Semantic Similarity for Adaptive Exploitation of Inter-Query Learning

Iker Gondra, and Douglas R. Heisterkamp
Department of Computer Science, Oklahoma State University
Stillwater, Oklahoma 74078, USA

ABSTRACT

Relevance feedback has been used in many techniques for learning query modification and/or distance reweighting to improve the effectiveness of content-based image retrieval. These techniques, however, only use intra-query learning (i.e., short-term learning within a single query session). We propose a retrieval method that incorporates inter-query (i.e., long-term) learning into the query modification and distance reweighting framework. The semantic similarity of the current query with a set of past queries is used to control the exploitation of inter-query learning from historical data. For example, a local initial distance metric is created that is more informed than the commonly used default of Euclidean distance. The inter-query learning relies on the geometric view of one-class support vector machines for creating regions of support of past retrieval concepts. Our proposed method has been implemented using probabilistic feature relevance learning as the method for the intra-query distance reweighting and query modification. The efficacy of our approach is validated using real world data.

1. INTRODUCTION

Relevance feedback (RF) has been used in many techniques for learning query modification [6, 7, 12] or distance reweighting [6, 7, 10, 11]. Query modification aims at moving the query towards the region containing relevant images. Distance reweighting is based on updating the weights of a weighted distance metric so that relevant images are closer. Some systems incorporate both approaches [6, 7]. Probabilistic feature relevance learning (PFRL) [10] computes flexible metrics for producing retrieval neighborhoods that are elongated along less relevant dimensions and constricted along most influential ones. The technique has shown promise in a number of database applications. It, however, becomes less appealing in situations where all input variables have the same local relevance, and yet retrieval performance might still be improved by simple query shifting. On the other hand, MARS[12] attempts to improve retrieval performance by moving the query towards the region of the feature space containing the relevant images and away from the region containing non-relevant images. In [6], a principled approach is presented that combines PFRL, as in [10], with query shifting to try to achieve the best of both worlds. These techniques, however, only use intra-query learning (i.e., short-term learning within a single query session). Inter-query learning (i.e., long-term learning accumulated over the course of many query sessions) can also be used to enhance retrieval performance.

A few approaches [5, 8, 15] attempt inter-query learning. In [5] latent semantic analysis was used to provide a generalization of past experience. The images in a database are viewed as the fundamental vocabulary of the system. The RF from each query is considered as a document composed of many terms (images). Both [8] and [15] take the approach of complete memorization of prior history. In [8] the correlation between past image labeling is merged with low-level features to rank images for retrieval. The model estimates the semantic correlation between two images based on their co-occurrence frequency (i.e., the number of query sessions in which both images were labeled relevant). Intuitively, the larger the co-occurrence frequency of two images is, the more likely that they are semantically similar. In [15] the extra inter-query information is efficiently encoded by adding a virtual feature to the feature vector of an image.

In this paper, we propose a retrieval method that incorporates inter-query learning into the query modification and distance reweighting framework. The semantic similarity of the current query with a set of past queries is used to control the exploitation of inter-query learning from historical data. The inter-query learning relies on the geometric view of one class support vector machines (1SVM) for creating regions of support of past retrieval concepts. We implement our proposed method using PFRL as the method for the intra-query distance reweighting and query modification.

2. ONE-CLASS SUPPORT VECTOR MACHINE

In a one-class classification problem, data from only one of the classes (the target class) is available. For instance, user-labeled relevant images give us information about the user's high level concept. Thus, the task is to create a boundary around the target class such that most of the target data is included while, at the same time, minimizing the risk of accepting outliers [13].

The strategy that is followed in a 1SVM consists of mapping the training data to a higher dimensional feature space and then attempting to include most of it into a hypersphere of minimum size. Consider training data $\{\mathbf{x_j}\}_1^N$ where $\mathbf{x_j} \in \Re^d$ is the feature vector of the j^{th} user-labeled relevant image. Let $\Phi : \Re^d \rightarrow \Re^D$ be a non-linear mapping from the original (d dimensional) input space to the (D dimensional) feature space with $D \geq d$. The task is to minimize the following objective function (in primal form)

