Locality Sensitive Hashing for Fast Computation of Correlational Manifold Learning based Feature space Transformations

Vikrant Singh Tomar, Richard C. Rose

Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

vikrant.tomar@mail.mcgill.ca, rose@ece.mcgill.ca

Abstract

Manifold learning based techniques have been found to be useful for feature space transformations and semi-supervised learning in speech processing. However, the immense computational requirements in building neighborhood graphs have hindered the application of these techniques to large speech corpora. This paper presents an approach for fast computation of neighborhood graphs in the context of manifold learning. The approach, known as locality sensitive hashing (LSH), has been applied to a discriminative manifold learning based feature space transformation technique that utilizes a cosine-correlation based distance measure. Performance is evaluated first in terms computational savings at a given level of ASR performance. The results demonstrate that LSH provides a factor of 9 reduction in the computational complexity with minimal impact on speech recognition performance. A study is also performed comparing the efficiency of the LSH algorithm presented here and other LSH approaches in identifying nearest neighbors.

Index Terms: Locality sensitive hashing, correlation preserving discriminant analysis, discriminative manifold learning

1. Introduction

Manifold learning based feature space transformations assume that data points reside on or close to the surface of a lower dimensional manifold. The techniques attempt to capture the underlying manifold based relationships among data vectors in order to find a target feature representation, where the underlying relationships between feature vectors are preserved [1–3]. It has been suggested that the acoustic feature space is confined to lie on one or more low dimensional manifolds [4, 5]. Therefore, a feature space transformation technique that explicitly models and preserves the local relationships of data along the underlying manifold should be more effective for speech processing.

Multiple studies have demonstrated gains in automatic speech recognition (ASR) performance when using features derived from a manifold learning approach. Tang et. al. reported gains in ASR performance using features derived from locality preserving projections (LPP) [6]. In previous work [7, 8], the authors presented discriminative manifold learning techniques that led to significant improvements in ASR word error rates (WER) as compared to well-known techniques such as linear discriminant analysis (LDA) [9, 10] and LPP [1, 6]. However, despite having shown significant improvements in ASR performance on some tasks, manifold learning based algorithms have yet to find widespread usage in speech processing. This lack of acceptance can be credited to the high computational complexity and noise sensitivity of these algorithms [1, 3, 6, 11, 12].

This work is supported by Natural Sciences and Engineering Research Council of Canada, and McGill University.

The computational complexity of manifold learning techniques originates from the need to construct nearest neighborhood based relationships. Typically, a pair-wise distances measure is used. For a dataset containing N feature vectors of d dimensionality each, the construction of nearest neighborhood based graphs would require computational time amounting to $O(dN^2)$. Speech datasets typically have hundreds of millions of feature vectors each having dimensions in the range of 100-200. For these datasets, using an algorithm with $O(dN^2)$ can be computationally infeasible. Though, there exist a number of algorithms that allow for faster neighborhood calculations such as kd-trees, many of these algorithms reach the complexity of linear search as the dimensionality of data increases [13].

This work investigates a fast algorithm for neighborhood calculations, locality sensitive hashing (LSH) [14–16], as applied to manifold learning based correlation preserving discriminant analysis (CPDA) in ASR [8]. The algorithm acts by creating hashed signatures of feature vectors for distributing vectors into a number of discrete buckets. The underlying concept is that vectors with strong correlation are more likely to fall into the same bucket. It is shown that LSH can drastically reduce the computational complexity of manifold learning algorithms. In this work, LSH is shown to provide a factor of 10 speedup without significant impact on the ASR performance.

LSH is incorporated within the CPDA framework for fast computation of neighborhood graphs. CPDA is a supervised discriminative manifold learning algorithm that attempts to preserve the underlying local sub-manifold based relationships of feature vectors while at the same time tries to maximize a criterion related to the separability between classes of vectors. The algorithm utilizes a cosine-correlation based distance measure instead of the conventional Euclidean measures. The use of a cosine-correlation based measure is motivated by studies suggesting that additive noise in linear spectrum domain alters the norm of cepstrum feature vectors [17], and that the angles between cepstrum vectors are comparatively more robust to noise [18]. Thus, the techniques that use a correlation based distance measure for characterizing the relationships between features are less susceptible to ambient noise compared to techniques that use an Euclidean measure. Accordingly, CPDA has demonstrated significantly improved ASR performance in noisy environments [8]. Correlation preservation based techniques have also been used in other application domains [11, 19].

