BASIS VECTOR ORTHOGONALIZATION FOR AN IMPROVED KERNEL GRADIENT MATCHING PURSUIT METHOD

Yotaro Kubo, Shinji Watanabe[1]*, Atsushi Nakamura*

NTT Communication Science Labs.
Kyoto 619-0237, Japan

Simon Wiesler, Ralf Schlueter, Hermann Ney

RWTH Aachen University
Aachen 52062, Germany

ABSTRACT

With the aim of achieving a computationally efficient optimization of kernel-based probabilistic models for various problems, such as sequential pattern recognition, we have already developed the kernel gradient matching pursuit method as an approximation technique for kernel-based classification. The conventional kernel gradient matching pursuit method approximates the optimal parameter vector by using a linear combination of a small number of basis vectors. In this paper, we propose an improved kernel gradient matching pursuit method that introduces orthogonality constraints to the obtained basis vector set. We verified the efficiency of the proposed method by conducting recognition experiments based on handwritten image datasets and speech datasets. We realized a scalable kernel optimization that incorporated various models, handled very high-dimensional features ($>$ 100 K features), and enabled the use of large scale datasets ($>$ 10 M samples).

Index Terms— Kernel methods, hidden Markov models, orthogonal expansion, speech recognition

1. INTRODUCTION

Kernel methods are promising for handling nonlinear distortion problems in pattern recognition; however, due to their computational costs, the application fields for kernel methods remain limited. When kernel methods are employed, the computational time required for parameter estimation becomes at least $O(N^2)$, where N is the number of training samples. In some application fields, such as speech recognition, the approach is computationally prohibitive since N exceeds 10 million and continues to grow in common speech recognition tasks. Therefore, an efficient approximation method for kernel methods are necessary to enable kernel-based representation in a large scale task.

Computationally efficient approximation frameworks for kernel methods have been well studied in the last decade, e.g. [1], and some of these approaches achieved a computational time that was almost linear in relation to the number of samples used for training. However, since these approaches are only designed for specific model training criteria and for specific kinds of discriminant functions, it is not easy to apply them to arbitrary methods of model training. For example, they cannot be applied straight forwardly to latent models. Thus, if we are to enlarge the application fields of kernel methods, we must realize a flexible framework for approximating the kernel methods that can incorporate various models and training criteria with large scale datasets.

A flexible way to approximate kernel methods is to employ the Nyström method [2] that randomly selects a limited number of basis vectors from a given training dataset where the conventional kernel methods use all the training samples as basis vectors. However, the Nyström method cannot select appropriate basis vectors when the number of vectors is reduced to realize computational efficiency because the method is based on random selection. Alternative approaches, which involve dimensionality reduction in feature spaces, such as kernel principal component analysis [3] and kernel Fisher discriminant analysis [4], can identify efficient basis vectors; however, these kernel-based dimensionality reduction methods require $O(N^3)$ computational time for pre-processing.

Recently, novel methods based on addition of efficient basis vectors, rather than selection/ limitation, have been proposed [5–7]. In contrast to the dimensionality reduction methods these methods search efficient basis vectors and increase the dimensionality by adding these vectors iteratively. For example, the kernel matching pursuit (KMP) method [5] selects basis vectors from a given training dataset to approximate a solution by employing the matching pursuit theory [8], which greedily selects a basis vector that maximizes the projection of the optimal parameter vector. The cutting plane subspace pursuit (CPSP) method [6] extends the cutting plane-based training of support vector machines (SVMs) [9] by incorporating pre-image optimization that finds $\boldsymbol{y} \in \mathbb{R}^D$ such that $\boldsymbol{\phi}(\boldsymbol{y})$ represents an essential basis vector, where $\boldsymbol{\phi}$ is a feature mapping function implicitly defined by a given kernel function. The kernel gradient matching pursuit (KGMP) method [7], which is briefly described in the following section, generalizes the CPSP method by incorporating pre-image optimization to find pre-image vectors that approximate gradient vectors of a performance function of model training. Although the strategy suggested by the KGMP method is promising for achieving the above objectives, as described in the following sections, the KGMP method is inefficient in some cases.

In this paper, in order to increase dimensionality efficiently, we extend the KGMP method by introducing an efficient criterion that considers the orthogonality of the basis vectors as well as the improvement of a given performance function. This paper applied the proposed method for the various datasets (MNIST handwritten digits, TIMIT, and WSJ). These applications exceeded the scales of the state-of-the-art applications of the kernel methods in terms of the number of samples ($>$ 10 M samples). Moreover, these applications exceeded the scales of the conventional ASR in terms of the dimensionality of the feature vectors we used ($>$ 100 K features).

