Distance Metric Learning for Kernel Density-Based Acoustic Model Under Limited Training Data Conditions

Van Hai Do*†, Xiong Xiao†, Eng Siong Chng*† and Haizhou Li*†‡
* School of Computer Engineering, Nanyang Technological University, Singapore
† Temasek Laboratories@NTU, Nanyang Technological University, Singapore
‡ Institute for Infocomm Research, A*STAR, Singapore
E-mail: {dovanhai, xiaoxiong, aseschng}@ntu.edu.sg, hli@i2r.a-star.edu.sg

***Abstract*—Kernel density model works well for limited training data in acoustic modeling. In this paper, we improve the kernel density-based acoustic model for low resource language speech recognition. In our previous study, we demonstrated the effectiveness of the kernel density-based acoustic model on discriminative features such as cross-lingual bottleneck features. In this paper, we propose to learn a Mahalanobis-based distance, which is equivalent to a full rank linear feature transformation, to minimize training data frame classification error. Experimental results on the Wall Street Journal (WSJ) task show that the proposed Mahalanobis-based distance learning results in significant improvements over the Euclidean distance. The kernel density acoustic model with the Mahalanobis-based distance also outperforms deep neural network acoustic model significantly in limited training data cases.**

I. Introduction

Among the thousands of spoken languages used today, few of them are studied by the speech recognition community [1]. A major difficulty of ASR system deployment in a new language is that ASR systems rely on a large amount of training data for acoustic modeling. This requirement makes a full fledged acoustic modeling process impractical for under-resourced languages. Popular approaches are to transfer well-trained acoustic models to under-resourced languages such as universal phone set [2], cross-lingual tandem approach [3], cross-lingual subspace GMMs (SGMMs) [4], KL-HMM [5], [6], cross-lingual phone mapping [7], [8] and its extension, context-dependent phone mapping [9], [10], [11].

In our recent study [12], the kernel density model [13] - a special case of the exemplar-based approach [14] was applied for acoustic modeling for under-resourced languages. Unlike the parametric models, the kernel density model is a non-parametric technique that uses the training samples directly without estimating model parameters. This allows us to make full use of the limited training data. In [12], we replaced the GMM in the HMM/GMM model by the kernel density model to estimate the tied-state likelihood scores. Our study showed that by using cross-lingual bottleneck features, the kernel density model consistently outperforms the HMM/GMM and even the HMM/DNN models when the training data for target language is less than 4 hours. However, we found that using the kernel density with conventional MFCC feature yields worse results than the HMM/GMM model. The reason could be that the simple Euclidean distance used in the kernel density model is not optimal for speech recognition tasks. For example, the Euclidean distance treats the feature dimensions equally important and does not consider the correlation between feature dimensions. This prompts us to search for a better distance metric for the kernel density models.

In the exemplar-based learning literature, it is a popular approach to learn a distance metric from training data for a specific application, and this is called metric learning [19]. For example, the large margin nearest neighbors (LMNN) algorithm [21] is a supervised approach to learn a Mahalanobis-based distance metric. LMNN seeks a linear feature transformation such that, in the transformed space, the k nearest exemplars from the correct class and exemplars from other classes become separated by a larger margin. Another metric learning technique is the locality preserving projections (LPP) algorithm [20]. LLP learns a linear transformation $\boldsymbol{Q} : \mathcal{R}^d \mapsto \mathcal{R}^p$ where $(p \leq d)$ that aims to preserve the neighborhood structure of the data.

In this paper, we apply distance metric learning to kernel density-based acoustic model and learn a Mahalanobis-based distance metric that is optimized for speech recognition. The Mahanalobis-based distance learning is converted to a linear feature transformation problem. The feature transform is learnt in an iterative manner to optimize the frame classification accuracy on the training set using the maximum mutual information criterion (MMI).

The rest of this paper is organized as follows: Section II presents the kernel density acoustic model. Section III describes our proposed distance metric learning method. Section IV presents the experimental setups and results. Finally, we conclude in Section V.

