KERNEL INDEXING FOR RELEVANCE FEEDBACK IMAGE RETRIEVAL

Jing Peng

Electrical Engr. and Computer Science
Tulane University
New Orleans, LA 70118
jp@eecs.tulane.edu

Douglas R. Heisterkamp

Computer Science Department
Oklahoma State University
Stillwater, OK 74078
doug@ieee.org

ABSTRACT

Relevance feedback is an attractive approach to developing flexible metrics for content-based retrieval in image and video databases. Large image databases require an index structure in order to reduce nearest neighbor computation. However, flexible metrics can alter an input space in a highly nonlinear fashion, thereby rendering the index structure useless. Few systems have been developed that address the apparent flexible metric/indexing dilemma. This paper proposes kernel indexing to try to address this dilemma. The key observation is that kernel metrics may be non-linear and highly dynamic in the input space but remain Euclidean in induced feature space. It is this linear invariance in feature space that enables us to learn arbitrary relevance functions without changing the index in feature space. As a result, kernel indexing supports efficient relevance feedback retrieval in large image databases. Experimental results using a large set of image data are very promising.

1. INTRODUCTION

Relevance feedback (RF) is an attractive retrieval technique that allows a user to refine retrieval performance. A set of K nearest images to the query is first computed. The user then interacts with the retrieval algorithm by labeling the retrieved images as relevant or irrelevant. The algorithm dynamically adjusts its retrieval mechanism from the labeled images. This process repeats until the user is satisfied with the results. Relevance feedback retrieval has shown promise in a variety of image database applications [1, 2, 3, 4].

Formulating RF retrieval as a K nearest neighbor (NN) search allows such retrieval procedures to exploit highly customized retrieval metrics. However, NN search requires direct distance computation between the query and every image in the database. As such, the amount of computation required can be daunting when the image database is large. This demands effective strategies for database indexing or clustering in order to achieve greater computational efficiency. An index structure is optimized with respect to a given distance. Optimal retrieval demands dynamic metrics. However, when the metric is modified the index structure is no longer optimal and may not even be valid.

Few systems have been developed that address the apparent flexible metric/indexing dilemma in relevance feedback retrieval. While approximation based index techniques [5, 6] support relevance feedback retrieval with linear weightings (i.e., database objects still maintain the relative positions along each axis), they are unable to support nonlinear transformations, such as kernel distances [1, 7]. Kernel distances, however, often show significant performance improvement over Euclidean type of distance in relevance feedback retrieval [1, 7].

This paper presents a kernel indexing scheme that supports relevance feedback retrieval with kernel distances. We propose to build an index structure in a kernel induced feature space, where the images of database objects from the input space are obtained through a nonlinear mapping. Such an index structure potentially removes its dependency on the dimensionality of the input space, thereby mitigating its performance degradation with increasing input space dimensionality. In addition, since the construction of the feature space index is in the span of all data in the feature space, its behavior is inherently governed by the intrinsic dimensionality of the feature space.

2. KERNEL DISTANCE FOR RELEVANCE FEEDBACK RETRIEVAL

The kernel trick has been applied to numerous problems [8]. The kernel allows an algorithm to work in a feature space. If $\phi(\mathbf{x})$ is a mapping of a point $\mathbf{x}$ in the input space to the induced feature space

$$\mathbf{x} = (x_1, \cdots, x_q)^t \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \cdots, \phi_n(\mathbf{x})^t, \quad (1)$$

then the kernel calculates the dot product in the feature space of the images of two points from input space, $k(\mathbf{a}, \mathbf{b}) =<$

This work was supported by NSF Grant IIS-0136348.

$\phi(\mathbf{a}), \phi(\mathbf{b}) >$. Common kernels are Gaussian, $k(\mathbf{a}, \mathbf{b}) = e^{-\frac{||\mathbf{a}-\mathbf{b}||^2}{2\sigma^2}}$, and polynomial $k(\mathbf{a}, \mathbf{b}) = (1+ < \mathbf{a}, \mathbf{b} >)^d$.

