Fast Key-Word Searching Using 'BoostMap' based Embedding

Raid Saabni
Faculty of Engineering,
Tel-Aviv University.
Triangle R&D Center, Kafr Qarea, Israel
saabni@cs.bgu.ac.il

Alexander Bronstein
Faculty of Engineering,
Tel-Aviv University
Tel-Aviv, Israel
bron@eng.tau.ac.il

Abstract

Dynamic Time Warping (DTW), is a simple but efficient technique for matching sequences with rigid deformation. Therefore, it is frequently used for matching shapes in general, and shapes of handwritten words in Document Image Analysis tasks. As DTW is computationally expensive, efficient algorithms for fast computation are crucial. Retrieving images from large scale datasets using DTW, suffers from the constraint of linear searching of all sample in the datasets. Fast approximation algorithms for image retrieval are mostly based on normed spaces where the triangle inequality holds, which is unfortunately not the case with the DTW metric.

In this paper we present a novel approach for fast search of handwritten words within large datasets of shapes. The presented approach is based on the BoostMap [1] algorithm, for embedding the feature space with the DTW measurement to an euclidean space and use the Local Sensitivity Hashing algorithm (LSH) to rank the k-nearest neighbors of a query image. The algorithm, first, processes and embeds objects of the large data sets to a normed space. Fast approximation of k-nearest neighbors using LSH on the embedding space, generates the top k-ranked samples which are examined using the real DTW distance to give final accurate results. We demonstrate our method on a database of $45,800$ images of word-parts extracted from the IFN/ENIT database [11] and images collected from 51 different writers. Our method achieves a speedup of 4 orders of magnitude over the exact method, at the cost of only a 2.2% reduction in accuracy.

Keywords: Word Searching; Adaboost; BoostMap; Dynamic Time Warping; Embedding; Nearest Neighbor;

1 Introduction

Methods based on Dynamic Time Warping (DTW) have been proved relatively efficient and effective for matching shapes of handwritten words in script recognition tasks [16, 10, 13]. Manmatha *et al.* [10, 13], have used DTW with a set of features taken from the upper and lower profile to match shapes of words in handwritten documents. Saabni and El-Sana [16, 17], used DTW with features extracted from contours [16] or sliding windows [17] to compare shapes for keyword searching tasks. In keyword searching and script recognition tasks, shapes have to be matched to sets of multiple appearances of different words. To find the best match to a given shape, a similarity criterion has to be calculated to each shape in these large sets. Approximate Nearest Neighbors (ANNs) methods such as k-d trees, Locality Sensitive Hashing and Semantic Hashing, provide computationally efficient procedures for finding objects similar to a query object in large datasets. These methods have been successfully applied to search web-scale datasets that can contain millions of images. Unfortunately, the key assumption in these procedures is that objects in the dataset lie in a Euclidean space. This assumption is not valid with a metric space based on the DTW distance.

A lot of work has been done on embedding finite metric spaces into low-dimensional normed spaces in order to enable efficient and fast nearest neighbor extraction. [3, 20]. In domains with a computationally expensive distance measure, significant speed-ups can be obtained by embedding objects into a normed space with an efficient distance measure such as Euclidean or pseudo-Euclidean spaces. Among these methods used for efficient retrieval are the Lipschitz embeddings, FastMap [3], MetricMap [21] and BoostMap [1].

Athitsos *et al.* [1], presented the 'BoostMap', a method to embed any metric space into an Euclidean space, in which similarities can be rapidly measured using a weighted Manhattan distance. Embedding construction is formulated as a machine learning task, where Adaboost is used to combine many simple, 1D embeddings into a multidimensional embedding that preserves a significant

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.204

amount of the proximity structure in the original space. In this paper, we use a variation of BoostMap algorithm for embedding the feature space of all different shapes of a large lexicon words to an euclidean space. Embedding is performed using the DTW distance and the Adaboost algorithm to determine a small subclass of samples to be used for embedding to a low dimensional euclidean space. We use Linear Discriminant Analyses(LDA) and Principal Component Analyses(PCA), to help Adaboost to determine the subclass of classifiers and to generate the right training sets. In the next step, we find the approximate k-nearest neighbors using the LSH method, which in it's turn produces the required short list. On this list, we apply the expensive matching methods, yet keep the search time reasonable and constant.