$$\min_{R\in\Re,\zeta\in\Re^N,\mathbf{a}\in\Re^D} R^2 + C\sum_{i=1}^{N}\zeta_i$$

with constraints that (almost) all the training data are within the hypersphere (i.e., $||\Phi(\mathbf{x_i}) - \mathbf{a}||^2 \leq R^2 + \zeta_i,\ \zeta_i \geq 0,\ i = 1, 2, \ldots, N$), where R and $\mathbf{a}$ are the radius and center of the hypersphere, respectively. The parameter $0 \leq C \leq 1$ is the soft-hard margin penalty and it gives the tradeoff between the size of the hypersphere and the number of training data that can be included. By setting partial derivatives to 0 in the corresponding Lagrangian we obtain $\mathbf{a} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x_i})$. Replacing partial derivatives into the Lagrangian and noticing that $\mathbf{a}$ is a linear combination of the training data (which allows us to use a kernel function), the following objective function (in dual form) is obtained

$$\min_{\alpha} \sum_{i=j=1}^{N} \alpha_i\alpha_j K(\mathbf{x_i},\mathbf{x_j}) - \sum_{i=j=1}^{N} \alpha_i K(\mathbf{x_i},\mathbf{x_i})$$

with constraints $0 \leq \alpha_i \leq C,\ \sum_{i=1}^N \alpha_i = 1$, where K is an appropriate Mercer kernel. We use the Gaussian kernel $K(\mathbf{x_i}, \mathbf{x_j}) = e^{-||\mathbf{x_i}-\mathbf{x_j}||^2/\sigma^2}$. A quadratic programming method is used to find the optimal α values in the objective function [13].

3. PROPOSED METHOD

Suppose that we have a retrieval method that performs query modification and distance reweighting. After the tth RF iteration for the ith query, let $\mathbf{q}_i^t \in \Re^d$ be the adjusted query point, $\mathbf{w}_i^t \in \Re^d$ be the adjusted query weights for an arbitrary weighted distance metric, and $\mathcal{R}_i = \{\mathbf{x_j}, y_j\}_1^m$ be the set of cumulative retrievals for the ith query, where $\mathbf{x_j}$ denotes the feature vector representing the jth retrieved image, and y_j is either 1 (relevant image) or 0 (non-relevant image) marked by the user as the class label associated with the jth retrieved image. At the end of the search session, after T RF iterations, intra-query learning is given by $\mathbf{q}_i^T$, $\mathbf{w}_i^T$, and $\mathcal{R}_i$. In general, this intra-query learning is lost when the search session is over.

Inter-Query Learning

Because of their straightforward interpretation as the density of past interaction in a local area of the feature space, we have chosen 1SVMs as our long term learning structure. Let $\mathcal{R}_i^+ = \{\mathbf{x_j} | \{\mathbf{x_j}, 1\} \in \mathcal{R}_i\}$ and $\mathcal{R}_i^- = \{\mathbf{x_j} | \{\mathbf{x_j}, 0\} \in \mathcal{R}_i\}$ be the set of cumulative relevant and non-relevant retrieved images, respectively. At the end of the search session, after T RF iterations, we use $\mathcal{R}_i^+$ as training data for a 1SVM. Then, we associate $\mathbf{q}_i^T$ and $\mathbf{w}_i^T$ with the resulting region of support (i.e., hypersphere) in feature space. Thus, the inter-query learning relies on the geometric view of 1SVMs for creating regions of support of past retrieval concepts. The basic idea is that future query images that fall within the same region of support can take advantage of inter-query learning. Thus, instead of "starting from scratch", the previously learned $\mathbf{q}_i^T$ and $\mathbf{w}_i^T$ can be exploited.

We expect to have overlapping regions of support and thus queries that fall into more than one hypersphere. Thus, in order to identify the regions of support that are most likely to contain relevant images, we have to determine semantic similarity between the query image's concept and the concepts associated with the hyperspheres into which it falls. By storing the user's RF about each retrieved image on a particular search session (i.e., $\mathcal{R}_i$) along with the resulting hypersphere h_i, we are able to capture the semantics of the retrieval concept associated with h_i, denoted by $\mathcal{C}(h_i)$. This information can then be used as a basis for determining semantic similarity. Therefore, in addition to 1SVM parameters, other information is stored in our long term learning structure which we will refer from now on as a hypersphere and is defined as $h_i = \{\mathbf{q}_i^T, \mathbf{w}_i^T, \mathcal{R}_i, \mathbf{a_i}, R_i\}$, where $\mathbf{a_i}$, and R_i are the center and radius of the hypersphere, respectively.

