Combining Derivative and Parametric Kernels for Speaker Verification

C. Longworth*, *Student Member, IEEE,* and M.J.F. Gales, *Member, IEEE*

***Abstract*—Support Vector Machine-based speaker verification (SV) has become a standard approach in recent years. These systems typically use dynamic kernels to handle the dynamic nature of the speech utterances. This paper shows that many of these kernels fall into one of two general classes, derivative and parametric kernels. The attributes of these classes are contrasted and the conditions under which the two forms of kernel are identical are described. By avoiding these conditions gains may be obtained by combining derivative and parametric kernels. One combination strategy is to combine at the kernel level. This paper describes a maximum-margin based scheme for learning kernel weights for the SV task. Various dynamic kernels and combinations were evaluated on the NIST 2002 SRE task, including derivative and parametric kernels based upon different model structures. The best overall performance was 7.78% EER achieved when combining five kernels.**

***Index Terms*—Speaker Recognition, Dynamic Kernels, Support Vector Machines, Classifier Combination.**

I. Introduction

SPEAKER Verification (SV) is a binary classification task. The objective is to decide, given an utterance of speech and an associated identity claim, whether the speech was uttered by the claimed speaker or by an imposter. Traditional approaches to text-independent SV have made use of generative models, normally Gaussian Mixture Models (GMMs) to approximate the distribution of the speech associated with each target speaker. The classification decision is then made based upon the log-likelihood ratio between the target speaker model and a Universal Background Model (UBM) trained to represent all speakers.

Recent approaches have examined how to apply Support Vector Machines (SVMs) to the SV task [1][2]. SVMs are a form of discriminative classifier based upon a maximum-margin training criterion that have been successfully applied to many different applications. SVMs can only use input of a fixed dimensionality. Hence, they can not be directly applied to classification of speech, which is typically parameterised as variable length sequences of observations. This has lead to the development of *dynamic kernels* that implicitly map variable length speech utterances into a fixed dimensional representation. A variety of dynamic kernels have been proposed and applied to the SV task. Common examples include Fisher Kernels [3], GMM-supervector kernels [4], MLLR kernels [5] and CAT kernels [6]. SVMs have generally been found to outperform traditional log-likelihood based approaches.

C. Longworth and M.J.F. Gales are with the Engineering Department, Cambridge University, Cambridge CB2 1PZ, UK (email: cl336@eng.cam.ac.uk; mjfg@eng.cam.ac.uk). C. Longworth was supported by a Schiff studentship.

This paper attempts to unify various forms of dynamic kernel within a general framework. Many commonly used kernels can be placed into one of two classes, *parametric kernels* and *derivative kernels*. The two types of kernel are closely related and under certain conditions, described in this paper, the features obtained will be identical. As well as establishing the conditions under which kernels are the same, the framework may also be used to motivate new forms of kernel. For each form of parametric kernel a corresponding derivative kernel can be defined and inversely, for each derivative kernel there exists a corresponding parametric kernel. In many cases the derivative equivalents to many commonly used parametric kernels have not previously been applied to SV.

In the second half of this paper a general scheme is described for combining multiple complementary kernels for the SV task. In many recent approaches such as [7], combination is performed by fusing the output scores from multiple SVMs. A scheme such as logistic regression can be used to train a suitable weighting for each score [8]. For SVM-based systems, an alternative approach is to combine classifiers at the kernel level. Here a suitable set of kernel weights must be obtained. This task is known as Multiple Kernel Learning (MKL).

One approach to MKL is to select the weighting that minimises the cross-validation error by performing a grid search over all possible weightings. Unfortunately this is only feasible when the number of kernels is small and is generally unsuitable for anything other than pairwise combination. An efficient alternative approach was proposed in [9] and extended in [10]. Here kernel weights are obtained using a maximum margin criterion. A standard SVM implementation can be used to efficiently select suitable weights even when the number of kernels is large. This approach also has several advantages over score-fusion. Firstly, a consistent criterion is used to train both the SVM parameters and the kernel weights. A second advantage is that unlike score-fusion this maximum-margin MKL scheme does not require a separate development dataset.

