Document Image Retrieval Using Feature Combination in Kernel Space

Ehtesham Hassan, Santanu Chaudhury, M Gopal
Department of Electrical Engineering, IIT Delhi
Email: hassan.ehtesham@gmail.com, sc chaudhury@gmail.com, mgopal@ee.iitd.ernet.in

Abstract—The paper presents application of multiple features for word based document image indexing and retrieval. A novel framework to perform Multiple Kernel Learning for indexing using the Kernel based Distance Based Hashing is proposed. The Genetic Algorithm based framework is used for optimization. Two different features representing the structural organization of word shape are defined. The optimal combination of both the features for indexing is learned by performing MKL. The retrieval results for document collection belonging to Devanagari script are presented.

Keywords—Document Indexing, Multiple Kernel Learning, Shape Descriptor

I. INTRODUCTION

The process of archiving the old manuscripts, books and scriptures through digitization have created large number of document images. The indexing of these document images is a challenging task. Different methodologies have been proposed for indexing the scanned document images. In the context of Indian script document image indexing, OCR based frameworks have been the primary approach. However the applicability of OCR based approach is limited for old and degraded documents and for scripts which does not have mature OCR technology. In [1], novel word image based document document indexing scheme using Distance Based Hashing (DBH) is presented. We present novel approach for application of multiple features for indexing using the Kernel based DBH. The features are optimally combined to improve indexing performance by performing Multiple Kernel Learning (MKL). The MKL have shown significant improvement in many classification problems by learning the optimal combination of different data sources, or by learning the optimal kernel for the classifier from the data itself [2]–[4]. The paper presents novel framework for performing MKL in indexing problem. In indexing problems, the optimization objective is maximization of retrieval performance. Considering the discrete nature of optimization objective, Genetic Algorithm based framework is proposed to perform MKL.

We define two different descriptors for word image representation utilizing the object shape informations like outer boundary, inner contours, upper and lower envelope curve on both side of the principal axis of object. The shape descriptors have been applied for indexing independently and by combination through MKL. The transformation of DBH to Kernel based DBH is obtained by the application of kernel trick. The kernelization of DBH provides the platform to perform indexing in kernel space. Because of the highly complex geometric structure of word object space, the traditional kernel embeddings may not be suitable. Therefore, the selection of initial kernel embedding for both shape descriptors is also learned by performing MKL. The organization of paper is as follows. The section II presents brief review of DBH and its transformation to kernel space. The section III presents MKL framework for indexing. The features used for experimentation are presented in section IV. The retrieval results for Devanagari script document collection is presented in section V. Finally, we conclude and give perspective of our work.

II. DISTANCE BASED HASHING IN KERNEL SPACE

A. Review: The DBH [5]

The DBH performs projection of objects on a carefully selected line such that inter object distances are preserved. The line projection is applied for performing object mapping. For two objects \((x_1, x_2)\) in space \((X, D)\), the line projection \(F_{x_1, x_2} : X \rightarrow R\) for object \(x\) is defined as

\[
F_{x_1, x_2}(x) = \frac{D(x_1, x)^2 - D(x_2, x)^2 + D(x_1, x_2)^2}{2D(x_1, x_2)}
\]

(1)

Here \(R\) represents the line connecting point representation of pivot objects \((x_1, x_2)\). The equation (1) can be used to define a rich family of functions having \(N(N-1)/2\) unique functions for each pairs of \(N\) objects from the database. The discretization of real hash functions defined by (1) is performed by using threshold parameters \(t_1, t_2 \in R\) as

\[
F_{t_1, t_2}^{x_1, x_2}(x) = \begin{cases} 
1 & \text{if } F_{x_1, x_2}^{x_1, x_2}(x) \in [t_1, t_2] \\
0 & \text{otherwise}
\end{cases}
\]

(2)

The selection of \((t_1, t_2)\) should be such that, \(F_{t_1, t_2}^{x_1, x_2}(x)\) maps approximately half the data points in \(X\) to 0 and half to 1, i.e. \(F\) generates balanced hash tables. Therefore the set \(V(x_1, x_2)\) of intervals \([t_1, t_2]\) for all pairs \((x_1, x_2) \in X\) is defined as

\[
V(x_1, x_2) = [t_1, t_2] \mid Pr_{x \in X}(F_{t_1, t_2}^{x_1, x_2}(x) = 0) = 0.5
\]

(3)

Now, the hash function family \(H_{DBH}\) is defined as

\[
H_{DBH} = F_{t_1, t_2}^{x_1, x_2}(x) \mid x_1, x_2 \in X, [t_1, t_2] \in V(x_1, x_2)
\]

(4)

The \(H_{DBH}\) can be generated by selecting \(N\) sample objects. The equation (4) can be used to define an indexing scheme.
by generating $L$ hash tables where each hash table corresponds to a $k$-bit hash function formed by concatenation of $k$ functions selected randomly from $H_{DBH}$. The retrieval process includes query hashing on hash tables (mapping of query on each hash table) and performing similarity search over the pool of objects collected from all the query buckets. The hash table parameters ($L$, $k$) are adjustable parameters and are defined by performance requirements.