Semantic Similarity

Since for every query image, there is a corresponding hypersphere, we only need to be able to determine semantic similarity between concepts associated with hyperspheres. The intuition for determining semantic similarity between $\mathcal{C}(h_i)$ and $\mathcal{C}(h_j)$ is that if images are jointly labeled as relevant in both $\mathcal{R}_i$ and $\mathcal{R}_j$, it is likely that $\mathcal{C}(h_i)$ and $\mathcal{C}(h_j)$ have similar semantic content. Also, the larger the number of overlapping relevant images, the higher the semantic similarity between them can be expected. The number of overlapping images for which there is RF disagreement should also have an important negative effect on the semantic similarity. We now explain how the semantic similarity function is derived.

The basic idea is based on the observation that semantic similarity between $\mathcal{C}(h_i)$ and $\mathcal{C}(h_j)$ should be based on similarity between their corresponding RF distributions (i.e., $\mathcal{R}_i$ and $\mathcal{R}_j$). Let X_i be a random variable with sample space $S_i = \{(\mathbf{x_j}, y_j) | (\mathbf{x_j}, y_j) \in \mathcal{R}_i\}$ (i.e., an event is the labeling of an image as relevant or non-relevant). Let $P_i((\mathbf{x_j}, y_j)|\mathcal{R}_i)$ be the probability that a user assigns label y_j to $\mathbf{x_j}$ when searching for images belonging to $\mathcal{C}(h_i)$. Thus, $\forall(\mathbf{x_j}, y_j) \in S_i,\ P_i((\mathbf{x_j}, y_j)|\mathcal{R}_i) = 1$. Let's assume that $\mathcal{C}(h_i) = \mathcal{C}(h_j)$. Then, let X_{ij} be a random variable with sample space $S_{ij} = \{(\mathbf{x_j}, y_j) | \mathbf{x_j} \in (\mathcal{R}_i^+ \cap \mathcal{R}_j^+) \cup (\mathcal{R}_i^+ \cap \mathcal{R}_j^-) \cup (\mathcal{R}_i^- \cap \mathcal{R}_j^+) \cup (\mathcal{R}_i^- \cap \mathcal{R}_j^-)\}$ (i.e., events involving images that appear in both $\mathcal{R}_i$ and $\mathcal{R}_j$). Similarly, $P_{ij}((\mathbf{x_j}, y_j)|\mathcal{R}_i, \mathcal{R}_j)$ is the probability that a user assigns label y_j to $\mathbf{x_j}$ when searching for images belonging to $\mathcal{C}(h_i) = \mathcal{C}(h_j)$. Thus, $P_{ij}((\mathbf{x_j}, y_j)|\mathcal{R}_i, \mathcal{R}_j) = 1$ if $\mathbf{x_j} \in \mathcal{R}_i^+ \cap \mathcal{R}_j^+$ or $\mathbf{x_j} \in \mathcal{R}_i^- \cap \mathcal{R}_j^-$. Otherwise, $P_{ij}((\mathbf{x_j}, y_j)|\mathcal{R}_i, \mathcal{R}_j) = 0.5$ if $\mathbf{x_j} \in \mathcal{R}_i^+ \cap \mathcal{R}_j^-$ or $\mathbf{x_j} \in \mathcal{R}_i^- \cap \mathcal{R}_j^+$. We can use the *entropy impurity* [4] of X_{ij}'s distribution to measure the distance between the distributions of X_i and X_j. The entropy impurity (or just *entropy*), $i(X)$, of random variable X with sample space S is defined as $i(X) = -\sum_{x \in S} P(x)\ log_2 P(x)$, where $P(x)$ is the probability of event x. Observe that $i(X_{ij}) = |\mathcal{R}_i^+ \cap \mathcal{R}_j^-| + |\mathcal{R}_i^- \cap \mathcal{R}_j^+|$ (i.e., number of mismatches). Quantifying semantic distance in this way makes intuitive sense. As the number of mismatches increases, their corresponding event probabilities decrease, entropy (impurity) increases, and support for our initial assumption (i.e., that $\mathcal{C}(h_i) = \mathcal{C}(h_j)$) decreases.