2. KERNEL GRADIENT MATCHING PURSUIT

In this section, the underlying theory behind the KGMP method is first briefly described. Then, this theory is applied to realize kernel-log-linear models. Both the proposed method and the KGMP method are based on the theory described in this section, which involves iterative dimensionality expansion performed by appending

[1] The author is currently with Mitsubishi Electric Research Laboratories.

Copyright 2012 IEEE. Published in the IEEE 2012 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), scheduled for 25-30 March 2012 in Kyoto, Japan. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

efficient basis vectors during a gradient ascent optimization process.

2.1. General Formulation

The KGMP method is an optimization method that assumes the following condition in derivative vectors, with respect to a parameter vector $\boldsymbol{\lambda}$, of the given objective function $F(\boldsymbol{\lambda})$.

$$\nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) = \sum_{n=1}^{N} d_n(\boldsymbol{\lambda}) \boldsymbol{\phi}(\boldsymbol{x}_n) - c\boldsymbol{\lambda}, \tag{1}$$

where $\boldsymbol{\phi}$ is a feature mapping function that transforms low-dimensional observation vectors into very high-dimensional feature vectors, $d_n(\boldsymbol{\lambda})$ is the weight of each training sample depending on the choice of the performance function F, and c is a regularization constant. Most of the training criteria of kernel linear models with L2-regularization terms have this form.

We assume that the parameter vector is represented by a small number, denoted by M, of basis vectors $\boldsymbol{\phi}(\boldsymbol{y}_m)$ $(m \in \{1, .., M\})$ that have pre-image vectors $\boldsymbol{y}_m$ in the observation space $(\mathbb{R}^D)$, as follows:

$$\boldsymbol{\lambda} = \sum_{m=1}^{M} \beta_m \boldsymbol{\phi}(\boldsymbol{y}_m). \tag{2}$$

To identify a small number $M \ll N$ of basis vectors $\boldsymbol{y}_m$ that lead to the improvement of performance function F, a kernel matching pursuit algorithm [5] and pre-image optimization are introduced. As in [5], a new basis vector that realizes a good approximation of the gradient vector $\nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda})$ is appended during the optimization of β_m in Eq. (2). However, a new basis vector is chosen not only from the training dataset, but also from the entire observation space by solving a pre-image problem. The new basis vector to be appended can be obtained by solving the following pre-image optimization problem:

$$\hat{Y} = \underset{Y}{\operatorname{argmin}} \min_{\beta_{M+1}, \cdots, \beta_{M+R}} \left\| \nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) - \sum_{m=M+1}^{M+R} \beta_m \boldsymbol{\phi}(\boldsymbol{y}_m) \right\|^2, \tag{3}$$

where $Y \stackrel{\text{def}}{=} \{\boldsymbol{y}_{M+1}, \boldsymbol{y}_{M+2}, \cdots \boldsymbol{y}_{M+R}\}$. Although the above optimization problem is not analytically solvable in general, the numerical approach is still acceptable if the kernel function is differentiable. In contrast to the approximation method based on data selection, such as KMP [5], solving this numerical optimization is important for both approximation accuracy and computational efficiency because finding the best Y from the entire vector space $\mathbb{R}^D$ is more computationally efficient than finding the best Y from the given training samples when the number of training samples is large. Unfortunately, this pre-image optimization is no longer convex. However, since this optimization is only used to increase the dimensionality of the subspace, the use of the basis vectors obtained by solving this optimization always contributes to improvements in the main objective function F, even though the pre-image optimization (Eq. (3)) converges to a local optimum.

2.2. Application to Kernel-Log-Linear Models

As an example, we employ a kernel-log-linear model that defines the conditional probability of a label $l \in \mathcal{L}$, given observation vector $\boldsymbol{x} \in \mathbb{R}^D$, by using a parameter vector $\boldsymbol{\lambda} = \left[\boldsymbol{\theta}_1^{\mathsf{T}}, \cdots, \boldsymbol{\theta}_l^{\mathsf{T}}, \cdots\right]^{\mathsf{T}}$, as follows:

$$P(l|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{\exp\left\{\sum_{m=1}^{M} \beta_{l,m} K(\boldsymbol{y}_m, \boldsymbol{x})\right\}}{\sum_{l'} \exp\left\{\sum_{m=1}^{M} \beta_{l',m} K(\boldsymbol{y}_m, \boldsymbol{x})\right\}}. \tag{4}$$

