Performance comparison of different similarity models for CBIR with relevance feedback

Daniel Heesch, Alexei Yavlinsky and Stefan Rüger

Department of Computing
South Kensington Campus, Imperial College London
London SW7 2AZ, England
{dh500,agy02,s.rueger}@imperial.ac.uk

Abstract. This paper reports on experimental results obtained from a comparative study of retrieval performance in content-based image retrieval. Two different learning techniques, k-Nearest Neighbours and support vector machines, both of which can be used to define the similarity between two images, are compared against the vector space model. For each technique, we determine both absolute retrieval performance as well as the relative increase in performance that can be achieved through relevance feedback.

1 Introduction

Content-based information retrieval remains a formidable challenge. The fundamental problem appears to be linked to our gaping ignorance regarding content-representation in the human brain, and a major breakthrough may thus have to wait for a corresponding breakthrough in the neurosciences. There undoubtedly exist taxing technological challenges arising from time and storage considerations. With grid computing and nano-technology advancing rapidly, however, these challenges are likely to be met in the foreseeable future. The problem of the semantic gap between primitive features on the one hand and image meaning on the other hand, however, is likely to persist for some time. One promising attempt to at least alleviate the problem has been relevance feedback. It has long been used profitably in traditional information retrieval [13] and is beginning to establish itself as a core paradigm in content-based information retrieval. In the context of image retrieval, relevance feedback requires the user to label retrieved images according to their relevance. A commonly employed techniques to achieve supervised learning through relevance feedback involves updating weights of a parametrized similarity function over several iterations of user-system interaction (see for example [12, 4]). This paper is concerned with the question of how retrieval performance and the effectiveness of relevance feedback is affected by the choice of the similarity model that underlies the ranking of retrieved images. The first similarity model is associated with support vector machines (SVM). This learning technique constitutes the fruit of a number of relatively recent advances in the theory of statistical learning [15]. Owing to their remarkable

generalization performance, they have in many areas replaced neural networks for prediction and classification problems alike. In CBIR, SVMs have recently been employed in [5] where they are used to determine feature weights following relevance feedback. Theoretical progress in support estimation [14] has recently provided the basis for the application of one-class SVMs to CBIR. In [1], for example, linear and non-linear kernels are used to capture single and multi-modal distributions of relevant images by finding closest-fitting spheres around positive examples.

The second learning technique is based on k-Nearest Neighbours (k-NN), a technique that has been widely used in non-parametric density estimation and classification [3]. A recent application to CBIR is found in the context of key-frame based video retrieval [11].

The paper is structured as follows. Section 2 briefly introduces the image features used. An exposition of the learning techniques and how they can be employed for similarity computation is given in section 3. Section 4 provides a brief description of the relevance feedback technique implemented and section 5 details the experimental set-up. Results will be presented in section 6 and the paper will end with conclusions in section 7.

2 Image features

Our study concentrates on the use of colour features for capturing image content. While any two such features will necessarily be correlated, we sought to reduce the representational overlap by defining features in different colour spaces and by giving different spatial emphasis as detailed below.

2.1 HSV colour histogram

We use a uniform quantization of the HSV colour space using 8 bins for each dimension. Two different features are defined. One feature is a global histogram with no preservation of spatial information, while the other feature consists of an array of five local histograms with the first four forming a partition of the image and the fifth covering the central 25%. This second feature achieves some preservation of local colour information. Moreover, by associating a high weight with the central area it lends itself well for capturing content of images where a centrally located object of interest is enveloped by irrelevant background. The size of the vector for the latter feature is thus $8^3 \times 5 = 2560$.

2.2 HMMD colour histogram

The HMMD (Hue, Min, Max, Diff) colour space, which is part of the MPEG-7 standard, derives from the HSV and RGB spaces. The Hue component is the same as in the HSV space, and Max and Min denote the maximum and minimum among the R, G, and B values, respectively. The Diff component is defined as the difference between Max and Min. Three components suffice to uniquely locate

a point in the colour space and thus the space is effectively three-dimensional. Following the MPEG-7 specification, we quantize the HMMD non-uniformly into 184 bins (for details about this quantization see [6]) with the three dimensions being Hue, Sum and Diff (Sum being defined as (Max+Min)/2). Two features are defined with respect to the HMMD colour space, a standard global histogram and the colour structure descriptor as detailed below.

