alpha_i \Phi(\mathbf{x}_i)$$

where $\alpha_i \in \mathfrak{R}$ is calculated by the 1SVM computation. Instead of storing in the

“concept database” each hypersphere \mathcal{H} that results from each query \mathbf{x} , let the accumulated inter-query learning \mathcal{E} be summarized by a fixed number of cluster representatives. A cluster representative $\mathcal{C} \in \mathcal{E}$ is defined as follows

$$\begin{aligned}\mathcal{C} &= \{\mathbf{p}, \mathcal{W}\} \\ \mathcal{W} &= \{(\mathbf{x}_i, y_i, w_i)\}_1^m\end{aligned}$$

where $\mathbf{p} \in \mathbb{R}^d$ is the pre-image of \mathcal{C} 's center in feature space, which is computed as explained later. Thus, the center of \mathcal{C} in feature space is $\mathbf{c} = \Phi(\mathbf{p})$. The m images (each with corresponding “semantic weight” $w_i \in \mathbb{R}$) in \mathcal{W} contribute to $\Psi(\mathcal{C})$ (i.e., \mathcal{C} 's high-level concept). Intuitively, \mathcal{W} describes the high-level semantics (i.e., the concept) associated with \mathcal{C} . For each \mathbf{x}_i , let the set $\mathcal{A}_{\mathbf{x}_i}$ be defined as follows

$$\mathcal{A}_{\mathbf{x}_i} = \sum_{\mathcal{C} \in \mathcal{E}} \{w_i \mid (\mathbf{x}_i, 1, w_i) \in \mathcal{W}\}$$

That is, $\mathcal{A}_{\mathbf{x}_i}$ is the sum of all “semantic weights” of \mathbf{x}_i from all cluster representatives in which \mathbf{x}_i appears as a relevant image. Given that the user has labelled \mathbf{x}_i as a relevant image and given \mathcal{E} , we define the single-image probability that the user's concept is $\Psi(\mathcal{C})$ as follows

$$p(\Psi(\mathcal{C}) \mid \mathbf{x}_i, \mathcal{E}) = \frac{w_i}{\mathcal{A}_{\mathbf{x}_i}} \text{ if } \mathbf{x}_i \in \mathcal{W}, \quad 0 \text{ otherwise}$$

Thus, among all $\mathcal{C} \in \mathcal{E}$ in which \mathbf{x}_i is relevant, the \mathcal{C} in which \mathbf{x}_i has the largest “semantic weight” has the highest probability of matching the user's concept. The total probability for each $\mathcal{C} \in \mathcal{E}$ is obtained by summing the single-image probabilities of images that are co-occurring and relevant in both \mathcal{W} and \mathcal{R} . Therefore, the $\mathcal{C} \in \mathcal{E}$ that has the largest semantic overlap will have the highest probability of coinciding with the user's concept. Thus, given \mathcal{R} and \mathcal{E} , we define the overall probability that

the user’s concept is $\Psi(\mathcal{C})$ as follows

$$P(\Psi(\mathcal{C}) \mid \mathcal{R}, \mathcal{E}) = \frac{\sum_{\mathbf{x}_i \in \mathcal{S}} p(\Psi(\mathcal{C}) \mid \mathbf{x}_i, \mathcal{E})}{|\mathcal{G}|}$$

where

$$\begin{aligned} \mathcal{S} &= \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{R}^+ \text{ and } (\mathbf{x}_i, 1, *) \in \mathcal{W}\} \\ \mathcal{G} &= \{\mathbf{x}_i \mid (\mathbf{x}_i, *, *) \in \mathcal{R} \text{ and } (\mathbf{x}_i, *, *) \in \mathcal{W} \\ &\quad \text{and not}(\mathbf{x}_i \in \mathcal{R}^- \text{ and } (\mathbf{x}_i, 0, *) \in \mathcal{W})\} \end{aligned}$$

where $*$ is a “don’t-care” symbol indicating that the corresponding tuple element is ignored when determining set membership. For each cluster representative \mathcal{C} , we compute its distance to \mathcal{H} with the following measure

$$Dist(\mathcal{C}, \mathcal{H}) = (1 - 2P(\Psi(\mathcal{C}) \mid \mathcal{R}, \mathcal{E}))\Delta + \|\mathbf{c} - \mathbf{a}\|^2$$

where $0 \leq \Delta \leq 1$ is a distance adjustment. Thus, the distance between \mathbf{c} and \mathbf{a} in feature space is adjusted based on the probability that the user’s concept is $\Psi(\mathcal{C})$. Therefore, the proposed measure adjusts the distance between the resulting hypersphere and the cluster representatives based on an estimate of their conceptual similarity, which is derived from both the current intra-query and accumulated inter-query learning.

As previously shown, the center of a hypersphere (i.e., 1SVM) is expressed as an expansion in terms of its corresponding support vector images. The center of a cluster representative $\mathcal{C} \in \mathcal{E}$ is the mean of the centers of all the hyperspheres that have been merged with \mathcal{C} . Therefore, its location in feature space would have to be expressed in terms of the support vector images of all of those hyperspheres’ centers. However, the complexity of distance computations scales with the number of support vectors.

Thus, this would result in a system that is both considerably slower and not scalable since the memory needed for storing cluster representatives' centers would continually increase as more queries are processed. This fact motivated us to use pre-images for approximating the centers of cluster representatives.

The pre-image problem is to find a point $\mathbf{x} \in \mathcal{R}^d$ in input space such that, for a given $\mathcal{V} \in \mathcal{R}^{d'}$ in feature space, $\mathcal{V} = \Phi(\mathbf{x})$. However, since the map Φ into the feature space is nonlinear, this is often impossible (i.e., the pre-image \mathbf{x} may not exist). Therefore, instead, we can find an approximate pre-image $\mathbf{p} \in \mathcal{R}^d$ such that $\|\mathcal{V} - \Phi(\mathbf{p})\|^2$ is minimized [124] (See Figure 4.7).

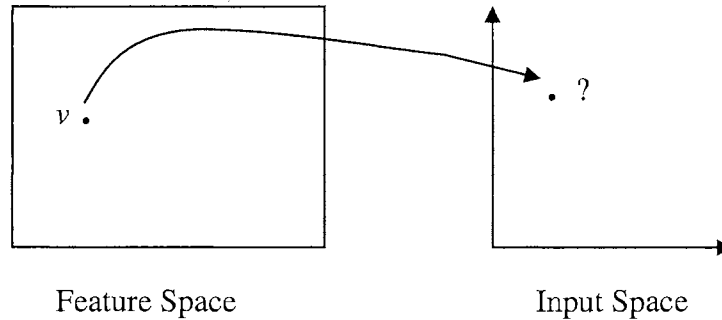


Figure 4.7: The pre-image problem is to find a point \mathbf{x} in input space such that, for a given point \mathcal{V} in feature space, $\mathcal{V} = \Phi(\mathbf{x})$. Not every point in the feature space is necessarily the image of some point in the input space. Therefore, finding an exact pre-image point is not always possible.

Traditional methods [12, 95] solve this optimization problem by performing iteration and gradient descent. The disadvantage of those methods is that the optimization procedure can be expensive and may result in finding a local optimum [124]. The basic idea of the approach presented in [70] is to use distance constraints in the feature space to approximate the location of the pre-image. That is, distances between \mathcal{V} and its neighbors in feature space are found. Then, the corresponding input-space distances are computed and used to constraint the location of the pre-image [70] (See Figure 4.8). We apply this method to our problem.

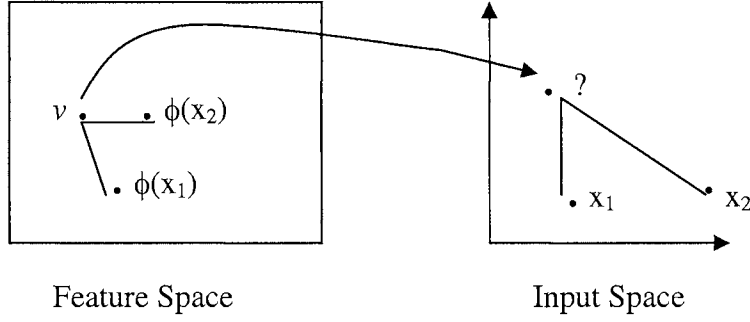


Figure 4.8: Method for Estimating Location of Pre-Image. The distances between ϑ and neighboring points can be used to constraint the location of the pre-image in input space.

Let d_{ci} be the feature space distance between a cluster representative's center \mathbf{c} and an image \mathbf{x}_i . Using the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2}$ we solve for the corresponding input-space distance \hat{d}_{ci} between \mathbf{c} and \mathbf{x}_i

$$\hat{d}_{ci} = -\sigma^2 \log\left(1 - \frac{d_{ci}}{2}\right)$$

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ be the k nearest neighbor images to \mathbf{c} in feature space. Each image \mathbf{x}_i is represented by a d -dimensional feature vector, $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T$. Then, the problem is to find the least-squares solution $\mathbf{c} = [c_1, c_2, \dots, c_d]^T$ to the system of equations

$$\|\mathbf{c} - \mathbf{x}_i\|^2 = \hat{d}_{ci}, \quad i = 1, \dots, k$$

After expanding, grouping like terms, and subtracting the k -th equation from the rest we obtain a system of the form $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} is a $(k-1)$ by d matrix with row vectors

$$[2(x_{k1} - x_{i1}), \quad 2(x_{k2} - x_{i2}), \quad \dots, \quad 2(x_{kd} - x_{id})], \quad i = 1, \dots, k$$

and \mathbf{b} is a $(k-1)$ by 1 vector with rows

$$[-(\mathbf{x}_i \cdot \mathbf{x}_i) + (\mathbf{x}_k \cdot \mathbf{x}_k) - \hat{d}_{ck} + \hat{d}_{ci}], \quad i = 1, \dots, k$$

and $\mathbf{x} = [c_1, c_2, \dots, c_d]^T$. We then use the singular value decomposition of \mathbf{A} to solve this least-squares problem.

The merging of clusters (i.e., hyperspheres) is the core of our modified first approach. It is used to accelerate query processing by considering only a small number of cluster representatives rather than the entire set of hyperspheres. The c -means [90] algorithm is one of the simplest and most commonly used clustering algorithms. It starts with a random partitioning of patterns to clusters and keeps reassigning patterns to clusters based on their similarity to cluster centers until there is no reassignment of any pattern from one cluster to another or a convergence criterion is met [90]. We use a modified c -means algorithm in which training is done incrementally one pattern (i.e., one hypersphere) at a time as successive queries are processed. The modified algorithm is summarized in Figure 4.9. The proposed method for merging a hypersphere with the closest cluster representative is composed of two stages. First, move the cluster’s center in feature space towards the hypersphere’s center. Then, update the cluster’s concept so that it is more similar to the hypersphere’s semantics. At the first stage, a weighted average between the support vector images that make up the hypersphere’s center and the cluster’s pre-image is taken. Then, the pre-image of the cluster center’s new location in feature space is computed. At the second stage, the union between images in \mathcal{H} and \mathcal{W}^{winner} is taken. Then, the “semantic weight” of co-occurring relevant images is increased. Similarly, the “semantic weight” of images with opposite RF is decreased. For any cluster representative \mathcal{C} , only a fixed number of images is kept in \mathcal{W} . Thus, when the number of images in \mathcal{W} is too large, images with lowest “semantic weight” are deleted.

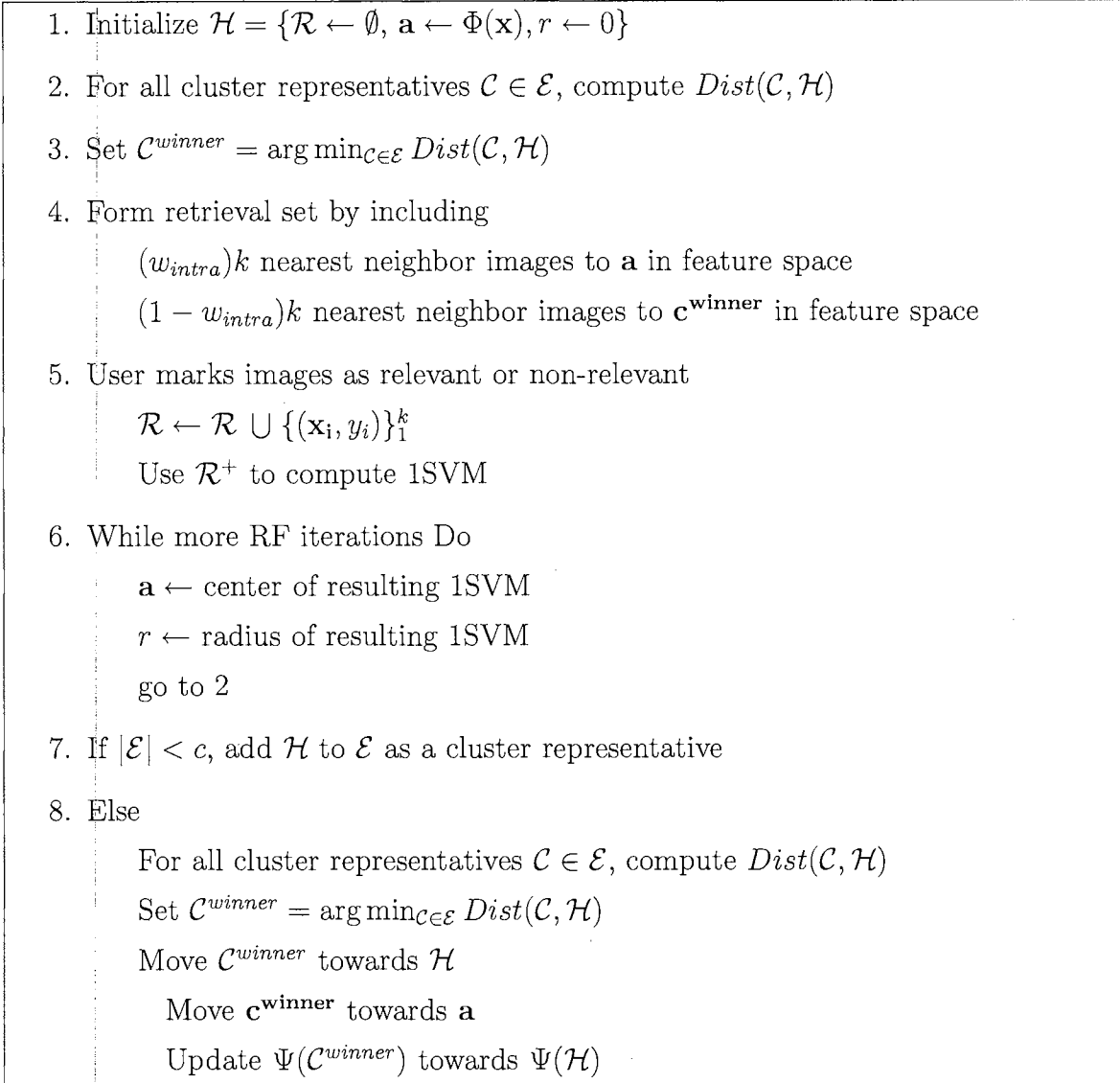


Figure 4.9: Algorithm of Modified First Approach.

4.3.2 Overview of Second Approach

The second approach incorporates inter-query learning into the query modification and distance reweighing framework. For example, a local initial distance metric is created that is more informed than the commonly used default of Euclidean distance. The semantic similarity of the current query with a set of past queries is used to control the exploitation of inter-query learning from historical data.

Suppose that we have a retrieval method that performs query modification and

distance reweighing. Then, after each RF iteration, \mathbf{x} and \mathbf{w} are modified according to the particular query modification and/or distance reweighing approach (See Figure 4.10).

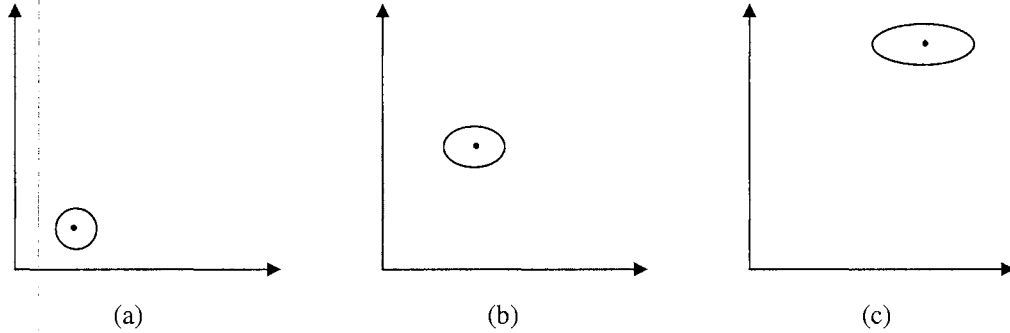


Figure 4.10: Query modification and distance reweighing framework: a) initial (input space) location of query \mathbf{x} and feature weights \mathbf{w} (the circle indicates an equal weighting of every feature dimension); b) new query location \mathbf{x}' and new feature weights \mathbf{w}' after 1 iteration of RF; c) new query location \mathbf{x}'' and feature weights \mathbf{w}'' after two RF iterations.

For example, PFRL [103] (described in Section 4.1) combined with query shifting could be used. PFRL becomes less appealing in situations where all the input variables have the same local relevance and yet retrieval performance might still be improved by simple query shifting towards $\mu_r = \frac{1}{|\mathcal{R}^+|} \sum_{\mathbf{x} \in \mathcal{R}^+} \mathbf{x}$. A PFRL algorithm combined with query shifting (PFRL+ μ_r) is summarized in Figure 4.11.

Note that training data in PFRL+ μ_r (for computing the relative feature relevances used to determine the k nearest neighbors in the next iteration) consists of all previous (cumulative) retrieved images. This is an improvement over the original PFRL (as presented in [103]) where training data consists only of images retrieved at the current RF iteration.

At the end of the search session for \mathbf{x} , intra-query learning is given by \mathcal{R} and by the final values for \mathbf{x} and \mathbf{w} . In general, this intra-query learning is lost when the search session is over. We now describe our proposed method for accumulating and

1. Initialize $\mathbf{w} \leftarrow \{1/d\}^d$, $\mathcal{R} \leftarrow \emptyset$
2. Find k nearest neighbor images to \mathbf{x} using \mathbf{w}
3. User marks the k images
4. While More RF Iterations Do
 - $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\mathbf{x}_i, y_i)\}_1^k$
 - Update \mathbf{w} using PFRL with \mathcal{R}
 - Compute μ_r ; $\mathbf{x} \leftarrow \mu_r$
 - Find k nearest images to \mathbf{x} using \mathbf{w}
 - User marks the k images

Figure 4.11: PFRL with Query Shifting (PFRL+ μ_r).

incorporating inter-query learning into this query modification and distance reweighing framework. As in the first approach, at the end of the search session for \mathbf{x} , we use \mathcal{R}^+ as training data for a 1SVM. Then, we associate the final values for \mathbf{x} and \mathbf{w} with the resulting region of support (i.e., hypersphere) \mathcal{H} in feature space. The basic idea is that future query images that fall within the same region of support can take advantage of inter-query learning. Thus, instead of “starting from scratch”, the previously learned final values for \mathbf{x} and \mathbf{w} can be exploited (See Figure 4.12).

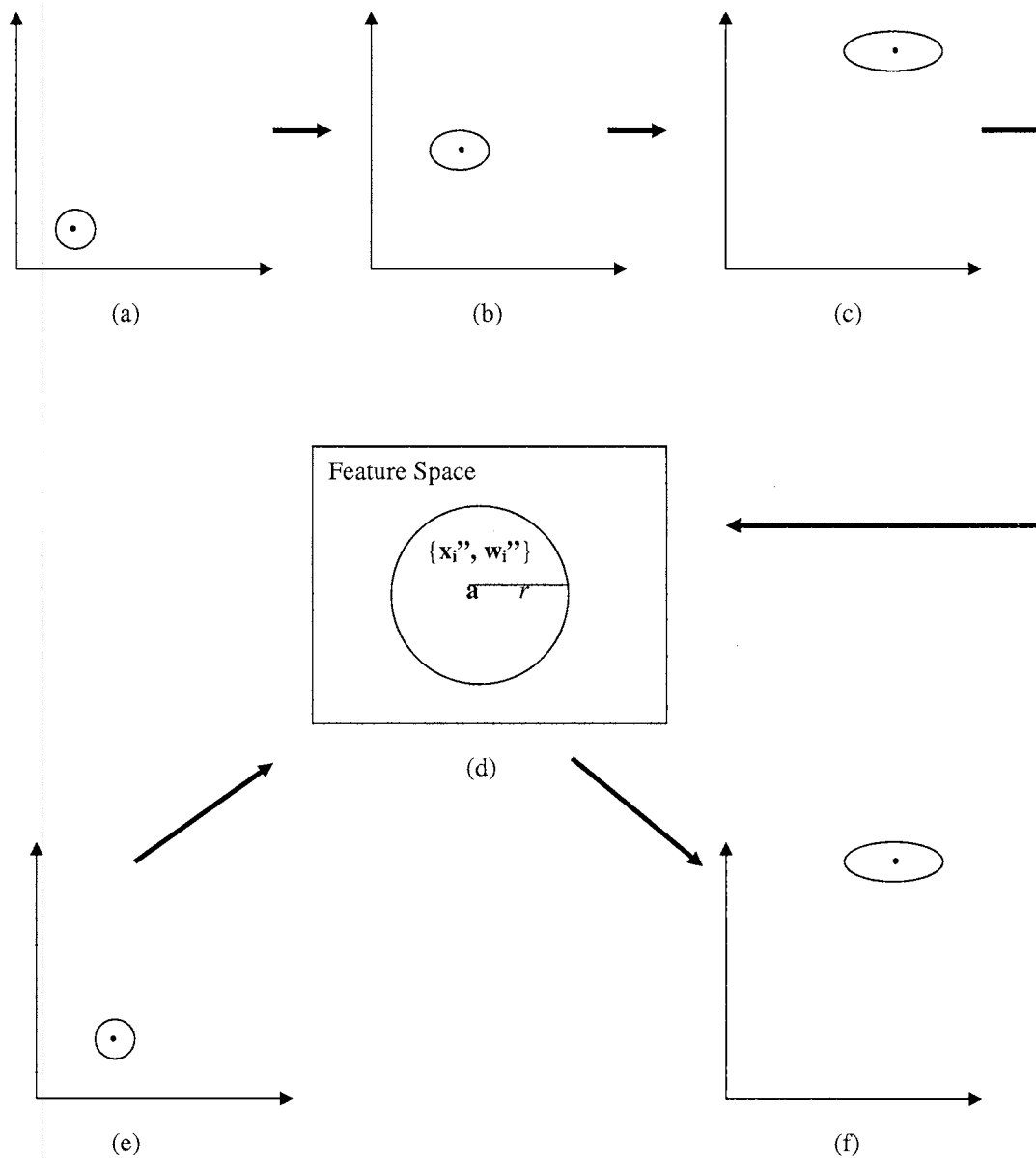


Figure 4.12: Basic idea of second approach: a) initial (input space) location \mathbf{x}_i and feature weights \mathbf{w}_i for the i -th query image; b) new query location of \mathbf{x}_i' and new feature weights \mathbf{w}_i' after one iteration of RF; c) new query location \mathbf{x}_i'' and new feature weights \mathbf{w}_i'' after two RF iterations; d) 1SVM is computed based on \mathcal{R}^+ and $\{\mathbf{x}_i'', \mathbf{w}_i''\}$ is associated with resulting hypersphere; e) the (feature space) representation of future query \mathbf{x}_j falls inside this hypersphere. The 1SVM classifies \mathbf{x}_j into the same query concept as \mathbf{x}_i ; f) more informed initial value for \mathbf{x}_j and \mathbf{w}_j is obtained based on stored $\{\mathbf{x}_i'', \mathbf{w}_i''\}$.