It should be noted that we changed the suffix of β to the class variable l and the basis vector variable m to share the same basis vector set among all classes. By employing a zero-mean Gaussian prior probability density function (pdf) of the parameter vector and the training dataset $\mathcal{D} = \{(\boldsymbol{x}_1, l_1), \cdots (\boldsymbol{x}_N, l_N)\}$, the maximum-a-posteriori (MAP) estimation of the above kernel-log-linear model is defined as follows:

$$F(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \left(\boldsymbol{\beta}_{l_n}^{\mathsf{T}} \boldsymbol{k}_n - \log \sum_{l'} \exp \boldsymbol{\beta}_{l'}^{\mathsf{T}} \boldsymbol{k}_n \right) - \frac{c}{2} \sum_{l} \boldsymbol{\beta}_l^{\mathsf{T}} \mathbf{G} \boldsymbol{\beta}_l, \tag{5}$$

where $\boldsymbol{\beta}_l$ denotes a coefficient vector, $\boldsymbol{\beta}_l \stackrel{\text{def}}{=} [\beta_{l,1}, \beta_{l,2}, \cdots]^{\mathsf{T}}$, $\boldsymbol{k}_n$ is a projection of the t^{th} training vector, $\boldsymbol{k}_n = [K(\boldsymbol{x}_n, \boldsymbol{y}_1), K(\boldsymbol{x}_n, \boldsymbol{y}_2), \cdots]$, and the $(i, j)^{\text{th}}$ element in the gram matrix $\mathbf{G}$ is $g_{i,j} = K(\boldsymbol{y}_i, \boldsymbol{y}_j)$.

This example yields a gradient vector $\nabla_{\boldsymbol{\theta}_l} F(\boldsymbol{\lambda})$ as follows:

$$\nabla_{\boldsymbol{\theta}_l} F(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \underbrace{\left(\delta(l, l_n) - P(l|\boldsymbol{x}_n, \boldsymbol{\lambda})\right)}_{d_{n,l}(\boldsymbol{\lambda})} \boldsymbol{\phi}(\boldsymbol{x}_n) - c\boldsymbol{\theta}_l. \tag{6}$$

As shown in the above equation, the gradient vector of the MAP estimation of the presented kernel-log-linear models can be expressed in the form defined in Eq. (1). Thus, incremental subspace expansion is accomplished by solving optimization Eq. (3).

3. ORTHOGONAL KGMP

In this paper, we propose an improved method for resolving the inefficiency caused by redundant basis vectors by explicitly introducing (nearly) orthogonal constraints to obtain a more accurate basis set. We focus on *near* orthogonality because completely orthogonal basis vectors cannot be expressed by the pre-image form $\boldsymbol{\phi}(\boldsymbol{y})$ with some kernels, e.g. the Gaussian kernel.

As with the algorithm in the previous section, we consider the incremental expansion of a basis vector set. Thus, we consider that a nearly orthogonal basis vector set $\boldsymbol{\phi}(\boldsymbol{y}_m)$ $(m \leq M)$ is already obtained, and aim at finding another basis vector $\boldsymbol{\phi}(\boldsymbol{y}_{M+1})$ that realizes orthogonality as well as a good approximation of the gradient vector. In this section, we introduce the orthogonalized gradient vector $\hat{\nabla}_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda})$, which is orthogonal to the existing basis vectors, and substitute the original gradient vector $\nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda})$ in Eq. (3) with the orthogonalized gradient vector. The orthogonal gradient vector is constructed by subtracting components that can be represented as a linear combination of the existing basis vectors. By using this orthogonal gradient vector instead of the original gradient vector as the approximation target, the basis expansion process is modified to find a nearly orthogonal basis vector set.

To obtain the orthogonal gradient vector, we adapted the Gram-Schmidt orthonormalization method to kernel-based approaches. The Gram-Schmidt orthonormalization of the existing basis vectors is performed by finding an orthonormalization matrix $\mathbf{Q} = \{q_{i,j} | i, j \in \{1, .., M\}\} \in \mathbb{R}^{M \times M}$ that leads vectors $\boldsymbol{b}_i = \sum_{j=1}^{M} q_{i,j} \boldsymbol{\phi}(\boldsymbol{y}_j) (i \in \{1, .., M\})$ to form an orthogonal basis vector set. The vector $\boldsymbol{q}_i$ corresponding to the i^{th} row in the matrix $\mathbf{Q}$ can be computed by the following recursive procedure:

$$\boldsymbol{q}_i = \frac{1}{\sqrt{\tilde{\boldsymbol{q}}_i^{\mathsf{T}} \mathbf{G} \tilde{\boldsymbol{q}}_i}} \tilde{\boldsymbol{q}}_i, \text{ where } \tilde{\boldsymbol{q}}_i = \boldsymbol{e}_i - \sum_{j=1}^{i-1} \boldsymbol{e}_i^{\mathsf{T}} \mathbf{G} \boldsymbol{q}_j \boldsymbol{q}_j, \tag{7}$$

