SUBSPACE PURSUIT METHOD FOR KERNEL-LOG-LINEAR MODELS

Yotaro Kubo [1,2,3], *Simon Wiesler*[2], *Ralf Schlueter*[2], *Hermann Ney*[2],
Shinji Watanabe[3], *Atsushi Nakamura*[3], *Tetsunori Kobayashi*[1]

[1] Dept. of Computer Science and Engineering, Waseda University, Tokyo, Japan.
[2] Computer Science Department, RWTH Aachen University, Aachen, Germany.
[3] NTT Communication Science Laboratories, Kyoto, Japan.

ABSTRACT

This paper presents a novel method for reducing the dimensionality of kernel spaces. Recently, to maintain the convexity of training, log-linear models without mixtures have been used as emission probability density functions in hidden Markov models for automatic speech recognition. In that framework, nonlinearly-transformed high-dimensional features are used to achieve the nonlinear classification of the original observation vectors without using mixtures. In this paper, with the goal of using high-dimensional features in kernel spaces, the *cutting plane subspace pursuit* method proposed for support vector machines is generalized and applied to log-linear models. The experimental results show that the proposed method achieved an efficient approximation of the feature space by using a limited number of basis vectors

Index Terms— Automatic speech recognition, kernel method, subspace method, log-linear model, dimensionality reduction

1. INTRODUCTION

In general, the conventional training methods for mixture models used in automatic speech recognition have often suffered from local optimum problems due to the non-convexity of training. The use of convex optimization for training appears to be attractive way of preventing these local optimum problems by considering the success of support vector machines (SVMs). However, probabilistic models with latent variables are no longer available for making an optimization convex. Because current speech recognition systems perform nonlinear classification by employing Gaussian mixture models (GMMs), local-optima problems cannot be ignored even if the hidden Markov model (HMM) state alignment is fixed to avoid the latent variable of HMMs.

Recently, it has been confirmed that the use of high-dimensional features obtained by nonlinearly warping the observation vectors enables the nonlinear classification of the observation vectors without using mixture models. Wiesler *et al.* achieved comparable performance with a conventional GMM-based speech recognizer for large vocabulary continuous speech recognition (LVCSR) task by employing polynomial features and sparse cluster features obtained by probabilistic clustering [1]. It should be noted that high-dimensional features are also used for non-convex models. Povey *et al.* applied a subspace method, called fMPE, to high-dimensional feature vectors to obtain modified features that maximize the non-convex performance function [2].

With these approaches, the high-dimensional features are explicitly defined and computed. Therefore, the computational time has at least to be proportional to the number of features. However, with SVMs, several methods based on the kernel trick have been developed to handle extremely high-dimensional features. Kernel methods handle high-dimensional transformed features $\boldsymbol{\phi}(\boldsymbol{x})$ of the observation vector $\boldsymbol{x}$ simply by focusing on the inner product function, called the "kernel function," of the transformed features $K(\boldsymbol{x}, \boldsymbol{y}) \overset{\mathrm{def}}{=} \boldsymbol{\phi}(\boldsymbol{x})^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{y})$. Since the most commonly used kernel functions can be computed simply by using the observation vectors, the kernel methods do not necessitate the computation of the high-dimensional features $\boldsymbol{\phi}(.)$. Although the efficiency of kernel methods in speech recognition tasks has been partially confirmed by phoneme classification experiments [3, 4], the efficiency in continuous speech recognition tasks has not been evaluated yet since the kernel methods require an enormous amount of computational time. In kernel-based methods, computational time $O(T^2)$, where T is the number of observation vectors in the training dataset, is the minimum requirement because parameter vectors in feature spaces are represented as a linear combination of all vectors in the training dataset. Thus, the naive applications of kernel methods cannot be used for automatic speech recognition tasks because these tasks generally involve over 1 million observation vectors. To make computation tractable, several techniques that use a limited number of basis vectors in kernel spaces instead of using all the vectors in the training dataset have been developed in the machine learning field. For example, principal component analysis (PCA) and linear discriminant analysis (LDA) are enhanced by using kernel methods [5, 6]. Although these methods can reduce the computational time required for model parameter estimation, the computational time for pre-processing becomes $O(T^3)$ by introducing these methods. The Nyström method uses basis vectors derived from randomly sampled feature vectors from the training dataset [7]. Therefore, the computational cost of pre-processing is sufficiently low; however, the efficiency of randomly sampled basis vectors is questionable. On the other hand, the cutting-plane subspace pursuit (CPSP) method proposed by Joachims and Yu [8] provides a basis optimization approach that can be performed with a computational time of approximately $O(T)$. However, since the method is closely associated with the SVM training criterion, it is difficult to apply it directly to the conventional automatic speech recognition models.