B. Kernel based DBH

In the following discussion, the Kernel based DBH is presented. Considering $X$ as Euclidean vector space and $D$ Euclidean distance, the squared distance $D^2(x_1, x_2)$ can be expanded as $x_1^T x_1 + x_2^T x_2 - 2 x_1^T x_2$. Equation (1) is redefined as

$$F_{x_1,x_2}(x) = \frac{x_1^T x_1 - x_1^T x + x_2^T x - x_2^T x_2}{\sqrt{x_1^T x_1 - 2 x_1^T x_2 + x_2^T x_2}}$$

The above expression represents the line projection computation using dot products. The kernel methods increase the computational power of linear learning algorithms by mapping the data to high dimensional feature space [6]. The mapping $\phi : X \to S$ i.e. $x \to \phi(x)$ from input space $X$ to kernel space $S$, defines dot product $x^T x'$ in the kernel space as $\phi^T(x) \phi(x')$. It is clear that direct mapping to space $S$ can be implicitly performed by selecting a feature space which supports the direct computation of dot product using a nonlinear function in input space. The kernel function $k$ which performs such mapping is defined as

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \phi^T(x) \phi(x')$$

The expression shows the mapping to space $S$ by function $k$ happens implicitly without considering the actual form of $\phi$. The kernel space equivalent of the squared distance $D^2(x_1, x_2)$ is defined as $k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2)$. Therefore kernel space representation of (5) is defined as

$$F_{\phi(x_1), \phi(x_2)}(\phi(x)) = \frac{k(x_1, x_1) - k(x_1, x) + k(x_2, x) - k(x_1, x_2)}{\sqrt{k(x_1, x_1) - 2k(x_1, x_2) + k(x_2, x_2)}}$$

The above expression gives the formulation of line projection in kernel space defined by pivot objects $\langle \phi(x_1), \phi(x_2) \rangle$. Equation (6) can be discretized by defining the thresholds as discussed in the section II-A. Following the procedure discussed in section II-A, we can generate family of hash functions $H_{DBH}$ by applying the discrete hash functions defined for mapping function (6). The indexing framework and retrieval procedure remains same as traditional DBH (section II-A).

III. MKL BASED FEATURE FUSION FOR INDEXING

Equation (6) represents line projection computation in kernel space which defines the Kernel based DBH. The expression provides option to learn the kernel for indexing using set of base kernels. The problem is defined as MKL for indexing. In this case instead of single kernel, parameterized combination of kernels i.e. $k(x_1, x_2) = \sum_{i=1}^{N} w_i k_i(x_1, x_2)$ is considered. The resultant kernel should satisfy Mercer’s condition therefore all the weights should be positive real number i.e. $\forall i, w_i \geq 0$. The optimization objective of MKL problem is to maximize the retrieval performance of indexing scheme. The weights ($w'$) are optimization parameters. The existing MKL formulations have continuous objective functions and apply conventional gradient based optimization algorithms. The current optimization problem is discrete in nature while the parameter space is continuous. The discrete nature of optimization objective does not allow application of gradient based algorithms. For such optimization tasks evolutionary algorithm (EA) can provide efficient solution. The Genetic Algorithm (GA) is type of EA which is well suited for global optimal parameter search in complex spaces. Additionally GA works with raw objectives when compared with conventional techniques. Therefore we formulate MKL for indexing in GA like paradigm. The fitness function for GA population string is evaluated as retrieval performance for a validation query set.

Using base kernel set computed for different features, the above framework learns optimal combination of features for indexing. In the document indexing problem, selection of suitable kernel is difficult because of complex geometric structure of word object space. Therefore the kernel selected by traditional methods may not provide suitable kernel space representation. In this case, the above discussed framework is applied to learn the optimal kernel for each feature sets too. Therefore, the document indexing will follow 2-stage MKL process. The 1st stage MKL learns the optimal kernel for each feature set. The 2nd stage MKL learns the optimal combination of the feature sets for indexing.

The precision oriented retrieval i.e. correct matches in K nearest neighbors in retrieval result is taken as GA fitness value. The tournament selection is applied to chose individuals for successive population generation. It selects $p$ individuals randomly from the population and individual with highest fitness among the selected $p$ is placed in Mating Pool. The process is repeated for $M$ times, here $M$ is population size and $p$ is tournament size. The reproduction operators for offspring generation from the Mating Pool consists single point crossover and uniform mutation. We have applied elitist selection strategy for constructing new population for successive iteration. The elitist selection combines offspring with current population and selects $M$ best individuals with based on their fitness value. The distance computation for nearest neighbor search is performed in kernel space using equation (5).