One-class SVM kernel distance is based on the idea of using a hypersphere to describe relevant retrievals in the feature space. We want to compute the smallest hypersphere possible to include most relevant retrievals. This can be stated more precisely as [7] $\min_{R,\Xi,c} R^2 + \frac{1}{vl}\sum_i \xi_i$, subject to $\|\phi(\mathbf{x}_i) - c\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0$ for all i. Here $v \in [0, 1]$ controls the number of relevant retrievals that can be included in the hypersphere. Using Lagrangian multipliers one can solve the optimization problem and obtain

$$c = \sum_i \alpha_i \phi(\mathbf{x}_i) \tag{2}$$

with $\sum_i \alpha_i = 1$ and $0 \leq \alpha_i \leq \frac{1}{vl}$. Note that the center c (2) is a convex combination of relevant retrievals in the feature space. The resulting one-class SVM kernel distance is

$$D(\mathbf{x}, \mathbf{q}) = k(\mathbf{q}, \mathbf{q}) - 2\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{q}) + \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{3}$$

3. KERNEL INDEX

None of the index schemes [9, 5] developed so far support dynamic kernel distances. This motivates us to investigate kernel index algorithms for efficient relevance feedback retrieval with kernel distances.

The key to kernel indexing is this: while kernel distances are highly nonlinear and dynamic in the input space, they remain Euclidean in the kernel induced feature space. One-class SVMs adaptively computes the kernel distance (3) with relevance feedback by calculating the center (2) that is a linear combination of relevant retrievals. It is the moving center in the feature space that chases user's perceived similarity. While changing centers result in different kernel distances in the input space, the squared ($\|\phi(\mathbf{x}) - \phi((y)\|^2$) distance remains the same in the feature space.

We therefore build index structures in the feature space that are invariant to dynamic kernel distances in the input space. Note that individual dimensions cannot be directly accessed. However, there are two approaches to dodging this problem. One is to choose a orthonormal basis of the feature space by carrying out the Gram-Schmidt procedure. From the basis of the feature space one can apply VA-Files [5, 6] or tree structures to index the database for efficient NN computation.

The second approach to kernel indexing is metric space based techniques such as M-trees [9] and vantage-point trees (vp-trees) [10], where partitions are based on relative distance rather than absolute coordinate values. By selecting appropriate kernels, hyperspheres in the feature space thus implicitly yield a nonlinear structure in the input space. For the work in this paper, we explore metric space based kernel indexing.

3.1. M-tree in Feature Space

There are a number of metric space based index methods, any of which can be constructed in the feature space to give rise to a kernel index scheme. For the purpose of experimentation, we have used the M-tree indexing scheme in view of its effectiveness on large databases.

The internal nodes of an M-tree hold a collection of routing objects, whereas database objects are stored at leaf nodes. A routing object consists of the object itself, its covering radius that represents the maximum distance between the routing object and objects stored at the leaf nodes of its subtrees, the distance between the covering object and its parent node, and a pointer to the root. For each leaf node object, there is an associated distance between the object and the covering object of its parent. One of the key elements of the basic M-tree algorithm is to determine new covering objects such that the corresponding regions have minimum overlap and minimum covering volume. For details, see [9].

To build a M-tree in the feature space, we can apply the M-tree construction algorithm [9] using the distance $D(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{x}) - 2k(\mathbf{x}, \mathbf{y}) + k(\mathbf{y}, \mathbf{y})$. The same kernel distance is also used for NN search. In RF retrieval using the one-class SVM kernel distance, c (2) is initialized to the query $\phi(\mathbf{q})$. After each iteration, c is updated according to (2) with RF. Because ϕ is a nonlinear mapping, in general we have $\sum \alpha_1 \phi(\mathbf{x}_i) \neq \phi(\sum \alpha_i \mathbf{x}_i)$. Thus, NN search becomes $\|\phi(\mathbf{x}) - \sum \alpha_i \phi(\mathbf{x}_i)\|^2$. Effectively we would need two distance functions to segregate tree building and querying. However, this is simply an implementation issue. The distance in the feature space remains $\| \cdot - \cdot \|^2$.

