Learning to Rank for Content-Based Image Retrieval

Fabio F. Faria
Institute of Computing
University of Campinas
Campinas, SP, Brazil
fabiof@lis.ic.unicamp.br

Adriano Veloso
Dep. of Computer Science
Fed. Univ. Minas Gerais
Belo Horizonte, MG, Brazil
adrianov@dcc.ufmg.br

Humberto M. Almeida
Dep. of Computer Science
Fed. Univ. Minas Gerais
Belo Horizonte, MG, Brazil
hmossri@dcc.ufmg.br

Eduardo Valle
Institute of Computing
University of Campinas
Campinas, SP, Brazil
mail@eduardovalle.com

Ricardo da S. Torres
Institute of Computing
University of Campinas
Campinas, SP, Brazil
rtorres@ic.unicamp.br

Marcos A. Gonçalves,
and Wagner Meira Jr.
Dep. of Computer Science
Fed. Univ. Minas Gerais
Belo Horizonte, MG, Brazil
{mgoncalv,meira}@dcc.ufmg.br

ABSTRACT

In Content-based Image Retrieval (CBIR), accurately ranking the returned images is of paramount importance, since users consider mostly the topmost results. The typical ranking strategy used by many CBIR systems is to employ image content descriptors, so that returned images that are most similar to the query image are placed higher in the rank. While this strategy is well accepted and widely used, improved results may be obtained by combining multiple image descriptors. In this paper we explore this idea, and introduce algorithms that learn to combine information coming from different descriptors. The proposed learning to rank algorithms are based on three diverse learning techniques: Support Vector Machines (CBIR-SVM), Genetic Programming (CBIR-GP), and Association Rules (CBIR-AR). Eighteen image content descriptors (color, texture, and shape information) are used as input and provided as training to the learning algorithms. We performed a systematic evaluation involving two complex and heterogeneous image databases (Corel e Caltech) and two evaluation measures (Precision and MAP). The empirical results show that all learning algorithms provide significant gains when compared to the typical ranking strategy in which descriptors are used in isolation. We concluded that, in general, CBIR-AR and CBIR-GP outperforms CBIR-SVM. A fine-grained analysis revealed the lack of correlation between the results provided by CBIR-AR and the results provided by the other two algorithms, which indicates the opportunity of an advantageous hybrid approach.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

General Terms

Experimentation, Performance

Keywords

Content-based image retrieval, Learning to rank, Suport Vector Machines, Genetic Programming, Association Rules

1. INTRODUCTION

Traditional image retrieval approaches, based on keywords and textual metadata, face serious challenges. Describing the image content with textual features is intrinsically very difficult, and the task has not been made easier by the growth and diversification of image databases. Many applications, especially those dealing with large general image databases face obstacles to obtain textual descriptors, where manual annotation is prohibitively expensive, contextual text is scarce or unreliable, and user needs are impossible to anticipate.

On those contexts, Content-Based Image Retrieval (CBIR, [34]), can be very helpful, since it forsakes the need of keywords or other textual metadata. Often, it consists of retrieving the most similar images to a given query image, a form of query-by-example that makes concrete the intuition of the famous proverb: "a picture is worth a thousand words". However, satisfying the user needs involves answering the conceptual query which is represented by the sample image – an open research issue.

A critical aspect of the system is the final ordering – the ranking – of the images. CBIR systems will rank the images in the result set according to their similarity to the query image. Because the result set is often large, the users will only inspect the topmost results, so their perception of the system quality depends critically on the relevance of those results.

Similarity is computed using image content descriptors, which combine a feature vector and a similarity measure to express a specific perceptual quality of the image [6]. The feature vectors encode visual features associated with the images, such as color, texture, and shape [12, 36–38]. The similarity measures (which range from simple metrics, like the one based on the Euclidean distance, to very elaborate algorithms, like the Earth Mover's Distance [25]) determine

how the feature vectors are distributed in the description space – affecting critically how the vectors correspond to perceptual qualities.

Obviously, different descriptors produce different rankings. Also, the best descriptor to employ is data-dependent, and impossible to know before query time. Further, it is intuitive that different descriptors may provide different but complementary information about images, so that the combination of multiple descriptors may improve ranking performance. Combining multiple descriptors is clearly a better strategy than relying on a single one, but the optimal combination of descriptors is, again, data-dependent and impossible to obtain in advance.

In this paper we propose an alternative approach for content-based image retrieval, which applies learning algorithms to effectively combine multiple descriptors in order to improve ranking performance. We provide as input to the learning algorithms a set of query images. Associated with each query image, we also provide a set of sample images which are represented by the corresponding similarities to the query image. The similarities are computed usind multiple descriptors. The relevance of an image to the query image is also informed as input (e.g., an image is relevant if it is trully similar to the query image, otherwise it is irrelevant). This information is used as training, so that the learning algorithms produce a ranking function that maps similarities to the level of relevance of images to query images. When a new query image is given, the relevance of the returned images is estimated according to the learned function (i.e., this function gives a score to an image indicating its relevance to the query image).

The main contributions of this paper are: the application of the learning to rank approach to CBIR, and the introduction of a new rank learning scheme, based on Association Rules. Also, to the best of our knowledge, this is the first attempt at comparing algorithms of learning to rank, in the context of CBIR.

Three algorithms have been evaluated, representing very different learning strategies: (i) CBIR-SVM, which is based on Support Vector Machines [3, 21, 44], (ii) CBIR-GP, which is based on Genetic Programming [7, 8, 23], and (iii) CBIR-AR, which is based on Association Rules [1, 42]. We have performed a systematic set of experiments using two image databases (Corel and Caltech).

Our results indicate that algorithms that learn to rank achieve superior ranking performance when compared to traditional approaches that simply employ isolated descriptors. We also found out that CBIR-AR and CBIR-GP outperform CBIR-SVM in terms of overall ranking quality and that the strengths of CBIR-AR are complementary to those of CBIR-GP and CBIR-SVM, indicating that synergy could be obtained by combining the former with one of the latter.

