Robustness of Several Kernel-based Fast Adaptation Methods on Noisy LVCSR

Brian Mak

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
mak@cse.ust.hk

*Roger Hsiao**

Language Technologies Institute,
School of Computer Science,
Carnegie Mellon University,
Pittsburgh, Pennsylvania, USA.
wrhsiao@cs.cmu.edu

Abstract

We have been investigating the use of kernel methods to improve conventional linear adaptation algorithms for fast adaptation, when there are less than 10s of adaptation speech. On clean speech, we had shown that our new kernel-based adaptation methods, namely, *embedded kernel eigenvoice* (eKEV) and *kernel eigenspace-based MLLR* (KEMLLR) outperformed their linear counterparts. In this paper, we study their unsupervised adaptation performance under additive and convoluted noises using the Aurora4 Corpus, with no assumption or prior knowledge of the noise type and its level. It is found that both eKEV and KEMLLR adaptation continue to outperform MAP and MLLR, and the simple reference speaker weighting (RSW) algorithm continues to perform favorably with KEMLLR. Furthermore, KEMLLR adaptation gives the greatest overall improvement over the speaker-independent model by about 19%.

Index Terms: fast adaptation, kernel method, kernel eigenspace-based MLLR, embedded kernel eigenvoice, MAP, MLLR, reference speaker weighting.

1. Introduction

Noise robustness is a key issue for successful commercialization of speech recognition systems. This influences the design of any new speech processing algorithm: an algorithm that is noise robust is preferable to one that is not.

In the past few years, we have been trying to improve various conventional linear adaptation algorithms by the use of kernel methods [1] for fast speaker adaptation. In fast adaptation, the amount of adaptation speech is less than 10s. The idea is that for an algorithm that is based on some linear operation, one may try to exploit possible nonlinearity in the data using the kernel trick. Conceptually, a nonlinear function is used to map the operands of the linear operation to a kernel-induced high-dimensional feature space, where conventional linear methods are then applied. It turns out that the nonlinear map need not be known; instead, the computation depends only on the inner products in the feature space, which can be obtained efficiently with a suitable nonlinear kernel function. That is, the nonlinearity is captured by the kernel function. Thus, the use of kernel method provides an elegant nonlinear generalization of existing linear algorithms. In [2, 3] we kernelize the eigenvoice adaptation method [4] in our new *embedded kernel eigenvoice* (eKEV) adaptation method, whereas in [5, 6], the eigen-MLLR adaptation is kernelized to our new *kernel eigenspace-based MLLR* (KEMLLR) adaptation method. On clean speech, the two kernel-based adaptation algorithms were shown to outperform their linear counterparts as well as the conventional adaptation methods such as the Bayesian-based *maximum a posteriori* (MAP) adaptation [7] and the transformation-based *maximum likelihood linear regression* (MLLR) adaptation [8].

Although eKEV and KEMLLR had been shown to perform well in fast speaker adaptation from small-vocabulary task (using TIDIGITS [9]), medium-vocabulary task (using the Resource Management Corpus [10]), to large-vocabulary task (using the 5K Wall Street Journal Corpus [11]), their effectiveness under noisy conditions has yet to be shown. In this paper, we report and compare the adaptation performance of eKEV and KEMLLR on the noisy large-vocabulary Aurora4 [12] recognition task, which is basically the noisy version of the 5K Wall Street Journal task.

2. Aurora4 Evaluation

2.1. Aurora4 Corpus

The Aurora4 Corpus was derived from the 5K WSJ0 Corpus [11]. The training set was modified from the SI-84 WSJ0 training set, and the evaluation set was modified from the November'92 NIST evaluation set. It consists of several different training sets and evaluation sets, depending on how the data were downsampled, filtered, and how noises were added.

In this study, we use only the Training Set 1 (TS1) for training our baseline acoustic models. TS1 consists of the 7138 utterances from 83 speakers in the SI-84 WSJ0 training set (recorded using the Sennheiser microphone and sampled at 16kHz) but were filtered by P.341 filtering. Similarly the evaluation set of WSJ0, consisting of 330 utterances, was adopted by Aurora4 after P.341 filtering. Fourteen versions of this filtered evaluation set were created as follows. Each of the clean filtered versions of the evaluation set recorded with Sennheiser microphone and a secondary microphone were selected to form evaluation set 1 and 8. Six different types of noises: car noise, babble, restaurant noise, street noise, airport noise, and train noise, of signal-to-noise level between 5–15 dB were digitally added to evaluation set 1 and 8 to form evaluation set 2–7 and 9–14 respectively. Thus, there are totally $14 \times 330 = 4620$ utterances in Aurora4's evaluation set.