Note that $0 \le i(X_{ij}) \le |S_{ij}|$. The normalized distance function $dist(\mathcal{C}(h_i), \mathcal{C}(h_j)) = \frac{i(X_{ij})}{|S_{ij}|}$ could be used as a measure of semantic distance between $\mathcal{C}(h_i)$ and $\mathcal{C}(h_j)$. For convenience, we convert to the normalized similarity measure $sim(\mathcal{C}(h_i), \mathcal{C}(h_j)) = \frac{|S_{ij}| - 2i(X_{ij})}{|S_{ij}|}$. Note that $-1 \le sim(\mathcal{C}(h_i), \mathcal{C}(h_j)) \le 1$. The reason for rescaling to the range [-1,1] is that it allows semantic disagreement to have an effect on the voting scheme that we use for combining evidence. This does not affect the ranking based on semantic similarity. Thus, the semantic similarity between $\mathcal{C}(h_i)$ and $\mathcal{C}(h_j)$ is defined as

$$sim(\mathcal{C}(h_i), \mathcal{C}(h_j)) = \frac{|S_{ij}| - 2i(X_{ij})}{|S_{ij}|}$$
$$= \frac{|\mathcal{R}_i^+ \cap \mathcal{R}_j^+|}{|S_{ij}|} - \frac{|\mathcal{R}_i^+ \cap \mathcal{R}_j^-| + |\mathcal{R}_i^- \cap \mathcal{R}_j^+|}{|S_{ij}|}$$

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

Proposed Method

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

The first stage sets $\mathbf{w}_z^0 = \{1/d\}_1^d$, $\mathbf{q}_z^0 = \mathbf{z}$ and computes an n by n "concept similarity" matrix C whose $(i,j)^{th}$ entry is $sim(\mathcal{C}(h_i), \mathcal{C}(h_j))$. Intuitively, $C_i = \sum_{j=1}^n sim(\mathcal{C}(h_i), \mathcal{C}(h_j))$ is the degree by which $\mathcal{C}(h_i), \forall h_i \in \mathcal{H}$ agree with (or are semantically similar to) $\mathcal{C}(h_i)$. Then, $\mathbf{q}_z^0$ and $\mathbf{w}_z^0$ are updated as follows

$$\mathbf{q}_z^0 \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{q}_i^T \right) + (1-\alpha)\mathbf{q}_z^0$$
$$\mathbf{w}_z^0 \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{w}_i^T \right) + (1-\alpha)\mathbf{w}_z^0$$

$$\gamma_i = \frac{max(0, C_i)}{\sum_{i=1}^n max(0, C_i)} \quad \alpha = \frac{\sum_{i=1}^n max(0, C_i)}{n^2}$$

Thus α adapts based on the density of homogeneous semantic concepts. For instance, if there is complete semantic agreement among $\mathcal{C}(h_i), \forall h_i \in \mathcal{H}$, $\alpha = 1$ and inter-query learning is completely exploited by setting $\mathbf{q}_z^0 = (1/n)\sum_{i=1}^n \mathbf{q}_i^T$ and $\mathbf{w}_z^0 = (1/n)\sum_{i=1}^n \mathbf{w}_i^T$. On the other hand, when there is complete semantic disagreement, $\alpha = 0$ and inter-query learning is not used.

With each RF iteration, $\mathcal{R}_z$ grows. In the second stage, the system uses this new information to revise its previous choices. Thus, after the tth RF iteration, the semantic similarity between the query image's concept and $\mathcal{C}(h_i), \forall h_i \in \mathcal{H}$ is determined. Then, based on this information, past inter-query learning choices are revised

$$\mathbf{q}_z^t \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{q}_i^T \right) + (1-\alpha)\mathbf{q}_z^0$$

$$\mathbf{w}_z^t \leftarrow \alpha \left(\sum_{i=1}^{n} \beta_i \mathbf{w}_i^T \right) + (1-\alpha)\mathbf{w}_z^0$$

$$\beta_i = \frac{max(0, sim(\mathcal{C}(h_z), \mathcal{C}(h_i)))}{\sum_{i=1}^{n} max(0, sim(\mathcal{C}(h_z), \mathcal{C}(h_i)))}$$

$$\alpha = \frac{\sum_{i=1}^{n} max(0, sim(\mathcal{C}(h_z), \mathcal{C}(h_i)))}{n}$$