2. Correlation Preserving Discriminant Analysis

This section summarizes the CPDA algorithm. A more detailed discussion can be found in [8]. CPDA is a discriminative manifold learning technique that attempts to maximize class separa-

bility while preserving local sub-manifold based relationships of the data vectors. An important aspect of CPDA is that it measures the relationships between feature vectors in terms of a cosine correlation based kernel.

The goal in CPDA is to estimate the parameters of a projection matrix $\boldsymbol{P} \in \mathbb{R}^{d\times m}$, with $m \leq d$ (generally $m \ll d$), which maximizes the class discrimination in the projected feature space while retaining the inherent data structure. The first step is to normalize the training feature vectors. This discards the magnitude information while retaining information related to the cosine correlation among feature vectors. Consider a normalized ASR training dataset $\boldsymbol{X} = [\boldsymbol{x}_1^T, \cdots \boldsymbol{x}_N^T]^T$, where each $\boldsymbol{x}_i \in \mathbb{R}^d$ represents a vector projected onto the surface of a d-dimensional unit hypersphere, and belongs to the class/label $C(\boldsymbol{x}_i)$. For an arbitrary source space vector $\boldsymbol{x}_i$, the corresponding target space vector $f(\boldsymbol{x_i}, \boldsymbol{P})$ is obtained as,

$$f(\boldsymbol{x}_i, \boldsymbol{P}) = \frac{\boldsymbol{P}^T\boldsymbol{x}_i}{||\boldsymbol{P}^T\boldsymbol{x}_i||} \qquad \forall i = 1, 2, \cdots, N. \tag{1}$$

where $f(\boldsymbol{x}_i, \boldsymbol{P}) = 0$ for $||\boldsymbol{P}^T\boldsymbol{x}_i|| = 0$.

CPDA follows the graph-embedding framework for characterizing the relationship between feature vectors [3]. The training data is embedded into two undirected weighted graphs, the intrinsic graph $\mathcal{G}_{int} = \{\boldsymbol{X}, \boldsymbol{W}_{int}\}$, and the penalty graph $\mathcal{G}_{pen} = \{\boldsymbol{X}, \boldsymbol{W}_{pen}\}$. Here, $\boldsymbol{X}$, the set of training vectors, corresponds to the nodes of the graphs. The matrices $\boldsymbol{W}_{int}$ and $\boldsymbol{W}_{pen} \in \mathbb{R}^{N\times N}$ are the intrinsic and penalty correlation-affinity edge-weight matrices, respectively. The affinity matrices characterize the statistical and geometrical similarities of the feature vectors. The elements of the two affinity matrices are defined in terms of a cosine-correlation kernel as,

$$w_{ij}^{int} = \begin{cases} exp\left(-\frac{1-\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle}{\rho}\right) & ; C(\boldsymbol{x}_i) = C(\boldsymbol{x}_j), e(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ 0 & ; \text{Otherwise} \end{cases} \tag{2}$$

$$w_{ij}^{pen} = \begin{cases} exp\left(-\frac{1-\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle}{\rho}\right) & ; C(\boldsymbol{x}_i) \neq C(\boldsymbol{x}_j), e(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ 0 & ; \text{Otherwise} \end{cases} \tag{3}$$

where ρ is the kernel scale parameter and $C(\boldsymbol{x}_i)$ refers to the class or label of vector $\boldsymbol{x}_i$. The indicator function $e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is equal to 1 when $\boldsymbol{x}_i$ lies in the near neighborhood of $\boldsymbol{x}_j$ in correlation sense. The LSH algorithm described in Section 3 is utilized for fast computation of nearest neighbors. In the intrinsic graph, $\mathcal{G}_{int}$, a node $\boldsymbol{x}_i$ is connected to the K_{int} nearest neighbors belonging to the same class $C(\boldsymbol{x}_i)$. Similarly, in the penalty graph, $\mathcal{G}_{pen}$, a node $\boldsymbol{x}_i$ is connected to the K_{pen} largest affinity neighbors *not* belonging to the class $C(\boldsymbol{x}_i)$.

