Using Biased Support Vector Machine in Image Retrieval with Self-Organizing Map

Chan Chi Hang

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. Irwin King

Abstract of thesis entitled:

Using Biased Support Vector Machine in Image Retrieval with Self-Organizing Map

Submitted by Chan Chi Hang
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in August 2004

With the rapid growth in the volume of digit images, searching and browsing in a large collection of images is gaining importance. In a traditional image retrieval system, it uses text keywords or text descriptors for indexing and retrieval. However, the keyword-based image retrieval systems require large amount of manual effort to annotate the images in database, and it reduces the scalability of the system. Content-Based Image Retrieval (CBIR) has been proposed to overcome the difficulties of keyword-based image retrieval approach in early 1990's. In contrast to the keyword-based approach, CBIR uses the visual features, such as color, texture, and shape feature, for indexing and retrieval. Since the feature extraction process can be made automatic, this greatly reduces the difficulties of the keyword-based approach. However, it is difficult to use low-level image features to represent high-level image concepts, and the CBIR systems have a very limited recall even the best feature extraction and similarity measure algorithms are used.

In this thesis, we make use of the relevance feedback architecture to learn image similarity through interactions with users. The goal of relevance feedback is to learn user's preference from their interaction, and it is a powerful technique to improve the retrieval result in CBIR. In recent years, many intra-query learning techniques have been proposed to solve the relevance feedback problem, in which the prior information from past queries are ignored. Among these techniques, Support Vector Machines (SVM) have shown promising results in the area. More specifically, in relevance feedback applications the SVMs are typically been used as binary classifiers with the balanced input data assumption. In other words, they do not consider the imbalanced dataset problem in relevance feedback, i.e., the non-relevant examples outnumbered the relevant examples. In this thesis, we propose to apply our Biased Support Vector Machine (BSVM) to address this problem.

Moreover, we apply our Self-Organizing Map-based inter-query technique to reorganize the feature vector space, in order to incorporate the information provided by past queries and improve the retrieval performance for future queries. The proposed combined scheme is evaluated against real world data. Promising results demonstrating the effectiveness of our proposed approach.

中文摘要

由於電腦圖像數量的迅速增長，搜尋和瀏覽電腦圖像獲取了很大的重要性。傳統圖像檢索系統使用了關鍵字作爲分度法和檢索法。但是，基於關鍵字的圖像檢索系統要求很多的人力，並且它減低了系統的擴展性。在九十年代初期，研究員提議用影像擷取檢索(CBIR) 克服關鍵字圖像檢索方法困難。與關鍵字圖像檢索方法對比，影像擷取檢索利用圖像的視覺特性，譬如顏色、紋理，和形狀特性，作爲分度法和檢索。因爲特性提取過程可以自動進行，這大大地減少了關鍵字圖像檢索方法的困難。但是，這是很難使用低級圖像特性代表高級圖像概念，影像擷取檢索系統有非常有限的收回即使使用了最佳的算法。

在這份論文，我們利用相關性反饋由用戶上學會圖像間的相似性。相關性反饋的目標是學會用戶的喜好，並且這是一個強而有力的技術去改進影像擷取檢索系統。在過去幾年，研究員提議了許多相關性反饋的技術，但這些技術忽略了過去詢問中的資訊。在這些技術之中，支援向量機(SVM) 顯示了良好的結果。更加具體地說，在相關性反饋中應用傳統的支援向量機可被視用作爲平衡的二進制分類。換句話說，他們不考慮相關性反饋中的不平衡資料集問題。亦即，非相關的例子在數量上遠超過了相關的例子。在這份論文，我們提議應用我們的偏心支援向量機 (BSVM) 去解決這個問題。而且，我們使用自我組織映射圖網路 (SOM) 去學習過去詢問中的資訊和改進未來詢問的檢索表現。從使用現實世界資料的實驗中，我們提出的算法展示了良好的結果。

Acknowledgement

Here I wish to express my sincere gratitude to my supervisors Prof. Irwin King. He has given me excellent guidance from the inception of research directions to the approach of solving problems. The knowledge I acquired is not only beneficial to my research, but also to my future career. In addition, I would like to thank those who have helped, supported and influenced me, those who have given me chances to explore, those who were patient in waiting for my improvement and those who have understood and have backed me up psychologically.

The support from my colleagues is also indispensable. In particular, I am grateful to Richard Sia, Steven Choi and Kaizhu Huang, for sharing their knowledge and insightful discussions in my research. Last but not least, I would like to thank my fellow classmates, including Gordon Lam, Chi-Wai Leung, Johnson Hung, Chi-Wing Wong, Chi-Hung Law, Sam Ip, Eddie Ng, Vesta Lee, Pheobe Lau, Edith Ngai, Cheuk-Luk Fung and Kim-Fung Jang for their continuous warmth in the midst of my setbacks.

Lastly, I want to express my warm thanks to family. Their love and care are the energy of life.

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the rapid growth in the volume of digital images, searching and browsing in a large collection of images are gaining importance. In a traditional image retrieval system, it uses text keywords or text descriptors for indexing and retrieval. However, there are two main difficulties in keyword-based image retrieval [21, 46]; they are,

- **Differences in interpretation of image content**: There are always inconsistencies in keyword assignments, since different indexers may use different keywords to describe the same image concept. Moreover, the retrieval system that uses words to describe image concept suffers from two well-known language related problem called *synonymy* and *polysemy*. *Synonymy* describes that several words have the same meaning. *Polysemy* describes that the words have multiple meanings.

- **Non-scalability**: Large amount of manual effort is required to annotate the images in database. Since it is hard to extract the keywords from an image automatically, and many image retrieval systems that adopted the keyword-based approach need human to extract the keywords from images.

To overcome the difficulties of keyword-based image retrieval approach, Content-Based Image Retrieval (CBIR) has been proposed in early 1990's. In contrast to the keyword-based approach, CBIR uses the visual feature of images, such as color, texture, and shape feature, for indexing and retrieval. This greatly reduces the difficulties of the keyword-based approach, since the feature extraction process can be made automatic and the image's own content is always consistent. The CBIR process can be summarized as follows:

- **Feature Extraction**: Image processing and computer vision techniques are used to extract low-level visual features from images, color, texture, and shape for example. These features are usually represented by high-dimensional vectors in the real domain.
- **Retrieval**: For a given feature, a notation of similarity measure is determined. The similarity measure is used to rank the images in the collection.

Despite the extensive research effort, the retrieval techniques used in CBIR systems have a very limited recall even when the best feature extraction and similarity measure algorithms are used. That is only a very limited relevant items are retrieved to the user in response to the initial query. This problem is recognized as a major difficulty in information retrieval [27, 40]. There are two major reasons that lead to this problem [61]; they are,

- **The gap between high-level concepts and low-level features**: In a traditional CBIR system, it assumes that the mapping from low-level features to high-level concepts is easy for the user to do. However, this assumption may not be true. One example is to map a picture of a smiling old man to low-level feature representation.

- **Subjectivity of human perception**: For a particular image, different users or the same user under different circumstances may perceive differently. Thus, it is almost impossible to find a feature extraction or similarity measure algorithm to satisfy all situations.

In light of this, researchers figure out that refinement of the query and similarity measurement during the retrieval process are required to further improve the retrieval performance.

Relevance feedback is suggested as a solution for the problem of user subjectivity. The goal of relevance feedback is to learn user's preference from their interaction, and it is a powerful technique to improve the retrieval result in CBIR. Under this framework, a set of images is presented to the user according to the query. The user marks those images as either relevant or non-relevant and then feeds back this information into the system. Based on this feedback information, the system presents another set of images to the user. The system learns user's preference through this iterative process and improves the retrieval performance. From the experimental results of various CBIR systems, it shows that relevant feedback is a promising direction for CBIR.

1.1 Problem Statement

Most of the current relevance feedback systems are based on the intra-query learning approach [9, 20, 26, 61, 75]. In this approach, the system refines the query and improves the retrieval result by using the feedback information provided by the user. The learning process starts from ground up for each query, and the prior experiences from past queries are ignored. In the intra-query approach, the system presents a set of images, X_t, to the user in each iteration, and the user gives feedback, A_t, to the system based on these images. Thus, the system learns the user's preference from these feedback information, and the information provided to the system up to the t-th iteration can be represented as $H = \{X_1, A_1, X_2, A_2, \cdots, X_t, A_t\}$. Among the intra-query learning techniques, recent research shows that SVM-based techniques are more promising and effective techniques than other intra-query approaches [9, 82]. The regular SVM [5, 78] and one-class SVM [51, 64, 65, 72, 73] are introduced into the relevance feedback problem. The regular SVM-based technique [20, 26, 75, 76] treats the relevance feedback problem as a strict binary classification problem. However, this technique does not consider the imbalanced dataset problem, in which the number of non-relevant images is significantly larger than the relevant images. This imbalanced dataset problem will lead the positive data (relevant images) be overwhelmed by the negative data (non-relevant images) [9]. The one-class SVM-based technique [9] uses only the relevant images in the learning process, and treats the problem as a density estimation problem. The one-class SVM-based technique seems to avoid the imbalanced dataset problem. However, it cannot work well without the help of negative information [82].

Recently, researchers propose the use of inter-query information to further improve the retrieval result in relevance feedback process [24, 28, 42, 83]. In the inter-query learning approach, feedback information from past queries are accumulated to train the system to determine what images are of the same semantic meaning. Let us assume that the system has processed k queries before. For the $(k+1)$-th query, the information provided to the system is $\{H_1, H_2, \cdots, H_{k+1}\}$ instead of H_{k+1} alone. Thus, the inter-query approach has more information to learn the user's preferences. In [28, 42, 83], the system analyzes the correlation between images labelled in the past queries. The inter-query information is used to improve similarity measure in the retrieval process. In [24], the inter-query information is used to capture users' query concepts. The image concepts are then used to select the set of images presented to the user, and improve the retrieval result. These approaches examined the possibility of incorporating the inter-query information to the relevance feedback process. They show that the retrieval performance can be benefited from the inter-query learning.

The problem we are facing are:

- To develop a relevance feedback system that has the advantages of the existing SVM-based relevance feedback techniques, and able to address the imbalanced dataset problem.
- To incorporate the inter-query information in the system, so as to improve the retrieval result and reduce the number of iteration required.

1.2 Major Contributions

The main contributions of our work are as follows:

- We propose a Biased Support Vector Machine (BSVM) [6, 31] technique to capture the user's individual preferences in the relevance feedback process. Moreover, BSVM addresses the imbalanced dataset problem in relevance feedback process. Our strategy is to construct a SVM that

  - classifies the positive data (relevant images) and negative data (non-relevant images) correctly, and
  - contains a parameter to control the importance of positive data and negative data.

  Thus, the positive data will not be overwhelmed by the negative data.

- We propose a Self-Organizing Map (SOM)-based technique [6, 7] to incorporate the inter-query information in the system. We use a SOM to represent the images in the database, and use the inter-query information to modify the feature vector space, in which the SOM of images is stored. This allows for transforming the images distributions and improving their organization in the modified vector space. Thus, the images are organized in a fashion that ease the retrieval process.

Our experimental results show that:

- BSVM produces better retrieval performance than regular SVM, one-class SVM and other techniques in the literature in the relevance feedback problem.
- the retrieval performance of BSVM can be further improved by applying the SOM-based inter-query learning.

1.3 Publication List

- **Chi-Hang Chan**, Ka-Cheung Sia, and Irwin King. Utilizing inter- and intra-query relevance feedback for content-based image retrieval. In Special Session of the International Conference on

Neural Information Processing (ICONIP2003), Istanbul, Turkey, May 2003.

- Chu-Hong Hoi, **Chi-Hang Chan**, Kaizhu Huang, Michael Lyu, and Irwin King. Biased support vector machine for relevance feedback in image retrieval. In The Proceedings to the 2004 International Joint Conference on Neural Networks, Budapest, Hungary, July 25-29, Accepted. IEEE Computer Society.

- **Chi-Hang Chan** and Irwin King. Using Biased Support Vector Machine to Improve Retrieval Result in Image Retrieval with Self-Organizing Map. In Proceedings to the International Conference on Neural Information Processing (ICONIP2004), Calcutta, India, November, 2004, Accepted.

1.4 Thesis Organization

In this thesis, we review current techniques in the literature in document retrieval as well as content-based image retrieval in Chapter 2. In particular, we discuss the major characteristic and properties of relevance feedback problem. We also analyze a variety of relevance feedback algorithms to point out the current direction of relevance feedback research. In Chapter 3, we propose the BSVM for relevance feedback in CBIR. The formulation and properties of BSVM are discussed in this chapter. In Chapter 4, we present our SOM-based inter-query learning algorithm. Experiments using synthetic and real data are shown in both Chapter 3 and Chapter 4 to illustrate the characteristics and performance of our algorithm. Lastly, we give concluding remarks in Chapter 5.

□ **End of chapter.**

Chapter 2

Background Survey

2.1 Relevance Feedback Framework

The relevance feedback process is an automatic and iterative process to improve retrieval result using the feedback information given by the users. In the relevance feedback framework, the system automatically selects and presents a set of objects, documents or images for example, to the user. The user then provides feedback to the system based on the degree of relevance between each presented object and his desired target. With the feedback information, the system captures the user's preferences and improves the retrieval performance. A typical relevance feedback framework consists of two major steps,

1. the system selects a set of objects from the database and presents to the user, and
2. the system captures the user's preferences and refines the query based on the feedback information given by the user.

This two steps repeat iteratively until the process is terminated. We use Fig. 2.1 to illustrate this framework. Various approaches have been proposed to optimize these two steps in recent years, we will further discuss about the query learning in relevance feedback in section 2.3, and the presentation set selection in section 2.4.

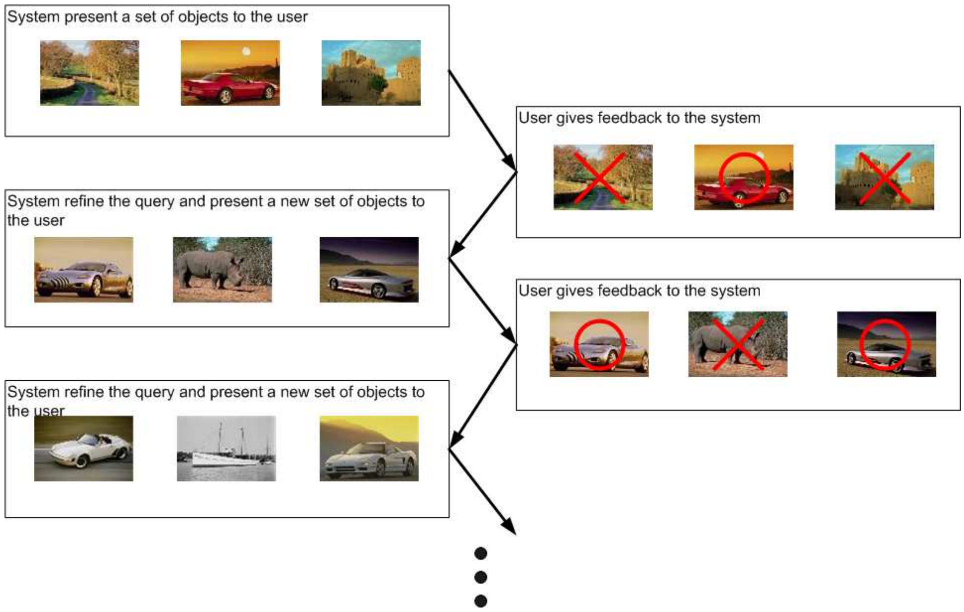


Figure 2.1: Relevance Feedback Framework

The relevance feedback problem can be considered as either a classification problem or a density estimation problem. Distinguishing the relevant and non-relevant objects in a collection is a common objective in a relevance feedback system, and this objective can be treated as a classification problem. Since it can be considered as classifying the data into two classes (relevant and non-relevant) based on a limited set of labelled data (objects identified by the user). Another common objective in relevance feedback is to rank the objects in the collection based on their degree of relevance to the query target, and it can be treated as a density estimation problem. Since the ranking function can be considered as the distribution of relevant objects and non-relevant objects, and the task of estimating these two distributions from the labelled data is a density estimation problem.

In the rest of this section, the major aspects and properties for relevance feedback are discussed.

2.1.1 Relevance Feedback Types

Different relevance feedback algorithms assume the user gives different types of feedback for the presented objects. There are mainly three different assumptions on the feedback types for the current relevance feedback algorithms; they are “relevant only”, “relevant score”, and “binary feedback”.

Relevant only ($\{O_R\}$) - It has been applied in the early relevance feedback systems [32, 53, 54, 85], in which only objects labelled as relevant are used in the learning process. The major drawback is the information contained in the non-relevant objects is ignored, since the users are looking for a small portion of objects in the collection, and a large portion of objects are expected to be non-relevant. Thus, there is a high chance for a presented object to be non-relevant, and a large amount of information is ignored if the non-relevant objects are not used in the learning process.

Relevant score ($\{\{O_R, O_N\}, S\}$) - It has been applied in various relevance feedback algorithms [34, 61], in order to obtain more information from the user. The relevant score can be discrete or continuous, and it indicates the degree of relevance of the presented objects to the user’s target. However, it is a difficult task for the user to quantize the degree of relevance, and different users may have different interpretations for that. Thus, the “binary feedback” is proposed to overcome this problem.

Binary feedback ($\{O_R, O_N\}$) - It is a common scheme adopted by the recent relevance feedback systems [9, 20, 25, 28, 29, 75], it assumes that the user labels the presented objects as either relevant or non-relevant. The advantage of this assumption is that it can make use of both relevant and non-relevant objects in the learning process, and the workload for the user is low when compare with the relevant score scheme.

2.1.2 Data Distribution

The data distribution can be separated into the feature distribution and the target distribution. The feature distribution refers to the distribution among all the objects in the collection in a particular feature vector space. The target distribution refers to the distribution of the desired target in a query in the feature vector space, and it is a major concern in most of the relevance feedback algorithms. The assumption on data distribution is an important issue in relevance feedback, since it is a crucial prior knowledge for estimating the target distribution.

Feature Distribution

The feature distributions of images and documents databases are usually sparse and highly clustered. This is due to the fact that feature representations have high dimensionality, and similar objects are located near to each others in the feature vector space. For the document databases, the concept of distribution is seldom applied. It is because the similarity measure used in the document retrieval is not a distance measure between feature vectors, but the cosine between the feature vectors. In general, similar documents are assumed to be clustered together on the surface of a unit sphere when the feature vectors are normalized. For the image databases, it is a common assumption that similar images follow a Gaussian distribution, and a Gaussian Mixture

Model (GMM) is used to model the feature distribution of the image database.

Target Distribution

In document retrieval systems, a general assumption for the target distribution is that those relevant objects are clustered together. Document retrieval approaches in the vector space model [32, 54, 58] define the similarity measure as the cosine between the term vectors of documents, and the document with the highest similarity measure to the query is considered as the query's target. Thus, the relevant documents are assumed to be clustered among the query vector in the vector space model. For the approaches in the probabilistic model [17, 36, 53, 85], they assume that if two documents with more presence or absence of search terms in common (similar term vector pattern), then they are considered more similar to each others. In the retrieval process for relevance feedback in the probabilistic model, documents with search term pattern similar to that of the labelled relevant documents are more likely to be the query's target. Thus, the probabilistic model also assumes that relevant documents are clustered together.

In CBIR systems, the Gaussian distribution is a common and convenient assumption for the target distribution. The single isotopic Gaussian assumption is adopted by the earliest relevance feedback systems [11, 61] for CBIR, and the components of the feature vectors are considered to be independent in these approaches. MindReader [34] proposes the use of the general Gaussian distribution as the target distribution, in order to consider the correlations among the components of feature vectors in the similarity measure. In [44], it does not restrict the target distribution to follow any particular class of statistical distribution, and uses a Parzen window estimation to model the target distribution. Recently, the support vector machine [20, 26, 75, 76] is applied in relevance feedback, where the problem of relevance feedback is modelled as a two-class classification problem, and the non-linearity of the target distribution is addressed in it.

2.1.3 Training Set Size

The size of the training set in the relevance feedback problem is usually small compared to the dimension of data, since each training sample requires user's annotation, and the number of user's annotation is considered as the user's workload. The size of the training set can be divided into two parts; they are the number of iteration and the number of sample in each iteration. If the total number of samples is constrained, then more iteration can provide better retrieval results in most cases. The intuition behind is that more iteration can give more opportunities for the system to refine the query. When the number of iteration is reduced to one, the system becomes a traditional one-shot retrieval system.

The limited size of the training set is the major concern in a relevance feedback system, since most of the classical learning techniques for the classification and the density estimation problem are based on the law of large number, and the target estimator tends to optimal when the number of training set is sufficiently large. Various relevance feedback systems apply these classical techniques on the problem. However, the size of training set is usually small, and it may not be sufficient for these systems to provide stable or meaningful results.

2.1.4 Inter-Query Learning and Intra-Query Learning

The learning process can be classified as inter-query learning or intra-query learning. Most of the current relevance feedback systems are based on the intra-query learning approach [9, 20, 26, 61, 75]. In this approach, the system refines the query and improves the retrieval result by using the feedback information that the user provided. The learning process starts from ground up for each query, and the prior experience from past queries are ignored. In the intra-query approach, the system presents a set of objects, X_t, to the user in each iteration, and the user gives feedback, A_t, to the system based on these objects. Thus, the system learns the user's preference from these feedback information, and the information provided to the system up to the t-th iteration can be represented as $H = \{X_1, A_1, X_2, A_2, \cdots, X_t, A_t\}$.

Recently, researchers propose the use of the inter-query information to further improve the retrieval result [7, 24, 28, 42, 83]. In the inter-query learning approach, feedback information from past queries is accumulated to train the system in order to determine which images are of the same semantic meaning. Let us assume that the system has processed k queries before. For the $(k+1)$-th query, the information provided to the system is $\{H_1, H_2, \cdots, H_{k+1}\}$ instead of H_{k+1} alone. Thus, the inter-query approach has more information to learn the user's preferences. In [28, 29], the system applies latent semantic indexing (LSI) [19] in relevance feedback. LSI is a classical document retrieval algorithm. It analyzes the correlation of documents and terms in the database. In [42, 83], the system analyzes the correlation between objects labelled in past queries. The inter-query information is used to improve the similarity measure in the retrieval process. In [28, 29], previous feedback information are stored in the system to build the latent semantic index. Then, they treat the query result as a document and the object in the collection as a term. In [24], the inter-query information is used to capture users' query concepts. These concepts are then used to select the set of objects presented to the user, and improve the retrieval result. These approaches examined the possibility of incorporating the inter-query information to the relevance feedback process, and they show that the retrieval performance can be benefited from the inter-query learning.

2.2 History of Relevance Feedback Techniques

The concept of relevance feedback is first introduced by Rocchio and Salton [56] to the document retrieval in the early 1960's. Rocchio et al. [54] argue that the initial query in the document retrieval may not be able to represent the user's need, so that they define the optimal query as the one that maximizing the similarity measure of the relevant documents and minimizing the similarity measure of the non-relevant documents to the query in the vector space model. The relevance feedback technique is then applied to estimate the optimal query, and these techniques are referred as the vector space model (Secion 2.3.1). Salton et al. [85] analyze this problem with a probabilistic model (Section 2.3.4). The statistical information of the query is gathered during the relevance feedback process, and the information is used to estimate the probability function of a document belonging to the relevant and the non-relevant set. These probability functions are then used to determine which documents are more similar to the query in the retrieval process.