In the third stage, $\alpha \leftarrow f_\alpha(t)$, where $f_\alpha(t+1) < f_\alpha(t)$ (i.e., α decreases so that, as the number of RF iterations increases, we rely more on intra-query learning). Then, intra and inter-query learning are combined

$$\mathbf{q}_z^t \leftarrow \alpha \mathbf{q}_z^t + (1-\alpha)\mathbf{q_{intra}}$$
$$\mathbf{w}_z^t \leftarrow \alpha \mathbf{w}_z^t + (1-\alpha)\mathbf{w_{intra}}$$

where $\mathbf{q_{intra}}$ and $\mathbf{w_{intra}}$ are the modified query location and distance weights computed by the particular query modification and reweighting method, based on intra-query learning $\mathcal{R}_z$. Thus, in this case, α determines the ratio of intra to inter-query learning to be used in processing the query. It adapts based on the density of homogeneous semantic concepts and the number of RF iterations. The second and third stages are repeated after each RF iteration.

4. PFRL

In PFRL [10], retrieved images with RF are used to compute local feature relevance. If we let the class label $y \in \{0, 1\}$ at query $\mathbf{x}$ be treated as a random variable from a distribution with the probabilities $\{Pr(1|\mathbf{x}), Pr(0|\mathbf{x})\}$, we have $f(\mathbf{x}) \doteq Pr(y=1|\mathbf{x}) = E(y|\mathbf{x})$. In the absence of any variable assignments, the least-squares estimate for $f(\mathbf{x})$ is $E[f] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$ where $p(\mathbf{x})$ is the joint density. Now given only that $\mathbf{x}$ is known at dimension $x_i = z_i$. The least-squares estimates becomes $E[f|x_i = z_i] = \int f(\mathbf{x})p(\mathbf{x}|x_i = z_i)d\mathbf{x}$. Here $p(\mathbf{x}|x_i = z_i)$ is the conditional density of the other input variables. In image retrieval, $f(\mathbf{z}) = 1$, where $\mathbf{z}$ is the query. Then $[(f(\mathbf{z}) - 0) - (f(\mathbf{z}) - E[f|x_i = z_i])] = E[f|x_i = z_i]$ represents a reduction in error between the two predictions. Thus, a measure of feature relevance at query $\mathbf{z}$ can be defined as $r_i(\mathbf{z}) = E[f|x_i = z_i]$. The relative relevance can be used as a weighting scheme for a weighted K-nearest neighbor search (KNN) with $w_i(\mathbf{z}) = \frac{e^{Vr_i(\mathbf{z})}}{\sum_{l=1}^{q} e^{Vr_l(\mathbf{z})}}$, where V is a parameter that can be chosen to maximize(minimize) the influence of r_i on w_i. For further details, see [10].

PFRL with Query Shifting

PFRL becomes less appealing in situations where all the input variables have the same local relevance and yet retrieval performance might still be improved by simple query shifting towards $\mu_r = \frac{1}{\mathcal{R}_z^+} \sum_{\mathbf{x} \in \mathcal{R}_z^+} \mathbf{x}$. A PFRL algorithm combined with query shifting (PFRL+$\mu_\mathbf{r}$) is summarized in Figure 1.

1. Initialize $t = 0$, $\mathbf{q}_z^t = \mathbf{z}$, $\mathbf{w}_z^t = \{1/d\}^d$, $\mathcal{R}_z = \emptyset$
2. Compute K nearest images to $\mathbf{q}_z^t$ using $\mathbf{w}_z^t$
3. User marks the K images
4. While More RF Iterations Do
   - 4.1. $t \leftarrow t + 1$
   - 4.2. $\mathcal{R}_z \leftarrow \mathcal{R}_z \cup \{\mathbf{x_j}, y_j\}_1^K$
   - 4.3. Update $\mathbf{w}_z^t$ using $\mathcal{R}_z$
   - 4.4. Compute $\mu_\mathbf{r}$; $\mathbf{q}_z^t \leftarrow \mu_\mathbf{r}$
   - 4.5. Compute K nearest images to $\mathbf{q}_z^t$ using $\mathbf{w}_z^t$
   - 4.6. User marks the K images

Figure 1: PFRL with query shifting (PFRL+$\mu_\mathbf{r}$)

Note that training data in PFRL+$\mu_\mathbf{r}$ (for computing the relative feature relevances used to determine the KNN in the next iteration) consists of all previous (cumulative) retrieved images. This is an improvement over the original PFRL where training data consists only of images retrieved at the current RF iteration.