II. Kernel density model for acoustic modeling

In this study, instead of using a GMM to model the feature distribution of a triphone tied-state as in the conventional HMM/GMM acoustic model, we use the kernel density model similar to the one used in [13], [12]. Specifically, the likelihood of feature vector $\boldsymbol{o}_t$ for speech class i.e. HMM tied-state s_j, is estimated as follows:

$$p(\boldsymbol{o}_t|s_j) \quad = \quad \frac{1}{ZN_j}\sum_{i=1}^{N_j}\exp\Big(-\frac{||\boldsymbol{o}_t-\boldsymbol{e}_{ij}||^2}{\sigma}\Big) \qquad (1)$$

where $\boldsymbol{e}_{ij}$ is the i^{th} training exemplar of class s_j, $||\boldsymbol{o}_t-\boldsymbol{e}_{ij}||$ is the Euclidean distance between $\boldsymbol{o}_t$ and $\boldsymbol{e}_{ij}$, σ is a scale variable, N_j is the number of exemplars in class s_j, and Z is a normalization term to make Eq. (1) be a valid distribution.

From Eq. (1), the likelihood function is mathematically similar to a GMM with a shared scalar variance for all dimensions and Gaussians. Effectively, Eq. (1) puts a Gaussian-shaped function at each training exemplar and sums all these Gaussians with a normalization factor to the likelihood function.

There are four main steps to build an LVCSR system with the kernel density acoustic model [12]:

Step 1 Build a triphone-based HMM/GMM acoustic model.

Step 2 Generate tied-state label for each frame (exemplar) of training data using forced alignment. The training exemplars are then grouped based on their tied-state label.

Step 3 Use kernel density model to estimate HMM tied-state likelihood probability $p(\boldsymbol{o}_t|s_j)$ as in Eq. (1)[1].

Step 4 Plug the tied-state likelihood probability into a standard decoder such as Viterbi decoder for decoding.

III. Distance metric learning (DML) for kernel density acoustic model

To augment the kernel density model, a popular approach is to learn a Mahalanobis-based distance as [19]:

$$d^2(\boldsymbol{o}_t,\boldsymbol{e}_{ij}) = (\boldsymbol{o}_t-\boldsymbol{e}_{ij})^T\boldsymbol{M}(\boldsymbol{o}_t-\boldsymbol{e}_{ij}) \qquad (2)$$

where $d(\boldsymbol{o}_t,\boldsymbol{e}_{ij})$ is a Mahalanobis-based distance between test feature vector $\boldsymbol{o}_t$ and exemplar $\boldsymbol{e}_{ij}$, $\boldsymbol{M}$ is a matrix to be learnt. Since $\boldsymbol{M}$ is a symmetric positive semi-definite matrix, it can be factored as $\boldsymbol{M}=\boldsymbol{Q}^T\boldsymbol{Q}$. Hence Eq. (2) can be rewritten as

$$d^2(\boldsymbol{o}_t,\boldsymbol{e}_{ij}) = (\boldsymbol{Q}\boldsymbol{o}_t-\boldsymbol{Q}\boldsymbol{e}_{ij})^T(\boldsymbol{Q}\boldsymbol{o}_t-\boldsymbol{Q}\boldsymbol{e}_{ij}). \qquad (3)$$

This implies the Mahalanobis-based distance can be interpreted as the Euclidean distance in a transformed space, $\boldsymbol{o}_t \rightarrow \boldsymbol{Q}\boldsymbol{o}_t$ [19]. The purpose of metric learning is to learn transformation $\boldsymbol{Q}$ to transform the input space to a new space where the kernel density model can perform better. A similar spirit has been applied widely for HMM/GMM-based acoustic models where principal component analysis (PCA) [24], linear discriminant analysis (LDA) [25], heteroscedastic LDA (HLDA) [26] and multiple HLDA [27] are used to project input feature to a new space.