The rest of this paper is organized as follows: in Section 2 we briefly overview some of the related work, in Section 3, we describe our approach in details and analyze results of the experimental tests in Sections 4.

2 Related Work

Shape matching algorithms, constituting the core of all handwriting recognition systems, can be roughly classified as *pixel*- and *feature-based* approaches [10]. Pixel-based matching approaches measure the similarity between the two images in the pixel domain using various metrics, such as the Euclidean distance map (EDM), XOR difference, or the sum of square differences (SSD). In feature-based matching, images are compared using representative features extracted from them. Similarity measurements such as DTW and point correspondence are typically defined on the feature domain.

Many systems for word spotting and searching presented in previous work are based on variants of DTW using different sets of features and giving relatively good results comparing to the competing techniques [10]. Manmatha *et al.* [10] were among the first to introduce DTW for word spotting. They examined several matching techniques and showed that DTW, in general, provides better results. Rath and Manmatha [13] preprocessed segmented word images to create sets of one-dimensional features, which were subsequently compared using DTW. They also analyzed a range of features suitable for matching words using DTW [12]. Shrihari *et al.* [19] presented a design of a search engine for handwritten documents. They indexed documents using global image features, such as stroke width, slant, word gaps, as well as local features that describe the shapes of characters and words. A segmentation-free approach for keyword search in historical documents was proposed by Gatos *et al.* [4]. Their system combines image preprocessing, synthetic data creation, word spotting and user feedback techniques. Saabni and El-Sana [16] presented an algorithm for searching Arabic keywords in handwritten documents. In their approach, they used geometric features taken from the contours of the word-parts to generate feature vectors. DTW uses these real valued feature vectors to measure similarity between word-parts.

A lot of work has been done on embedding finite metric spaces into low-dimensional normed spaces in order to enable efficient and fast nearest neighbor extraction. Such embeddings have been extensively studied in pure mathematics [9, 22], and have found application in a variety of settings [3, 20]], Usually using one of the l_p norms. In domains with a computationally expensive distance measure, significant speed-ups can be obtained by embedding objects into another space with a more efficient distance measure. Several methods have been proposed for embedding arbitrary spaces into a Euclidean or pseudo-Euclidean space. Some of these methods, in particular Multidimensional Scaling (MDS)[22], Bourgain embeddings [7], Locally Linear Embedding (LLE) [14], need to evaluate exact distances between the query and most or all database objects, and thus are not designed for efficient nearest neighbor retrieval. Methods that can be used for efficient retrieval include Lipschitz embeddings, FastMap [3], MetricMap [21] and BoostMap [1].

Efficient embedding of the EMD metric (a closely related metric to DTW), to a normed space have been presented by Indyk and Thaper[8] and used later by Grauman and Darrell [6]. Indyk and Thaper[8] use a randomized multi-scale embedding of histograms into a space equipped with the l_1 norm. Additional efficient embedding of the EMD metric have been presented by Shridonkar and Jacobs[18] using a novel method for approximating the EMD distance of two histograms using a new metric on the weighted wavelet coefficients of the difference histogram using the l_1 distance. This new metric was experimentally shown to follows EMD closely without any significant performance difference. The wavelet EMD metric can be computed in $O(n)$ time.

3 Our Approach

In the presented approach we treat words in a holistic manner, and use the Multi Angular Descriptor (MAD) descriptor [15], to convert words to vectors in a feature space. The MAD descriptor can deal directly with gray level and binary shapes representing words with single and multi components. Even though, all generated feature vectors can have the same dimension, any l_p distance is not an adequate measurement due to time shift and non linear deformation, Therefore the DTW metric is used for shapes similarity measurement. To find the best match to a given query image, a similarity criterion has to be calculated to each shape in large sets of shape samples. For example, in a typical task for word recognition, a given shape have to be matched against multiple appearances of all different words in the lexicon which may exceed few millions when considering many appearances of each word in the lexicon. In such cases, it is desirable to make as few distance calculations as possible. A known and familiar technique to achieve such improvement includes mapping (embedding) the set of shapes represented as feature vectors in some feature space into a normed low-dimensional space and then conduct the search in that space using multidimensional indexing methods. Intuitively, such embedding is required to be distance preserving, i.e, distances in the embedding space have to approximate distances of the objects in the original space, but searching time in the embedding space is significantly reduced.