Relevance feedback is introduced to image retrieval during mid 1990's [39, 45, 49, 61]. In a typical CBIR system, images are represented as data points by extracting the image features by some feature extraction methods, color moment and co-occurrence matrix for example. Then the nearest neighbor to the query in the vector space is considered as its target. Some earlier works in relevance feedback for CBIR is aimed at using the feedback information to modify the distance function of the image representations in the vector space. MARS [61] is one of the earliest systems that applied this idea. They construct the distance function as the weighed combination of the feature components, and the feedback information is used to update the weights of the distance function. One common weight updating method is to assign the weight of the feature component inverse proportional to the standard deviation of the relevant data. Since feature components with smaller standard deviations should be more important than those with larger standard deviations. These methods are referred as the ad-hoc re-weighting approach (Secion 2.3.2) or the standard deviation approach.

PicHunter [14] is also among the earliest work in the field of relevance feedback for CBIR. It assumes that the user is looking for a single image instead of a category of images. PicHunter is analogous to the probability model in the sense that they both use the feedback information to gather the statistical information for the query. In PicHunter, Bayes' rule is applied to construct the probability function of the image being the query's target, then this function is used to retrieve images for the query based on the feedback information. These techniques are referred as the Bayesian approach (Section 2.3.5).

MindReader [34] argues that the ad-hoc re-weighting approach lacks an optimal claim, and it addresses this problem by providing an optimization function. It replaces the weighted distance function in the ad-hoc re-weighting approach by a quadratic distance, and the query is optimized when the distance of relevant images to the query is minimized. The feedback information is used to obtain the mean vector and covariance matrix in the quadratic distance function. Thus, the MindReader has an optimality claim by modelling the problem as a minimization problem. These techniques are referred as the distance optimization approach (Section 2.3.3).

The density estimation approach (Section 2.3.6) is a combination of the probabilistic model and the distance optimization approach. In the density estimation approach, the relevance feedback system uses either parametric or non-parametric approach to estimate the distribution of the target. And the feedback information is used to obtain the parameters in the distribution function, since the probability function in the probabilistic model can be considered as a distribution, and the quadratic distance in the distance optimization can be replaced by a Gaussian distribution. Thus, the density estimation approach can be viewed as a generalization of these two approaches. In [44], they apply the non-parametric density estimation method, and model the target distribution with the Parzen window estimation. For the parametric approach [48, 71], Gaussian distribution is a common assumption for the distribution.

Recently, researchers apply the support vector machine (Section 2.3.7) in the statistical learning theory to the relevance feedback problem. SVM is a classification and regression technique with strong theoretical foundation and good generalization ability, and it provides good experimental results in many different domains. The regular SVM is used to solve the two-class classification problem. In [20, 25, 26, 75], they apply the regular SVM in relevance feedback problem by treating the relevant and non-relevant data as two separate classes. In [9], it applies the one-class SVM in relevance feedback, and treats the relevance feedback problem as a density estimation problem.

2.3 Relevance Feedback Approaches

In this section, various major approaches are discussed, including the relevance feedback systems in document retrieval and CBIR. The road map of relevance feedback development is shown in Fig. 2.2. The abbreviations and symbols used in this thesis is shown in Table 2.1 and Table 2.2.

Table 2.1: List of Abbreviations

CBIR	Content-based Image Retrieval
GMM	Gaussian Mixture Model
LSI	Latent Semantic Indexing
MARS	Multimedia Analysis and Retrieval System
QBC	Query By Committee
SOM	Self-Organizing Map
SVM	Support Vector Machine
idf	inverse document frequency
tf	term frequency

2.3.1 Vector Space Model

Vector space model is the earliest relevance feedback approach introduced. It is designed to be used in document retrieval. In this model, each document is commonly represented by the search terms it contains. A particular expression for the document can be written as,

$$\mathbf{x} = (x_1, x_2, \ldots, x_j, \ldots, x_J), x_j \in \mathbb{R} \text{ and } x_j \geq 0 \tag{2.1}$$

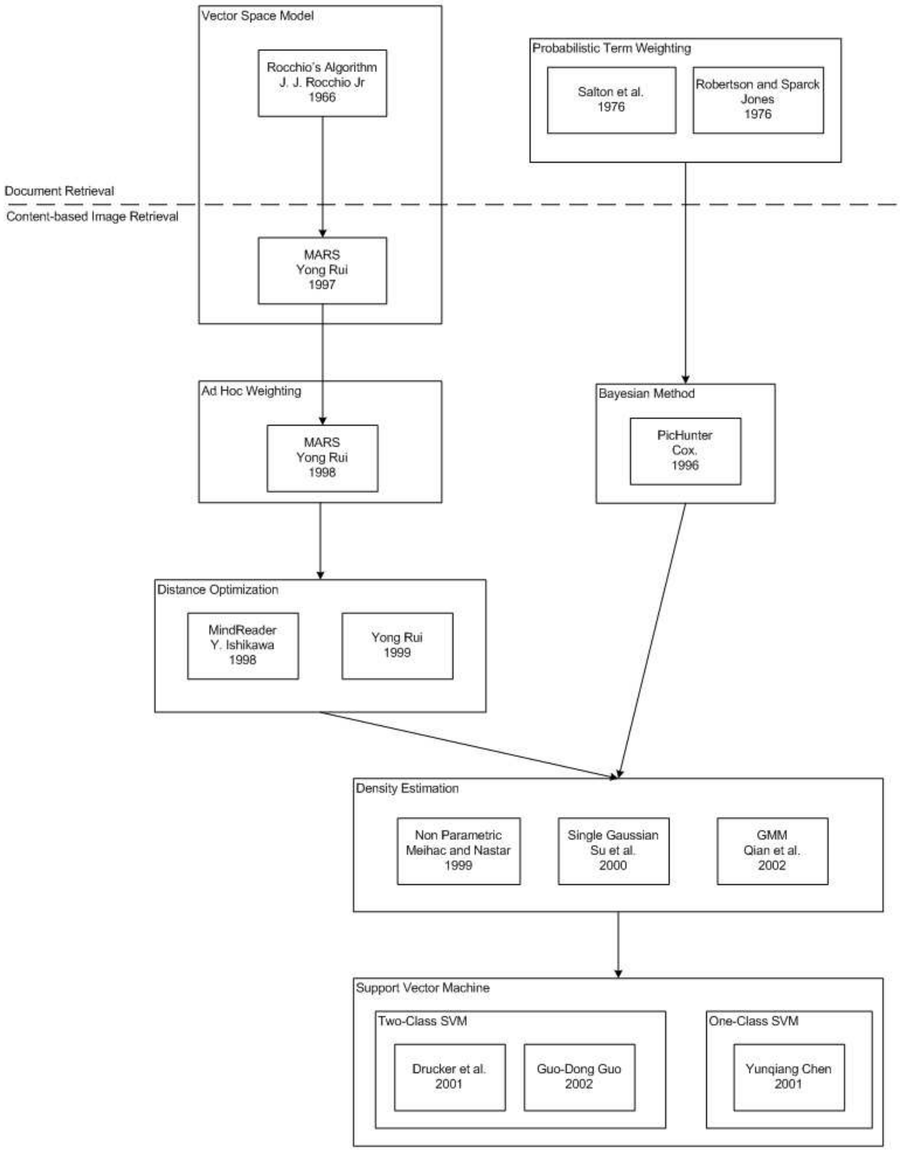


Figure 2.2: Road Map of Relevance Feedback

where x_j represents the importance of the j-th search term in the object $\mathbf{x}$. A commonly used importance measure in document retrieval is the product of term frequency (tf) and the inverse document frequency (idf). The term frequency measures the occurrence of a search term

Table 2.2: List of Symbols

$\mathbf{q}$	query vector
$\mathbf{x}$	object vector
t	index for iteration
i	index for object
j	index for feature component
J	maximum number of components
x_{ij}	j-th component of the i-th object
q_{tj}	j-th component of the query in the t-th iteration
s_i	the relevance score of the i-th object labelled by the user
S	the set of the relevance set, i.e. $\{s_1, s_2, \ldots, s_n\}$
O	objects in the whole collection
O_{Rt}	relevant objects identified in the t-th iteration
O_{Nt}	non-relevant objects identified in the t-th iteration
O_R	relevant objects identified up to the current iteration
O_N	non-relevant objects identified up to the current iteration
$\sigma_{O_R j}$	standard deviation of the j-th component among all relevant objects
$\sigma_{O_N j}$	standard deviation of the j-th component among all non-relevant objects
$\mu_{O_R j}$	mean of the j-th component among all relevant objects
$\mu_{O_N j}$	mean of the j-th component among all non-relevant objects
n	Number of objects in the collection

in the document and the document frequency measures the occurrence of a search term in the whole collection. A typical expression can be written as,

$$x_j = tf_{x_j} \times \log \frac{n}{df_j}, \tag{2.2}$$

where tf_{x_j} is the number of occurrence of the term j in document $\mathbf{x}$, df_j is the number of documents contain the term j, and n is the total number of documents.

In the vector space model, a typical query-document similarity measure can be computed as follows,

$$S(\mathbf{q},\mathbf{x}) = \frac{\mathbf{q}^T\mathbf{x}}{|\mathbf{q}||\mathbf{x}|}, S(\mathbf{q},\mathbf{x}) \in [0,1]. \tag{2.3}$$

In this equation, $S(\mathbf{q},\mathbf{x})$ is the cosine of the angle between the two vectors, $\mathbf{q}$ and $\mathbf{x}$, such that only the direction of the vector is considered in the similarity measure. The similarity measure of the vector space model in 2-dimensional case is illustrated in Fig. 2.3. The search terms used in this example are "relevance" and "feedback". The magnitude of object vector $\mathbf{x}_1$ is larger than that of $\mathbf{x}_2$, but the similarity values of object vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ to the query $\mathbf{q}$ are the same, since their angle to the query are equal. Thus, the magnitude of the query and object vectors would not affect the similarity measure.

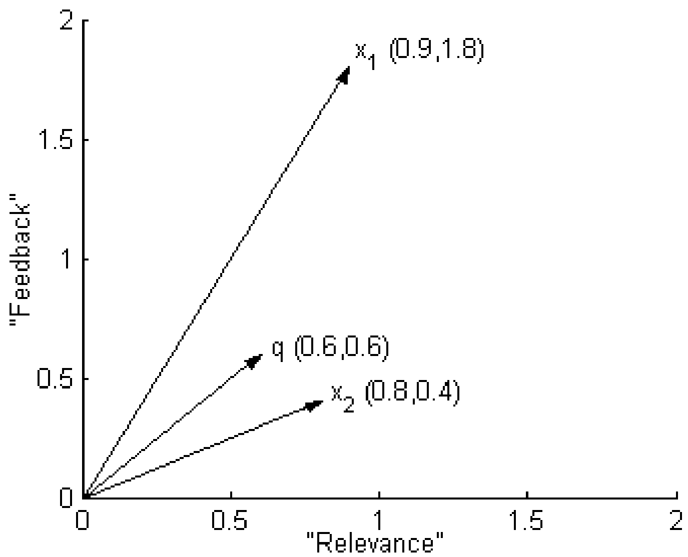


Figure 2.3: Similarity Measure in Vector Space Model

Document Retrieval

Among the approaches in the vector space model [32, 33, 54, 55] for document retrieval, Rocchio's technique [54, 55] is the earliest one. Rocchio's technique is based on the query modification in the vector space, and aims to obtain an approximation of the optimal query. Rocchio assumes that all the objects in the collection can be divided into two sets, relevant set and non-relevant set. Rocchio also defines an optimal query $\mathbf{q}_{opt}$ which maximizes the similarity to the relevant set and minimizes the similarity to the non-relevant set. The optimal query $\mathbf{q}_{opt}$ is mathematically defined by,

$$\mathbf{q}_{opt} = \arg\max_{\mathbf{q}} \left[\frac{1}{|O'_R|} \sum_{\mathbf{x}_i \in O'_R} S(\mathbf{q}, \mathbf{x}_i) - \frac{1}{|O'_N|} \sum_{\mathbf{x}_i \in O'_N} S(\mathbf{q}, \mathbf{x}_i) \right], \qquad (2.4)$$

where O'_R and O'_N are the sets of relevant and non-relevant objects in the whole collection. By deriving Eq. (2.4), we obtain

$$\begin{aligned}
& \frac{1}{|O'_R|} \sum_{\mathbf{x}_i \in O'_R} S(\mathbf{q}, \mathbf{x}_i) - \frac{1}{|O'_N|} \sum_{\mathbf{x}_i \in O'_N} S(\mathbf{q}, \mathbf{x}_i) && (2.5) \\
= \quad & \frac{1}{|O'_R|} \sum_{\mathbf{x}_i \in O'_R} \left(\frac{\mathbf{q}^T}{|\mathbf{q}|} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} \right) - \frac{1}{|O'_N|} \sum_{\mathbf{x}_i \in O'_N} \left(\frac{\mathbf{q}^T}{|\mathbf{q}|} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} \right) && (2.6) \\
= \quad & \frac{\mathbf{q}^T}{|\mathbf{q}|} \left[\frac{1}{|O'_R|} \sum_{\mathbf{x}_i \in O'_R} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} - \frac{1}{|O'_N|} \sum_{\mathbf{x}_i \in O'_N} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} \right] && (2.7) \\
= \quad & \frac{\mathbf{q}^T}{|\mathbf{q}|} \mathbf{A}. && (2.8)
\end{aligned}$$

The similarity measure is maximized when the query is equal to $c\mathbf{A}$ for any arbitrary scalar c. Thus, the optimal query is defined by,

$$\mathbf{q}_{opt} = \frac{1}{|O'_R|} \sum_{\mathbf{x}_i \in O'_R} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} - \frac{1}{|O'_N|} \sum_{\mathbf{x}_i \in O'_N} \frac{\mathbf{x}_i}{|\mathbf{x}_i|}. \qquad (2.9)$$

However, the optimal query in Eq. (2.9) cannot be used in practice as an initial query formulation, because the sets O'_R and O'_N are not known in advance. Thus, Rocchio defines an iterative relevance feedback technique to approximate the optimal query. In this technique, the relevant and non-relevant objects labelled by the user are used to replace O'_R and O'_N in Eq. (2.9). This technique is formulated as follows,

$$\mathbf{q}_{t+1} = \mathbf{q}_t + \beta(\frac{1}{|O_{R_t}|}\sum_{\mathbf{x}_i \in O_{R_t}} \mathbf{x}_i) - \gamma(\frac{1}{|O_{N_t}|}\sum_{\mathbf{x}_i \in O_{N_t}} \mathbf{x}_i), \tag{2.10}$$

where β and γ are suitable constants ($\beta, \gamma \in [0, 1]$ such that $\beta + \gamma = 1.0$).

CBIR

MARS [43, 58, 59, 60, 61] is among the first CBIR system which employs relevance feedback as an interactive tool to refine the image query. One of the methods proposed by MARS is the query point movement, which directly applied Rocchio's formula in relevance feedback for image retrieval. They propose a technique to construct a pseudo term vector from the feature vector. They use the component importance (ci) and inverse collection importance (ici) to replace the factors tf and idf in document retrieval. The factor ci measures the importance of a component in an object while the factor ici measures the importance of a component across different objects in the whole collection. The objects in the collection are represented by pseudo term vector $\mathbf{v}_i$, and it is constructed from the factors ci and ici. The factor ci of the i-th object is written as,

$$ci_i = [\frac{x_{i1}}{\mu_1}, \frac{x_{i2}}{\mu_2}, \ldots, \frac{x_{ik}}{\mu_k}], \tag{2.11}$$

where x_i is the feature vector of the i-th object, and μ_j is the mean of the j-th component among the feature vector in the whole collection. The factor ici of the i-th object is written as,

$$ici_i = [\log_2(\sigma_{i1} + 2), \log_2(\sigma_{i2} + 2), \ldots, \log_2(\sigma_{ik} + 2)], \tag{2.12}$$

where σ_j is the standard deviation of the j-th component among the feature vector in the whole collection. The pseudo term vector $\mathbf{v}_i$ of the i-th object is the product of ci and ici,

$$\mathbf{v}_i = ci_i \times ici_i. \tag{2.13}$$

The similarity measure between two objects is computed by the cosine of their weight vectors. The query is also represented by a pseudo term vector and updated with the Rochhio's formula.

Table 2.3: Characteristics of Vector Space Model approaches

	Rochioo	**MARS**
Reference	[54, 55]	[58, 59]
Domain	document	image
Objective	estimate the optimal query	
Representation	term vector	pseudo term vector
Vector Space	$\mathbf{x} \in \mathbb{R}^dand\forall x_j \geq 0$	
Similarity measure	cosine of angle	
Data involved	relevant objects	

Table 2.4: Equations in Vector Space Model approaches

Rochhio	$\mathbf{q}_{t+1} = \mathbf{q}_t + \beta(\frac{1}{\lvert O_{R_t}\rvert}\sum_{\mathbf{x}_i \in O_{R_t}} \mathbf{x}_i) - \gamma(\frac{1}{\lvert O_{N_t}\rvert}\sum_{\mathbf{x}_i \in O_{N_t}} \mathbf{x}_i)$
MARS	$\mathbf{q}_{t+1} = \mathbf{q}_t + \beta(\frac{1}{\lvert O_{R_t}\rvert}\sum_{\mathbf{x}_i \in O_{R_t}} \mathbf{x}_i) - \gamma(\frac{1}{\lvert O_{N_t}\rvert}\sum_{\mathbf{x}_i \in O_{N_t}} \mathbf{x}_i)$

Discussion

In [62], several relevance feedback systems for document retrieval have been investigated. It includes three techniques in the vector space model and three techniques in the probabilistic model (Section 2.3.4). Their result shows that the vector space model in [33] provides the best retrieval performance among these techniques.

2.3.2 Ad-hoc Re-weighting

The ad-hoc re-weighting approach [10, 11, 12, 43, 58, 59, 60, 61, 63] is among the earliest work of relevance feedback proposed in the field of CBIR. The common theme in different ad-hoc re-weighting techniques is that the similarity measure is modelled as the weighted combination of component similarity in the feature vector. The similarity measure is defined by,

$$S(\mathbf{q}, \mathbf{x}) = \sum_j w_j S_j(\mathbf{q}_j, \mathbf{x}_j), \tag{2.14}$$

where w_j is the weight corresponding to the j-th feature component and S_j is usually a Euclidean measurement.

The idea behind this technique is very intuitive, it associates the user interested components with higher weights and associates lower weights for the other components. The standard deviation method is a common approach for this objective. In this approach, each component in the feature vector is first normalized into a particular range in the preprocess status, so that the scale of each component is almost equal. In the relevance feedback learning process, the system computes the standard deviation of each component among the relevant objects labelled by the user, and these values are used to measure the user's preferences on different components. If the standard deviation of the relevant examples is high along the j-th feature component, then we can deduce that the user is not interested in the j-th feature component and a lower weight should be assigned to this component. Therefore, the weight of the feature component is assigned inverse proportional to the standard deviation,

$$w_j \propto \frac{1}{\sigma_{O_{R_j}}}. \tag{2.15}$$

MARS

The MARS system [43, 58, 59, 60, 61] is among the first one to propose a re-weighting scheme for relevance feedback in CBIR. In the MARS system, two independent query refinement techniques are introduced; they are named the re-weighting and the query point movement. Each component of the feature is normalized to a particular range in the preprocessing. For the re-weighting process in the relevance feedback process, the component weights are updated with the standard deviation method,

$$w_j = \frac{1}{\sigma_{O_{R_j}}}. \tag{2.16}$$

For the query point movement in the relevance feedback process, the MARS system uses a modified Rocchio's formula [54, 55] and updates the query point with the following equation,

$$\mathbf{q}_{t+1_j} = \alpha \mathbf{q}_{t_j} + \beta\left(\frac{1}{|O_{R_t}|} \sum_{\mathbf{x}_i \in O_{R_t}} \mathbf{x}_{i_j}\right) - \gamma\left(\frac{1}{|O_{N_t}|} \sum_{\mathbf{x}_i \in O_{N_t}} \mathbf{x}_{i_j}\right), \tag{2.17}$$

where α, β and γ are suitable constants.

Table 2.5: Characteristics of Ad-hoc Re-weighting approaches

	MARS
Reference	[43, 60, 61]
Domain	image
Objective	estimate components' importance
Representation	feature vector
Vector Space	$x \in \mathbb{R}^d$
Similarity measure	distance metric
Data involved	relevant objects

Table 2.6: Equations in Ad-hoc Re-weighting approaches

MARS	Query point movement	$\mathbf{q}_{t+1_j} = \alpha \mathbf{q}_{t_j} + \beta(\frac{1}{\lvert O_{R_t} \rvert} \sum_{\mathbf{x}_i \in O_{R_t}} \mathbf{x}_{i_j}) - \gamma(\frac{1}{\lvert O_{N_t} \rvert} \sum_{\mathbf{x}_i \in O_{N_t}} \mathbf{x}_{i_j})$
	Component re-weighting	$w_j = \frac{1}{\sigma_{O_{R_j}}}$

Discussion

Lack of optimality claim is the major problem in ad-hoc re-weighting approach, as there is no justification on the choice of the weight updating function and many of them can be good candidate. For example both $1/\sigma$ and $1/\sqrt{\sigma}$ can also be the components' weight updating function, but we cannot tell which one is better. Since the Ad-hoc Re-weighting approach can only tell us that the weight of a component should be increased if its standard deviation is decreased, but it does not tell which updating function is more suitable.

Dependence between components is not assumed in ad-hoc re-weighting approach. Since the weighting of each component is computed independently in ad-hoc re-weighting approach, the correlation among the components is ignored. However, the components of the feature vectors are not necessary to be independent.

2.3.3 Distance Optimization Approach

The distance optimization approach [2, 34, 57, 74] addresses the two problems; lack of optimality claim and dependence between components. In this approach, the relevance feedback problem is modelled as obtaining the optimal distance measure of the data. The distance measure in this approach is in the quadratic form,

$$D(\mathbf{q}, \mathbf{x}) = (\mathbf{x} - \mathbf{q})^T \mathbf{M} (\mathbf{x} - \mathbf{q}), \tag{2.18}$$

where $\mathbf{M}$ is a symmetric matrix and its components define the correlations between the components in the feature vector. Since this distance function is in ellipse shape in the feature space, the dependence of components can be measured. Moreover, by using this formula, we can claim that optimal distance measure is the one that minimize the total distance between the relevant objects and the query. Thus, the objective of the distance optimization approach is to find out the matrix $\mathbf{M}$ and query point $\mathbf{q}$ which minimize the total distance.

The similarity measure of distance optimization approach is a generalization of that in the ad-hoc re-weighting method. The similarity measure is in a sphere shape in the vector space if the Euclidean distance is directly used. The ad-hoc re-weighting approach generalizes it, and gives different weighting for the components in the feature vectors. This can be considered as an ellipse with its axis aligned with the coordinate axis. For the distance optimization approach, it further relaxes the constraint by not forcing the axis of the ellipse to align with the coordinate axis. We illustrate these three cases in Fig. 2.4. The distance optimization approach can be considered as a generalization of the ad-hoc re-weighting method.