It is common for an image to be ascribed into different concepts by different users or to be a combination of different concepts. Therefore, we expect to have overlapping regions of support and thus queries that fall into more than one hypersphere (See Figure 4.13). Thus, in order to identify the regions of support that are most likely to contain relevant images, we have to determine semantic similarity between the query image’s concept (i.e., $\Psi(\mathbf{x})$) and the concepts associated with the hyperspheres into which $\Phi(\mathbf{x})$ falls. By storing the user’s RF about each retrieved image on a particular search session (i.e., \mathcal{R}) along with the resulting hypersphere \mathcal{H} , we are able to capture the semantics of the retrieval concept associated with \mathcal{H} (i.e., $\Psi(\mathcal{H})$). This information can then be used as a basis for determining semantic similarity. Therefore, in addition to 1SVM parameters, other information is stored in a hypersphere descriptor, which is extended as follows

$$\mathcal{H} = \{\mathbf{x}, \mathbf{w}, \mathcal{R}, \mathbf{a}, r\}$$

where \mathbf{x} is the final (input space) query location, \mathbf{w} are the final feature weights, and \mathbf{a} , and r are the center and radius of the resulting hypersphere respectively.

4.3.2.1 Semantic Similarity

For every query image \mathbf{x} , there is a corresponding hypersphere \mathcal{H} (obtained by training a 1SVM on the user’s cumulative RF). Thus, we only need to be able to determine semantic similarity between concepts associated with hyperspheres. That is, if $\Phi(\mathbf{x})$ falls into more than one hypersphere, we compute the semantic similarity between every hypersphere into which $\Phi(\mathbf{x})$ falls and \mathbf{x} ’s own hypersphere. The intuition for determining semantic similarity between $\Psi(\mathcal{H}_i)$ and $\Psi(\mathcal{H}_j)$ is that if images are jointly labelled as relevant in both \mathcal{R}_i and \mathcal{R}_j , it is likely that $\Psi(\mathcal{H}_i)$ and $\Psi(\mathcal{H}_j)$ have similar semantic content. Also, the larger the number of overlapping relevant images, the higher the semantic similarity between them can be expected. The number of

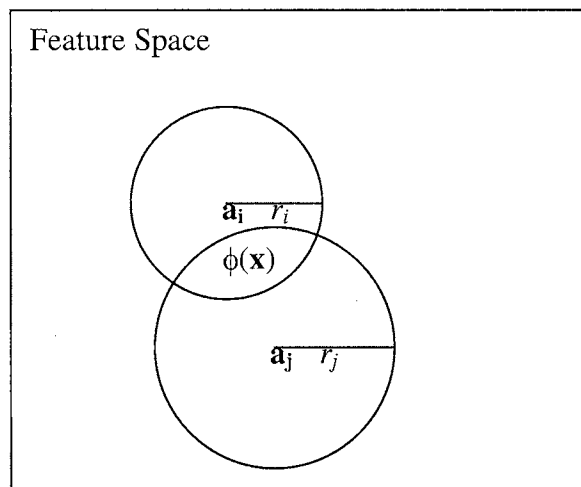


Figure 4.13: The (feature space) representation of a query \mathbf{x} may fall into more than one hypersphere. All the 1SVMs into which \mathbf{x} falls classify \mathbf{x} into their corresponding query concepts. The semantic similarity between \mathbf{x} 's concept and the concepts of those hyperspheres should be approximated in order to decide what previous knowledge to exploit.

overlapping images for which there is RF disagreement should also have an important negative effect on the semantic similarity. We now explain how the semantic similarity measure is derived.

The basic idea is based on the observation that semantic similarity between $\Psi(\mathcal{H}_i)$ and $\Psi(\mathcal{H}_j)$ should be based on similarity between their corresponding RF distributions (i.e., \mathcal{R}_i and \mathcal{R}_j). Let X be a random variable with sample space

$$\mathcal{S} = \{(\mathbf{x}_i, y_i) \mid (\mathbf{x}_i, y_i) \in \mathcal{R}\}$$

(i.e., an event is the labelling of an image as relevant or non-relevant). Let $P((\mathbf{x}_i, y_i) \mid \mathcal{R})$ be the probability that a user assigns label y_i to \mathbf{x}_i when searching for images belonging to $\Psi(\mathcal{H})$. Thus, $\forall (\mathbf{x}_i, y_i) \in \mathcal{S}, P((\mathbf{x}_i, y_i) \mid \mathcal{R}) = 1$. Let's assume that $\Psi(\mathcal{H}_i) = \Psi(\mathcal{H}_j)$. Then, let X_{ij} be a random variable with sample space

$$\mathcal{S}_{ij} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in (\mathcal{R}_i^+ \cap \mathcal{R}_j^+) \cup (\mathcal{R}_i^+ \cap \mathcal{R}_j^-) \cup (\mathcal{R}_i^- \cap \mathcal{R}_j^+) \cup (\mathcal{R}_i^- \cap \mathcal{R}_j^-)\}$$

(i.e., events involving images that appear in both \mathcal{R}_i and \mathcal{R}_j). Similarly, $P_{ij}((\mathbf{x}_j, y_j) \mid \mathcal{R}_i, \mathcal{R}_j)$ is the probability that a user assigns label y_j to \mathbf{x}_j when searching for images belonging to $\Psi(\mathcal{H}_i) = \Psi(\mathcal{H}_j)$. Thus, $P_{ij}((\mathbf{x}_j, y_j) \mid \mathcal{R}_i, \mathcal{R}_j) = 1$ if $\mathbf{x}_j \in \mathcal{R}_i^+ \cap \mathcal{R}_j^+$ or $\mathbf{x}_j \in \mathcal{R}_i^- \cap \mathcal{R}_j^-$. Otherwise, $P_{ij}((\mathbf{x}_j, y_j) \mid \mathcal{R}_i, \mathcal{R}_j) = 0.5$ if $\mathbf{x}_j \in \mathcal{R}_i^+ \cap \mathcal{R}_j^-$ or $\mathbf{x}_j \in \mathcal{R}_i^- \cap \mathcal{R}_j^+$. We can use the *entropy impurity* [26] of X_{ij} 's distribution to measure the distance between the distributions of X_i and X_j . The entropy impurity (or just *entropy*), $i(X)$, of random variable X with sample space \mathcal{S} is defined as

$$i(X) = - \sum_{x \in \mathcal{S}} P(x) \log_2 P(x)$$

where $P(x)$ is the probability of event x . Observe that $i(X_{ij}) = |\mathcal{R}_i^+ \cap \mathcal{R}_j^-| + |\mathcal{R}_i^- \cap \mathcal{R}_j^+|$ (i.e., number of mismatches). Notice that quantifying semantic distance in this way makes intuitive sense. As the number of mismatches increases, their corresponding event probabilities decrease, entropy (impurity) increases, and support for our initial assumption (i.e., that $\Psi(\mathcal{H}_i) = \Psi(\mathcal{H}_j)$) decreases.

Note that $0 \leq i(X_{ij}) \leq |\mathcal{S}_{ij}|$. The normalized distance function

$$dist(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j)) = \frac{i(X_{ij})}{|\mathcal{S}_{ij}|}$$

could be used as a measure of semantic distance between $\Psi(\mathcal{H}_i)$ and $\Psi(\mathcal{H}_j)$. For convenience, we convert to the normalized similarity measure

$$sim(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j)) = \frac{|\mathcal{S}_{ij}| - 2i(X_{ij})}{|\mathcal{S}_{ij}|}$$

Note that $-1 \leq sim(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j)) \leq 1$. The reason for rescaling to the range $[-1, 1]$ is that it allows semantic disagreement to have an effect on the voting scheme that we use for combining evidence. This does not affect the ranking based on semantic

similarity. Thus, the semantic similarity between $\Psi(\mathcal{H}_i)$ and $\Psi(\mathcal{H}_j)$ is defined as

$$\begin{aligned} \text{sim}(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j)) &= \frac{|\mathcal{S}_{ij}| - 2i(X_{ij})}{|\mathcal{S}_{ij}|} \\ &= \frac{|\mathcal{S}_{ij}| - i(X_{ij})}{|\mathcal{S}_{ij}|} - \frac{i(X_{ij})}{|\mathcal{S}_{ij}|} \\ &= \frac{|\mathcal{R}_i^+ \cap \mathcal{R}_j^+|}{|\mathcal{S}_{ij}|} - \frac{|\mathcal{R}_i^+ \cap \mathcal{R}_j^-| + |\mathcal{R}_i^- \cap \mathcal{R}_j^+|}{|\mathcal{S}_{ij}|} \end{aligned}$$

Notice that, intuitively, the first and second term in the formula are the maximum possible semantic agreement and disagreement respectively.

4.3.2.2 Query Modification and Distance Reweighting

Let $\mathcal{Z} = \{\mathcal{H}_j\}_1^n$ be the set of hyperspheres into which $\Phi(\mathbf{x})$ falls. In the following, we assume that $n > 0$ and go through the main stages of our proposed method. In the case that $n = 0$, inter-query learning is not exploited. At the beginning of the search session, the system does not have any knowledge about the semantics of \mathbf{x} (i.e., $\mathcal{R} = \emptyset$). Nevertheless, we can still identify the set of $\mathcal{H}_i \in \mathcal{Z}$ that are most likely to contain relevant images. The basic assumption is that if a majority of $\Psi(\mathcal{H}_i)$, $\mathcal{H}_i \in \mathcal{Z}$ are semantically similar, their concept has a higher density in that particular region of the feature space and thus there is more evidence that \mathbf{x} belongs to that concept. In other words, each $\mathcal{H}_i \in \mathcal{Z}$ classifies \mathbf{x} as belonging to $\Psi(\mathcal{H}_i)$. Therefore, the semantic similarity between every $(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j))$ pair determines the degree to which \mathcal{H}_i and \mathcal{H}_j are “voting” for the same concept. Thus, the set of $\mathcal{H}_i \in \mathcal{Z}$ whose $\Psi(\mathcal{H}_i)$ has highest semantic agreement are the most likely to contain relevant images.

The first stage of the algorithm sets $\mathbf{w} = \{1/d\}_1^d$ and computes an n by n “concept similarity” matrix \mathbf{Y} whose (i, j) -th entry is $\text{sim}(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j))$. Intuitively,

$$\mathbf{Y}_i = \sum_{j=1}^n \text{sim}(\Psi(\mathcal{H}_i), \Psi(\mathcal{H}_j))$$

is the degree by which $\Psi(\mathcal{H}_j), \forall \mathcal{H}_j \in \mathcal{Z}$ agree with (or are semantically similar to) $\Psi(\mathcal{H}_i)$. Then, \mathbf{x} and \mathbf{w} are updated as follows

$$\begin{aligned}\mathbf{x} &\leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{x}^i \right) + (1 - \alpha) \mathbf{x} \\ \mathbf{w} &\leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{w}^i \right) + (1 - \alpha) \mathbf{w}\end{aligned}$$

where

$$\begin{aligned}\gamma_i &= \max(0, \mathbf{Y}_i) / \sum_{j=1}^n \max(0, \mathbf{Y}_j) \\ \alpha &= \sum_{i=1}^n \max(0, \mathbf{Y}_i) / n^2\end{aligned}$$

where \mathbf{x}^i and \mathbf{w}^i are respectively the final query location and feature weights associated with hypersphere \mathcal{H}_i . Thus α adapts based on the density of homogeneous semantic concepts. For instance, if there is complete semantic agreement among $\Psi(\mathcal{H}_i), \forall \mathcal{H}_i \in \mathcal{Z}$, then $\alpha = 1$ and inter-query learning is completely exploited by setting

$$\begin{aligned}\mathbf{x} &\leftarrow \frac{1}{n} \sum_{i=1}^n \mathbf{x}^i \\ \mathbf{w} &\leftarrow \frac{1}{n} \sum_{i=1}^n \mathbf{w}^i\end{aligned}$$

On the other hand, when there is complete semantic disagreement, $\alpha = 0$ and inter-query learning is not used.

With each RF iteration, \mathcal{R} grows. In the second stage, the system uses this new information to revise its previous choices. Thus, after each RF iteration, the semantic similarity between $\Psi(\mathbf{x})$ and $\Psi(\mathcal{H}_i), \forall \mathcal{H}_i \in \mathcal{Z}$ is determined. Then, based

on this information, past inter-query learning choices are revised

$$\mathbf{x} \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{x}^i \right) + (1 - \alpha) \mathbf{x}^{\text{initial}}$$

$$\mathbf{w} \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{w}^i \right) + (1 - \alpha) \mathbf{w}^{\text{initial}}$$

where

$$\beta_i = \max(0, \text{sim}(\Psi(\mathcal{H}), \Psi(\mathcal{H}_i))) / \sum_{i=1}^n \max(0, \text{sim}(\Psi(\mathcal{H}), \Psi(\mathcal{H}_i)))$$

$$\alpha = \sum_{i=1}^n \max(0, \text{sim}(\Psi(\mathcal{H}), \Psi(\mathcal{H}_i))) / n$$

where $\mathbf{x}^{\text{initial}}$ and $\mathbf{w}^{\text{initial}}$ refer to the initial (i.e., before any RF iterations) values of \mathbf{x} and \mathbf{w} respectively. In the third stage, α decreases so that, as the number of RF iterations increases, we rely more on intra-query learning. Then, intra and inter-query learning are combined

$$\mathbf{x} \leftarrow \alpha \mathbf{x} + (1 - \alpha) \mathbf{x}^{\text{intra}}$$

$$\mathbf{w} \leftarrow \alpha \mathbf{w} + (1 - \alpha) \mathbf{w}^{\text{intra}}$$

where $\mathbf{x}^{\text{intra}}$ and $\mathbf{w}^{\text{intra}}$ are the modified query location and distance weights computed by the particular query modification and reweighing method (e.g., PFRL+ μ_r), based on intra-query learning \mathcal{R} . Thus, in this case, α determines the ratio of intra to inter-query learning to be used in processing the query. It adapts based on the density of homogeneous semantic concepts and the number of RF iterations. The second and third stages are repeated after each RF iteration.

This approach can be implemented using PFRL+ μ_r as the method for the intra-

query distance reweighing and query modification. This implementation of our approach (PFRL+ μ_r +1SVM) is summarized in Figure 4.14. In the Figure, \mathbf{w}^{pfrl} refers to the feature weights as computed by PFRL.

1. Initialize $\mathbf{w} \leftarrow \{1/d\}^d$, $\mathcal{R} \leftarrow \emptyset$, $\alpha \leftarrow 1$
2. Form $\mathcal{Z} \leftarrow \{\mathcal{H}_i\}_1^n$
3. If $|\mathcal{Z}| = 0$ go to 5
4. Exploit Inter-Query Learning
 - 4.1. Compute $\{\gamma_i\}_1^n$, α
 - 4.2. $\mathbf{x} \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{x}^i \right) + (1 - \alpha) \mathbf{x}$
 - 4.3. $\mathbf{w} \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{w}^i \right) + (1 - \alpha) \mathbf{w}$
5. Compute k nearest images to \mathbf{x} using \mathbf{w}
6. User marks the k images
7. While More RF Iterations Do
 - 7.1. $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\mathbf{x}_i, y_i)\}_1^k$
 - 7.2. If $|\mathcal{Z}| = 0$ go to 7.4
 - 7.3. Revise Inter-Query Learning
 - 7.3.1. Compute $\{\beta_i\}_1^n$, α
 - 7.3.2. $\mathbf{x} \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{x}^i \right) + (1 - \alpha) \mathbf{x}^{\text{initial}}$
 - 7.3.3. $\mathbf{w} \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{w}^i \right) + (1 - \alpha) \mathbf{w}^{\text{initial}}$
 - 7.4. Compute \mathbf{w}^{pfrl} , μ_r ; decrease α
 - 7.5. $\mathbf{x} \leftarrow \alpha \mathbf{x} + (1 - \alpha) \mu_r$
 - 7.6. $\mathbf{w} \leftarrow \alpha \mathbf{w} + (1 - \alpha) \mathbf{w}^{\text{pfrl}}$
 - 7.7. Compute k nearest images to \mathbf{x} using \mathbf{w}
 - 7.8. User marks the k images
8. Use \mathcal{R}^+ as training data for a 1SVM
9. Save $\mathcal{H} = \{\mathbf{x}, \mathbf{w}, \mathcal{R}, \mathbf{a}, r\}$

Figure 4.14: PFRL with Query Shifting and Inter-Query Learning (PFRL+ μ_r +1SVM)

4.3.3 Experimental Results

In this section we present experimental results obtained with the approaches described in Sections 4.3.1 and 4.3.2. The retrieval performance is measured by precision (1.1) and recall (1.2). The following data sets were used for evaluation:

1. *Texture* - the texture data set, obtained from MIT Media Lab [108]. There are 40 different texture images that are manually classified into 15 classes. Each of those images is then cut into 16 non-overlapping images of size 128x128. Thus, there are 640 images in the database. The images are represented by 16-dimensional feature vectors. We use 16 Gabor filters (2 scales and 4 orientations). Sample images are shown in Figure 4.15.
2. *Letter* - the letter data set, obtained from the UCI repository of machine learning databases [93]. It consists of 20,000 character images, each represented by a 16-dimensional feature vector. There are 26 classes of the 2 capital letters “O” and “Q”. The images are based on 20 different fonts with randomly distorted letters. Sample images are shown in Figure 4.16.

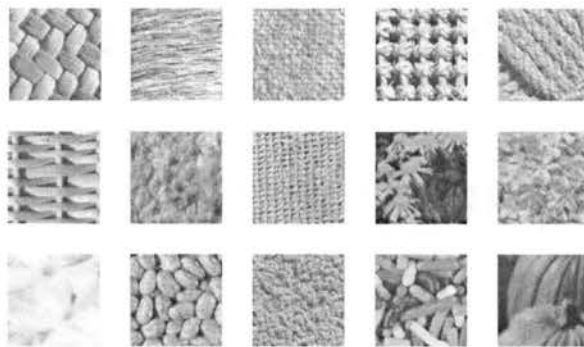


Figure 4.15: Sample images from *Texture* data set.

Because the images in the data sets are labelled according to their category, it is known whether an image in a retrieval set would be labelled as relevant or non-relevant by a user. To determine the free parameters, a ten-fold cross-validation

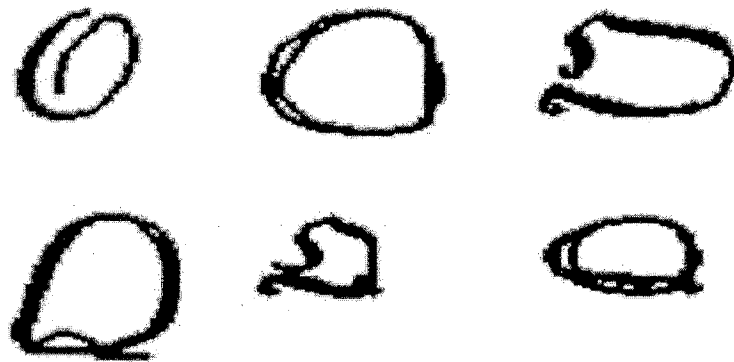


Figure 4.16: Sample images from *Letter* data set. First row contains images of the letter “O”; images on the second row are of the letter “Q”.

was performed for the *Texture* and *Letter* data sets. Each data set was divided into ten partitions. Each partition in turn was left out and the other nine were used to determine values for the free parameters. The left out partition was then used to test the algorithms. The values reported are the average of the ten tests. We make no claim on using optimal values for *Letter* as the parameters were selected after a very coarse sampling.

In order to compare the performance of our first method, we have implemented the virtual feature (VF) approach [155], and the statistical correlation technique (SC) [75] (both described in Section 4.2). Those approaches and our first method exploit inter-query learning. Their response with respect to different amounts of experience (data level) is investigated. The data level is the amount of accumulated inter-query learning (i.e., number of queries processed) relative to the number of images in the data set. We also compare the performance of our first method against that of traditional intra-query-learning-only RF approaches. For that purpose, we have also implemented the probabilistic feature relevance learning (PFRL) [103] method (described in Section 4.1).

Figures 4.17 and 4.18 show the precision in the initial retrieval set (i.e., with no iterations of RF) with respect to different data levels. In the Figures, 1SVM refers

to our first method. In order to create the initial retrieval set, a traditional intra-query-learning-only RF approach performs a k nearest neighbor (knn) search. Both VF and PFRL require at least one iteration of RF. Thus, for the initial retrieval set, they have the same performance as knn. As we can observe from those Figures, precision in the initial retrieval set can be drastically improved by integrating inter-query learning. Also, precision keeps improving as the data level increases. This results in a reduction on the number of RF iterations that are needed to satisfy a query. Thus, from the user’s point of view, it is very beneficial since users cannot stand too many RF iterations. On the other hand, if we use solely a knn search, there is no gain on the initial retrieval precision along the number of processed queries.

From those Figures, we can also observe that, with low data levels, there may be an initial decrease in precision. This is due to the fact that the retrieval set is formed based on a fixed ratio of intra to inter-query learning. Both VF and SC use a similar concept, the “maximal distance adjustment” and the “semantic weight” respectively, which is also based on a fixed weighting of inter-query learning. Intuitively, initially we would like to rely heavily on current intra-query learning since, at the beginning, there is not much historical information. Similarly, we would like to increase the exploitation of inter-query learning as more queries are processed and experience accumulates. Thus, we could adaptively change the ratio of intra to inter-query learning so that at the beginning, when there is little historical information, w_{intra} is large and, as experience accumulates, it becomes increasingly smaller (i.e., we rely more on inter-query learning). We plan to investigate the possibility of using a machine learning approach such as artificial neural networks or reinforcement learning to have a principled way of exploiting intra and inter-query learning that adapts to the current situation. In our approach, the optimal ratio of intra to inter-query learning was defined as the one resulting in highest precision with large data levels and was determined to be 0.25:0.75. Note that choosing 1:0 as the ratio of intra to inter-query

learning is the same as using an intra-query-learning-only 1SVM learning approach. Thus, our method outperforms 1SVM approaches that do not exploit inter-query learning.

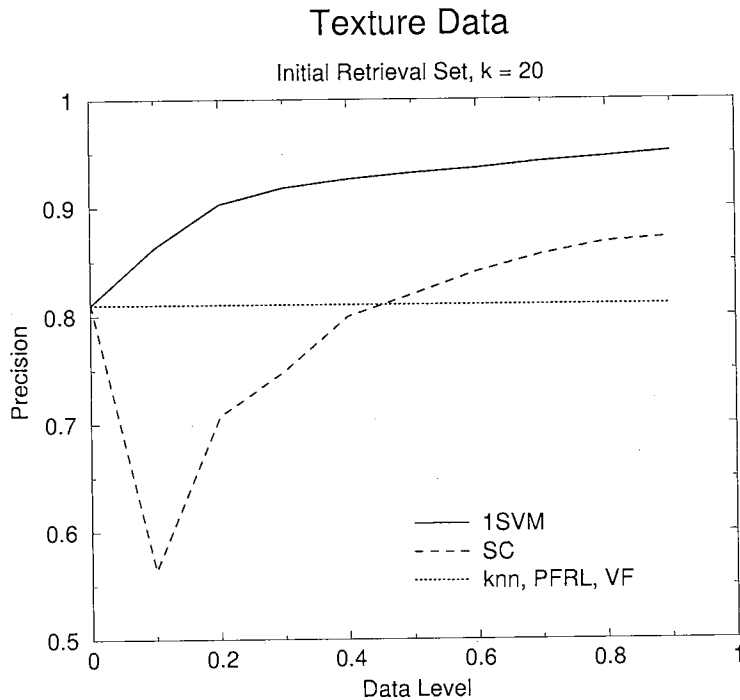


Figure 4.17: Retrieval performance in initial retrieval set with first approach and on other methods on *Texture* data.