PFRL with Query Shifting and Inter-Query Learning

In PFRL and PFRL+$\mu_\mathbf{r}$, all information collected during a search session is lost at the end of the session. We implement our proposed method using PFRL+$\mu_\mathbf{r}$ as the method for the intra-query distance reweighting and query modification. The M-tree [2] data structure is used for the efficient search of hyperspheres. This implementation of our proposed method (PFRL+$\mu_\mathbf{r}$+1SVM) is summarized in Figure 2. In the figure, $\mathbf{w_{pfrl}}$ refers to the distance weights as computed by PFRL.

5. EXPERIMENTAL RESULTS

In the following we compare the retrieval performance of PFRL, PFRL+$\mu_\mathbf{r}$, and PFRL+$\mu_\mathbf{r}$+1SVM on real data sets. We have also implemented two techniques that exploit inter-query learning, the virtual feature (VF) approach [15] and the statistical correlation (SC) method [8]. The retrieval performance is measured by *precision*,which is the fraction of relevant images in the retrieval set. The following data sets were used for evaluation:

Texture-there are 40 different texture images that are manually classified into 15 classes. Each of those images is then cut into 16 non-overlapping images of size 128x128. Thus, there are 640 images in the database. The

1. Initialize $t = 0$, $\mathbf{q}_z^t = \mathbf{z}$, $\mathbf{w}_z^t=\{1/d\}^d$, $\mathcal{R}_z = \emptyset$, $\alpha = 1$
2. Form $\mathcal{H} = \{h_j\}_1^n$
3. If $|\mathcal{H}| = 0$ go to 5
4. Exploit Inter-Query Learning
   - 4.1. Compute $\{\gamma_i\}_i^n$, α
   - 4.2. $\mathbf{q}_z^t \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{q}_i^T\right) + (1-\alpha)\mathbf{q}_z^t$
   - 4.3. $\mathbf{w}_z^t \leftarrow \alpha \left(\sum_{i=1}^n \gamma_i \mathbf{w}_i^T\right) + (1-\alpha)\mathbf{w}_z^t$
5. Compute K nearest images to $\mathbf{q}_z^t$ using $\mathbf{w}_z^t$
6. User marks the K images
7. While More RF Iterations Do
   - 7.1. $t \leftarrow t+1$
   - 7.2. $\mathcal{R}_z \leftarrow \mathcal{R}_z \cup \{\mathbf{x_j}, y_j\}_i^K$
   - 7.3. If $|\mathcal{H}| = 0$ go to 7.5
   - 7.4. Revise Inter-Query Learning
     - 7.4.1. Compute $\{\beta_i\}_i^n$, α
     - 7.4.2. $\mathbf{q}_z^t \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{q}_i^T\right) + (1-\alpha)\mathbf{q}_z^0$
     - 7.4.3. $\mathbf{w}_z^t \leftarrow \alpha \left(\sum_{i=1}^n \beta_i \mathbf{w}_i^T\right) + (1-\alpha)\mathbf{w}_z^0$
   - 7.5. Compute $\mathbf{w_{pfrl}}$, $\mu_\mathbf{r}$; $\alpha \leftarrow f_\alpha(t)$
   - 7.6. $\mathbf{q}_z^t \leftarrow \alpha \mathbf{q}_z^t + (1-\alpha)\mu_\mathbf{r}$
   - 7.7. $\mathbf{w}_z^t \leftarrow \alpha \mathbf{w}_z^t + (1-\alpha)\mathbf{w_{pfrl}}$
   - 7.8. Compute K nearest images to $\mathbf{q}_z^t$ using $\mathbf{w}_z^t$
   - 7.9. User marks the K images
8. Use $\mathcal{R}_z^+$ as training data for a 1SVM
9. Save $h_z = \{\mathbf{q}_z^t, \mathbf{w}_z^t, \mathcal{R}_z, \mathbf{a}_z, R_z\}$

Figure 2: PFRL with query shifting and inter-query learning (PFRL+$\mu_\mathbf{r}$+1SVM)

images are represented by 16 dimensional feature vectors. We use 16 Gabor filters (2 scales and 4 orientations).