In the transformed space, a correlation measure for a graph $\mathcal{G}$ is given by,

$$\begin{aligned} S &= \sum_{i\neq j\in\mathcal{G}} (1 - \langle f(\boldsymbol{x}_i, \boldsymbol{P}), f(\boldsymbol{x}_j, \boldsymbol{P})\rangle) w_{ij}, \\ &= \sum_{i\neq j\in\mathcal{G}} \left(1 - \frac{f_{ij}}{f_i f_j}\right) w_{ij}, \end{aligned} \tag{4}$$

where for two arbitrary vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $f_i = \sqrt{\boldsymbol{x}_i^T\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{x}_i}$, and $f_{ij} = \boldsymbol{x}_i^T\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{x}_j$.

The goal is to penalize the properties inherent to the penalty graph while at the same time preserve the properties inherent to the intrinsic graph. Thus, the following function is defined as a measure of class separability and graph-preservation [19],

$$F(\boldsymbol{P}) \;=\; S_p - S_i \;=\; \sum_{i\neq j} \left(1 - \frac{f_{ij}}{f_i f_j}\right) \cdot w_{ij}^{p-i} \tag{5}$$

where $w_{ij}^{p-i} = w_{ij}^{pen} - w_{ij}^{int}$. An optimal projection matrix is the one to maximize the above function, *i.e.*,

$$\boldsymbol{P}_{CPDA} = \arg\max_{\boldsymbol{P}} F(\boldsymbol{P}). \tag{6}$$

To optimize expression (5), the gradient ascent rule can be utilized as follows: $P \leftarrow P + \alpha\nabla_P F$, with

$$\nabla_P F = \sum_{i\neq j} \left[\frac{f_{ij}\boldsymbol{x}_i\boldsymbol{x}_i^T}{f_i^3 f_j} + \frac{f_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^T}{f_i f_j^3} - \frac{\boldsymbol{x}_i\boldsymbol{x}_j^T + \boldsymbol{x}_j\boldsymbol{x}_i^T}{f_i f_j}\right] \boldsymbol{P} \cdot w_{ij}^{p-i} \tag{7}$$

where α is the gradient scaling factor, and $\nabla_P F$ represents the gradient of the expression (5) with respect to $\boldsymbol{P}$.

It is important use a good initialization for the gradient ascent in order to avoid converging to a local optima [8]. To this end, the mapping function can be approximated to that of a linear transformation rooted in graph-embedding by neglecting the normalization term, *i.e.*, setting $f(\boldsymbol{x}_i, \boldsymbol{P}) = \boldsymbol{P}^T\boldsymbol{x}_i$. A closed-form solution for a good initial projection matrix is then achieved by solving the eigenvalue problem [3, 7]

$$(\boldsymbol{X}(\boldsymbol{D}_p - \boldsymbol{W}_p)\boldsymbol{X}^T)\boldsymbol{p}_j = \lambda_j(\boldsymbol{X}(\boldsymbol{D}_i - \boldsymbol{W}_i)\boldsymbol{X}^T)\boldsymbol{p}_j, \tag{8}$$

where $\boldsymbol{D}$ is a diagonal matrix whose elements correspond to the row sum of the matrix $\boldsymbol{W}$. The subscript i and p signifies 'intrinsic' and 'penalty' matrices respectively. The vector $\boldsymbol{p}_j$ indicates the j^{th} column of the transformation matrix $\boldsymbol{P}_{CPDA}$.

3. Locality Sensitive Hashing

LSH is a class of algorithms that promise fast nearest neighborhood search in a high dimensional space with a high degree of accuracy. A number of different implementations exist for these schemes [20]. This work utilizes a simple extension of a particular LSH scheme referred to as exact Euclidean LSH (E2LSH) that attempts to find the nearest neighbors in the Euclidean space using random projections [15]. Section 3.1 presents a general description of the E2LSH algorithm. Some details regarding LSH incorporation within the CPDA framework are given in Section 3.2. Though meant for a Euclidean space, in this work, E2LSH is used for finding nearest neighbors in a cosine-correlation based unit hyperspace; a discussion on the validity of this implementation is included in Section 3.3. Further discussion on LSH can be found in [14–16, 21].