The remaining of the paper is organized as follows. Next section discusses related work on the application of learning algorithms for CBIR. A formal definition of the problem, as well as a description of the algorithms analyzed, are presented in Section 3. In Section 4, we evaluate empirically the effectiveness of those algorithms. In Section 5, we present our conclusions.

2. RELATED WORK

Several machine learning techniques have been used for learning to rank different kinds of objects (e.g., text documents, images) and have provided good results. For text documents, some approaches [2, 8] use Genetic Programming (GP) to optimize ranking functions and obtain the better performance in search for documents. Zobel and Mofat present no less than one million possibilities to compute such ranking functions [47]. Other approaches based on Support Vector Machine (SVM) have been proposed [4, 15, 20] to discover the best search function. Furthermore, Veloso et al. [41] find patterns (or rules) associating document features using Association Rules. Later those rules are used to rank documents.

In the CBIR domain, the use of machine learning techniques to rank images tries to alleviate the so-called *semantic gap* problem: translation of high-level user concepts into low-level feature vectors, provided by descriptors. One common approach is to use learning methods to combine different descriptors. In [11, 35], those approaches rely on assigning weights to indicate the importance of a descriptor. Basically, the higher the weight the more important a descriptor is assumed to be. Frome et al. [11] apply a maximal-margin formulation for learning linear combination of elementary distances defined by triplets of images. Shao et al. [35] use Genetic Algorithms to determine the best weights for available descriptors. Kernels and SVM have also been used for CBIR. Examples include [13, 46]. Torres et al. [7], in turn, exploit GP for combining image descriptors and finding the best weight for each descriptor. Those algorithms optimize image retrieval and try to minimize the *semantic gap*.

In order to include the user in the CBIR process, relevance feedback techniques have been proposed to improve the effectiveness of retrieval systems. In [10, 16, 29], learning techniques are used to meet user needs. In these techniques, the user indicates to the system which images are relevant and the system learns from these indications trying to return more relevant images at next iterations. Those relevance feedback algorithms try to characterize specific user perceptions/needs.

There are very few works in the literature concerned with learning to rank specifically for CBIR. In [17], it is proposed the use of multiple-instance ranking based on the max margin framework (method that is adapted from the RankSVM algorithm [15]), where local information is extracted from images. This approach considers that is more flexible to use the relative ranking (an image is more relevant than another one) for image retrieval than to use the traditional relevance feedback methods, where images are grouped into relevant and irrelevant sets.

3. LEARNING TO RANK IMAGES

In this section we introduce the concept of CBIR, and give a formal definition of the problem of learning to rank images. Then, we present three algorithms, based on different learning techniques.

3.1 Content-Based Image Retrieval (CBIR)

CBIR systems are designed to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). Their goal is to support image retrieval based on content properties (e.g., shape, color, texture), encoded into feature vectors.

CBIR is strongly based upon the concept of descriptor. A descriptor is a mathematical object which tries to express some perceptual quality of the images, and is composed by:

(1) a feature extraction algorithm that encodes image properties, such as color, shape, and texture into a feature vector; and (2) a similarity measure (distance function) that computes the similarity between two images as a function of the distance between their corresponding feature vectors [6]. Both the feature vector and the distance function affect how the descriptor encodes the perceptual qualities.

Images are usually coded in a way that is both extensive (images are large) and semantically poor (there is very few semantic content in the pixels themselves). Thus, descriptors play a fundamental role in CBIR systems, since they provide a more compact and semantically richer representation for images.

The CBIR system will usually pre-process the images stored in its database, by extracting and indexing the feature vectors. This process is usually performed off-line, once per image.

Once the database is ready, the CBIR system allows the user to specify the queries by means of a query pattern (which can be a sample image). The query is also processed by the feature vector extractor, and the similarity function is used to evaluate its similarity to the database images. Then, the database images will be ranked in decreasing order of similarity to the query, and shown to the user in that order.

There is a huge array of descriptors available in the literature, with their corresponding strengths and weaknesses. The choice of the descriptors affects critically the overall effectiveness of the CBIR system.

Table 1 lists the set of descriptors considered in our study.

Descriptor	Content Type
GCH [38]	Color
BIC [36]	Color
COLORBITMAP [28]	Color
ACC [19]	Color
CCV [33]	Color
CGCH [37]	Color
CSD [31]	Color
JAC [43]	Color
LCH [38]	Color
CCOM [22]	Texture
LAS [39]	Texture
LBP [32]	Texture
QCCH [18]	Texture
SASI [5]	Texture
SID [45]	Texture
UNSER [40]	Texture
EOAC [30]	Shape
SPYTEC [24]	Shape

Table 1: The eighteen image descriptors used in our experiments.

3.2 Problem Definition

Different descriptors provide different but complementary information about the similarity between images. This is because descriptors may employ different content types (i.e., color, texture or shape), or may do it in different ways. Certain descriptors may be more effective for some images, and less effective for others. There is no perfect descriptor, and no descriptor is consistently superior than all others in all possible cases.

Our approach is to use learning algorithms, which are able to combine different information provided by multiple descriptors (eighteen in our experiments) in order to improve efficiently ranking performance.

We present the problem in a classical learning setting. The learning algorithm is presented with a set of annotated examples composed by query images and their corresponding ground-truth (i.e., the degree of relevance of the images in the database for that query). The degree of relevance is chosen among a few discrete possibilities (e.g., 0 for irrelevant images, 1 for relevant ones).

The learning algorithms have, at their disposal, a precomputed matrix of the distances between the queries and all images in the database, for each descriptor considered in the study.

The idea is that the algorithm should take into account both the training set (with the ground-truth annotations) and the available distances (which provide the similarity between the query and the database images) to learn the optimal way to combine the evidence from the descriptors in order to answer the queries. The algorithms produce a ranking model, learned from the training set, which is used to estimate the relevance of arbitary images.