This paper considers a number of refinements to this scheme for the SV task. The standard scheme has a known tendency to find sparse kernel weightings. For a given set of kernels the optimal level of sparsity may vary depending on the task. In this work a regularisation term is applied to the objective function. This allows the user to select the desired level of sparsity by adjusting a constant. Unlike grid-search based MKL, this constant may be efficiently selected via cross-validation even when the number of kernels is high. This paper also considers cross-speaker tying of kernel weights. By defining the objective function over all speakers a robust estimate for the kernel weights may be obtained even when

the amount of enrollment data available per speaker is limited.

This paper is organised as follows. The next section describes SVM-based speaker verification and the use of dynamic kernels. Two general categories of dynamic kernel, derivative and parametric kernels, are introduced and the conditions under which they will be complementary described. In Section III, Multiple Kernel Learning is discussed. In Section IV, experimental results on the NIST 2002 SRE dataset are presented. Finally conclusions are drawn.

II. SVM-Based Speaker Verification

The objective of the speaker verification (SV) task is to determine, given an utterance $\boldsymbol{O}$ and associated identity claim s, whether $\boldsymbol{O}$ was uttered by the target speaker s or by an imposter. If a function $S(\boldsymbol{O}, s)$ is available to assign a score to each utterance, decisions can be made by comparing each score to a fixed threshold.

$$S(\boldsymbol{O}, s) \quad \begin{matrix} \text{target speaker} \\ > \\ < \\ \text{imposter} \end{matrix} \quad \text{Threshold} \tag{1}$$

A standard approach to SV assigns a score to each utterance based upon the log-likelihood ratio (LLR) between a Gaussian Mixture Model (GMM) trained to represent speaker s and a Universal Background Model (UBM) GMM representing all speakers [11]. As the amount of available enrollment speech per speaker is usually limited, the parameters of the speaker-dependent models are typically estimated by MAP-adapting the mean parameters of the UBM [12].

Recently there has been interest in obtaining scores using the output of a Support Vector Machine (SVM) classifier. The SVM is a binary, discriminative classification scheme that has been successfully applied to a wide variety of tasks. Given N training samples, $\boldsymbol{X} = \{\boldsymbol{x}_1 \dots \boldsymbol{x}_N\}$, where each sample $\boldsymbol{x}_i$ has associated binary label $y_i \in \{-1, 1\}$, the SVM algorithm will train a linear decision boundary with form

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0 \tag{2}$$

SVMs use a *maximum-margin* training criterion. The optimal decision boundary parameters $\{\boldsymbol{w}, b\}$ are those that both minimise the empirical risk and maximise the distance between the decision boundary and the closest training example, known as the margin. The optimal decision boundary is defined by

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{w.r.t.} \quad & \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi} \\ \text{s.t.} \quad & y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i \ \forall i \\ & \xi_i \geq 0 \ \forall i \end{aligned} \tag{3}$$

where ξ_i is the training error associated with $\boldsymbol{x}_i$ and C is an constant that controls the trade-off between maximising the margin and reducing the empirical risk. The SVM optimisation function can also be defined using an equivalent dual form.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\ \text{w.r.t.} \quad & \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \ \forall i \end{aligned} \tag{4}$$

where α_i is the dual variable associated with sample $\boldsymbol{x}_i$. The primal weight vector can then be reclaimed using $\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i y_i x_i$. The non-zero elements of $\boldsymbol{\alpha}$ correspond to samples that lie on or within the margin. These samples are termed *support-vectors* and entirely determine the position of the decision boundary. An interesting property of this form is that during training and inference all references to data are in the form of inner-products between pairs of examples. It is therefore possible to replace these inner products with a *kernel function* $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ that implicitly calculates the inner-product between two vectors in some, possibly very high dimensional, *feature-space*.

SVMs cannot be directly applied to tasks involving speech as they can only use input of some fixed dimensionality. However, speech utterances are typically parameterised as variable length sequences of observations $\boldsymbol{O} = \{\boldsymbol{o}_1, \dots, \boldsymbol{o}_T\}$. This has lead to the development of *dynamic kernels*, also referred to as sequence kernels [13]. These have the form

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = < \phi(\boldsymbol{O}_i; \boldsymbol{\lambda}), \phi(\boldsymbol{O}_j, \boldsymbol{\lambda}) > \tag{5}$$

where $\phi(\boldsymbol{O}; \boldsymbol{\lambda})$ is a function that maps a speech utterance into a fixed dimensional *score-space*. The kernel also defines a distance metric between two feature vectors. Since SVMs are not invariant to feature scaling it is useful to use a metric that is maximally non-committal. One such metric is given by