2.3 Colour Structure Descriptor

This feature lends itself well for capturing local colour structure in an image. A 8×8 sliding window is moved over the image. Each of the 184 bins of the HMMD histogram contains the number of window positions for which there is at least one pixel in the area covered by the window with a colour that falls into the bin under consideration. This feature is capable of discriminating between images that have the same global colour distribution but different local colour structures. Although the number of samples in the 8×8 structuring window is kept constant (64), the spatial extent of the window differs depending on the size of the image. Thus, for larger images appropriate sub-sampling is employed to keep the total number of samples per image roughly constant (see [6] for details). The bin values are normalized by dividing by the number of locations of the structuring window and fall in the range $[0, 1]$.

3 Similarity models

3.1 Support Vector Machines

The idea behind support vector machines (SVMs) is to map n-dimensional vectors $\mathbf{x}$ of the input space non-linearly into high-dimensional vectors of the feature space and to then construct an optimal hyperplane in that feature space. Training a support vector machine, i.e. determining the optimal hyperplane, requires the solution of a quadratic optimization problem of the following form [15]:

$$\begin{aligned} \text{minimize} \quad & W(\alpha) = -\sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \qquad (1) \\ \text{subject to} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & \forall i : 0 \leq \alpha_i \leq C \end{aligned}$$

where the number of training examples is denoted by l and α is a vector of l variables where each component α_i corresponds to a training example $(\mathbf{x_i}, \mathbf{y_i})$. Here y_i denotes whether an image is relevant or not and $\mathbf{x}$ is a vector consisting of the bin frequencies of a colour histogram. The size of the optimization problem depends on the number of training examples l. An implementation with

good performance characteristics is SVM^{light} (available at http://www-ai.cs.uni-dortmund.de/svm_light) which breaks up the quadratic programming (QP) problem into smallest possible QP sub-problems and solves these analytically.

For our study, SVM^{light} is provided with sets of positive examples (the query images) and negative examples (randomly selected images that are not in the same category as the query) and then finds the optimal hyperplane separating positive from negative examples. Each new image is then assigned a score $s \in \mathbb{R}$, where $|s|$ measures the distance of the image from the hyperplane. A positive score indicates that the image is likely to be relevant. We here make the somewhat stronger assumption that the higher the score, the more relevant the image is likely to be. This relationship is also tacitly assumed in [5]. To turn the s score into a bounded dissimilarity value d, we compute for each image i to be ranked

$$d(i) = 1 - \frac{s_i - \min_j s_j}{\max_j s_j - \min_j s_j}. \tag{2}$$

3.2 k-Nearest Neighbour

We use a variant of the distance-weighted k-NN approach [8]. For each image i to be ranked we identify those images from the two sets of positive (P) and negative (N) images that are among the k nearest neighbours of i (nearness being defined by the l_1-norm). Using these neighbours, we determine the dissimilarity

$$d(i) = \frac{\sum_{n \in N} \mathrm{dist}^{-1}(i,n)}{\sum_{p \in P} \mathrm{dist}^{-1}(i,p)}. \tag{3}$$

In practice a small term is added to the distances so as to avoid division by zero. For all experiments, k is set to 5 and the number of positive and negative examples to 4 and 10, respectively.

3.3 Vector space model

All colour features are histograms which have been normalized such that all bins sum up to 1. A similarity metric for histogram features which has given good results in the past is the l_1-norm, $\sum_{j=0}^{N-1} |h_1(j) - h_2(j)|$, where the sum is over all bins. This value is divided by 2 to give an upper bound of 1. Since a query consists of multiple images, the overall distance between an image and a query Q is obtained by averaging the distance over all query images. Hence, the dissimilarity of a query to image i is

$$d(i) = \frac{1}{2|Q|} \sum_{q \in Q} \sum_{j=0}^{N-1} |h_i(j) - h_q(j)|. \tag{4}$$

Since the l_1-norm is defined in vector space, we will henceforth refer to this model as the vector space model (VSM).

3.4 Combination of image features

For each colour feature j we determine the distance $d_j(i)$ between the query to each of the images to be ranked according to one of the three similarity models described above. These feature-specific distances are then combined in a weighted sum $D(i) = \sum_j w_j d_j$, where $0 \leq w_j \leq 1$ and $\sum_j w_j = 1$. $D(i)$ is our estimate of the overall dissimilarity between a query and the ith image. The weights w_j provide the plasticity to be exploited by relevance feedback.