*Roger Hsiao finished this work when he was a graduate student at the Department of Computer Science of HKUST.

2.2. Acoustic Modeling

The feature extraction and acoustic modeling procedure follows the practice in Aurora4 evaluation. In particular, the ETSI advanced frontend [13] was used to extract the conventional 39-dimensional MFCC vectors at every 10ms over a window of 25ms.

The speaker-independent (SI) model consists of 15,449 cross-word triphones based on 39 base phonemes. Each of them was modeled as a continuous density HMM (CDHMM) which is strictly left-to-right and has three states with a Gaussian mixture density of four components per state. The number of tied states is around 3000. The reduced complexity of the HMMs was adopted by Aurora4 evaluation community to allow more experiments to be run to test different configurations in terms of compression, frontend algorithm, etc.

2.3. Adaptation Experiments

Unsupervised adaptation was carried out using one single utterance at a time. That is, each utterance in an evaluation set was first used to adapt the clean SI models, and the adapted model obtained was then used to re-decode the same utterance to get the final recognition result. The utterance duration varies from the minimum 2.06s to the maximum 14.19s with an average of 7.31s. Thus, the task requires fast adaptation methods when only one utterance is available for adaptation; this is the focus of this research.

2.3.1. Model and Adaptation Methods

The following model and adaptation methods are compared on the Aurora4 task [12].

SI : speaker-independent model.

MAP : MAP adaptation [7]

MLLR : MLLR adaptation [8].

RSW : reference speaker weighting [14, 15].

eKEV : embedded kernel eigenvoice adaptation [2, 3].

KEMLLR : kernel eigenspace-based MLLR adaptation [5, 6].

Since the five adaptation methods under investigation have been well documented in the literature, they are not reviewed here again due to the limited space. The readers are referred to the citations for details. Briefly speaking, eKEV and KEMLLR generalize their linear counterparts, eigenvoice (EV) and eigen-MLLR (EMLLR), using kernel principal component analysis (KPCA). They differ mainly in their representation of speaker supervectors. In EV/eKEV, each speaker supervector consists of all the Gaussian mean vectors in his hidden Markov models; in EMLLR/KEMLLR, a speaker-dependent model is created by MLLR adaptation from a speaker-independent model, and his speaker supervector is created by stacking up all his MLLR transforms. The major challenge in the generalization of EV/EMLLR to eKEV/KEMLLR is to formulate the acoustic likelihoods in terms of the kernel values during decoding as well as maximum-likelihood estimation of the eigenvoice weights.

We had not run comparison experiments with the linear counterparts of eKEV and KEMLLR, namely EV and EMLLR. The reason is that in our past investigation on speaker adaptation [2, 3, 5, 6], eKEV and KEMLLR always perform better than EV and EMLLR respectively.

2.3.2. Adaptation Details

Among the five adaptation methods under investigation, MAP and MLLR were performed using the HTK toolkit [16]. The remaining three adaptation methods were implemented by us. Their system parameters were tuned to give the best performance in the Resource Management task, and were then simply adopted without any change for Aurora4 evaluation.

The settings of various system parameters are described below.

MAP : Scaling factors between 1 and 20 were tried, and the result from the best value was reported.

MLLR : Due to the short duration of adaptation utterances (with an average duration of 7.31s), it was found that a single global MLLR transformation with full covariance gave the best overall result[1].

RSW : All 83 training speakers were used as reference speakers.

eKEV : Due to the use of nonlinear kernel function in eKEV adaptation, there is no analytical solution for the maximum-likelihood estimation of its eigenvoice weights. Instead, they were found by the quasi-Newton BFGS numerical algorithm.

- Number of eigenvoices to use = 10.
- Number of maximum-likelihood reference speakers = 5.
- Gaussian composite kernels of the form $k(\mathbf{u}, \mathbf{v}) = \exp(-\beta\|\mathbf{u}-\mathbf{v}\|^2)$ were adopted; and $\beta = 0.005$.

KEMLLR : Again the eigenvoice weights were estimated using the same numerical method as in eKEV.

- The number of eigenvoices to use equals to the number of speakers, which is 83 in Aurora4.
- Again Gaussian composite kernels were adopted, and $\beta = 0.001$.

Finally, all speech decoding were carried out using the HTK software.

2.4. Results and Discussions

Performance of each adaptation method on the 14 evaluation data sets is detailed in Table 1. The results in the table are also plotted in two figures: Fig. 1 compares the various adaptation methods on the evaluation data sets 1–7 which were recorded by the same microphone that was used to record the training data, whereas Fig. 2 shows the comparison on evaluation data sets 8–14 which were recorded by a different microphone. (The results of MAP are not plotted in Fig. 1 and 2 because they are too close to SI's.)