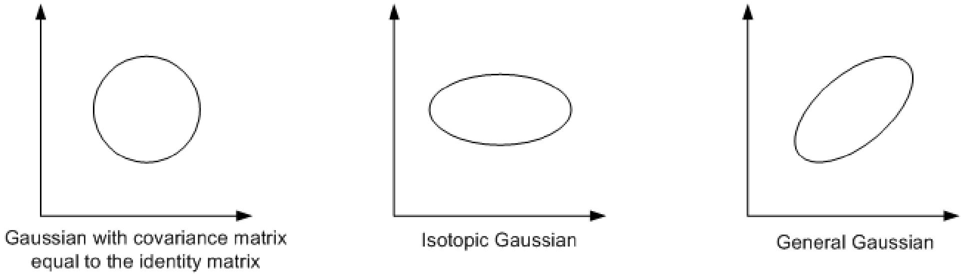


Figure 2.4: Illustration of distance measures in Euclidean distance, ad-hoc re-weighting and distance optimization approach

MindReader

MindReader [34] is the first distance optimization approach proposed. It formulates the objective function as follows,

$$\min_{\mathbf{q},\mathbf{M}} \sum_{\mathbf{x}_i \in O_R}^{|O_R|} s_i(\mathbf{x}_i - \mathbf{q})^T \mathbf{M}(\mathbf{x}_i - \mathbf{q}), \tag{2.19}$$

where s_i is the relevance score of the i-th sample. If no constraint is introduced, the zero matrix would give the minimum. Thus, the following constraint is introduced,

$$|\mathbf{M}| = 1. \tag{2.20}$$

This optimization problem can be solved with Lagrange multipliers. The optimal query point is the weighted average of feature vectors among relevant objects. It is mathematically written as,

$$\mathbf{q}_{t+1_j} = \frac{\sum_{\mathbf{x}_i \in O_R} s_i \mathbf{x}_{i_j}}{\sum_{\mathbf{x}_i \in O_R} s_i}. \tag{2.21}$$

The optimal matrix $\mathbf{M}$ is defined by,

$$\mathbf{M}_{t+1} = |\mathbf{C}_{t+1}|^{\frac{1}{J}} \mathbf{C}_{t+1}^{-1}, \tag{2.22}$$

$$c_{t+1j_1j_2} = \sum_{\mathbf{x}_i \in O_R} s_i(\mathbf{x}_{ij_1} - \mathbf{q}_{t+1j_1})(\mathbf{x}_{ij_2} - \mathbf{q}_{t+1j_2}), \tag{2.23}$$

where $\mathbf{C}_{t+1}$ is the weighted covariance matrix of the feature vectors among relevant objects in the t-th iteration, and $c_{t+1j_1j_2}$ is the scalar of matrix $\mathbf{C}_{t+1}$ at the j_1 row and j_2 column.

In order to obtain the matrix $\mathbf{M}$, we need to calculate the inverse of the covariance matrix $\mathbf{C}$. However, when the number of relevant objects is small, the covariance matrix $\mathbf{C}$ is not invertible. This situation occurs when the number of relevant objects is less than the number of feature components, and the covariance matrix $\mathbf{C}$ become singular. Thus, MindReader proposes to use Moore-Penrose inverse matrix [23] to deal with this situation.

Table 2.7: Characteristics of Distance Optimization approaches

	MindReader
Reference	[34]
Domain	image
Objective	estimate a optimal distance measure
Representation	feature vector
Vector Space	$x \in \mathbb{R}^d$
Similarity measure	distance metric
Data involved	relevant objects

Table 2.8: Equations in Distance Optimization approaches

MindReader	$\mathbf{q}_{t+1_j} = \frac{\sum_{\mathbf{x}_i \in O_R} s_i \mathbf{x}_{i_j}}{\sum_{\mathbf{x}_i \in O_R} s_i}$ $\mathbf{M}_{t+1} = \|\mathbf{C}_{t+1}\|^{\frac{1}{J}} \mathbf{C}_{t+1}^{-1}$ $cc_{t+1j_1j_2} = \sum_{\mathbf{x}_i \in O_R} s_i(\mathbf{x}_{ij_1} - \mathbf{q}_{t+1j_1})(\mathbf{x}_{ij_2} - \mathbf{q}_{t+1j_2})$

Discussion

The distance measure in distance optimization approach is analogous to a Gaussian distribution. The Gaussian distribution is modelled as,

$$G(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}} \exp\left[-(\mathbf{x}-\overline{\mathbf{x}})^T\mathbf{C}^{-1}(\mathbf{x}-\overline{\mathbf{x}})\right], \tag{2.24}$$

where $\overline{\mathbf{x}}$ is the mean of the data, and $\mathbf{C}$ is the covariance matrix of the data. This equation is analogous to the similarity measure Eq. (2.18) in the sense that,

$$G(\mathbf{x}_1) > G(\mathbf{x}_2) \quad \text{iff} \quad D(\mathbf{q},\mathbf{x}_1) < D(\mathbf{q},\mathbf{x}_2), \tag{2.25}$$

$$G(\mathbf{x}_1) = G(\mathbf{x}_2) \quad \text{iff} \quad D(\mathbf{q},\mathbf{x}_1) = D(\mathbf{q},\mathbf{x}_2). \tag{2.26}$$

The proof for these two statements is as follows,

$$G(\mathbf{x}_1) \quad > \quad G(\mathbf{x}_2) \tag{2.27}$$

$$\frac{\exp\left[-(\mathbf{x}_1-\overline{\mathbf{x}})^T\mathbf{C}^{-1}(\mathbf{x}_1-\overline{\mathbf{x}})\right]}{\sqrt{(2\pi)^n|\mathbf{C}|}} \quad > \quad \frac{\exp\left[-(\mathbf{x}_2-\overline{\mathbf{x}})^T\mathbf{C}^{-1}(\mathbf{x}_2-\overline{\mathbf{x}})\right]}{\sqrt{(2\pi)^n|\mathbf{C}|}} \tag{2.28}$$

$$(\mathbf{x}_1-\overline{\mathbf{x}})^T\mathbf{C}^{-1}(\mathbf{x}_1-\overline{\mathbf{x}}) \quad < \quad (\mathbf{x}_2-\overline{\mathbf{x}})^T\mathbf{C}^{-1}(\mathbf{x}_2-\overline{\mathbf{x}}). \tag{2.29}$$

By substituting Eq. (2.21) and Eq. (2.22) into it, we obtain

$$G(\mathbf{x}_1) \quad > \quad G(\mathbf{x}_2) \tag{2.30}$$

$$(\mathbf{x}_1-\mathbf{q})^T\mathbf{M}(\mathbf{x}_1-\mathbf{q}) \quad < \quad (\mathbf{x}_2-\mathbf{q})^T\mathbf{M}(\mathbf{x}_2-\mathbf{q}) \tag{2.31}$$

$$D(\mathbf{q},\mathbf{x}_1) \quad < \quad D(\mathbf{q},\mathbf{x}_1). \tag{2.32}$$

Eq.(2.26) can be proven with a similar method. Thus, the distance optimization approach and the Gaussian distribution measure always produce the same similarity measure. Moreover, it leads to the development of the density estimation approach in relevance feedback.

2.3.4 Probabilistic Model

The probabilistic model is a relevance feedback methodology designed for document retrieval. In this model, the object representation is similar to that of the vector space model. Each object in the collection is represented by a term vector, but each component in the term vector is restricted to be a binary value,

$$\mathbf{x} = (x_1, x_2, \ldots, x_j) \text{ and } x_j = \{0, 1\}. \tag{2.33}$$

The value of the j-th component depends on whether the object contains the j-th search term or not. The term vectors in the probabilistic model can also be considered as the vertices in a hypercube.

The main objective in the probabilistic model is to formulate a decision rule to classify the object into either relevant or non-relevant set. The obvious way to classify an object as relevant is to compare two probability functions; they are the probability of the document being relevant and the probability of the object being non-relevant. It is mathematically written as,

$$P(\mathbf{x} \in O_R|\mathbf{x}) > P(\mathbf{x} \in O_N|\mathbf{x}). \tag{2.34}$$

By applying Bayes' theorem, the decision rule can be transformed into a similarity measure,

$$S(\mathbf{q}, \mathbf{x}) = \log P(\mathbf{x}|\mathbf{x} \in O_R) - \log P(\mathbf{x}|\mathbf{x} \in O_N). \tag{2.35}$$

The objects are ranked by this equation instead of a strict decision on whether it is relevant or not. The probability $P(\mathbf{x}|\mathbf{x} \in O_R)$ and

$P(\mathbf{x}|\mathbf{x} \in O_N)$ are difficult to compute if no assumption is made. Thus, the probabilistic model always assumes that the search terms occur independently, and the probability $P(\mathbf{x}|\mathbf{x} \in O_R)$ and $P(\mathbf{x}|\mathbf{x} \in O_N)$ are expressed as,

$$\begin{aligned}
P(\mathbf{x}|\mathbf{x} \in O_R) &= \prod_{j=1}^{J} P(x_j|\mathbf{x} \in O_R) && (2.36)\\
&= \prod_{j=1}^{J} p_j^{x_j}(1-p_j)^{1-x_j}, && (2.37)\\
P(\mathbf{x}|\mathbf{x} \in O_N) &= \prod_{j} P(x_j|\mathbf{x} \in O_N) && (2.38)\\
&= \prod_{j=1}^{J} u_j^{x_j}(1-u_j)^{1-x_j}, && (2.39)\\
p_j &= P(x_j = 1|\mathbf{x} \in O_R), && (2.40)\\
u_j &= P(x_j = 1|\mathbf{x} \in O_N). && (2.41)
\end{aligned}$$

By substituting these equations into Eq.(2.35), the similarity measure becomes

$$S(\mathbf{q}, \mathbf{x}) = \sum_{j=1}^{J} x_j \log \frac{p_j(1-u_j)}{u_j(1-p_j)} + \sum_{j=1}^{J} \log \frac{1-u_j}{1-p_j}. \qquad (2.42)$$

Since the second term will not be affected by the object $\mathbf{x}$, this term can be removed from the equation, and the similarity measure can be written as,

$$S(\mathbf{q}, \mathbf{x}) = \sum_{j=1}^{J} x_j \log \frac{p_j(1-u_j)}{u_j(1-p_j)}. \qquad (2.43)$$

This equation is then treated as a weighted summation of the object's vector components, and it is expressed as,

$$
\begin{aligned}
S(\mathbf{q}, \mathbf{x}) &= \sum_{j=1}^{J} w_j x_j, & (2.44)\\
w_j &= \frac{p_j(1-u_j)}{u_j(1-p_j)}. & (2.45)
\end{aligned}
$$

where w_j is considered as a term weight and x_j indicates the presence or absence of the j-th search term in the object $\mathbf{x}$. Since these probability values p_j and u_j cannot be known in advance, various methods have been proposed to estimate these quantities with the distributions of relevant and non-relevant sets.

Salton et al. [62, 80, 84, 85] use the statistical information gathered in the relevance feedback process to estimate the quantities p_j and u_j. Referring to the notation of term occurrence data in Table 2.9, the quantities p_j and u_j are estimated with the following equations,

$$
\begin{aligned}
p_j &= \frac{r_j}{R}, & (2.46)\\
u_j &= \frac{n_j - r_j}{N - R}, & (2.47)
\end{aligned}
$$

where R is the number of relevant objects, N is the total number of objects in the collection, r_j is the number of relevant objects containing the j-th search term, and n_j is the number of objects in the whole collection containing the j-th search term. This equation assumes that the search term distribution in the retrieved relevant objects is the same as the distribution for the relevant objects in the whole collection, and the non-retrieved objects are treated as non-relevant. By substituting

Eq.(2.46), the similarity measure is expressed as,

$$S(\mathbf{q}, \mathbf{x}) = \sum_j x_j \log \frac{\frac{r_j}{R-r_j}}{\frac{n_j-r_j}{N-R-n_j+r_j}}. \tag{2.48}$$

Since all unlabelled objects in the collection are treated as non-relevant data, only relevant objects labelled by the user can improve the retrieval result, and the non-relevant objects provide no addition information to the relevance feedback system. Thus, the system should present the most probable objects to the user in the presentation set selection.

	Relevant objects	Non-relevant objects	All objects
$x_j = 1$	r_j	$n_j - r_j$	n_j
$x_j = 0$	$R - r_j$	$N - R - n_j - r_j$	$N - n_j$
All objects	R	$N - R$	N

Table 2.9: Occurrence of search term j in a collection of N objects

Robertson and Jones [36, 53] investigate several similar schemes and provide a theoretical explanation to explain which scheme is most suitable. In [53], four different term weight formulae are derived from Table 2.9. They are,

$$w_j^1 = \log \frac{\frac{r_j}{R}}{\frac{n_j}{N}}, \tag{2.49}$$

$$w_j^2 = \log \frac{\frac{r_j}{R}}{\frac{n_k-r_j}{N-R}}, \tag{2.50}$$

$$w_j^3 = \log \frac{\frac{r_j}{R-r_j}}{\frac{n_j}{N-n_j}}, \tag{2.51}$$

$$w_j^4 = \log \frac{\frac{r_j}{R-r_j}}{\frac{n_j-r_j}{N-n_j-R+r_j}}, \tag{2.52}$$

and the weight formula w_j^4 is the same as the formula proposed by Salton [62]. Robertson and Jones [52] analyze the properties and assumptions behind these weight formulae, and explain which weight formula is most suitable in document retrieval.

The weight formulae are constructed under different assumptions, and by analyzing these assumptions, the most suitable weight formula can be obtained. The weight formulae w_j^1 and w_j^3 compare the term distribution in the relevant set to the whole collection, so there are two assumptions behind it; they are,

1. the distributions of terms in relevant objects are independent, and

2. their distributions in all objects are independent.

Roberston [52] points out that these two assumptions are not strictly compatible. Since the search terms contained in the query occur more frequently in the relevant set than the non-relevant set, the first assumption implies that the cooccurrence of search terms in the relevant set is higher in the whole collection, and it is not compatible with the second assumption. Moreover, the weight formulae w_j^1 and w_j^2 consider the proportion of search terms that is present in the collection, while the ratio between the presence and absence of a search term in the collection is considered in w_j^3 and w_j^4. Since the objects should be ranked according to which search terms are present and absent in it, the weight formulae w_j^3 and w_j^4 should be more suitable. It shows that the weight formula w_j^4 is more suitable among these four weight formulae.

When the statistical value of some quantities is too small, problems may arise in computing the similarity measure, $r_j = 0$ for example, because of the logarithmic expression in the weight formula. For this reason, a small value 0.5 is added into each of the four elements in Table 2.9 to allow for some uncertainty, and the weight formula w_j^4 is modified as,

$$w_j^4 = \log \frac{\frac{r_j+0.5}{R-r_j+0.5}}{\frac{n_j-r_j+0.5}{N-n_j-R+r_j+0.5}}. \tag{2.53}$$

By substituting this equation, the similarity measure is written as,

$$S(\mathbf{q}, \mathbf{x}) = \sum_j x_j \log \frac{\frac{r_j+0.5}{R-r_j+0.5}}{\frac{n_j-r_j+0.5}{N-R-n_j+r_j+0.5}}. \tag{2.54}$$

So that the similarity measure can always yields a real number.

Table 2.10: Characteristics of Probabilistic Model approaches

	Salton	**Robertson and Jones**
Reference	[62, 80, 85, 84]	[53, 36]
Domain	document	
Objective	estimate the probability of an object being relevant	
Representation	term vector	
Vector Space	$x \in \mathbb{N}^d$ and $\forall x_j = \{0, 1\}$	
Similarity measure	probability measure	
Data involved	relevant objects	

Table 2.11: Equations of Probabilistic Model approaches

Salton	$S(\mathbf{q}, \mathbf{x}) = \sum_j x_j \log \frac{\frac{r_j}{R-r_j}}{\frac{n_j-r_j}{N-R-n_j+r_j}}$
Robertson and Jones	$S(\mathbf{q}, \mathbf{x}) = \sum_j x_j \log \frac{\frac{r_j+0.5}{R-r_j+0.5}}{\frac{n_j-r_j+0.5}{N-R-n_j+r_j+0.5}}$

Discussion

In the probabilities model, it models the feature representation as a binary vector, and it ignores some useful information in the objects. Since the occurrence of a search term in the document can tell us its importance in the document, when we compare it with the vector space model in section 2.3.1, more information is provided in the vector space model, and it should be able to perform better. Croft and Harper [17, 18] address this problem by modelling the feature representation in the same way as the vector space model, and use a probabilistic model to update its components in relevance feedback process.

2.3.5 Bayesian Approach

The Bayesian approach considers the relevance feedback problem as estimating the probability distribution of the query among the objects in the collection. Since the probability distribution is known, the probability associated with each object in the collection can be calculated, and can be used to rank the objects in the collection according to the query. Different modelings for Bayesian approach in relevance feedback are proposed, and the major models are target-based [13, 14, 15, 16, 79] and category-based [44, 47, 48, 50, 70, 69, 71]. In the target-based model, the relevance feedback process is modelled as a target search in the collection. It assumes that the user is searching for a particular target in the collection, and the similarity measure is modelled as the probability value of an object being the query's target. For the category-based model, it assumes that the user is searching for one or more objects from a category. The details of the category-based model will be described in section 2.3.6, the rest of this section describes the construction of the target-based model.

PicHunter

PicHunter [13, 14, 15, 16] estimates the probability of an object $\mathbf{x}$ being the query's target, $\mathbf{q}_{opt}$, with the relevant feedback history, H_t, given, and this probability function is written as,

$$P(\mathbf{x} = \mathbf{q}_{opt}|H_t). \tag{2.55}$$

The relevant feedback history consists of the set of objects, D_t, presented to the user, and the action, A_t, taken by the user in the t iteration. The relevant feedback history, H_{t-1}, is accumulated up to the $t-1$ iteration, so that the feedback history can be defined iteratively,

$$H_t = \{D_t, A_t, H_{t-1}\}. \tag{2.56}$$

The PicHunter system estimates the probability $P(\mathbf{x} = \mathbf{q}_t|H_t)$ incrementally from $P(\mathbf{x} = \mathbf{q}_t|H_{t-1})$ by applying Bayes' rule,

$$\begin{aligned}
& P(\mathbf{x} = \mathbf{q}_{opt}|H_t) && (2.57)\\
= \;& P(\mathbf{x} = \mathbf{q}_{opt}|D_t, A_t, H_{t-1}) && (2.58)\\
= \;& \frac{P(D_t, A_t|\mathbf{x} = \mathbf{q}_{opt}, H_{t-1})P(D_t, \mathbf{x} = \mathbf{q}_{opt}|H_{t-1})}{\sum_{\mathbf{x}_i \in O} P(D_t, A_t|\mathbf{x}_i = \mathbf{q}_{opt}, H_{t-1})P(D_t, \mathbf{x}_i = \mathbf{q}_{opt}|H_{t-1})} && (2.59)\\
= \;& \frac{P(A_t|\mathbf{x} = \mathbf{q}_{opt}, D_t, H_{t-1})P(\mathbf{x} = \mathbf{q}_{opt}|H_{t-1})}{\sum_{\mathbf{x}_i \in O} P(A_t|\mathbf{x}_i = \mathbf{q}_{opt}, D_t, H_{t-1})P(\mathbf{x}_i = \mathbf{q}_{opt}|H_{t-1})} && (2.60)
\end{aligned}$$

where $P(D_t, A_t|\mathbf{x} = \mathbf{q}_{opt}, H_{t-1})$ is written as $P(A_t|\mathbf{x} = \mathbf{q}_{opt}, D_t, H_{t-1})$ because D_t is determinate by H_{t-1}. When $t = 1$, where no past relevance feedback history is available, the probability $P(\mathbf{x} = \mathbf{q}_{opt}|H_1)$ is assumed to be evenly distributed, and equals to $\frac{1}{|O|}$.

The key part of the PicHunter system is to estimate the probability,

$$P(A_t|\mathbf{x} = \mathbf{q}_{opt}, D_t, H_{t-1}). \tag{2.61}$$

This probability can be considered as an estimation of user's behavior, because it predicts the user's response from the feedback information given. Since PicHunter assumes the user searches for a particular target, the system restricts the user to pick only one of the $|D_t|$ presented objects as relevant, the Eq.(2.61) is modelled as,

$$P(A_t|\mathbf{x} = \mathbf{q}_{opt}, D_t, H_{t-1}) = \frac{\exp\left(-d(\mathbf{x}_{A_t}, \mathbf{x})/\sigma_d\right)}{\sum_{\mathbf{x}_{A_t} \in D_t} \exp\left(-d(\mathbf{x}_{A_t}, \mathbf{x})/\sigma_d\right)}, \quad (2.62)$$

where the function $d(.)$ is a distance function, and σ_d is the standard deviation of the distances of function $d(.)$ among the objects in the collection.

Table 2.12: Characteristics of Bayesian approaches

	PicHunter
Reference	[13, 14, 15, 16]
Domain	image
Objective	estimate the probability of an object is relevant
Representation	feature vector
Vector Space	$x \in \mathbb{R}^d$
Similarity measure	probability measure
Data involved	relevant and non-relevant objects

Table 2.13: Equations in Bayesian approaches

PicHunter	$P(\mathbf{x} = \mathbf{q}_{opt}|H_t) = \frac{P(A_t|\mathbf{x}=\mathbf{q}_{opt}, D_t, H_{t-1}) P(\mathbf{x}=\mathbf{q}_{opt}|H_{t-1})}{\sum_{\mathbf{x}_i \in O} P(A_t|\mathbf{x}_i=\mathbf{q}_{opt}, D_t, H_{t-1}) P(\mathbf{x}_i=\mathbf{q}_{opt}|H_{t-1})}$ $P(A_t|\mathbf{x} = \mathbf{q}_{opt}, D_t, H_{t-1}) = \frac{\exp\left(-d(\mathbf{x}_{A_t}, \mathbf{x})/\sigma_d\right)}{\sum_{\mathbf{x}_{A_t} \in D_t} \exp\left(-d(\mathbf{x}_{A_t}, \mathbf{x})/\sigma_d\right)}$

Discussion

In PicHunter, the system is too sensitive to the user's response in the latest iteration. In Eq.(2.60), the system estimates the probability of an object being the target of the query, and the latest user's response is a major factor in the equation. The retrieval result is highly dependent on the user's response in the latest iteration. Thus, PicHunter cannot provide a stable retrieval result in the relevance feedback process.

2.3.6 Density Estimation Approach

The density estimation approach is developed from the Bayesian approach and can be considered as a generalization of the distance optimization approach. The objective of the density estimation approach is to estimate the distribution of the relevant objects in the collection, and it uses the distribution function to rank the relevance of the objects to the query. This objective is analogous to that of the Bayesian approach, but the density estimation approach assumes the user is searching for a set of relevant objects instead of a single target. Moreover, the distance optimization approach can be considered as a special case in the density estimation approach, since their formulations are the same when a target distribution is assumed to be a Gaussian distribution.

The density estimation approach can be further divided into parametric approach [47, 48, 50, 69, 70, 71, 81] and non-parametric approach [44]. In the parametric approach, the relevance feedback system assumes that the target distribution is governed by a certain statistical law, Gaussian distribution and GMM for example, and the objective of the system is to estimate the parameters of the statistical distribution for the queries. In the non-parametric approach, no apriori information about the statistical law underlying the query's target is required.

Non-Parametric Approach

Meihac and Nastar [44] propose a non-parametric density estimation approach for the relevance feedback problem. The similarity measure in this approach is defined by,

$$S(\mathbf{q}, \mathbf{x}) = \log P(\mathbf{x}|\mathbf{x} \in O_R) - \log P(\mathbf{x}|\mathbf{x} \in O_N), \tag{2.63}$$

where $P(\mathbf{x}|\mathbf{x} \in O_R)$ and $P(\mathbf{x}|\mathbf{x} \in O_N)$ are the distributions of relevant and non-relevant sets respectively. The feedback information given by the user is used to estimate these two distributions.

Since the probability distributions, $P(\mathbf{x}|\mathbf{x} \in O_R)$ and $P(\mathbf{x}|\mathbf{x} \in O_N)$, of the relevant and non-relevant sets cannot be known in advance, the system applies the Parzen window estimation in maximum likelihood estimation to estimate these two distributions. The Parzen window estimation is a non-parametric density estimation method. It does not assume the form of the distribution for data, and it uses the empirical data to model the distribution. The distribution of relevant set is modelled as follow,

$$\begin{aligned} P(\mathbf{x}|\mathbf{x} \in O_R) &= \prod_j f(\mathbf{x}_j|O_R), & (2.64)\\ &= \prod_j \sum_{y_i \in O_R} f_{G\sigma}(x_{i_j} - y_i), & (2.65)\\ f_{G\sigma}(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, & (2.66) \end{aligned}$$

where $f_{G\sigma}$ is a Gaussian smoothing function, and the feature components are assumed to be independent. The formulation of $P(\mathbf{x}|\mathbf{x} \in O_N)$ is similar to that of $P(\mathbf{x}|\mathbf{x} \in O_R)$.