Algorithm 1 Orthogonal Kernel Gradient Matching Pursuit

$M \leftarrow M^{\mathrm{init}}, Y \leftarrow \mathrm{RandomSample}(\{\boldsymbol{x}_n | \forall t\}, M^{\mathrm{init}})$
while $M < \hat{M}$ **do**
  Optimize $\beta_{l,m}$ $(m \in \{1, .., M\})$; Compute $\nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda})$
  for $r = 1$ to R **do**
    Compute $\mathbf{Q}$ from the given $\nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda})$ and Y (Eq. (7))
    Optimize $\hat{\boldsymbol{y}}_{M+1} = \underset{\boldsymbol{y}_{M+1}}{\mathrm{argmin}} \| \hat{\boldsymbol{\nabla}}_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) - \boldsymbol{\phi}(\boldsymbol{y}_{M+1}) \|^2$
    $Y \leftarrow Y \cup \{\hat{\boldsymbol{y}}_{M+1}\}$; $M \leftarrow M + 1$
  end for
end while

where the i^{th} element of the unit vector $\boldsymbol{e}_i$ is 1.

By using this orthonormalization matrix $\mathbf{Q}$ and the obtained orthonormalized basis vectors $\boldsymbol{b}_m$, the orthogonalized gradient vector can be derived by subtracting all basis vectors $\boldsymbol{b}_m$, as follows:

$$\hat{\nabla}_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) - \boldsymbol{\Phi}^{\mathsf{T}} \mathbf{Q}^{\mathsf{T}} \mathbf{Q} \left(\sum_{n=1}^{N+M} \alpha_n(\boldsymbol{\lambda}) \boldsymbol{k}_n \right), \quad (8)$$

where the definition of the n^{th} projection vector $\boldsymbol{k}_n$ is the same as that in Eq. (5), and the m^{th} row of the matrix $\boldsymbol{\Phi}$ is $\boldsymbol{\phi}(\boldsymbol{y}_m)$.

Thanks to the orthogonality constraint, we can obtain an approximation of the solution of the optimization in Eq. (3) by employing a greedy strategy, without solving the complex simultaneous optimization directly. The greedy strategy we used to approximate the orthogonalized gradient vector is also based on the matching pursuit approach. First, the gradient vector is approximated by using only one pre-image basis vector $\boldsymbol{\phi}(\boldsymbol{y}_{M+1})$; and then, the remaining basis vectors are obtained from the gradient vector that is orthogonalized from the obtained basis vector $\boldsymbol{\phi}(\boldsymbol{y}_{M+1})$. This iterative orthogonalization is performed by updating orthogonalization matrix $\mathbf{Q}$, iteratively. Algorithm 1 is an example of the specific algorithm used in the experimental section. In the algorithm, we avoid multiple basis optimization, but attempt to reconstruct a gradient vector that is orthogonalized by all previously obtained basis vectors. In the experimental section, we call this algorithm "Orthogonal KGMP."

4. EXPERIMENTS

In this section, we evaluate the efficiency of the proposed method by conducting recognition experiments on image and speech datasets

4.1. Handwritten digit classification

To evaluate the basic performance of the proposed method, we carried out handwritten digit classification experiments as preliminary experiments. In the experiments, we used the MNIST handwritten digit dataset [10]. We designed a binary classification task and a multiclass classification task. In the binary classification task, models are trained to classify digit images into two classes, "0"/ "1"/ "2"/ "3"/ "4" vs "5"/ "6"/ "7"/ "8"/ "9", and, in the multiclass task, models are trained to predict the posterior probability of the corresponding digit. We used the first 50,000 images in the training dataset for parameter optimization, the remaining 10,000 images in the training dataset for validation, and all 10,000 images in the test dataset for evaluation. The observation vector consisted of the intensities of 28 $\times$ 28 pixels.

