Derivative Kernels for Noise Robust ASR

A. Ragni and M. J. F. Gales

Cambridge University Engineering Department
Trumpington St., Cambridge CB2 1PZ, U.K.
{ar527,mjfg}@eng.cam.ac.uk

***Abstract*—Recently there has been interest in combined generative/discriminative classifiers. In these classifiers features for the discriminative models are derived from generative kernels. One advantage of using generative kernels is that systematic approaches exist how to introduce complex dependencies beyond conditional independence assumptions. Furthermore, by using generative kernels model-based compensation/adaptation techniques can be applied to make discriminative models robust to noise/speaker conditions. This paper extends previous work with combined generative/discriminative classifiers in several directions. First, it introduces derivative kernels based on context-dependent generative models. Second, it describes how derivative kernels can be incorporated in continuous discriminative models. Third, it addresses the issues associated with large number of classes and parameters when context-dependent models and high-dimensional features of derivative kernels are used. The approach is evaluated on two noise-corrupted tasks: small vocabulary AURORA 2 and medium-to-large vocabulary AURORA 4 task.**

I. Introduction

Most automatic speech recognition (ASR) systems use generative models, hidden Markov models (HMM), as the acoustic models. Likelihoods from these models are combined with the prior, the language model, using Bayes' rule to yield the sentence posterior. Although successful, it is widely known that the underlying models are not correct. This has lead to interest in discriminative classifiers which directly model sentence posteriors/decision boundaries given a set of features extracted from the observation sequence. There are several options how features can be extracted from observation sequences. This includes event detectors [1], generative kernels [2] and other parametric and non-parametric approaches [3]. Event detectors make use of multiple parallel feature streams which operate at different levels of granularity such as word, multi-phone and phone. This flexibility enables a wide range of short and long-spanning dependencies. However, the current applications of event detectors do not attempt to improve the underlying acoustic models, the recognition results from these models are used to derive features. Additionally, the issues associated with adapting feature streams to noise/speaker conditions are not easy to handle.

Generative kernels derive features from generative models and have several advantages. First, the use of competing log-likelihoods, first and higher order derivatives of log-likelihood offers a systematic approach of adding new acoustic features. In contrast to log-likelihoods the derivatives do not inherit conditional independence assumptions from generative models and enable other short and long-spanning dependencies. Second, the generative kernels can be adapted to noise/speaker conditions using model-based compensation/adaptation approaches [4]. Third, since generative kernels derive features from generative models the parameters of these models can be re-estimated to extract more discriminative features. Previous work with generative kernels has examined several feature configurations. The use of log-likelihood features extracted from whole-word and context-dependent HMMs was investigated in [5] and [6] respectively. However, the features in these approaches inherited the underlying HMM conditional independence assumptions. Derivative features have been examined in [4] and [7]. However, the generative models used in these approaches were whole-word models and small vocabulary recognition tasks were considered.

This paper extends the previous work with derivative features to handle medium/large vocabulary speech recognition tasks. This requires three fundamental issues to be addressed. The first issue is the large number of context-dependent discriminative classes. The approach based on phonetic decision tree clustering [6] to ensure that sufficient amount of training data exists for robust parameter estimation is adopted. With derivative features parameter tying introduces another issue. When more than one distinct generative model is used to extract features the discriminative parameters become sensitive to the order of components in these models. A simple approach is proposed where discriminative parameters associated with derivatives are tied within the states. The third issue is that large margin training should be used with high-dimensional features and limited amount of training data, however, current implementations [8] can not handle high-dimensional features. In this paper an on-the-fly variant of minimum Bayes' risk training is performed.

II. Combined generative and discriminative classifiers

Generative models are well known for their natural handling of variable length sequences, adaptability to varying noise/speaker conditions, efficient learning and inference algorithms. For discriminative classifiers these issues are not easy to handle. This section provides details on a combined approach which offers the benefits of generative models with the additional power of discriminative classifiers.

Consider a framework illustrated by Figure 1 where the shaded part corresponds to generative models and the rest to discriminative classifiers. The generative part is a standard model-based HMM compensation/adaptation framework. Given noise and speaker-dependent observation sequence $\mathbf{O}$

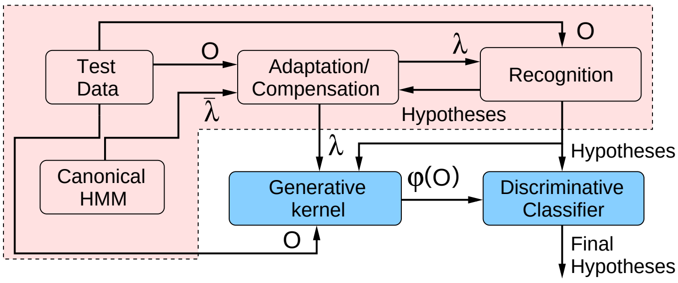

Fig. 1. Combined generative and discriminative framework

the parameters of the canonical HMMs are compensated to the target conditions using model-based techniques. The discriminative part makes use of these compensated HMMs and observation sequences to extract a set of features. These features handle the mapping from variable length sequences to a fixed dimension and incorporate a range of short and long-spanning dependencies. The advantage of this framework is that the features extracted from the compensated HMMs will be automatically adapted to target noise/speaker conditions. It is then possible to train noise/speaker-independent discriminative classifiers.