4 Relevance feedback

Thumbnails of retrieved images are displayed such that their respective distance from the centre of the screen is proportional to their dissimilarity to the query Q as computed by the system, $D_s(i)$ (for the GUI see [4]). By moving images further away or closer towards the center, the user provides a real-valued vector of distances, $D_u(i)$, which, in general, differ from the distances computed by the system. To update the feature weights, we minimize the sum of squared errors (SSE) between the user distances and the system-computed distances, i.e.

$$\mathrm{SSE}(w) = \sum_{i=1}^{N} [D_s(i) - D_u(i)]^2 = \sum_{i=1}^{N} \Big[\sum_j w_j d_j(i) - D_u(i) \Big]^2 \qquad (5)$$

subject to the constraint of convexity ($\sum_j w_j = 1$). Using one Lagrangian multiplier, this problem can readily be solved analytically. We have previously studied the effectiveness of different feedback scenarios and identified positive feedback as being superior to both mixed and negative feedback [4]. In the present study, we shall therefore confine our analysis to the case of positive feedback.

5 Experimental set-up

5.1 Selection of the image corpus

It is known that the choice of a test collection can heavily influence the perceived performance of a system [9]. We hence sought to use a corpus with a wide range of images and one that could easily be reproduced by other groups. It is desirable for evaluation purposes to work with a set of images for which a ground truth is available. We opted for images obtained from the Corel Gallery 380.000 package, which contains some 30.000 photographs, sorted into 450 categories.

The image collection was created with the aim of being able to assess the performance on realistic search tasks. Some Corel categories were of a very similar kind and were therefore merged. Other categories were of a very abstract nature (e.g. "Lifestyles") and were removed from the corpus. From the resulting collection, 50 categories were randomly chosen. A training set and a test set were then generated by randomly choosing for each set 20 images from each of the 50 categories leaving us with two mutually exclusive sets of 1000 images each.

5.2 Evaluation

For the present study we employ the two measures *recall* and *precision* to measure retrieval performance for *category searches* [7, 2]. Image queries are formed as follows: for each training image, three other images are randomly selected from the remaining set of images of that category. Each query thus consists of four images. For the k-NN and SVM model, ten negative examples are randomly chosen from the training set subject to the condition that they do not belong to the same category as the query images. Retrieval is performed from the test set and performance measured in terms of *mean average precision* which can be thought of being the area enclosed by the precision-against-recall graph and the recall axis (details in [16]).

As the Corel categorization provides us with a ground truth for each image, it is possible to automate the relevance feedback process. Among the 40 top ranked images as many at most three relevant images are randomly selected and their entries in the distance vector $d_u(i)$ set to 0. The weights are then updated as described in section 4. We allow for five such iterations of relevance feedback and evaluate performance for each iteration.

6 Results

6.1 Time performance

All three learning methods scale linearly with the number of images n to be ranked. The performance results for different values of n, obtained with a Pentium IV 1.8GHz processor, are depicted in Figure 1. The least computationally demanding model is VSM. With the l_1-norm as the similarity metric, computation of the distance between an image and a query of size q requires but $2qm$ additions (qm additions and the same number of subtractions), where m is the length of the feature vector $\mathbf{x}$.

In the case of support vector machines, training needs to be done only once per query since relevance feedback does not affect the position of the hyperplane. With the size of each colour feature vector ranging between 184 and 2560, and only 14 training examples (4 positive examples constituting the query and 10 negative examples), the time required for training the SVM is negligible. Once trained, computing the distance from the hyperplane for each of the n images requires the evaluation of

$$f(\mathbf{x}) = \sum_{i=1}^{l_s} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{6}$$

where l_s is the number of support vectors. Since the number of training examples place an upper bound on the number of support vectors, this function evaluation is also done swiftly. The slightly worse performance of the SVM compared to VSM results from the need to evaluate the polynomial kernel function k.

k-NN is computationally more expensive than both SVM and VSM. This is a consequence of the fact that for each image to be ranked, its k-nearest neighbours from among the training examples need to be determined. Given t training examples (here $t = 14$), this requires $2tqm$ additions to be carried out.

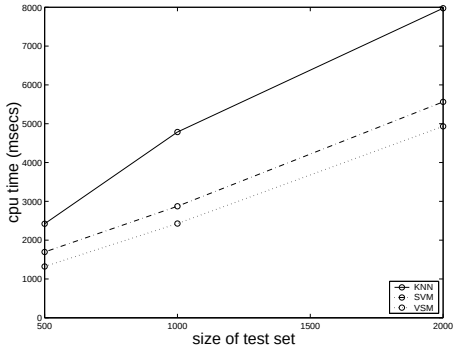

Fig. 1. All three similarity models scale linearly with the number of queries in the test collection. The computationally most expensive method is k-NN.