In this paper, by using the same strategy as that used with the CPSP method, we propose a generalized version of the CPSP method that can be used for log-linear models trained with conventional discriminative training criteria, This method provides an efficient subspace approximation method for kernel-based speech recognition with the aim of realizing the practical use of kernel methods in speech recognition. The rest of this paper is organized as follows. In Section 2, we describe the model definition and how to train the models with the fixed basis vectors. In Section 3, we propose the basis vector optimization method. In Section 4, we present the experimental setup, experimental results, and discussions.

2. SUBSPACE KERNEL-LOG-LINEAR MODELS

Hereafter, although several training criteria can be associated with the proposed method, we describe the case of the frame-wise maximum mutual information (MMI) criterion [1, 9]. In frame-wise

978-1-4577-0539-7/11/$26.00 ©2011 IEEE

MMI, the training dataset is decomposed into observation vectors and corresponding HMM state-level labels, i.e., the training dataset $\mathcal{Z}$ is denoted as $\mathcal{Z} \overset{\text{def}}{=} \{(\boldsymbol{x}_t, l_t)|\forall t \in [1..T]\}$ where T is the number of frames in the training dataset. The state-level labels (l_t) can be obtained in practice by using the forced alignment of baseline HMM systems. The frame-wise MMI method is performed by maximizing the sum of the logarithmic posterior probabilities of the corresponding HMM states, as follows:

$$\hat{\Lambda} = \operatorname*{argmax}_{\Lambda} \underbrace{\sum_t \log P(l_t|\boldsymbol{x}_t, \Lambda) - c\sum_s ||\boldsymbol{\lambda}_s||^2}_{F(\Lambda)}, \tag{1}$$

where Λ is a set of parameter vectors, i.e, $\Lambda \overset{\text{def}}{=} \{\boldsymbol{\lambda}_s|\forall s\}$, s is a variable that denotes an HMM state, and c is a hyper-parameter that represents a scale-factor of the above regularization term.

The following log-linear model is used as $P(s|\boldsymbol{x}_t, \Lambda)$, which includes the nonlinear transformation function $\boldsymbol{\phi}$, as follows:

$$P(s|\boldsymbol{x}_t, \Lambda) \overset{\text{def}}{=} \frac{\exp\left\{\boldsymbol{\phi}(\boldsymbol{x}_t)^{\mathsf{T}}\boldsymbol{\lambda}_s\right\}}{\sum_{s'} \exp\left\{\boldsymbol{\phi}(\boldsymbol{x}_t)^{\mathsf{T}}\boldsymbol{\lambda}_{s'}\right\}}. \tag{2}$$

The emission probability used in a decoding process is obtained as $P(\boldsymbol{x}_t|s) = P(s|\boldsymbol{x}_t, \Lambda)P(\boldsymbol{x}_t)/P(\boldsymbol{s})$ where $P(\boldsymbol{x}_t)$ can be omitted without any loss of strictness because this probability is the same for all hypotheses in the decoding process. $P(s)$ can be obtained by computing the frequency of the HMM states in the training dataset, or assuming the uniform distribution.

We introduce a subspace approximation that assumes that the parameter vector $\boldsymbol{\lambda}_s$ of the s^{th} HMM state is in the space spanned by M basis vectors denoted by $\boldsymbol{\phi}(\boldsymbol{y}_m)$ ($m \in [1..M]$), as follows:

$$\boldsymbol{\lambda}_s \approx \sum_{m=1}^{M} \beta_{s,m}\boldsymbol{\phi}(\boldsymbol{y}_m), \tag{3}$$

where $\boldsymbol{y}_m$ is a vector in the observation space corresponding to the basis vector $\boldsymbol{\phi}(\boldsymbol{y}_m)$ in the feature space, and $\beta_{s,m}$ is a scale variable of the m^{th} basis vector.