Figures 4.19 and 4.20 show the precision after one iteration of RF with respect to different data levels. As we can observe from those Figures, precision increases after one RF iteration. The amount of improvement obtained when going from one to two RF iterations is much smaller. This is a desirable property since users do not want to perform many RF iterations. We can also observe that, with at least one RF iteration, 1SVM and VF have similar performance. On the other hand, our approach can provide improvement in the initial retrieval set. It can also be seen that, as the data level increases, both methods result in a very significant performance improvement over PFRL. On the other hand, with PFRL, there is no gain on the retrieval precision along the number of processed queries. As a result, the precision stays at a fixed value. This demonstrates that methods which exploit both short and

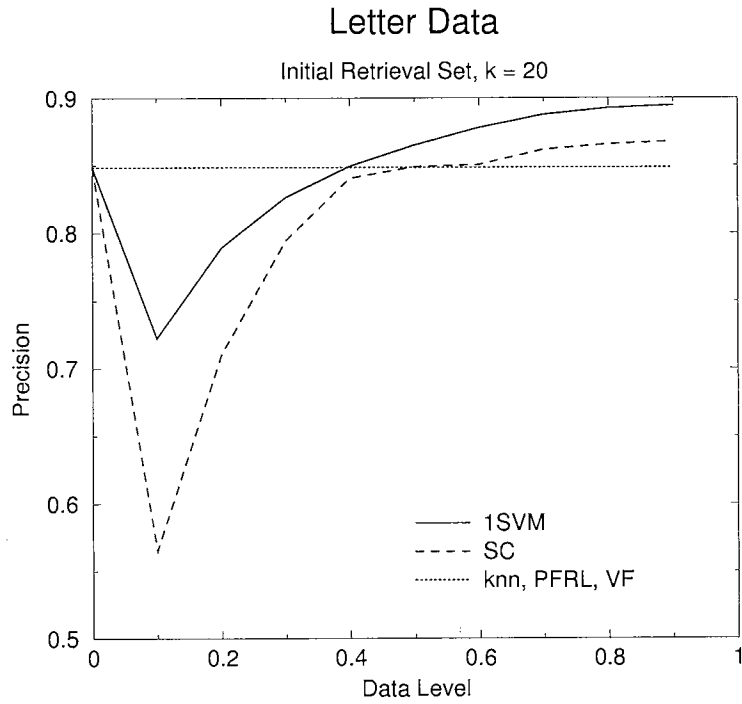


Figure 4.18: Retrieval performance in initial retrieval set with first approach and other methods on *Letter* data.

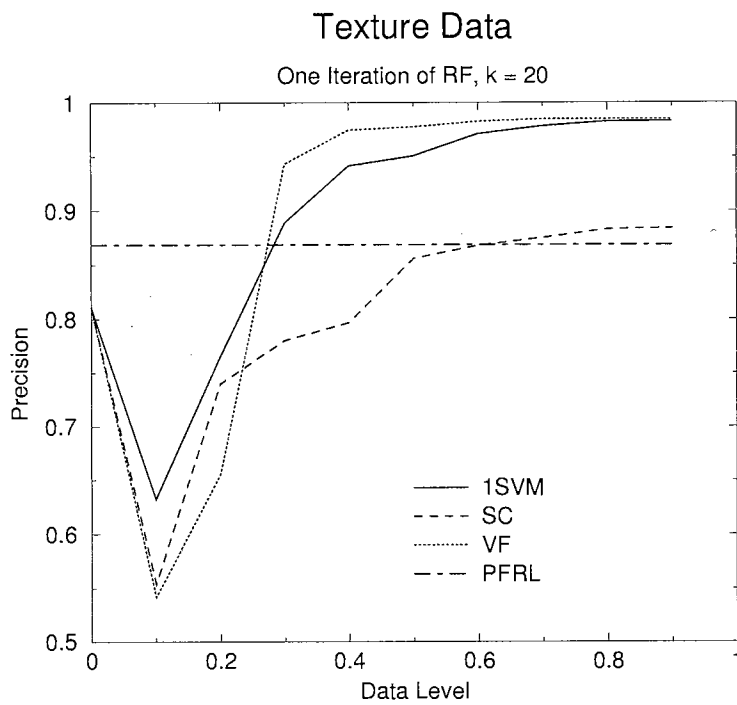


Figure 4.19: Retrieval performance after one RF iteration with first approach and other methods on *Texture* data.

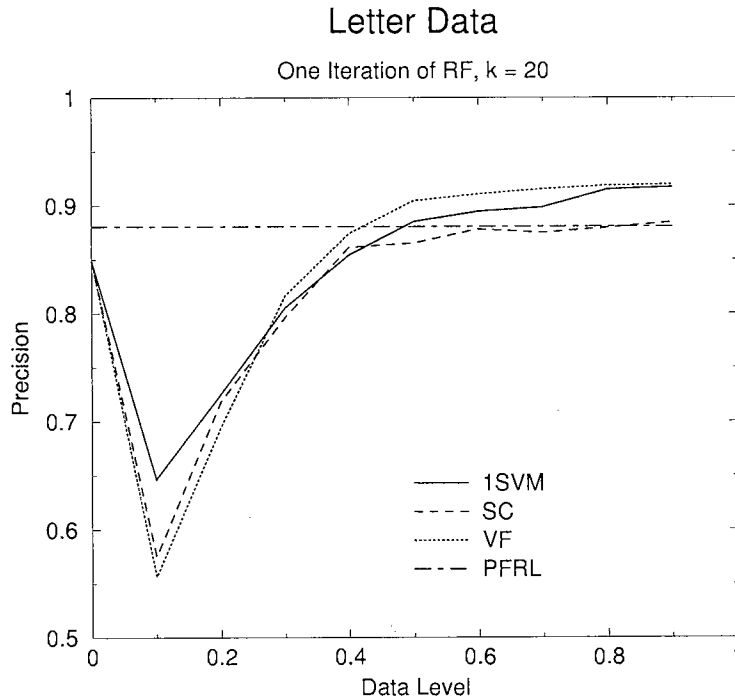


Figure 4.20: Retrieval performance after one RF iteration with first approach and other methods on *Letter* data.

long-term information perform better than intra-query-learning-only techniques.

Figure 4.21 shows the precision-recall graph of our approach for different data levels. Both high recall and high precision is desired, though not often obtainable. The values are the average over 64 random queries from *Texture*. From that Figure we can observe that increasing the data level has the desirable effect of pulling the precision-recall curve towards the upper right. As a last illustration, Figure 4.22 shows a particular retrieval result obtained by performing a nearest neighbor search in feature space on a random query from the *Texture* data set. A retrieval precision of 0.25 is achieved. This shows the inconsistency between content-based and semantic similarity. In contrast, Figure 4.23 shows the retrieval results obtained with our approach. In this case, a retrieval precision of 0.95 is achieved. This illustrates that exploiting inter-query learning can dramatically help to reduce the semantic gap and improve retrieval performance.

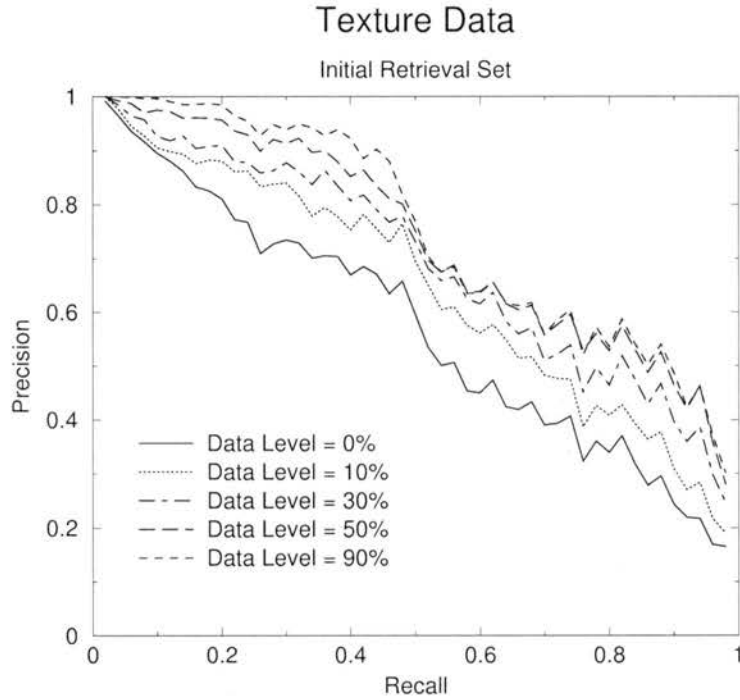


Figure 4.21: Retrieval performance at different data levels with first approach and other methods on *Texture* data.

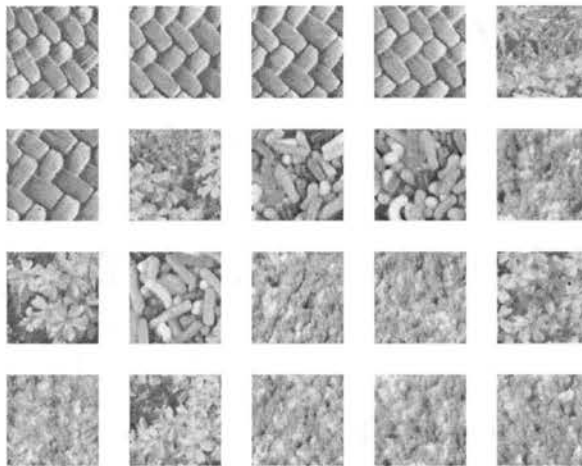


Figure 4.22: Retrieval results after performing a nearest neighbor search in feature space on a random query from the *Texture* data set. The top leftmost image represents the query image. The images are sorted based on their similarity to the query. The ranks descend from left to right and from top to bottom. Retrieval precision is 0.25.

Next, we compare the performance of our original approach (i.e., with no merging of 1SVMs) against that of the modified approach (i.e., with merging of 1SVMs), which summarizes inter-query learning. The goal is to determine whether high retrieval per-

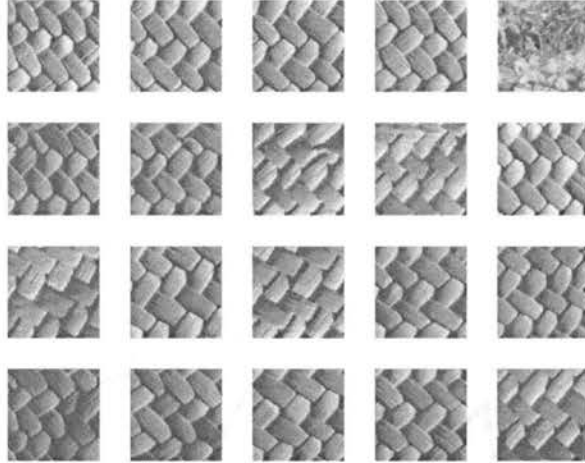


Figure 4.23: Retrieval results with our method on a random query from the *Texture* database. The top leftmost image represents the query image. The images are sorted based on their similarity to the query. The ranks descend from left to right and from top to bottom. Retrieval precision is 0.95.

formance can still be obtained when summarizing inter-query learning. Figures 4.24 and 4.25 show the precision of the initial retrieval set with respect to different data levels. These Figures also show the performance obtained by running the modified approach without using pre-images to approximate cluster representatives's centers (i.e., by keeping their full expansions). Based on those Figures, we can observe that the performance loss that results from using pre-images to approximate cluster representative's centers is small. We also make the observation that with the proposed cluster-merging approach precision does not degrade with low data levels. It is higher on low data levels and slightly smaller with high levels of data.

In order to evaluate the performance of our second approach, we have implemented it using PFRL combined with query shifting (PFRL+ μ_r) (described in Section 4.3.2). This implementation of our approach (PFRL+ μ_r +1SVM) is summarized in Figure 4.14. In PFRL and PFRL+ μ_r , all information collected during a search session is lost at the end of the session. We compare the retrieval performance of PFRL, PFRL+ μ_r , and PFRL+ μ_r +1SVM. Figures 4.26 and 4.28 show precision in the initial retrieval set with respect to different data levels. An intra-query-learning-only RF approach

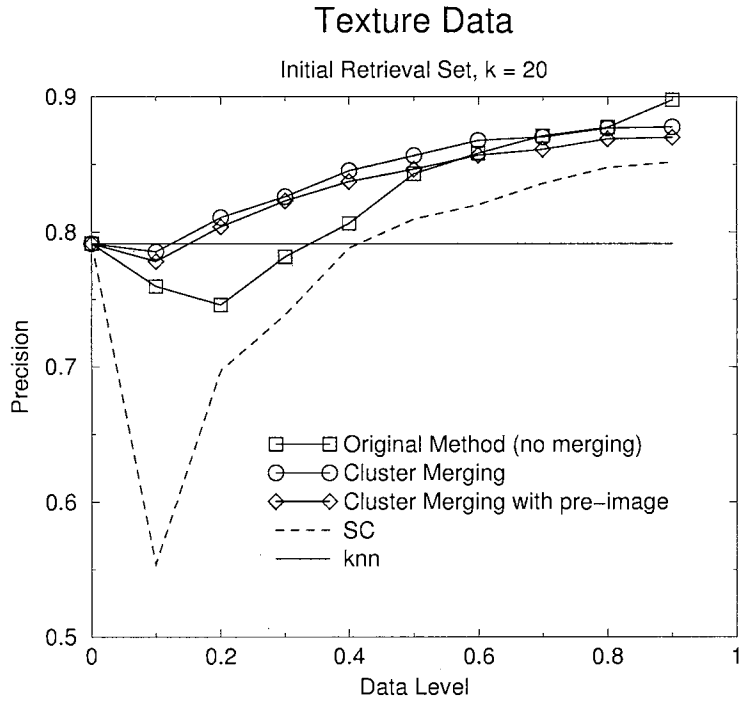


Figure 4.24: Retrieval performance in initial retrieval set with modified first approach and other methods on *Texture* data.

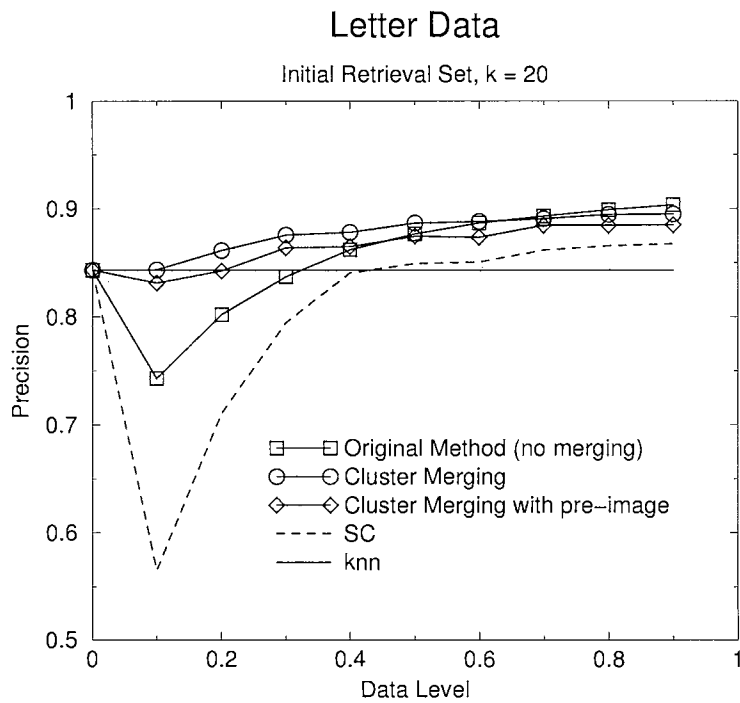


Figure 4.25: Retrieval performance in initial retrieval set with modified first approach and other methods on *Letter* data.

forms the initial retrieval set by doing a knn search. The VF approach requires at least one RF iteration. Thus, on initial retrieval, VF, PFRL and PFRL+ μ_r have the same performance as a knn search. Again, as we can observe from those Figures, precision in the initial retrieval set can be drastically improved by exploiting inter-query learning and keeps improving as the data level increases. This results in a reduction on the number of RF iterations that are needed to satisfy a query. Thus, from the user's perspective, it is very beneficial since users cannot stand too many RF iterations.

Figures 4.27 and 4.29 show precision after one RF iteration with respect to different data levels. As we can observe, precision increases after one RF iteration. The amount of improvement obtained when going from one to two RF iterations is much smaller. This is a desired property since users do not want to perform many RF iterations. We can observe that, with low data levels, there is an initial decrease in precision in both VF and SC. This is due to the fact that those methods use a fixed ratio of intra to inter-query learning to form the retrieval set. Our second approach is based on an adaptive weighting of inter-query learning and thus, does not suffer from this problem.

We can learn from these results that the image retrieval performance is constantly improved by the integration of inter-query learning. Furthermore, performance can be improved in the initial retrieval set where a traditional intra-query-learning-only approach would require at least one iteration of RF to provide some improvement. Thus, user interaction can be reduced by reducing the number of iterations that are needed to satisfy a query.

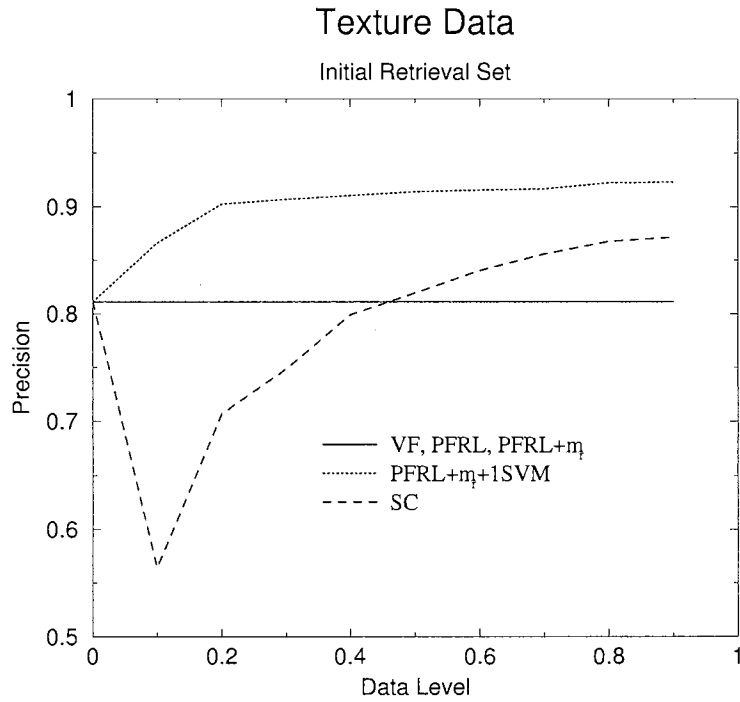


Figure 4.26: Retrieval performance in initial retrieval set with $\text{PFRL} + \mu_r + 1\text{SVM}$ and other methods on *Texture* data.

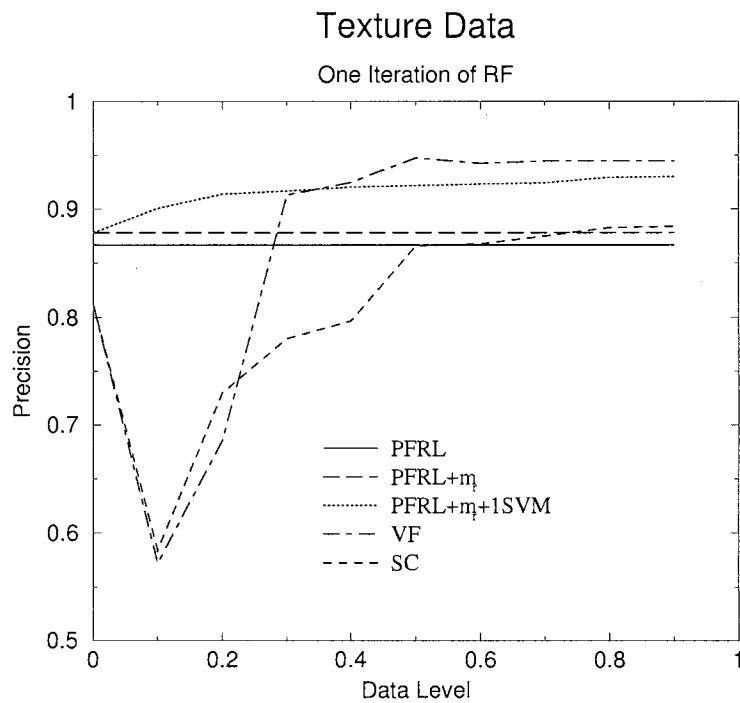


Figure 4.27: Retrieval performance after one RF iteration with $\text{PFRL} + \mu_r + 1\text{SVM}$ and other methods on *Texture* data.

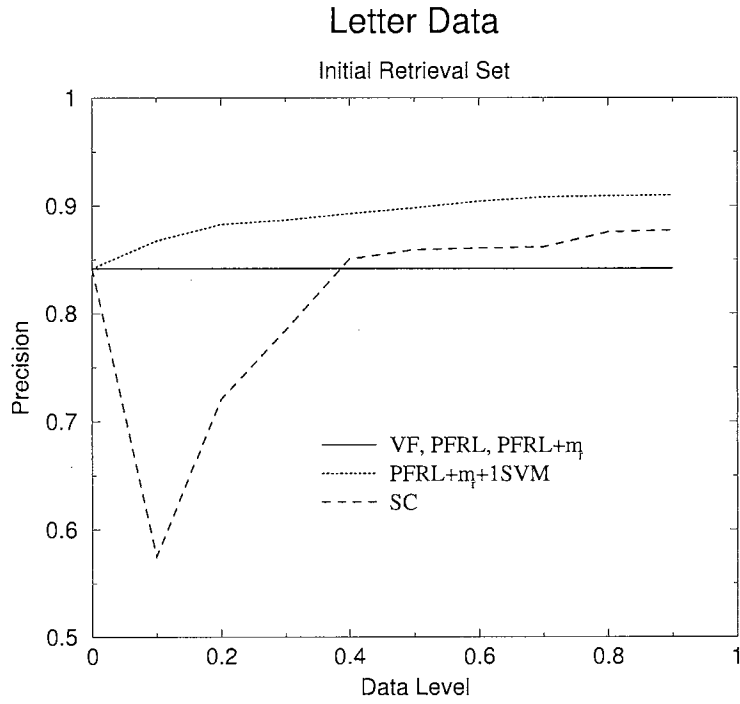


Figure 4.28: Retrieval performance in initial retrieval set with $\text{PFRL} + \mu_r + 1\text{SVM}$ and other methods on *Letter* data.

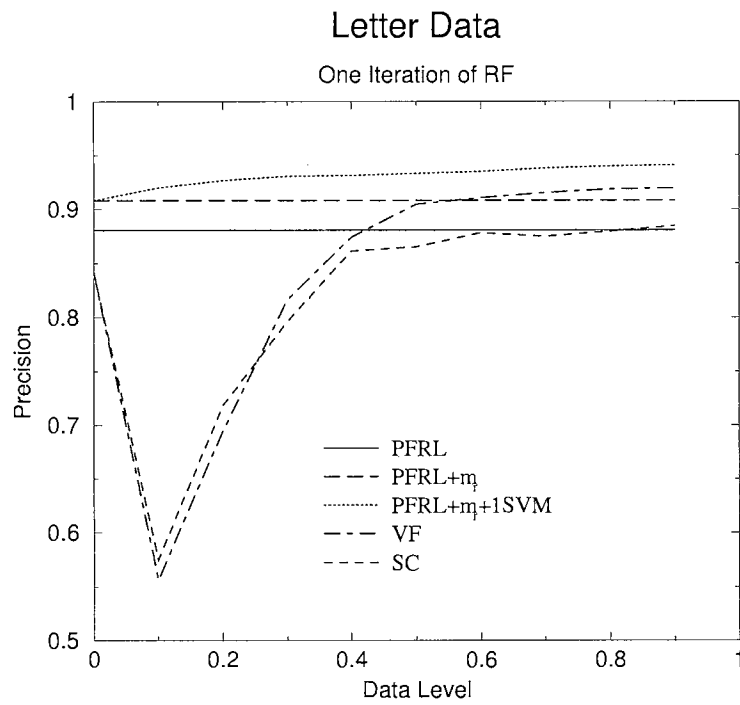


Figure 4.29: Retrieval performance after one RF iteration with $\text{PFRL} + \mu_r + 1\text{SVM}$ and other methods on *Letter* data.

Chapter 5

Learning with Region-Based Image Representations

In this chapter, we first summarize related work on intra-query learning with region-based image representations. Next, we present two novel intra-query learning approaches for content-based image retrieval (CBIR) with region-based image representations. The first approach, probabilistic region relevance learning (PRRL) [38], is based on the observation that regions in an image have unequal importance for computing image similarity. It automatically estimates region relevance based on user's feedback. It can be used to set region weights in region-based image retrieval frameworks that use an overall image-to-image similarity measure.

The second approach, presented in [37], is based on support vector machine (SVM) learning. Traditional approaches based on SVM learning require the use of fixed-length image representations because SVM kernels represent an inner product in a feature space that is a non-linear transformation of the input space. However, many CBIR methods that use region-based image representations create a variable-length image representation and define a similarity measure between two variable-length representations. Thus, the standard SVM approach cannot be applied because it violates the requirements that a SVM places on the kernel. Fortunately, a generalized

support vector machine (GSVM) [84] (described in Section 3.1.5) has been developed that allows the use of an arbitrary kernel. We present a learning algorithm based on GSVMs. Since a GSVM does not place restrictions on the kernel, any image similarity measure can be used.