Parametric Approach - Gaussian Model

In [69, 70, 71], a Gaussian distribution is used to model the distribution of the relevant set. The similarity measure here is similar to that in the non-parametric approach. In this approach, only the probability distribution of the relevant set is considered, and the probability distribution of the non-relevant set is ignored.

In this approach, the parametric density estimation is used to estimate the distribution of relevant set in contrast to the non-parametric technique in the non-parametric approach. The isotopic Gaussian distribution is used to characterize the distribution of the relevant set. It is mathematically written as,

$$P(\mathbf{x}|\mathbf{x} \in O_R) = \frac{1}{(2\pi)^{j/2}|\mathbf{C}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{C}^{-1}(\mathbf{x}-\mu)}, \tag{2.67}$$

where j is the dimension of the feature vectors, μ is the mean vector of the relevant objects, $\mathbf{C}$ is a diagonal matrix, and each component c_{jj} is the variance of the relevant objects in the j-th dimension. Since the class probability $P(\mathbf{x} \in O_R)$ is invariant among the object $\mathbf{x}$, this term can be ignored in the similarity measure. Thus, the distance measure is then written as,

$$S(\mathbf{q}, \mathbf{x}) = (\mathbf{x} - \mu)^T\mathbf{C}^{-1}(\mathbf{x} - \mu). \tag{2.68}$$

The relevant examples provided by the user are used to compute the mean vector μ and the diagonal matrix $\mathbf{C}$. Since the distance measure in the distance optimization approach can be considered as a Gaussian distribution, this updating scheme is analogous to the distance optimization appraoch.

Since the non-relevant objects does not follow any particular distribution in general, we cannot assume the distribution of the non-relevant set follows a Gaussian distribution. In order to utilize the information provided by the non-relevant objects, these objects are used to penalize the objects which are located nearby. The intuition behind it is that objects located near to the non-relevant objects are considered not as relevant as other objects. A penalizing function $f(\cdot)$ is constructed as follow,

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in O_N} g(d(\mathbf{x}, \mathbf{x}_i)), \tag{2.69}$$

where $g(\cdot)$ is a Gaussian function, and $d(\cdot)$ is a distance function. The distance measure is then defined by,

$$S(\mathbf{q}, \mathbf{x}) = (\mathbf{x} - \mu)^T \mathbf{C}^{-1} (\mathbf{x} - \mu) + f(\mathbf{x}). \tag{2.70}$$

Thus, the distance of objects that located near to the non-relevant objects are increased.

Parametric Approach - Gaussian Mixture Model

In [50], they use the Gaussian Mixture Model (GMM) to model the distribution of the relevant set. Modelling the target distribution as a single Gaussian distribution can only retrieve the relevant objects around the query in a local area, and it fails to model the distribution when some relevant objects are far away from the query. Thus, the target distribution is assumed to follow GMM in [50], in order to retrieve more relevant objects, and improve the retrieval performance. The similarity measure in this approach is mathematically written as,

$$\begin{aligned} S(\mathbf{q}, \mathbf{x}) &= P(\mathbf{x} \in O_R | \mathbf{x}), && (2.71) \\ &= \sum_{k=1}^{K} \alpha_k G(\mathbf{x} | \mu_k, \mathbf{C}_k), && (2.72) \\ &= \sum_{k=1}^{K} \alpha_k \frac{1}{(2\pi)^{j/2} |\mathbf{C}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x} - \mu_k)^T \mathbf{C}_k^{-1} (\mathbf{x} - \mu_k)}, && (2.73) \end{aligned}$$

where K is the number of Gaussian components, j is the subscript for the dimension of the feature vectors, α_k, μ_k and C_k are the weight, mean vector and the covariance matrix of the k-th Gaussian respectively.

In order to construct the similarity measure for a query, the parameters in the Eq.(2.73) have to be estimated. Firstly, we have to estimate the number of Gaussian components K. In this approach, it proposes to find a set of coverings S_{Cover}, in which each covering contains as much relevant objects as possible, but none of the non-relevant objects. Let O_R and O_N be the sets of relevant and non-relevant objects. If O_R is not empty, the object $\mathbf{x}_m$ with the largest likelihood of being relevant is picked and a new covering is created in the set S_{Cover},

$$\mathbf{x}_m = \arg\max_{\mathbf{x}} S(\mathbf{q}, \mathbf{x}). \tag{2.74}$$

Then, we calculate the maximal distance between $\mathbf{x}_m$ and O_R, and the minimal distance between $\mathbf{x}_m$ and O_N,

$$d_{\max} = \max D(\mathbf{x}_m, \mathbf{x}), \forall \mathbf{x} \in O_R, \tag{2.75}$$

$$d_{\min} = \min D(\mathbf{x}_m, \mathbf{x}), \forall \mathbf{x} \in O_N. \tag{2.76}$$

The radius of the covering is defined by,

$$r = \begin{cases} d_{\max} + d_{\min}/2 & \text{if } d_{\min} \geq d_{\max} \\ \rho d_{\min} & \text{otherwise} \end{cases}, \tag{2.77}$$

where ρ is a suitable constant and $0 < \rho < 1$. All the relevant objects in the set O_R fall in this covering are removed. This procedure repeats until the set O_R becomes empty.

With a set of coverings S_{Cover}, it considers each covering as a Gaussian component in the target distribution, and estimates their parameters. The weight of the k-th Gaussian component is set to the portion of relevant objects fall in the corresponding covering. Since the number of relevant objects in each covering may not be sufficient to estimate the covariance matrix of the corresponding Gaussian components. The system simplifies the covariance matrix, $\mathbf{C}_k$, as a diagonal matrix, and combines the unlabelled objects fall in this coverings with the relevant examples to estimate the mean μ_k and covariance matrix $\mathbf{C}_k$.

Table 2.14: Characteristics of Density Estimation Approach

	Non-Parametric Approach	**Gaussian**	**GMM**
Reference	[44]	[69, 70, 71]	[50]
Domain	image		
Objective	estimate the distribution of the relevant set		
Representation	feature vector		
Vector Space	$x \in \mathbb{R}^d$		
Similarity measure	probability measure		
Data involved	relevant and non-relevant objects		

Table 2.15: Equations in Density Estimation Approach

Non-Parametric Approach	$S(\mathbf{q}, \mathbf{x}) = \log P(\mathbf{x} \vert \mathbf{x} \in O_R) - \log P(\mathbf{x} \vert \mathbf{x} \in O_N)$ $P(\mathbf{x} \vert \mathbf{x} \in O_R) = \prod_j \sum_{y_i \in O_R} f_{G\sigma}(x_{i_j} - y_i)$ $f_{G\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ The formulation of $P(\mathbf{x} \vert \mathbf{x} \in O_N)$ is similar to that of $P(\mathbf{x} \vert \mathbf{x} \in O_R)$
Gaussian	$S(\mathbf{q}, \mathbf{x}) = (\mathbf{x} - \mu)^T \mathbf{C}^{-1} (\mathbf{x} - \mu) + f(\mathbf{x})$ $f(x) = \sum_{x_i \in O_N} g(d(x, x_i))$
GMM	$S(\mathbf{q}, \mathbf{x}) = \sum_{k=1}^{K} \alpha_k \frac{1}{(2\pi)^{j/2} \vert \mathbf{C}_k \vert^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \mathbf{C}^{-1} (\mathbf{x}-\mu_k)}$

Discussion

Large amount of feedback data are required in the non-parametric approach, since it uses a non-parametric estimation to estimate the distributions of relevant and non-relevant objects, and this estimation method requires a large amount of data to estimate the underlying distribution. However, the size of the training set in relevance feedback is usually small, and it is not sufficient to provide enough information to estimate the target distribution.

The major problem of modelling the target distribution as a single Gaussian distribution is that it is only able to retrieve the relevant objects in a local area, and it fails to retrieve the relevant objects that are far away from the query. For the real world data, the relevant objects are usually organized in several separated clusters, and this approach is only able to retrieve the relevant objects in a local area.

The major problem of modelling the target distribution as GMM is that a large amount of feedback information is required. Since there are K Gaussian distributions in this approach, and the number of relevant examples required to estimate these K Gaussian distributions is proportion to K. The number of relevant examples provided in relevance feedback process may not be able to estimate the parameters of the GMM efficiently.

2.3.7 Support Vector Machine

Support vector machine (SVM) [5, 78] is a core technique for regression and pattern classification problem in machine learning theory. It has strong theoretical foundation and excellent empirical successes, and it has been applied in many different problem domains, handwritten digit recognition, object recognition and text classification, for example. Recently, the SVM is applied in relevance feedback problem. The regular SVM is applied in relevance feedback by considering it as a classification problem, and the one-class SVM is applied in relevance feedback by considering it as a density estimation problem.

Regular Support Vector Machine

The regular SVM is used to solve the two-class classification problem. The two-class classification problem can be formalized as estimating a classifier $f : \mathbb{R}^d \rightarrow \{-1, 1\}$ from a set of training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ is a feature vector, and y_i is its label. For the case when the solution is linear separable, the objective of the SVM is to find the hyperplane that separates the training data by the maximal margin. All vectors lying on one side of the hyperplane are labelled as +1, and all vectors lying on the other side of the hyperplane are labelled as -1. When the algorithm is applied to non-separable data, no feasible solution can be found, so that a further cost is introduced for the misclassification of the training data. In the case of non-linear separation, the training data are transformed into a high dimensional feature space through a Mercer kernel, and the technique in linear separation is applied in the new feature space. This objective can be modelled as the following equation,

$$\min_{\mathbf{w} \in \mathcal{F}} \quad \frac{1}{2}||\mathbf{w}||^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n} \xi_i, \tag{2.78}$$

$$s.t. \quad y_i(\mathbf{w} \cdot \Phi(\mathbf{x_i})) \geq \rho - \xi_i, \tag{2.79}$$

$$\xi_i \geq 0, \rho \geq 0, \tag{2.80}$$

where ξ_i represents the margin errors for the non-separable training data, and $\nu \in [0, 1]$ is a parameter to control the tradeoff in the number of support vectors and margin errors.

The regular SVM has been directly applied in relevance feedback as a two-class classification problem. In [20, 25, 26, 75], the relevant set and non-relevant set are considered as two different classes, and SVM is applied to classify these two datasets. For the relevance feedback problem, we need an evaluation function to output the relevance of the objects in the collection. A common approach for the evaluation function is to model it as the distance between the object and the classifier.

One-class SVM

One-class SVM [51, 64, 72, 73] is developed from the regular SVM, and it is used to solve the density estimation problem. The objective of the one-clss SVM is to construct a decision hypersphere that includes most of the positive data and minimizes the size of the hypersphere. This objective can be formulated as follows,

$$\min_{R\in\mathbb{R},\mathbf{c}\in\mathcal{F}} \quad R^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i, \tag{2.81}$$

$$s.t. \quad ||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 \leq R^2 + \xi_i, \tag{2.82}$$

$$\xi_i \geq 0, \tag{2.83}$$

where ξ_i are the slack variables for margin error, $\mathbf{c}$ and R are the center and radius of the hypersphere, and $\nu \in [0, 1]$ is a parameter to control the tradeoff between the radius of the hypersphere and the number of positive examples.

In [9], it treats the relevance feedback problem as a density estimation problem. It applies the one-class SVM to estimate the distribution of the relevant set. It uses only the relevant objects labelled by the user in the relevance feedback process. These relevant objects are presented to the one-class SVM, and used to estimate the distribution of the relevant set.

Table 2.16: Characteristics of Support Vector Machine Approach

	Regular SVM	**One-class SVM**
Reference	[20, 25, 26, 75]	[9]
Domain	image	
Objective	Find a classifier that separate the relevant and non-relevant set with maximum margin	Maximizes the number of relevant objects lying inside the classifier and minimize the volume of the classifier
Representation	feature vector	
Vector Space	$x \in \mathbb{R}^d$	
Similarity measure	distance to the classifier	
Data involved	relevant and non-relevant objects	relevant objects

Table 2.17: Equations in Support Vector Machine Approach

Regular SVM	$\min\limits_{\mathbf{w} \in \mathcal{F}} \frac{1}{2}\|\|\mathbf{w}\|\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i$
one-class SVM	$\min\limits_{R \in \mathbb{R}, \mathbf{c} \in \mathcal{F}} R^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$

Discussion

The imbalance between relevant and non-relevant sets makes the regular SVM not suitable for the classification task in relevance feedback problem. Since the regular SVM treats the relevant set and non-relevant set equally, and it does not consider that the number of non-relevant images is significantly larger than the relevant images. This imbalanced dataset problem will lead to the positive data (relevant images) being overwhelmed by the negative data (non-relevant images) [9].

In the one-class SVM, a large amount of useful information is ignored. Since the non-relevant examples are ignored in the relevance feedback process. It only uses the relevant examples to estimate the decision boundary for the query, and large portion of objects in the database is non-relevant to the query. Since only a small portion of the training data belongs to the relevant set, the information provided to the one-class SVM is limited.

2.4 Presentation Set Selection

The strategy for selecting objects to present to the user plays an important role in a relevance feedback system. Since the set of presented objects is the training data in the relevance feedback learning process. This training set is usually small, and it is valuable information for the system. There are two main directions for the presentation set selection strategy; they are the *most-probable* and the *most-informative*.

2.4.1 Most-probable strategy

In the most-probable strategy, the system presents the most relevant objects to the user in each iteration of the training process. Thus, the user retrieves the current best objects in each iteration, and the system uses this set of objects in the learning process in the next iteration. This presentation set selection strategy is adopted by most of the relevance feedback algorithms. However, the major drawback of this strategy is that most probable object are usually very similar to those labelled as relevant objects, and provides little information for the system in the further learning process.

2.4.2 Most-informative strategy

In the most-informative strategy, the system presents the most ambiguous objects to the user, that is the object that the system is most uncertain about. Thus, the system obtains more information from the user's feedback and clarifies user's intention in the query. However, identifying which objects are most informative in respect to the query is not an easy task. The disadvantage of this strategy is that it is more difficult for the user to judge when to stop the relevance feedback process. Since the user does not know the retrieval result can satisfy his need or not.

Background of the most-informative strategy

The problem of selecting the most informative object in the database is first studied in the theory of learning. Most of the research in the learning theory is based on a paradigm which the learner is trained and tested by examples drawn from the same random distribution. In this paradigm the learner is passive and has no control over the information it receives. Queries in learning is then studied in [1, 77], in which the learner has the power to ask the queries he wants.

In [3, 4], Baum and Lang propose a queries learning algorithm in neural network, and use it to train and classify handwritten characters. However, constructing a query by generating examples may provide an unexpected problem. They found that many of the generated examples by the algorithm are not recognizable by the user. It shows that an example generated by machine may contain no natural meaning. Thus, researchers suggest picking the examples in the collection, instead of generating new examples.

In [22, 66], Freund and Seung propose the query by committee (QBC) algorithm. In QBC, several different classifiers are learned with the same training set, and all the data in the collection are tested with these classifiers. Then, the data with the greatest entropy among these classification results are selected as query. Since revealing these examples can provide the greatest information to the system. This algorithm has been applied in a number of domains, and recently applied in text retrieval [41] and relevance feedback [75].

The presentation set selection algorithm proposed in the relevance feedback can be divided into three different categories; they are iteration minimization, vector space minimization, and maximum entropy approach.

Iteration Minimization

PicHunter [14] proposes an iteration minimization technique to obtain the presentation set. PicHunter is a relevance feedback system which is designed to look for a single target image. The goal of the presentation set selection scheme in PicHunter is to minimize the total amount of iterations required. This scheme tries to retrieve as much information from the user as possible, so that the search process can end quickly. The estimated number of iteration in PicHunter is defined by,

$$E(D_t) = P(\text{target not found}) \sum C(P(\mathbf{x}|A_t))P(A_t|D_t) \tag{2.84}$$

where

$$P(\text{target not found}) = 1 - P(\mathbf{x}_1 = \mathbf{q}_{opt}) - P(\mathbf{x}_2 = \mathbf{q}_{opt}) - \cdots - P(\mathbf{x}_{|D_t|} = \mathbf{q}_{opt}), \tag{2.85}$$

$C(P(\mathbf{x}|A_t))$ is an estimate to the number of iterations left based on the probability of the object $\mathbf{x}$ being the target of the search when the user's response A_t in iteration t is given, $P(A_t|D_t)$ is the probability of A_t being the response of the user, and D_t is the presentation set selected.

In order to estimate the number of iterations left, PicHunter uses the entropy in information theory to estimate $C[P(\mathbf{x}|A_t)]$. Since entropy is a measure of amount of information hidden in a probability distribution, and can be used to estimate the effort required to resolve the ambiguity specified by $P(\mathbf{x}|A_t)$. The function $C(P(\mathbf{x}|A_t))$ is defined by,

$$C(P(\mathbf{x}|A_t)) \approx -\alpha \sum_{\mathbf{x}_i \in O} P(\mathbf{x}_i|A_t) \log P(\mathbf{x}_i|A_t), \tag{2.86}$$

where α is a positive constant and it is irrelevant for the goal of minimizing the number of iterations.

Finding the presentation set D_t that minimize the $E(D_t)$ is not a trivial task. The problem of obtaining the optimal solution is very costly. Thus, PicHunter uses a Monte Carlo approach to select the presentation set. It samples several random presentation sets, and selects the one that minimizes the function $E(D_t)$.

Vector Space Minimization

In [75, 76], it proposes an active learning algorithm to select the presentation set in the SVM-based relevance feedback approaches. Their objective is to find the presentation set that reduces the version space of the classifiers maximally. The version space is the area that spanned the classifiers which is able to separate the training data according to their label. Thus, the confident of the classifier increases as the volume of the version space decreases. The best strategy to reduce the version space is to halve the version space in each iteration. It shows that selecting the data on the boundary of the SVM classifier as the presentation set can halve the version space approximately.

Maximum Entropy Approach

In [37], it proposes a presentation set selection algorithm based on the principle of maximum entropy. This algorithm can be applied on various relevance feedback techniques. The objective of the algorithm is to find a presentation set with maximum entropy. It is achieved by dividing the vector space into k sector, and the integral probability of each sector is $\frac{1}{k}$, where k is the size of the presentation set. The entropy is maximized when the probability of each outcome is the same, so that this presentation set has the maximum entropy.

Discussion

It is hard to tell which presentation set selection algorithm is the best. Since they should be applied in different situations.

- The iteration minimization approach should be applied in the case that the system assumes the user is looking for a single target image.

- The vector space minimization approach should be applied in the case that the system treats the relevance feedback problem as a classification problem.

- The maximum entropy approach should be applied in the case that the system treats the relevance problem as a density estimation problem.

□ **End of chapter.**

Chapter 3

Biased Support Vector Machine for Content-Based Image Retrieval

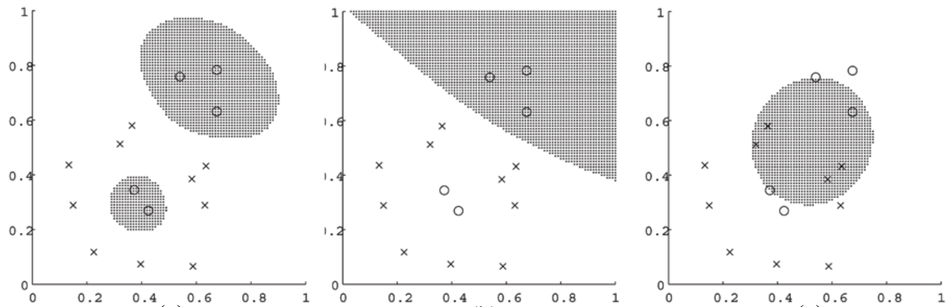
3.1 Motivation

In the past years, relevance feedback techniques in CBIR have evolved from early ad-hoc re-weighting techniques to recent machine learning techniques. Inspired by the term-weighing and relevant techniques in document retrieval [56], ad-hoc re-weighting technique [12, 61] has been proposed in CBIR, and it shows that relevance feedback is a powerful technique to improve the retrieval result. Later on researchers began to look at this problem from a more systematic point of view by formulating it into an optimization, classification or density estimation problem. Many relevance feedback techniques are suggested, such as distance optimization approach [34], Bayesian approach [14], Gaussian and GMM in parametric density estimation [50, 70], and Parzen window estimation in non-parametric density estimation [44]. Recently there are attempts to incorporate SVM into relevance feedback problem, and it shows that SVM-based techniques are more promising and effective techniques than other techniques [9, 75].

Typical relevance feedback approaches by SVMs are based on strict binary classification [75] or one-class classification [9]. However, the strict binary classification does not consider the imbalance problem in relevance feedback, that is the number of non-relevant images are significantly larger than the relevant images. This imbalanced dataset problem will lead to the positive data be overwhelmed by the negative data. The one-class technique seems to avoid the imbalance problem. However, it cannot work well without the help of negative information. We illustrate these problems in Fig. 3.1. The circles and crosses represent the positive and negative data respectively. The boundaries of the shadow regions represent the decision boundaries. The optimal decision boundary is shown in Fig. 3.1a. Fig. 3.1b shows the decision boundary of the regular SVM. We can see that the positive data is being overwhelmed by the negative data, and the system treats the positive data as outliers. Fig. 3.1c shows the decision boundary of the one-class SVM. We can see that without the help of the negative data, the system classifies the negative data as positive. In order to overcome the imbalanced dataset problem and fuse the negative information, we propose the Biased Support Vector Machine derived from the one-class SVM to construct the relevance feedback technique in CBIR.

3.2 Background

In the following, we introduce the basic ideas and formulations of regular SVMs, one-class SVMs and our BSVM. SVMs implement the principle of structural risk minimization by minimizing Vapnik-Chervonenkis



(a) (b) (c)

Figure 3.1: Drawbacks of Regular SVM and One-class SVM

dimensions. On pattern classification problems, SVMs provide very good generalization performance in empirical applications.

3.2.1 Regular Support Vector Machine

Let us consider the regular SVMs in binary classification problem. Assume we are given training data $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ in some space $\mathcal{X} \in \mathbb{R}^d$ and their corresponding class labels $\{y_1, y_2, \ldots, y_n\}$ where $y_i \in \{-1, 1\}$. The goal of learning in regular SVMs is to find the hyperplane that can classify the data correctly, and the margin between two sets of data is maximized. We illustrate the linear separating hyperplane of regular SVMs for separable data in Fig. 3.2. The circles and crosses are the positive data and negative data, respectively. The circles and the crosses on the two solid lines are called support vectors. The dashed line between the two solid lines is the decision hyperplane. It separates the positive and negative data with maximum margin.

By applying the Mercer kernel theory, the data in the original space $\mathcal{X}$ can be projected to a higher dimensional space $\mathcal{F}$ which is spanned by a mapping function Φ. The mapping function corresponds to Mercer kernel $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$ which implicitly computes the dot product in $\mathcal{F}$. The use of kernels allows the SVMs to deal with non-linearity of the distribution of training images in an efficient way. Hence, the goal of SVMs is to find the optimal separating hyperplane depicted by a vector $\mathcal{F}$ in the feature space,

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}), \tag{3.1}$$

where $\mathbf{w}$ is the normal to the hyperplane, and $\Phi(\mathbf{x})$ is the mapping function. The task to find the optimal hyperplane turns to solving the primal optimization problem in the form of soft margin SVMs,

$$\begin{aligned}
\min_{\mathbf{w}\in\mathcal{F}} \quad & \frac{1}{2}||\mathbf{w}||^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i, & (3.2)\\
s.t. \quad & y_i(\mathbf{w}\cdot\Phi(\mathbf{x_i})) \geq \rho - \xi_i & (3.3)\\
& \xi_i \geq 0, \rho \geq 0, & (3.4)
\end{aligned}$$

where ξ_i represents the margin errors for the non-separable training data, and $\nu \in [0, 1]$ is a parameter to control the tradeoff in the number of support vectors and margin errors. To understand the role of ρ, note that when the margin errors $\xi_i = 0$, one can show that the two classes are separated by a margin with $2\rho/||\mathbf{w}||$ from Eq.(3.3). By introducing the Lagrange multipliers, the optimization problem can be transformed to its dual form, and solved with quadratic programming techniques.