In this work vector Taylor series (VTS) model-based compensation is applied to map HMM parameters to target noise conditions. The first-order VTS scheme described in [9] is used. The mismatch function between the static part of clean $\mathbf{x}_t^{\mathsf{s}}$ and noise-corrupted $\mathbf{o}_t^{\mathsf{s}}$ observation is given by

$$\mathbf{o}_t^{\mathsf{s}} = \mathbf{x}_t^{\mathsf{s}} + \mathbf{h} + \mathbf{C}\log\left(\mathbf{1} + \exp\left(\mathbf{C}^{-1}\left(\mathbf{n}_t^{\mathsf{s}} - \mathbf{x}_t^{\mathsf{s}} - \mathbf{h}\right)\right)\right) \quad (1)$$

where $\mathbf{C}$ is a discrete cosine transformation matrix, $\mathbf{1}$ is a unit vector, $\mathbf{n}_t^{\mathsf{s}}$ and $\mathbf{h}$ are additive and convolutional noise vectors. Applying the first-order VTS expansion and taking expectation with respect to static parameters of component θ^{jm} yields the following form of updated mean and covariance

$$\boldsymbol{\mu}_{jm}^{\mathsf{s}} = \mathbf{C}\log\left(\exp\left(\mathbf{C}^{-1}\left(\overline{\boldsymbol{\mu}}_{jm}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathtt{h}}\right)\right) + \exp\left(\mathbf{C}^{-1}\boldsymbol{\mu}_{\mathtt{n}}\right)\right) \quad (2)$$

$$\boldsymbol{\Sigma}_{jm}^{\mathsf{s}} = \mathbf{J}_{jm}\overline{\boldsymbol{\Sigma}}_{jm}^{\mathsf{s}}\mathbf{J}_{jm}^{\mathsf{T}} + (\mathbf{I} - \mathbf{J}_{jm})\,\boldsymbol{\Sigma}_{\mathtt{n}}^{\mathsf{s}}\,(\mathbf{I} - \mathbf{J}_{jm})^{\mathsf{T}} \quad (3)$$

where $\boldsymbol{\mu}_{\mathtt{h}}$ and $\boldsymbol{\mu}_{\mathtt{n}}, \boldsymbol{\Sigma}_{\mathtt{n}}$ are convolutional and additive noise parameters estimated using maximum likelihood (ML) training [10], $\mathbf{I}$ is identity matrix, $\mathbf{J}_{jm}$ is a component-specific Jacobian matrix computation of which is fully described in [9].

Several options exist to estimate the canonical HMM parameters. One approach is to train HMMs on clean data. Another approach is to adaptively train HMMs using ML [11], [12] or minimum phone error (MPE) [10], [13] training on multistyle data collected in various noise/speaker conditions. This allows more data to be used in estimating canonical model parameters.

III. Derivative kernels

Generative kernels in Figure 1 extract features from generative models. The simplest example are *log-likelihood kernels*. The base (b) feature in equation (4) is a log-likelihood of generative model computed for class ω_i from observation sequence $\mathbf{O}$ [2]

$$\boldsymbol{\phi}_{\mathtt{b}}^{0}(\mathbf{O}|\omega_i) \;=\; \left[\log\left(p(\mathbf{O}|\omega_i)\right)\right] \quad (4)$$

Another example is shown in equation (5) where, in addition to the correct class ω_i, log-likelihoods of competing classes are also appended (a) [2]

$$\boldsymbol{\phi}_{\mathtt{a}}^{0}(\mathbf{O}|\omega_i) \;=\; \begin{bmatrix} \log\left(p(\mathbf{O}|\omega_1)\right) \\ \log\left(p(\mathbf{O}|\omega_2)\right) \\ \vdots \\ \log\left(p(\mathbf{O}|\omega_K)\right) \end{bmatrix} \quad (5)$$