Next, we present an intra/inter-query learning approach that addresses the problem of semantically-meaningful image segmentation. A large number of image segmentation techniques have been proposed in the literature. However, most image segmentation algorithms create regions that are homogeneous with respect to one or more low-level features according to some similarity measure. Unfortunately, homogeneous regions based on low-level features usually do not correspond to meaningful objects. We propose an algorithm based on multiple-instance learning (MIL) [25, 85, 87] (described in Section 3.2.1) that exploits both intra and inter-query learning for automatically improving the segmentation of images in a database. The main advantage of this approach is that it can automatically refine the segmentation of images into semantically-meaningful objects.

5.1 Related Work in Intra-Query Learning

Although relevance feedback (RF) learning has been successfully applied to CBIR systems that use global image representations, not much research has been conducted on RF learning methods for region-based CBIR.

By referring to an image as a bag and a region in the image as an instance, MIL has been applied to image classification and retrieval [2, 86, 154, 158]. The diverse density (DD) technique [87] (described in Section 3.2.1) is applied in [86, 154, 158]. Basically, an objective function is used that looks for a feature vector that is close to many instances from different positive bags and far from all instances from negative bags. Such a vector is likely to represent the concept (i.e., object in the image) that matches the concept the user has in mind.

In [86], MIL was applied to the task of learning to recognize a person from a set of images that are labelled positive if they contain the person and negative otherwise. They also used this model to learn descriptions of natural images (such as a sunsets or mountains) and then used the learned concept to retrieve similar images from an image database. Their system uses the set of cumulative user-labelled relevant and non-relevant images to learn a scene concept which is used to retrieve similar images. This is done by using the DD algorithm to find out what regions are in common between the relevant images and the differences between those and the non-relevant images. The confidence that an image is relevant to the user's query concept can be measured by the distance from the ideal point (as computed by the DD algorithm) to the closest region in the image. However, not all region features are equally important. Thus, in this approach, the distance measure is not restricted to a normal Euclidean distance, but may be defined as a weighted Euclidean distance (such as (2.1)) where important features have larger weights. The DD algorithm is also capable of determining these weights. However, by introducing weights, the number of dimensions over which DD has to be maximized is doubled.

This method is improved in [154] by allowing a broader range of images. In [154], the image similarity measure is defined as the correlation coefficient of corresponding regions. This similarity measure is further refined by allowing different weights for different locations. In [158], a comparison of performance obtained with the DD and EM-DD [157] (described in Section 3.2.1) algorithms when using a wide variety of image processing techniques and a broader range of images is presented.

Based on the assumption that important regions should appear more often in relevant images than unimportant regions, a $RF * IIF$ (region frequency * inverse image frequency) weighting scheme is proposed in [65]. Let $\mathcal{D} = \{\mathbf{x}_i\}_1^m$ be the set of all images in the database, \mathbf{x} be the query image, $\{\mathcal{R}_i\}_1^n$ be the set of all regions in \mathbf{x} , and \mathcal{R}^+ be the set of cumulative relevant retrieved images for \mathbf{x} . The region

frequency (RF) of a region \mathcal{R}_i is defined as

$$RF(\mathcal{R}_i) = \sum_{\mathbf{x}_j \in \mathcal{R}^+} s(\mathcal{R}_i, \mathbf{x}_j)$$

where $s(\mathcal{R}_i, \mathbf{x}_j) = 1$ if at least one region of \mathbf{x}_j is similar to \mathcal{R}_i and 0 otherwise. Two regions are deemed similar if their L1-distance (also known as the Manhattan distance or city-block distance) is smaller than a predefined threshold. The inverse frequency (IIF) of \mathcal{R}_i is defined as

$$IIF(\mathcal{R}_i) = \log \left(\frac{m}{\sum_{\mathbf{x}_j \in \mathcal{D}} s(\mathcal{R}_i, \mathbf{x}_j)} \right)$$

The region importance (RI) (i.e., weight) of \mathcal{R}_i is then

$$RI(\mathcal{R}_i) = \frac{RF(\mathcal{R}_i) * IIF(\mathcal{R}_i)}{\sum_{j=1}^n (RF(\mathcal{R}_j) * IIF(\mathcal{R}_j))}$$

Traditional RF schemes based on SVM learning have been applied to significantly improve retrieval performance in CBIR systems that use global image representations [18, 56, 156]. Those approaches require the use of fixed-length image representations because SVM kernels represent an inner product in a feature space that is a non-linear transformation of the input space. However, many CBIR methods that use region-based image representations create a variable-length image representation and define a similarity measure between two variable-length representations. Thus, the standard SVM approach cannot be applied because it violates the requirements that a SVM places on the kernel. To resolve the issue of common SVM kernels not allowing variable-length representations, a generalization of the Gaussian kernel is introduced in [65]

$$K_{GGaussian}(\mathbf{x}, \mathbf{y}) = e^{\frac{-d(\mathbf{x}, \mathbf{y})}{2\sigma^2}} \quad (5.1)$$

where $d(\mathbf{x}, \mathbf{y})$ is a distance measure in the input space between the two variable-length representations of images \mathbf{x} and \mathbf{y} . A particular form of (5.1) in which d is the earth mover’s distance (EMD) [116] is proposed in [65]. The EMD computes the distance between two distributions represented by sets of weighted features. It is the minimal cost of changing one distribution into the other. The cost is defined in terms of a user-defined ground distance that measures the distance between two features. A distribution can have any number of features. Thus, EMD can operate on variable-length representations of distributions. An image can be seen as a distribution with a variable number of regions. The kernel proposed in [65] is

$$K_{GEMD}(\mathbf{x}, \mathbf{y}) = e^{\frac{-EMD(\mathbf{x}, \mathbf{y})}{2\sigma^2}}$$

where $EMD(\mathbf{x}, \mathbf{y})$ is the EMD distance between the two variable-length representations of images \mathbf{x} and \mathbf{y} . In order for EMD to be a true metric, the ground distance must be a metric [116]. For example, in [65], the ground distance between two regions is set to the Euclidean distance. Therefore, this approach does not allow for arbitrary image similarity measures.

5.2 Probabilistic Region Relevance Learning

A key factor in region-based CBIR approaches that consider all the regions to perform an overall image-to-image similarity is the weighting of regions. The weight that is assigned to each region for determining similarity is usually based on prior assumptions such as that larger regions, or regions that are close to the center of the image, should have larger weights. For example, in integrated region matching (IRM) [75] (described in Section 2.3), an *area percentage scheme*, which is based on the assumption that important objects in an image tend to occupy larger areas, is used to assign weights to regions. The location of a region is also taken into consideration.

For example, higher weights are assigned to regions in the center of an image than to those around boundaries. These region weighting heuristics are often inconsistent with human perception. For instance, a facial region may be the most important when the user is looking for images of people while other larger regions such as the background may be much less relevant.

Based on the observation that regions in an image have unequal importance for computing image similarity (See Figure 5.1), we propose a probabilistic method inspired by probabilistic feature relevance learning (PFRL) [103] (described in Section 4.1), probabilistic region relevance learning (PRRL), for automatically capturing region relevance based on user’s feedback. PRRL can be used to set region weights in region-based image retrieval frameworks that use an overall image-to-image similarity measure.

5.2.1 Region Relevance Measure

Inspired by PFRL, we learn the differential region relevance by estimating the strength of each region in predicting the class of a given query. Given a query image \mathbf{x} . Let \mathbf{x} be represented by a set of regions $\{\mathcal{R}_i\}_1^n$, where $\mathcal{R}_i = \{\mathbf{r}_i\}$ is the descriptor of the i -th region and $\mathbf{r}_i \in \mathbb{R}^d$ is a feature vector extracted from the i -th region. Let the class label $y \in \{1, 0\}$ at \mathbf{x} (i.e., relevant or non-relevant) be treated as a random variable from a distribution with the probabilities $\{Pr(1 | \mathbf{x}), Pr(0 | \mathbf{x})\}$. Consider the function f of n arguments

$$f(\mathbf{x}) \doteq Pr(1 | \mathbf{x}) = Pr(y = 1 | \mathbf{x}) = E(y | \mathbf{x})$$

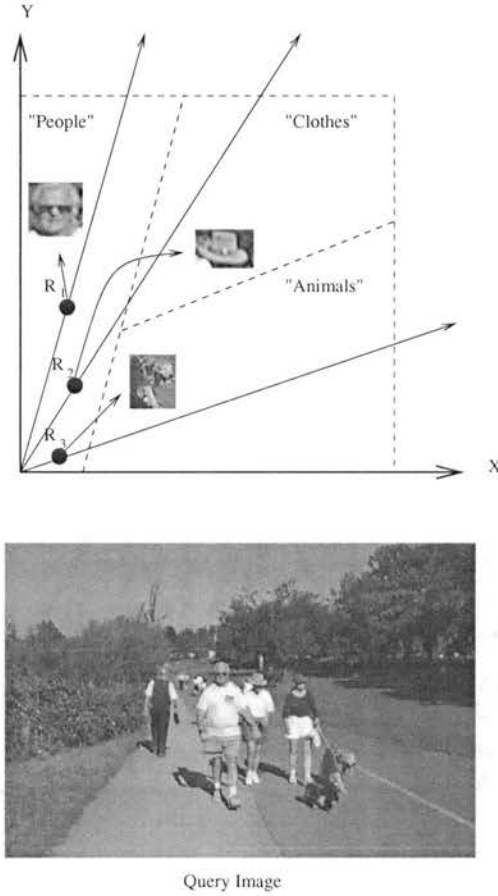


Figure 5.1: Regions are unequal in their differential relevance for computing similarity. Given that the user is looking for images of people, region \mathcal{R}_1 may be the most important, perhaps followed by \mathcal{R}_2 and \mathcal{R}_3 . Thus, the neighborhood of the similarity metric should be elongated along the direction of \mathcal{R}_1 and constricted along the direction of \mathcal{R}_3 .

In the absence of any argument assignments, the least-squares estimate for $f(\mathbf{x})$ is simply its expected (average) value

$$E[f] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

where $p(\mathbf{x})$ is the joint probability density. Now, suppose that we know the value of \mathbf{x} at a particular \mathbf{r}_i . The least-squares estimate becomes

$$E[f | \mathbf{r}_i] = \int f(\mathbf{x})p(\mathbf{x} | \mathbf{r}_i)d\mathbf{x}$$

where $p(\mathbf{x} \mid \mathbf{r}_i)$ is the conditional density of the other regions. Because $f(\mathbf{x}) = 1$ (i.e., the query image is always relevant), $(f(\mathbf{x}) - 0)$ is the maximum error that can be made when assigning 0 to the probability that \mathbf{x} is relevant when the probability is in fact 1. On the other hand, $(f(\mathbf{x}) - E[f \mid \mathbf{r}_i])$ is the error that is made by predicting $E[f \mid \mathbf{r}_i]$ to be the probability that \mathbf{x} is relevant. Therefore,

$$[(f(\mathbf{x}) - 0) - (f(\mathbf{x}) - E[f \mid \mathbf{r}_i])] = E[f \mid \mathbf{r}_i]$$

represents a reduction in error between the two predictions. Therefore, a measure of the relevance of the i -th region for \mathbf{x} can be defined as

$$r_i(\mathbf{x}) = E[f \mid \mathbf{r}_i] \tag{5.2}$$

Thus $r_i(\mathbf{x}) = 0$ when $f(\mathbf{x})$ is independent of \mathbf{r}_i (at \mathbf{x}) and $r_i(\mathbf{x}) = 1$ when $f(\mathbf{x})$ depends only on \mathbf{r}_i (at \mathbf{x}). Values in between these extremes indicate varying degrees of relevance for \mathbf{r}_i . Also, it can be viewed as a measure of local relevance in the sense that its value depends on the particular \mathbf{x} at which it is evaluated [31]. We can then use a weighted similarity measure where the weight of the i -th region is given by

$$w_i = \frac{e^{ar_i(\mathbf{x})}}{\sum_{j=1}^n e^{ar_j(\mathbf{x})}} \tag{5.3}$$

where a is a parameter that can be chosen to maximize (minimize) the influence of r_i on w_i [103].

5.2.2 Estimation of Region Relevance

Similarly to PFRL for estimating feature relevance, we use the retrieved images with RF to estimate region relevance. Let $\mathcal{R} = \{(\mathbf{x}_j, y_j)\}_1^m$ be the set of cumulative retrievals for \mathbf{x} , where \mathbf{x}_j is the j -th retrieved image and $y_j \in \{1, 0\}$ is its class label

(i.e., relevant or non-relevant). Let \mathbf{x}_j be represented by a set of regions $\{\mathcal{R}'_j\}_1^z$ where $\mathcal{R}'_j = \{\mathbf{r}'_j\}$ is the descriptor of the j -th region and $\mathbf{r}'_j \in \mathfrak{R}^d$ is a feature vector extracted from the j -th region. Let $0 \leq s(\mathbf{r}_i, \mathbf{r}'_j) \leq 1$ denote the similarity between \mathbf{r}_i from \mathbf{x} and \mathbf{r}'_j from \mathbf{x}_j in a region-based CBIR system. Also, let

$$\hat{s}(\mathbf{r}_i, \mathbf{x}_j) = \max_{j \in \{1, 2, \dots, z\}} s(\mathbf{r}_i, \mathbf{r}'_j)$$

We can use \mathcal{R} to estimate (5.2), hence (5.3). Note that $E[f \mid \mathbf{r}_i] = E[y \mid \mathbf{r}_i]$. Thus, (5.2) can be estimated by

$$\hat{E}[y \mid \mathbf{r}_i] = \frac{\sum_{j=1}^m y_j 1(\hat{s}(\mathbf{r}_i, \mathbf{x}_j) == 1)}{\sum_{j=1}^m 1(\hat{s}(\mathbf{r}_i, \mathbf{x}_j) == 1)}$$

where $1(\cdot)$ returns 1 if its argument is true, and 0 otherwise. However, (5.2) cannot be directly estimated in this manner since there may be no (or at most a few) \mathbf{r}'_j such that $s(\mathbf{r}_i, \mathbf{r}'_j) = 1$ (i.e., no \mathbf{r}'_j such that $\mathbf{r}_i = \mathbf{r}'_j$). Therefore, instead, we follow an strategy suggested in [31] and look for data in the vicinity of \mathbf{r}_i (i.e., we allow $s(\mathbf{r}_i, \mathbf{r}'_j) < 1$). Thus, (5.2) is estimated by

$$\hat{E}[y \mid \mathbf{r}_i] = \frac{\sum_{j=1}^m y_j 1(\hat{s}(\mathbf{r}_i, \mathbf{x}_j) > \varepsilon)}{\sum_{j=1}^m 1(\hat{s}(\mathbf{r}_i, \mathbf{x}_j) > \varepsilon)} \quad (5.4)$$

where $0 \leq \varepsilon \leq 1$ is an adaptive similarity threshold that changes so that there is sufficient data for the estimation of (5.2). The value of ε is chosen so that

$$\sum_{j=1}^m 1(\hat{s}(\mathbf{r}_i, \mathbf{x}_j) > \varepsilon) = g$$

where $g \leq m$. The PRRL algorithm is summarized in Figure 5.2.

- | |
|--|
| <ol style="list-style-type: none"> 1. Initialize region weights, $\mathcal{R} \leftarrow \emptyset$ 2. Retrieve the k most similar images to query image \mathbf{x} 3. While More RF Iterations Do <ol style="list-style-type: none"> (a) User marks the k images as relevant or non-relevant (b) $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathbf{x}_j, y_j\}_1^k$ (c) Update weights of regions in \mathbf{x} with (5.4) and (5.3) using \mathcal{R} (d) Retrieve the k most similar images to \mathbf{x} |
|--|

Figure 5.2: The probabilistic region relevance learning (PRRL) algorithm.

5.2.3 Experimental Results

In this section we present experimental results obtained with PRRL. The retrieval performance is measured by precision (1.1) and recall (1.2). The following data set was used for evaluation:

1. *Corel* - A subset of 2000 labelled images from the general purpose COREL image database. There are 20 image categories, each containing 100 pictures. The region-based feature vectors of those images are obtained with the IRM/UFM segmentation algorithm described in Section 2.3¹. Sample images are shown in Figure 5.3.

We tested the performance of unified feature matching (UFM) [17] (described in Section 2.3), UFM with PRRL (UFM+PRRL), and UFM with the RF*IIF method [65] (described in Section 5.1) (UFM+RFIIF). Every image is used as a query image. A uniform weighting scheme is used to set the region weights of each query and target images. For UFM+PRRL, and UFM+RFIIF, user’s feedback was simulated by carrying out 3 RF iterations for each query. Because the images in the data set are labelled according to their category, it is known whether an image in a retrieval set would be labelled as relevant or non-relevant by a user.

¹We would like to thank Yixin Chen for providing us with this data

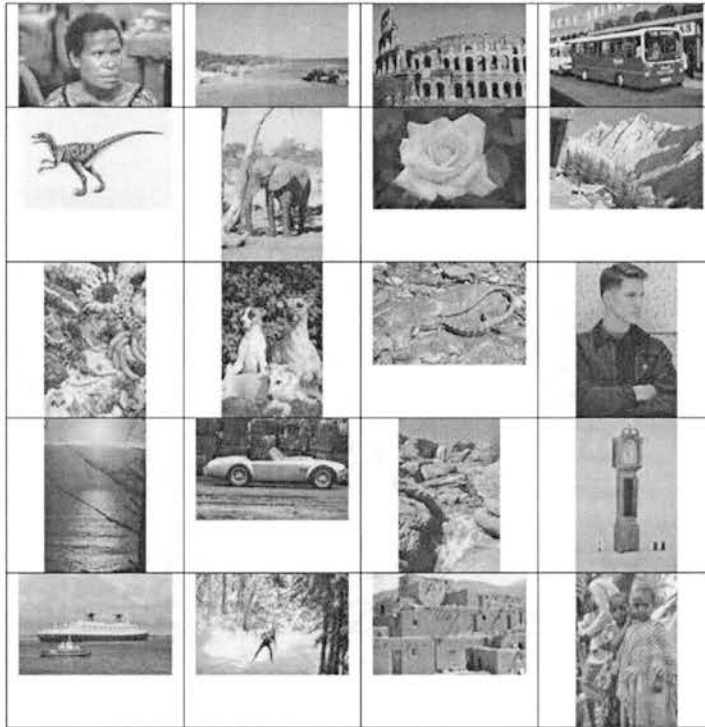


Figure 5.3: Sample images from *Corel* data set.

The average precision of the 2000 queries with respect to different number of RF iterations is shown in Figure 5.4. The size of the retrieval set is 20. Figures 5.5 through 5.8 show the precision recall curves after each RF iteration. We can observe that UFM+PRRL has the best performance. It can be seen that, even after only 1 RF iteration, the region weights learned by PRRL result in a very significant performance improvement.

Figure 5.9 shows the retrieval results obtained on a random query image. It is difficult to make objective comparisons with other region-based image retrieval systems such as Netra [81] or Blobworld [15] which require additional information from the user (i.e., important regions and/or features) during the retrieval process.

Currently, PRRL only performs intra-query learning. That is, for each given query, the user's feedback is used to learn the relevance of the regions in the query and the learning process starts from ground up for each new query. However, it is also possible

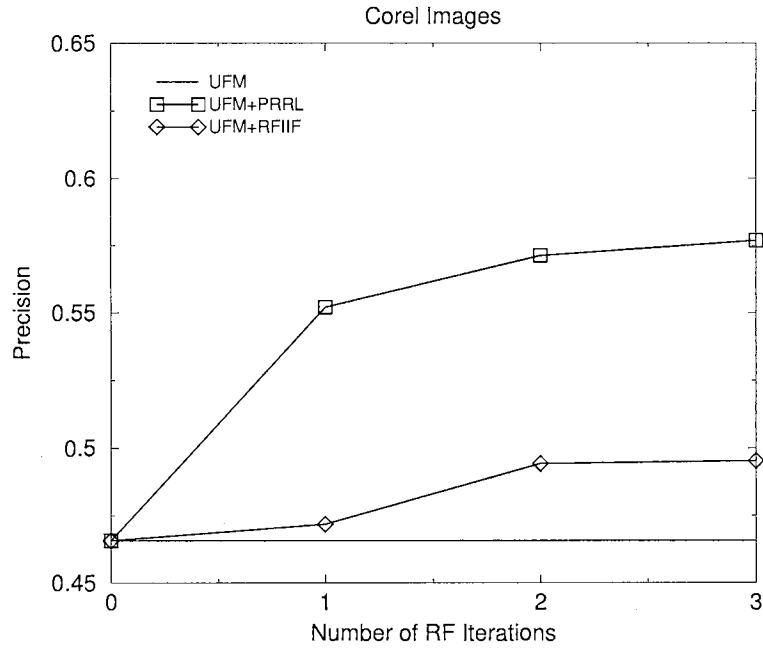


Figure 5.4: Retrieval performance at different number of RF iterations with PRRL and other methods on *Corel* data.

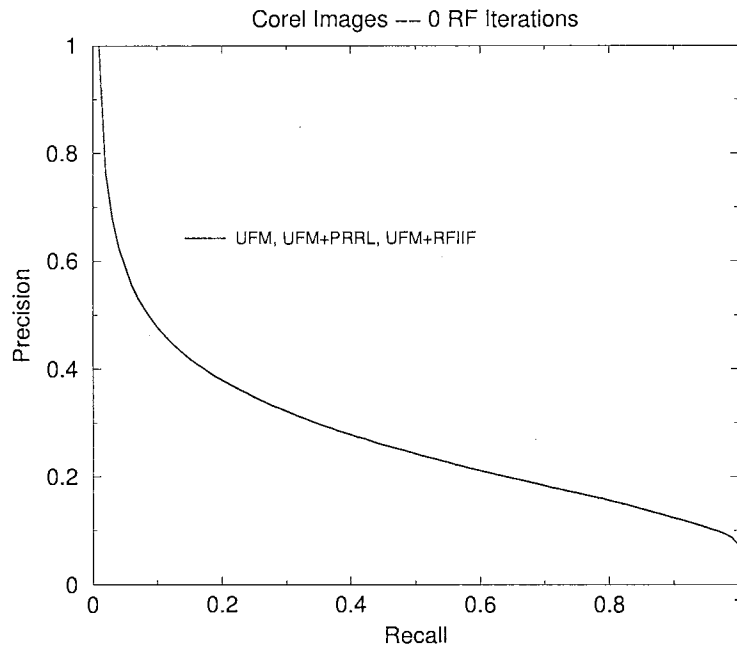


Figure 5.5: Retrieval performance in initial retrieval set with PRRL and other methods on *Corel* data.

to exploit inter-query learning to enhance the retrieval performance of future queries. Thus, for a new query, instead of starting the learning process from ground up, we

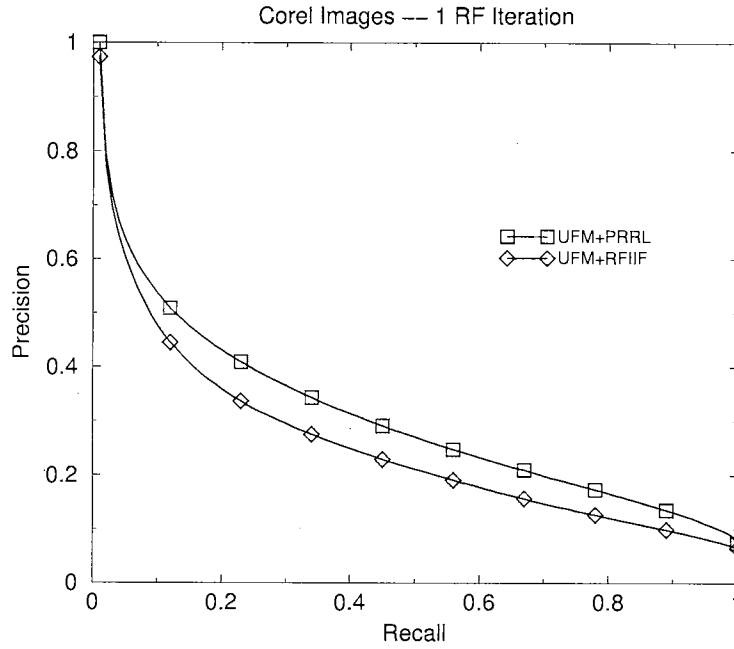


Figure 5.6: Retrieval performance after one RF iteration with PRRL and other methods on *Corel* data.