3.1. Exact Euclidean LSH (E2LSH)

E2LSH attempts to hash given feature vectors to various buckets on the real line in a "locality sensitive" manner, *i.e.*, with a goal that features close to each other in the original space will fall into the same buckets. This is achieved by projecting each vector $\boldsymbol{x}_i$ onto real line by a family of hash functions $\mathcal{H} = \{h : \mathbb{R}^d \rightarrow \mathbb{N}\}$. The hash function used in this work is given by,

$$h(\boldsymbol{x}) = \left\lfloor \frac{<\bar{\boldsymbol{a}}, \boldsymbol{x}> + \bar{b}}{w} \right\rfloor, \tag{9}$$

where $\bar{\boldsymbol{a}}$ is a d-dimensional random vector whose entries are chosen from a p-stable distribution, w is the width of each segment or bucket on the real-line, and the bias, $\bar{b}$, is a uniform random number taken from $[0, w]$. The projection of all the vectors in this manner results in a chain or table of hash buckets, each having pointers to one or more vectors.

The hash function given in Eq. (9) is locality sensitive in the Euclidean space because of the following property of p-stable distributions. If $\bar{a}_i$ for $i \in \{1 \ldots M\}$ are independently

and identically distributed random variables that follow a p-stable distribution, then their p^{th} -order-rooted linear combination, $(\overline{a}_1^p + \overline{a}_2^p + \dots)^{1/p}$, also follows the same distribution. For example, a Gaussian distribution is a 2-stable distribution. Thus, it can be inferred that for two arbitrary feature vectors, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and a random vector $\overline{\boldsymbol{a}}$ whose elements have been independently sourced from a 2-stable distribution, the distance between their inner-products, $< \bar{\boldsymbol{a}}, \boldsymbol{x_i} >$ and $< \bar{\boldsymbol{a}}, \boldsymbol{x_j} >$, is distributed as $||\boldsymbol{x}_i - \boldsymbol{x}_j||_2 \bar{\boldsymbol{z}}$ where $\bar{\boldsymbol{z}}$ is a random variable that follows the same 2-stable distribution as the elements of $\overline{\boldsymbol{a}}$ [15, 22]. This property *guarantees* that if two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close together in the original space then they should have high probability of collision or hashing to the same bucket: $Pr[h(\boldsymbol{x}_i) = h(\boldsymbol{x}_j)] \geq P_1$. If the two points are far apart then the collision probability should be small: $Pr[h(\boldsymbol{x}_i) = h(\boldsymbol{x}_j)] \leq P_2$, where $P_1/P_2 > 1$.

For optimal performance, k-different projections are used to create a family of composite hash functions $\mathbb{G} = \{g : \mathbb{R}^d \rightarrow \mathbb{N}^k\}$ such that $g(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \dots, h_k(\boldsymbol{x})]$, where $h_i(\boldsymbol{x}) \in \mathcal{H}$. This increases the hashing discriminative power as $(P_1/P_2)^k > P_1/P_2$. Furthermore, L independent hash tables are created for which hash functions, $g_1, \dots, g_L$, are uniformly chosen from $\mathbb{G}$. By choosing the optimal values of w, k and L, one can find the true neighbors with an arbitrarily high probability.

In summary, as a result of performing LSH, each feature vector, $\boldsymbol{x}_i$, is associated with one of the buckets in each of the L tables. The number of buckets in different tables can be different. Ideally, two points that are close to each other should fall in the same buckets in all the tables. Given the hash tables, the nearest neighbors are identified for a new data vector (query) by hashing the vectors into a bucket in each table and forming the union of all points in these buckets.

3.2. Implementation Details

The LSH algorithms are ideally targeted at applications with a small query set that is typically separate from the training set. In such cases, the nearest neighbors are found by iterating over the query points to identify the target hashing bucket and candidate neighbors of each. However, often in manifold learning, it is required to search for nearest neighbors for all the query points in the training set itself. In such cases, it is not feasible to *re*-iterate over the entire training set. For example, for the CPDA algorithm described in Section 2, the nearest neighbors were calculated for all 1.4 million feature vectors from the training set in order to populate the affinity matrices $\boldsymbol{W}_{int}$ and $\boldsymbol{W}_{pen}$.

This issue can be avoided by calculating the pair-wise distances between all hashed vectors within a bucket to create multiple candidate neighborhood structures for each vector. The final neighbors for each vector can be selected from a union of all such structures.