In this paper, a novel distance metric learning (DML) approach is proposed for kernel density model. The proposed method learns the Mahanalobis-based distance in Eq. (3) to optimize the frame accuracy on the training set by maximizing posterior probability $p(s_C|\boldsymbol{o}_t)$ of correct HMM state s_C for each input feature $\boldsymbol{o}_t$. In this work, for each $\boldsymbol{o}_t$, the cost function f which is a function of transformation $\boldsymbol{Q}$ is defined as the MMI (Maximum Mutual Information) criterion as follows:

$$f(\boldsymbol{Q}) = \log\left(p(s_C|\boldsymbol{o}_t)\right) = \log\left(\frac{p(\boldsymbol{o}_t|s_C)p(s_C)}{\sum\limits_{j=1}^{J}p(\boldsymbol{o}_t|s_j)p(s_j)}\right) \qquad (4)$$

where J is the number of HMM tied-states, s_C is the correct state label for input vector $\boldsymbol{o}_t$, $p(s_j)$ is the state prior estimated from training data, $p(\boldsymbol{o}_t|s_j)$ is the likelihood probability estimated by the kernel density model as in Eq. (5) for input feature $\boldsymbol{o}_t$ and state s_j.

$$p(\boldsymbol{o}_t|s_j)=\frac{1}{ZN_j}\sum_{i=1}^{N_j}\exp\Big(-(\boldsymbol{Q}\boldsymbol{o}_t-\boldsymbol{Q}\boldsymbol{e}_{ij})^T(\boldsymbol{Q}\boldsymbol{o}_t-\boldsymbol{Q}\boldsymbol{e}_{ij})\Big). \qquad (5)$$

Eq. (5) is derived from Eq. (1) by using the Mahalanobis-based distance defined in Eq. (3) and the scaling factor σ is set to 1.

The goal of distance metric learning is to find $\boldsymbol{Q}$ to maximize $f(\boldsymbol{Q})$. In this study, the gradient descent algorithm is applied to update $\boldsymbol{Q}$ iteratively. The proposed distance metric learning procedure is described as follows:

Step 1 Initialize transformation $\boldsymbol{Q} \in \mathcal{R}^{D\times D}$ as an identity matrix where D is the dimension of the input feature, $\boldsymbol{o}_t$.

Step 2 Compute the derivative of f with respect to $\boldsymbol{Q}$, $\frac{\partial f}{\partial \boldsymbol{Q}}$ [2].

Step 3 Update $\boldsymbol{Q}$ using the gradient descent method: $\boldsymbol{Q}^{new} \leftarrow \boldsymbol{Q}^{old} + \alpha\frac{\partial f}{\partial \boldsymbol{Q}}$, where α is the learning rate.

Step 4 Estimate the likelihood scores of all samples in the development set using Eq. (5). Convert these scores into state posteriors to compute the frame accuracy in the next step.

Step 5 The recognized state label for each sample (frame) in the development set is determined by picking the state label of the highest posterior score in that frame. Compute the frame accuracy in the development set by comparing the recognized state label and the ground truth, i.e. state label provided by forced alignment in the development set. If the frame accuracy still increases go back to Step 2, otherwise we stop the learning procedure.

IV. Experiments

A. Experimental procedures

We evaluate the performance of the proposed method on the WSJ task[3]. The WSJ task has been chosen as the target under-resourced language as the effect of sufficient training

[1] In fact, scaled likelihood is used since the normalization term, Z in Eq. (1) is the same for all classes and never needs to be computed [12].

[2] The detailed derivation for computing derivative of f with respect to $\boldsymbol{Q}$ is presented at http://www3.ntu.edu.sg/home2009/dova0001/DML.pdf

[3] We actually used the Aurora-4 corpus with clean training and test setting recorded at 16kHz. The Aurora-4 clean data is a filtered version of the WSJ0 SI84 training data and Nov92 test data.

data for it is well known, and we can hence clearly demonstrate the effect of the proposed work on small training sizes. Four different training data sizes of 7, 16, 55 and 220 minutes are randomly selected from the full 15 hours of SI84 training set. For each data size, we randomly extract around 10% from the training data to build the development set. The rest is used to train the models. The HMM/GMM system provides both the state-tying decision tree and frame level state label for the building of the hybrid HMM/DNN and the kernel density (HMM/KD) systems. Since the training data sizes are small, a small-scaled DNN architecture is used which consists of 3 hidden layers with 500 neurons in each hidden layer. The DNNs are initialized using RBM pre-training [16]. The test data are 166 clean utterances, or about 20 minutes of speech.