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = \phi(\boldsymbol{O}_i; \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{Q}^{-1}\phi(\boldsymbol{O}_j; \boldsymbol{\lambda}) \tag{6}$$

where $\boldsymbol{Q}$ is the Fisher information matrix defined as

$$\boldsymbol{Q} = \mathcal{E}\left\{(\phi(\boldsymbol{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi)(\phi(\boldsymbol{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi)^{\mathrm{T}}\right\} \tag{7}$$

$$\boldsymbol{\mu}_\Phi = \mathcal{E}\{\phi(\boldsymbol{O}; \boldsymbol{\lambda})\} \tag{8}$$

where $\mathcal{E}\{\}$ is the expectation with respect to $\boldsymbol{O}$. $\boldsymbol{Q}$ is often approximated by the covariance matrix of the training data in the feature space. Also it is often set to be diagonal.

Using a dataset of speech utterances $\{\boldsymbol{O}_1, \dots, \boldsymbol{O}_N\}$ from a variety of speakers, a binary classifier that distinguishes between the target speaker s and all other, imposter, speakers may be trained. A binary-labelled training dataset is required. This may be constructed by assigning each utterance $\boldsymbol{O}_i$ label $y_i = 1$ if $\boldsymbol{O}_i$ was spoken by speaker s, otherwise $y_i = -1$. A speaker-dependent decision boundary can then be trained using the kernel given in equation 6. Given this model, test utterance $\boldsymbol{O}^{\mathrm{v}}$ can be scored using:

$$S(\boldsymbol{O}^{\mathrm{v}}, s) = \sum_{i=1}^{N}\alpha_i^{(s)} y_i K(\boldsymbol{O}_i, \boldsymbol{O}^{\mathrm{v}}) + b^{(s)} \tag{9}$$

The nature of the dynamic kernel $K(\boldsymbol{O}_i, \boldsymbol{O}_j)$ depends upon the form of $\phi(\cdot)$. Dynamic kernels based upon vector-averaging [14] or Dynamic Timewarping [15] have been developed but these have now been largely superseded by *gener-*

ative kernels where the score-operator depends upon an associated generative model. A number of different dynamic kernels of this form have been proposed for speaker verification, for example the Fisher kernel [3], GMM-supervector kernel [4] and MLLR-kernel [5]. These kernels can be characterised into two broad classes depending upon the form of $\phi(\boldsymbol{O};\boldsymbol{\lambda})$. These will be referred to as *parametric kernels* and *derivative kernels*.

A. Parametric Kernels

Parametric kernels are a form of dynamic kernel where the feature-space consists of a set of parameters $\boldsymbol{\lambda}$ associated with a generative model. A variable length utterance is mapped to a fixed dimensional feature representation by training a generative model to represent the utterance and then concatenating the model parameters into a feature vector. Hence the location of an utterance within the feature space is determined by maximum likelihood parameter estimates given the verification utterance $\boldsymbol{O}^{(\mathrm{v})}$. Thus

$$\phi_{\boldsymbol{\lambda}}(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda}) = \left[\hat{\boldsymbol{\lambda}}\right], \quad \hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}}\{\log p(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda})\} \tag{10}$$

One property of this form of kernel is that the derivative with respect to the parameters of the generative model is zero when differentiated at the ML estimate, i.e.

$$\nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda})\Big|_{\hat{\boldsymbol{\lambda}}} = \mathbf{0} \tag{11}$$

The precise nature of the parametric kernel is determined by the generative model used to represent the speaker. One parametric kernel that has been successfully used for speaker verification is the GMM-supervector kernel [4]. In this kernel, the feature-space is formed from the concatenated means of an utterance-dependent GMM. As there are typically not enough observations per component to robustly estimate the parameters, MAP adaptation, using the UBM as a prior, is used instead. Here