The features derived from base $\boldsymbol{\phi}_{\mathtt{b}}^{0}$ and appended $\boldsymbol{\phi}_{\mathtt{a}}^{0}$ log-likelihood kernels inherit conditional independence assumptions of the underlying generative models. In contrast to log-likelihood kernels features derived from *derivative kernels* have different conditional independence assumptions. Consider ρ-th order base derivative kernel where feature vector has the following form

$$\boldsymbol{\phi}_{\mathtt{b}}^{\rho}(\mathbf{O}|\omega_i) = \begin{bmatrix} \log\left(p(\mathbf{O}|\omega_i)\right) \\ \nabla_{\boldsymbol{\lambda}}\log\left(p(\mathbf{O}|\omega_i)\right) \\ \vdots \\ \nabla_{\boldsymbol{\lambda}}^{\rho}\log\left(p(\mathbf{O}|\omega_i)\right) \end{bmatrix} \quad (6)$$

In addition to correct class log-likelihood the feature vector in equation (6) incorporates derivatives up to the order ρ with respect to generative model parameters. Consider the first-order derivatives taken with respect to component θ^{jm} output distribution parameters $\boldsymbol{\lambda}_{jm} = \{\boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\}$

$$\nabla_{\boldsymbol{\lambda}_{jm}}\log(p(\mathbf{O}|\omega_i)) = \sum_{t=1}^{T} P(\theta_t^{jm}|\mathbf{O})\nabla_{\boldsymbol{\lambda}_{jm}}\log(p(\mathbf{o}_t|\theta_t^{jm})) \quad (7)$$

These derivatives are functions of component posterior probabilities, $P(\theta_t^{jm}|\mathbf{O})$, which depend on the whole observation sequence. This means that the use of derivatives introduces additional dependencies into the features. Higher-order derivatives offer more complex dependencies [7].

Since not all first and higher order derivatives are equally discriminative a subset of them are normally used. In [14] the derivatives with respect to the mean vectors ($1m$) were found to be the most discriminative first-order derivatives. The feature vector in this case has the following form

$$\boldsymbol{\phi}_{\mathtt{b}}^{1m}(\mathbf{O}|\omega_i) = \begin{bmatrix} \log\left(p(\mathbf{O}|\omega_i)\right) \\ \sum_{t=1}^{T} P(\theta_t^{1,1}|\mathbf{O})\boldsymbol{\Sigma}_{1,1}^{-1/2}(\mathbf{o}_t - \boldsymbol{\mu}_{1,1}) \\ \vdots \\ \sum_{t=1}^{T} P(\theta_t^{1,M}|\mathbf{O})\boldsymbol{\Sigma}_{1,M}^{-1/2}(\mathbf{o}_t - \boldsymbol{\mu}_{1,M}) \\ \sum_{t=1}^{T} P(\theta_t^{2,1}|\mathbf{O})\boldsymbol{\Sigma}_{2,1}^{-1/2}(\mathbf{o}_t - \boldsymbol{\mu}_{2,1}) \\ \vdots \\ \sum_{t=1}^{T} P(\theta_t^{N,M}|\mathbf{O})\boldsymbol{\Sigma}_{N,M}^{-1/2}(\mathbf{o}_t - \boldsymbol{\mu}_{N,M}) \end{bmatrix} \quad (8)$$

where N is the number of states and M is the number of components in every state. Note that consistently with other work in this area standard deviation rather than variance normalisation is performed [4].

In order to illustrate the advantages of using first and higher derivatives consider the following example [7]. A discrete

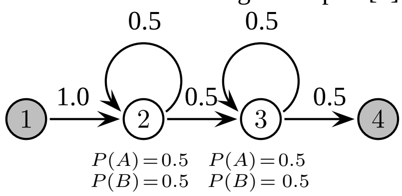

Fig. 2. Example discrete HMM topology, transition and output probabilities

HMM with the topology shown in Figure 2 is used to model two classes ω_1 and ω_2. The data for the two classes are

$$\omega_1 : \quad \texttt{AAAA}, \texttt{BBBB}$$
$$\omega_2 : \quad \texttt{AABB}, \texttt{BBAA}$$

When ML training is used to estimate HMM parameters then the state transition and output probabilities shown in Figure 2 are obtained. Since all estimated distributions yield equal probabilities the HMM is not capable of distinguishing between the two classes. The situation is different with derivative kernels. Table III shows values of selected derivatives. When the first and second order derivatives are computed with

TABLE I
FEATURE VECTOR VALUES FOR SECOND-ORDER GENERATIVE KERNEL

Feature	Class ω_1		Class ω_2	
	AAAA	BBBB	AABB	BBAA
∇_{2A}	0.50	-0.50	0.33	-0.33
$\nabla_{2A}\nabla_{2A}^{\mathsf{T}}$	-3.83	0.17	-3.28	-0.61
$\nabla_{2A}\nabla_{3A}^{\mathsf{T}}$	-0.17	-0.17	-0.06	-0.06

respect to output symbol A in state 2 (line 1 and 2) then all training examples may be correctly classified provided non-linear decision boundaries can be modelled. With the cross-state second order derivative $\nabla_{2A}\nabla_{3A}^{\mathsf{T}}$ (line 3) a linear decision boundary is sufficient. This second order derivative is capable of capturing whether label changes or not on transition from state 2 to state 3.