By using this approximation, inner products between the vector $\boldsymbol{x}_t$ and the parameter vector $\boldsymbol{\lambda}_s$ can be denoted as a linear combination of the kernel function, as follows:

$$\boldsymbol{\phi}(\boldsymbol{x}_t)^{\mathsf{T}}\boldsymbol{\lambda}_s = \sum_m \beta_{s,m}K(\boldsymbol{x}_t, \boldsymbol{y}_m). \tag{4}$$

From the representer theorem [10], the approximation is strict when $M = T$ and $\boldsymbol{y}_t = \boldsymbol{x}_t$. This is the essential reason why a kernel-based method without any approximation requires $O(T^2)$ computational time at least. In this case, each likelihood computation requires $O(T)$ computational time, and optimization of the conventional training criteria requires at least T times of likelihood computation. In this study, we attempt to represent the subspace (Eq. (3)) by using a limited number of basis vectors, i.e. $M < T$, to make the likelihood computation as fast as $O(M)$.

By plugging the model definition (Eq. (2)) and the approximation (Eq. (3)) into the training criterion (Eq. (1)), the following training objective function is obtained.

$$\begin{aligned} F(B;Y) \overset{\text{def}}{=} &\sum_t \log \frac{\exp\left\{\sum_m \beta_{l_t,m}K(\boldsymbol{x}_t, \boldsymbol{y}_m)\right\}}{\sum_{s'} \exp\left\{\sum_m \beta_{s',m}K(\boldsymbol{x}_t, \boldsymbol{y}_m)\right\}} \\ &- c\sum_s \sum_{m,m'} \beta_{s,m}\beta_{s,m'}K(\boldsymbol{y}_m, \boldsymbol{y}_{m'}), \end{aligned} \tag{5}$$

where $B \overset{\text{def}}{=} \{\beta_{s,m}|\forall s, \forall m\}$, $Y \overset{\text{def}}{=} \{\boldsymbol{y}_m|\forall m\}$. Note that, in this objective function, B is used as a parameter variable to be optimized instead of Λ because the parameter space is restricted to the subspace denoted by Y. The optimization of B with fixed Y can be achieved by using standard optimization methods, such as the Newton-Raphson method and the Rprop method [11]. However, the identification of an essential subspace Y is a non-trivial problem.

3. GRADIENT DESCENT SUBSPACE PURSUIT METHOD

In principle, the best basis vector set $\hat{Y}$ satisfies the following equation:

$$\hat{Y} = \operatorname*{argmax}_{Y} \max_{B} F(B;Y). \tag{6}$$

If the number of basis vectors is known, the joint optimization of B and Y can be performed by using the back-propagation algorithm. However, this joint optimization is non-convex even if F is a convex function with respect to B. In this study, this non-convexity is avoided by appending the basis vectors gradually as required to continue the B-optimization. Because Y denotes the basis vectors used to approximate the true parameter vectors, the condition in Eq. (6) can always be satisfied if the number of basis vectors $|Y|$ is sufficient, i.e., $|Y| \geq \min\{T, D\}$ where D is the dimensionality of $\boldsymbol{\phi}(.)$ [10]. From this perspective, we can omit the non-convex optimization of the basis vectors by avoiding joint optimization and appending basis vectors so that we achieve the maximum of the objective function (Eq. (6)). However, because the number of basis vectors might be very large without any optimization (cf. the Nyström method), the CPSP method and the method proposed in this paper use basis selection criteria to obtain a small set of basis vectors.

One strategy for selecting an efficient set of basis vectors that can effectively maximize the objective function is to provide a basis vector that is directed to the global maximum. If the main objective function is a convex function, the global maximum must be in the direction of the gradient vectors in the feature space. Thus, basis vectors corresponding to the gradient vector are needed to achieve the global maximum after the optimization converges in a certain subspace. With the proposed method, these basis vectors are appended to the subspace variable to expand the subspace, and then the parameter variable B is optimized after each subspace expansion.