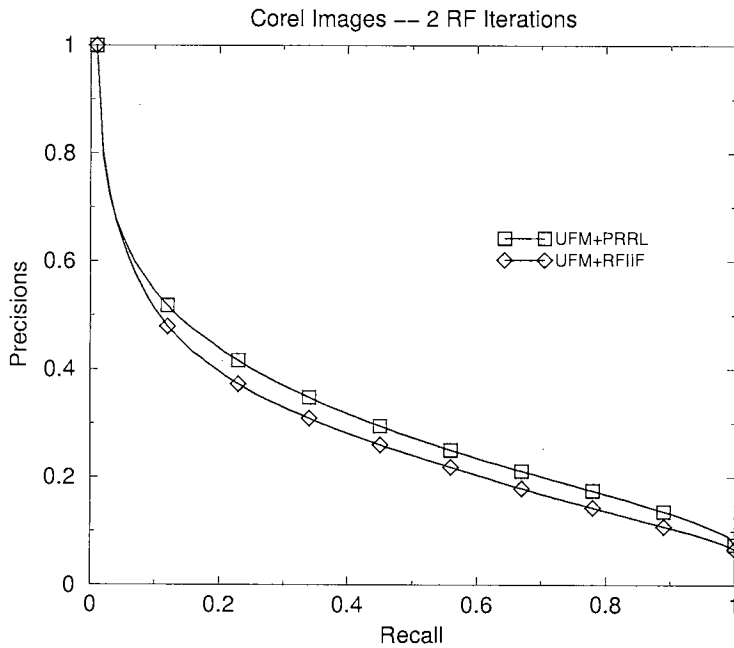


Figure 5.7: Retrieval performance after two RF iterations with PRRL and other methods on *Corel* data.

could exploit the previously learned region importances of similar queries. This would be very beneficial specially in the initial retrieval set since, instead of using uniform

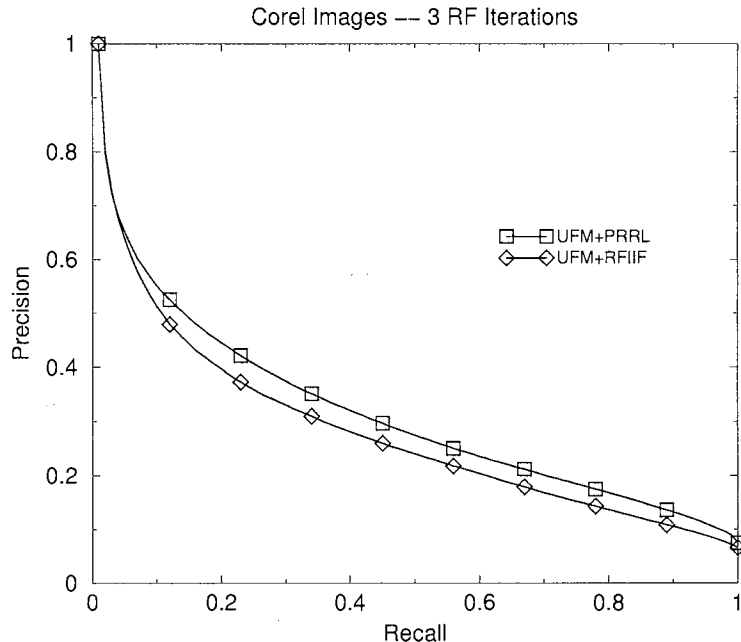
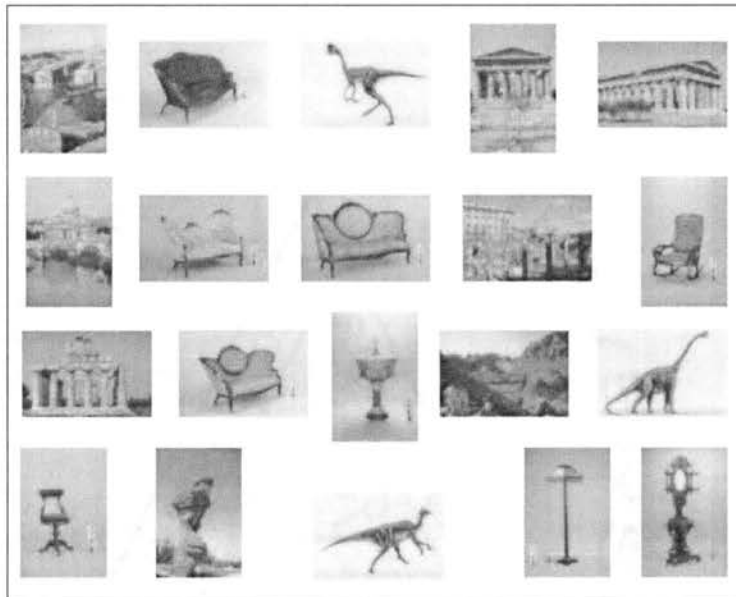


Figure 5.8: Retrieval performance after three RF iterations with PRRL and other methods on *Corel* data.

weighting or some other weighting heuristic, we could make a more informed initial estimate of the relevance of regions in the new query. We plan to investigate the possibility of incorporating inter-query learning into the PRRL framework as part of our future work.

5.3 Intra-Query Learning with Generalized Support Vector Machines

Several RF schemes based on SVM learning [18, 56, 156] have been applied to significantly improve retrieval performance in CBIR systems that use fixed-length global image representations. In [18], relevant images are used to estimate the distribution of target images by fitting a tight hypersphere in the non-linearly transformed feature space. In [156], the problem is regarded as a two-class classification problem and a maximum margin hyperplane in the non-linearly transformed feature space is used



Initial Retrieval Set with UFM, precision = 0.3



Retrieval Set with UFM+PRRL after 2 RF iterations, precision = 0.75

Figure 5.9: Retrieval results on random query image (top leftmost). The images are sorted based on their similarity to the query image. The ranks descend from left to right and from top to bottom.

Fortunately, a GSVM [84] (described in Section 3.1.5) allows the use of an arbitrary kernel and it can lead to a decision function that is as satisfactory as that of a conventional SVM. Because GSVM does not place restrictions on the kernel, any similarity measure (i.e., not necessarily an inner product one) can be used.

We now describe our GSVM-based learning approach. Let an image \mathbf{x} be represented by a set of regions $\{\mathcal{R}_i\}_1^n$, where $\mathcal{R}_i = \{\mathbf{r}_i\}$ is the descriptor of the i -th region and $\mathbf{r}_i \in \mathfrak{R}^d$ is a feature vector extracted from the i -th region. Let $S(\mathbf{x}_i, \mathbf{x}_j)$ be an arbitrary similarity measure between two images. During the RF process for a particular query image, the user marks each retrieved image \mathbf{x}_i as relevant ($y_i = 1$) or non-relevant ($y_i = 0$). We use the set of cumulative retrievals $\mathcal{R} = \{(\mathbf{x}_i, y_i)\}_1^m$ as training data in (3.6). Set $K(\mathbf{x}_i, \mathbf{x}_j) = S(\mathbf{x}_i, \mathbf{x}_j)$ and let

$$s_{\mathbf{x}_i} = [S(\mathbf{x}_i, \mathbf{x}_1), S(\mathbf{x}_i, \mathbf{x}_2), \dots, S(\mathbf{x}_i, \mathbf{x}_m)]^T$$

That is, $s_{\mathbf{x}_i}$ is the vector of similarities of \mathbf{x}_i to all training images. Then, the (i, j) -th entry of matrix \mathbf{A} in (3.6) is $s_{\mathbf{x}_i} \cdot s_{\mathbf{x}_j}$ (i.e., the dot product of $s_{\mathbf{x}_i}$ and $s_{\mathbf{x}_j}$). The equivalent (non-matrix) notation for (3.6) is then as follows

$$\min_{\boldsymbol{\alpha} \in \mathfrak{R}^m} \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j s_{\mathbf{x}_i} \cdot s_{\mathbf{x}_j} - \sum_{i=1}^m \alpha_i$$

$$s.t. \quad \sum_{i=1}^m \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq c$$

Let $K_{\mathcal{R}}(\mathbf{x}_i, \mathbf{x}_j) = s_{\mathbf{x}_i} \cdot s_{\mathbf{x}_j}$ (i.e., identity kernel over this new representation). Note that the above optimization problem is that of a standard SVM with an identity

kernel over this new representation

$$\min_{\boldsymbol{\alpha} \in \mathfrak{R}^m} \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K_{\mathcal{R}}(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^m \alpha_i \quad (5.5)$$

$$s.t. \quad \sum_{i=1}^m \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq c$$

Thus, by representing each image as a vector of its similarity (as given by the arbitrary region-based similarity measure S) to all training images, we can use an ordinary SVM. The proposed learning algorithm is summarized in Figure 5.11.

1. Retrieve the k most similar images to query image \mathbf{x}
2. While More RF Iterations Do
 - (a) User marks the k images as relevant or non-relevant
 - (b) $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\mathbf{x}_j, y_j)\}_1^k$
 - (c) Compute standard SVM by solving (5.5) on training data \mathcal{R}
 - (d) Compute the score $f(\mathbf{x})$ of each database image \mathbf{x} using resulting SVM decision function $f(\mathbf{x}) = \sum_{i=1}^{|\mathcal{R}|} \alpha_i y_i K_{\mathcal{R}}(\mathbf{x}, \mathbf{x}_i) + b$
 - (e) Retrieve the k highest-score database images

Figure 5.11: GSVM-based RF Learning Algorithm.

5.3.1 Experimental Results

In this section we present experimental results obtained with the proposed GSVM-based learning approach. The retrieval performance is measured by precision (1.1) and recall (1.2). The *Corel* data set (described in Section 5.2.3) was used for evaluation.

We tested the performance of UFM, UFM with the proposed GSVM-based learn-

ing method (UFM+GSVM), UFM with PRRL (UFM+PRRL), UFM with RFIIF (UFM+RFIIF), IRM, IRM with the proposed GSVM-based learning approach (IRM+GSVM), EMD, and GEMD (the method that uses a generalized Gaussian kernel with EMD described in Section 5.1). Every image is used as a query image. The size of the retrieval set is 20. A uniform weighting scheme is used to set the region weights of each query and target images. For UFM+GSVM, UFM+PRRL, UFM+RFIIF, GEMD, and IRM+GSVM, user’s feedback was simulated by carrying out 3 RF iterations for each query. Because the images in the data set are labelled according to their category, it is known whether an image in the retrieval set would be labelled as relevant or non-relevant by the user. After each RF iteration in UFM+GSVM, GEMD, and IRM+GSVM, the set of labelled cumulative retrieved images is used as training data for a SVM and the resulting decision function is used to rank database images. We used the *libsvm* [16] package for computing the SVM. Similarly, the set of cumulative retrieved images is used as training data in UFM+PRRL and UFM+RFIIF.

The average precision of the 2000 queries with respect to different number of RF iterations is shown in Figure 5.12. Figures 5.13 through 5.16 show the precision recall curves after each RF iteration. We can observe that UFM+GSVM has the best performance. Also, both UFM+GSVM and IRM+GSVM continue to have a significant improvement in performance after the first RF iteration. The initial decrease in performance with IRM+GSVM may be due to the initial lack of relevant training data because of the low initial retrieval precision of IRM.

The experimental results on general-purpose images show convincingly the efficacy of the proposed method in improving image retrieval performance. Currently, for each query, the user’s RF is used as training data and the learning process starts from ground up for each new query. However, it is also possible to exploit the long term learning accumulated over the course of many query sessions. This would be very beneficial specially in the initial retrieval set since, instead of ranking images based

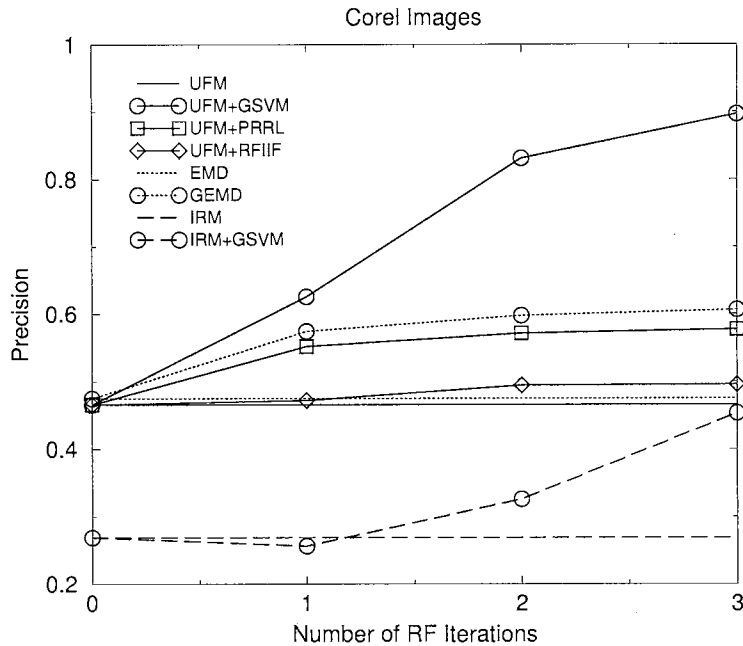


Figure 5.12: Retrieval performance at different number of RF iterations with the proposed GSVM-based approach and other methods on *Corel* data.

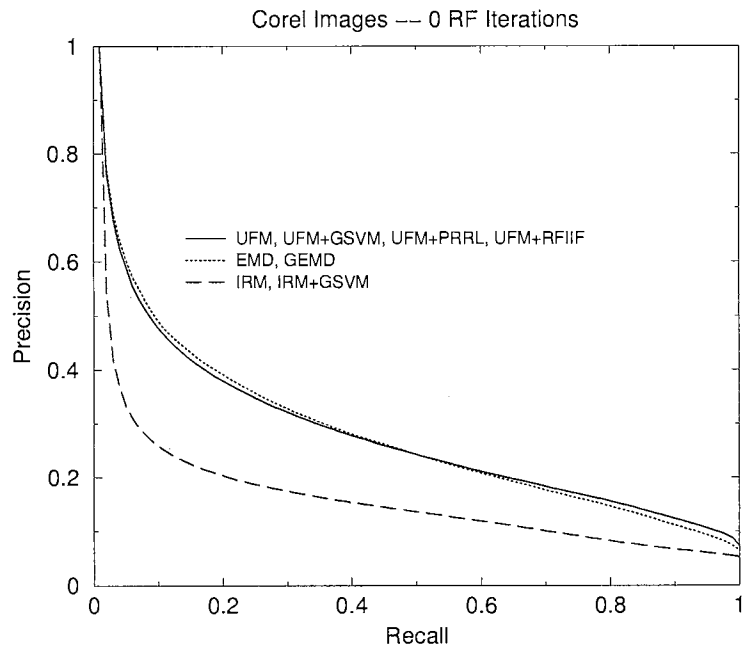


Figure 5.13: Retrieval performance in initial retrieval set with the proposed GSVM-based approach and other methods on *Corel* data.

only on the similarity measure, we could make a more informed initial estimate of the relevance of images to the user's query concept. We plan to investigate the possibility

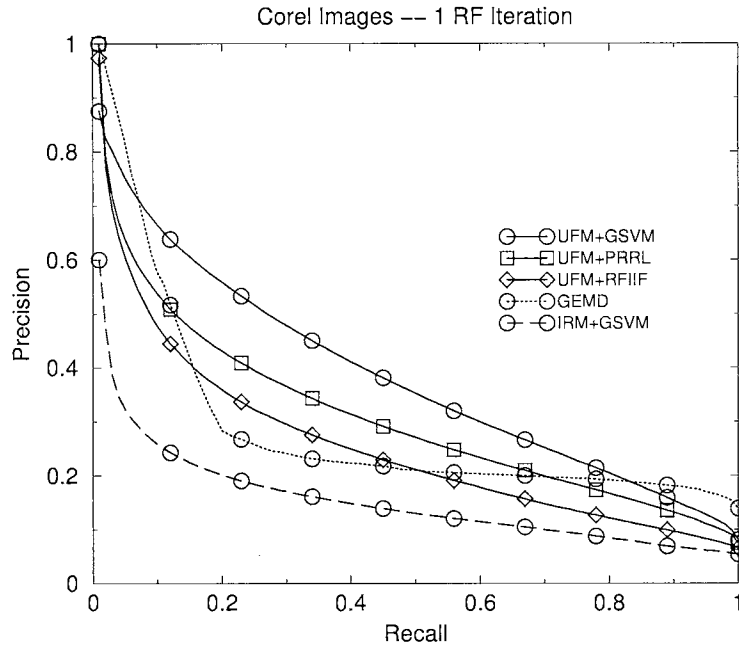


Figure 5.14: Retrieval performance after one RF iteration with the proposed GSVM-based approach and other methods on *Corel* data.

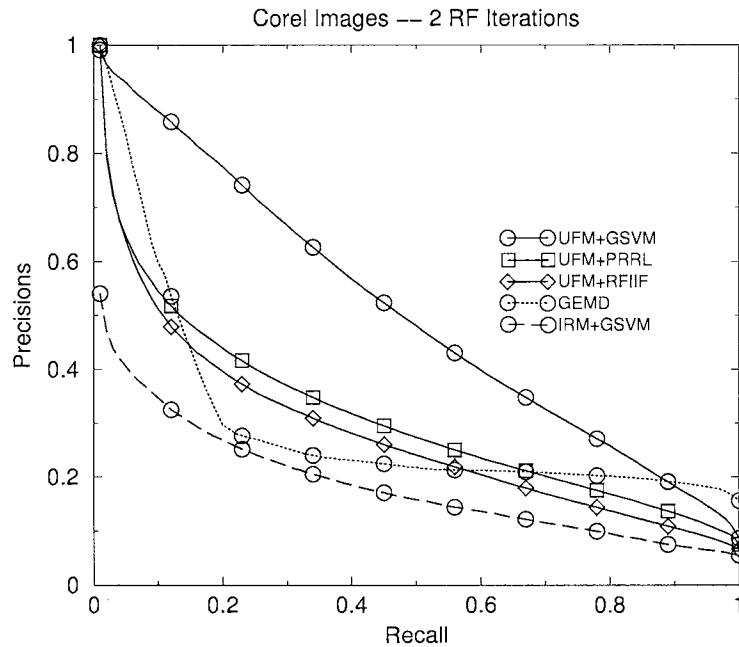


Figure 5.15: Retrieval performance after two RF iterations with the proposed GSVM-based approach and other methods on *Corel* data.

of incorporating long-term learning into this GSVM-based learning framework as part of our future work.

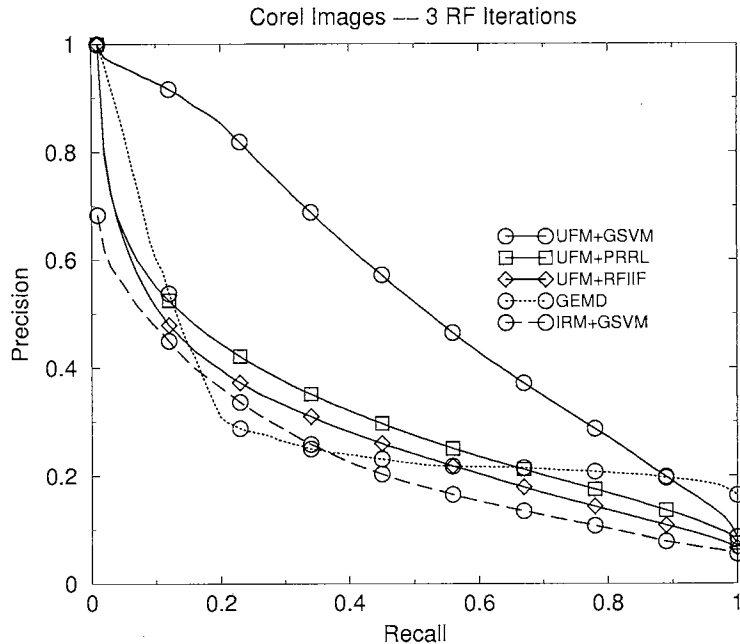


Figure 5.16: Retrieval performance after three RF iterations with the proposed GSVM-based approach and other methods on *Corel* data.

5.4 Improving Image Segmentation

A large number of image segmentation techniques have been proposed in the literature. However, semantically meaningful image segmentation still remains an open and difficult problem. This is mainly due to the fact that most image segmentation algorithms create regions that are homogeneous with respect to one or more low-level features. Unfortunately, homogeneous regions based on low-level features usually do not correspond to meaningful objects. Because what an object is ultimately depends on high-level human knowledge, it is very difficult to design segmentation algorithms that can extract semantic objects from images. To the best of our knowledge, no approach has been proposed that exploits intra/inter-query learning for automatically improving image segmentation. We propose an algorithm that exploits both intra and inter-query learning for automatically improving the segmentation of images in a database.

We assume the existence of a region-based CBIR system with a set of database

images that have been segmented. Thus, there is an initial segmentation of the images in the database. Then, through the use of intra and inter-query learning, this initial segmentation (and subsequent ones) is improved. We use a generic and simple clustering-based image segmentation algorithm based on k -means clustering. Because the focus of our approach is on improving an initial segmentation through the use of intra and inter-query learning, this algorithm (which produces a “not so good” segmentation) serves our purpose. The major advantage of this segmentation procedure is its low computational cost.

The segmentation algorithm is shown in Figure 5.17. To segment an image, it is first partitioned into blocks of $n \times n$ pixels. Then, a feature vector $\mathbf{b} \in \mathcal{R}^d$ is extracted from each block. The k -means algorithm is used to cluster the set of feature vectors into several classes with every class corresponding to a region in the segmented image. Each cluster is represented by cluster center $\mathbf{c} \in \mathcal{R}^d$ and a weight vector $\mathbf{w} \in \mathcal{R}^d$. The weight vector specifies the weight/importance of each feature dimension in a cluster. This agrees with our intuition that the weight/importance of each feature may be different in each image region. We assume the use of a weighted distance measure $dist_{\mathbf{w}}(\mathbf{b}, \mathbf{c})$ that can be used for classifying blocks into clusters. In order to determine the number of clusters k to use, the segmentation algorithm is run with increasing values of k up to a maximum number max_k . For each value of k , after running the segmentation algorithm, a clustering validity measure is used to assess the goodness of the resulting clustering. For example, the Xie-Bene (XB) validity measure [153] could be used. It is a measure of the compactness-to-separation ratio and is defined as

$$XB = \frac{1}{k} \frac{\sum_{i=1}^k \sigma_i^2}{D_{min}}$$

where D_{min} is the smallest distance between two cluster centers (i.e., separation), and σ_i is the sum of variances for the i -th cluster (i.e., compactness). Thus, a smaller

value of XB indicates a better clustering.