IV. Classification with derivative kernels

A. Isolated case kernels

The derivative kernels from Section III can be directly applied for isolated word classification tasks. One option to extend them to classify sequences rather than isolated words is to use acoustic code-breaking [15]. In this approach recognition of continuous speech is broken down into classification of a sequence of isolated speech segments. Given a word-level hypothesis with alignment an isolated discriminative classifier is sequentially applied to every segment. One classifier used for this task are binary support vector machines (SVM). For the SVMs the feature vector of the first-order derivative kernel has the following form [14]

$$\boldsymbol{\phi}(\widetilde{\mathbf{O}}, \omega_i, \omega_j) = \begin{bmatrix} \log\left(p(\widetilde{\mathbf{O}}|\omega_i)\right) - \log\left(p(\widetilde{\mathbf{O}}|\omega_j)\right) \\ \nabla_{\boldsymbol{\lambda}} \log\left(p(\widetilde{\mathbf{O}}|\omega_i)\right) \\ \nabla_{\boldsymbol{\lambda}} \log\left(p(\widetilde{\mathbf{O}}|\omega_j)\right) \end{bmatrix} \tag{9}$$

where ω_i and ω_j are two classes and $\widetilde{\mathbf{O}}$ is a segment of observation sequence. Note that the feature vector in equation (9) is a *joint feature vector* which incorporates features simultaneously from two classes. For multi-class classification with SVMs schemes such as majority voting [14], [4] and tree-based reductions [16] have been examined. However, with large number of words the number of binary SVMs required in these approaches becomes large. One option to address this issue is to use a multi-class SVM [5]. However, with high-dimensional derivative features and large number of classes the total dimensionality of the joint feature vector becomes huge. This makes constraint satisfaction of maximum margin training computationally infeasible. Therefore in [5] log-likelihood rather than derivative kernels were used.

The use of acoustic code-breaking approach is suboptimal in several ways. The first issues is that the discriminative model is defined on a word-level which is not useful for medium/large vocabulary tasks. The use of subword models is complicated as phone boundaries are hard to reliably estimate. Another issue with acoustic code-breaking is that isolated segments rather than continuous sequences are modelled.

B. Continuous case kernels

This paper extends derivative kernels to classify continuous sequences by using continuous discriminative models [7], [5], [6]. The model considered in this work has a log-linear form

$$P(\mathbf{W}|\mathbf{O}) = \frac{\exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}, \mathbf{W}, \boldsymbol{\theta}))}{\sum_{\mathbf{W}'} \exp(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}, \mathbf{W}', \boldsymbol{\theta}'))} \tag{10}$$

where $\boldsymbol{\alpha}$ are discriminative parameters and $\boldsymbol{\theta}$ is an alignment. An important decision to make is at which level $\boldsymbol{\theta}$ will segment the data as this defines the level of conditional independence assumption in the model. Segmenting data at the word-level is not useful for medium/large vocabulary acoustic models. Similarly to the work in [6] here the data is segmented at the phone level. Figure 3 illustrates the structure of the model in equation (10) by a lattice typically used in discriminative HMM training [17]. There every word arc is segmented into

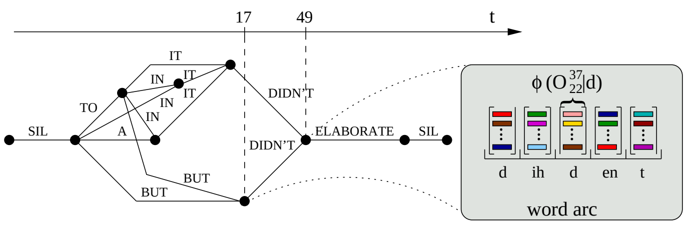

Fig. 3. Structure modelling approach in continuous discriminative models

a sequence of phone arcs. This allows context-dependent generative models attached to phone arcs to be directly used in derivative kernels. The features extracted are shown in Figure 3 as the column vectors. Note that for simplicity context-independent labels are shown.