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}}\{\log p(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda}) + \log p(\boldsymbol{\lambda})\} \tag{12}$$

where $p(\boldsymbol{\lambda})$ is based on the UBM parameters. In this case the property in equation 11 will not be satisfied. For a GMM the ML or MAP estimate has no closed-form solution. Iterative approaches based on EM are commonly used. For component m the MAP-adapted mean at iteration k is given by

$$\boldsymbol{\mu}_m^{(k)} = \frac{\sum_{t=1}^{T} \gamma_m^{(k-1)}(t)\boldsymbol{o}_t^{(\mathrm{v})} + \tau\tilde{\boldsymbol{\mu}}_m}{\sum_{t=1}^{T} \gamma_m^{(k-1)}(t) + \tau} \tag{13}$$

where $\tilde{\boldsymbol{\mu}}_m$ is the UBM mean vector associated with component m (which is also used as the initial parameters $\boldsymbol{\mu}_m^{(0)}$), $\gamma_m^{(k)}(t) = P(m|\boldsymbol{o}_t^{(\mathrm{v})};\boldsymbol{\lambda}^{(k)})$, the posterior probability of component m at time t given observation $\boldsymbol{o}_t^{(\mathrm{v})}$ and $\boldsymbol{\lambda}^{(k)}$, and τ is the standard MAP adaptation constant that controls the influence of the prior on the final model. If k iterations of mean-only MAP adaptation are performed the feature-space for the GMM-supervector kernel is

$$\phi_{\boldsymbol{\lambda}}(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda}^{(k)}) = \left[\boldsymbol{\mu}_1^{(k)\mathrm{T}}, \ldots, \boldsymbol{\mu}_M^{(k)\mathrm{T}}\right]^{\mathrm{T}} \tag{14}$$

In [4], a distance metric is defined such that the kernel function is an upper bound on the KL divergence between the two utterance-dependent models. This normalises each component mean by the associated mixture weight and the inverse of the covariance matrix. In this work the distance metric given in equation 6 is used, which is consistent with the metrics used for the other kernels.

Parametric Kernels may be based around alternative forms of generative model. The MLLR kernel [5] and CMLLR kernel [16] are examples of parametric kernels where the generative model includes an utterance-dependent linear transform. In both cases the feature space consists of the concatenated transform parameters only. Another parametric kernel proposed for speaker verification is the Cluster-Adaptive-Training (CAT) kernel used in [6]. Here the feature space consists of the utterance-dependent cluster weights associated with a trained CAT generative model [17].

B. Derivative Kernels

Derivative kernels provide an interesting contrast to parametric kernels. Rather than using model parameters as the feature-space, the partial derivatives of the utterance log-likelihood with respect to individual model parameters are used instead. The set of partial derivatives form a sufficient statistic for the utterance log-likelihood. They are therefore a natural choice for an utterance-dependent fixed-dimensional feature set. For a set of model parameters, $\boldsymbol{\lambda}$, the derivative feature-space generated from a verification utterance $\boldsymbol{O}^{(\mathrm{v})}$ has the form

$$\phi_{\nabla}(\boldsymbol{O}^{(\mathrm{v})};\hat{\boldsymbol{\lambda}}) = \frac{1}{T}\left[\nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda})\Big|_{\hat{\boldsymbol{\lambda}}}\right] \tag{15}$$

where $\hat{\boldsymbol{\lambda}}$ is the model parameter value at which the derivative is evaluated. Equation 15 includes an optional term to normalise by the number of frames T in $\boldsymbol{O}^{(\mathrm{v})}$. This is important if the utterances in the dataset vary greatly in duration. Derivative kernels may also include higher-order derivative terms in the feature-space. This is not possible for parametric kernels. However, generally only first-order derivatives have been found to contain useful discriminative information [18]. Some derivative kernels, such as the log-likelihood ratio kernel [19], also include a log-likelihood ratio term in the score-space. This approach is closely related to kernel combination as described in section III. In this paper, the forms of derivative kernel considered contain only first-order derivatives.

When using derivative kernels it is necessary to define the point around which the derivative kernel feature-space will be evaluated. Various points are possible. The point may be based on the UBM parameters, which is similar to using the Fisher kernel [3]. Another possibility is to use speaker-specific parameters associated with a speaker-dependent model. As a GMM is typically used, iterative approaches are used to obtain the speaker-specific parameters. To clearer specify the iteration at which the derivative is evaluated, $\log p(\boldsymbol{O}^{(\mathrm{v})};\boldsymbol{\lambda}^{(k)})$, will be used for the feature-space evaluated at the k^{th} iteration. This approach resembles the log-likelihood ratio kernel.