3.3 Machine Learning Techniques

The three algorithms studied are based on three very different machine learning techniques, which are employed to learn to rank images from a training set of rankings and a set of descriptors. We have called the algorithms CBIR-SVM, CBIR-GP, and CBIR-AR, accordingly to the underlying machine learning scheme used on each one of them.

3.3.1 CBIR-SVM: Learning using Support Vector Machines

The Support Vector Machines (SVM), introduced by [3], is a supervised learning method for solving classification problems. The main idea of SVM is to construct a separating hyperplane in an n-dimensional space (where n is the number of attributes of the entity being classified, i.e., its dimensionality), that maximizes the margin between two classes. Intuitively, the margin can be interpreted as a measure of separation between two classes and can be interpreted as a measure of the quality of the classification. Figure 1 illustrates this idea.

More recently [15, 20], SVM was applied for learning ranking functions in the context of information retrieval. It has also been employed specifically for CBIR [14, 17]

The learning process using SVM for ranking images works as follow:

Given an input space $X \in R^n$, where n is number of features and an output space of ranks represented by labels $Y = \{r_1, r_2, ..., r_q\}$, where q denotes the number of ranking positions. In these ranks there exist an order, $r_q \succ r_{q-1} \succ ... \succ r_1$, where $\succ$ represents a preference relation [4].

Each instance $\vec{x}_i \in X$ denotes a query-image pair (α,β), where α denotes an image of the database and β the query image, and it is labeled with one rank. It is represented by a feature vector in which each feature is an image descriptor defined as a function of the query and a database image. A preference relation between instances exists, $\vec{x}_i$ is preferable

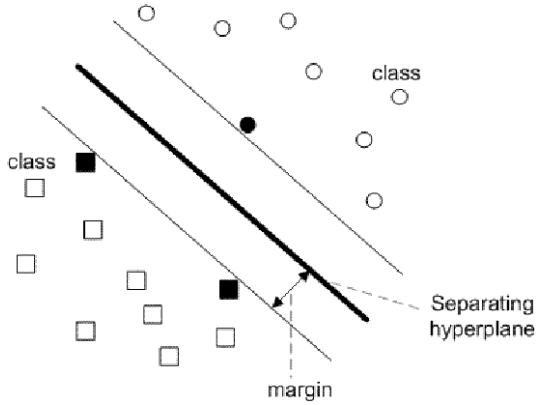

Figure 1: The SVM classifier finds the hyperplane which separates the two classes (here exemplified by squares and circles) with the widest possible margin.

to $\vec{x}_j$ is denoted by $\vec{x}_i \succ \vec{x}_j$.

A ranking function $f \in F$ can be used to give a score value for each instance $\vec{x}_i \in X$. Thus, it is possible to determine preference relations between instances from a set of ranking functions F, such that:

$$\vec{x}_i \succ \vec{x}_j \Leftrightarrow f(\vec{x}_i) > f(\vec{x}_j) \quad (1)$$

The ranking problem can be interpreted as a learning problem for classification on pairs of instances ($\vec{x}_i$,$\vec{x}_j$) as being either well or badly ranked. The preference relation $\vec{x}_i \succ \vec{x}_j$ is represented by a new vector $\vec{x}_i - \vec{x}_j$ such that

$$\left(\vec{x}_i - \vec{x}_j, z = \begin{cases} +1 & y_i \succ y_j \\ -1 & y_j \succ y_i \end{cases}\right) \quad (2)$$

Next, to classify each pair of images ($\vec{x}_i$,$\vec{x}_j$), two classes are considered: correctly ranked (+1) and incorrectly ranked pairs (-1), the former being those where $\vec{x}_i$ should be ahead of $\vec{x}_j$, and the latter being those where the converse is true.

From all original instances $\vec{x}_i \in X$, it is created a new training data set S' containing new labeled vectors according to Equation 2. We take S' as classification data and construct a SVM model that will assign either positive label (z=+1) or negative label (z=-1) to all vectors $\vec{x}_i - \vec{x}_j$.

Thus, the task is to select the best function $f^* \in F$ that minimizes a given loss function, given ranked instances, resulting the ranking model of Ranking SVM.

For a more detailed description of learning ranking functions using SVMs, the reader is referred to [4, 17, 21].

3.3.2 CBIR-GP: Learning using Genetic Programming

Genetic Programming (GP) is an inductive learning method introduced by Koza [23] as an extension to Genetic Algorithms (GAs). It is a problem-solving system designed following the principles of inheritance and evolution, inspired by the idea of *Natural Selection*. The space of all possible solutions to the problem is investigated using a set of optimization techniques that imitate the theory of evolution.

In order to apply GP to solve a given problem, several key components of a GP framework need to be defined. In our application, we have modeled the "population" in evolution as arithmetic combination functions (uniquely represented as expression trees), whose non-leaf nodes are numerical operators and the leaf node set is composed of the similarity values obtained from different descriptors [7].

The essential GP components are mapped as follows:

- Terminals: Leaf nodes in the tree structure. Terminals are the similarity functions of each descriptor.
- Functions: Non-leaf nodes used to combine the leaf nodes. The following functions were used in our implementation: $+, \times, /, -, log, exp$. This function set is widely used in common GP experiments and is suitable to validate our ideas.
- Initial Population Generation: The initial set of trees randomly generated by the *ramped half-and-half method* [23].
- Fitness Function: The objective function GP aims to optimize. A fitness function measures how effective a combination function represented by an individual tree is for ranking images. In our study, we use mean average precision (MAP) as fitness function.
- Crossover: A genetic operator that exchanges subtrees from two parents to form two new children. Its aim is to improve the diversity as well as the genetic fitness of the population. This process is shown in Figure 2(b).
- Reproduction: A genetic operator that copies the individuals with the best fitness values directly into the population for the next generation without going through the crossover operation.
- Mutation: A genetic operator that replaces a selected individual's subtree, whose root is a picked mutation point, with a randomly generated subtree.