Given observation sequence $\mathbf{O}$ and hypothesised word sequence $\mathbf{W}$ aligned by $\boldsymbol{\theta}$ the model in equation (10) assigns a *score* equal to the exponent of the dot-product below

$$\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{W},\boldsymbol{\theta})=\sum_{i=1}^{L_{\mathtt{p}}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})},w_i)+\sum_{j=1}^{L_{\mathtt{w}}}\log\left(P(\mathbf{w}_j)\right) \quad (11)$$

The dot product in equation (11) is a summation of phone-level dot products and word-level language model probabilities. Features used at the phone-level are those extracted by derivative kernels from generative models

$$\boldsymbol{\phi}(\mathbf{O},w)=\begin{bmatrix}\delta(w,\omega_1)\boldsymbol{\phi}^{\rho}_{\mathtt{b}}(\mathbf{O}|\omega_1)\\ \vdots\\ \delta(w,\omega_{K_{\mathtt{p}}})\boldsymbol{\phi}^{\rho}_{\mathtt{b}}(\mathbf{O}|\omega_{K_{\mathtt{p}}})\end{bmatrix} \quad (12)$$

where w is one of $K_{\mathtt{p}}$ context-dependent classes. The vector in equation (12) is a high-dimensional *joint feature vector*, the use of delta functions ensures that only one $\boldsymbol{\phi}^{\rho}_{\mathtt{b}}(\mathbf{O}|\omega_i)$ is active on every phone arc. The language model probabilities in equation (11) are obtained in this work from a n-gram model.

During training/decoding the most likely alignment

$$\widehat{\boldsymbol{\theta}}/\{\widehat{\boldsymbol{\theta}},\widehat{\mathbf{W}}\}=\arg\max_{\boldsymbol{\theta}/\{\boldsymbol{\theta},\mathbf{W}\}}\left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{W},\boldsymbol{\theta})\right\} \quad (13)$$

with respect to discriminative parameters $\boldsymbol{\alpha}$ theoretically should be used. The inference problem in equation (13) can be solved using the semi-Markov equivalent [18] of the Viterbi algorithm. In [19] the impact of using the most likely alignment was investigated with the appended log-likelihood features $\boldsymbol{\phi}^{0}_{\mathtt{a}}$ on a digit string recognition task. Small improvements were observed over using alignments produced by HMMs. In this work the use of optimal alignment was not investigated and the alignment provided by HMMs was adopted.

V. Parameter tying and estimation

A. Parameter tying

When context-dependent generative models are used the number of possible classes becomes large. It is unlikely that the amount of training data available will be sufficient to robustly estimate parameters of all classes. The standard approach with generative models is to use *state-level* phonetic decision trees to cluster phonetically similar states together [20]. Since discriminative classes in this work are defined on a model rather than state level the trees created for generative models can not be re-used. Therefore, another set of *model-level* decision trees is created as described in [6].

When derivative features are used there is an additional issue to consider. In contrast to log-likelihoods the derivatives are computed with respect to components of generative model states. The clustering procedure applied in phonetic decision tree building is insensitive to the order of states/components and components when used at the model and state level respectively. The derivative features however have a fixed order. Consider an example on the left in equation (14) where for simplicity the class label is omitted and one-state HMM is assumed.

$$\begin{bmatrix}\alpha_0\\ \boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_M\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\log\left(p(\mathbf{O})\right)\\ \nabla_{\boldsymbol{\lambda}_1}\log\left(p(\mathbf{O})\right)\\ \vdots\\ \nabla_{\boldsymbol{\lambda}_M}\log\left(p(\mathbf{O})\right)\end{bmatrix}\quad\begin{bmatrix}\alpha_0\\ \boldsymbol{\alpha}\\ \vdots\\ \boldsymbol{\alpha}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\log\left(p(\mathbf{O})\right)\\ \nabla_{\boldsymbol{\lambda}_1}\log\left(p(\mathbf{O})\right)\\ \vdots\\ \nabla_{\boldsymbol{\lambda}_M}\log\left(p(\mathbf{O})\right)\end{bmatrix} \quad (14)$$

When several generative models extract features for one discriminative class then the order of components in these models can adversely affect the discriminative ability of derivatives. One option to overcome this is to ensure that a small number of generative models is used by any discriminative class. However, this can introduce robustness issues as fewer training examples will be available. The option considered in this work is to tie parameters associated with derivatives within states as shown on the right in equation (14). With limited amount of training data this approach can improve robustness by reducing the number of parameters by a factor of M.

B. Parameter estimation

The standard criterion to train log-linear models is a conditional maximum likelihood (CML). For tasks such as ASR minimum Bayes' risk (MBR) training is a popular alternative approach. The objective function in MBR training is given by

$$\mathcal{F}_{\mathtt{mbr}}(\boldsymbol{\alpha})=\sum_{r=1}^{R}\sum_{\mathbf{W}}P(\mathbf{W}|\mathbf{O}^{(r)})\mathcal{L}(\mathbf{W},\mathbf{W}^{(r)}_{\mathtt{ref}}) \quad (15)$$

where loss function $\mathcal{L}(\mathbf{W},\mathbf{W}_{\mathtt{ref}})$ may be defined on a sentence, word, phone or frame level. In this work the standard phone-level loss function [17] is used. Alternatively, with large dimensional features and limited amount of training data large margin training may be more appropriate. However, current implementations of large margin training [8] can not handle large dimensional features. Therefore in this work a variant of MPE training [6] is used.