Two types of feature are investigated in this paper. The first feature is 39 dimensional MFCC, including 13 static features and its time derivatives. Utterance-based mean and variance normalization (MVN) is applied to reduce recording mismatch between training and testing conditions. The second feature is cross-lingual bottleneck feature [10], [12]. Using bottleneck feature generated by a bottleneck network which is well trained with a source language is a good option to improve the performance in the case of limited training data conditions. Cross-lingual bottleneck feature has been used effectively for HMM/GMM models [22], hybrid HMM/MLP [10] and in our recent kernel density model [12]. The cross-lingual bottleneck network is trained from more than 100 hours of Malay read speech data [15]. The detailed description of the bottleneck network architecture and the training procedure can be found in [10], [12].

In this work, the focus is on acoustic model training with limited training data, hence we assumed that the language model and pronunciation dictionary are available. The standard WSJ bigram LM and the 5k vocabulary are used in decoding. In the hybrid model (HMM/DNN) and the kernel density model (HMM/KD), for each HMM state, the probability of jumping to the next state is simply set to 0.5.

B. Baseline models

The results in word error rate (WER) obtained by various systems with four different amounts of target language training data are presented in Table I. The first and second system are the conventional HMM/GMM and hybrid HMM/DNN using MVN-processed MFCC features. As expected, the WER gets worse when less training data are used. The HMM/DNN system outperforms the HMM/GMM system significantly for all training data sizes.

The third row of Table I shows the results obtained by using MFCC feature with the plain HMM/KD [13], i.e. without distance metric learning. In this experiment, scaling factor σ in Eq. (1) is set to 1. Unfortunately, the kernel density produces worse results than the HMM/GMM baseline. As discussed in Section III, the reason is that the Euclidean distance is not robust to feature variation of MFCC.

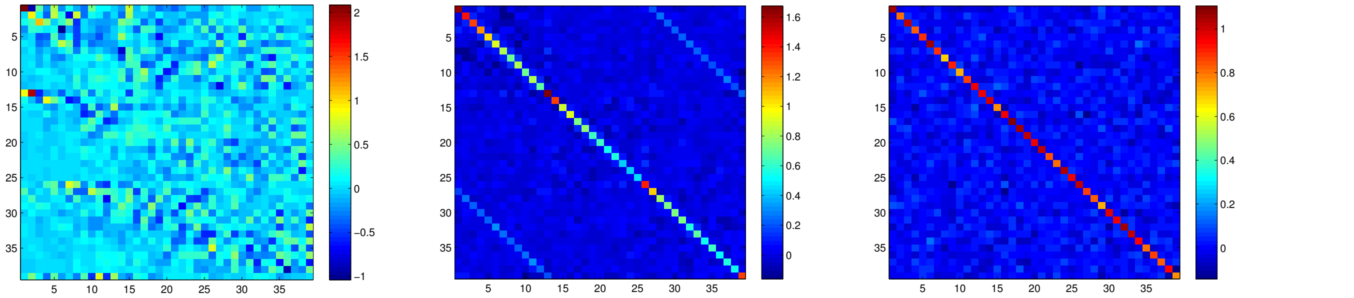
Next, let's examine the results obtained by using the cross-lingual bottleneck feature. In Table I, row 8 and 9 are the same as the row 1 and 2, except that MFCC feature is replaced by bottleneck feature. It is observed that using bottleneck feature significantly improves both the HMM/GMM and HMM/DNN systems especially for the case of very limited target language training data. These results shows the benefit of using cross-lingual features generated by the well-resourced language models. Note that at 220 minutes, the HMM/GMM and HMM/DNN with bottleneck feature actually perform worse than those systems with MFCC feature. This could be due to that the gain of bottleneck feature is offset by the loss due to the mismatch between the two corpora. This can be solved by applying adaptation (re-training) for the bottleneck network when more target language training data are available. Now, we focus on row 10 of Table I when the cross-lingual bottleneck feature is used for the HMM/KD model. A large improvement is observed over the result in row 3 where MFCC feature is used. Moreover, in the case of using the cross-lingual bottleneck feature, the HMM/KD model outperforms the HMM/GMM system in row 8 significantly and even compatible with the HMM/DNN system in row 9. This shows that the HMM/KD model can be well employed when good input features are used.