The regular SVMs have been applied in relevance feedback by treating it as a two-class classification problem. The relevant images labeled by the user are treated as positive data, and the non-relevant images labeled by the user are treated as negative data. The SVMs training is applied in every iteration in the relevance feedback process. However, this technique does not consider the imbalanced dataset problem, in which the number of non-relevant images are significantly larger than the relevant images. This imbalanced dataset problem will lead to the

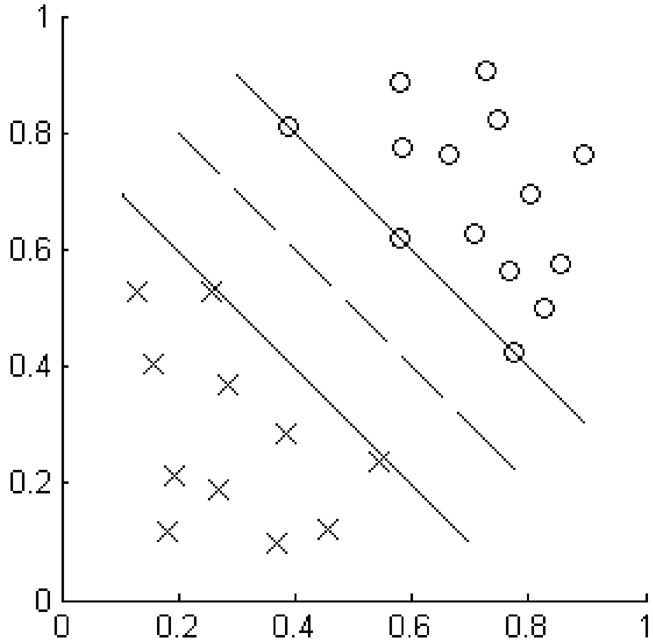


Figure 3.2: Illustration of Regular SVM

positive data (relevant images) being overwhelmed by the negative data (non-relevant images) [9].

3.2.2 One-class Support Vector Machine

One-class SVMs are derived from regular SVMs for solving density estimation problem. In typical formulation of 1-SVMs, only positive data are considered for estimating the density of the data. There are several kinds of different formulations of 1-SVMs in the literature. Here, we choose to illustrate the sphere-based approach with an explicit and good geometric property. In this approach, the goal is to construct a decision hypersphere that includes most of the positive data and minimizes the size of the hypersphere. Fig. 3.3 illustrates an example of 1-SVMs. It illustrates the sphere hyperplane in 1-SVM for constructing the smallest soft sphere that contains most of the positive data. The circles outside of the hyperplane are called outliers.

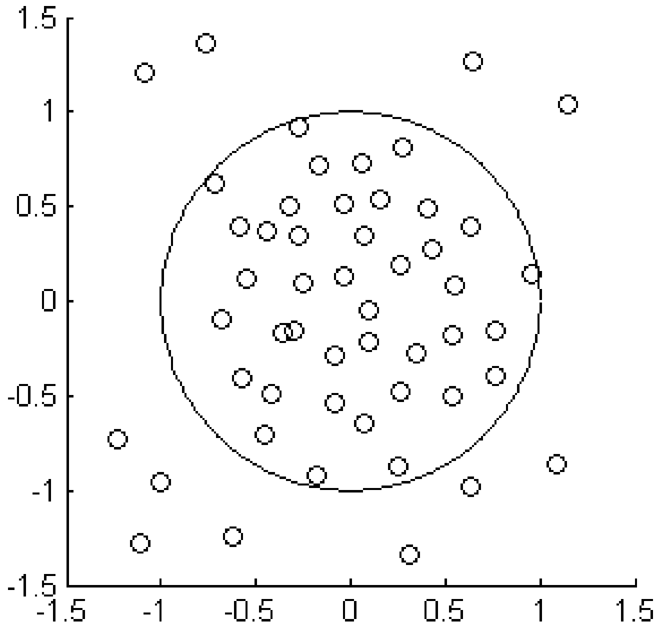


Figure 3.3: Illustration of One-class SVM

The optimal decision function of the sphere-based approach of 1-SVMs can be found by solving the optimization problem as follows,

$$\min_{R\in\mathbb{R},\mathbf{c}\in\mathcal{F}} \quad R^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i, \tag{3.5}$$

$$s.t. \quad ||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 \leq R^2 + \xi_i, \tag{3.6}$$

$$\xi_i \geq 0, \tag{3.7}$$

where ξ_i are the slack variables for margin error, $\mathbf{c}$ and R are the center and radius of the hypersphere, and $\nu \in [0, 1]$ is a parameter to control the tradeoff between the radius of the hypersphere and the number of positive training samples.

The one-class SVM is applied in relevance feedback to avoid the imbalanced dataset problem. The one-class SVM techniques treat the relevance feedback problem as estimating the density of the relevant images. It only considers the relevant images labeled by the user, and ignores the non-relevant images. However, when large portion of images are non-relevant to the query's target, the one-class SVM cannot work well without the help of non-relevant images [82].

3.3 Biased Support Vector Machine

In order to incorporate the negative information, we propose the Biased Support Vector Machine derived from 1-SVMs for overcoming the imbalance dataset problem of relevance feedback tasks. Our strategy is to describe the data by employing a pair of sphere hyperplanes in which the inner one captures most of the positive samples while the outer one pushes out the negative samples. Therefore, the goal of our problem is to find an optimal sphere hyperplane which not only can contain most of positive data but also can push most of negative data out of the sphere. The problem can be visually illustrated in Fig. 3.4. The dashed sphere in the figure is the desired spherehyperplane. The task can be formulated as an optimization problem and the mathematical formulation of our technique is given as follows.

Let us consider the training data:

$$(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n) \in \mathbb{R}^d \times Y, Y \in \{1, -1\} \tag{3.8}$$

where n is the number of training samples and d is the dimension of the input space.

The objective function for finding the optimal sphere hyperplane

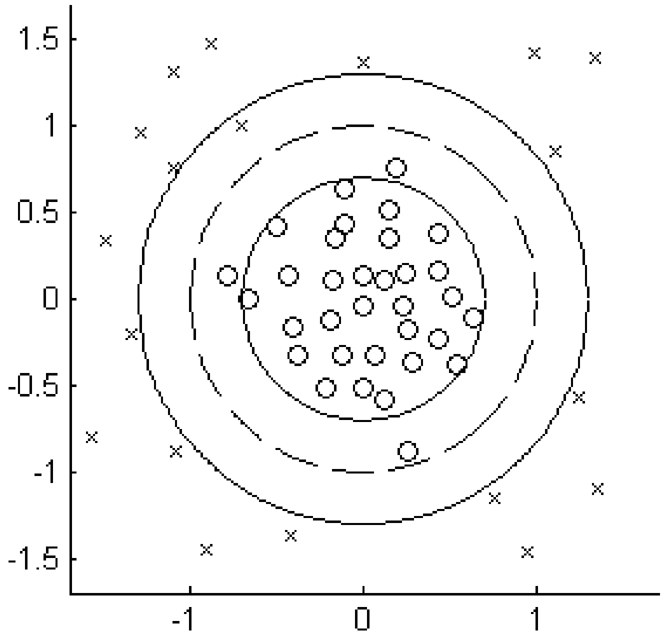


Figure 3.4: Illustration of Biased Support Vector Machine

can be formulated as,

$$\min_{R\in\mathbb{R},\xi\in\mathbb{R},\rho\in\mathbb{R}} \quad bR^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i, \tag{3.9}$$

$$s.t. \quad y_i(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2) \leq -\rho + \xi_i, \tag{3.10}$$

$$\xi_i \geq 0, \tag{3.11}$$

$$\rho \geq 0, \tag{3.12}$$

$$0 \leq \nu \leq 1, \tag{3.13}$$

where ξ_i are the slack variables for margin error, $\Phi(\mathbf{x_i})$ is the mapping function, $\mathbf{c}$ and R are the center and radius of the optimal hypersphere, ρ is the width of the margin, b is a parameter to control the bias, and $\nu \in [0, 1]$ is a parameter to control the tradeoff between the number of support vectors and margin errors. In the objective function Eq.(3.9), the term bR^2 is used to minimize the volume of the hypersphere, the term ρ is used to maximize the width of the margin, and the term $\frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$ is used to minimize the error.

The optimization task can be solved by introducing the Lagrange multipliers,

$$\begin{aligned} L(R,\xi,c,\alpha,\beta,\lambda) = bR^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\beta_i\xi_i - \lambda\rho \\ + \sum_{i=1}^{n}\alpha_i[y_i(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2) + \rho - \xi_i], \end{aligned} \tag{3.14}$$

where α_i, β_i, and λ are the Lagrange multipliers.

The objective function L reaches its minimum when its partial derivatives equal to 0. Let us take the partial derivative of L with respect to R, ξ_i, $\mathbf{c}$, and ρ. The partial derivative of L with respect to R is,

$$\frac{\partial L}{\partial R} = 0 \tag{3.15}$$

$$2bR - \sum_{i=1}^{n} 2y_i\alpha_i R = 0 \tag{3.16}$$

$$\sum_{i=1}^{n} y_i\alpha_i = b \tag{3.17}$$

The partial derivative of L with respect to ξ_i is,

$$\frac{\partial L}{\partial \xi_i} = 0 \tag{3.18}$$

$$\frac{1}{n\nu} - \beta_i - \alpha_i = 0 \tag{3.19}$$

Since $\beta_i \geq 0$, we obtain,

$$0 \leq \alpha_i \leq \frac{1}{n\nu} \tag{3.20}$$

The partial derivative of L with respect to $\mathbf{c}$ is,

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{c}} &= 0 && (3.21)\\ \sum_{i=1}^{n} 2\alpha_i y_i(\Phi(\mathbf{x_i}) - \mathbf{c}) &= 0 && (3.22)\\ \mathbf{c}\sum_{i=1}^{n} \alpha_i y_i &= \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x_i}) && (3.23)\\ \mathbf{c} &= \frac{1}{b}\sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x_i}) && (3.24) \end{aligned}$$

The partial derivative of L with respect to ρ is,

$$\begin{aligned} \frac{\partial L}{\partial \rho} &= 0 && (3.25)\\ -1 - \lambda + \sum_{i=1}^{n} \alpha_i &= 0 && (3.26) \end{aligned}$$

Since $\lambda \geq 0$, we obtain,

$$\sum_{i=1}^{n} \alpha_i \geq 1 \tag{3.27}$$

By summarizing the above equations, we obtain,

$$\sum_{i=1}^{n} y_i \alpha_i = b, \tag{3.28}$$

$$0 \leq \alpha_i \leq \frac{1}{n\nu}, \tag{3.29}$$

$$\mathbf{c} = \frac{1}{b}\sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x_i}), \tag{3.30}$$

$$\sum_{i=1}^{n} \alpha_i \geq 1. \tag{3.31}$$

By substituting the above derived results to the objective function in Eq. (3.14), the dual of the primal optimization can be shown to take the form

$$\begin{aligned}
\max_{\alpha} \quad & \sum_i \alpha_i y_i k(\mathbf{x_i}, \mathbf{x_i}) - \frac{1}{b} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x_i}, \mathbf{x_j}) && (3.32)\\
s.t. \quad & \sum_i \alpha_i y_i = b, && (3.33)\\
& 0 \leq \alpha_i \leq \frac{1}{n\nu}, && (3.34)\\
& \sum_i \alpha_i \geq 1 && (3.35)
\end{aligned}$$

where k is the mapping function corresponds to Mercer kernel $\mathbf{x}$. This dual problem can be solved with quadratic programming techniques. The decision function is defined by,

$$f(x) = \mathrm{sgn}(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2). \tag{3.36}$$

where $\mathbf{c}$ can be obtained from Eq.(3.30), and R can be solved by the support vectors. Based on this decision function, the data point that lie inside the classifier will be predicted as positive, and negative otherwise.

3.4 Interpretation of parameters in BSVM

In order to provide more natural interpretation for the parameters in BSVM, the formulation of BSVM follows the ν-SVM instead of the classical ϵ-SVM. To formulate it, let us first define the term margin error. The data points with $\xi_i \geq 0$, that are either errors or lie within margin, are considered as margin error. Formally, the fraction of margin errors is defined by,

$$\frac{1}{n}|\{i|y_i \times f(x_i) < \rho\}|. \tag{3.37}$$

Proposition 1 *Suppose BSVM is applied on some data, the following statements hold,*

1. ν *is an upper bound on the fraction of margin errors.*

2. ν *is a lower bound on the fraction of support vectors.*

3. *BSVM turns to regular SVM when* b *tends to zero.*

4. *BSVM turns to one-class SVM when* b *tends to its maximum.*

Proof 1 1. *By KKT conditions,* $\rho > 0$ *implies* $\lambda = 0$. *Hence the inequality Eq.(3.31) becomes an equality. Thus, at most* $n\nu$ *examples can have* $\alpha_i = \frac{1}{n\nu}$. *All examples with* $\xi_i > 0$ *do satisfy* $\alpha_i = \frac{1}{n\nu}$, *because* α_i *could grow to reduce* ξ_i *if not. Since examples are margin errors have* $\xi_i > 0$, *the fraction of margin error is upper bounded by* ν.

2. *Support vectors can contribute at most* $\frac{1}{n\nu}$ *from Eq.(3.29). Hence there must be at least* $n\nu$ *of them from Eq.(3.31). Thus,* ν *is an lower bound on the fraction of support vectors.*

3. *By replacing* y_i *with +1 for the positive class and -1 for the negative one, the constraint in Eq. (3.35) can be written as*

$$\sum_{i \in S^+} \alpha_i - \sum_{i \in S^-} \alpha_i = b, \tag{3.38}$$

where S^+ *denotes the positive class and* S^- *denotes the negative one. When* b *tends to zero, we have,*

$$\sum_{i \in S^+} \alpha_i = \sum_{i \in S^-} \alpha_i. \tag{3.39}$$

The constraint is the same as the one in ν*-SVM, and it makes* *the positive and negative classes have the same importance. Thus, BSVM turns to a regular SVM when b tends to zero.*

4. *When b tends to its maximum max_b with respects to ν, we have*

$$\sum_{i \in S^+} \alpha_i - \sum_{i \in S^-} \alpha_i = max_b, \tag{3.40}$$

Since all α_i of negative examples must take their minimums, in order to satisfy $b = max_b$. The α_i of negative examples reach their minimum when $\alpha_i = 0$, so that the negative examples are ignored in constructing the BSVM classifier. Thus, the BSVM turns to a one-class SVM when b tends to its maximum.

We illustrate the effect of parameter b in BSVM on the construction of decision boundaries in Fig 3.5. The circles and crosses represent the positive and negative data respectively. The boundaries of the shadow regions represent the decision boundaries. We can see that when b tends to zero, the behavior of BSVM is similar to regular SVM. When we increase the value of b, the behavior of BSVM is similar to 1-SVM.

3.5 Soft Label Biased Support Vector Machine

In the relevance feedback problem, the user's feedback is restricted to be either relevant or non-relevant. In various relevance feedback systems, the user give rating as feedback to the system. In order to support this type of feedback, we propose the soft label BSVM, which allows the class label to be a real number ranged from -1 to 1.

Let us consider the training data:

$$(\mathbf{x_1}, s_1), \ldots, (\mathbf{x_n}, s_n) \in \mathbb{R}^d \times S, S \in [1, -1] \tag{3.41}$$

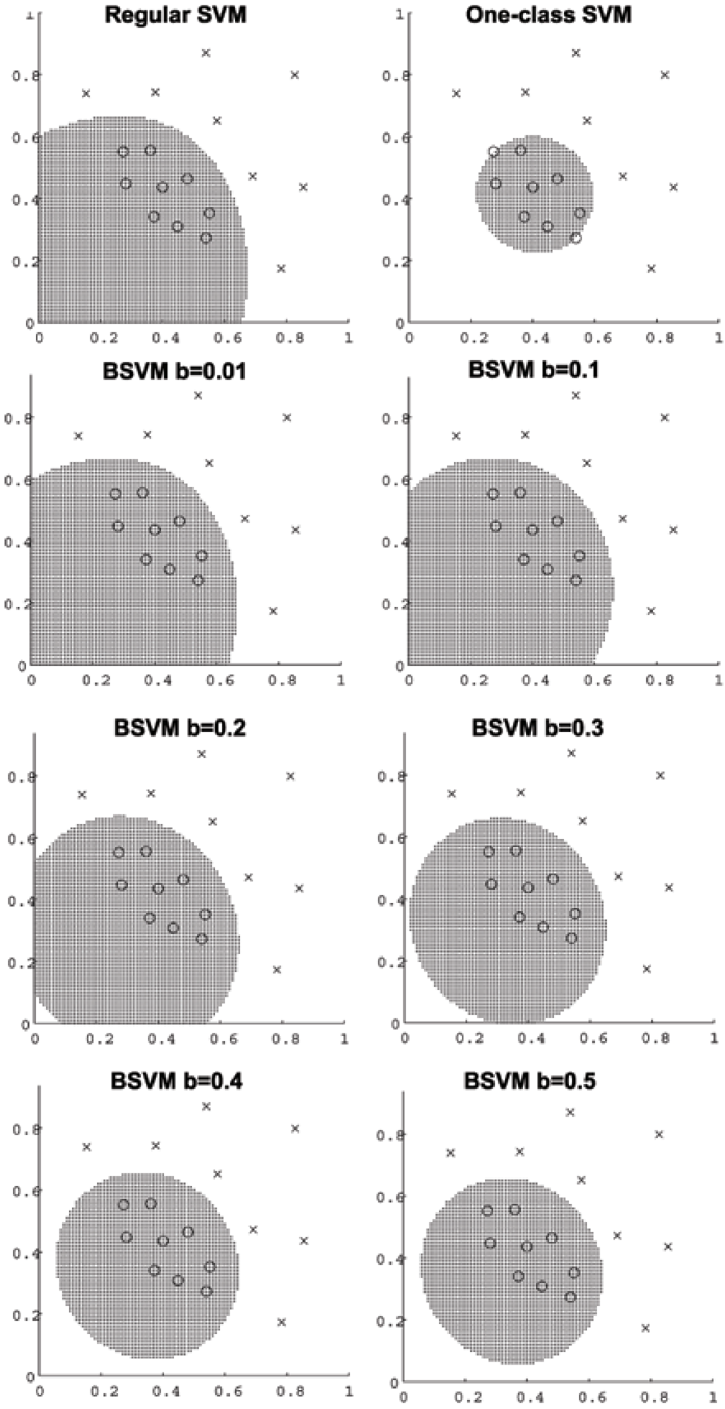


Figure 3.5: Illustration of parameter b in BSVM

where n is the number of training samples, d is the dimension of the input space, and s_i is the score given to the data point $\mathbf{x_1}$. The magnitude of s_i indicates the importance of the corresponding data point. Examples with $s_i > 0$ are expected to lie inside the classifier, and examples with $s_i < 0$ are expected to lie outside the classifier.

The objective function for finding the optimal sphere hyperplane can be formulated as,

$$\min_{R\in\mathbb{R},\xi\in\mathbb{R},\rho\in\mathbb{R}} \quad bR^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n} y_i s_i \xi_i, \tag{3.42}$$

$$s.t. \quad y_i(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2) \leq -y_i s_i \rho + \xi_i, \tag{3.43}$$

$$b \geq 0, \tag{3.44}$$

$$\xi_i \geq 0, \tag{3.45}$$

$$\rho \geq 0, \tag{3.46}$$

$$0 \leq \nu \leq 1, \tag{3.47}$$

$$y_i = 1 \text{ if } s_i \geq 0, \text{ and} \tag{3.48}$$

$$y_i = -1 \text{ otherwise}, \tag{3.49}$$

where ξ_i are the slack variables for margin error, $\Phi(\mathbf{x_i})$ is the mapping function, $\mathbf{c}$ and R are the center and radius of the optimal hypersphere, ρ is the width of the margin, b is a parameter to control the bias, and $\nu \in [0, 1]$ is a parameter to control the tradeoff between the number of support vectors and margin errors.

The optimization task can be solved by introducing the Lagrange multipliers,

$$\begin{aligned} L(R,\xi,\mathbf{c},\alpha,\beta,\lambda) = bR^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n} y_i s_i \xi_i - \sum_{i=1}^{n}\beta_i\xi_i - \lambda\rho \\ + \sum_{i=1}^{n}\alpha_i[y_i(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2) + y_i s_i \rho - \xi_i], \end{aligned} \quad (3.50)$$

where α_i, β_i, and λ are the Lagrange multipliers.

Let us take the partial derivative of L with respect to R, ξ, $\mathbf{c}$ and ρ respectively. By setting their partial derivatives to 0, we obtain the following equations,

$$2R(b - \sum_{i=1}^{n} y_i\alpha_i) = 0 \;\Rightarrow\; \sum_{i=1}^{n} y_i\alpha_i = b, \quad (3.51)$$

$$\frac{y_i s_i}{n\nu} - \alpha_i - \beta_i = 0 \;\Rightarrow\; 0 \le \alpha_i \le \frac{y_i s_i}{n\nu}, \quad (3.52)$$

$$\sum_{i=1}^{n} 2\alpha_i y_i(\Phi(\mathbf{x_i}) - \mathbf{c}) = 0 \;\Rightarrow\; \mathbf{c} = \frac{1}{b}\sum_{i=1}^{n}\alpha_i y_i \Phi(\mathbf{x_i}), \quad (3.53)$$

$$-1 + \sum_{i=1}^{n} y_i s_i \alpha_i - \lambda = 0 \;\Rightarrow\; \sum_{i=1}^{n} y_i s_i \alpha_i \ge 1. \quad (3.54)$$

By substituting the above derived results to the objective function in Eq. (3.50), the dual of the primal optimization can be shown to take the form

$$\max_{\alpha} \quad \sum_i \alpha_i y_i k(\mathbf{x_i},\mathbf{x_i}) - \frac{1}{b}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(\mathbf{x_i},\mathbf{x_j}) \quad (3.55)$$

$$s.t. \quad \sum_i \alpha_i y_i = b, \quad (3.56)$$

$$0 \le \alpha_i \le \frac{y_i s_i}{n\nu}, \quad (3.57)$$

$$\sum_i y_i s_i \alpha_i \ge 1 \quad (3.58)$$

This dual problem can be solved with quadratic programming techniques.

The main difference between the original BSVM and soft label BSVM is in the constraints Eq(3.57) and Eq(3.58). With these two constrains, the examples with smaller score in magnitude have smaller α_i. It means that these examples have smaller influence on the constructing of decision boundary. In this way, we can control the importance of different examples with the score s_i.

3.6 Interpretation of parameters in Soft Label BSVM

As in the BSVM, the parameters in soft label provide natural interpretation for the parameters in it.

Proposition 2 *Suppose soft label BSVM is applied on some data, the following statements hold,*

1. $\frac{\nu}{min_s}$ *is an upper bound on the fraction of margin errors, where* min_s *is the minimum value of the magnitude of* s_i.
2. ν *is a lower bound on the fraction of support vectors.*
3. *All examples with* $s_i = 0$ *are ignored by the machine.*

Proof 2

1. *By KKT conditions,* $\rho > 0$ *implies* $\lambda = 0$*. Hence the inequality Eq.(3.54) becomes an equality. At most* $\frac{n\nu}{min_s}$ *examples can have* $\alpha_i = \frac{min_s}{n\nu}$*. All examples with* $\xi_i > 0$ *do satisfy* $\alpha_i = \frac{s_i}{n\nu}$*, it is because* α_i *could grow to reduce* ξ_i *if not. Since examples are margin errors have* $\xi_i > 0$*, all margin errors have* $\alpha_i \geq \frac{min_s}{n\nu}$*. Thus, the fraction of margin error is upper bounded by* $\frac{\nu}{min_s}$.
2. *Support vectors can contribute at most* $\frac{1}{n\nu}$ *from Eq.(3.52). Hence there must be at least* $n\nu$ *of them from Eq.(3.54). Thus,* ν *is an lower bound on the fraction of support vectors.*

3. For any examples with $s_i = 0$, the constraint in Eq.(3.43) becomes,

$$y_i(||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 - R^2) \leq \xi_i. \tag{3.59}$$

The error of the example is controlled by slack variable ξ_i solely. However, the machine is minimizing $y_i s_i \xi_i$ in the objective function Eq.(3.42), and this term become 0 when $s_i = 0$. Thus, all examples with $s_i = 0$ are ignored by the machine.