Let L be a lexicon of n words, S_{w_i} be the set of all available shapes of different appearances of the word $w_i \in L$ and SL, the set of all shapes in all sets S_{w_i} for each $w_i \in L$. Given a shape s of a word to be recognized (searched), we will have to conduct a matching procedure comparing s to all shapes in SL calculating the minimum distance or minimum average distances. The complexity of this procedure is linear to the size of SL, which is not affordable in many recognition tasks. Since DTW distances are pseudo distances that do not satisfy the triangle inequality, embedding SL into an Euclidean space, will enable approximate and fast searching with sub-linear time of the size of SL. After embedding, the approximate $k-NNs$, can be used efficiently in a filter and refine approach. In a second phase, we calculate the real DTW distance to a short list of the $k-top$ ranked words to determine final results.

The system goes through three stages to search for a given shape of a word within a large lexicon. In the first stage, all shapes in SL are embedded into the Euclidean space R^n where n is the size of SL. This is done by embedding each sample s in SL to n-coordinated vector, where the coordinate i is the DTW distance of the sample s to the shape s_i in SL. As a result, to search a shape s in the embedded space, we will have first to map S into R^n using the same process, which means, we will have to perform n calculations of the DTW distance to each sample in SL. Reducing the dimensionality of the space R^n to R^d where $d << n$ .i.e. d is sub linear to n, while preserving the $K-NN$ property in the reduced space, will enable embedding s and searching it in sub linear time. To search the approximate k-Nearest Neighbors efficiently in sub linear time, we use the Locality sensitive hashing (LSH) since it can manage higher dimensional spaces compared to $k-d$ trees.

3.1 Embedding to Euclidean Space

BoostMap [1] is a method for constructing embeddings that are optimized for preserving the similarity structure of the original space. Formally, let X denote a set of objects (Shapes of words in our case), and let the metric $D(x1, x2)$ be the DTW distance of the object $x1$ to $x2$ represented by their MAD feature vectors. A 1-D Euclidean embedding of space X is simply a function $F : X \to R$. Any given object $r \in X$ can define a 1-D embedding using the formula : $Fr(x) = DTW(x, r)$. The object r used to define Fr, is typically called a reference object. If D obeys the triangle inequality, Fr intuitively maps nearby points in X to nearby points on the real line R. Embedding The DTW metric using Fr, may violate the triangle inequality for some triples of objects but Fr may still map nearby points in X to nearby points in R, at least most of the time. On the other hand, distant objects may also map to nearby points.

An embedding $F : X \to R^d$, is a function that maps any object $x \in X$ into a d-dimensional vector $F(x) \in R^d$. Such embedding is meaningful, if it is distance preserving, i.e., it embeds close neighbors in X under D to close neighbor in the embedding space under some l_p metric. We are interested in constructing an embedding F that, given a query object q, can provide good approximate similarity rankings of objects to the query object q, in order of decreasing similarity. We follow the 'BoostMap' methodology [1], to construct the embedding F (strong classifier) as a linear combination of weighted 1-D embeddings (weak classifiers). The Distance between two objects in the embedded space will be measured using the weighted $L1$ metric, while the Adaboost learning methodology guaranties distance preserving embedding.

If X is a set of objects, and DTW(x, y) is the DTW distance measure between objects $x1, x2 \in X$. for a triple (q, x, y) of objects in X, we define the proximity order $P(q, x, y)$ to be a function that outputs whether q is closer to x or to y using DTW, see the following definition:

$$P(q,x,y) = \left\{ \begin{array}{rll} 1 & if & DTW(x,q) < DTW(y,q) \\ 0 & if & DTW(x,q) = DTW(y,q) \\ -1 & if & DTW(x,q) > DTW(y,q) \end{array} \right\}$$

In our application, the domain X is a set of images of handwritten words, and D is the DTW similarity distance between shapes in X. An embedding $F : X \to R^d$ is a function that maps any object $x \in X$ into a d-dimensional vector $F(x) \in R^d$, where distances between vectors in R^d are measured using a weighted Manhattan distance ($L1$) metric.