The nature of derivative kernels is again determined by the generative model used to represent a speaker. Derivatives with

respect to the means of the GMM can be used [13]. Here elements of the feature space have the form

$$\nabla_{\boldsymbol{\mu}_m} \log p(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda})\Big|_{\boldsymbol{\lambda}^{(k)}} = \sum_{t=1}^{T} \gamma_m^{(k)}(t)\boldsymbol{\Sigma}_m^{-1}(\mathbf{o}_t^{\mathrm{v}} - \boldsymbol{\mu}_m^{(k)}) \quad (16)$$

Derivative kernels may also be defined using alternative forms of generative model. If the generative model contains a linear transform, derivatives with respect to this transform may be used as features leading to derivative equivalents of MLLR and CMLLR kernels. Transform-based derivative kernels are discussed further in [20]. Given a generative model and associated subset of model parameters both a parametric kernel and derivative kernel can be simply defined. In fact for each form of parametric kernel there is a directly related derivative kernel. This can be used to motivate new kernels to apply to the SV task.

C. Relationship between Parametric and Derivative Kernels

It is interesting to briefly contrast derivative kernels with parametric kernels. From equation 11, the derivative of the parametric kernel features at the ML-estimate of the model parameters will be zero for the verification data $\boldsymbol{O}^{\mathrm{v}}$. In general this will not be the case for the derivative kernel. Instead the features of the derivative kernel will be zero for the enrollment data if the ML-estimate is used as the point to specify the derivative

$$\hat{\boldsymbol{\lambda}}_e = \arg\max_{\boldsymbol{\lambda}}\{\log p(\boldsymbol{O}^{\mathrm{e}};\boldsymbol{\lambda})\}, \quad \boldsymbol{\phi}_{\nabla}(\boldsymbol{O}^{\mathrm{e}};\hat{\boldsymbol{\lambda}}_e) = \mathbf{0} \quad (17)$$

In addition derivative kernels commonly use a length normalisation term. This is not necessary for parametric kernels, where there is an implicit normalisation for the lengths, for example the normalisation term in equation 13. A consequence of this is that when a component is not observed ML-based parametric kernels are undefined, whereas derivative kernels tend to zero.

Both parametric and derivative kernels have been used successfully for speaker-verification. The respective feature-spaces can express different types of speaker-discriminant information and thus may be complementary. It is useful to establish under what conditions the two forms of kernel are the same, as this yields information as to how to make the features complementary to one another. The parametric kernel feature-space at the kth iteration of training can be expressed in the form of a gradient ascent update.

$$\boldsymbol{\phi}_{\boldsymbol{\lambda}}(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda}^{(k+1)}) = \left[\boldsymbol{\lambda}^{(k)} + \tilde{\eta}\nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda})\Big|_{\boldsymbol{\lambda}^{(k)}}\right] \quad (18)$$

where $\tilde{\eta}$ is the learning rate. This may then be expressed as a function of a derivative feature-space evaluated at $\boldsymbol{\lambda}^{(k)}$

$$\boldsymbol{\phi}_{\boldsymbol{\lambda}}(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda}^{(k+1)}) = \left[\boldsymbol{\lambda}^{(k)} + \eta\boldsymbol{\phi}_{\nabla}(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda}^{(k)})\right] \quad (19)$$

where $\eta = T\tilde{\eta}$ if duration normalisation is used, as in equation 15, otherwise $\eta = \tilde{\eta}$. The two classes of dynamic kernel are thus related to each other. Compared to the derivative kernel feature-space, the parametric kernel features includes a term $\boldsymbol{\lambda}^{(k)}$ which introduces a translation of the feature space. If a kernel that is invariant to translation is used, such as a stationary kernel, this will have no effect. Note stationary kernels, such as the Kullback-Leibler Divergence kernel [21], have the general form $K(\boldsymbol{O}_i,\boldsymbol{O}_j) = \mathcal{F}(\boldsymbol{\phi}(\boldsymbol{O}_i) - \boldsymbol{\phi}(\boldsymbol{O}_j))$ where $\mathcal{F}()$ is the function that defines the kernel.