3.7 Relevance Feedback Using Biased Support Vector Machine

3.7.1 Advantages of BSVM in Relevance Feedback

From the above formulation, one may see that the optimization equation is similar to the one in the ν-SVM. Now, we explain the mathematical differences compared with regular SVMs and the advantages of our BSVM from the geometric perspective for solving the relevance feedback problems.

From the results of mathematic deduction in the optimization function, we see that BSVM is with the following constraint from Eq. (3.35),

$$\sum_i \alpha_i y_i = b, \tag{3.60}$$

When replacing y_i with $+1$ for the positive class and -1 for the negative one, the constraint can be written as

$$\sum_{i \in S^+} \alpha_i - \sum_{i \in S^-} \alpha_i = b, \tag{3.61}$$

where S^+ denotes the positive class and S^- denotes the negative one.

However, in the regular SVMs (ν-SVM), the constraint is with the form

$$\sum_{i \in S^+} \alpha_i - \sum_{i \in S^-} \alpha_i = 0. \tag{3.62}$$

The difference indicates that the weight allocated to the positive support vectors in BSVM will be larger than the negative ones when setting a positive bias factor b. This can be useful for solving the imbalance dataset problem. However, regular SVMs (ν-SVM) treat the two classes without any bias which is not effective enough to model the relevance feedback problem.

Moreover, we can also see the difference from the geometric perspective. Fig. 3.6, Fig. 3.7 and Fig. 3.8 provide the comparison of the decision boundaries of regular SVM, 1-SVM and BSVM on the synthetic data with the same kernels (Radial Basis Function) and parameters (ρ=0.1). The circles and crosses represent the positive and negative data respectively. The boundaries of the shadow regions represent the decision boundaries. We can see that the geometric property of BSVM is better than the regular SVM and 1-SVM. BSVM can describe the data in a cluster behavior by the sphere based boundary and can flexibly control the weight of the positive class for the imbalanced dataset by adjusting the bias factor. Therefore, compared with regular SVM and 1-SVM, BSVM is more reasonable and effective to model the relevance feedback tasks.

3.7.2 Relevance Feedback Algorithm By BSVM

From the above comparisons, we have shown the benefits of BSVM for solving relevance feedback issues. Here, we describe how to formulate the relevance feedback algorithm by employing the BSVM technique. Applying SVMs based techniques in relevance feedback is similar to the

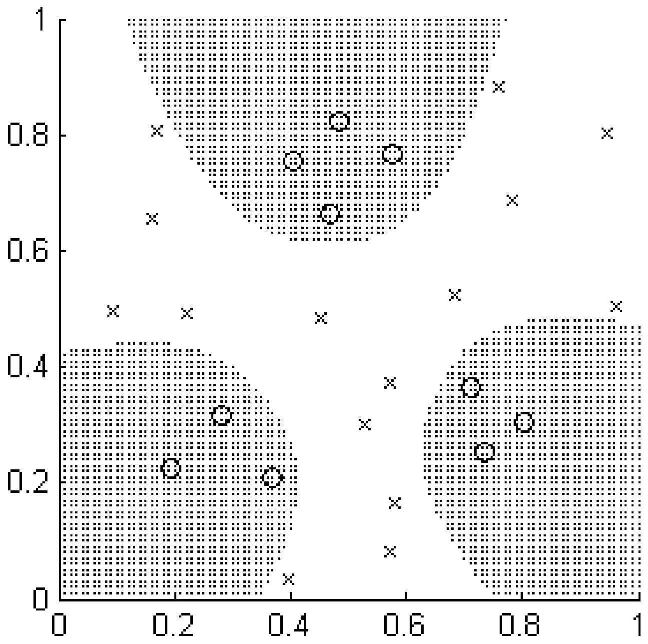


Figure 3.6: Decision Boundary of Regular SVM

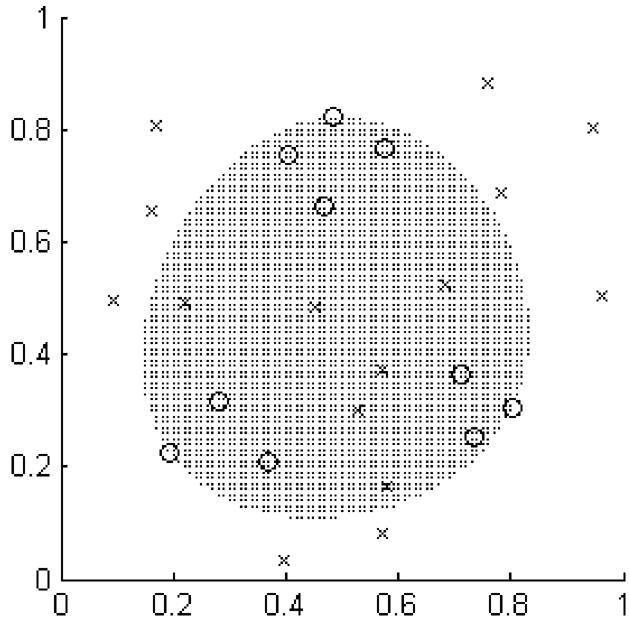


Figure 3.7: Decision Boundary of One-class SVM

classification task. However, the relevance feedback need to construct the evaluation function to output the relevance value of the retrieval

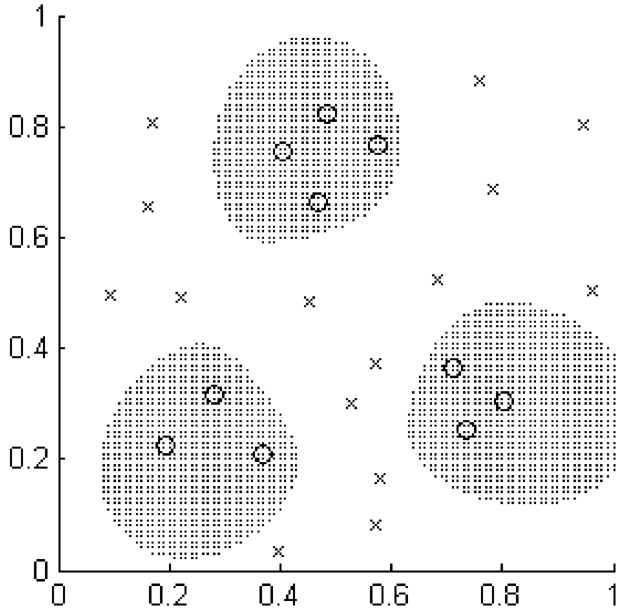


Figure 3.8: Decision Boundary of BSVM

instances. From the decision function, we build the evaluation function with the similar form by substituting the equation in

$$f(\mathbf{x_i}) = R^2 - ||\Phi(\mathbf{x_i}) - \mathbf{c}||^2 \tag{3.63}$$

where the center $\mathbf{c}$ can be solved by a set of support vectors. However, for the relevance evaluation purpose, constant values can be eliminated. Hence, the evaluation function can be shown to take the concise form

$$f(\mathbf{x_i}) = \frac{2}{b} \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x}). \tag{3.64}$$

Once the parameters α_i are solved, the evaluation function can be constructed. Consequently, we can rank the images based on the scores of the evaluation function $f(\mathbf{x_i})$. The images with higher scores will be more likely be chosen as the targets.

3.8 Experiments

Here, we present the experimental results of our BSVM both on the synthetic data and the real-world images. The relevance feedback systems involved in the experiment are listed on Table 3.1. The ν-SVM in the experiment is equivalent to the regular SVM. For the purpose of objective measure of performance, we assume that the query judgement is defined on the image categories. And the metric of evaluation is the Average Precision which is defined as the average ratio of the number of relevant images of the returned images over the number of total returned images.

Table 3.1: List of Relevance Feedback Systems in Experiment

Support Vector Machine	BSVM
Support Vector Machine	ν-SVM
Support Vector Machine	One-class SVM
Ad-hoc Re-weighting	MARS
Distance Optimization Approach	MindReader
Bayesian Approach	PicHunter
Density Estimation	Non Parametric
Density Estimation	Single Gaussian
Density Estimation	GMM

In the experiment, we evaluate the retrieval performance of various relevance feedback methods on CBIR. A category is first picked from the database randomly, and this category is assumed to be the user's query target. The system then improves retrieval results by relevance feedback. In each iteration of the relevance feedback process, five images are picked from the database and labelled as either relevant or non-relevant based on the ground truth of the database. For the first iteration, two relevant images and three non-relevant images are randomly picked, and all methods are run based on the same set of initial data points. For the iterations afterward, each method selects five images based on their own display set selection algorithm. The precision of each method is then recorded, and the whole process is repeated for 200 times to produce the average precision in each iteration for each method.

For the SVM-based techniques in the experiment, we implement the algorithms by modifying the codes in the *libsvm* library [8]. We notice that the experimental settings are important to impact on the evaluation results. To enable an objective measure of performance without bias, we choose the same kernel and parameters for all SVM-based methods. In order to select the best kernel function for the current dataset, we performed an experiment to evaluate the performance of different kernels. The kernel functions involved in the experiment are listed on Table 3.2. The image dataset used in this experiment are chosen from the COREL image collection. The datasets is with 20 categories (20-Cat). Each category includes 100 images belonging to a same semantic class. We evaluate the performance of different kernel functions by measuring their average precision on the top 10 retrieval result.

Table 3.2: List of Kernel Functions in Experiment

Radial Basis Function
Sigmoid Function
Polynomial Function
Linear Function

Fig. 3.9 show the evaluation result on different kernel functions. The Radial Basis Function (RBF) outperforms other kernels in the experiments, and it is a common practice to use RBF as the kernel function on the image-based experiment. Thus, we choose the RBF as

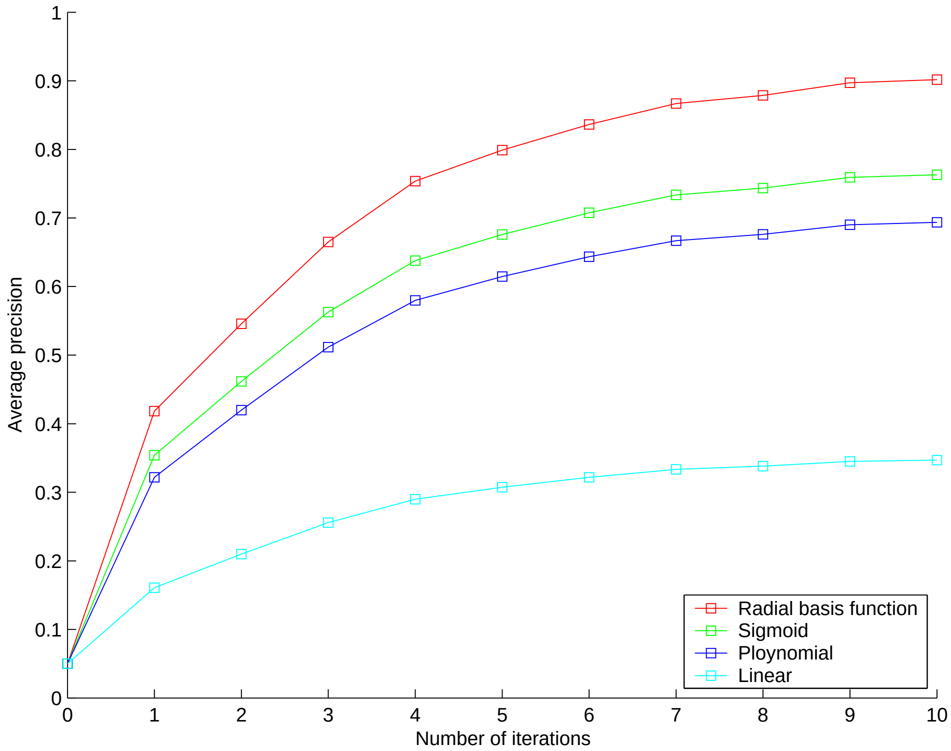


Figure 3.9: Retrieval performance of different kernel functions

the kernel function in the other experiments.

3.8.1 Synthetic Dataset

We generate a synthetic dataset to simulate the real-world image dataset. The dataset consists 40 categories each of them contains 100 data points randomly generated by seven Gaussians in a 40-dimensional space. The means and covariance matrices of the Gaussians for each category are randomly generated from the range of [0,10].

Fig. 3.10 and Fig. 3.11 show the evaluation result on the synthetic dataset.

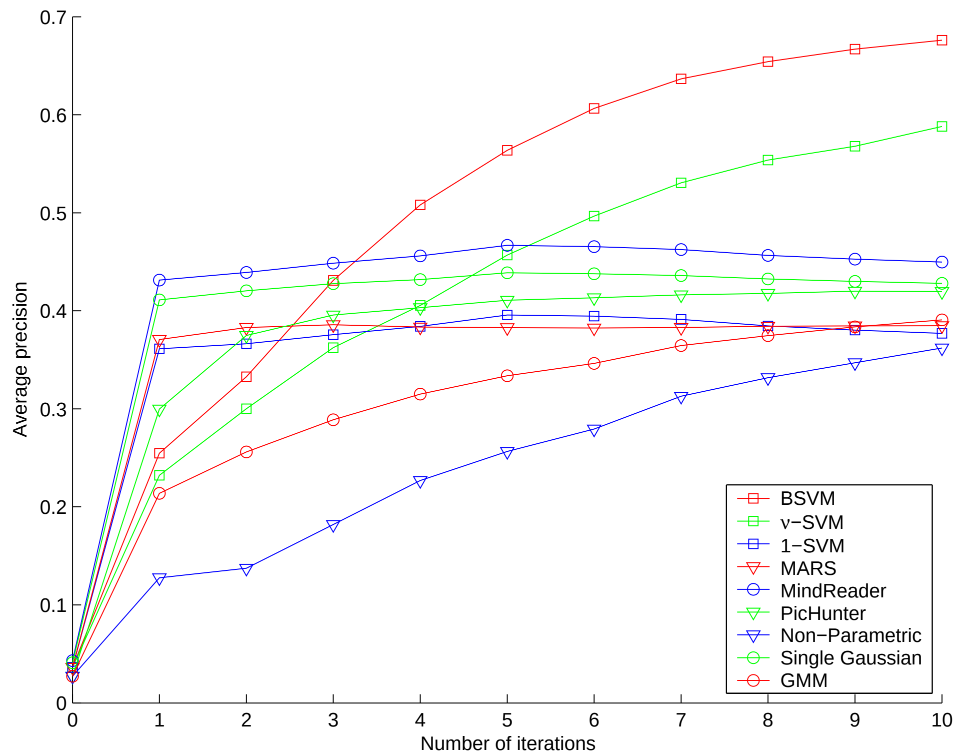


Figure 3.10: Top 10 average precision on synthetic dataset

3.8.2 Real-World Dataset

The real-world images are chosen from the COREL image collection. We organize two datasets containing various images with different semantic meanings, such as antique,aviation, balloon, botany, butterfly, car and cat, etc. One of the datasets is with 20 categories (20-Cat) and another is with 50 categories (50 -Cat). Each category includes 100 images belonging to a same semantic class.

The dataset used in the experiment is the real-world images chosen from the COREL image collection. We organize two datasets containing various images with different semantic meanings, such as antique,aviation, balloon, botany, butterfly, car and cat, etc. One of the datasets is with 20 categories (20-Cat) and another is with 50 cate-

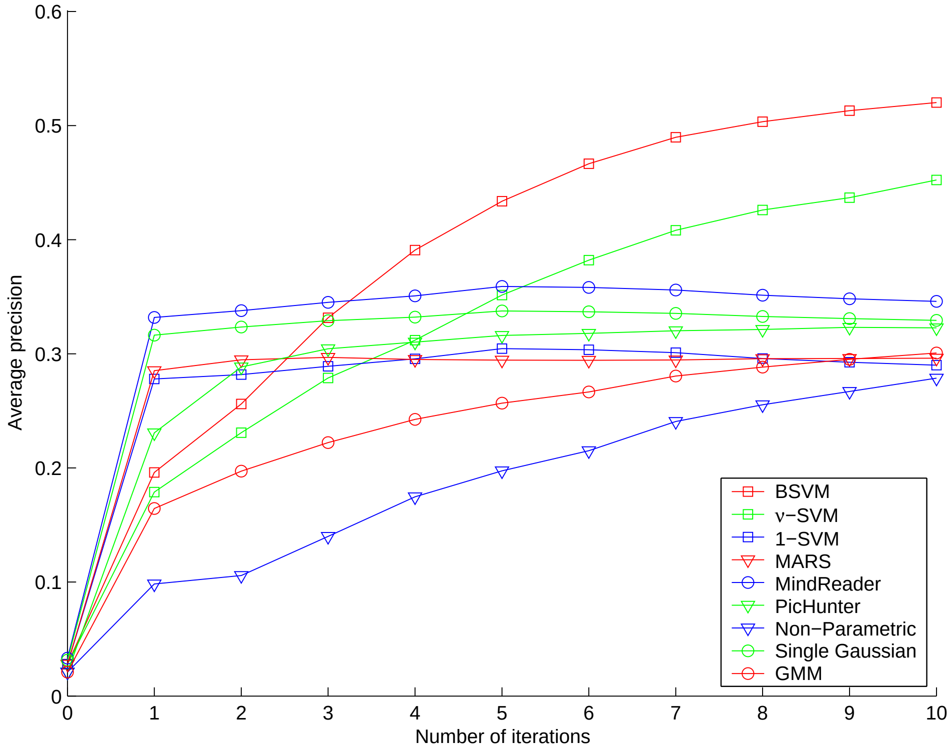


Figure 3.11: Top 30 average precision on synthetic dataset

gories (50 -Cat). Each category includes 100 images belonging to a same semantic class.

For the real-world image retrieval, the image representation is an important step for evaluating the relevance feedback algorithms. We extract three different features to represent the images: color, shape and texture. The color feature engaged is the color moment since it is closer to human perception naturally. We extract three moments: color mean, color variance and color skewness in each color channel (H, S, and V), respectively [68]. Thus, 9-dimensional color moment is employed as the color feature in our experiments.

We employ the edge direction histogram as the shape feature in our experiments [35]. Canny edge detector is applied to obtain the edge images. From the edge images, the edge direction histogram can then be computed. The edge direction histogram is quantized into 18 bins of 20 degrees each, hence an 18-dimensional edge direction histogram is used to represent the edge feature.

We use the wavelet-based texture feature for its effectiveness. We perform the Discrete Wavelet Transformation (DWT) on the gray images employing a Daubechies-4 wavelet filter. In total, we perform 3-level decompositions and obtain ten subimages in different scales and orientations [67]. Then, we choose nine subimages with most of the texture information and compute the entropy of each subimage. Hence, a 9-dimensional wavelet-based texture feature is obtained to describe the texture information for each image.

Fig. 3.12 and Fig. 3.13 show the evaluation result on the 20-cat dataset. Fig. 3.14 and Fig. 3.15 show the evaluation result on the 50-cat dataset.

3.8.3 Experimental Results

With some minor exception, the result in these figures are homogeneous. It means that the best retrieval results are produced by the same relevance feedback system for the two datasets, and the same hold for the poorest one. From the experiment, it draws the following observations,

- The retrieval results of the included relevance feedback techniques have been improved after the first iteration. The retrieval result in the first iteration can be considered as the retrieval result of the traditional one-shot approach. Thus, it shows that the retrieval performance of CBIR can be improved with relevance feedback techniques.

- The relevance feedback techniques that assume the target dis-

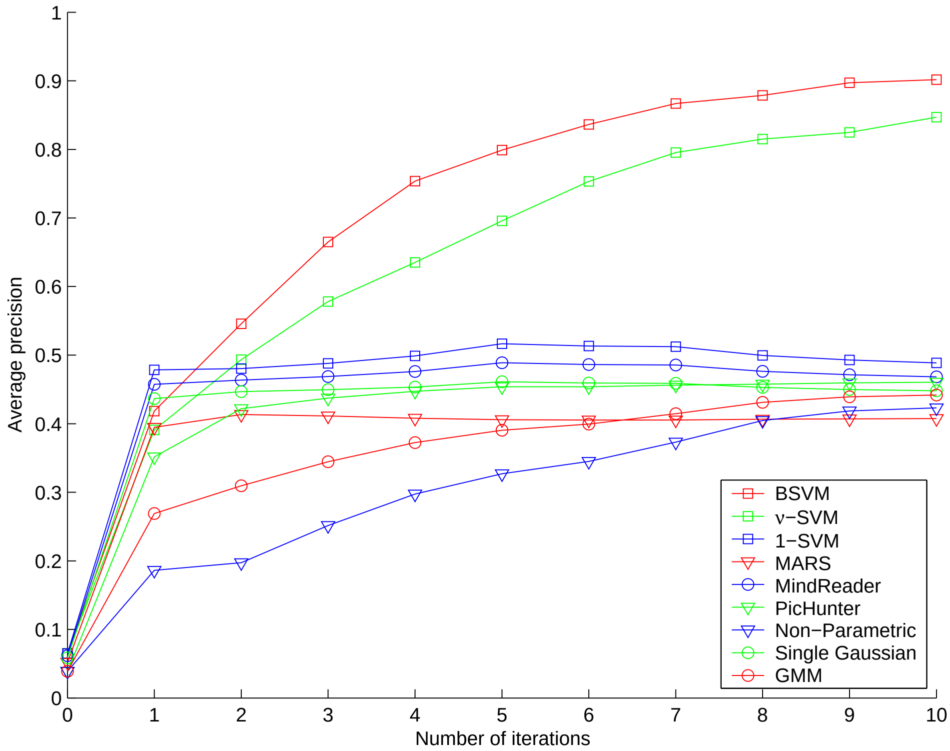


Figure 3.12: Top 10 average precision on 20-cat image dataset

tribution follows a single Gaussian distribution fail to improve the retrieval performance after the first few iterations. These techniques include MARS, Mindreader and single Gaussian approach. The reason behind it is that these techniques are able to retrieve the relevant images around the query, and fail to retrieve the relevant images that far away from it. Thus, these techniques are only able to retrieve the relevant images in a local area, and fail to retrieve other relevant images and improve the retrieval performance afterward.