Here, by denoting the number of basis vectors to be added as E and the newly adding basis vectors as $\mathcal{Y} = \{\boldsymbol{y}_m|m \in [M + 1..M + E]\}$, the basis vectors $\hat{\mathcal{Y}}$ to be added to the current solution (Y, B) are obtained by minimizing the sum of the distances between the true gradient vector and a gradient vector approximated by using $\boldsymbol{\phi}(.)$, as follows:

$$\hat{\mathcal{Y}} = \operatorname*{argmin}_{\mathcal{Y}} \sum_s \underbrace{\min_{\gamma_s} \left|\left| \Delta_{\boldsymbol{\lambda}_s}F(\Lambda) - \sum_{m=M+1}^{M+E} \gamma_{s,m}\boldsymbol{\phi}(\boldsymbol{y}_m) \right|\right|^2}_{\mathcal{E}_s(\mathcal{Y})}, \tag{7}$$

where Λ is the current estimate of the parameters, $\gamma_{s,m}$ is a scale variable that is chosen freely so that $\mathcal{E}_s(\mathcal{Y})$ is minimized, and $\Delta_{\boldsymbol{\lambda}_s}F(\Lambda)$ is a gradient vector of the performance function $F(\Lambda)$ (not $F(B;Y)$) with respect to $\boldsymbol{\lambda}_s$.

To apply the kernel trick, we assume that the gradient vector of $F(\Lambda)$ has the following form:

$$\Delta_{\boldsymbol{\lambda}_s}F(\Lambda) \overset{\text{def}}{=} \sum_t \alpha_{s,t}\boldsymbol{\phi}(\boldsymbol{x}_t). \tag{8}$$

In fact, most conventional training criteria (e.g., MMI, minimum classification error (MCE), and minimum phoneme error (MPE)) of

Algorithm 1 Subspace pursuit method

```
Input: $M, \hat{M}, E$ /* The initial/required/increase # of basis vectors */
Input: $\boldsymbol{y}_m$ ($m \in [1..M]$) /* The initial value */
while $M < \hat{M}$ do
    Optimize $\beta_{s,m}$ ($m \in [1..M]$)
    Optimize $\boldsymbol{y}_m$ ($m \in [M+1..M+E]$)
    $M \leftarrow M + E$
end while
```

log-linear models have the above form (Eq. (8)) [1]. For example, with frame-wise MMI, the gradient vector can be expressed as follows:

$$\Delta_{\boldsymbol{\lambda}_s} F(\Lambda) = \sum_t \underbrace{(\delta(s,l_t) - P(s|\boldsymbol{x}_t,\Lambda))}_{\alpha_{s,t}} \boldsymbol{\phi}(\boldsymbol{x}_t), \tag{9}$$

where $\delta(s,l_t)$ is Kronecker's delta.

By substituting the assumption (Eq. (8)) into the basis vector identification criterion ($\mathcal{E}_s(\mathcal{Y})$), Eq. (7) can be expressed by using a kernel function, as follows:

$$\begin{aligned}\mathcal{E}_s(\mathcal{Y}) = &\sum_{m=M+1}^{M+E} \sum_{m'=M+1}^{M+E} \gamma_{s,m}\gamma_{s,m'} K(\boldsymbol{y}_m, \boldsymbol{y}_{m'}) \\ &- 2\sum_{m=M+1}^{M+E} \sum_t \alpha_{s,t}\gamma_{s,m} K(\boldsymbol{y}_m, \boldsymbol{x}_t) + \text{constant}.\end{aligned} \tag{10}$$

By using this expression, the direct evaluation of $\boldsymbol{\phi}(.)$ is omitted, and the use of enormously high-dimensional features is enabled.