4. EXPERIMENTAL RESULTS

In the following we compare two relevance feedback retrieval methods using relatively large sets of real image data: (1) OCKD - One-class SVM kernel distance (3); and (2) OCKD-MT - One-class SVM kernel distance coupled with M-tree indexing in the feature space. We used the Hemera Photo-Object image data set to evaluate our kernel index performance. This data set consists of 94800 images that are very heterogeneous and having annotated ground truth. Two sets of color histogram features (space spanned by the histogram features corresponds the input space discussed in the previous section) are used to represent the images. The first set of features uses a 1:1 scale and 6 regions, giving rise to 66 dimensions. The second set of features uses all scales and 5 regions, resulting in 198 dimensions. We are curious to know how an increase in dimensionality affects kernel indexing performance. The dimensions are arranged by scales, by regions and by zones. A set of 200 randomly selected images is used as query images.

In all the experiments, the features are first normalized to lie between 0 and 1. The kernel function we used is the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$. Procedural parameter γ was determined empirically.

The left panel in Figure 1 shows the average retrieval precision obtained by OCKD and OCKD-MT on the 66 dimensional image data, respectively, while the right panel in Figure 1 shows the average retrieval precision obtained by OCKD and OCKD-MT on the 198 dimensional image data, respectively. The two methods basically registered the same average retrieval precision.

Note that the overall poor retrieval precision can be attributed to the poor features we have been able to compute and highly heterogeneous nature of the Hemera Photo-Object image data that are available to us. In many cases, there are fewer than 30 images per class. In addition, great variations exist within each class. Better representation should improve retrieval performance. Average precision combined with performance cost still validates our proposed kernel indexing technique.

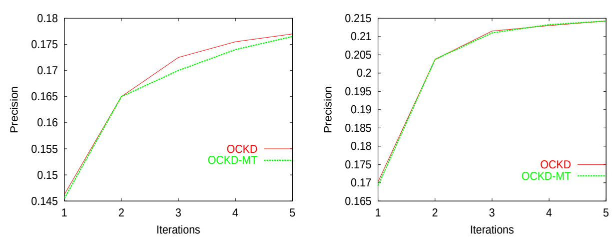

Fig. 1. Average retrieval precision achieved by OCKD and OCKD-MT on the 66 dimensional image data (left) and the 198 dimensional image data (right).

Table 1 shows the average number of distance calculations per query registered by the two methods in order to compute 20 nearest neighbors on the 66 (rows two and three) and 198 (rows four and five) dimensional image data as a function of iterations. The total number of distance calculations registered by OCKD over 5 iterations is 47.4×10^4 on both data sets. In contrast, the total number of distance calculations registered by OCKD-MT is 26214 on the 66 dimensional image data, and 178902 on the 198 dimensional image data. The results show that metric space based indexing in the kernel induced feature space is effective and can indeed significantly improve computational efficiency.

Note that there is an increase in distance calculation by OCKD-MT on the 198 dimensional image data. There are two factors that might contribute to the increase. First is the increase in dimensionality. Second is the γ values used in the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$. γ affects the distributions of the images in the feature space, which

Table 1. Average distance calculations by OCKD and OCKD-MT on the 66 and 198 dimensional image data sets.

Method	1	2	3	4	5
OCKD($\times 10^4$)	9.48	9.48	9.48	9.48	9.48
OCKD-MT	5191	5261	5258	5254	5250
OCKD($\times 10^4$)	9.48	9.48	9.48	9.48	9.48
OCKD-MT	35077	36123	36003	35872	35827

in turn influences the M-tree construction. Since we only used fixed γ values in the experiments ($\gamma = 1/45$ for the 66 dimensional image data, and $\gamma = 1/2$ for the 198 dimensional image data), the relative positions of the images in the feature space might have been skewed, thereby increasing distance calculations. The effect of γ on distance calculations is currently under investigation [1].

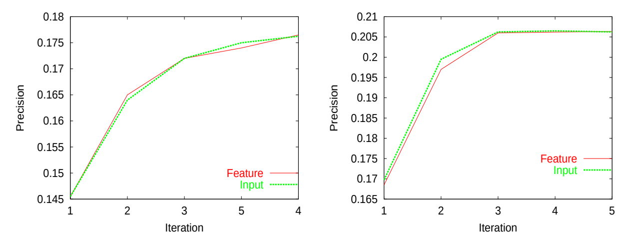

Fig. 2. Average retrieval precisions on the 66 dimensional (left) and 198 dimensional image data (right) over the 200 queries in the feature and input spaces, respectively.