The GP evolution process starts with an initial population of individuals, composed by terminals and functions. Usually, the initial population is generated randomly. Each individual denotes a solution to the examined problem and is represented by a tree, as shown in Figure 2(a). To each one of these individuals is associated a fitness value. This value is determined by fitness function that calculates how good the individual is. The individuals will evolve generation by generation through genetic operations such as reproduction, crossover, and mutation. Thus, for each generation, after the genetic operations are applied, a new population replaces the current one.

The process is repeated over many generations until the termination criterion has been satisfied. This criterion can be, for example, a maximum number of generations, or some level of fitness to be reached.

The GP framework is an iterative process with a training and a validation phase, both needing a set of annotated queries. In the training phase, the annotated set is used to discover candidate individuals. The best ones are then evaluated on the validation set, to select the individual that presents the best performance in both sets. This validation phase is used to avoid overfitting.

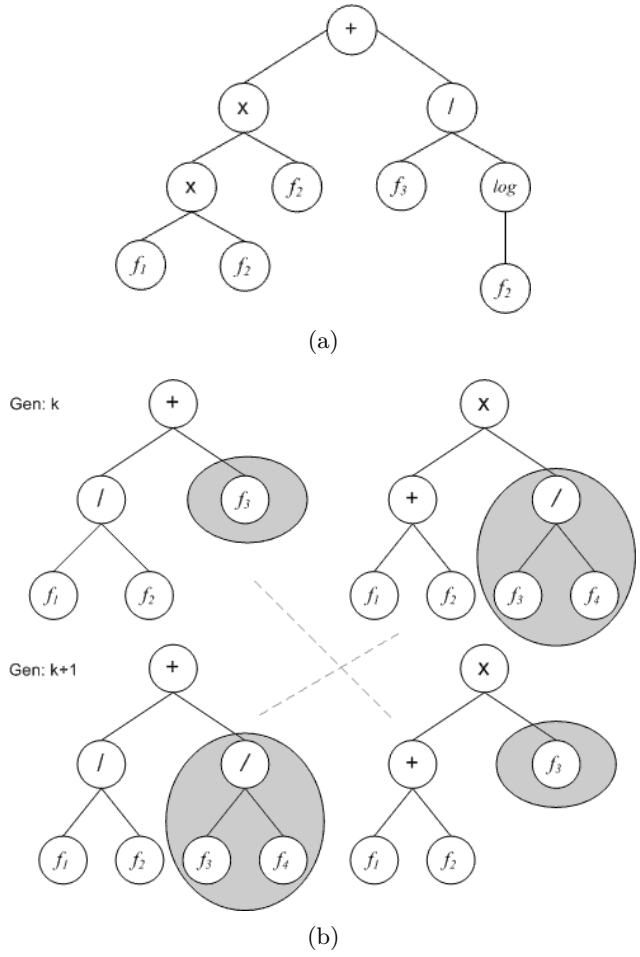

Figure 2: CBIR-GP individuals (a) are similarity functions (from potentially different descriptors) combined by arithmetic operators; the Genetic Programming scheme combines "material" from different individuals in the population from one generation to another using, for example, a crossover operations (b).

Please, refer to [2, 7, 8] for a more detailed description of learning ranking functions and descriptor combining strategies for content-based image retrieval using Genetic Programming.

3.3.3 CBIR-AR: Learning using Association Rules

Association rules are patterns describing implications of the form $\mathcal{X} \rightarrow \mathcal{Y}$, where we call $\mathcal{X}$ the antecedent of the rule, and $\mathcal{Y}$ the consequent. The rule does not express a classical logical implication where $\mathcal{X}$ necessarily entails $\mathcal{Y}$. Instead it denotes the tendency of observing $\mathcal{Y}$ when $\mathcal{X}$ is observed. Association rules have been originally conceived for data mining [1] and have also been used for textual information retrieval [41].

In the context of leaning to rank images, we are interested in using the training set, to associate descriptor similarity values calculated to relevance levels. The similarity values are first discretized using the procedure proposed in [9]. Then the rules become of the form the form $\mathcal{X} \rightarrow r_i$, where the antecedent of the rule is a set of similarity values (potentially coming from different descriptors) and the consequent is a relevance level.

Two measures are used to estimate the quality of a rule:

- The support of $\mathcal{X} \rightarrow r_i$, represented by $\sigma(\mathcal{X} \rightarrow r_i)$, is the fraction of examples in the training set containing the feature-set $\mathcal{X}$ and relevance r_i.
- The confidence of $\mathcal{X} \rightarrow r_i$, represented by $\theta(\mathcal{X} \rightarrow r_i)$, is the conditional probability of r_i given $\mathcal{X}$. The higher confidence, the stronger is the association between $\mathcal{X}$ and r_i.

In order to avoid a combinatorial explosion while extracting rules, a minimum support threshold, is employed.

In order to estimate the relevance of an image, it is necessary to combine the predictions performed by different rules [41]. Our strategy is to interpret each rule $\mathcal{X} \rightarrow r_i$ as a vote given by $\mathcal{X}$ for relevance level r_i. Votes have different weights, depending on the confidence of the corresponding rules The weighted votes for relevance r_i are summed and then averaged by the total number of rules predicting r_i, as shown in Equation 3, where $\mathcal{R}$ is the set of rules used in the voting process:

$$s(r_i) = \frac{\sum_{\mathcal{X} \rightarrow r_i \in \mathcal{R}} \theta(\mathcal{X} \rightarrow r_i)}{|\mathcal{R}|} \quad (3)$$

Thus, the score associated with relevance r_i, $s(r_i)$, is essentially the average confidence associated with rules that predict r_i. Finally, the relevance of an image is estimated by a linear combination of the normalized scores associated with each relevance level ($r_i \in \{0, 1\}$), as shown in Equation 4:

$$relevance = \sum_{i \in \{0,1\}} \left(r_i \times \frac{s(r_i)}{\sum_{j \in \{0,1\}} s(r_j)} \right) \quad (4)$$

The reader is referred to [41] for a more detailed description of the process of estimating relevance using association rules.