TABLE I
WER (%) OBTAINED BY VARIOUS SYSTEMS AT FOUR DIFFERENT TRAINING DATA SIZES. ROW 1-7 ARE RESULTS OBTAINED BY USING MFCC FEATURE. ROW 8-13 SHOW RESULTS OBTAINED BY USING CROSS-LINGUAL BOTTLENECK FEATURE. KD STANDS FOR KERNEL DENSITY USED FOR ACOUSTIC MODELING, DML STANDS FOR DISTANCE METRIC LEARNING, DST STANDS FOR DISCRIMINATIVE SCORE TUNING.

No	Acoustic model	Training data (minutes)			
		7	16	55	220
Monolingual (MFCC feature)					
1	HMM/GMM	30.9	23.1	14.0	9.1
2	HMM/DNN	24.1	17.9	11.3	7.8
3	Plain HMM/KD	33.7	26.2	15.5	11.5
4	HMM/KD+LDA	33.3	25.6	15.1	11.0
5	HMM/KD+DML	25.6	19.5	11.5	8.3
6	HMM/KD+DST	26.7	19.8	12.4	8.7
7	HMM/KD+DML+DST	**23.3**	**17.1**	**9.7**	**7.0**
Cross-lingual (cross-lingual bottleneck feature)					
8	HMM/GMM	24.6	18.5	11.3	10.1
9	HMM/DNN	17.5	15.3	10.3	8.5
10	Plain HMM/KD	18.8	15.8	10.6	8.2
11	HMM/KD+DML	18.4	15.6	10.3	8.0
12	HMM/KD+DST [12]	15.8	13.3	9.9	7.9
13	HMM/KD+DML+DST	**15.7**	**13.1**	**9.8**	**7.6**

C. Distance metric learning for kernel density model

The experiments in the previous section indicated that the kernel density model [13], [12] with the simple Euclidean distance is outperformed by the conventional HMM/GMM model when MFCC is used as the input feature. However, using bottleneck feature, the kernel density model achieves a significantly better performance than the HMM/GMM model. To improve the kernel density model with MFCC feature, the Euclidean distance is replaced by the Mahalanobis-based distance as in Eq. (3). The transformation matrix $\boldsymbol{Q}$ is trained following the procedure in Section III. We use the mini-batch mode update strategy with the mini-batch size of 50, i.e. $\boldsymbol{Q}$



(a) MFCC + LDA (b) MFCC + distance metric learning (c) BN + distance metric learning

Fig. 1. Illustration of the linear feature transformation matrices. BN stands for cross-lingual bottleneck features. The MFCC feature vector are ordered as [c1,...,c12,c0], and their delta and acceleration versions.

is updated after every 50 frames using accumulative gradient. The learning rate, α is set to 0.0002.

The result of the kernel density model with distance metric learning (DML) for MFCC feature is listed in row 5 of Table I. A large improvement is achieved over the plain kernel density model in row 3. This improvement remains stable over different training data sizes i.e. from 24% to 28% relative. The result in row 5 is also significant better than the HMM/GMM model in row 1 and approaching the performance of the HMM/DNN model in row 2.

For comparison, we also apply LDA which uses the label of training data to linearly separate classes. In this experiment, MFCC is first applied LDA to keep all 39 dimensions and then it is used as the input for the conventional kernel density model. As shown in row 4 of Table I, using LDA just results in a small improvement in WER over the plain kernel density model with MFCC in row 3. The study in [23] indicated that LDA suffers from a small sample size problem when dealing with high-dimensional data. Our experimental results show that LDA can give 4.3% relative improvement for the case of 4 hours, this improvement drops to 1.2% when 7 minutes of training data are used. In another experiment, we use the standard Mahalanobis distance as in Eq. (2) where $\boldsymbol{M} = \boldsymbol{S}^{-1}$ with $\boldsymbol{S}$ is the covariance matrix of input feature $\boldsymbol{o}_t$. However, the result is even not as good as using the Euclidean distance and hence is not reported here.

We also apply the DML for cross-lingual bottleneck feature, the result is shown in row 11 of Table I. Unlike the case of MFCC feature, applying DML for bottleneck feature achieves only a small improvement over the plain kernel density model in row 10. It indicates that DML is more important when we apply the kernel density model to low level features such as MFCC.