As in BoostMap, the simple (1-D) embeddings are used to construct the embedding F, where any object $r \in X$ can be used to define a one-dimensional embedding $Fr : X \to R$. Fr maps each object of X to a single real number which is it's distance to r, to be called a reference point. Mostly, if the objects, x and y are similar under the DTW measurement, then it is expected that their embeddings $Fr(X)$ and $Fr(Y)$ will be nearby points on the real line. Generally, each 1-D embedding Fr acts as a weak classifier for the following binary classification problem: given three objects $q, x, y \in X$, Fr provides an answer to the questions which object (x or y) is closer to q by simply checking if $Fr(q)$ is closer to $Fr(a)$ or to $Fr(b)$ using the l_1 metric. This classifier in many cases will probably have a high error rate, but still it is expected to be more accurate than a random guess.

For ranking nearest neighbors in a filter and refine manner we have decided to adopt only reference points to act as 1-D embeddings, but not to use pivot 1-D embeddings as presented in [1]. Therefore the training set S is a set of triples $< q, x, y >$, picked randomly as in BoostMamp but with some further considerations. In our approach, we use the Principal Component Analyses (PCA), and the Linear Discriminant Analyses (LDA) dimensionality reduction techniques, to guide for efficient generation of training sets, which in their turn will determine the right reference points for the embedding process. The algorithm constructs, using Adaboost, an embedding $F : X \rightarrow R^d$, optimized for classification accuracy on triples of objects where distances, are measured using weighted $L1$ metric.

The training algorithm for BoostMap as presented in [1], is an adaptation of the Adaboost algorithm to the problem of embedding construction. The inputs to the training algorithm are the following:

- A training set $T = (q_1, x_1, y_1), ..., (q_t, x_t, y_t))$ of t triples of objects from X and the set Y of labels from $-1, 1$ to each triple.
- A set $C \subset X$ of candidate reference objects to define 1-D embeddings.
- A matrix of mutual distances of all pairs of objects in X.

The training algorithm combines many classifiers associated with $1 - D$ embeddings Fr for $r \in C$, into a strong classifier H, where $H(x) = \sum_{i=1}^{d} \alpha_i Fr_i(x)$.

The methodology of Adaboost is used to determine the final set of 1-D embeddings, and their weights to generate the strong classifier H. In Adaboost methodology, weak classifiers are chosen and weighted so that they complement each other. Even when individual classifiers are highly inaccurate, the combined classifier can be accurate.

We have slightly modified the BoostMap method in order to guarantee a better learning and boosting process. we have modified the generation of the initial set C of reference point to be guided by (PCA) to insure capturing the main direction of the variance of the domain X and the (LDA), to support more discriminance power between the different words. Therefore, we have used the leading $2 * (logn)$ components of the PCA and LDA main direction for variance and separation respectively. Following the improvement in [2], we have generated the triples to support more accuracy in k-NN ranking than real similarity by using the initial class C to generate triples including reference point from C and many occurrences of objects from their k-Nearest Neighbors. After Embedding to the Euclidean space, we implement the known Locality Sensitivity Hashing (LSH) method for fast approximations of the k-nearest neighbors. The top k-ranked neighbors of a query object are retrieved and the original DTW measurement is calculated to each object in that list.

3.2 Fast k-Nearest Neighbor approximation

Computing exact nearest neighbors in high dimensional spaces, is a very difficult task. Few methods seem to be significantly better than a brute-force computation of all distances. However, it has been shown that by computing nearest neighbors approximately, it is possible to achieve significantly faster running times with a relatively small actual errors. Methods and structures for both exact and approximate nearest neighbor searching such as kd-tree and box decomposition trees can not be used due to the high dimensionality of the given feature space. In the presented approach, we use the Locality Sensitive Hashing (LSH) which manages high dimensional points and guaranties better performance.