The hyper-parameters of Algorithm 1 were set at $M^{\mathrm{init}} = 10$, $\hat{M} = 1000$, and $R = 10$. The remaining hyper-parameters (the regularization constant c and the γ variable in Gaussian kernels) were tuned by using 10,000 validation images. As a reference, we compared the prediction error rates with those of SVMs. For multiclass classification of SVMs, we prepared one-vs-rest SVMs.

Table 1. Prediction error rates of handwritten digit classification.

Method	binary	multiclass
Log-linear (linear)	13.7	8.2
Log-linear (Nyström)	6.0	5.4
Log-linear (KGMP)	3.3	4.1
Log-linear (Orthogonal KGMP)	1.9	2.5
SVM (linear)	12.3	7.9
SVM (CPSP)	1.8	3.2

Table 1 shows the prediction error rates of the digit classification tasks. We confirmed that using nonlinearity via kernel methods led to better performance. Moreover, since KGMP-based basis selection outperformed the random basis selection method (the Nystöm method), we could confirm that the strategy of KGMP, which reconstructs gradient vectors incrementally, successfully captured an efficient representation of high-dimensional parameter vectors. Although the generalization performance of the log-linear models appeared worse than that of linear SVMs, this discrepancy becomes insignificant if we introduce the orthogonal KGMP method. We considered that this disadvantage of conventional KGMP is mainly due to the redundancy between the basis vectors that are added simultaneously by optimizing Eq. (3). In the multiclass experiments, the orthogonal KGMP outperformed the one-vs-rest support vector machines even though the same number of basis vectors were used. We considered this to be due to the advantage gained by direct multiclass formulation. Since the proposed method is based on the general gradient ascent-based formulation, we can directly apply the proposed method to various problems including multiclass problems.

We carried out benchmark tests to evaluate the computational time required by the proposed method. We prepared a naive implementation of kernel-log-linear models that uses all the training vectors as basis vectors for comparison. In the experiments, we performed the abovementioned binary classification task, and measured the computational time by varying the number of training samples. The basis expansion process in the proposed methods was iterated until the value of the objective function of the proposed method exceeded 95 percent of that of the naive method. Figure 1 shows average computational time (5 trials for each setting) as a function of the number of training data. We confirmed that the training time of the proposed method was almost linear with respect to the number of training samples although that of the naive implementation increased quadratically.

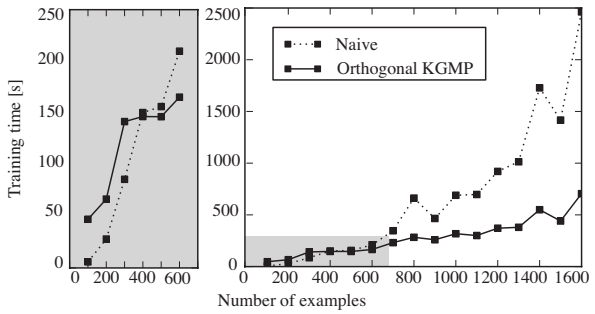

Fig. 1. Average training time as a function of the number of training samples (the left figure is a magnified view of the right figure).

Table 2. Phoneme error rates (PERs) of phoneme recognition.

Method	# basis	# dim. of ϕ	PER
GMM (MMIE, Full cov.) [11]	–	–	30.7
Linear	430	430	37.1
Second-order polynomial	750	101,475	27.9
Third-order polynomial	736	1,528,890	30.1
Gaussian	2560	∞	35.8

4.2. Speech applications

Following the first objective, to confirm the performance in terms of flexible usage, we performed phoneme recognition experiments based on HMMs. The HMMs we used is defined by substituting the emission pdf $p(\boldsymbol{x}_t|q_t)$ in the conventional HMM-based acoustic models for ASR, as $p(\boldsymbol{x}_t|q_t) \propto P(q_t|\boldsymbol{x}_t)/P(q_t)$, where $P(q_t|\boldsymbol{x}_t)$ was modeled by using the kernel-log-linear models, and $P(q_t)$ is the distribution of the HMM state, which was assumed to be a uniform distribution. $P(\boldsymbol{l})$ and $P(\boldsymbol{q}|\boldsymbol{l})$ were estimated by using the maximum likelihood procedure. The kernel-log-linear models $P(q_t|\boldsymbol{x}_t)$ were estimated using the MAP criterion (Eq. (5)).