3.3. The choice of LSH scheme

The CPDA algorithm is based on finding nearest neighbors in a unit hyperspace, where affinity between feature vectors is measured using an exponentially decaying cosine correlation kernel, *i.e.*, $w_{ij} = exp\left(-\frac{1-\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{\rho}\right)$. For two unit vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, it is trivial to see that, for $p = 2$, the p-stable property of E2LSH is also valid in this desired kernel space:

$$\begin{aligned} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 &= \boldsymbol{x}_i^T \boldsymbol{x}_i + \boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{x}_i^T \boldsymbol{x}_j \\ &= 1 + 1 - 2\boldsymbol{x}_i^T \boldsymbol{x}_j \\ &= 2(1 - \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle). \end{aligned}$$

Thus, extending the p-stability based arguments of E2LSH, it can be said that for two unit vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, their projection by a vector with elements chosen from a p-stable distribution will vary as $(1-\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle)\bar{\boldsymbol{z}}$. In other words, the E2LSH hashing scheme is locally sensitive in the correlation space.

Authors in [23, 24] have used a different LSH scheme for correlation based neighborhood search. This involves hashing each vector, $\boldsymbol{x}_i$, using a k-dimensional bit representation, $\{0, 1\}^k$, by projection with a unit normal random vector $\boldsymbol{r} \in \{\boldsymbol{r}_1 \dots \boldsymbol{r}_k\}$,

$$h(\boldsymbol{x}_i) = \begin{cases} 1 & ; \quad \langle \boldsymbol{r}, \boldsymbol{x}_i \rangle \geq 0 \\ 0 & ; \quad \langle \boldsymbol{r}, \boldsymbol{x}_i \rangle < 0 \end{cases} . \tag{10}$$

The cosine distance is then approximated by Hamming distance in the bit vector space,

$$cos(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx cos(\frac{H(h(\boldsymbol{x}_i), h(\boldsymbol{x}_j))\pi}{k}), \tag{11}$$

where $H(\cdot, \cdot)$ denotes Hamming distance between two bit-vectors, and k refers to the number of random projections per vector, *i.e.*, the dimensionality of the resultant bit vectors.

The above cosine LSH scheme, henceforth referred to as cos-LSH, suffers from many limitations. The scheme requires on the order of hundreds independent tables for acceptable accuracy [23, 24]. Furthermore, the approximation of Hamming distance between bit vectors to that of cosine distance between the original feature vectors only holds when high dimensional bit representations ($k \approx 1000$) are used [21, 24]. The approximation approaches equality when k goes to infinity. Though finding Hamming distance between two bit vectors is a fast and memory efficient task, the advantages diminish as the dimensionality of the bit vectors increases. The high dimensionality of target bit representations also increases the cost of the random projections. E2LSH, in comparison, is more well-behaved. The scheme is supported by the property of p-stable distributions, and provides a high degree of accuracy with an easy to understand dependence on the parameters [15,25]. The optimal range of dimensions of hash functions varies between 1 and 5, and the number of tables is less than 10. These claims are supported by the results presented in Section 4.4.

4. Experimental Study and Discussion

This section describes the experiments performed to evaluate the effectiveness of LSH for building neighborhood graphs when incorporated within discriminative manifold learning based CPDA as applied to ASR feature space transformations. The effectiveness is measured in terms of reduction in time required to train the CPDA projection matrix, $\boldsymbol{P}$, and ASR word error rate (WER) obtained using the transformed features. A comparison of E2LSH and cos-LSH [21, 23] in terms of their ability to find the true nearest neighbors is given in Section 3.3.

4.1. Task Domain and Setup

The experimental evaluations are performed on the Aurora-2 speech in noise task. The training set contains a total of 8440 utterances recorded from 55 male and 55 female speakers and artificially noise corrupted by a mixture of noise conditions. The corpus represents a simulation of a speech in noise task, and one must be careful when generalizing these results to other tasks.

The ASR system is configured using whole word continuous density hidden Markov models (CDHMMs). There are 16 states per word-model, and 4 states for silence models for a total

of 180 states. Features used for the baseline condition are composed of 13-dimensional Mel-frequency cepstrum coefficients (MFCCs) augmented with log energy and first and second order differences.

The feature space transformations are trained on 9 concatenated MFCC frames. The states of CDHMMs are used as class labels for the supervised transforms. A neighborhood size of $K_{int} = K_{pen} = 200$ is used for the calculations of affinity matrices, $\boldsymbol{W}_{int}$ and $\boldsymbol{W}_{pen}$ in CPDA. Systems labeled as CPDA-LSH use the cosine E2LSH for building the nearest neighborhood graphs. The resultant projection matrix $\boldsymbol{P}$ is then used to project the 117-dimensional training and test vectors to a 39 dimensional space. Semi-tied covariance (STC) transformations are applied prior to recognition to account for the correlation introduced to the transformed features by the CPDA and LDA projections, as described in [8, 26].