4. EXPERIMENTAL EVALUATION

In this section we present the experimental results for the evaluation of the proposed learning to rank algorithms in terms of ranking performance. Our evaluation is based on a comparison of the CBIR-SVM, CBIR-GP, CBIR-AR algorithms.

We first present general information about how the experiments were conducted (databases, evaluation metrics, parameters etc.), and then we present and discuss the results obtained.

4.1 Image Databases

We have employed subsets from two large image databases in our evaluation. The first database, Corel, was extracted from a database containing 20,000 images from the Corel GALLERY Magic - Stock Photo Library 2. Our subset is composed by 3,906 images and 123 query images. There

are 85 classes of images and the number of images per class varies from 7 to 98.

The second database, Caltech, contains 8,677 color images extracted from the Caltech101 database [26]. Those images are grouped into 101 classes (i.e., planes, ants, brains, cameras etc.) and the number of images per class varies from 40 to 800. There are 122 query images.

In both databases, classes are mutually exclusive, i.e., an image can be associated to only one class. An image is considered relevant to a query image if both belong to the same class.

For each database, a matrix of the similarity values between each pair of images has been computed, using each one of the eighteen descriptors showed in Table 1.

Figure 3 shows the percentage of relevant images for each query image used. This is related to the difficulty of image retrieval on each database, since a query with too few potential answers is, all other things being equal, more challenging to the CBIR system.

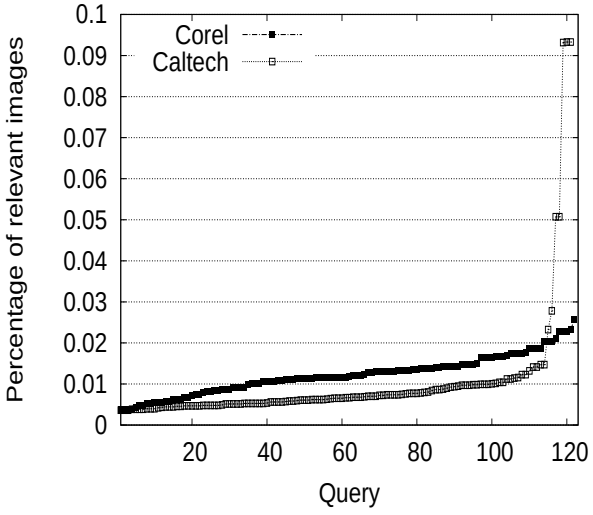

Figure 3: Distribution of relevant images, indicating the expected abundance/scarsity of correct answers for each query. This has an impact on the degree of difficulty of the retrieval task (this distribution indicates that the Caltech database is more challenging).

4.2 Evaluation Metrics

To evaluate the ranking performance of the algorithms, we have used two metrics: precision (taken at a few topmost positions), and MAP (Mean Average Precision, which gives a summarized measure of the precision x recall curve).

While both measures tend to emphasize quality at the top of the rank, the effect is less pronounced on MAP, which is sensitive to the entire ranked list of images. The precision measure is much more focused on the top of the ranked list. For detailed discussion about the metrics, the reader is referred to [27].

4.3 Setup

To evaluate the ranking performance of the algorithms, we conducted five-fold cross validation. Thus, each database is arranged in five folds, including training, validation and test. At each run, three folds are used as training set, one fold is used as validation-set and the remaining fold as test-set. Parameters for each learning algorithm were chosen using the validation-set (i.e., the parameters that lead to the best performance in the validation-set were used in the test-set), and are shown in Tables 2 and 3. Exhaustive variations of GP parameteres were performed in a small sample of the training set in order to find the optimal combination of them. However, because of the deviation among them has been minimum, it was used the same configuration set in all folds.

Run	Corel		Caltech	
	CBIR-AR (σ_{min})	CBIR-SVM (C)	CBIR-AR (σ_{min})	CBIR-SVM (C)
1	0.0025	0.2000	0.0005	0.9000
2	0.0050	0.2000	0.0001	0.2000
3	0.0050	0.0900	0.0002	0.0900
4	0.0025	0.1000	0.0001	30.000
5	0.0075	10,000	0.0001	0.0900

Table 2: Parameters used on the CBIR-AR and CBIR-SVM algorithms, for each database.

GP Parameter	Value
Population	600
Generations	30
Crossover	0.85
Mutation	0.10
Reproduction	0.05
Tree Depth	8

Table 3: Parameters used for CBIR-GP for both Corel and Caltech databases.

4.4 Results

We first report results obtained from evaluation of all descriptors used in our experiments (Section 4.4.1). Next (Section 4.4.2), we will discuss the results from a coarse grained analysis, averaged by query, and by run. This analysis is intended to give an overall picture concerning the ranking performance of the algorithms. In section 4.4.3 we will discuss the results obtained using a finer grained analysis.

4.4.1 Evaluation of Image Descriptors

Table 4 shows the precision values for Corel database, considering eigthteen descriptors (those presented in Table 1). As it can be observed, the BIC descriptor yields the best results in terms of precision values for different numbers of retrieved images. For Caltech database, similar results were observed.

The results of the experiments using the BIC descriptor was used to confirm that the combination of different descriptors provides better results than the use of a single one.

4.4.2 Coarse Grained Analysis

Table 5 shows MAP values for Corel and Caltech databases. The result for each run is obtained by averaging partial results considering each query in the run. The final result is obtained by averaging the five runs.

For the Corel database, CBIR-AR and CBIR-GP are the best performers. On average, CBIR-AR and CBIR-

GP present similar performance, being superior than CBIR-SVM and BIC. CBIR-AR showed improvements of about 4.2% and 21.0% when compared to CBIR-SVM and BIC, respectively. CBIR-GP, in turn, showed improvements of about 4.7% and 21.6%, when compared to CBIR-SVM and BIC, respectively.