The analytic solution of $\gamma_{s,m}$ that minimizes $\mathcal{E}_s$ can be obtained when the basis vector set $\mathcal{Y}$ is given. The gradient of $\mathcal{E}_s$ with respect to the vector $\boldsymbol{\gamma}_s$, where the m^{th} element is $\gamma_{s,m}$, can be expressed as $\Delta_{\boldsymbol{\gamma_s}} \mathcal{E}_s(\mathcal{Y}) = 2\mathbf{G}\boldsymbol{\gamma}_s - 2\sum_t \alpha_{s,t}\boldsymbol{k}_t$ where $\mathbf{G}$ is a matrix whose $(m,m')^{\text{th}}$ element is $K(\boldsymbol{y}_m, \boldsymbol{y}_{m'})$, and $\boldsymbol{k}_t$ is a vector whose m^{th} element is $K(\boldsymbol{y}_m, \boldsymbol{x}_t)$. Setting this gradient vector at $\mathbf{0}$ yields the following analytical solution:

$$\hat{\boldsymbol{\gamma}}_s = \mathbf{G}^{-1} \sum_t \alpha_{s,t}\boldsymbol{k}_t \tag{11}$$

By plugging this equation into the criterion $\mathcal{E}_s(\mathcal{Y})$, the gradient of $\mathcal{E}_s$ with respect to $\boldsymbol{y}_m$ used for $\mathcal{Y}$-optimization is computed as follows:

$$\begin{aligned}\Delta_{y_m}\mathcal{E}_s(\mathcal{Y}) = &\hat{\gamma}_{s,m}^2 \Delta_{\boldsymbol{y_m}} K(\boldsymbol{y}_m, \boldsymbol{y}_m) \\ &+ \sum_{m' \neq m} \hat{\gamma}_{s,m}\hat{\gamma}_{s,m'} \Delta_{\boldsymbol{y_m}} K(\boldsymbol{y}_{m'}, \boldsymbol{y}_m) \\ &- 2\sum_t \hat{\gamma}_{s,m}\alpha_{s,t} \Delta_{\boldsymbol{y_m}} K(\boldsymbol{x}_t, \boldsymbol{y}_m).\end{aligned} \tag{12}$$

This gradient vector can be calculated in computational time $O(TES)$.

Algorithm 1 is derived by inserting this basis selection during the main optimization of B. In the algorithm, $\mathcal{Y}$-optimization and B-optimization are performed alternatively. In the following experiments, we implemented each optimization by using the Rprop algorithm [11].

It should be noted that the use of the modified-MMI criterion [12] with a certain configuration produces the equivalent criterion of the CPSP method because $\alpha_{s,t}$ in Eq. (8) becomes equal to the variable used to represent the cutting planes in the CPSP method. Thus, we can consider that the proposed method is a straightforward generalization of the original CPSP method.

4. EXPERIMENTS

Speech recognition experiments were carried out to evaluate the efficiency of the proposed method by performing a dimensionality reduction of the second order features of Mel-frequency cepstral coefficients (MFCCs), expressed as follows:

$$\phi(\boldsymbol{x}) \overset{\text{def}}{=} \left\{x_i^2, \sqrt{2}x_i x_j, \sqrt{2}x_i, 1 | \forall i, \forall j \neq i\right\}, \tag{13}$$

where x_i denotes the i^{th} component of an observation vector. Observation vectors consist of 12 dimensional MFCCs and the log-energy augmented by their first/ second order time-derivatives (total: 39 dimensions). The input MFCC features are whitened by subtracting the global mean vector and multiplying the global inverse-covariance matrix obtained from the training dataset. The number of dimensions of $\boldsymbol{\phi}(\boldsymbol{x})$ is 819. Because this number is not very high, we can also apply conventional dimensionality reduction techniques to these features directly for comparison. In this feature setting, compared with the direct computation of $\boldsymbol{\phi}(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{y})$, we can efficiently compute the inner-product between two feature vectors by using the following kernel function:

$$K(\boldsymbol{x}, \boldsymbol{y}) \overset{\text{def}}{=} (\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} + 1)^2. \tag{14}$$

We implemented the following two algorithms as conventional feature reduction techniques.

- The **LDA** method reduces the dimensions of D-dimensional vectors by using an $(M \times D)$-dimensional matrix obtained by maximizing Fisher's discriminant criterion [6]. In these experiments, because the training criterion is convex, the performance of the LDA features exactly converges to that of the expanded second order features.
- The **relief** method estimates the importance of the features by employing local metrics obtained using nearest neighbour samples [13]. In this paper, we reduce dimensionality by using only M features with the highest importance. To improve the performance, we modified the estimated feature selection rule so that the first order features (x_i) are always included. Note that this method is a feature selection method unlike the LDA method and the proposed method.