In MPE training of log-linear models standard gradient-based optimisation is performed

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_{\mathtt{mpe}}(\boldsymbol{\alpha})=\sum_{r=1}^{R}\sum_{\mathbf{a}\in\mathbf{L}^{(r)}_{\mathtt{den}}}\mathcal{C}(\mathbf{a})P(\mathbf{a}|\mathbf{O}^{(r)})\boldsymbol{\phi}(\mathbf{O}^{(r)}_{t(\mathbf{a})},w) \quad (16)$$

where $\mathcal{C}(\mathbf{a})$ is phone arc $\mathbf{a}$ contribution to the average accuracy, $P(\mathbf{a}|\mathbf{O})$ is arc posterior probability and $\boldsymbol{\phi}(\mathbf{O},w)$ is given by equation (12). Storing high-dimensional features attached to every phone arc as in Figure 3 is impractical for medium/large vocabulary tasks. In this paper on-the-fly training is performed where every lattice is passed through twice. The first pass extracts derivative features on the fly and accumulates dot products with discriminative parameters. These phone-level dot products are then combined with language model probabilities in a lattice-based forward-backward algorithm [17] to yield arc posterior probabilities and contributions. In the second pass the gradient in equation (16) is accumulated.

Although every derivative is computed twice there is no need to keep features attached to phone arcs. The derivatives can be computed on-the-fly and destroyed once arc $\mathbf{a}$ is finished.

Regularisation is important when estimating parameters of log-linear models. In this work regularised training is performed where the final objective function to maximise has the following form

$$\mathcal{F}(\boldsymbol{\alpha}) = \mathcal{F}_{\mathtt{mpe}}(\boldsymbol{\alpha}) - \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \tag{17}$$

The second term in equation (17) originates from a Gaussian prior. The mean of the prior has the form

$$\boldsymbol{\alpha}_0^{(\omega)} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^{\mathsf{T}} \tag{18}$$

which in equation (10) would yield the generative model performance. The weight matrix $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ in this work has a diagonal form where a separate $\sigma_{\mathtt{l}}$ and $\sigma_{\mathtt{d}}$ weights are used for parameters associated with log-likelihood and derivatives respectively.

VI. RESULTS

This section describes experiments with derivative kernels in AURORA 2 and AURORA 4 task. Only first-order derivatives with respect to mean vectors are considered. For all systems the continuous discriminative models are initialised with the sparse parameter vector in equation (18) to yield generative model performance on the first iteration. Similarly to other work in this area RProp optimisation is performed [1]. To prevent over-training a subset of test data was used to stop training, Set A for AURORA 2 and Set B for AURORA 4.

A. AURORA 2

AURORA 2 is a connected digit string recognition task. The number of classes is 11 plus the `sil` and `sp` model. The generative model of digits is a whole-word HMM with 16 states and 3 components/mixture trained using ML on the clean data. The setup used follows the one described in [4]. The continuous discriminative model is based on derivative features $\phi_{\mathtt{b}}^{1m}$ in equation (8), no language model is used. As a contrast another model is built based on one-dimensional features $\phi_{\mathtt{b}}^{0}$ in equation (4). The number of discriminative parameters is 21,554 and 13 respectively. The multi-style data is used for training.

The word error-rate (WER) averaged over 0-20 dB test data of the VTS-compensated HMMs (VTS) and discriminative classifiers is shown in Table II. The first block quotes the

TABLE II
AURORA2 RECOGNITION RESULTS BASED ON CLEAN-TRAINED HMMS

System	Test set A	B	C	Avg
VTS	9.8	9.1	9.5	**9.5**
SVM	7.5	7.4	8.1	**7.6**
$\phi_{\mathtt{b}}^{0}$	8.1	7.4	8.2	**7.8**
$\phi_{\mathtt{b}}^{1m}$	7.0	6.6	7.6	**7.0**

acoustic code-breaking results with binary SVMs [4] as described in Section IV-A. As can be seen from Table II the use of isolated discriminative classifier with derivative kernels yielded large gains over the VTS. The second block in Table II shows the performance of continuous discriminative models. The one-dimensional features $\phi_{\mathtt{b}}^{0}$ show results comparable to the performance of SVMs but have significantly fewer parameters. The derivative features $\phi_{\mathtt{b}}^{1m}$ improve the result of $\phi_{\mathtt{b}}^{0}$ relatively by 10%, however, the number of added parameters is approximately half of those available to the HMMs. Comparing the performance of the isolated SVMs and continuous derivative kernels it can be seen that modelling whole sentences rather than isolated segments gives consistent gains. The same was observed with the appended log-likelihood kernels in [5].