The same trend holds in the Caltech database. Again, CBIR-AR and CBIR-GP present the best results on average. Specifically, for run 3, CBIR-AR and CBIR-SVM are statistically tied. CBIR-AR showed improvements of about 14.3% and 24.0. CBIR-GP, in turn, showed improvements of about 9.7% and 19.8%, when compared to CBIR-SVM and BIC, respectively.

The next set of experiments evaluates the effectiveness of CBIR-AR, CBIR-GP, CBIR-SVM, and BIC in terms of the precision measures. Table 6 shows precision values obtained for each algorithm.

In the Corel database, CBIR-AR showed the best ranking performance at the first four positions of the ranking. However, CBIR-GP showed a slightly better performance at the final positions. For the Caltech database, CBIR-AR presents the best ranking performance for all positions considering precision measure.

MAP results, which take into account the entire rank, show a statistical tie between CBIR-AR and CBIR-GP, both showing better results than CBIR-SVM. The precision results favor CBIR on the topmost (and most critical, from the user point of view) positions of the rank. The single descriptor retrieval (BIC) is almost always outperformed by all other methods.

The significance of the results were confirmed by statistical tests. We have conducted two sets of significance tests considering each database. The first set of significance tests was carried on the average of the results for each query, while the second considered the average of the five runs.

4.4.3 Fine Grained (Query-Level) Analysis

We were interested in studying the correlation between learning algorithms, so that we could analyze the situations in which one of the algorithms outperformed the other.

Figure 4 shows scatter plots of MAP values obtained by different algorithms for each query. The coordinates associated with each point in one of the graphs are given by the MAP values obtained by the two algorithms indicated in the axes. For example, the point p_1 (labeled in the topmost graph) represents a query in Corel database for which CBIR-AR achieves a MAP value of 1.0 and CBIR-GP achieves a MAP value of 0.10 (x-axis). Similarly, the point labeled p_2 represents another query, where CBIR-AR achieves a MAP of 0.01 and CBIR-GP achieves a MAP of 1.0.

As it can be seen, CBIR-GP and CBIR-SVM are strongly correlated (correlation coefficients are above 0.90), indicating that those algorithms tend to achieve similar ranking performance in roughly the same queries. On the other hand, the performance of CBIR-AR is little correlated with the performances of the other two algorithms, showing that it tends to perform well where the others perform badly and vice-versa. This phenomenon is also observed in Caltech database (correlation coefficients shown in Table 7).

A manual inspection of the queries revealed that, for CBIR-GP and CBIR-SVM, the key property that leads to a good ranking performance is the number of relevant images for each query. Specifically, CBIR-GP and CBIR-SVM achieve higher MAP values in queries with several relevant images, and lower MAP values in queries with few relevant images. CBIR-AR, on the other hand, is less sensitive to this property, being able to achieve good ranking performance even for queries that have only few relevant images.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have evaluated three different machine leaning algorithms for ranking images in CBIR systems: CBIR-SVM (based on Support Vector Machines), CBIR-GP (based on Genetic Programming) and CBIR-AR (based on Association Rules). CBIR-AR is an original contribution.

We have shown that the learning algorithms, used to combine evidence from multiple descriptors, largely outperform the results obtained from the single best descriptor (BIC), showing the advantage of the learning schemes.

Among the learning schemes, CBIR-AR and CBIR-GP yield similar performances considering the whole ranking, but CBIR-AR outperforms CBIR-GP on the topmost (and most critical) positions of the rank. Both outperform CBIR-SVM in all considered metrics.

The fine-grained analysis showed an interesting lack of correlation between the quality of the results of CBIR-AR and the other two schemes, which indicates the opportunity to combine the schemes to obtain an even better ranking. We are currently working on that direction.

6. ACKNOWLEDGMENTS

Authors are grateful to FAPESP, CAPES, FINEP, CNPq, CNPq by BIO-CORE project, FAPEMIG, INCTWeb (CNPq grant number 573871/2008-6) and Microsoft Research for financial support.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[2] H. M. Almeida, M. Gonçalves, M. Cristo, and P. Calado. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *SIGIR '07*, pages 399–406, 2007.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152. Springer, 1992.

[4] Y. Cao, X. Jun, T. Liu, H. Li, Y. Huang, and H. Hon. Adapting ranking svm to document retrieval. In *SIGIR '06*, pages 186–193, 2006.

[5] A. Çarkacioglu and F. Yarman-Vural. Sasi: a generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615–2633, 2003.

[6] R. da S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[7] R. da S. Torres, A. X. Falcão, M. A. Goncalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009.

[8] W. Fan, M. D. Gordon, and P. Pathak. Genetic programming-based discovery of ranking functions for effective web search. *J. Manage. Inf. Syst.*, 21(4):37–56, 2005.