Locality sensitive hashing, is a technique for grouping points in space into 'buckets' based on some distance metric operating on the points. Points that are close to each other under the chosen metric are mapped to the same bucket with high probability. This is based on the simple idea that, if two points are close together, then after a projection operation, these two points will remain close together. The basic idea is to hash the input items so that similar items are mapped to the same buckets with high probability (the number of buckets being much smaller than the universe of possible input items). LSH [5], uses several hash functions of the same type to create a hash value for each point of the dataset. Each function reduces the dimensionality of the data by projection onto random vectors. The data is then partitioned into bins by a uniform grid. Since the number of bins is still too high, a second hashing step is performed to obtain a smaller hash value. At query time, the query point is mapped using the hash functions and all the data points that are in the same bin as the query point are returned as candidates. The final nearest neighbors are selected by a linear search through candidate data points.

3.3 Feature Extraction

The Multi Angular Descriptor (MAD) for shape based object recognition presented by Saabni and Bronstein [15], works for Binary and gray level shapes, and with one and multi components shapes. In the binary case, from each contour point, the Angular descriptor captures the angular view to multi resolution rings in different heights. In the gray level case, it captures the weighted distribution over relative positions of the shape points to multi resolution rings around the centroid. The angular descriptor is robust to noise and small deformations and have very flexible variables which can be tuned for different applications. The extension of this descriptor to the gray level case can be seen as an extension of the shape context to gray level images which enables dealing with low quality images. Given a binary image I of a connected component(CC), We start by calculating the centroid C and the diameter D of the image I. The calculated values of C and D are used to determine and draw a set of rings centered by C with different radius values which are derived from the diameter D. We treat the rings as lying on different heights above the given shape where larger rings overlay closer to the shape. In the next step, we treat each ring as a set of k points taken uniformly distant from each other. Each point in each ring serves as an upper view point watching each pixel (contour point) in the shape. The main idea of the presented descriptor is to generate a sequential concatenation of upper view points from different heights and resolution to the 2-D shape. The multi resolution and heights will enable capturing more information in different resolutions and by that enabling a local and semi-global description of the given shape.

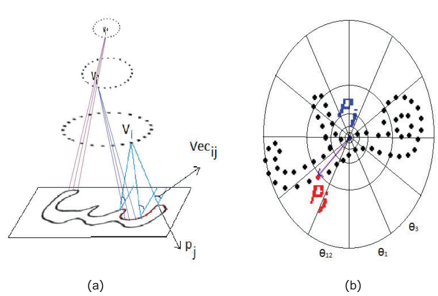

Figure 1. (a) The Multi Angular Descriptor with three view points from three different layers, and (b) the Shape Context Descriptor from one contour point view.

Let I be a binary image with the size nXm including one Connected Component(CC), and Let C and D be the centroid and the diameter of the shape respectively. Let $P = \{P_i\}_{i=1}^{l}$ a set of l point taken uniformly from the extracted contour of the CC. Given a view point V_j from a given ring with height h over the shape, the $l-coordinate$ vector of angles, obtained by connecting the point V_j with each point $P_i \in P$ and the plain of the shape is a rich description of the shape from this view point. As expected, one view point have some limitations, therefore, the key idea is to give additional view points from different directions, therefore, additional points of view will enable richer and more accurate description of the shape. Lifting the points of view up to different heights from the $2-D$ shape gives additional views to the shape, avoids intersection with segments of the shape, and integrates the distances into the angle value.

4 Experimental results

To evaluate the proposed approach we have used the IFN/ENIT off-line Arabic database [11], a set of Arabic words taken from a data base for Arabic handwritten text recognition research, and a set of multiple appearances of 500 words written by 51 students. from that collection, we have extracted 1632 words with $45,800$ different shapes. 10% of this set have been randomally taken out of the lexicon to be used as testing samples. To compare results, 1) we have used the basic DTW algorithm directly on the samples without preprocessing and 2) we have first pre-processed the dataset using the presented method and used (LSH) to extract the k-Top ranked list in a filter and refine approach. In this case we have picked K to be 10 closest neighbors from all shapes. The BoostMap process have used $d = O(\log(n))$ reference object for embedding to the R^d Euclidean space. Table 4, shows the accuracy rates and reduction in time.