4.2. Computational Analysis

LSH reduces the computational requirements for neighborhood search from $O(dN^2)$ to $O(dkNL) + O(dN_B^2)$, where d is the dimensionality of feature vectors, N is total number of feature vectors, N_B is average number of points in each bucket, k is dimension of hash functions $g(\boldsymbol{x})$, and L is number of hash tables. This is a significant improvement. Typically, N_B is several orders of magnitude smaller than N; for example, N_B had a value in the range of 300 compared to $N = 1.4$ million for $w = 1$ and $k = 3$ in this work.

The ASR results presented in the next section are obtained using a training set containing 1.4 million vectors of dimensionality 117. The execution time for different feature space transformation techniques on the same computing system is: LDA– 90 seconds, LPP – 28 hours, CPDA without LSH – 36 hours, and CPDA-LSH – 4 hours (for $k = 3, L = 6, w = 1$), respectively. Thus, LSH provides a factor of 9 speed up to the CPDA algorithm. This remarkable speedup should enable the application of manifold learning based feature space transformation techniques to generally large speech databases.

4.3. ASR Results

ASR WER results for the Aurora-2 speech in noise task are given in Table 1. The test results are given for clean testing, and noise levels ranging from 20dB to 5dB SNR. At each SNR level, the results are averaged over four different noise types (subway, car, exhibition, and airport). Each row of the table presents ASR WER results obtained using a particular feature type. The last row, labeled "CPDA-LSH" refers to ASR results when CPDA transformation is obtained while using the fast LSH scheme for nearest neighbors calculations.

The most interesting observation from Table 1 is that CPDA-LSH shows almost no impact on ASR performance as compared to CPDA without LSH for high SNR cases. Though the performance of the randomized LSH algorithms seems to be affected by the presence of high noise, the ASR performance is still superior to other techniques like LDA, and LPP. Especially, comparisons with LPP – which is an unsupervised manifold learning technique – show the discriminative cosine manifold learning based CPDA with LSH solves the problem of high computational complexity as well as noise corruption.

4.4. Comparison of LSH schemes

The arguments presented in Section 3.3 in favor of E2LSH over cos-LSH are supported by the performance comparisons given in Table 2. The table compares the two LSH schemes for their ability to provide computational performance gains and accuracy of finding true nearest neighbors by varying the number of random projections k. The speedup and %-accuracy are given with respect to the linear neighborhood search. For each hashing scheme, optimal value of other parameters – w and L for E2LSH, and the number of permutations, L, and beam width, B – are empirically chosen. Further details about the cos-LSH scheme and related parameters can be found in [21, 23, 24].

For this experiment 100,000 feature vectors are randomly selected from the mixed-condition noisy training set described in Section 4.1. Two separate tables are show, one for each LSH scheme. Note that both schemes operate in different parameter spaces, therefore a fair comparison can only be drawn with respect to the trade-off between the gained speedup and accuracy of finding true neighbors. It is evident from the comparisons 2 that E2LSH provides much better search accuracy for the same amount of speedup. Similar comparisons can also be drawn by varying the number of tables or permutations L. In the experiments conducted in this work, it is found that increasing L does not lead to significant accuracy gains after certain point. However, the speedup decreases with increases in L. These results are not reported due to lack of space.

Table 1: WER for mixed noise training and noisy testing on Aurora-2 speech corpus.

Features	SNR (dB)				
	Clean	20	15	10	5
MFCC	1.86	3.07	4.62	7.22	14.05
LDA	1.82	2.71	3.54	6.40	15.36
LPP	1.77	2.60	4.06	7.56	16.79
CPDA	1.53	2.43	3.31	5.64	14.11
CPDA-LSH	1.39	2.34	3.51	5.95	15.06

Table 2: Comparison of E2LSH and cos-LSH schemes for their ability of finding true nearest neighbors.