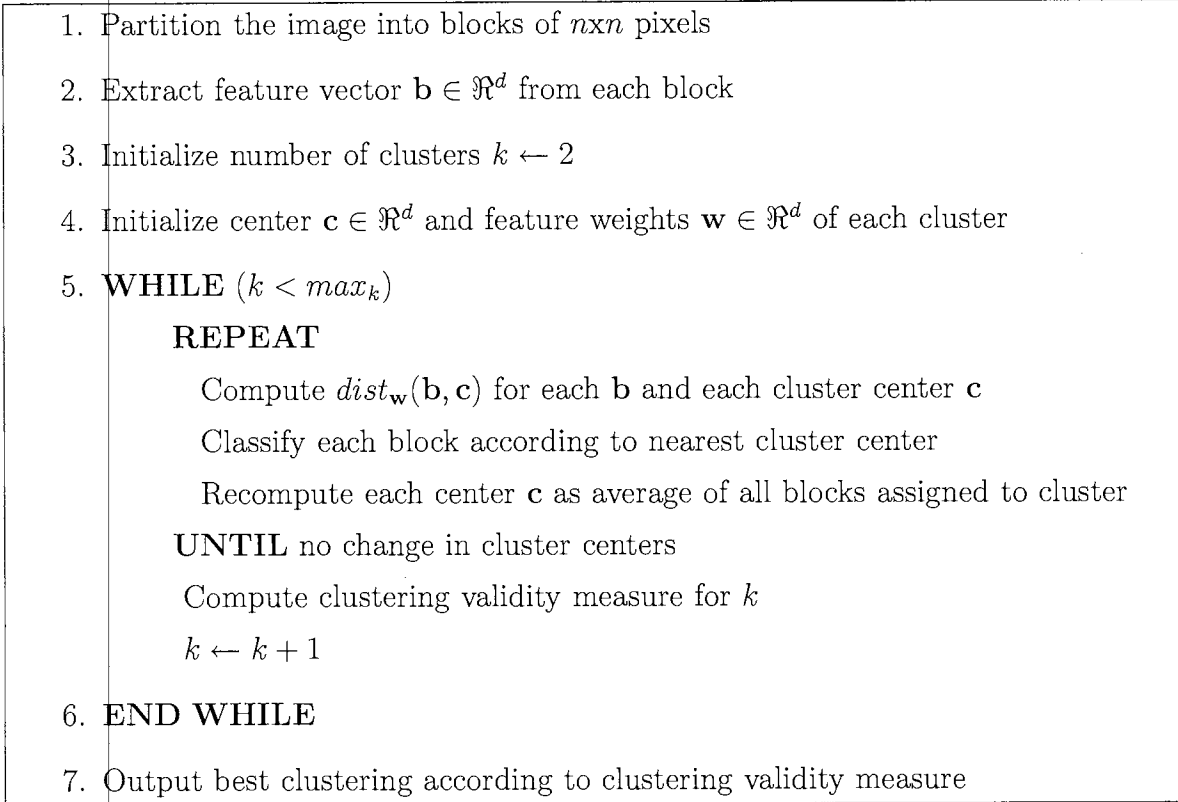


Figure 5.17: Simple Segmentation Algorithm.

The learning framework that we use in our approach is multiple-instance learning (MIL) [25, 85, 87] (described in Section 3.2.1). We view a segmented image as a bag consisting of a collection of instances (i.e., regions). For a given query, we assume that the user’s decision to label an image in the retrieval set as relevant is based on the presence of at least one particular object in the image. Similarly, a user labels an image as non-relevant if none of the objects in the image correlate with the user’s concept. At the end of the query session, given the set of cumulative user-labelled images (i.e., intra-query learning), we use MIL to find commonalities among the relevant images that do not appear in the non-relevant images. Such commonalities can be captured by the DD function. Intuitively, the larger the DD value at a point $(\mathbf{t}, \mathbf{w}')$, the more likely that image regions whose center \mathbf{c} is close to \mathbf{t} (measured by $dist_{\mathbf{w}'}(\mathbf{c}, \mathbf{t})$) appear in relevant images. Thus, if $DD(\mathbf{t}, \mathbf{w}')$ is large,

\mathbf{t} is the prototypical feature vector of a region that is common among the relevant images and uncommon among the non-relevant images. Also, \mathbf{w}' gives the relative importance of the different features in discriminating that particular region. This information can be used to improve the segmentation of the relevant images. For example, in the simple segmentation algorithm described in Figure 5.17, we could make a more informed decision on the initialization of (one or more) (\mathbf{c}, \mathbf{w}) pairs. The (possibly improved) segmentation of those images can then be stored in the database so that future queries can benefit from it (i.e, inter-query learning). That is, better segmentations will result in both future better retrieval performance and future improved updating of image segmentations. This basic idea is illustrated in Figure 5.18.

The DD function may have multiple local maxima (See Figure 5.19). A low value for $DD(\mathbf{t}, \mathbf{w}')$ means that this point is not particularly useful in discriminating between relevant and non-relevant images. Thus, points with low DD value are not useful. We can use a threshold to discriminate between points. Thus, only points whose DD value is above the threshold are considered for further exploitation (See Figure 5.20).

A segmented image \mathbf{x} consists of a set of regions, with each region represented by a (\mathbf{c}, \mathbf{w}) pair. Given the set of cumulative retrieved images $\mathcal{R} = \{(\mathbf{x}_i, y_i)\}_1^n$, where $y_i \in \{1, 0\}$ is the class label (i.e., relevant or non-relevant). Let $\mathcal{R}^+ = \{\mathbf{x}_i \mid (\mathbf{x}_i, 1) \in \mathcal{R}\}$.

The first step in our approach is to start an optimization of the DD function at the (\mathbf{c}, \mathbf{w}) of each region from every $\mathbf{x} \in \mathcal{R}^+$ and find the corresponding maximizer $(\mathbf{t}, \mathbf{w}')$. Let \mathcal{T} be the set of all such maximizers. Thus, we follow the heuristic applied in [87] to search for maxima of the DD function. That is, start an optimization of the DD function at each instance from every positive bag. Since, according to the definition of the DD function, a maximum DD point is made of contributions from some set of

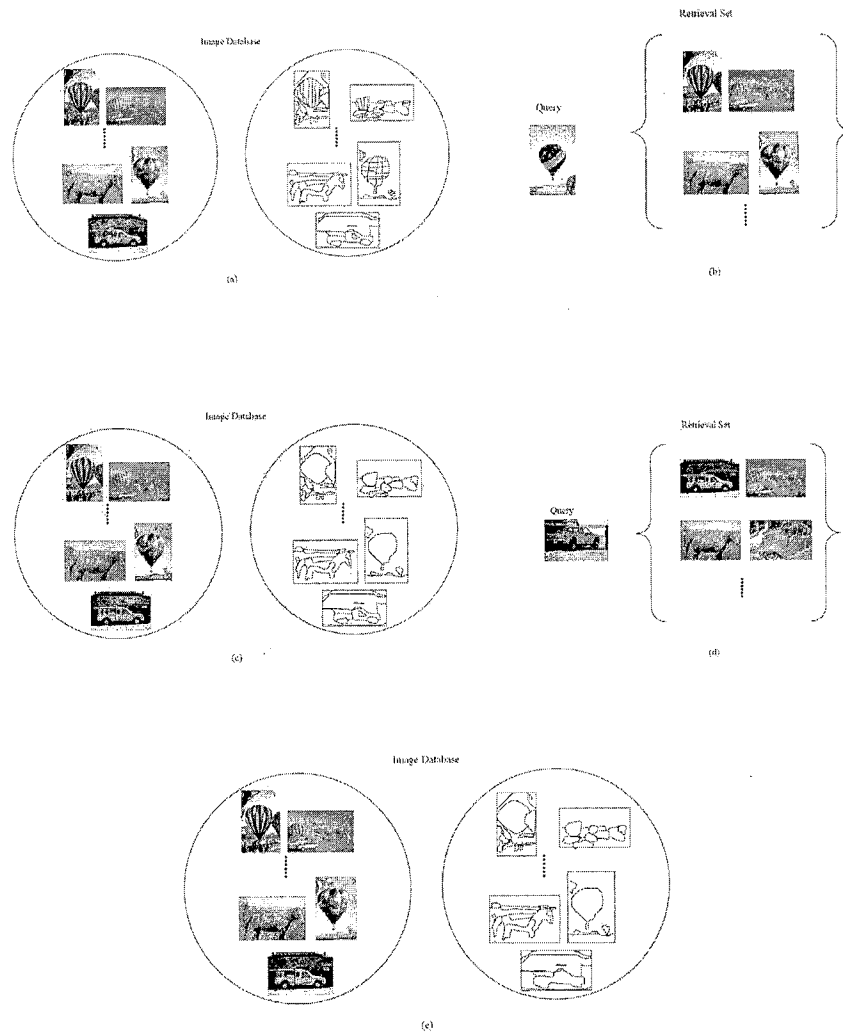


Figure 5.18: Basic idea of MIL-based approach: a) set of images in a database along with their initial segmentations/region-based representations; b) query image and corresponding retrieval set; c) MIL is performed on set of user-labelled relevant images to improve and update the segmentation/region-based representation of those images in the database; d) query image and corresponding retrieval set; e) MIL is performed on set of user-labelled relevant images to improve and update the segmentation/region-based representation of those images in the database.

positive bags, each maximum DD point is likely to be close to one or more instances from positive bags. The optimization can be solved by Powell's method [109].

The next step is to determine which maximizers in \mathcal{T} are useful and thus we want

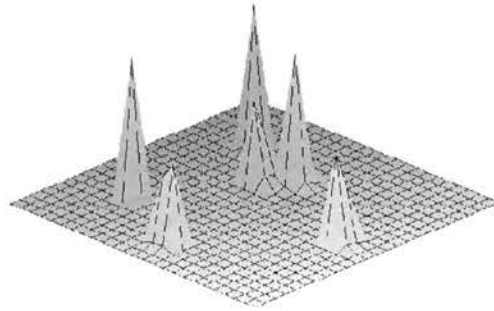


Figure 5.19: The space defined by the DD function may have multiple maximizers.

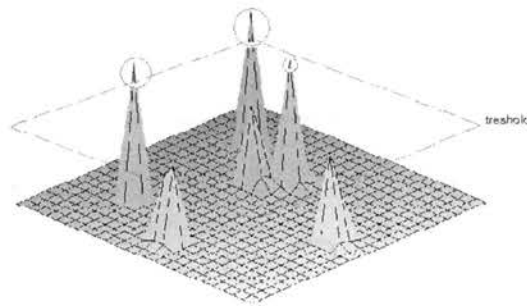


Figure 5.20: A threshold can be used to discriminate between useful and not-useful maximizers of the DD function.

to keep (See Figure 5.20). For example, we could use an adaptive threshold which is equal to the average of the maximum and minimum DD value of all the maximizers. Thus, after this filtering step, maximizers with low DD value have been removed from \mathcal{T} . In order to avoid having maximizers that are duplicates (or slight variations) of one another, for every pair of maximizers in \mathcal{T} that are very similar, we can remove the one with lowest DD value from \mathcal{T} .

Next, based on \mathcal{T} , we consider possible updates to the segmentation of each $\mathbf{x} \in$

\mathcal{R}^+ . That is, instead of completely re-segmenting an image, we consider possible changes that could improve the existing segmentation. This is desirable because an image may be relevant under different query concepts. For example, the image in Figure 5.21 is relevant under both the “balloons” and the “cars” query concepts. Thus, the segmentation of the “balloons” object(s) in the image can be improved at the end of a query session for which the user’s concept was “balloons”. Similarly, the segmentation of the “cars” object(s) in the image can be improved at the end of a query session for which the user’s concept was “cars” (See Figure 5.18).



Figure 5.21: An example of an image that is relevant under different user’s concepts (e.g., “balloons”, and “cars”).

Therefore, for each $\mathbf{x} \in \mathcal{R}^+$, the segmentation of only those regions/objects in \mathbf{x} that resulted in the user labelling \mathbf{x} as relevant is considered for updating. Thus, we first have to determine the mapping between maximizers in \mathcal{T} and regions in \mathbf{x} . This could be done by computing the distance from every maximizer $(\mathbf{t}, \mathbf{w}') \in \mathcal{T}$ to the (\mathbf{c}, \mathbf{w}) of every region in \mathbf{x} . If the distance is smaller than some threshold, we say that $(\mathbf{t}, \mathbf{w}')$ maps to (\mathbf{c}, \mathbf{w}) (i.e., $(\mathbf{t}, \mathbf{w}')$ is a “prototype” of that region) (See Figure 5.22).

The proposed updates to the segmentation of \mathbf{x} vary according to the type of mapping between maximizers in \mathcal{T} and regions in \mathbf{x} . The following are the different

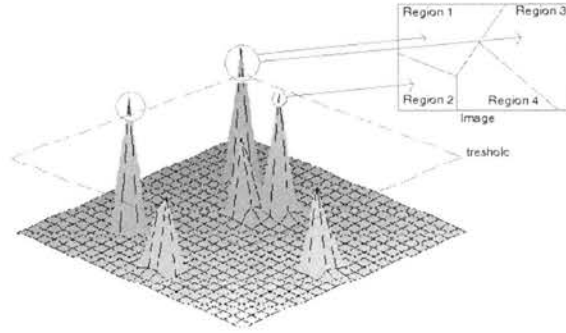


Figure 5.22: Regions in the segmented image can be associated with the closest maximizer of the DD function.

types of possible mappings along with their corresponding proposed segmentation update:

1. A maximizer $(\mathbf{t}, \mathbf{w}') \in \mathcal{T}$ does not map to the (\mathbf{c}, \mathbf{w}) of any region in \mathbf{x} . Intuitively, this means that the important/common region whose prototype is given by $(\mathbf{t}, \mathbf{w}')$ either does not appear as an independent region in the segmentation of \mathbf{x} or does not appear in \mathbf{x} at all. The proposed change is to add $(\mathbf{t}, \mathbf{w}')$ as a new cluster center (See Figure 5.23). Then, after re-clustering all the \mathbf{b} in \mathbf{x} , we can determine the validity of the proposed change. For instance, if after re-clustering, the value of the newly inserted cluster center is far from its original value of $(\mathbf{t}, \mathbf{w}')$ or only a few (or none) of the \mathbf{b} have been assigned to the new cluster, we may conclude that \mathbf{x} in fact does not contain that important/common region. In such case, the proposed change can be undone simply by removing the newly inserted cluster center and keeping the original segmentation.
2. A maximizer $(\mathbf{t}, \mathbf{w}') \in \mathcal{T}$ maps to the (\mathbf{c}, \mathbf{w}) of exactly one region in \mathbf{x} (See Figures 5.24 and 5.25). Intuitively, this means that the important/common

region whose prototype is given by $(\mathbf{t}, \mathbf{w}')$ does appear as an independent region in the segmentation of \mathbf{x} . However, since $(\mathbf{t}, \mathbf{w}')$ is a prototype for such region, the segmentation of that region in \mathbf{x} may still be improved by moving (\mathbf{c}, \mathbf{w}) towards $(\mathbf{t}, \mathbf{w}')$ (and re-clustering). We assume that this is always a good update to the segmentation of \mathbf{x} .

3. A maximizer $(\mathbf{t}, \mathbf{w}') \in \mathcal{T}$ maps to the (\mathbf{c}, \mathbf{w}) of more than one region in \mathbf{x} (See Figures 5.26 and 5.27). Intuitively, this means that the important/common region whose prototype is given by $(\mathbf{t}, \mathbf{w}')$ appears as more than one independent regions in the segmentation of \mathbf{x} . The proposed change is to merge those regions by removing the (\mathbf{c}, \mathbf{w}) of each and adding a new (\mathbf{c}, \mathbf{w}) that is the average of the all the (\mathbf{c}, \mathbf{w}) that were removed. Then, after re-clustering all the \mathbf{b} in \mathbf{x} , we can determine the validity of the proposed change.
4. More than one maximizer $(\mathbf{t}, \mathbf{w}') \in \mathcal{T}$ maps to the (\mathbf{c}, \mathbf{w}) of one or more regions in \mathbf{x} (See Figures 5.28 and 5.29). Intuitively, if more than one maximizer maps to the (\mathbf{c}, \mathbf{w}) of exactly one region in \mathbf{x} , this means that the important/common regions whose prototype are given by the maximizers appear as a single region in the segmentation of \mathbf{x} . The proposed change is to split that region in \mathbf{x} by removing its corresponding (\mathbf{c}, \mathbf{w}) and adding the maximizers. Then, after re-clustering all the \mathbf{b} in \mathbf{x} , we can determine the validity of the proposed change. However, if more than one maximizer maps to the (\mathbf{c}, \mathbf{w}) of more than one region in \mathbf{x} , there is no intuitive update to the segmentation of \mathbf{x} (i.e., both a merge and a split operation would have to be done at the same time). This case should not occur very often. We do not make any update to the segmentation of \mathbf{x} in this case.

After all mappings between maximizers in \mathcal{T} and regions in \mathbf{x} are obtained, the proposed updates to the segmentation of \mathbf{x} are carried out all at once. Then, after

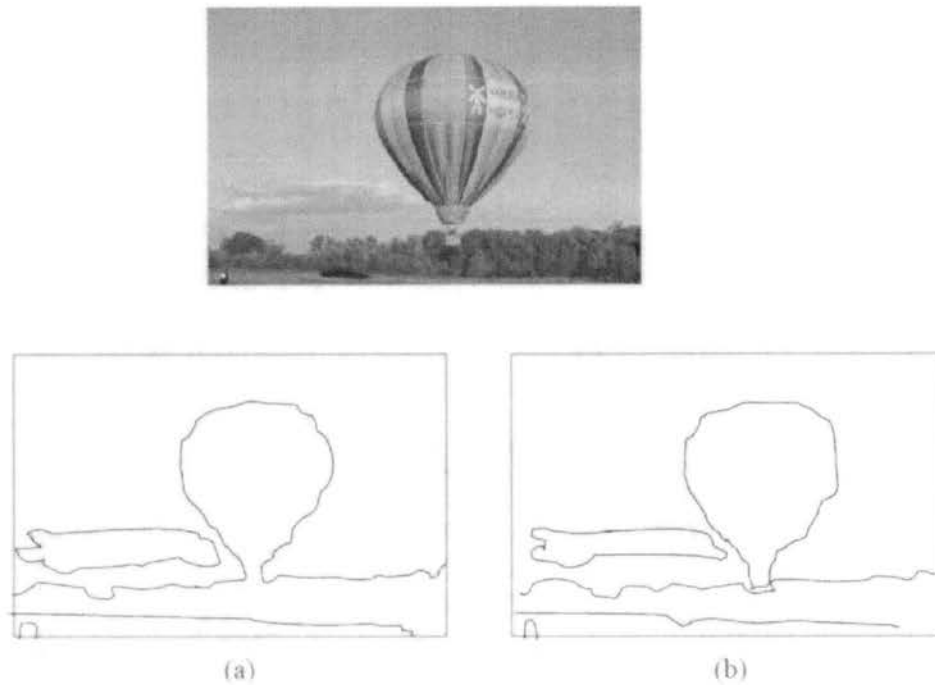


Figure 5.23: Sample of under-segmentation: a) an important object (i.e., the balloon) does not appear as an independent region in the original image segmentation; b) the important object appears as an independent region after adding a new cluster with $(\mathbf{t}, \mathbf{w}')$ that is prototypical of the important object, and re-clustering.

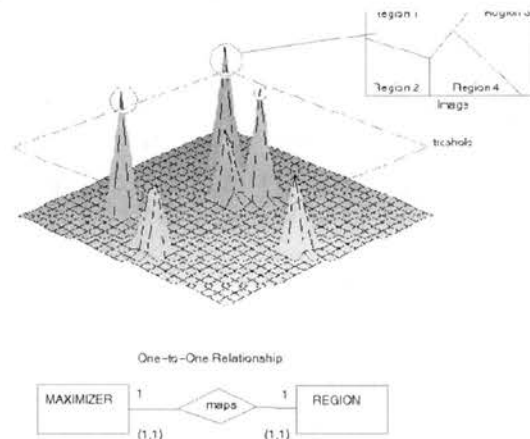


Figure 5.24: There is a one-to-one mapping between a maximizer and a region. The proposed change is to update (\mathbf{c}, \mathbf{w}) by moving it towards the maximizer $(\mathbf{t}, \mathbf{w}')$.

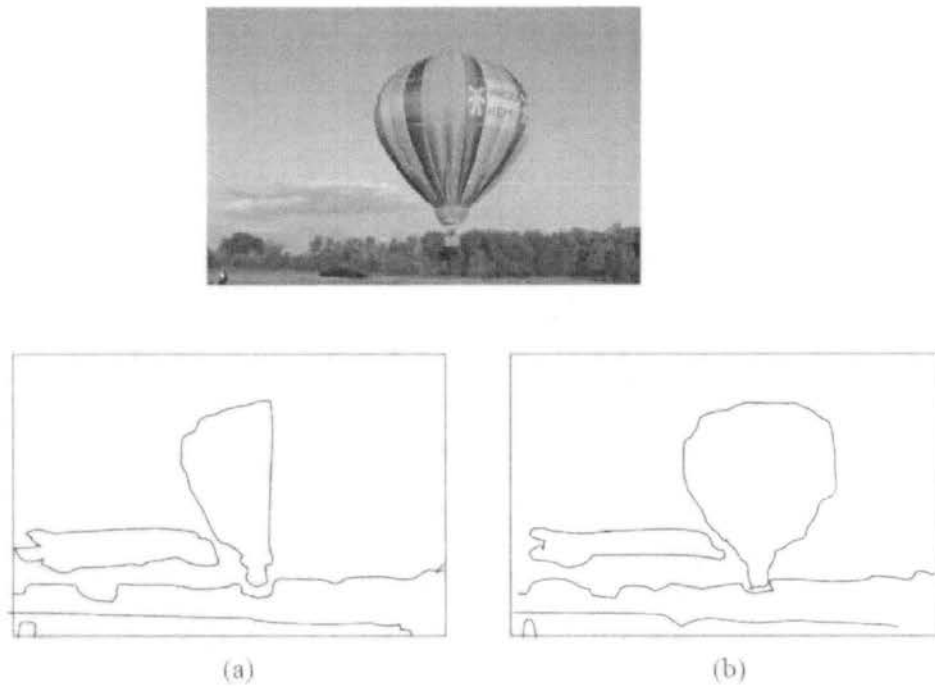


Figure 5.25: Sample of poor segmentation: a) an important object (i.e., the balloon) is not well segmented in the original image segmentation; b) the segmentation of the important object improves after moving the corresponding (\mathbf{c}, \mathbf{w}) towards the prototypical $(\mathbf{t}, \mathbf{w}')$ of the important object, and re-clustering.

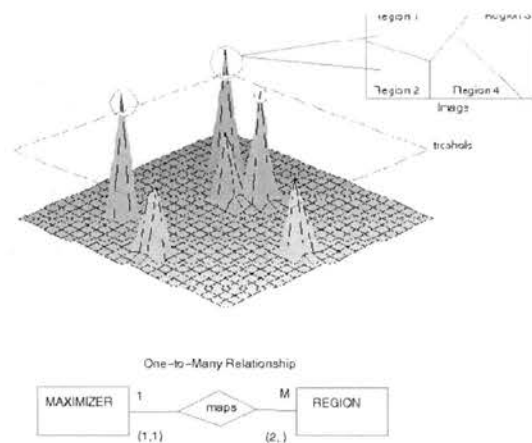


Figure 5.26: There is a one-to-many mapping between a maximizer and more than one region classifiers. The proposed change is to merge the regions by removing their (\mathbf{c}, \mathbf{w}) and adding the maximizer as the cluster prototype of a new region.

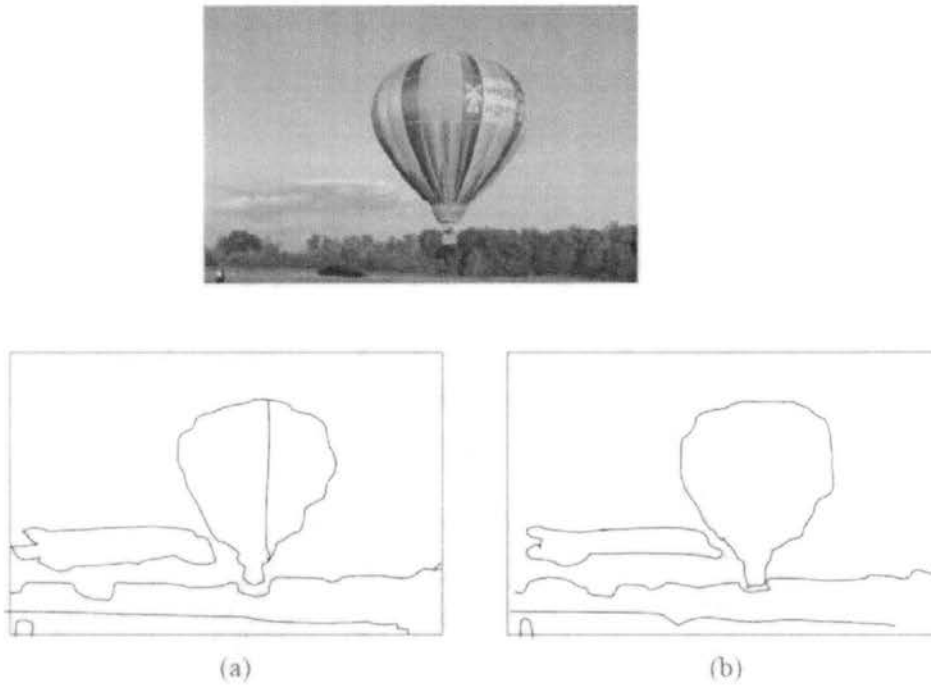


Figure 5.27: Sample of over-segmentation: a) an important object (i.e., the balloon) appears as more than one independent regions in the original image segmentation; b) the segmentation of the important object improves after merging the corresponding (\mathbf{c}, \mathbf{w}) by removing them, adding the prototypical $(\mathbf{t}, \mathbf{w}')$ of the important object, and re-clustering.

re-clustering all the \mathbf{b} in \mathbf{x} , the updates are evaluated in an incremental fashion and, if necessary, undone. Figure 5.30 shows the proposed algorithm for improving image segmentation.