- Some relevance feedback techniques, PicHunter, GMM and non-parametric approach, have relaxed the assumption on the target distribution. However, they cannot outperform the techniques

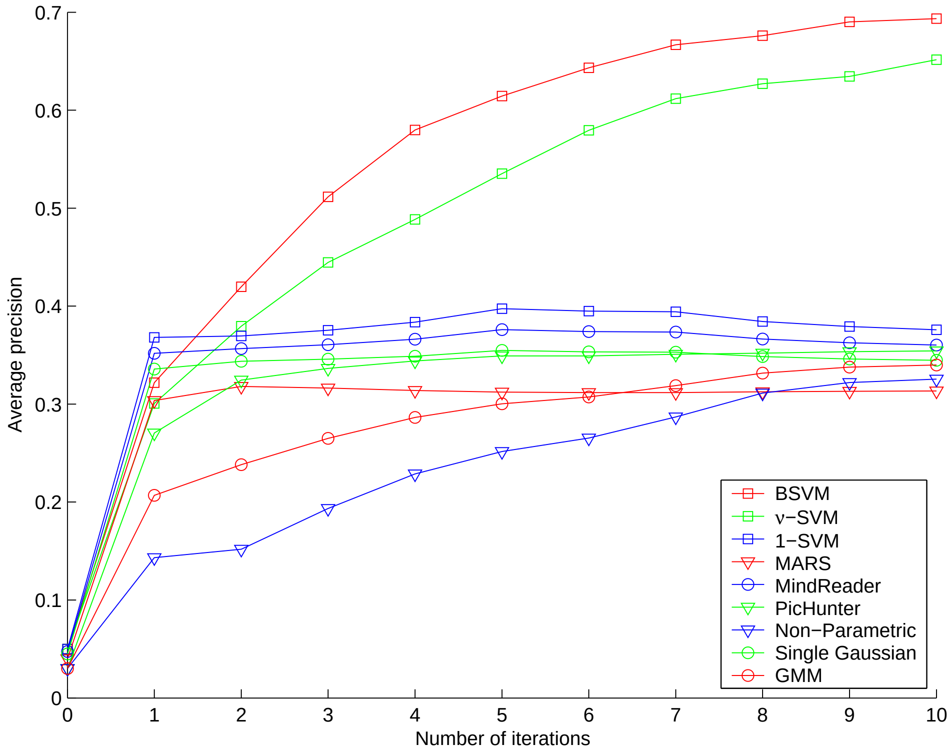


Figure 3.13: Top 30 average precision on 20-cat image dataset

with single Gaussian assumption. The reason behind it is that these techniques require a large amount of feedback data to provide sufficient statistical information to estimate the target distribution. However, the number of training samples in the relevance feedback process is usually small, and these techniques fail to estimate the target distribution in the relevance process.

- The SVM-based techniques, BSVM, ν-SVM and 1-SVM, perform better than other techniques in the experiment. SVM has a strong theoretical foundations and excellent empirical successes in pattern classification problem. The SVM maps the data points form the original vector space to a high dimensional vector space with the Mercer kernel. By using this technique, the SVM is able to

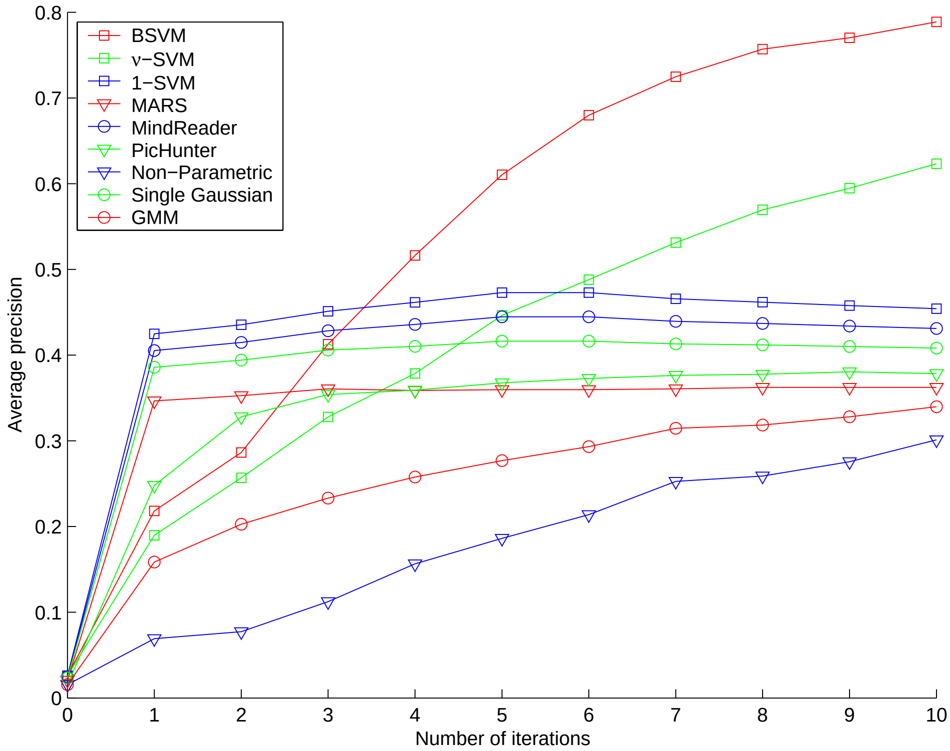


Figure 3.14: Top 10 average precision on 50-cat image dataset

address the non-linearity in the target distribution.

- BSVM outperforms the other approaches in the experiment. However, we notice that the performances of 1-SVM in the beginning feedback steps are better than those of other approaches. The reason is that 1-SVM can reach the enclosed positive region soon, but it cannot be further improved without the help of the negative information in further steps.

3.9 Conclusion

We have investigated SVM-based relevance feedback techniques for solving the relevance feedback problems in CBIR. We addressed the imbalanced dataset problem in relevance feedback and proposed a novel

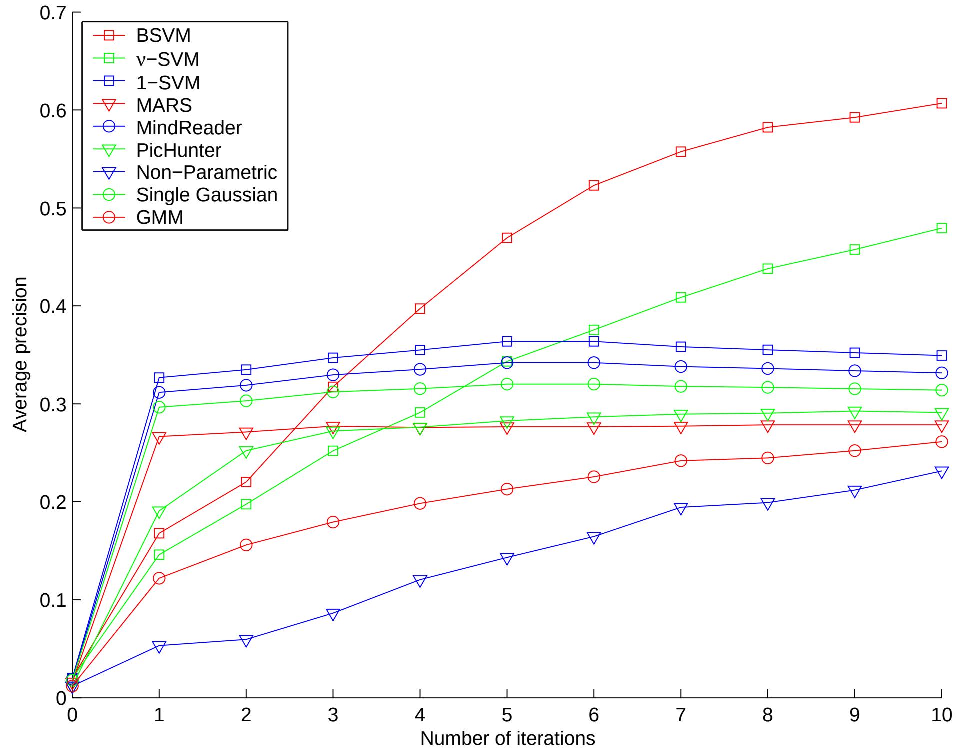


Figure 3.15: Top 30 average precision on 50-cat image dataset

relevance feedback technique with Biased Support Vector Machine. The advantages of our proposed techniques are explained and demonstrated compared with traditional approaches. We performed the experiments both on synthetic data and real-world image datasets. The experimental results demonstrate that our BSVM based relevance feedback algorithm is effective and promising for improving the retrieval performance in CBIR.

□ **End of chapter.**

Chapter 4

Self-Organizing Map-based Inter-Query Learning

4.1 Motivation

In most of the relevance feedback systems, only the intra-query feedback information is used to learn the user's preference. However, a small training data set is difficult to provide enough statistical information for achieving this goal and providing good retrieval result. In order to address this problem, we use the inter-query information to modify the feature vector space and cluster the neurons with similar images together, so that the neurons are organized in a way that ease the process of intra-query learning. In the proposed approach, we update the similarity measure between images dynamically according to the feedback information given by each past query. It is achieved by further training the neurons on the SOM. Neurons representing relevant images are moved closer to the estimated user target and those represent non-relevant images are moved away from the estimated user target.

Figure 4.1 shows a 2-dimensional feature vector space of a collection of images with 4 different classes. A SOM is trained based on the underlying distribution. In analyzing the image data, images from the same class often form clusters which are sparse and irregular in shape. This makes the retrieval process more difficult to find target images. With the help of inter-query feedback information described above, we organize the feature vector space in a fashion that ease the retrieval process.

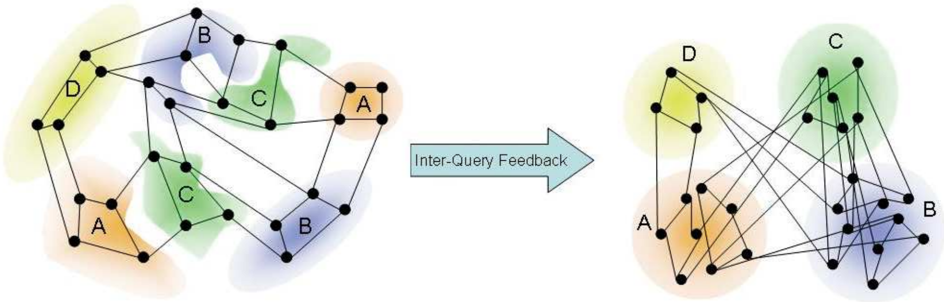


Figure 4.1: Illustration of SOM-based Inter-query learning

4.2 Algorithm

4.2.1 Initialization and Replication of SOM

In the preprocessing procedure, the system performs feature extraction on the images in the database, and uses a SOM to represent the distribution of the data. We perform a low-level feature extraction on the set of images in the database, and each image is then represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ in a high dimensional vector space. We construct and train a SOM M with feature vectors extracted from the images. After the SOM training, the model vectors in the neurons of M are arranged to match the distribution of the feature space. The model vectors $\mathbf{m}_i \in M$ of neurons in the SOM are used to partition the feature vector space based on the minimum distance classifier, each image is classified into different groups represented by $\mathbf{m}_i$. By doing so, we reduce the size of data from $|O|$ to $|M|$, where $|O|$ and $|M|$ are the number of images and neurons in the SOM respectively. The similarity measure between images is then defined as the Euclidean distance between the model vectors which represent them.

The relationship between the neurons and the images in the database depends on the coordinates of the model vectors, any changes on the model vectors of neurons may alter this relationship. Our proposed approach is to modifying the model vectors in the SOM to update the similarity measure. Thus, we duplicate another SOM from the original one. The new SOM contain a set of neurons with model vectors $\mathbf{m}'_i \in M'$ and has a one-to-one mapping, $f : M \to M'$, between the set M and M'. To obtain the set of images represented by model vector $\mathbf{m}'_i$, we can get the original model vector $\mathbf{m}_i$ by f^{-1}, and then by minimum distance classifier in M. Initially, the layout of the two SOMs are the same. We update the similarity measure by modify the model vectors in M' instead of M, so that the relationship between images in the database and the model vectors in M can be preserved during the whole learning process.

4.2.2 SOM Training for Inter-Query Learning

In order to update the similarity measure based on the inter-query feedback information, we modify the model vectors $\mathbf{m}'_i$ in the new SOM, such that neurons contain similar images as indicated in the feedback are moved closer to each others. The idea of this process is similar to the clustering algorithm in rival penalized competitive learning [38]. Consider that there are K past queries stored in the system, and inter-query information provided to the system is represented by $\{q^1, \ldots, q^K\}$. Each past query q^k is used to reorganize the vector space of the SOM, and improve the structure of data. Assume in the k-th query, the user marked a set of relevant images D_R^k and a set of non-relevant images D_N^k during the whole retrieval process, ${M'}_R^k$ and ${M'}_N^k$ are the corresponding sets of model vectors respectively. Let $\mathbf{c'}^k$ be the model vector with highest relevance score in Eq. (4.7), and it is most likely to be the user's target for that query. We then modify the model vectors with the following equations,

$$
\begin{aligned}
&\forall \mathbf{m} \in {M'}_R^k \\
&\qquad \mathbf{m} \;=\; \mathbf{m} + \alpha_R^k(\mathbf{c'}^k - \mathbf{m}), \qquad (4.1) \\
&\forall \mathbf{m} \in {M'}_N^k \\
&\qquad \mathbf{m} \;=\; \mathbf{m} + \alpha_N^k(\mathbf{m} - \mathbf{c'}^k), \qquad (4.2)
\end{aligned}
$$

where α_R^k and α_N^k are the learning rates and they are monotonic decreasing functions of k. Thus, neurons represent relevant images are moved closer to the estimated user's target and those represent non-relevant images are moved away from the estimated user's target. For a long run, the vector space will be modified, in which neurons represent the same image concept are clustered together.

In a SOM, the nearby neurons in the topology are representing similar units, so that the learning process can be improved by moving also the neurons near to the neurons in the sets ${M'}_R^k$ and ${M'}_N^k$. The idea of this process is similar to the SOM training process. The equations for modifying the model vectors are defined by,

$$
\begin{aligned}
&\forall \mathbf{m} \in N({M'}^{k}_{R}) \\
&\qquad \mathbf{m} \;=\; \mathbf{m} + h^{k}_{Ri}(\mathbf{c}'^{j} - \mathbf{m}), \qquad (4.3)\\
&\forall \mathbf{m} \in N({M'}^{k}_{N}) \\
&\qquad \mathbf{m} \;=\; \mathbf{m} + h^{k}_{Ni}(\mathbf{m} - \mathbf{c}'^{j}), \qquad (4.4)
\end{aligned}
$$

where $N(M)$ is the set of nearby neurons for M in the SOM topology, h^{k}_{Rci} and h^{k}_{Nci} are the neighborhood functions. The neighborhood functions are defined by

$$
\begin{aligned}
h^{k}_{Ri} &= \alpha^{k}_{R} \cdot \exp\left(-\frac{\mathrm{dis}(\mathbf{m}, {M'}^{k}_{R})}{2(\sigma^{k}_{R})^{2}}\right), \text{ and} \qquad (4.5)\\
h^{k}_{Ni} &= \alpha^{k}_{N} \cdot \exp\left(-\frac{\mathrm{dis}(\mathbf{m}, {M'}^{k}_{N})}{2(\sigma^{k}_{N})^{2})}\right), \qquad (4.6)
\end{aligned}
$$

where σ^{k}_{R} and σ^{k}_{N} are some monotonic decreasing functions of k, dis($\mathbf{m}$, ${M'}^{k}_{R}$) and dis($\mathbf{m}$, ${M'}^{k}_{N}$) denote the distance between the model vector $\mathbf{m}$ and the corresponding nearby neuron in the set ${M'}^{k}_{R}$ and ${M'}^{k}_{N}$ in the SOM topology respectively. Thus, the value of the neighborhood function for neuron $\mathbf{m}$ decreases as the distance dis($\mathbf{m}$, ${M'}^{k}_{N}$) increases.

4.2.3 Incorporate with Intra-Query Learning

In the intra-query learning process, the system presents a set of images D_t to the user in each iteration t, and the user gives response A_t by marking them as either relevant or non-relevant. The information provided in the k-th query at iteration t is represented by $q^{k}_{t} = \{D_1, A_1, \ldots, D_t, A_t\}$, and the system uses it to refine the query. We define D_R and D_N as the set of relevant images and the set of non-relevant images marked by the user from first iteration to the current iteration respectively. The sets D_R and D_N are then represented by the corresponding model vector set M'_R and M'_N. The BSVM in Section 3.3 is used to train a decision boundary to classify this two sets of model vectors.

In order to retrieve images from the database, we need to construct an evaluation function to output the relevance value of the neurons, and it is defined by,

$$g(\mathbf{m}'_i) = R^2 - ||\Phi(\mathbf{m}'_i) - \mathbf{c}||^2 \tag{4.7}$$

where $\mathbf{c}$ is the center of the sphere hyperplane of the BSVM. The center $\mathbf{c}$ can be solved by the set of support vectors, and the constant values can be eliminated. We can rank the neurons based on the scores of the evaluation function $g(\mathbf{m}'_i)$. The neurons with higher scores will be more likely to be chosen as the targets. The relevance score between an image and its corresponding neuron is measured by their Euclidean distance. Thus, we can rank the images in the database by combining it with the function $g(\mathbf{m}'_i)$.

4.3 Experiments

Here, we present the experimental results of our SOM-based inter-query learning both on the synthetic data and the real-world images. For the purpose of objective measure of performance, we assume that the query judgement is defined on the image categories. And the metric of evaluation is the Average Precision which is defined as the average ratio of the number of relevant images of the returned images over the number of total returned images.

In the experiment, we evaluate the retrieval performance of our

SOM-based inter-query learning technique by applying it to various intra-query learning techniques. The intra-query learning techniques involved in the experiment are listed on 4.1.

Table 4.1: List of Relevance Feedback Systems in Experiment

Support Vector Machine	BSVM
Support Vector Machine	ν-SVM
Support Vector Machine	One-class SVM
Ad-hoc Re-weighting	MARS

In the experiment, a category is first picked from the database randomly, and this category is assumed to be the user's query target. The system then improves retrieval results by relevance feedbacks. In each iteration of the relevance feedback process, five images are picked from the database and labelled as either relevant or non-relevant based on the ground truth of the database. For the first iteration, two relevant images and three non-relevant images are randomly picked, and all methods are run based on the same set of initial data points. For the iterations afterward, each method selects five images based on their own display set selection algorithm. The precision of each intra-query learning method is then recorded, and the whole process is repeated for 200 times to produce the average precision in each iteration for each method. After that, a SOM of size 30×30 is trained by using the the feature vectors of images in the database. Our SOM-based inter-query is then applied to reorganize the SOM with the 200 past queries. Finally, we generated another 200 queries and recorded the precision of each intra-query learning techniques after our SOM-based inter-query learning is applied. In the experiment, we implement the algorithm by modifying the codes in SOM Toolbox [30].

4.3.1 Synthetic Dataset

We generate a synthetic dataset to simulate the real-world image dataset. The dataset consists 40 categories each of them contains 100 data points randomly generated by seven Gaussians in a 40-dimensional space. The means and covariance matrices of the Gaussians for each category are randomly generated from the range of [0,10].

Fig. 4.2 and Fig. 4.3 show the evaluation result on the synthetic dataset.

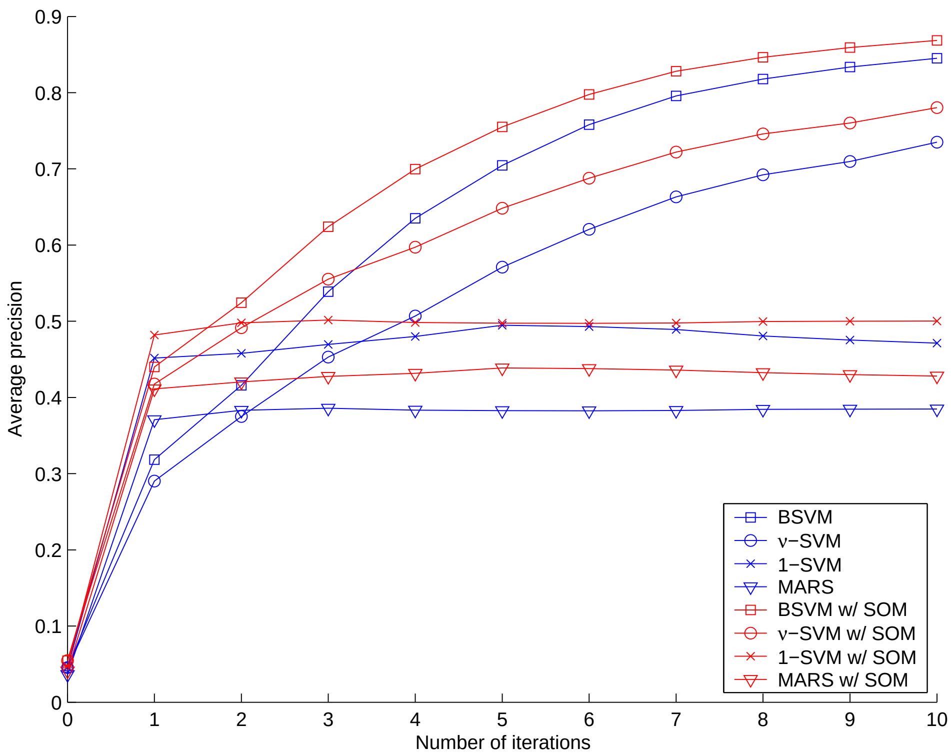


Figure 4.2: Top 10 average precision on synthetic dataset

4.3.2 Real-World Dataset

The real-world images are chosen from the COREL image collection. We organize two datasets containing various images with different se-

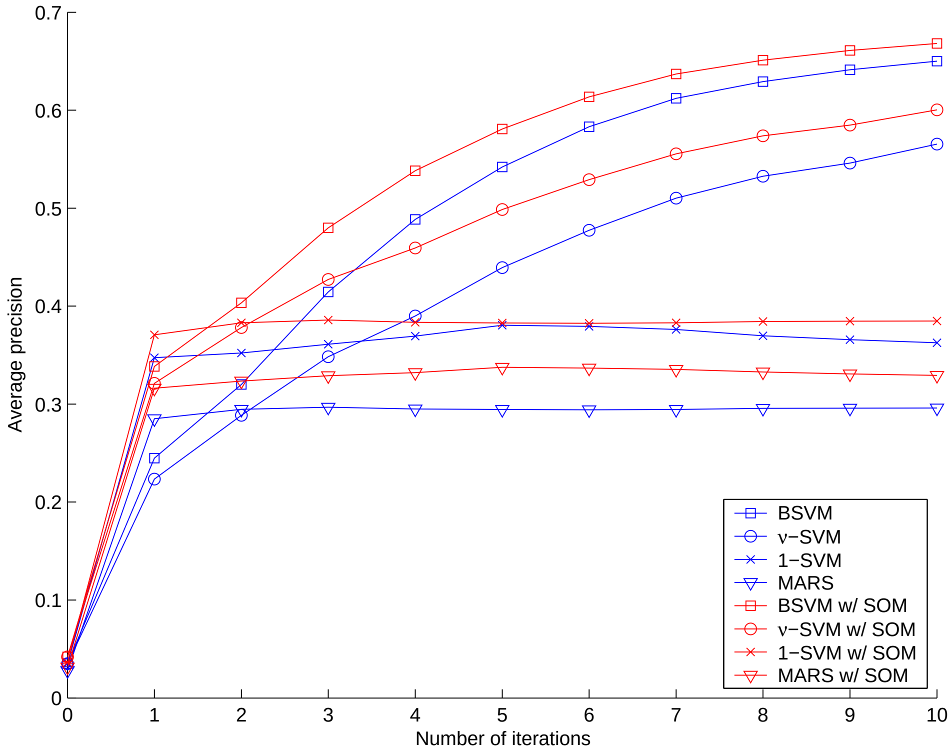


Figure 4.3: Top 30 average precision on synthetic dataset

mantic meanings, such as antique,aviation, balloon, botany, butterfly, car and cat, etc. One of the datasets is with 20 categories (20-Cat) and another is with 50 categories (50 -Cat). Each category includes 100 images belonging to a same semantic class.

The dataset used in the experiment is the real-world images chosen from the COREL image collection. We organize two datasets containing various images with different semantic meanings, such as antique,aviation, balloon, botany, butterfly, car and cat, etc. One of the datasets is with 20 categories (20-Cat) and another is with 50 categories (50 -Cat). Each category includes 100 images belonging to a same semantic class.

For the real-world image retrieval, the image representation is an important step for evaluating the relevance feedback algorithms. We extract three different features to represent the images: color, shape and texture. The color feature engaged is the color moment since it is closer to human perception naturally. We extract three moments: color mean, color variance and color skewness in each color channel (H, S, and V), respectively [68]. Thus, 9-dimensional color moment is employed as the color feature in our experiments.