We also implemented the following system to evaluate log-linear models without any dimensionality reduction technique.

- Log-linear models with explicitly expanded **second order** features.

Furthermore, as reference results, we also considered the following three methods in the experiments:

- Log-linear models with **first order** features, i.e. $\boldsymbol{\phi}(\boldsymbol{x}) \overset{\text{def}}{=} \boldsymbol{x}$.
- **Gaussian mixture models (GMM)** with diagonal covariance matrices trained by using the **MCE** criterion (presented in [14]).
- **Gaussian mixture models (GMM)** with full covariance matrices trained by using the **MMI** criterion (presented in [15]).

We used the TIMIT dataset for the experiments. The dataset consists of 3,696 utterances (1,124,823 frames) for training, and 192 utterances (57,919 frames) for testing. As described in [16], we used 48 phonetic classes for training and decoding, and we calculated the phoneme error rates by using 39 broader phonetic categories. All HMMs have 3 left-to-right states for each 48 monophone model. A bi-gram (bi-phoneme) grammar model is employed during all decoding processes. The scale factor for this grammar model and the hyperparameter c (Eq. (1)) are tuned in advance to maximize the performance of the second order system. The initial number of the basis vectors (M in Algorithm 1) is set at 78, and the increased number of the basis vectors (E in Algorithm 1) is set at 16. The initial

[1] Note that this method can also be used with non-convex criteria such as MCE although the first motivation is to maintain convexity.

basis vector set is set at $\{\boldsymbol{\phi}(\boldsymbol{e}_m), \boldsymbol{\phi}(-\boldsymbol{e}_m) | \forall m \in [1..39]\}$ where the m^{th} element of the unit vector $\boldsymbol{e}_m$ is 1.

Figure 1 shows the phoneme error rates of the compared methods obtained by varying the number of basis vectors. We confirmed that all the compared methods interpolated the results of the first order features and the second order features. The results of log-linear models with second order features were better than conventional GMM results. This advantage might be attributed to the convexity of the training method and the efficiency of the log-linear models with second order features. Further, we confirmed that the LDA technique was not very efficient in the experiments. We consider that this was due to the non-Guassianity of the distribution of the second order features. The *relief* feature selection method worked efficiently even if the method only performed the binary feature selection. The proposed method worked effectively even when such non-Gaussian features were used. In fact, the proposed method achieved the comparable performance with the full second order feature system by only using about 600 basis vectors. Although the differences are not significant, the performance of the proposed method was also superior to that of the full second order system when the number of basis vectors M was in the $[606..798]$ range. This advantage might be due to de-noising in the feature space.

We consider the proposed method to have two advantages. The first is that the proposed method can directly optimize the objective function used for parameter training unlike the LDA system, which maximizes Fisher's discriminant criterion. The second advantage is that the proposed method can maintain the convexity of the original problem unlike the perceptron approach. The experimental results suggest the first advantage. Although the second advantage is proved analytically by the formulation described in this paper, the experimental verification of the second advantage will be our future work.

In the experiments, because the number of second order features is not very high, the second order system without dimensionality reduction is the most efficient in terms of training time. However, if we use the naive kernel method to handle these second order features, the kernel function has to be computed at least $T^2 \simeq 1.27 \times 10^{12}$ times. In the experiments, if we stopped the optimization at $M = 606$ the count of the kernel function computations was $(M+OE)T \simeq 4.52 \times 10^{10}$ where O is the total number of iterations for $\mathcal{Y}$-optimization. Furthermore, the proposed method will perform well when kernel functions that produce infinite dimensional features are used.