E2LSH ($w = 1, L = 6$)			cos-LSH ($B = 32, L = 50$)		
k	% Acc.	Speedup	k	% Acc.	Speedup
3	98.3	8.0	5000	65.9	2.1
4	86.9	9.7	1000	45.6	7.1
5	85.1	15.6	500	23.7	10.2
6	57.1	19.3	100	17.6	13.8
7	49.9	27.9	3	2.0	16.4

5. Conclusion

This paper has presented an investigation of utilizing fast LSH algorithms for populating the affinity matrices in manifold learning based feature space transformations. The discriminative manifold learning based CPDA algorithm is used as an example manifold learning technique. CPDA utilizes a cosine-correlation based distance measure to characterize the manifold domain relationships among feature vectors. For this reason, a cosine adaptation of exact Euclidean LSH (E2LSH) scheme is chosen for hashing. It is demonstrated that use of LSH within CPDA framework leads to significant computational gains without much affect on ASR performance. Furthermore, the chosen LSH scheme is compared to another commonly used cosine distance based LSH scheme in terms of the trade-off between computational gains and nearest neighbor search accuracy.

6. References

[1] Xiaofei He and Partha Niyogi, "Locality preserving projections," in *Neural Information Processing Systems (NIPS)*, 2002.

[2] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, 2003.

[3] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin, "Graph embedding and extensions: a general framework for dimensionality reduction.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[4] K. N. Stevens, *Acoustic Phoenetics*, MIT Press, Cambridge, MA, USA, 1998.

[5] Aren Jansen and Partha Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," in *ICASSP: IEEE International Conference on Acoustics Speech and Signal Processing*, 2006.

[6] Yun Tang and Richard Rose, "A study of using locality preserving projections for feature extraction in speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, NV, Mar. 2008, pp. 1569–1572, IEEE.

[7] Vikrant Singh Tomar and Richard C. Rose, "Application of a locality preserving discriminant analysis approach to ASR," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, Montreal, QC, Canada, July 2012, pp. 103–107, IEEE.

[8] Vikrant Singh Tomar and Richard C Rose, "A correlational discriminant approach to feature extraction for robust speech recognition," in *Interspeech*, Portland, OR, USA, 2012.

[9] K. Beulen, L. Welling, and H. Ney, "Experiments with linear feature extraction in speech recognition," in *European Conference on Speech Communication and Technology*, 1995.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2nd edition, 2000.

[11] Yong Ma, Shihong Lao, Erina Takikawa, and Masato Kawade, "Discriminant analysis in correlation similarity measure space," *Proceedings of the 24th international conference on Machine learning - ICML '07*, , no. 1, pp. 577–584, 2007.

[12] Vikrant Singh Tomar and Richard C. Rose, "Noise aware manifold learning for robust speech recognition," in *ICASSP: IEEE International Conference on Acoustics Speech and Signal Processing*, 2013.

[13] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 1st edition, 2006.

[14] Piotr Indyk and R Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth Annual ACM Symposium on Theory of Computing*, 1998, pp. 604–613.

[15] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," *Proceedings of the twentieth annual symposium on Computational geometry - SCG '04*, p. 253, 2004.

[16] Alexandr Andoni, Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press, 2006.

[17] David Mansour and B.H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *Speech and Signal Processing, IEEE*, vol. 37, no. 11, pp. 4–7, 1989.

[18] B.a. Carlson and M.a. Clements, "Speech recognition in noise using a projection-based likelihood measure for mixture density HMM's," *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 237–240 vol.1, 1992.

[19] Yun Fu, Ming Liu, and Thomas S. Huang, "Conformal embedding analysis with local graph modeling on the unit hypersphere," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, June 2007.

[20] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348–1358, Aug. 2010.

[21] Moses S. Charikar, "Similarity estimation techniques from rounding algorithms," *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC '02*, p. 380, 2002.

[22] V. M. Zolotarev, "One-Dimensional Stable Distributions," in *Vol. 65 of Translations of Mathematical Monographs*. American Mathematical Society, 1986.

[23] Aren Jansen and Benjamin Van Durme, "Efficient spoken term discovery using randomized algorithms," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, Dec. 2011, pp. 401–406.

[24] Deepak Ravichandran, Patrick Pantel, Eduard Hovy, Marina Rey, and I S I Edu, "Randomized Algorithms and NLP : Using Locality Sensitive Hash Functions for High Speed Noun Clustering," .

[25] Sariel Har-Peled, P Indyk, and R Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," *Theory of Computing*, vol. 8, no. 1, pp. 321–350, 2012.

[26] M. J. F. Gales, "Adapting Semitied Full Covariance Matrix HMMs," Tech. Rep., Cambridge University, UK, 1997.