We employ the edge direction histogram as the shape feature in our experiments [35]. Canny edge detector is applied to obtain the edge images. From the edge images, the edge direction histogram can then computed. The edge direction histogram is quantized into 18 bins of 20 degrees each, hence an 18-dimensional edge direction histogram is used to represent the edge feature.

We use the wavelet-based texture feature for its effectiveness. We perform the Discrete Wavelet Transformation (DWT) on the gray images employing a Daubechies-4 wavelet filter. In total, we perform 3-level decompositions and obtain 10 subimages in different scales and orientations [67]. Then, we choose 9 subimages with most of the texture information and compute the entropy of each subimage. Hence, a 9-dimensional wavelet-based texture feature is obtained to describe the texture information for each image.

Fig. 4.4 and Fig. 4.5 show the evaluation result on the 20-cat dataset. Fig. 4.6 and Fig. 4.7 show the evaluation result on the 50-cat dataset.

4.3.3 Experimental Results

With some minor exception, the result in these figures are homogeneous. From the experiment, all four intra-query techniques perform better when our SOM-based inter-query learning technique is applied, and it performs the best when incorporating with our BSVM.

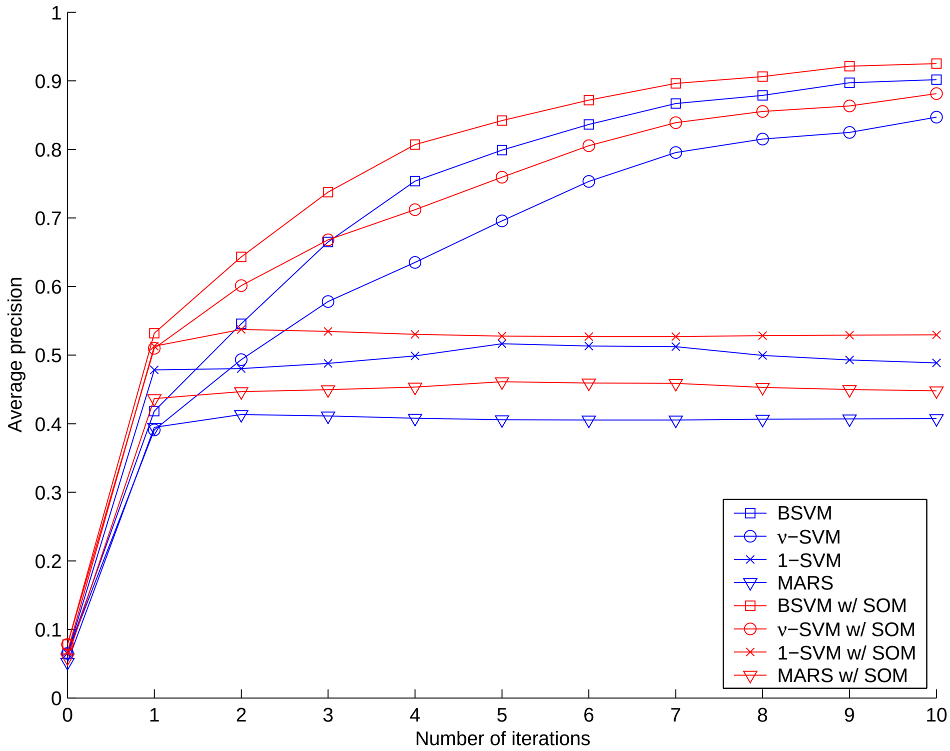


Figure 4.4: Top 10 average precision on 20-cat image dataset

It shows that the SOM-based inter-query learning can help the intra-query learning process and improve the retrieval result.

4.4 Conclusion

We have proposed a SOM-based inter-query learning technique to re-organize the feature vector space of image data, such that the information provided in past queries is utilized and the retrieval result is improved. Moreover, our SOM-based inter-query learning reduced the size of data from the number of images in the collection to the number of neurons in the SOM. Thus, the time complexity of the intra-query learning can be reduced. We performed experiments on real-world image datasets. The experimental results demonstrate that combining

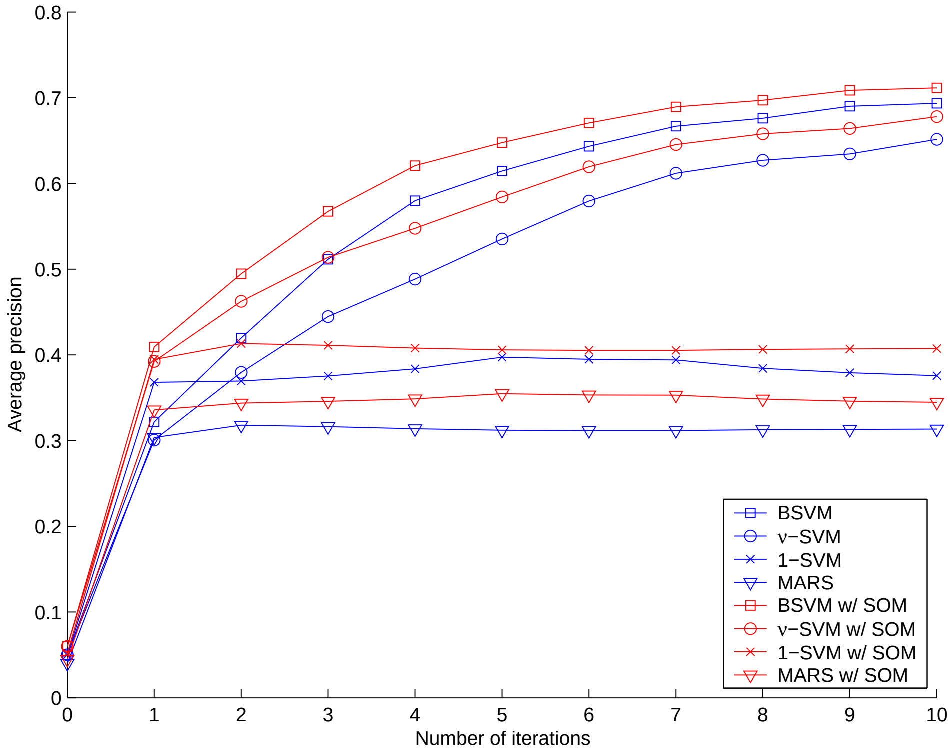


Figure 4.5: Top 30 average precision on 20-cat image dataset

our BSVM-based relevance feedback algorithm and SOM-based inter-query learning technique is effective and promising for improving the retrieval performance in CBIR.

□ **End of chapter.**

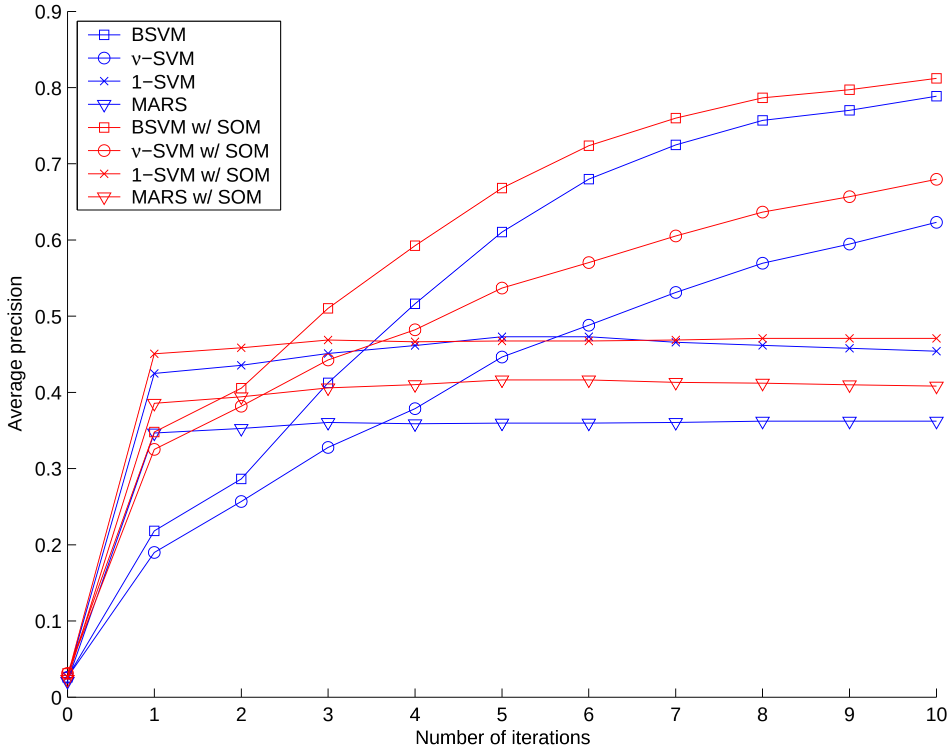


Figure 4.6: Top 10 average precision on 50-cat image dataset

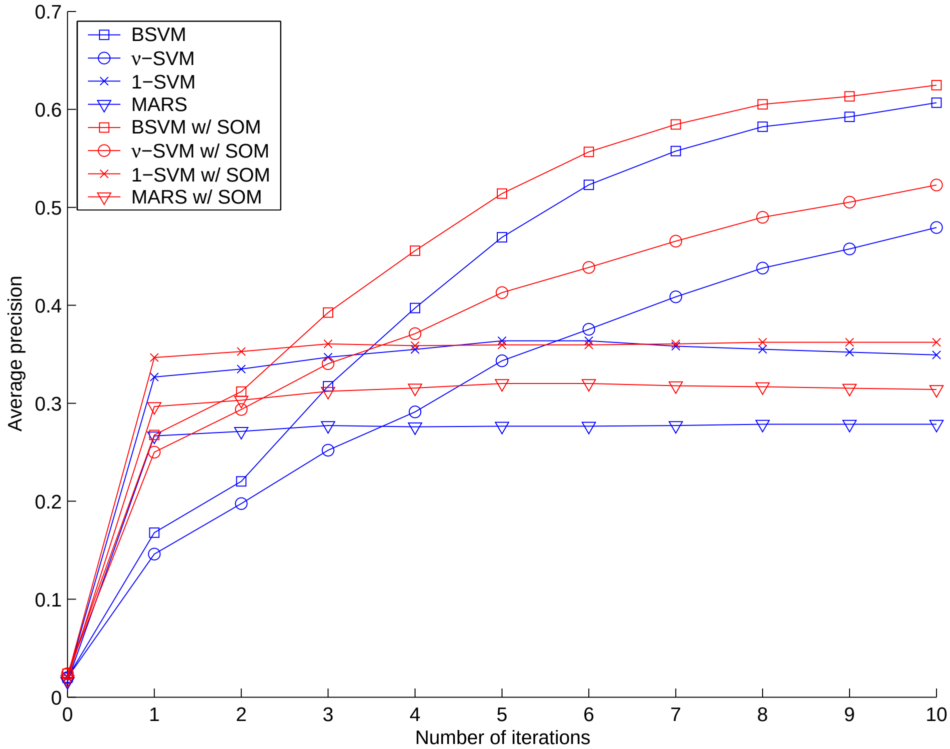


Figure 4.7: Top 30 average precision on 50-cat image dataset

Chapter 5

Conclusion

In this thesis, we have proposed to apply our BSVM in relevance feedback process to capture user's preference in CBIR. Moreover, we also proposed a SOM-based inter-query learning technique to incorporate the past queries in the process to further improve the retrieval performance.

The relevance feedback approach is a powerful technique in CBIR tasks. The goal of relevance feedback is to learn user's preference from their interaction, and it is a powerful technique to improve the retrieval result in CBIR. Under this framework, a set of images is presented to the user according to the query. The user marks those images as either relevant or non-relevant and then feeds back this into the system. Based on these feedback information, the system presents another set of images to the user. The system learns user's preference through this iterative process, and improves the retrieval performance.

Most of the current relevance feedback systems are based on the intra-query learning approach. In this approach, the system refines the query and improves the retrieval result by using feedback information that the user provided. The learning process starts from ground up for each query, and the prior experience from past queries are ignored. Among these techniques, Support Vector Machines (SVM) have shown promising results in the area. In this thesis, we propose to apply our BSVM technique to capture the user's individual preferences in the relevance feedback process, and address the imbalanced dataset problem in relevance feedback process. Our BSVM is able to classify the positive and negative data with maximum margin, and minimize the volume of the positive area. Moreover, our BSVM contains a parameter to control the importance of positive and negative data. The experimental results demonstrate that our BSVM-based relevance feedback algorithm is effective and promising for improving the retrieval performance in CBIR.

Recently, researchers propose the use of inter-query information to further improve retrieval result. In the inter-query approach, feedback information from past queries are accumulated to train the system to determine what images are of the same semantic meaning. In this thesis, we propose a relevance feedback technique to incorporate both inter-query and intra-query information for modifying the feature vector space and estimating the users' target. SOM is used to cluster and index the images in the database. We apply our SOM-based inter-query technique to modify the feature vector space, in which the SOM of images is stored. This allows for transforming the images distributions and improving their organization in the modified vector space. Thus, the images are organized in a fashion that ease the retrieval process. We demonstrate improvement in retrieval precision using both synthetic and real world image data.

In the thesis, we have exploited very minimal potential of the BSVM. The main objective of our BSVM is to overcome the imbalanced dataset problem, i.e., the number of negative examples outnumbered the positive examples, and this problem occurred in many different real world applications. For example, for the task of classifying an image contains a human face or not, the number of negative examples is much more than the positive examples usually. Hence, our future work invlolves generalizing the formulation of the BSVM, and making it available in different problem domains.

□ **End of chapter.**

Bibliography

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

[2] T. Ashwin, N. Jain, and S. Ghosal. Improving image retrieval performance with negative relevance feedback. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1637–1640, 2001.

[3] E. B. Baum. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Network*, 2:5–19, 1991.

[4] E. B. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference in Neural Networks*, pages 335–340, 1992.

[5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] C.-H. Chan and I. King. Using biased support vector machine to improve retrieval result in image retrieval with self-organizing map. In *Proceedings of the 11th International Conference on Neural Information Processing (ICONIP 2004) accepted*, 2004.

[7] C.-H. Chan, K.-C. Sia, and I. King. Utilizing inter and intra-query relevance feedback for content-based image retrieval. In *Special Session of Joint 13th International Conference on Artificial Neural Network and 10th International Conference on Neural Information Processing*, 2003.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machine. In *http://www.csie.ntu.edu.tw/∼cjlin/libsvm/*, 2001.

[9] Y. Chen, X. Zhou, and T. S. Huang. One-class svm for learning in image retrieval. In *Proceedings of the IEEE international conference on Image Processing*, pages 34–37, 2001.

[10] G. Ciocca, I. Gagliardi, and R. Schettini. Quicklook: A content-based image retrieval system with learning capabilities. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, pages 1028–1029, 1999.

[11] G. Ciocca and R. Schettini. A relevance feedback mechanism for content-based image retrieval. *Information Processing and Management*, pages 605–632, 1999.

[12] G. Ciocca and R. Schettini. The quicklook image search engine. *Journal of Image and Graphics*, pages 645–648, 2000.

[13] I. J. Cox, M. L. Miller, T. Minka, and P. N. Yianilos. An optimized interaction strategy for bayesian relevance feedback. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, (CVPR'98)*, pages 553–558, 1998.

[14] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation and psychological experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.

[15] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *Proceedings of International Conference on Pattern Recognition*, pages 361–369, 1996.

[16] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Target testing and the pichunter bayesian multimedia database retrieval system. In *Proceedings of 3rd Forum on Research and Technology Advances in Digital Libraries (ADL'96)*, pages 66–75, 1996.

[17] W. B. Croft. Experiments with representation in a document retrieval system. *Inofrmation Technology: Research and Development*, 2(1):1–21, 1983.

[18] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevanve information. *Journal of Documentation*, 35(4):285–295, 1979.

[19] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Amercian Society of Information Scienece*, 41(6):391–407, 1990.

[20] H. Drucker, B. Shahrary, and D. C. Gibbon. Relevance feedback using support vector machines. In *Proceedings of the 2001 International Conference on Machine Learning*, pages 122–129, 2001.

[21] 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. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):46 – 52, 1995.

[22] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[23] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press Baltimore and London, third edition, 1996.

[24] I. Gondra, D. R. Heisterkamp, and J. Peng. Improving the initial image retrieval set by inter-query learning with one-class svms. *Neural Computing and Applications*, pages 1–10, 2004.

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

[26] G.-D. Guo, A. K. Jain, W.-Y. Ma, and H.-J. Zhang. Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks*, 13(4):811–820, 2002.

[27] D. Harman. Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10. ACM Press, 1992.

[28] X. He, O. King, W.-Y. Ma, M. Li, and H.-J. 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, January 2003.

[29] D. R. Heisterkamp. Building a latent semantic index of an image database from patterns of relevance feedback. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 132–135, 2002.

[30] Helsubki University of Technology. SOM Toolbox: a function package implementing the Self-Organizing Map (SOM) algorithm. In *http://www.cis.hut.fi/projects/somtoolbox/*, 2003.

[31] C.-H. Hoi, C.-H. Chan, K. Huang, M. R. Lyu, and I. King. Biased support vector machine for relevance feedback in image retrieval. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'04) accepted*, 2004.

[32] E. Ide. New experiments in relevance feedback. In *The Smart system - experiments in automatic document processing*, pages 337–354. Englewood Cliffs, NJ: Prentice Hall Inc., 1971.

[33] E. Ide and G. Saltion. Interative search strategies and dynamic file organization in information retrieval. In *The Smart system - experiments in automatic document processing*, pages 373–393. Englewood Cliffs, NJ: Prentice Hall Inc., 1971.

[34] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proceedings of*

24th International Conference on Very Large Data Bases, VLDB, pages 218–227, 1998.

[35] A. K. Jain and A. Vailaya. Shape-based retrieval: a case study with trademark image database. *Pattern Recognition*, 9:1369–1390, 1998.

[36] K. S. Jones. Search term relevance weighting given little relevance information. *Journal of Documentation*, 35(1):30–48, 1979.

[37] I. King and Z. Jin. Relevance feedback content-based image retrieval using query distribution estimation based on maximum entropy principle. In *Proceedings of the International Conference on Neural Information Processing (ICONIP2001)*, pages 699–704, 2001.

[38] I. King, L. Xu, and L. Chan. Using rival penalized competitive clustering for feature indexing in hong kong's textile and fashion image database. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'98)*, pages 237–240, 1998.

[39] T. Kurita and T. Kato. Learning of personal visual impression for image database systems. In *Proceedings of the second International Conference on Document Analysis and Recognition*, pages 547–552, 1993.

[40] F. W. Lancaster. Medlars: Report on the evaluation of its operating efficiency. *American Documentation*, 20(2):119–148, 1969.

[41] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, 1994.

[42] M. Li, Z. Chen, and H. J. Zhang. Statistical correlation analysis in image retrieval. *Pattern Recognition*, 35(12):2687–26937, 2002.

[43] S. Mehrotra, K. Chakrabarti, M. Ortega, Y. Rui, and T. S. Huang. Multimedia analysis and retrieval system. In *Proceedings the 3rd International Workshop on Information Retrieval Systems*, pages 39–45, 1997.

[44] C. Meilhac and C. Nastar. Relevance feedback and category search in image databases. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 512–517, 1999.

[45] T. Minka and R. Picard. Interactive learning using a 'society of models'. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition(CVPR-1996)*, pages 447–452, 1996.

[46] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. 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.

[47] J. Peng. A multi-class relevance feedback approach to image retrieval. In *Proceedings of IEEE International Conference on Image Processing*, pages 46–49, 2001.

[48] J. Peng. Multi-class relevance feedback content-based image retrieval. *Comput. Vis. Image Underst.*, 90(1):42–67, 2003.

[49] R. W. Picard, T. P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 777–780, Lausanne, Switzerland, 1996.

[50] F. Qian, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang. Gaussian mixture model for relevance feedback in image retrieval. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 26–29, 2002.

[51] G. Ratsch, S. Mika, B. Scholkopf, and K.-R. Muller. Constructing boosting algorithms from svms: an application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, 2002.

[52] S. E. Robertson. Specificity and weighted retrieval. *Journal of Documentation*, 30:41–46, 1974.

[53] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[54] J. J. Rocchio. Document retrieval systems - optimization and evaluation, doctoral dissertation, harvard university. In *Report ISR-10, to the National Science Foundation*, Harvard Computional Laboratory, Cambridge, MA, 1966.

[55] J. J. Rocchio. Relevance feedback in information retrieval. In *The Smart system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice Hall Inc., 1971.

[56] J. J. Rocchio and G. Salton. Information search optimization and iterative retrieval techniques. In *Proceedings of the American Federation of Information Processing Societies AFIPS*, volume 27, pages 293–305, 1965.

[57] Y. Rui and T. S. Huang. Optimizing learning in image retrieval. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 236–243, 2000.

[58] Y. Rui, T. S. 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.

[59] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Proceedings IS&T and SPIE Storage and Retrieval of Image and Video Databases VI*, pages 25–36, 1997.

[60] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture in content-based multimedia information retrieval systems. In *Proceedings IEEE Workshop on Content-based Access of Image and Video Libraries, in conjunction with CVPR'97*, pages 82–89, 1997.

[61] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrota. Relevance feedback: A power tool for interactive content-based image retrieval.

IEEE Transactions on Circuits and Video Technology, pages 644–655, September 1998.

[62] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[63] R. Schettini, G. Ciocca, and I. Gagliardi. Content-based color image retrieval with relevance feedback. In *Proceedings IEEE International Conference on Image Processing*, pages 75–79, 1999.

[64] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[65] B. Scholkopf, A. Smola, R. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[66] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[67] J. Smith and S. F. Chang. Automated image retrieval using color and texture. *Technical Report CU/CTR 408- 95-14*, 1995.

[68] M. A. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[69] Z. Su, S. Li, and H. Zhang. Extraction of feature subspaces for content-based retrieval using relevance feedback. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 98–106, 2001.

[70] Z. Su, H. Zhang, S. Li, and S. Ma. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *IEEE Transactions on Image Processing*, 12(8):924–937, 2003.

[71] Z. Su, H. Zhang, and S. Ma. Using bayesian classifier in relevant feedback of image retrieval. In *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000)*, pages 258–261, 2000.

[72] D. M. J. Tax and R. P. W. Duin. Data domain description using support vectors. In *Proceedings of the European Symposium Artificial Neural Network*, pages 251–256, 1999.

[73] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recignition Letters*, 20(11-13):1191–1199, 1999.

[74] J. Tesic and B. Manjunath. Nearest neighbor search for relevance feedback. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 643–648, 2003.

[75] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM Press, 2001.

[76] S. Tong and D. Koller. Support vector machine active learning with application sto text classification. In *Proceedings of the 2000 Internation Conference on Machine Learning*, pages 999–1006, 2000.

[77] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[78] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[79] N. Vasconcelos and A. Lippman. Bayesian relevance feedback for content-based image retrieval. In *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, pages 63–67, 2000.

[80] H. Wu and G. Salton. The estimation of term relevance weights using relevance feedback. *Journal of Documentation*, 37(4):194–214, 1981.

[81] Y. Wu, Q. Tian, and T. Huang. Discriminant-EM algorithm with application to image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pages 1222–1227, 2000.

[82] R. Yan, A. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *Proceedings of the 11th Annual ACM International Conference on Multimedia (MM2003)*, pages 343–346, 2003.

[83] P. Yin, B. Bhanu, K. Chang, and A. Dong. Improving retrieval performance by long-term relevance information. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 533–536, 2002.

[84] C. T. Yu, W. S. Luk, and G. Salton. On models of information retrieval processes. *Information Systems*, 4(3):205–218, 1979.

[85] C. T. Yu and G. Salton. Precision weighting - an effictive automatic indexing method. *Journal of the ACM*, 23(1):76–88, 1976.