Table 1. Results shows a 4 orders of magnitude over the exact method with only 2.2% lose in precesion rates.

Factor	$BasicDTW$	$BoostMap + LSH$
Precision	83.4%	81.2%
Time (msec)	173,528	84

As seen in Table 4, Our approximate method achieves a 4-time decrease in the search time at the expense of an insignificant drop in precision. The presented times are for an average query searching a given shape within the data set with an average home desktop computer.

5 Acknowledgments

We would like to thank Prof' D. Ron from Tel-Aviv University for her insightful comments which led to several improvements in the presentation of this paper. This work has been supported in part by the 'Vatat and Pikovsky Valazzi Post Doc scholarship' and the DFG-Trilateral Grant No.8716.

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *in IEEE Conference on Computer Vision and Pattern Recognition*, pages 268–275, 2004.

[2] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: An embedding method for efficient nearest neighbor retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(1):89 –104, jan. 2008.

[3] C. Faloutsos and K. Lin. fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings ACM SIGMOD Conference*, pages 163–174, 1995.

[4] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis, and S. Perantonis. A segmentation-free approach for keyword search in historical typewritten documents. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, volume 1, pages 54–58, 29 Aug.-1 Sept. 2005.

[5] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *In Proceedings of the 25th Intl Conf on Very Large Data Bases*, 1999.

[6] K. Grauman and T. Darrell. Fast contour matching using approximate earth movers distance. In *In IEEE Conference on CVPR*, volume 01, page 220227, 2004.

[7] G. Hjaltason and H. Samet. .properties of embedding methods for similarity searching in metric spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:530–549, 2003.

[8] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *In 3rd International Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[9] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Math*, 26:189–206, 1984.

[10] R. Manmatha and T. Rath. Indexing handwritten historical documents - recent progress. *the Proc. of the Symposium on Document Image Understanding (SDIUT-03)*, pages 77–85, 2003.

[11] M. Pechwitz, S. S. Maddouri, V. Mrgner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. In *In Proc. of CIFED 2002*, pages 129–136, 2002.

[12] T. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, volume 1, pages 218–222, 3-6 Aug. 2003.

[13] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 521–527, 18-20 June 2003.

[14] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science 22 December*, 290:2323–2326, 2000.

[15] R. Saabni and A. Bronstein. The multi angular descriptor: new binary and gray images descriptor for shape recognition. Technical report, Triangle Research & Development Center, 2011.

[16] R. Saabni and J. El-Sana. Keyword searching for arabic handwritten documents. In *The 11'th International Conference on Frontiers in Handwriting recognition (ICFHR2008), Montreal*, pages 716–722, 2008.

[17] R. Saabni and J. El-Sana. Word spotting for handwritten documents using chamfer distance and dynamic time warping. In *Document Recognition and Retrieval XVIII*, 2011.

[18] S. Shirdhonkar and D.W. Jacobs. Approximate earth movers distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference.*, pages 1–8, 2008.

[19] S. Srihari, C. Huang, and H. Srinivasan. A search engine for handwritten documents. *Document Recognition and Retrieval XII, San Jose, CA,Society of Photo Instrumentation Engineers (SPIE)*, pages pp. 66–75, January 2005.

[20] J.T.L. Wang, K.I. Lin X. Wang, D. Shasha, B.A. Shapiro, and K. Zhang. Evaluating a class of distance-mapping algorithms for data mining and clustering. *Proc. ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining*, pages 307–311, 1999.

[21] X. Wang, J. T. L. Wang, K. I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang. An index structure for data mining and clustering. *Knowledge and Information Systems*, vol. 2:161–184, 2000.

[22] F. Young and R. Hamer. Multidimensional scaling: History, theory and applications. *Hillsdale, New Jersey: Lawrence Erlbaum Associates*, 1987.