The description of the proposed approach is very generic since there are still many important open questions that need to be addressed. For instance, informed ways of determining the thresholds that are used are needed. We will develop an specific implementation of this approach as part of our future work.

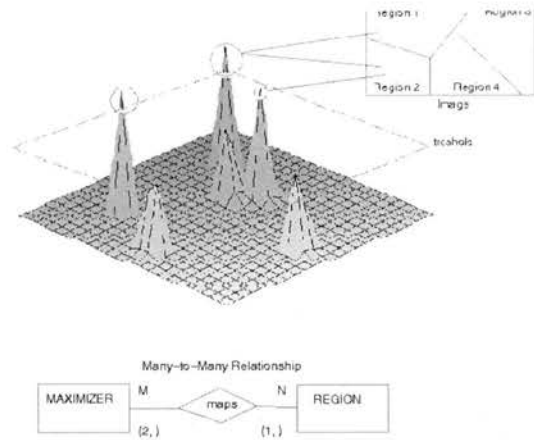


Figure 5.28: There is a many-to-many mapping between more than one maximizer and the (c, w) of at least one region. If the maximizers map to just one region, the proposed change is to split the region by removing its (c, w) and adding the prototypical maximizers as new regions.

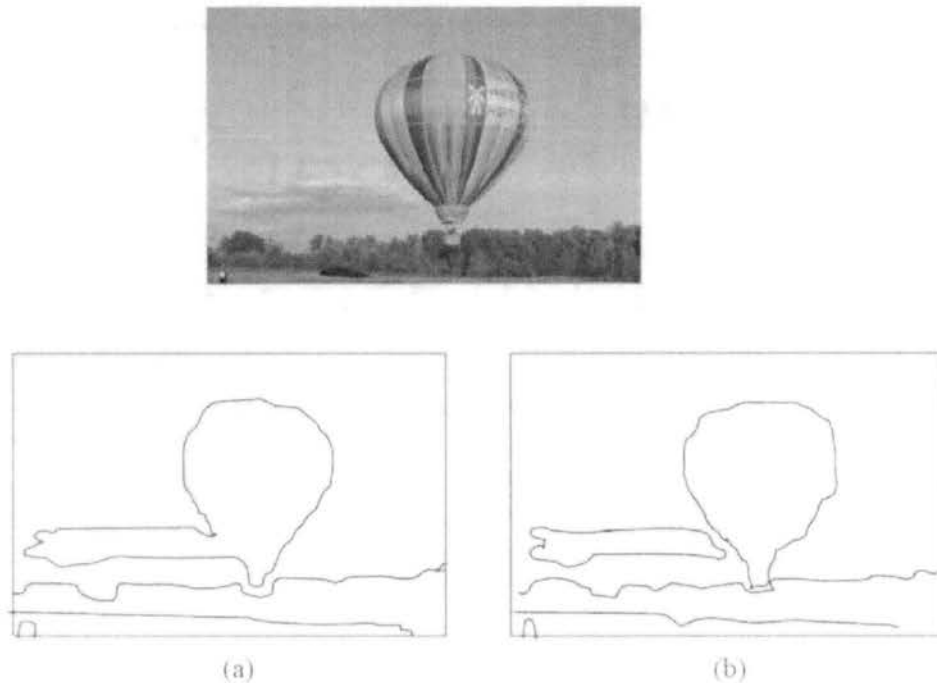


Figure 5.29: Sample of under-segmentation: a) important objects (i.e., the balloon and the cloud) appear as a single region in the original image segmentation; b) the segmentation of the important objects improves after splitting them by removing the (c, w) of their original single region, adding the prototypical maximizers of the important object, and re-clustering.

```

1.  $\mathcal{T}$  is set of maximizers of the DD function. Initialize  $\mathcal{T} \leftarrow \emptyset$ 
2. FOR (every  $(\mathbf{c}, \mathbf{w})$  of each region from every  $\mathbf{x} \in \mathcal{R}^+$ )
    Find maximizer  $(\mathbf{t}, \mathbf{w}')$  of DD function from starting point  $(\mathbf{c}, \mathbf{w})$ 
     $\mathcal{T} \leftarrow \mathcal{T} \cup (\mathbf{t}, \mathbf{w}')$ 
3. Remove duplicated maximizers and maximizers with low DD value from  $\mathcal{T}$ 
4. FOR (every  $\mathbf{x} \in \mathcal{R}^+$ )
    FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}$ )
        FOR (each  $(\mathbf{c}, \mathbf{w})$  from every region in  $\mathbf{x}$ )
            Compute  $dist_{\mathbf{w}'}(\mathbf{t}, \mathbf{c})$ 
 $\mathcal{T}_{(1-null)}$  is set of maximizers in  $\mathcal{T}$  that do not map to any region in  $\mathbf{x}$ 
 $\mathcal{T}_{(1-1)}$  is set of maximizers in  $\mathcal{T}$  with one-to-one mappings
 $\mathcal{T}_{(1-M)}$  is set of maximizers in  $\mathcal{T}$  with one-to-many mappings
 $\mathcal{T}_{(M-1)}$  is set of maximizers in  $\mathcal{T}$  with many-to-one mappings
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(1-1)}$ )
            UPDATE corresponding  $(\mathbf{c}, \mathbf{w})$  by moving it towards  $(\mathbf{t}, \mathbf{w}')$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(1-null)}$ )
            ADD  $(\mathbf{t}, \mathbf{w}')$  as a new cluster prototype to segmentation of  $\mathbf{x}$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(1-M)}$ )
            MERGE corresponding regions in  $\mathbf{x}$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(M-1)}$ )
            SPLIT corresponding region in  $\mathbf{x}$ 
    REPEAT
        FOR (every  $\mathbf{b}$  in  $\mathbf{x}$ )
            Compute  $dist_{\mathbf{w}}(\mathbf{b}, \mathbf{c})$  for the  $(\mathbf{c}, \mathbf{w})$  of every region in  $\mathbf{x}$  and classify  $\mathbf{b}$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$   $\in \mathcal{T}_{(1-null)}$ )
            IF ADD not valid, UNDO and remove  $(\mathbf{t}, \mathbf{w}')$  from  $\mathcal{T}_{(1-null)}$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(1-M)}$ )
            IF MERGE not valid, UNDO and remove  $(\mathbf{t}, \mathbf{w}')$  from  $\mathcal{T}_{(1-M)}$ 
        FOR (every maximizer  $(\mathbf{t}, \mathbf{w}')$  in  $\mathcal{T}_{(M-1)}$ )
            IF SPLIT not valid, UNDO and remove  $(\mathbf{t}, \mathbf{w}')$  from  $\mathcal{T}_{(M-1)}$ 
    UNTIL (no change to segmentation of  $\mathbf{x}$ )

```

Figure 5.30: Algorithm for MIL-based Segmentation.

Chapter 6

Other Image Representations

The main idea of content-based image retrieval (CBIR) is to search on images directly. That is, instead of searching based on assigned keywords, we search visual content directly. However, we still need to use a set of features to represent visual content. In this chapter, we present an initial investigation into what we believe is the logical continuation of the CBIR idea of searching visual content directly. It is based on the observation that, since ultimately, the entire visual content of an image is encoded into its raw data (i.e., the raw pixel values), in theory, it should be possible to determine image similarity based on the raw data alone. That is, everything that we need to know regarding the visual content of the image is in the raw data itself. Humans are very good at looking at an image (i.e., the raw data) and extracting all the important features from it. Thus, all the important features are “hidden” in the raw data somewhere. The problem of feature extraction is just that we do not entirely know yet how (we, humans) “find” them. Thus, instead of attempting to determine image similarity based on a small set of (probably incomplete) set of features, why not have a similarity measure that is based on the raw data itself (since everything is in the raw data). We present an initial investigation, conducted in [41], into an image dissimilarity measure following from the theoretical foundation of the recently proposed normalized information distance (NID) [74]. A very crude approximation

of the Kolmogorov complexity of an image is created by compression. Using this approximation, we can calculate the NID between images and use it as a metric for CBIR. The compression-based approximation to Kolmogorov complexity, though very rough, is shown to be valid by proving that it creates a statistically significant dissimilarity measure by testing it against a null hypothesis of random retrieval. Although the approximations used in this initial investigation may not currently be practical for CBIR, the results are encouraging that additional research into methods guided by the NID approach may be fruitful.

6.1 Image Similarity with Normalized Information Distance

We attempt to bypass the feature selection step (and the distance metric in the corresponding feature space) by taking the normalized information distance (NID) [74] approach. The NID approach is based on the notion of Kolmogorov complexity [68, 77]. The information distance between two strings a and b is the complexity of the transformations of a into b and b into a . The information distance is normalized by the individual complexities of a and b . In theory, the complexity of a is measured by the length of the shortest program that can compute a from scratch. The complexity of the transformation of a into b is the length of the shortest program that can compute b given a as an auxiliary input.

Kolmogorov complexity is not computable, but it has been used as the foundation for the minimum description length (MDL) principle [26, 112] and the minimum message length (MML) principle [145]. In [74], NID was successfully applied to the problems of determining whole mitochondrial genome phylogenies and classifying natural languages when using a compression-based approximation of complexity. It has also been shown to be applicable to chain letters [8].

6.1.1 The Normalized Information Distance

The NID presented in [74] is based on the incomputable notion of Kolmogorov complexity. The Kolmogorov complexity of a string x , $K(x)$, is defined as the length of the shortest effective binary description of x . Broadly speaking, $K(x)$ may be thought of as the length of the shortest program that, when run with no input, outputs x . It has been shown that, although there are many universal Turing machines (and thus many possible shortest programs), the corresponding complexities differ by at most an additive constant [33]. Thus, $K(x)$ is the smallest amount of information that is needed by an algorithm to generate x . Let x^* be the smallest program that generates x . Then, $K(x) = |x^*|$. Similarly, the conditional Kolmogorov complexity of x relative to another string y , $K(x | y)$, is the length of the shortest program that, when run with input y , outputs x . Also, $K(x, y)$ is the length of the smallest program that generates x and y along with a description of how to tell them apart. The theory and development of the notion of Kolmogorov complexity are described in detail in [77]. The *information in y about x* is defined as [68, 74]

$$I(x : y) = K(x) - K(x | y^*)$$

A result from [32] shows that, up to additive constants, $I(x : y) = I(y : x)$. Thus [74],

$$K(x) + K(y | x^*) = K(y) + K(x | y^*) \quad (6.1)$$

The *information distance* $E(x, y)$ is defined as the length of a smallest program that generates x from y and y from x [74]. A result from [7] indicates that, up to an additive logarithmic term,

$$E(x, y) = \max\{K(y | x), K(x | y)\} \quad (6.2)$$

Because it is not normalized, (6.2) may not be an appropriate distance measure. For instance, according to (6.2), the distance between two very long strings that differ only in a few positions would be the same as the distance between two short strings that differ by the same amount. In [74], the NID $d(x, y)$ is proposed

$$d(x, y) = \frac{\max\{K(x | y^*), K(y | x^*)\}}{\max\{K(x), K(y)\}} \quad (6.3)$$

The function $d(x, y)$ is a normalized information distance (i.e., it is a distance metric, takes values in $[0,1]$, and satisfies the normalization condition). It is also universal because it includes every computable type of similarity in the sense that, whenever two objects are similar in normalized information in some computable sense, then they are at least that similar in $d(x, y)$ sense [74]. For proofs and more details, refer to [74].

6.1.2 Image Similarity Measure

Let x and y be two raw images (i.e., strings containing byte streams describing color information). In order to be able to use (6.3) for determining distance between x and y , we need to estimate $K(x)$, $K(y)$ and their conditional complexities $K(x | y)$, $K(y | x)$. For the conditional complexities, by (6.1), $K(x | y) = K(x, y) - K(y)$ (up to an additive constant) [74]. Also, $K(x, y) = K(xy)$ (up to additive logarithmic precision) [74].

The size of the compressed x is used to approximate $K(x)$, similarly for $K(y)$. The compressed size of concatenation of x with y is used to estimate $K(xy)$, similarly for $K(yx)$. We justify this by the observation that compression algorithms take advantage of redundancy (i.e., spatial, color coherence) in an image to shrink the representation. Therefore, intuitively, if x is a more complex image than y , the size of the compressed x would be larger than that of y . Thus, this corresponds to the intuition that $K(y)$

should be smaller than $K(x)$. Similarly, when x and y are very different, the size of the compressed xy should be larger than when x and y are very similar. Thus, using (6.3), the distance between two raw images x and y can be defined as

$$d'(x, y) = \frac{\max\{|c(xy)| - |c(y)|, |c(yx)| - |c(x)|\}}{\max\{|c(x)|, |c(y)|\}} \quad (6.4)$$

where $c(i)$ is the compressed version of input i and $|c(i)|$ is its corresponding size. Note that $|c(x)|$, $|c(xy)|$ are very rough approximations to $K(x)$ and $K(x, y)$. Thus, we do not expect (6.4) to result in a performance that is high enough for (6.4) to be used as a practical tool. The purpose of this preliminary investigation was just to obtain some preliminary evidence to whether or not the NID could be applied to the problem of determining image similarity. Depending on the preliminary results obtained with (6.4), we will then decide whether to investigate implementations of the NID based on better approximations to the true Kolmogorov complexities (more about this on the next section).

6.1.3 Experimental Results

In this section we present some preliminary experimental results obtained with the NID approach. The retrieval performance is measured by precision (1.1) and recall (1.2). The following data sets were used for evaluation:

1. The *Texture* data set (described in Section 4.3.3).
2. *GroundTruth* - the University of Washington GroundTruth image database [1]. The images are photographs of different regions and topics. Sample images are shown in Figure 6.1. We use the set of 675 annotated images. Each image contains multiple annotations (i.e., keywords).
3. *IAPR-12* - the benchmark database and standard queries from technical committee 12 of IAPR[59]. The data consists of 1000 images and 30 standard

queries. Sample images from the queries can be found in Figure 6.2.

4. The *Corel* data set (described in Section 5.2.3).

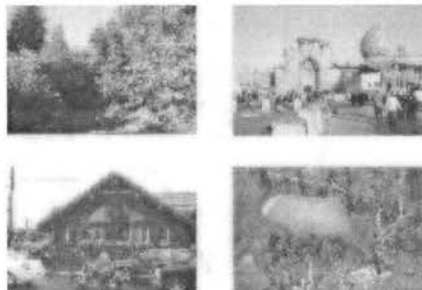


Figure 6.1: Sample images from *GroundTruth* data set.

The objective of our experiments was to obtain some preliminary evidence as to whether a crude approximation to the normalized information distance actually creates a statistically significant image similarity measurement. Therefore, we tested the performance against an uninformed method that used uniform random retrieval to select images. The *Texture* data set was used first. For this experiment, libucl [99] was used as the compressor. The image concatenation was a sequential placement of the raw bytes of the second image at the end of the first image. Each image was used as a query and the precision of a retrieval set of the twenty nearest images was measured. The results are presented in Table 6.1. The NID performed surprisingly well and is obviously statistically different than the random approach. It performs almost as well as 16-dimensional feature vector extracted using Gabor filters. Since the texture images contain the repeating patterns of the texture, they are probably the best case situation for approximation based on compression.

Table 6.1: *Texture* Data Set Performance

	Random	NID	Gabor
Precision at 20 images	0.079	0.80	0.81

The *GroundTruth* data set was used next. We define y as being relevant to x when x and y share at least one common annotation. For this experiment, gzip [48] was

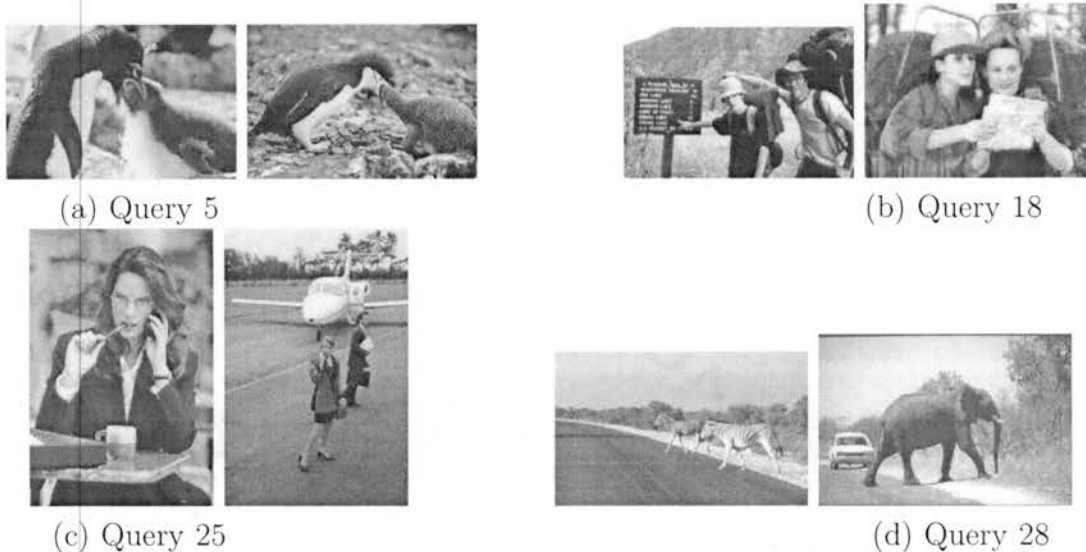


Figure 6.2: Sample query images from *IAPR-12* data set.

used as the compressor. The image concatenation was a sequential placement of the raw bytes of the second image at the end of the first image. Each image was used as a query and the precision of a retrieval set of the 20 nearest images was measured and presented in Table 6.2. The NID method had a precision of 0.578 and the random method has a precision of 0.414. To determine if the NID method is statistically different from the random method, McNemar's test [148] was used. In McNemar's test for two classifiers, A and B , the z statistic is

$$z = \frac{|n_{01} - n_{10}| - 1}{\sqrt{n_{10} + n_{01}}}$$

where n_{01} is the number of samples misclassified by A but not by B and n_{10} is the number of samples misclassified by B but not by A . In this case, $n_{01} = 2358$ and $n_{10} = 4572$ out of a total of 13500 classified samples (twenty for each of the 675 images) and $z = 26.58$. The quantity z^2 is distributed approximately as χ^2 with one degree of freedom. Thus we can reject the null hypothesis that the classifiers have the same error rate and assert that the NID is expressing a statistically significant similarity measure.

Table 6.2: *GroundTruth* Data Set Performance

	Random	NID
Precision at 20 images	0.414	0.578

The *IAPR-12* data set was used next. We used the queries that contained two images (queries 5, 18, 20, 21, 25, 26, and 28). Each image was used as a query image and the rank of the other image was determined by sorting the images based on distance from the query. For this experiment, *libucl* [99] was used as the compressor. Two methods of image concatenation were tried. In addition to the previous sequential concatenation, an interleaving of the two images was done by alternating the bytes from the two images. The sequential concatenation performed well on query 18 (Figure 6.2.(b)) with the desired retrieval image ranking first, but on query 25 (Figure 6.2.(c)) the desired image had rank 926. Over all of the queries, the average rank of the desired image was 501 and not different than random retrieval (which would average 499.5). Switching the concatenation to an interleaving approach improved the average rank to 395 but actually pushed the worst result from query 25 out to rank 981. Though the approach worked very well on some of the individual queries, further investigation of the *IAPR* data set is needed due to the difficulty of some the queries.

The *Corel* data set was used next. For this experiment, we used JPEG compression [104]. JPEG is a lossy compression algorithm that uses transform coding. First, the image is subdivided into blocks of 8x8 pixels. Then, a conversion to the frequency domain is performed by applying a two-dimensional discrete cosine transform (DCT) to each block. The results of psychophysical experiments suggest that the human eye is not so sensitive to high frequency brightness variation. Thus, the amount of information contained in the high frequency components can be greatly reduced without humans being able to perceive any significant difference in the image. Therefore, the next step is a quantization step in which each component in the frequency domain is

divided by a constant for that component and then rounded to the nearest integer. This is the main lossy step in the algorithm. The results of this quantization are then encoded by using a special form of lossless data compression known as entropy encoding. This involves arranging the quantized coefficients in a zig-zag order that groups similar frequencies together and then using Huffman coding [104]. Figure 6.3 shows the main steps of JPEG compression.

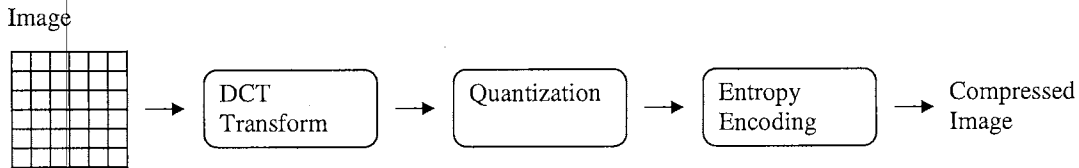


Figure 6.3: JPEG compression.

The image concatenation was a sequential placement of the quantized coefficients (resulting from the quantization step) of the second image at the end of the quantized coefficients of the first image. Then, the entropy encoding step was performed on the concatenated coefficients. Note that, in the quantization step, frequency components from both images that are close enough will be rounded to the same nearest integer (i.e., to the same quantized coefficient). Thus, the entropy encoder step will exploit not only redundancies between the two images but also implicitly, similarities between them. Each image was used as a query and the precision of a retrieval set of twenty nearest images was measured. The results are presented in Table 6.3. Once again, the NID performed surprisingly well and is obviously statistically different than the random approach. It does not perform much worse than unified feature matching (UFM) [17] (described in Section 2.3) using 9-dimensional feature vectors.

	Random	NID	UFM
Precision at 20 images	0.05	0.331	0.466

Although the NID measure is not computable and not even effectively approximable [76], it does provide insight into what we would want to do in the ideal case. This insight can be used to guide our attempts at simulating the NID measure at various levels of precision. We determined that even the very crude approximation to Kolmogorov complexity that compression generates was able to generate statistically significant dissimilarity measure for images when the NID approach was followed. This is an encouraging result that indicates that other attempts at simulating NID may yield good results.

We plan on exploring other methods of concatenating images and of compressing images, such as fractal and wavelet compression, that may better exploit the 2D nature of images. Another area where it may be useful to try the NID approach is in the matching of variable-length feature vectors. The NID approach may create a very practical method that goes beyond the individual region matching but does not require the expense of determining the higher level relationships among the regions. Another area of future research is the exploration of the NID approach as a feature-independent method of structuring an image data set.

Chapter 7

Conclusions and Future Work

In this dissertation, the problem of mapping the low-level physical characterization of images to high-level semantic concepts was addressed by focusing on inter-query learning in content-based image retrieval (CBIR) with both global and region-based image representations. While the focus was on inter-query learning, novel intra-query learning approaches as well as a novel image representation and similarity measure were also proposed.

We presented two novel techniques for performing inter-query learning with global image representations. Both techniques use support vector machines (SVM) for learning the class distributions of users' high-level query concepts from retrieval experience. They are based on a relevance feedback (RF) framework that learns one-class support vector machines (1SVM) from retrieval experience to represent the set memberships of users' high-level query concepts and stores them in a "concept database". The "concept database" provides a mechanism for accumulating inter-query learning obtained from previous queries. The geometric view of 1SVMs allows a straightforward interpretation of the density of past interaction in a local area of the feature space and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

The first approach does a fuzzy classification of a new query into the regions of

support represented by the 1SVMs in the “concept database”. In this way, past experience is merged with current intra-query learning. The second approach incorporates inter-query learning into the query modification and distance reweighing framework. One of the main advantages of these approaches is the capability of making an intelligent initial guess on a new query when the query is first presented to the system.