Letter-there are 20,000 character images, each represented by a 16-dimensional feature vector. There are 26 classes of the 2 capital letters O and Q. The images are based on 20 different fonts with randomly distorted letters.

To determine the free parameters, a ten-fold cross-validation was performed for both data sets. Figures 3 and 5 show precision in the initial retrieval set (i.e., with no RF iterations) with respect to different data levels. The data level is the amount of accumulated inter-query learning (i.e., number of queries processed) relative to the number of images in the data set. An intra-query-learning-only RF approach forms the initial retrieval set by doing a KNN search. The VF approach requires at least one RF iteration. Thus, on initial retrieval,VF, PFRL and PFRL+$\mu_\mathbf{r}$ have the same performance as a KNN search. As we can observe from those figures, precision in the ini-

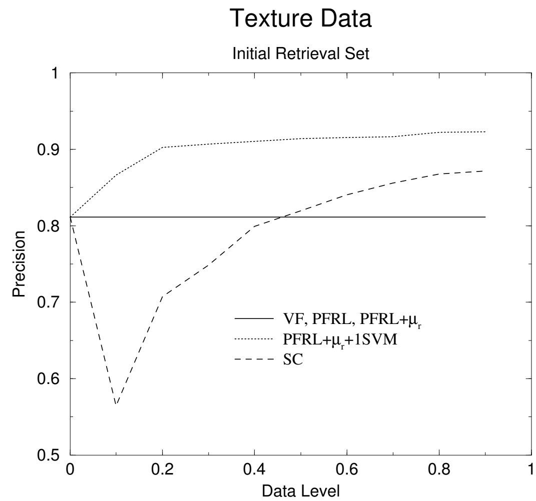

Figure 3: Precision vs. Data Level in Initial Retrieval Set.

tial retrieval set can be drastically improved by exploiting inter-query learning and keeps improving as the data level increases. This results in a reduction on the number of RF iterations that are needed to satisfy a query. Thus, from the user's perspective, it is very beneficial since users cannot stand too many RF iterations.

Figures 4 and 6 show precision after one RF iteration with respect to different data levels. As we can observe, precision increases after one RF iteration. The amount of improvement obtained when going from one to two RF iterations is much smaller. This is a desired property since users do not want to perform many RF iterations. We can observe that, with low data levels, there is an initial decrease in precision in both VF and SC. This is due to the fact that those methods use a fixed ratio of intra to inter-query learning to form the retrieval set. Our method is based on an adaptive weighting of inter-query learning and thus, does not suffer from this problem.

6. CONCLUSIONS

This paper presented a novel retrieval method that incorporates inter-query (i.e., long-term) learning into the query modification and distance reweighting framework. The experimental results show convincingly that PFRL with query shifting and inter-query learning outperformed either PFRL with query shifting or PFRL alone. The retrieval performance is constantly improved by the integration of inter-query learning. Furthermore, performance can be drastically improved in the initial retrieval set where both PFRL and PFRL+$\mu_\mathbf{r}$ require at least one iteration of RF to provide some improvement.

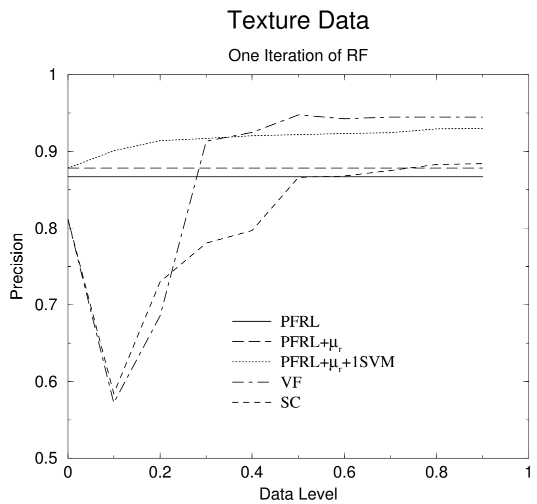

Figure 4: Precision vs. Data Level with One RF Iteration.