B. AURORA 4

AURORA 4 is a noise-corrupted medium/large vocabulary task based on the Wall Street Journal (WSJ) data. Two configurations of canonical HMMs were considered. The first repeats the previous setup where the HMMs are trained from clean data (SI-84 WSJ0 part, ∼14 hours). In the second more advanced VTS-adaptive training (VAT) is used to obtain the canonical HMM [10], [12]. For both setups the HMMs are state-clustered triphones (∼3140 states) with ∼16 components/mixture. Model-based noise compensation is done in two cycles where multiple (4) iterations of VTS compensation are performed for the training and test data, the supervision hypothesis is updated after each cycle. The discriminative model is based on $\phi_{\mathtt{b}}^{0}$ and $\phi_{\mathtt{b}}^{1m}$ features. The parameters of discriminative classes are tied to yield 47 and 4020 classes. Evaluation is performed using the standard 5000-word WSJ0 bigram model on four noise-corrupted test sets based on NIST Nov'92 WSJ0 test set.[1]

The first configuration investigates the usefulness of derivative kernels based on context-dependent HMMs with different number of discriminative classes. Table III shows AURORA 4 recognition results. The first block gives baseline performance

TABLE III
AURORA 4 RECOGNITION RESULTS BASED ON CLEAN-TRAINED HMMS

Classes	System	State tied $\boldsymbol{\alpha}$	Test set A	B	C	D	Avg
	VTS		7.1	15.3	12.1	23.1	**17.9**
47	$\phi_{\mathtt{b}}^{0}$		7.6	14.6	11.8	22.2	**17.2**
	$\phi_{\mathtt{b}}^{1m}$	yes	7.5	14.1	11.3	21.6	**16.6**
		no	7.4	14.3	11.7	21.9	**16.9**
4020	$\phi_{\mathtt{b}}^{0}$		6.6	14.2	10.7	21.8	**16.7**
	$\phi_{\mathtt{b}}^{1m}$	yes	6.8	13.7	10.6	21.3	**16.2**
		no	6.7	13.5	10.2	21.1	**16.0**

of the VTS-compensated HMMs. The second block gives results for 47-class discriminative model based on $\phi_{\mathtt{b}}^{0}$ and $\phi_{\mathtt{b}}^{1m}$ features. The first line in the second block shows that the use of one-dimensional $\phi_{\mathtt{b}}^{0}$ features gives gains over the VTS although the number of added parameters is only 47.

[1] Test set A is clean, set B has 6 types of noise added, set C has the channel distortion introduced and set D has both the additive noise and the channel distortion. Average SNR in noise-corrupted data is 10 dB.

The next two lines show that when ϕ_{b}^{1m} features are used then arbitrary ordering of components in HMMs has a clear impact on discriminative model performance. The third block in Table III shows results for 4020-class discriminative model in the three cases described above. As in the case of 47 classes the use of one-dimensional features yielded gains over the VTS. The addition of derivatives similarly improved the results further. As discussed in Section V with large number of classes the impact of arbitrary component ordering is expected to be small. The results in Table III confirm this by showing that tying parameters instead has lead to a small drop in classification accuracy, However, since the number of parameters in the tied case is less by a factor of 16 the within-state tying is useful for making compact discriminative models based on derivative kernels. Comparing the second and third block consistent gains can be observed from using more discriminative classes.

The second configuration used a VTS adaptively trained HMM system (VAT). Note in this configuration both the generative and discriminative models are trained on multistyle data. The following table shows the performance of baseline VAT, MPE-trained VAT (MPE-VAT) from [21] and 4020-class continuous discriminative models based on ϕ_{b}^{0} and ϕ_{b}^{1m} features. Comparing the VAT in Table IV (line 1) and the VTS in Table III (line 1) gain around 2% absolute can be observed from the adaptive training of generative parameters. The VAT+ϕ_{b}^{0} discriminative model gives gains over the VAT. The use of derivative features again improves the performance further. Comparing Tables III and IV shows that the use of derivative features in 4020-class VTS+ϕ_{b}^{1m} model allows to achieve the performance of more advanced VAT system. Further, by comparing lines 2 and 3 in Table IV it is interesting to note that the VAT+ϕ_{b}^{0} model gives the same level of performance as the MPE-VAT though the number of added parameteres in the former is just 4020 and the generative model is ML-trained. Finally, comparing the performance of VAT+ϕ_{b}^{1m} with the MPE-VAT system shows that derivative features on average yield 0.5% absolute improvement and the largest gain comes from the most noisy conditions.