Descriptors	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10	Avg
ACC	1.000	0.514	0.405	0.325	0.271	0.234	0.196	0.162	0.129	0.088	0.332
BIC	**1.000**	**0.536**	**0.425**	**0.349**	**0.293**	**0.249**	**0.218**	**0.179**	**0.139**	**0.096**	**0.348**
CCOM	1.000	0.344	0.259	0.223	0.181	0.154	0.128	0.106	0.088	0.064	0.255
CCV	1.000	0.380	0.255	0.203	0.175	0.153	0.118	0.094	0.076	0.052	0.251
CGCH	1.000	0.222	0.130	0.095	0.079	0.063	0.055	0.048	0.039	0.034	0.176
CSD	1.000	0.355	0.233	0.168	0.130	0.107	0.091	0.075	0.066	0.055	0.228
EOAC	1.000	0.234	0.147	0.107	0.091	0.077	0.065	0.057	0.048	0.039	0.186
GCH	1.000	0.404	0.263	0.213	0.183	0.156	0.129	0.104	0.074	0.044	0.257
JAC	1.000	0.467	0.344	0.276	0.231	0.182	0.147	0.119	0.090	0.052	0.291
LAS	1.000	0.207	0.134	0.103	0.084	0.066	0.055	0.046	0.041	0.034	0.177
LBP	1.000	0.139	0.083	0.064	0.056	0.046	0.040	0.038	0.035	0.032	0.153
LCH	1.000	0.351	0.243	0.190	0.162	0.135	0.114	0.093	0.075	0.058	0.242
LUCOLOR	1.000	0.316	0.199	0.155	0.132	0.110	0.093	0.078	0.064	0.052	0.220
QCCH	1.000	0.154	0.093	0.069	0.061	0.054	0.049	0.044	0.039	0.035	0.160
SASI	1.000	0.289	0.205	0.154	0.125	0.102	0.081	0.068	0.055	0.043	0.212
SID	1.000	0.150	0.087	0.069	0.059	0.051	0.044	0.040	0.036	0.031	0.157
SPYTEC	1.000	0.036	0.029	0.026	0.026	0.025	0.025	0.024	0.024	0.023	0.124
UNSER	1.000	0.086	0.043	0.034	0.032	0.030	0.029	0.028	0.027	0.026	0.133

Table 4: Precision values for Corel database. Best results are shown in bold.

Run	Corel				Caltech			
	CBIR-AR	**CBIR-GP**	**CBIR-SVM**	**BIC**	**CBIR-AR**	**CBIR-GP**	**CBIR-SVM**	**BIC**
1	**0.405**	0.399	0.374	0.279	0.051	**0.059**	0.051	0.058
2	0.312	**0.328**	0.308	0.298	**0.106**	0.057	0.038	0.018
3	0.366	**0.376**	0.351	0.341	**0.098**	0.089	**0.098**	0.093
4	**0.362**	0.354	0.344	0.290	0.046	0.049	0.039	**0.061**
5	**0.250**	0.246	0.251	0.193	0.064	**0.098**	0.092	0.063
Final(Avg)	**0.339**	**0.341**	0.326	0.280	**0.073**	**0.070**	0.064	0.058
CBIR-AR	-	-0.5%	4.2%	21.0%	-	4.1%	14.3%	24.0%
CBIR-GP	0.5%	-	4.7%	21.6%	-4.1%	-	9.7%	19.8%

Table 5: MAP values for Corel and Caltech databases. Best results, including statistical ties, are shown in bold. The percentage values represent the relative gain between techniques.

@	Corel				Caltech			
	CBIR-AR	**CBIR-GP**	**CBIR-SVM**	**BIC**	**CBIR-AR**	**CBIR-GP**	**CBIR-SVM**	**BIC**
1	**0.772**	0.711	0.750	0.633	**0.273**	0.196	0.206	0.237
2	**0.699**	0.660	0.668	0.606	**0.247**	0.196	0.184	0.196
3	**0.651**	0.641	0.622	0.569	**0.230**	0.197	0.182	0.178
4	**0.625**	0.620	0.597	0.545	**0.211**	0.185	0.175	0.169
5	**0.602**	**0.604**	0.581	0.537	**0.197**	0.174	0.166	0.155
6	**0.593**	**0.593**	0.564	0.515	**0.193**	0.165	0.155	0.149
7	**0.582**	**0.584**	0.556	0.495	**0.182**	0.160	0.151	0.140
8	0.561	**0.573**	0.545	0.481	**0.175**	0.163	0.147	0.134
9	0.549	**0.560**	0.531	0.476	**0.166**	0.163	0.146	0.132
10	0.540	**0.550**	0.521	0.464	**0.161**	0.156	0.141	0.127

Table 6: Precision values for the Corel and Caltech databases. Best results, including statistical ties, are shown in bold.

Techniques	Corel	Caltech
CBIR-AR×CBIR-GP	-0.064	0.318
CBIR-AR×CBIR-SVM	-0.091	0.257
CBIR-GP×CBIR-SVM	0.978	0.981

Table 7: Correlation coefficients between MAP numbers for each query.

[9] U. Fayyad and K. Irani. Multi interval discretization of continuous-valued attributes for classification learning. In *IJCAI.*, pages 1022–1027, 1993.