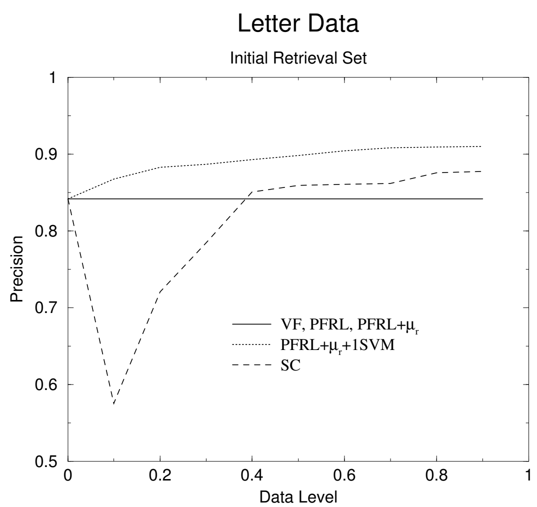

Figure 5: Precision vs. Data Level in Initial Retrieval Set.

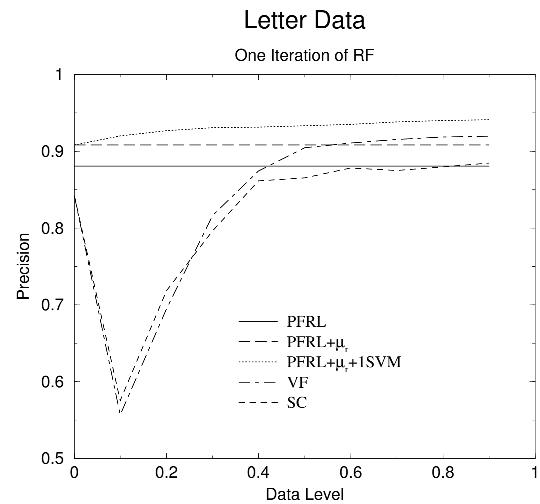

Figure 6: Precision vs. Data Level with One RF Iteration.

REFERENCES

[1] C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, Vol 2(2), pp. 121-167, June 1998.

[2] P. Ciaccia, M. Patella, P. Zezula, M-tree: An efficient access method for similarity search in metric spaces, *Proceedings of the 23rd International Conference on Very Large Databases*, pp.426-435, Athens, Greece, August 1997.

[3] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, UK, 2000.

[4] R. Duda, P. Hart, D. Stork, *Pattern Classification*, John Wiley and Sons, New York, NY, 2001.

[5] D. Heisterkamp, Building a latent-semantic index of an image database from patterns of relevance feedback, *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 132-135, Quebec City, Canada, August, 2002.

[6] D. Heisterkamp, J. Peng, H. Dai, Feature relevance learning with query shifting for content-based image retrieval, *Proceedings of 15th International Conference on Pattern Recognition*, pp. 250-253, Barcelona, Spain, September, 2000.

[7] Y. Ishikawa, R. Subramanys, C. Faloutsos, Mindreader: Querying databases through multiple examples, *Proceedings of the 24th International Conference on Very Large Databases*, pp. 433-438, New York, NY, August, 1998.

[8] M. Li, Z. Chen, H. Zhang, Statistical correlation analysis in image retrieval, *Pattern Recognition*, Vol. 35(12), pp. 2687-2693, December, 2002.

[9] J. Peng, B. Banerjee, D. Heisterkamp, Kernel index for relevance feedback retrieval in large image databases, *Proceedings of the 9th International Conference on Neural Information Processing*, pp. 187-191, Singapore, November, 2002.

[10] J. Peng, B. Bhanu, S. Qing, Probabilistic feature relevance learning for content-based image retrieval, *Computer Vision and Image Understanding*, pp. 150-164, Vol. 75(1/2), 1999.

[11] Y. Rui, T. Huang, Relevance feedback: A power tool for interactive content-based image retrieval, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8(5), pp. 644-655, September, 1998.

[12] Y. Rui, T. Huang, S. Mehrotra, Content-based image retrieval with relevance feedback in mars, *Proceedings of the IEEE International Conference on Image Processing*, pp. 815-818, Santa Barbara, CA, October, 1997.

[13] D. Tax, *One-class classification*, PhD thesis, Delft University of Technology, Delft, The Netherlands, June, 2001.

[14] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, NY, 1998.

[15] P. Yin, B. Bhanu, K. Chang, A. Dong, Improving retrieval performance by long-term relevance information, *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 533-536, Quebec City, Canada, August, 2002.