6.2 Absolute retrieval performance and gain through relevance feedback

Figure 2 depicts the precision-against-recall graphs for the three similarity models. Each graph is obtained by averaging over all 1000 queries. The upper graph in each plot represents the maximum performance that can be achieved in our retrieval model. It is obtained by determining for each individual query the feature weights that maximize mean average precision. These query-specific optima are found in turn by raster scanning the parameter space. The bottom graph represents performance prior to relevance feedback with each of the four features being given equal weight. The graph running nearly halfway between the upper bound and the baseline represents the result of five iterations of relevance feedback. The results are summarized in Table 1. We note a close similarity in

	before RF	after five iterations	upper bound
VSM	0.15 ± 0.17	0.18 ± 0.18	0.23 ± 0.19
k-NN	0.14 ± 0.16	0.17 ± 0.18	0.21 ± 0.19
SVM	0.13 ± 0.15	0.15 ± 0.17	0.19 ± 0.18

Table 1. Mean average precision values for the three similarity models ($\pm$ 1 stdev)

performance between the three models k-NN, VSM and SVM in terms of both absolute performance and their responsiveness to relevance feedback. The simple VSM offers a performance that is slightly superior to both k-NN and SVM.

In all three models, relevance feedback has a noticeable effect on performance, but it falls short of tuning the weights for maximum performance. In fact, when looking at the weights after five iterations, we have found that in many cases they were not closer to the optimal weights than the starting weights had been and may thus represent local optima.

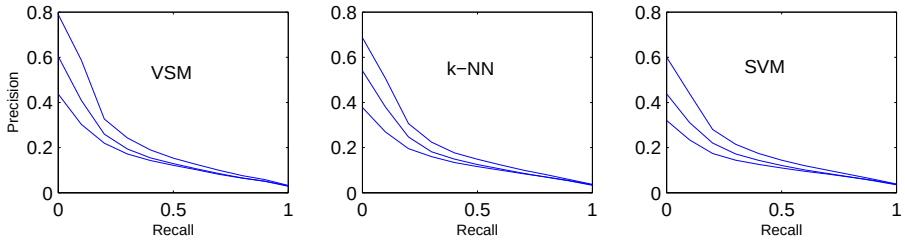

Fig. 2. Precision-against-recall graphs for the three similarity models. See text for details.

Figure 3 displays the relative gain through relevance feedback for all three models. It is evident that the models exhibit a difference in terms of their convergence behaviour under relevance feedback. While in the vector space model, performance increases after the first iteration and remains stable for all subsequent iterations, both k-NN and SVM display a steady increase over the first five iterations. It is known from previous studies (e.g. [10, 4]) that negative feedback tends to keep the system in a more flexible state than does positive feedback and results in a more gradual convergence as it is observed here. Although we explicitly restrict feedback to relevant images, both k-NN and SVM do rely on the presence of negative examples, while VSM does not. The conclusion that it is therefore the set of negative examples that moderates the effect of relevance feedback in k-NN and SVM is tempting. It is worth noting, however, that the negative examples are utilized once at the beginning of the query session for k-NN distance computation and hyperplane construction, and do not affect the computation of new weights during subsequent iterations. Their primary role is thus very different from the role of negative examples fed back by the user in [10] and [4], and further tests need to be carried out to substantiate such a claim.

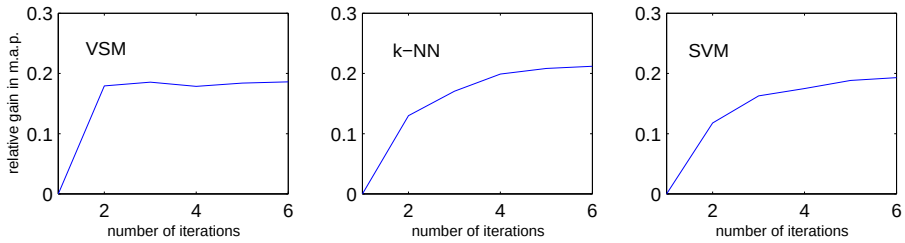

Fig. 3. Gains through relevance feedback for the three similarity models. The gradual increase for k-NN and SVM probably results from the effect of negative examples.

Although performances for the three models are very similar when averaged over all queries, the models exhibit notable differences for individual queries. In Figure 4, we have taken three colour features and plotted the resulting mean average

precision values for all weight combinations. Each row corresponds to a different query. For the upper query image all three similarity models produce similar performance surfaces with roughly coincident optima. For the lower query image the performance surface obtained for the VSM differs markedly from that of the other two models, suggesting that features that works well for one similarity model may not work quite so well for another model.