To simplify our discussion, results for data sets 2–7 and 9–14 are separately averaged and summarized in Table 2, and the corresponding reduction in word error rate (WER) are tabulated in Table 3.

From Table 1–3 and Fig. 1–2, we have the following observations:

[1] MLLR with a regression class tree of 32 nodes had been tried with different threshold counts in order to automatically select the right number of transforms for utterances of various durations. Although some utterances might benefit from using more than one transform, on the whole, we did not find a threshold count that could give a better result than simply using a single global transform.

Table 1: Recognition performance (in word error rate %) of various adaptation algorithms on each test subset of Aurora4.

Set	SI	MAP	MLLR	eKEV	RSW	KEMLLR
1	12.45	12.27	12.12	10.72	10.28	10.39
2	25.05	25.08	23.79	21.36	21.73	21.33
3	27.92	27.92	27.07	23.68	21.95	22.28
4	32.52	32.52	31.34	29.76	26.04	26.52
5	30.76	30.76	30.02	27.30	24.97	26.56
6	28.25	28.25	26.89	23.65	22.21	22.62
7	32.04	32.04	31.31	27.85	26.00	24.68
8	30.87	30.98	26.47	27.29	24.68	22.87
9	40.52	40.41	36.91	34.88	32.89	32.67
10	43.43	43.35	40.70	39.96	36.76	34.51
11	45.23	45.19	44.20	41.62	38.53	37.16
12	46.37	46.37	43.65	41.69	38.56	37.61
13	44.35	44.35	42.10	40.99	37.68	35.65
14	45.49	45.49	43.87	41.99	39.15	37.27

Table 2: Summary of average recognition performance (in word error rate %) of various adaptation algorithms on various groups of Aurora4 test subsets. (Data set 1 is the clean reference; set 2–7 contain different additive noises; set 8 is clean but with convoluted noise; set 9–14 contain different additive noises plus convoluted noise.)

Set	SI	MAP	MLLR	eKEV	RSW	KEMLLR
1	12.45	12.27	12.12	10.72	10.28	10.39
2–7	29.42	29.43	28.40	25.60	23.82	24.00
8	30.87	30.98	26.47	27.29	24.68	22.87
9–14	44.23	44.19	41.91	40.19	37.26	35.81
overall	34.66	34.64	32.89	30.91	28.67	28.01

- Compared with the SI baselines (on set 1 and 8), all adaptation methods except MAP, continue to help in combating noises.
- MAP is not effective for such small amount of adaptation data.
- The WER more than doubles when there is a mis-match in the channel in the absence of additive noises.
- When there is no channel mis-match between the training data and the test data, the presence of noises does not change the relative performance of the 5 adaptation methods under investigation that has been observed in the past (when they were tested using TIDIGITS, RM, or WSJ0). That is, the order of adaptation performance is:

$$\text{SI} \sim \text{MAP} < \text{MLLR} < \text{eKEV} < \text{RSW} \sim \text{KEMLLR}.$$

- When there is a channel mis-match, the relative order of the performance is similar except that now KEMLLR is obviously better than RSW. This can be explained by the fact that RSW is still using the reference speaker models that were trained only with clean speech recorded by the Sennheiser microphone, and the adapted model must be on the span of the Sennheiser acoustic space. On the other hand, the adapted model produced by KEMLLR does not have such limitation.
- That eKEV is outperformed by the simpler RSW can be explained by the fact that the pre-imaging algorithm used

Table 3: Summary of % word error rate reduction of various adaptation algorithms on various groups of Aurora4 test subsets. (Data set 1 is the clean reference; set 2–7 contain different additive noises; set 8 is clean but with convoluted noise; set 9–14 contain different additive noises plus convoluted noise.)

Set	MAP	MLLR	eKEV	RSW	KEMLLR
1	1.45	2.65	13.9	17.43	16.55
2–7	0.02	3.47	12.99	19.06	18.44
8	-0.36	14.25	11.60	20.02	25.92
9–14	0.09	5.26	9.14	15.76	19.04
overall	0.06	5.11	10.82	17.27	19.19

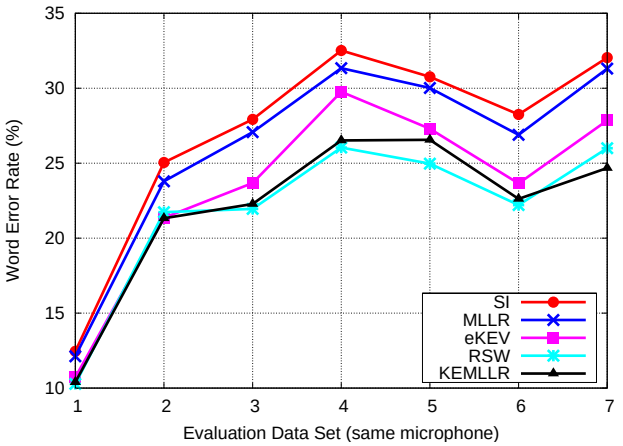

Figure 1: Recognition performance of various adaptation algorithms on test subsets 1–7 of Aurora4 which were recorded with the same microphone that recorded the training set.