We demonstrated the superior performance of the proposed approaches over other methods and confirmed that image retrieval performance can be improved by the integration of inter-query learning. Furthermore, performance increases in the initial retrieval set where a traditional intra-query-learning-only approach would require at least one iteration of RF to provide some improvement. Thus, user interaction can be reduced by decreasing the number of iterations that are needed to satisfy a query. We plan to investigate the possibility of using a machine learning approach such as artificial neural networks or reinforcement learning to have a more principled way of exploiting intra and inter-query learning that adapts to the current situation.

We also presented two novel intra-query learning approaches for CBIR with region-based image representations. The first method, probabilistic region relevance learning (PRRL), is based on the observation that regions in an image have unequal importance for computing image similarity. It automatically estimates region relevance based on user’s feedback. It can be used to set region weights in region-based image retrieval frameworks that use an overall image-to-image similarity measure. Currently, PRRL only performs intra-query learning. That is, for each given query, the user’s feedback is used to learn the relevance of the regions in the query and the learning process starts from ground up for each new query. However, it is also possible to exploit inter-query learning to enhance the retrieval performance of future queries. Thus, for a new query, instead of starting the learning process from ground up, we could exploit the previously learned region importances of similar queries. This would be very beneficial specially in the initial retrieval set since, instead of using uniform

weighting or some other weighting heuristic, we could make a more informed initial estimate of the relevance of regions in the new query. We plan to investigate the possibility of incorporating inter-query learning into the PRRL framework as part of our future work.

The second approach is based on SVM learning. Traditional approaches based on SVM learning require the use of fixed-length image representations because SVM kernels represent an inner product in a feature space that is a non-linear transformation of the input space. However, many CBIR methods that use region-based image representations create a variable-length image representation and define an arbitrary similarity measure between two variable-length representations. Thus, the standard SVM approach cannot be applied because the similarity measure may violate the requirements that a SVM places on the kernel. Fortunately, a generalized SVM has been developed that allows the use of an arbitrary kernel. We presented a learning algorithm based on generalized support vector machines (GSVM). Since a GSVM does not place restrictions on the kernel, any image similarity measure can be used. The experimental results on general-purpose images show convincingly the efficacy of the proposed method in improving image retrieval performance. Currently, for each query, the user's RF is used as training data and the learning process starts from ground up for each new query. However, it is also possible to exploit the long term learning accumulated over the course of many query sessions. This would be very beneficial specially in the initial retrieval set since, instead of ranking images based only on the similarity measure, we could make a more informed initial estimate of the relevance of images to the user's query concept. We plan to investigate the possibility of incorporating long-term learning into this GSVM-based learning framework as part of our future work.

A generic intra/inter-query learning approach that addresses the problem of semantically meaningful image segmentation was also proposed. A large number of

image segmentation techniques have been proposed in the literature. However, most image segmentation algorithms create regions that are homogeneous with respect to one or more low-level features according to some similarity measure. Unfortunately, homogeneous regions based on low-level features usually do not correspond to meaningful objects. We proposed an algorithm based on multiple-instance learning (MIL) that exploits both intra and inter-query learning for automatically improving the segmentation of images in a database. The main advantage of this approach is that it can automatically refine the segmentation of images into semantically-meaningful objects.

Finally, we presented an initial investigation into what we believe is the logical continuation of the CBIR idea of searching visual content directly. It is based on the observation that, since ultimately, the entire visual content of an image is encoded into its raw data (i.e., the raw pixel values), in theory, it should be possible to determine image similarity based on the raw data alone. We presented an initial investigation into an image dissimilarity measure following from the theoretical foundation of the recently proposed normalized information distance (NID). A very crude approximation of the Kolmogorov complexity of an image was created by compression. Using this approximation, we calculated the NID between images and used it as a metric for CBIR. The compression-based approximation to Kolmogorov complexity, though very rough, was shown to be valid by proving that it creates a statistically significant dissimilarity measure by testing it against a null hypothesis of random retrieval. Although the approximations used in this initial investigation may not currently be practical for CBIR, the results are encouraging that additional research into methods guided by the NID approach may be fruitful. We plan on exploring other methods of concatenating images and of compressing images, such as fractal and wavelet compression, that may better exploit the 2D nature of images. Another area where it may be useful to try the NID approach is in the matching of variable-length feature

vectors. The NID approach may create a very practical method that goes beyond the individual region matching but does not require the expense of determining the higher level relationships among the regions. Another area of future research is the exploration of the NID approach as a feature-independent method of structuring an image data set.

Bibliography

- [1] University of Washington, groundtruth image database.
<http://www.cs.washington.edu/research/imagedatabase/groundtruth>.
- [2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 15, pages 561–568, 2003.
- [3] M. Balabanovic. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction*, 8(1-2):71–102, 1998.
- [4] R. Bayer and E. M. McCreight. Organization and maintenance of large ordered indices. *Acta Informatica*, 1:173–189, 1972.
- [5] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, New Jersey, 1961.
- [6] A. B. Benitez. Using relevance feedback in content-based image metasearch. *IEEE Internet Computing*, 2(4):59–69, July 1998.
- [7] C. Bennett, P. Gács, M. Li, P. Vitányi, and W. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.
- [8] C. Bennett, M. Li, and B. Ma. Linking chain letters. *Scientific American*, (76-81), June 2003.

- [9] A. Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, San Francisco, California, 1999.
- [10] C. Bishop. Novelty detection and neural network validation. *IEEE Proceedings on Vision, Image, and Signal Processing*, 141(4):217–222, 1994.
- [11] C. Buckley and G. Salton. Optimization of relevance feedback weights. In *Proceedings of SIGIR*, pages 351–357, 1995.
- [12] C. Burges. Simplified support vector decision rules. In *Proceedings of the 13th International Conference on Machine Learning*, pages 71–77, 1996.
- [13] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998.
- [14] P. Campadelli, D. Medici, and R. Schettini. Color image segmentation using Hopfield networks. *Image and Vision Computing*, 15(3):161–166, 1997.
- [15] C. Carson. Blobworld: Image segmentation using expectation-maximization and its applications to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.
- [16] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [17] Y. Chen and J. Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1252–1267, 2002.
- [18] Y. Chen, X. Zhou, and T. Huang. One-class SVM for learning in image retrieval. In *Proceedings of IEEE International Conference on Image Processing*, pages 34–37, 2001.

- [19] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd International Conference on Very Large Databases*, pages 426–435, 1997.
- [20] J. Cox, M. L. Miller, T. P. Minka, and P. N. Yianilos. An optimized interaction strategy for Bayesian relevance feedback. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–558, 1998.
- [21] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.
- [22] I. Daubechies. *Ten Lectures on Wavelets*. Capital City Press, 1992.
- [23] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society*, 39(1):1–38, 1977.
- [25] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.
- [26] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, New York, NY, 2001.
- [27] C. Faloutsos, M. Flicker, W. Niblack, D. Petkovic, W. Equitz, and R. Barber. Efficient and effective querying by image content. Technical report, IBM, 1993.

- [28] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. The QBIC project: querying images by content using color, texture, and shape. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, pages 173–181, 1993.
- [29] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer graphics*. Addison Wesley, 1990.
- [30] J. Fournier and M. Cord. Long-term similarity learning in content-based image retrieval. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages 441–444, 2002.
- [31] J. Friedman. Flexible metric nearest neighbor classification. Technical report, Stanford University, Department of Statistics, 1994.
- [32] P. Gács. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, 15:1477–1480, 1974.
- [33] A. Gammernan and V. Vovk. Kolmogorov complexity: Sources, theory, and applications. *The Computer Journal*, 42(4):252–255, 1999.
- [34] A. Gersho. Asymptotically optimum block quantization. *IEEE Transactions on Information Theory*, IT-25(4):231–262, July 1979.
- [35] I. Gondra and D. R. Heisterkamp. Adaptive and efficient image retrieval with one-class support vector machines for inter-query learning. *WSEAS Transactions on Circuits and Systems*, 3(2):324–329, April 2004.
- [36] I. Gondra and D. R. Heisterkamp. Improving image retrieval performance by inter-query learning with one-class support vector machines. *Neural Computing and Applications*, 13(2):130–139, June 2004.

- [37] I. Gondra and D. R. Heisterkamp. Learning in region-based image retrieval with generalized support vector machines. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2004.
- [38] I. Gondra and D. R. Heisterkamp. Probabilistic region relevance learning for content-based image retrieval. In *Proceedings of International Conference on Imaging Science, Systems, and Technology*, pages 434–440, 2004.
- [39] I. Gondra and D. R. Heisterkamp. Semantic similarity for adaptive exploitation of inter-query learning. In *Proceedings of International Conference on Computing, Communications, and Control Technologies*, volume 1, pages 142–147, 2004.
- [40] I. Gondra and D. R. Heisterkamp. Summarizing inter-query knowledge in content-based image retrieval via incremental semantic clustering. In *Proceedings of IEEE International Conference on Information Technology*, volume 2, pages 18–22, 2004.
- [41] I. Gondra and D. R. Heisterkamp. A Kolmogorov complexity-based normalized information distance for image retrieval. In *Proceedings of International Conference on Imaging Science, Systems, and Technology: Computer Graphics*, 2005.
- [42] I. Gondra, D. R. Heisterkamp, and J. Peng. Improving the initial image retrieval set by inter-query learning with one-class support vector machines. In *Proceedings of International Conference on Intelligent Systems Design and Applications*, pages 393–402, 2003.
- [43] Y. Gong. *Intelligent Image Databases: Towards Advanced Image Retrieval*. Kluwer Academic Publishers, Norwell, MA, 1998.

- [44] A. Graf, A. Smola, and S. Borer. Classification in a normalized feature space using support vector machines. *IEEE Transactions on Neural Networks*, 14(3):597–605, May 2003.
- [45] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, May 1997.
- [46] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.
- [47] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1-2):113–133, October 1996.
- [48] gzip. GNU zip compression utility. <http://www.gzip.org>.
- [49] Y. Hara, K. Hirata, H. Takano, and S. Kawasaki. Hypermedia navigation and content-based retrieval for distributed multimedia databases. In *Proceedings of the 6th NEC Research Symposium on Multimedia Computing*, 1995.
- [50] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.
- [51] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1985.
- [52] X. He, O. King, W. Ma, M. Li, and H. Zhang. Learning a semantic space from user’s relevance feedback for image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):39–48, 2003.

- [53] D. R. Heisterkamp. Feature relevance learning with query shifting for content-based image retrieval. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 4, pages 250–253, 2000.
- [54] D. R. Heisterkamp. Building a latent semantic index of an image database from patterns of relevance feedback. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 132–135, August 2002.
- [55] D. R. Heisterkamp, J. Peng, and H. Dai. Adaptive quasiconformal kernel metric for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 288–393, 2001.
- [56] P. Hong, Q. Tian, and T. Huang. Incorporate support vector machines to content-based image retrieval with relevance feedback. In *Proceedings of IEEE International Conference on Image Processing*, pages 750–753, 2000.
- [57] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. John Wiley and Sons, New York, NY, 1999.
- [58] C. L. Huang. Parallel image segmentation using modified Hopfield model. *Pattern Recognition Letters*, 13(5):345–353, 1992.
- [59] IAPR. IAPR’s technical committee 12: Multimedia and visual information systems, benchmarking for visual information retrieval, <http://sci.vu.edu.au/clement/tc-12/benchmark>.
- [60] J. Ingemar and J. Cox. The Bayesian image retrieval system, PicHunter, theory, implementation, and psychological experiments. *IEEE Transactions on Image Processing*, 9:20–37, 2000.
- [61] Y. Ishikawa, R. Subramanys, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 433–438, 1998.

- [62] H. Iwata and H. Hagahashi. Active region segmentation of color images using neural networks. *Systems and Computers in Japan*, 29(4):1–10, 1998.
- [63] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.
- [64] N. Japkowicz. *Concept learning in the absence of counter-examples: an autoassociation-based approach to classification*. PhD thesis, The State University of New Jersey, 1999.
- [65] F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang. Learning in region-based image retrieval. In *Proceedings of 2nd International Conference on Image and Video Retrieval*, pages 206–215, 2003.
- [66] I. T. Jolliffe. *Principal component analysis*. Springer, New York, NY, 1986.
- [67] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the CANDID approach. In *SPIE Storage and Retrieval for Image and Video Databases*, volume 2420, pages 238–248, 1995.
- [68] A. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, 14:662–664, 1968.
- [69] M. Koskela and J. Laaksonen. Using long-term learning to improve efficiency of content-based image retrieval. In *Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems*, pages 72–79, 2003.
- [70] J. Kwok and I. Tsang. Finding the pre-images in kernel principal component analysis. In *NIPS Workshop on Kernel Machines*, 2002.
- [71] J. Laaksonen, M. Koskela, and E. Oja. Picsom: self-organizing maps for content-based image retrieval. In *Proceedings of International Joint Conference on Neural Networks*, volume 4, pages 2470–2473, 1999.

- [72] C. Lee, W. Y. Ma, and H. J. Zhang. Information embedding based on user's relevance feedback for image retrieval. In *Proceedings of SPIE International Conference on Multimedia Storage and Archiving Systems*, volume 4, pages 19–22, 1999.
- [73] J. Li, J. Wang, and G. Wiederhold. IRM: Integrated region matching for image retrieval. In *Proceedings of the 8th ACM International Conference on Multimedia*, pages 147–156, 2000.
- [74] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 863–872, 2003.
- [75] M. Li, Z. Chen, and H. Zhang. Statistical correlation analysis in image retrieval. *Pattern Recognition*, 35(12):2687–2693, December 2002.
- [76] M. Li and P. Vitányi. Reversibility and adiabatic computation: trading time and space for energy. *Proc. Royal Society of London, Series A*, 452:769–789, 1996.
- [77] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, New York, NY, 1997.
- [78] E. Littmann and H. Ritter. Adaptive color segmentation: a comparison of neural and statistical methods. *IEEE Transactions on Neural Networks*, 8(1):175–185, 1997.
- [79] L. Lucchese and S. K. Mitra. Advances in color image segmentation. In *Global Telecommunication Conference*, pages 2038–2044, 1999.
- [80] Swain. M. and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

- [81] W. Ma and B. Majunath. Netra: A toolbox for navigating large image databases. In *Proceedings of IEEE International Conference on Image Processing*, pages 568–571, 1997.
- [82] S. D. MacArthur, C. E. Bradley, and C. R. Shyu. Relevance feedback decision trees in content-based image retrieval. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 68–72, 2000.
- [83] O. Mangasarian. *Nonlinear Programming*. McGraw-Hill, New York, NY, 1969.
- [84] O. Mangasarian. Generalized support vector machines. In *A. Smola et al. (eds.), Advances in Large Margin Classifiers*, pages 135–146. MIT Press, 2000.
- [85] O. Maron. *Learning from Ambiguity*. PhD thesis, Massachusetts Institute of Technology, 1998.
- [86] O. Maron and A. Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, volume 15, pages 341–349, 1998.
- [87] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, 10:570–576, 1997.
- [88] D. Martin. *An empirical approach to grouping and segmentation*. PhD thesis, University of California, Berkeley, 2002.
- [89] D. Martin, C. Fowlkes, D. Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation and measuring ecological statistics. In *Proceedings of IEEE International Conference on Computer Vision*, pages 416–425, 2001.

- [90] J. McQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [91] S. Mehrotra, Y. Rui, M. Ortega, and T. Huang. Supporting content-based queries over images in MARS. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 632–633, 1997.
- [92] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London, A*(209):415–446, 1909.
- [93] C. Merz and P. Murphy. UCI repository of machine learning databases. <http://www.ics.uci.edu/mllearn/MLRepository.html>.
- [94] Y. Meyer. *Wavelets Algorithms and Applications*. SIAM, Philadelphia, 1993.
- [95] S. Mika, B. Schölkopf, A. Smola, K. Muller, M. Scholz, and G. Ratsch. Kernel PCA and denoising in feature spaces. *Advances in Neural Information Processing Systems*, 11:536–542, 1999.
- [96] T. Minka and R. Picard. Interactive learning using a society of models. *Pattern Recognition*, 30(4):565–581, April 1997.
- [97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [98] H. Muller, W. Muller, D. Squire, S. Marchand-Maillet, and T. Pun. Long-term learning from user behavior in content-based image retrieval. Technical Report 00.04, University of Geneva, Geneva, Switzerland, 2000.
- [99] M. Oberhumer. UCL compression library, version 1.02. <http://www.oberhumer.com/opensource/ucl>.

- [100] A. Ono, M. Amano, M. Hakaridani, T. Satoh, and M. Sakauchi. A flexible content-based image retrieval system with combined scene description keywords. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 201–208, 1996.
- [101] N. R. Pal and S. K. Pal. A review of image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
- [102] J. Peng, B. Banerjee, and D. R. Heisterkamp. Kernel index for relevance feedback retrieval in large image databases. In *Proceedings of the 9th International Conference on Neural Information Processing*, pages 187–191, 2002.
- [103] J. Peng, B. Bhanu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, 75(1/2):150–164, 1999.
- [104] W. B. Pennebaker and J. L. Mitchell. *The JPEG still image data compression standard*. Van Nostrand Reinhold, New York, 1993.
- [105] A. Pentland, R. Picard, and S. Sclaroff. PhotoBOOK: Tools for content-based manipulation of image databases. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, volume 2, pages 34–47, 1994.
- [106] E. Persoon and K. S. Fu. Shape discrimination using Fourier description. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3):170–179, 1977.
- [107] D. L. Pham and J. L. Prince. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. *Pattern Recognition Letters*, 20(1):57–68, 1999.
- [108] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell. MIT media lab: Vision texture database. <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/>.

- [109] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The art of scientific computing*. Cambridge University Press, New York, 2nd edition, 2002.
- [110] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *Graphical Models and Image Processing*, 54(5):438–460, 1992.
- [111] S. Ravela, R. Manmatha, and E. M. Riseman. Scale-space matching and image retrieval. In *Proceedings of the Image Understanding Workshop*, volume 2, pages 1199–1207, 1996.
- [112] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
- [113] G. Ritter and M. Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18:525–539, 1997.
- [114] J. Rocchio and G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance Feedback in Information Retrieval, pages 313–323. Prentice Hall, 1971.
- [115] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, (65), 386–408 1958.
- [116] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 59–66, 1998.
- [117] Y. Rui and T. Huang. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, September 1998.

- [118] Y. Rui, T. Huang, and S. Chang. Image retrieval: Past, present, and future. *Journal of Visual Communication and Image Representation*, 10:1–23, 1999.
- [119] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of IEEE International Conference on Image Processing*, pages 815–818, 1997.
- [120] Y. Rui, A. C. She, and T. S. Huang. Modified Fourier descriptors for shape representation - a practical approach. In *First International Workshop on Image Databases and Multimedia Search*, 1996.
- [121] G. Salton. Another look at automatic text-retrieval systems. *Communications of the ACM*, 29(7):648–656, 1986.
- [122] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1998.
- [123] R. Samadani, C. Han, and L. K. Katragadda. Content-based event selection from satellite image of the aurora. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, pages 50–59, 1993.
- [124] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. Smola. Input versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.
- [125] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87, Microsoft Research, 1999.
- [126] S. Sclaroff, L. Taycher, and M. L. Cascia. ImageRover: a content-based image browser for the world-wide web. Technical Report 97-005, Boston University CS Dept., 1997.

- [127] W. M. Shaw. Term-relevance computations and perfect retrieval performance. *Information Processing and Management*, 31(4):491–498, 1995.
- [128] H. T. Shen, B. C. Ooi, and K. L. Tan. Giving meanings to WWW images. In *Proceedings of ACM Multimedia*, pages 39–48, 2000.
- [129] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [130] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
- [131] A. R. Smith. Color gamut transformation pairs. *Computer Graphics*, (12):12–19, 1978.
- [132] J. Smith and S. Chang. VisualSEEk: a fully automated content-based image query system. In *Proceedings of ACM Multimedia*, pages 87–98, 1996.
- [133] J. Smith and S. Chang. An image and video search engine for the world-wide web. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, volume 5, pages 84–95, 1997.
- [134] J. R. Smith and C. S. Li. Image classification and querying using composite region templates. *Computer Vision and Image Understanding*, 75(1/2):165–174, 1999.
- [135] R. K. Srihari, Z. Zhang, and A. Rao. Intelligent indexing and semantic retrieval of multimodal documents. *Information Retrieval*, (2):245–275, 2000.
- [136] H. S. Stone and C. S. Li. Image matching by means of intensity and texture matching in the Fourier domain. In *Proceedings of SPIE Conference on Image and Video Databases*, pages 337–349, 1996.

- [137] S. Sull, J. Oh, S. Oh, S. Song, and S. Lee. Relevance graph-based image retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume 2, pages 713–716, 2000.
- [138] H. Tamura. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6), 1978.
- [139] D. Tax. *One-Class Classification*. PhD thesis, Delft University of Technology, Delft, The Netherlands, June 2001.
- [140] K. Tieu and P. Viola. Boosting image retrieval. In *Proceedings of IEEE Conference in Computer Vision and Pattern Recognition*, pages 228–235, 2000.
- [141] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 107–118, 2001.
- [142] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New York, NY, 1995.
- [143] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.
- [144] N. Vasconcelos and A. Lippman. Learning over multiple temporal scales in image databases. In *Proceedings of the 6th European Conference on Computer Vision*, volume 1, pages 33–47, 2000.
- [145] C. Wallace and D. Boulton. An information measure for classification. *Comput. J.*, 11:185–195, 1968.

- [146] J. Wang, G. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.
- [147] J. Wang, G. Wiederhold, O. Firschein, and X. Sha. Content-based image indexing and searching using Daubechies’ wavelets. *International Journal of Digital Libraries*, 1(4):311–328, 1998.
- [148] A. Webb. *Statistical Pattern Recognition*. John Wiley and Sons, LTD., Hoboken, NJ, 2nd edition, 2002.
- [149] L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. Semi-automatic image annotation. In *Proceedings of International Conference on Human-Computer Interaction*, volume 1, pages 326–334, 2001.
- [150] L. Williams and D. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and saliency. In *Proceedings of International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine*, pages 59–69, 1995.
- [151] P. Wolfe. A duality theorem for nonlinear programming. *Quarterly of Applied Mathematics*, 19:239–244, 1961.
- [152] P. Wu and B. S. Manjunath. Adaptive nearest neighbor search for relevance feedback in large image databases. In *Proceedings of 9th ACM Conference on Multimedia*, pages 89–97, 2001.
- [153] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.
- [154] C. Yang and T. Lozano-Pérez. Image database retrieval with multiple-instance learning techniques. In *Proceedings of IEEE International Conference on Data Engineering*, pages 233–243, 2000.

- [155] P. Yin, B. Bhanu, and K. Chang. Improving retrieval performance by long-term relevance information. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 533–536, 2002.
- [156] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, June 2002.
- [157] Q. Zhang and S. A. Goldman. EM-DD: an improved multiple-instance learning technique. *Neural Information Processing Systems*, 2001.
- [158] Q. Zhang, S. A. Goldman, W. Yu, and J. Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of the 19th International Conference on Machine Learning*, pages 682–689, 2002.
- [159] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.
- [160] Y. J. Zhang. A review of recent evaluation methods for image segmentation. In *International Symposium on Signal Processing and its Applications*, pages 13–16, 2001.
- [161] X. Zhou and T. Huang. Small sample learning during multimedia retrieval using BiasMap. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 11–17, 2001.

VITA



Iker Gondra

Candidate for the Degree of
Doctor of Philosophy

Thesis: INTER-QUERY LEARNING IN CONTENT-BASED IMAGE RETRIEVAL

Major Field: Computer Science

Biographical:

Personal Data: Born in Bilbao, Vizcaya, Spain, August 20, 1977, son of Maria Luisa Luja and José Enrique Gondra.

Education: Graduated from Sagrado Corazón High School, Sucre, Bolivia, in December 1994; received Bachelor of Science in Computer Science and Master of Science in Computer Science from Oklahoma State University, Stillwater, Oklahoma, US, in December 1998 and May 2002, respectively. Completed the requirements for the Doctor of Philosophy degree in Computer Science at Oklahoma State University in July 2005.

Experience: Employed by Oklahoma State University, Department of Computer Science as a graduate Teaching Assistant, August 1999 to December 2004; employed by the Department of Computer Science at St. Francis Xavier University, Antigonish, Nova Scotia, Canada, as an Assistant Professor, January 2005 to present.