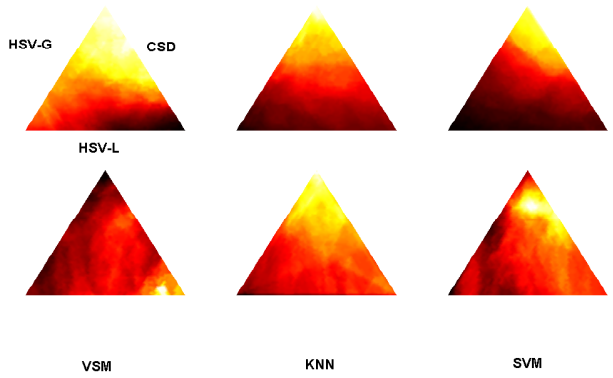

Fig. 4. Parameter simplices for three features and two butterfly queries (each row corresponding to a different query). Performance increases with decreasing gray-level. HSV-L: local HSV; HSV-G: globel HSV; CSD: colour structure descriptor

7 Conclusions

We have compared a number of performance aspects of two rather different techniques for computing similarities between images against a simple vector space model approach. The overall result from this study is that the more sophisticated learning techniques do not prove superior to the vector space model, while being computationally slightly more expensive. Both SVM and k-NN are known to work well with large sets of training examples so that one may not be able to fully exploit their potential in an interactive retrieval context where training is online involving but a small number of training examples.

All three models support an increase in performance through relevance feedback. We found that convergence for k-NN and SVM is more gradual and attributed this difference to the importance of negative examples in those models. Further investigations into the role of negative examples for a model's response to feedback are needed. In this context, the evaluation of one-class SVMs appears to be of particular interest as it relies solely on positive examples.

We have illustrated how the optimal feature combinations differ not only between queries but also between similarity models. If the set of top ranked images

is sufficiently different for each model using its optimal feature combination, combining models may help increase performance yet further.

Although the four features we considered are sufficiently different to allow performance gains through relevance feedback, they are all colour representations and thus capture but a fraction of the available image information. The addition of structure and texture features will undoubtedly help clarify some of the open issues this study has raised.

References

1. Y Chen, X Zhou, and T S Huang. One-class SVM for learning in image retrieval. In *Proc IEEE Image processing 2001*, 2001.
2. I J Cox, M L Miller, T P Minka, T V Papathomas, and P N Yianilos. The Bayesian image retrieval system, pichunter. *IEEE Transactions on Image Processing*, 9(1):20–38, 2000.
3. B V Dasarathy, editor. *Nearest Neigbour (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.
4. D Heesch and S Rüger. Performance boosting with three mouse clicks — relevance feedback for CBIR. In *Proceedings of the European Colloquium on IR Research 2003*. LNCS, Springer, 2003.
5. T S Huang. Incorporating support vector machines in content-based image retrieval with relevance feedback. In *Proc IEEE Image processing 2000*, 2000.
6. B S Manjunath and J-R Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11:703–715, 2001.
7. C Meilhac and C Nastar. Relevance feedback and category search in image databases. In *Proc. IEEE Int. Conf. Multimedia Comp. and Syst.*, pages 512–517, 1999.
8. T M Mitchell. *Machine Learning*. McGraw Hill, 1997.
9. H Müller, S Marchand-Maillet, and T Pun. The truth about Corel - evaluation in image retrieval. In *Proceedings of CIVR*, pages 38–49, 2002.
10. H Müller, W Müller, D M Squire, M.S Marchand-Maillet, and T Pun. Strategies for positive and negative relevance feedback in image retrieval. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000), IEEE, Barcelona, Spain*, 2000.
11. M Pickering and S Rüger. Evaluation of key-frame based retrieval techniques for video. *submitted to Elsevier Science and accepted for publication*, 2003.
12. Y Rui and T S Huang. A novel relevance feedback technique in image retrieval. In *ACM Multimedia (2)*, pages 67–70, 1999.
13. G Salton and M J Gill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., 1983.
14. B Schölkopf, J C Platt, J T Shawe, A J Smola, and R C Williamson. Estimation the support of a high-dimensional distribution. *Technical Report MSR-TR-99-87, Microsoft Research*, 1999.
15. V N Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
16. E M Voorhees and D Harman. Overview of the eigth Text REtrieval Conference (TREC-8). In *Proc. TREC*, pages 1–33 and A.17 – A.18, 1999.

Acknowledgements: This work was partially supported by the EPSRC, UK, and the ARC, Australia.