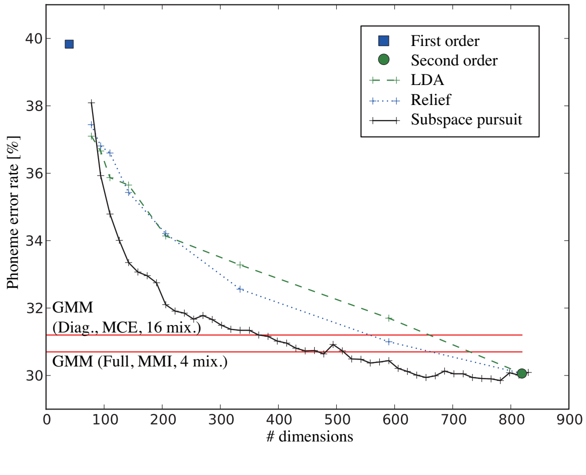

Fig. 1. Phoneme error rates as functions of the number of dimensions.

5. CONCLUSIONS

In this paper, we proposed a general framework for approximating log-linear models in kernel spaces. With the proposed method, the subspace of a kernel space spanned with a limited number of basis vectors is iteratively expanded and used to represent parameter vectors in the kernel space. Since the proposed method uses a limited number of basis vectors, the training method does not require $O(T^2)$ computational time, where T is the number of vectors in the training dataset. The experimental results show that the proposed method outperformed the conventional dimensionality reduction techniques.

In the future, we intend to evaluate the proposed method by using more diverse kernel functions. For example, extensions to indifferentiable structural kernels, such as dynamic time-alignment kernels [4], are promising because the use of these kernels is essential in natural language processing fields.

Acknowledgement This study was partially supported by a Grant-in-Aid for JSPS Fellows (21·04190) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

6. REFERENCES

[1] S. Wiesler, M. Nußbaum-Thom, G. Heigold, R. Schlueter, and H. Ney, "Investigations on features for log-linear acoustic models in continuous speech recognition," in *Proc. IEEE Workshop on ASRU*, Merano, Italy, 2009.

[2] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, Philadelphia, PA, USA, 2005.

[3] Y. Kubo, S. Watanabe, A. Nakamura, E. McDermott, and T. Kobayashi, "A sequential pattern classifier based on hidden Markov kernel machine and its application to phoneme classification," *IEEE J. Sel. Topics Signal Process.*, vol. 4, pp. 974–984, 2010.

[4] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama, "Dynamic time-alignment kernel in support vector machine," *Advances in Neural Information Processing Systems*, vol. 2, pp. 921–928, 2002.

[5] S. Mika, B. Schölkopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," *Advances in Neural Information Processing Systems*, vol. 11, no. 1, pp. 536–542, 1999.

[6] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comp.*, vol. 12, no. 10, pp. 2385–2404, 2000.

[7] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," *Advances in Neural Information Processing Systems*, vol. 13, pp. 682–699, 2001.

[8] T. Joachims and C.-N. J. Yu, "Sparse kernel SVMs via cutting-plane training," *Machine Learning Journal*, vol. 76, no. 2-3, pp. 179–193, 2009.

[9] G. Heigold, S. Wiesler, M. Nußbaum-Thom, P. Lehnen, R. Schlueter, and H. Ney, "Discriminative HMMs, log-linear models, and CRFs: what is the difference?" in *Proc. ICASSP*, Dallas, TX, USA, 2010.

[10] B. Schölkopf and A. Smola, *Learning with Kernels*. MIT Press, 2002.

[11] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. IEEE International Conference on Neural Networks*, San Francisco, CA, USA, 1993, pp. 586–591.

[12] G. Heigold, T. Deselaers, R. Schlueter, and H. Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc. ICML*, Helsinki, Finland, 2008.

[13] Z. Deng, F.-L. Chung, and S. Wang, "Robust relief feature weighting, margin maximization and fuzzy optimization," *Fuzz. Syst., IEEE Trans. on*, vol. 18, no. 4, pp. 726–744, 2010.

[14] E. McDermott and S. Katagiri, "String-level MCE for continuous phoneme recognition," in *Proc. EUROSPEECH*, Rhodes, Greece, 1997, pp. 123–126.

[15] S. Kapadia, V. Valtchev, and S. J. Young, "MMI training for continuous phoneme recognition on the TIMIT database," in *Proc. ICASSP*, vol. 2, Orlando, FL, USA, 2002, pp. 491–494.

[16] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *Acoust., Speech, Signal Process., IEEE Trans. on*, vol. 37, no. 11, pp. 1641–1648, 1989.