IV. FEATURE DESCRIPTION

The shape descriptor proposed in [1] represents the structural organization of an object in 2D histogram and is
fundamentally based on shape context [7]. For the set of $n$ descriptor points on object shape, there are $n(n - 1)$ point-pair arrangements. The histogram represents distance and orientation based distribution of these point-pair arrangements. We name the histogram as point distribution histogram ($pdh$) and consider its fourier coefficient as shape descriptor. The $pdh$ is computed as sum of shape contexts for the point set. The empirical evaluation have shown that robustness of descriptor to broken links and noisy ink dots increases by considering partition based approach for descriptor computation. In this case, the word image is divided in fix partitions of equal width (Figure 1). The descriptor is computed by arranging $pdh$ for each partition in respective order and taking the absolute of fourier transform of resulting histogram image. The $pdh$ for each partition is computed w.r.t. points lying in that partition. The partition based approach for descriptor computation nullifies the affect of distortion or noise in one partition to other partitions. Additionally, it also helps to retrieve partially similar matches and preserves character sequence information. The descriptor is computed as follows:

- descriptor points extraction followed by $pdh$ computation for all the partitions.
- descriptor is defined as $||Fourier(h_1 : \ldots : h_{parts})||$, $h_i$ represents the $pdh$ for $i^{th}$ partition.

The global shape features are predominantly represented by low frequency fourier coefficients and local shape features like internal contours, sharp curves and broken links contribute in high frequency fourier coefficients. We have adopted two different methodologies to extract descriptor points over object shape, leading to two different shape descriptors. The first methodology for descriptor point extraction involves envelope curve detection of word’s shape (Figure 2). A set of points is sampled on the envelope curves of word as descriptor points. These points are utilized to compute the **Envelope Curve based Shape Descriptor (ECBSD)**, which represents the outer boundary information of word shape. The envelope curves of a word remain considerably tolerant to different font properties and incorporates font invariance in the descriptor. The sampling is performed such that minimum distance between sampled points is maintained to ensure their uniform distribution over envelopes.

In many Indian scripts, some characters have similar outer boundary e.g. \{/sha, /pa\} and \{/ba, /va\} character pairs in Devanagari (Figure 3). However the appearance of inner contours distinguishes their semantics. We follow grid based approach to extract semantic information inherent in the inner contours of word shape to define their unique representation. For descriptor point extraction, we overlay a logical grid over the word shape. The transition points \{1 → 0, 0 → 1\} over grid are selected as descriptor points (The red points in figure 4). The process of point extraction gives due importance to inner contours for defining representation. Using the extracted set of points **Grid based Shape Descriptor (GBSD)** is computed as discussed earlier. The shape characteristics represented by the descriptors complement each other, therefore their optimal combination can improve the indexing performance by refining the grouping of similar points performed by hashing.

**V. Results**

The indexing and retrieval experiment is performed for Devanagari script document collection with 417 images. The gray to binary conversion of original images is performed by Otsu’s method. The word image segmentation is performed by projection profile method leading to dataset consisting 6680 images. The preprocessing steps for ECBSD and GBSD computation included bound detection and normalization of bounded image to $48 \times 144$. The $12 \times 36$ grid is overlaid on image for GBSD computation. In general, Devanagari script words consists of 3 to 6 characters, thus for descriptor computation word images have been split in $1 \times 4$ partitions. The $pdh$ dimension for each partition is selected as $40 \times 30$ i.e. shape context for each point is computed for 40 distance and 30 angle bins. The partitions and $pdh$ dimension parameters are same for both set of shape descriptors. The validation query set for GA fitness evaluation consists of 217 images. The complete setup is tested for query set containing 177 query images.
Table I

<table>
<thead>
<tr>
<th>Retrieval Results (F-score)</th>
<th>L = 15, k = 8</th>
<th>L = 20, k = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBH(GBSD)</td>
<td>0.791</td>
<td>0.796</td>
</tr>
<tr>
<td>DBH(ECBSD)</td>
<td>0.767</td>
<td>0.770</td>
</tr>
<tr>
<td>KernelDBH(GBSD)</td>
<td>0.803</td>
<td>0.806</td>
</tr>
<tr>
<td>KernelDBH(ECBSD)</td>
<td>0.779</td>
<td>0.781</td>
</tr>
<tr>
<td>KernelDBH(GBSD + ECBSD)</td>
<td>0.832</td>
<td>0.834</td>
</tr>
</tbody>
</table>

For learning the kernel for both descriptors (1st stage of MKL), base kernels included linear, polynomial (2nd order), gaussian (variance = \{1, 10\}) kernels. The kernel combination weight encoding is done in 6-bit binary string (For both stages). For both stage of MKL, initial GA population consisted 40 randomly generated strings and GA routine is simulated for 60 generations. The retrieval performance is measured by F-score (harmonic mean of precision and recall). The results for different hash table parameters are listed in table I. The increase in hash function length (k) from 8 to 10-bit creates more buckets. However object distribution in more populated buckets does not change much therefore retrieval results improve from 15 to 20 hash tables.

VI. CONCLUSION

The paper presents novel feature fusion approach for word based document indexing. The GA based framework to perform MKL for indexing problem is presented. The framework uses Kernel based DBH for performing MKL. Two different shape descriptors for word images are defined. The experiment have been performed for different Indian scripts, but results are reported only for Devanagari script documents.

REFERENCES