To further explore the merits of feature space (kernel) indexing, we built M-trees in both feature and input spaces, and their indexing performances are evaluated on the two sets (66 and 198 dimensions) of image data. Note that since in the input space the nonlinear mapping (1) is reduced to the identity mapping $\phi(\mathbf{x}) = \mathbf{x}$, the corresponding kernel distance (3) is reduced to $D(\mathbf{x}, \mathbf{q}) = \mathbf{q}^t\mathbf{q} - 2\sum_i \frac{1}{l}\mathbf{x}_i^t\mathbf{q} + \sum_{i,j} \frac{1}{l^2}\mathbf{x}_i^t\mathbf{x}_j$, where $\mathbf{x}_i$ again represent relevant retrievals. This distance is used for retrieval and indexing in the input space.

Figure 2 shows average retrieval precisions achieved the two approaches (Input and Feature) over the 200 queries in the feature and input spaces, respectively. Note that retrieval based on the above distance was unable to achieve the level of precision that can be computed by the kernel distance (3) on the 198 dimensional image data. To make a fair comparison, we chose a kernel function so that its retrieval results in a similar precision performance.

[1] γ can be leveraged to actively align points in the feature space so that a balanced tree can be built.

Table 2. Average distance calculations per query on the 66 and 198 dimensional image data sets.

Method	1	σ	2	σ	3	σ	4	σ	5	σ
Input	17038	7657	16572	7401	16466	7350	16404	7345	16375	7306
Feature	5191	2117	5261	2152	5258	2177	5254	2190	5250	2184
Input	28237	10775	27329	10580	27110	10542	26983	10535	26965	10544
Feature	9911	3849	10246	4113	10226	4127	10198	4127	10188	4136

Table 2 shows the average number of distance calculations per query registered by the two approaches to compute 20 NNs on the 66 (rows two and three) and 198 (rows four and five) dimensional image data, along with standard deviations (σ). Note that the observed difference in indexing performance between the two approaches is statistically significant. Given the same level of precision , the results show that metric space based indexing in the feature space is indeed effective and outperforms input space indexing on the image data we have experimented with. A further examination shows that the trees in both the feature and input spaces have the same depth. Therefore, it seems logical to attribute the superior performance of kernel indexing to the nonlinear transformation that it employs to map similar data to adjacent locations in feature space. A similar phenomenon is also observed in support vector machine research.

5. CONCLUSIONS

This paper proposes a novel kernel index scheme that supports RF retrieval using kernel distances. Specifically, we propose to build an index structure in the feature space, where the images of database objects from the input space are obtained through a nonlinear mapping. Indexed feature space can be accessed directly by means of the kernel trick. Such kernel indexing in feature space potentially removes its dependency on the dimensionality of the input space, thereby mitigating its performance degradation with increasing input space dimensionality. The experimental results using a large set of images with high dimensionality show that our kernel indexing technique can potentially improve computational efficiency of NN search for RF retrieval.

6. REFERENCES

[1] D. Heisterkamp, J. Peng, and H.K. Dai, "An adaptive quasiconformal kernel metric for image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Kauai Marriott, Hawaii*, 2001, pp. 236–243.

[2] J. Peng, "Multi-class relevance feedback content-based image retrieval," *Computer Vision and Image Understanding*, vol. 90, no. 1, pp. 42–67, 2003.

[3] J. Peng, B. Bhanu, and S. Qing, "Probabilistic feature relevance learning for content-based image retrieval," *Computer Vision and Image Understanding*, vol. 75, no. 1/2, pp. 150–164, 1999.

[4] Y. Rui and T.S. Huang, "Optimizing learning in image retrieval," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina*, 2000, pp. 236–243.

[5] R. Webber, J.J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional space," in *Proceedings of the International Conference on Very Large Databases*, August 1998, pp. 194–205.

[6] P. Wu and B.S. Manjunath, "Adaptive nearest neighbor search for relevance feedback in large image databases," in *Proceedings of the ACM Multimedia*, November 2000, pp. 202–209.

[7] Y. Chen, X. Zhou, and T. Huang, "One-class svm for learning in image retrieval," in *Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece*, October 2001, pp. 815–818.

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

[9] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: an efficient access method for similarity search in metric spaces," in *Proceedings of the International Conference on Very Large Databases*, August 1997, pp. 426–435.

[10] P. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 1992, pp. 311–321.