To give an insight to why the proposed DML approach can improve performance of the kernel density acoustic model, we compare the feature transformations learnt by different methods. The MFCC feature transformations learnt by LDA and the proposed DML are shown in Fig. 1(a), 1(b), respectively. It can be observed that the transformation learnt by DML has an obvious diagonal structure. From the values of the diagonal elements, the weights of MFCC features c0-c12 are almost monotonically decreasing, meaning that lower order MFCC features are more important than higher order MFCC features for frame classification. It is also observed that there are two off diagonal bars in the transformation, which model the correlation between static and corresponding acceleration features. On the other hand, there is no clear structure in LDA-derived transformation matrix.

The DML learnt transformation matrix for bottleneck features is shown in Fig. 1(c). It is observed that the BN transformation matrix is closer to the identity matrix than the MFCC transformation matrix, although both are learnt by DML. The diagonal values of the BN transformation are similar to each other, meaning that all BN features contribute similarly to the kernel density model and hence speech recognition. In addition, there is no clear off-diagonal structure, this may indicate that there is no obvious correlation between BN features that can be modeled for better frame classification performance. Above observations are reasonable considering that BN features are extracted by a deep neural network that was trained to discriminate sound classes. Hence there is less gain to apply DML on BN features than on MFCC features.

D. Discriminative score tuning (DST) for kernel density model

Recently, the combination of generative and discriminative models have been shown to improve performance of speech recognition [28], [29]. To improve the performance of the kernel density model, likelihood scores generated by the kernel density model are further refined by the discriminative score tuning (DST) module proposed in our previous study [12]. Specifically, a 2-layer neural network is placed on the top of the kernel density model to fine tune the likelihood scores. The number of inputs and outputs of the neural network is the same and equal to the number of HMM states. This neural network is trained with cross-entropy criterion to minimize the training frame classification error.

The results of the kernel density model with DST are shown in row 6 of Table I for the case of MFCC input and row 12 for the case of bottleneck feature input. It can be seen that using the discriminative score tuning can significantly improve the performance of the kernel density model for both the cases when MFCC and bottleneck features are used.

Now we examine whether further improvement can be achieved when both the DML and DST techniques are applied for the kernel density model. As shown in row 7 of Table I, applying the two techniques results in a significant improvement over using individual techniques for the case of MFCC feature. This demonstrates that DML and DST is highly complementary. While the DML makes more discriminative feature for the kernel density model, DST aims to fine tune the likelihood scores in a discriminative manner. The result of combination DML and DST for the case of cross-lingual bottleneck feature is presented in row 13 of Table I. We can see that the kernel density model with DML and DST provides the best performance over the GMM and even the DNN models for all four training data sizes with both the MFCC and bottleneck feature inputs.

V. Conclusion

In this paper, we proposed a method to augment the non-parametric kernel density-based acoustic model using distance metric learning method. Specifically, the Mahanalobis-based distance is realized by linear feature transformation and trained to improve the frame accuracy on the training set using the MMI criterion. With this approach, the performance of the kernel density model is improved significantly. The experimental results on the WSJ corpus showed that the proposed system produces consistently better results than both the HMM/GMM and HMM/DNN systems, up to 220 minutes of training data. This shows that the proposed system has advantages over conventional systems for small training sizes.

One issue of using the kernel density model is its high computational cost for decoding especially when training data size increases. To reduce the decoding time, pruning techniques will be investigated in the future works.

Acknowledgement

This work is supported by DSO National Laboratories, Singapore, Project MAISON DSOCL14045.

References

[1] H. Li, K. A. Lee, and B. Ma, "Spoken Language Recognition: From Fundamentals to Practice," Proceedings of the IEEE, Vol. 101, No. 5, May 2013, pp. 1136–1159.

[2] T. Schultz and A. Waibel, "Experiments On Cross-Language Acoustic Modeling," in Proc. International Conference on Spoken Language Processing (ICSLP), 2001, pp. 2721-2724.

[3] A. Stolcke, F. Grezl, M. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2006, pp. 321-324.

[4] L. Burget, et al. "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010, pp. 4334-4337.

[5] D. Imseng, H. Bourlard, and P. N. Garner, "Using KL-divergence and multilingual information to improve ASR for under-resourced languages," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 4869–4872.