Even if a stationary kernel is used, it is not sufficient to ensure that the two sets of features will be identical. Equation 19 contains a learning rate. Using an appropriate metric, the kernels will not depend on the learning rate if the learning rate is independent of the observation sequence since this dependency is removed by the metric (the metric used in equation 6 has this property but is not stationary). However this is not generally the case. To illustrate this consider the situation where the parametric kernel is obtained using an EM-based ML-estimation of the mean. At iteration $k+1$ the mean parametric feature-space for component m can be expressed as

$$\boldsymbol{\mu}_m^{(k+1)} = \boldsymbol{\mu}_m^{(k)} + \left(\frac{T\boldsymbol{\Sigma}_m}{\sum_{t=1}^{T}\gamma_m^{(k)}(t)}\right)\left[\frac{1}{T}\nabla_{\mu_m} \log p(\boldsymbol{O}^{\mathrm{v}};\boldsymbol{\lambda})\Big|_{\boldsymbol{\lambda}^{(k)}}\right] \quad (20)$$

EM is thus equivalent to gradient ascent using the derivative features with a learning rate that depends upon the total component occupancy for that observation sequence as well as the length of the observation sequence (when length normalisation is being used). It is more common to use MAP adaptation to successively update models. When the MAP prior $\tilde{\boldsymbol{\mu}}$ equals $\boldsymbol{\mu}^{(k)}$, this update is equivalent to gradient ascent with the following learning rate.

$$\eta = \left(\frac{T\boldsymbol{\Sigma}_m}{\sum_{t=1}^{T}\gamma_m^{(k)}(t) + \tau}\right) \quad (21)$$

If both parametric and derivative features are to be used, it is important that the features differ. This can be achieved using a non-stationary kernel, such as the kernel in equation 6, evaluating the derivative terms at a different point to the parametric features (effectively using a different number of iterations), or simply using either EM updates or MAP adaptation with a low value of τ. Combinations of these may make the features more complementary.

D. Generative Model Structure

For dynamic kernels that incorporate a generative model, such as parametric or derivative kernels, an appropriate form of model must be selected. If a GMM is used, the number of Gaussian components must be chosen. This is a trade-off between improving the ability of the model to approximate the distribution over the acoustic space and ensuring that the model parameters can be robustly estimated with the available data. A suitable model size is typically chosen by selecting a value that reduces the error rate on some development dataset. As the trade-off is data-dependent this strategy may not be optimal.

If a suitable scheme for combining classifiers is available, then other strategies may be used. Rather than selecting a single form of model, a series of dynamic kernels can

instead be defined, each based on different model structures. The associated classifiers can then be combined. Although this approach is more computationally expensive it has two advantages. Firstly, there is no need for prior knowledge about the task in order to select a suitable model size. Secondly, rather than making a single trade-off, the combined classifier can make use of features extracted from a range of different model structures, potentially leading to gains.

III. Multiple Kernel Learning

In recent SV evaluations there has been a focus on combining multiple classifiers to improve overall performance. In [7], gains were obtained by combining SVM classifiers based upon GMM-supervector kernels and MLLR kernels. For SVM classifiers two general combination strategies are available, score-fusion and kernel combination. In score-fusion a combined score is obtained by taking a weighted linear combination of the scores obtained from a set of individual classifiers. Various criteria are available to train an appropriate score-weighting. For example an additional SVM may be trained using the individual classifier scores as input features. Another criterion that is commonly used to optimise score weights is logistic regression [8]. Both of these methods require an auxiliary dataset in order to train the score weights.

An alternative approach to score-fusion is to to combine systems at the kernel level. Given a set of K kernels, a combined kernel function may be defined as the weighted sum of the individual kernels. The combined kernel has this form.

$$\boldsymbol{k}(\boldsymbol{O}_i, \boldsymbol{O}_j) = \sum_{k=1}^{K} \beta_k \boldsymbol{k}_k(\boldsymbol{O}_i, \boldsymbol{O}_j) \tag{22}$$

Function $\boldsymbol{k}_k(\boldsymbol{O}_i, \boldsymbol{O}_j)$, associated with kernel k, is defined by equation 6 for some function $\boldsymbol{\phi}_k(\boldsymbol{O}; \boldsymbol{\lambda})$ and β_k is the associated kernel weight. The weights are generally constrained such that $\beta_k > 0 \ \forall k$ and $\sum_{k=1}^{K} \beta_k = 1$. Learning a suitable set of kernel weights $\{\beta_1, \ldots, \beta_K\}$ is known as the Multiple Kernel Learning (MKL) problem [22]. This paper mainly focuses on kernel-based combination schemes.