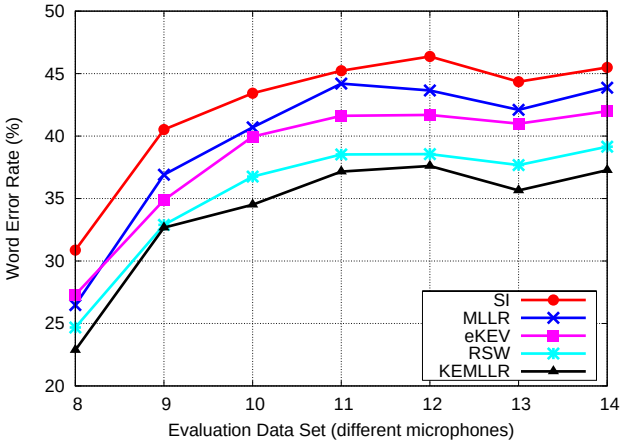

Figure 2: Recognition performance of various adaptation algorithms on test subsets 8–14 of Aurora4 which were recorded with a microphone different from that used to record the training set.

in eKEV to project the adapted model from the kernel-induced high-dimensional feature space back to the input space also makes use of some reference speakers. As a result, it is similar to RSW but with a smaller solution space that is constrained by the image in the feature space.

- The performance difference between MLLR and eKEV also shrinks when there is a channel mis-match.

3. Conclusions

In this paper, we compare the performance of five adaptation methods on noisy LVCSR. The five methods are MAP, MLLR, RSW, eKEV, and KEMLLR. The first three are linear methods while the last two make use of kernel methods. The presence of additive noises does not change the relative order of the performance of the five adaptation methods that we have observed in the past when we worked on clean speech; the order shows that KEMLLR and RSW have comparable performance and they significantly outperform the others. When there is convoluted noise (channel mis-match), it is obvious that KEMLLR has better adaptation performance than RSW and other methods.

Since fast adaptation is the focus of this research, we did not show any performance comparison between our new kernel-based adaptation methods with the conventional ones when there are more than 10s of adaptation speech. On the other hand, there is no reason to suggest that they will perform worse. In the worst case, they may simply switch to using linear kernel (instead of the current Gaussian kernels) for longer adaptation data, and the kernel-based adaptation methods will be reduced to their linear counterparts and match the proven performance of the latter.

4. Acknowledgments

This research is partially supported by the Research Grants Council of the Hong Kong SAR under the grant numbers HKUST617406.

5. References

[1] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[2] Brian Mak, J. T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 984–992, September 2005.

[3] Brian Mak, R. Hsiao, S. Ho, and J. T. Kwok, "Embedded kernel eigenvoice speaker adaptation and its implication to reference speaker weighting," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 4, pp. 1267–1280, July 2006.

[4] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 695–707, Nov 2000.

[5] R. Hsiao and Brian Mak, "Kernel eigenspace-based MLLR adaptation using multiple regression classes," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, March 18–23 2005, vol. 1, pp. 985–988.

[6] Brian Mak and R. Hsiao, "Kernel eigenspace-based MLLR adaptation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 784–795, March 2007.

[7] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, April 1994.

[8] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[9] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984, vol. 3, pp. 4211–4214.

[10] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-word Resource Management database for continuous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, vol. 1, pp. 651–654.

[11] D. B. Paul and J. M. Baker, "The design of the Wall Street Journal-based CSR corpus," in *Proceedings of the DARPA Speech and Natural Language Workshop*, Feb. 1992.

[12] N. Parihar and J. Picone, "DSR front end LVCSR evaluation," *AU/384/02, Aurora Working Group*, Dec. 2002, (http://www.isip.msstate.edu/projects/aurora).

[13] ETSI, *ES 202 050 v1.1.3 speech Processing, Transmission and Quality aspects (STQ); Distribution Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms.*, Nov. 2003.

[14] Tim J. Hazen, "A comparison of novel techniques for rapid speaker adaptation," *Speech Communications*, vol. 31, pp. 15–33, May 2000.

[15] Brian Mak, Tsz-Chung Lai, and Roger Hsiao, "Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 14–19 2006, vol. 1, pp. 229–232.

[16] Steve Young et al., *The HTK Book (Version 3.2)*, University of Cambridge, 2002.