[10] C. D. Ferreira, R. da S. Torres, M. A. Goncalves, and

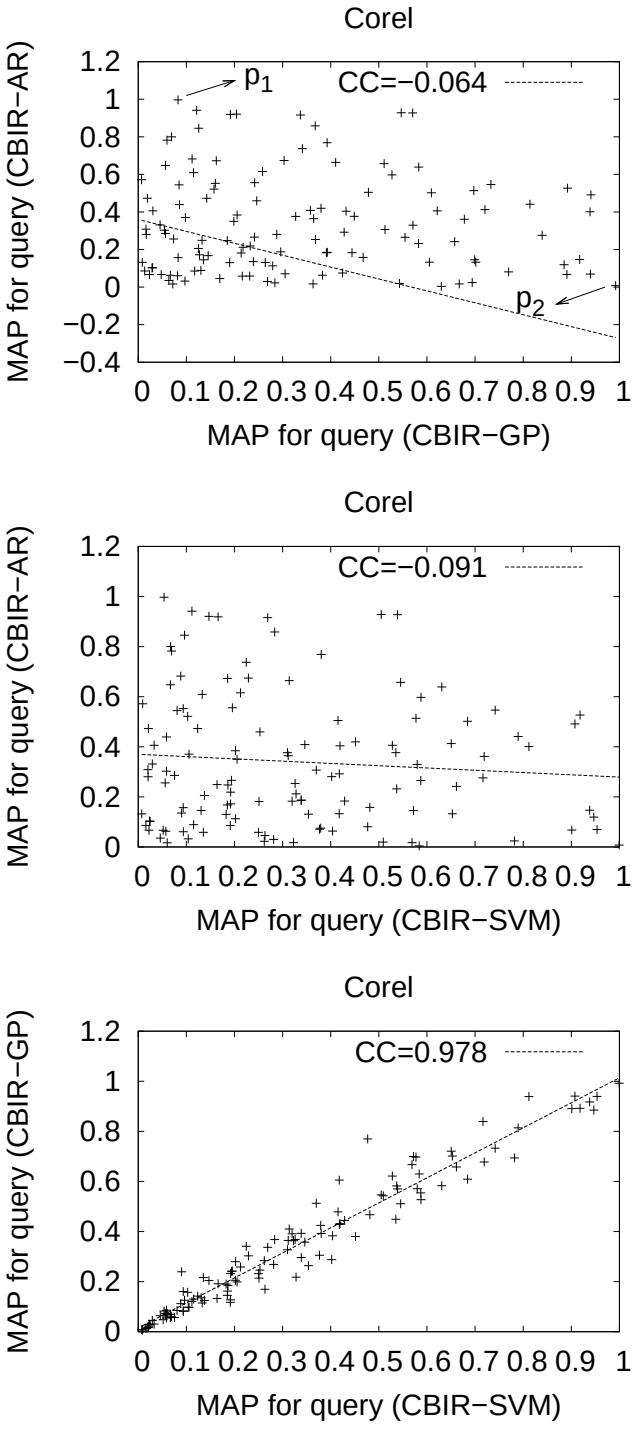

Figure 4: Each graph shows a scatter plot of the MAP values obtained for each query on two different schemes. On top: CBIR-AR x CBIR-GP; on middle CBIR-AR x CBIR-SVM; on bottom CBIR-GP x CBIR-SVM. The correlation coefficient (CC) is shown inside each graph.

W. Fan. Image Retrieval with Relevance Feedback based on Genetic Programming. In *SBBD*, pages 120–134, 2008.

[11] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 417–424, Cambridge, MA, 2007. MIT Press.

[12] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[13] P. Gosselin and M. Cord. Active learning methods for interactive image retrieval. *IEEE Transactions on Image Processing*, 17(7):1200–1211, 2008.

[14] J. Han, S. J. McKenna, and R. Wang. Learning query-dependent distance metrics for interactive image retrieval. In M. Fritz, B. Schiele, and J. H. Piater, editors, *ICVS*, volume 5815 of *Lecture Notes in Computer Science*, pages 374–383. Springer, 2009.

[15] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132, Cambridge, MA, 2000. MIT Press.

[16] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *In Proc. IEEE International Conference on Image Processing (ICIP)*, pages 750–753, 2000.

[17] Y. Hu, M. Li, and N. Yu. Multiple-instance ranking: Learning to rank images for image retrieval. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 1–8, 2008.

[18] C. Huang and Q. Liu. An orientation independent texture descriptor for image retireval. In *ICCCS*, pages 772–776, 2007.

[19] J. Huang, R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*, pages 762–768, 1997.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142, 2002.

[21] T. Joachims. Training linear SVMs in linear time. In *SIGKDD*, pages 217–226, 2006.

[22] V. Kovalev and S. Volmer. Color co-occurence descriptors for querying-by-example. In *MMM*, pages 32–38, 1998.

[23] J. Koza. *Genetic Programming: On the programming of computers by natural selection*. MIT Press, 1992.

[24] D. Lee and H. Kim. A fast content-based indexing and retrieval technique by the shape information in large image database. *Journal of Systems and Software*, 56(2):165–182, 2001.

[25] E. Levina and P. Bickel. The earth movers distance is the mallows distance: Some insights from statistics. In *Eighth IEEE International Conference on In Computer Vision (ICCV)*, volume 2, pages 251–256, 2001.

[26] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image*

Understanding, 106(1):59–70, 2007.

[27] Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *Learning to Rank Workshop in conjuntion with SIGIR*, 2007.

[28] T. Lu and C. Chang. Color image retrieval technique based on color features and image bitmap. *Inf. Processing and Management*, 43(2):461–472, 2007.

[29] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive content-based image retrieval using relevance feedback. *Comput. Vis. Image Underst.*, 88(2):55–75, 2002.

[30] F. Mahmoudi, J. Shanbehzadeh, A. Eftekhari-Moghadam, and H. Soltanian-Zadeh. Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern Recognition*, 36(8):1725–1736, 2003.

[31] B. Manjunath, J. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Trans. Circuits Syst. Video Techn.*, 11(6):703–715, 2001.

[32] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[33] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[34] D. Ritendra, J. Dhiraj, L. Jia, and Z. W. James. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, April 2008.

[35] H. Shao, J. W. Zhang, W. C. Cui, and H. Zhao. Automatic Feature Weight Assignment based on Genetic Algorithm for Image Retrieval. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, pages 731–735, 2003.

[36] R. Stehling, M. Nascimento, and A. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109, 2002.

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

[38] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[39] B. Tao and B. Dickinson. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327–342, 2000.

[40] M. Unser. Sum and difference histograms for texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):118–125, 1986.

[41] A. Veloso, H. M. Almeida, M. Gonçalves, and W. Meira. Learning to rank at query-time using association rules. In *SIGIR*, pages 267–274, 2008.

[42] A. Veloso, W. M. Jr., and M. J. Zaki. Lazy associative classification. In *ICDM*, pages 645–654, 2006.

[43] A. Williams and P. Yoon. Content-based image retrieval using joint correlograms. *Multimedia Tools Appl.*, 34(2):239–248, 2007.

[44] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, New York, NY, USA, 2007. ACM.

[45] J. A. M. Zegarra, N. J. Leite, and R. da S. Torres. Wavelet-based Feature Extraction for Fingerprint Image Retrieval. *Journal of Computational and Applied Mathematics*, 227(2):294–307, May 2009.

[46] L. Zhang, F. Lin, and B. Zhang. Support vector machine learning for image retrieval. In *ICIP (2)*, pages 721–724, 2001.

[47] J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.