TABLE IV
AURORA 4 RECOGNITION RESULTS BASED ON VAT HMMS AND COMPARISON TO MPE-VAT HMMS

System	Test set				Avg
	A	B	C	D	
VAT	8.6	13.8	12.0	20.1	**16.0**
MPE-VAT	7.2	12.8	11.5	19.7	**15.3**
VAT+ϕ_{b}^{0}	7.7	13.1	11.0	19.5	**15.3**
VAT+ϕ_{b}^{1m}	7.4	12.6	10.7	19.0	**14.8**

VII. Conclusion

This paper has described a continuous discriminative model based on derivative kernels which is suitable for noise-robust medium/large vocabulary speech recognition. Here the generative models are adapted to noise/speaker conditions using model-based techniques. The adapted models are then used to extract features from observation sequences. Previous work in small vocabulary tasks with whole word/phone models has been extended to allow context-dependent models to be used. At the phone level, in addition to log-likelihood, the first-order derivatives with respect to HMM mean vectors are used as the features. Parameter tying and estimation with large number of discriminative classes and high-dimensional features is described. The performance of continuous discriminative model was evaluated on two noise-corrupted tasks: AURORA 2 and AURORA 4. Consistent gains have been observed over VTS-compensated clean-trained ML, VTS adaptively trained ML and MPE HMM systems.

Acknowledgment

Anton Ragni is jointly funded by EPSRC, HTK and Toshiba Research Europe Limited.

References

[1] G. Zweig and P. Nguyen, “A segmental CRF approach to large vocabulary continuous speech recognition,” in *Proc. ASRU*, 2009.
[2] N. Smith, “Using Augmented Statistical Models and Score Spaces for Classification,” Ph.D. dissertation, Cambridge University, 2003.
[3] G. Zweig *et al.*, “Speech recognition with segmental conditional random fields: a summary of the JHU CLSP 2010 summer workshop,” in *Proc. ICASSP*, 2011.
[4] M. Gales and F. Flego, “Discriminative classifiers with adaptive kernels for noise robust speech recognition,” *Computer Speech and Language*, vol. 24, pp. 648–662, 2010.
[5] S.-X. Zhang, A. Ragni, and M. Gales, “Structured log-linear models for noise robust speech recognition,” *IEEE Sig. Proc. Let.*, 2010.
[6] A. Ragni and M. Gales, “Structured discriminative models for noise robust speech recognition,” in *Proc. ICASSP*, 2011.
[7] M. Layton, “Augmented statistical models for classifying sequence data,” Ph.D. dissertation, Cambridge University, 2006.
[8] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” *J. Mach. Learn. Res.*, pp. 1453–1484, 2005.
[9] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, “HMM Adaptation using Vector Taylor Series for Noisy Speech Recognition,” in *Proc. ICSLP*, Beijing, China, 2000.
[10] H. Liao and M. Gales, “Joint uncertainty decoding for robust large vocabulary speech recognition,” Cambridge Uni., Tech. Rep. CUED/F-INFENG/TR552, November 2006.
[11] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, “A compact model for speaker-adaptive training,” in *Proc. ICSLP*, 1996.
[12] O. Kalinli, M. Seltzer, and A. Acero, “Noise adaptive training using a vector Taylor series approach fo noise robust automatic speech recognition,” in *Proc. ICASSP*, 2009.
[13] F. Flego and M. Gales, “Discriminative adaptive training with VTS and JUD,” in *Proc. ASRU*, 2009.
[14] N. Smith and M. Gales, “Speech recognition using SVMs,” in *Adv. in Neural Inf. Proc. Systems*, 2001.
[15] V. Venkataramani, S. Chakrabartty, and W. Byrne, “Support vector machines for segmental minimum Bayes risk decoding of continuous speech,” in *Proc. ASRU*, 2003.
[16] M. Gales, A. Ragni, H. AlDamarki, and C. Gautier, “Support vector machines for noise robust ASR,” in *Proc. ASRU*, 2009.
[17] D. Povey, “Discriminative training for large vocabulary speech recognition,” Ph.D. dissertation, Cambridge University, 2004.
[18] S. Sarawagi and W. Cohen, “Semi-markov conditional random fields for information extraction,” in *Proc. NIPS*, 2005.
[19] S.-X. Zhang and M. Gales, “Structured support vector machines for noise robust continuous speech recognition,” in *Proc. Interspeech*, 2011.
[20] S. Young, J. Odell, and P. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” in *Proc. ARPA Workshop HLT*, 1994.
[21] F. Flego and M. Gales, “Factor analysis based VTS and JUD noise estimation and compensation,” Cambridge University, Tech. Rep. CUED/F-INFENG/TR653, 2011.