We used the TIMIT dataset for the experiments. The dataset consists of 3,696 utterances (1,124,823 frames) for training, 1,144 utterances (350,343 frames) for validation, and 192 utterances (57,919 frames) for testing. All the speech signals were first converted into sequences of 39-dimensional MFCC-based feature vectors (as in [11]); and then all the vectors were augmented by concatenating the preceding and subsequent 5 frames to construct 429-dimensional observation vectors. As described in [12], we used 48 phonetic classes for training and decoding, and we calculated the phoneme error rates by using 39 broader phonetic categories. The scale factor for the language model, the number of iterations, and the regularization constant c were tuned by using the validation dataset. In general, the estimation of model parameters of speech recognition involves unsupervised training; i.e. we can observe $\boldsymbol{l}$, but not $\boldsymbol{q}$. However, in the experiments, we used state sequences obtained by using a conventional HMM-based system for the sake of simplicity. We emphasize that the proposed method can also be employed for such unsupervised training since its training procedure satisfies the introduced assumption (Eq. (1)).

Table 2 shows the phoneme error rates (PERs) of the compared methods. We confirmed that using the standard log-linear models results in poor performance when compared with the conventional HMM-based approach with Gaussian mixture model (GMM)-type emission pdfs. This is mainly because of the nonlinearity in the observation vectors. However, by introducing kernels, we confirmed that the proposed approach outperformed conventional HMMs, even though latent variables were not considered in the emission pdfs. The use of third-order polynomial kernels and Gaussian kernels was not so effective in the experiments. The ineffectiveness of third-order polynomial kernels might be due to overfitting. The ineffectiveness of the Gaussian kernels might be due to nuisance information in the observation vectors caused by preceding and subsequent frames. However, the performance of second-order kernels is satisfactory, especially if we focus on the number of parameters. The number of parameters in the second-order system is 429,894, which is comparable to that of the baseline HMM systems; however, the performance improvement is significant. This advantage might be attributed to the efficiency of the proposed method that successfully identified an efficient subspace of the feature space.

We conducted the preliminary experiments to verify an applicability of the proposed method to a large scale task. In the experiments, the kernel-log-linear models were trained with the WSJ SI284 dataset to classify the corresponding HMM states (2656 classes) from the given observation vectors. The observation vector we used is 12-MFCCs with log-energies spliced with 11 frames (143 dims.). The optimization is stopped when the % change in the objective function is less than 0.1. We used the second-order polynomial kernel for the experiments; therefore the dimensionality of the feature vector $\boldsymbol{\phi}(.)$ is 10440. The hyper-parameters are set at $M^{\text{init}} = 78$, $R = 13$ and $c = 0.0$. Table 3 shows the results of the optimization. We realized the kernel-based optimization over 74 h ($10^{7.4}$ samples) dataset in practical computational time. These results exceed scales of the state-of-the-art applications of kernel methods. We confirmed that the proposed method is also available for such a large scale task.

Table 3. Computational time on the WSJ corpus ($10^{7.4}$ samples).

Kernel	# basis vectors	# dim. of ϕ	Obj. func. / # frame	Train. time
linear	143	143	-3.23	4 h 17 m
second	286	10440	-2.82	19 h 19 m

5. CONCLUSION

Aiming at the flexible use of kernel methods in various models and with various optimization criteria, we proposed an orthogonal kernel gradient matching pursuit method. The proposed method increases dimensionality by appending basis vectors that approximate a high-dimensional gradient vector obtained during model training optimization. To minimize redundancy in the obtained basis set, we introduced orthogonalization into the gradient vector to be approximated. We evaluated the proposed method by carrying out handwritten digit classification experiments and HMM-based continuous phoneme recognition experiments. We confirmed that the proposed method enabled the efficient realization of kernel approaches in terms of both computational efficiency and modeling accuracy.

6. REFERENCES

[1] K. Crammer, J. Kandola, and Y. Singer, "Online classification on a budget," *Advances in NIPS*, 2004.

[2] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," *Advances in NIPS*, vol. 13, pp. 682–699, 2001.

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

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

[5] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Mach. Learn.*, vol. 48, no. 1, pp. 165–187, 2002.

[6] T. Joachims and C.-N. John Yu, "Sparse kernel SVMs via cutting-plane training," *Mach. Learn. J.*, vol. 76, no. 2-3, pp. 179–193, 2009.

[7] Y. Kubo, S. Wiesler, R. Schlueter, H. Ney, S. Watanabe, A. Nakamura, and T. Kobayashi, "Subspace pursuit method for kernel-log-linear models," in *Proc. ICASSP*, 2011, pp. 4500–4503.

[8] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[9] T. Joachims, "Training linear SVMs in linear time," in *Proc. KDD*. ACM, 2006, pp. 217–226.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

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