[6] D. Imseng, P. Motlicek, H. Bourlard and P. N. Garner, "Using out-of-language data to improve an under-resourced speech recognizer," Speech communication, Vol. 56, 2014, pp. 142–151.

[7] K. C. Sim and H. Li, "Context Sensitive Probabilistic Phone Mapping Model for Cross-lingual Speech Recognition," in Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), 2008, pp. 2715-2718.

[8] K. C. Sim and H. Li, "Stream-based Context-sensitive Phone Mapping for Cross-lingual Speech Recognition," in Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), 2009, pp. 3019-3022.

[9] V. H. Do, X. Xiao, E. S. Chng, and H. Li, "Context dependant phone mapping for cross-lingual acoustic modeling," in Proc. International Symposium on Chinese Spoken Language Processing (ISCSLP), 2012, pp. 16–20.

[10] V. H. Do, X. Xiao, E. S. Chng, H. Li, "Context-dependent phone mapping for LVCSR of under-resourced languages," in Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), 2013, pp. 500–504.

[11] V. H. Do, X. Xiao, E. S. Chng, H. Li, "Cross-lingual phone mapping for large vocabulary speech recognition of under-resourced languages," IEICE Transactions on Information and Systems, Vol. E97-D, No. 2, 2014, pp. 285–295.

[12] V. H. Do, X. Xiao, E. S. Chng, H. Li, "Kernel Density-based Acoustic Model with Cross-lingual Bottleneck Features for Resource Limited LVCSR," in Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), 2014, pp. 6–10.

[13] T. Deselaers, G. Heigold, and H. Ney, "Speech recognition with state-based nearest neighbour classifiers," in Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), 2007, pp. 2093-2096.

[14] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernolle, K. Demuynck, J. F. Gemmeke, J. R. Bellegarda, and S. Sundaram, "Exemplar-based processing for speech recognition: An overview," IEEE Signal Processing Magazine, vol. 29, no. 6, 2012, pp. 98-113.

[15] X. Xiao, E. S. Chng, T. P. Tan, and H. Li, "Development of a Malay LVCSR System," in Proc. Oriental COCOSDA, 2010.

[16] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Computation 18, 2006, pp. 1527–1554.

[17] D. Povey, "Discriminative training for large vocabulary speech recognition", Ph.D. thesis, Cambridge University, 2004.

[18] S. Young and others, "The HTK book", Cambridge university engineering department, 2006.

[19] J. Goldberger, S. Roweis, G. E. Hinton, R. Salakhutdinov, "Neighbourhood components analysis," in Proc. Advances in Neural Information Processing Systems (NIPS), 2005, pp. 513–520.

[20] X. He, P. Niyogi, "Locality preserving projections," in Proc. Advances in Neural Information Processing Systems (NIPS), 2003.

[21] K. Q. Weinberger, L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," Journal of Machine Learning Research, Vol 10, 2009, pp. 207–244

[22] K. Vesely, M. Karafiat, F. Grezl, M. Janda, E. Egorova, "The Language-Independent Bottleneck Features," in Proc. Workshop on Spoken Language Technology (SLT), 2012, pp. 336-341.

[23] L. Chen, H. Liao, M. Ko, J. Lin, G. Yu, "A new lda-based face recognition system which can solve the small sample size problem," Pattern Recognition, 2000, pp. 1713-1726.

[24] C. M. Bishop, "Pattern Recognition and Machine Learning," New York NY: Springer, 2006.

[25] P. Brown, "The acoustic-modeling problem in automatic speech recognition," Ph.D. thesis, Computer Science Department, Carnegie-Mellon University, 1987.

[26] N. Kumar, A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rand HMMs for improved speech recognition," Speech Communication, vol. 26, pp. 283297, 1998.

[27] M. J. F. Gales, "Maximum likelihood multiple subspace projections for hidden Markov models," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 2, pp. 3747, 2002.

[28] M. J. F. Gales, F. Flego, "Discriminative classifiers with adaptive kernels for noise robust speech recognition," Computer Speech and Language, 2010, Vol 24, No 4, pp. 648–662.

[29] A. Ragni, M. J. F. Gales, "Derivative kernels for noise robust ASR," in Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2011, pp. 119-124.