A. Minimum-Equal Error Rate Criterion

One approach to solving the MKL problem is to select kernel weights that reduce the cross-validation error on some development dataset. This could be achieved by conducting a grid search over all possible weightings and selecting the weights that minimise the Equal Error Rate. In this work, this criterion is termed `minEER`. Unfortunately this approach is generally impractical for anything other than pairwise kernel combination. In cases where calculating the `minEER` kernel weights is feasible this metric can provide an upper bound for the gains that can be achieved using other criteria to select an appropriate set of kernel weights.

B. Maximum-Margin Based Multiple Kernel Learning

An efficient approach to MKL was developed in [9] and extended in [10]. Here the kernel weights are incorporated into the standard SVM objective function. For a set of N utterances $\{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_N\}$ each with associated label $y_i \in \{-1, 1\}$, the optimal set of weights are those that maximise the margin.

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{k=1}^{K} \frac{1}{\beta_k} ||\boldsymbol{w}_k||_2^2 + C \sum_{i=1}^{N} \xi_i \\ \text{w.r.t.} \quad & \boldsymbol{\beta}, \boldsymbol{w}_k, \boldsymbol{b}, \boldsymbol{\xi} \\ \text{s.t.} \quad & y_i \left(\sum_{k=1}^{K} \boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{\phi}_k(\boldsymbol{O}_i; \boldsymbol{\lambda}) + b \right) \geq 1 - \xi_i \ \forall i \\ & \xi_i \geq 0 \ \forall i, \quad \beta_k \geq 0 \ \forall k, \quad \sum_{k=1}^{K} \beta_k = 1 \end{aligned} \tag{23}$$

where $\boldsymbol{w}_k$ are the primal SVM weights associated with kernel k and b, $\boldsymbol{\xi}$ and C are the standard SVM bias, slack vector and regularisation term. In this formulation $\boldsymbol{\beta}$ is subsumed into the definition of the primal weights and hence does not directly appear in the marginal constraint. The primal weight vector is given by

$$\boldsymbol{w}_k = \sum_{i=1}^{N} \alpha_i y_i \beta_k \boldsymbol{\phi}_k(\boldsymbol{O}_i; \boldsymbol{\lambda}) \tag{24}$$

There are a number of issues to address when applying this form of MKL directly to speaker verification.

1) Regularisation Term: In equation 23, an l_1-norm constraint is applied to the kernel weights. A known consequence of this is to introduce a tendency towards sparse solutions [10]. For a given set of kernels, there is no guarantee that the level of sparsity will be optimal. One solution is to incorporate a regularisation term $\mathcal{R}$ into the objective function to allow the user to control the level of sparsity. A suitable form of regularisation is

$$\mathcal{R} = \varphi \sum_{k=1}^{K} (\beta_k - \frac{1}{K})^2 = \varphi \left(\sum_{k=1}^{K} \beta_k^2 - \frac{1}{K} \right) \tag{25}$$

due to the l_1-norm constraint on the kernel weights. Since the optimal solution is independent of any constant terms in the objective function, $\mathcal{R} = \varphi \sum_{k=1}^{K} \beta_k^2$ may be used instead. This allows the same form of regularisation term irrespective of K. Here φ is an empirically set constant. For positive values of φ the effect of this form of regularisation is to drive towards a uniform set of weights. When φ is negative the solution will tend to be sparse and the objective function will perform kernel selection. Although an additional parameter has been introduced, note that an appropriate value for φ may be obtained through cross-validation even when the number of kernels is large.

2) Cross-Speaker Tying: In most SVM-based speaker verification systems, such as [7][23], a distinct set of SVM parameters is trained for each speaker. However, the amount of enrollment data available per speaker is typically limited. Learning a set of speaker-dependent kernel weights in addition to the SVM parameters may lead to over-training. One way to obtain a more robust set of weights is to tie $\boldsymbol{\beta}$ over all enrolled speakers. This can be achieved by redefining the MKL objective function to sum over all speakers, while maintaining

a separate set of marginal constraints for the enrollment data associated with each speaker.

3) Dynamic Range Normalisation: The form of objective function given in (23) is biased towards those kernels for which the average magnitude of the associated feature vectors is greatest. Under a maximally non-committal kernel metric, this corresponds to the kernels for which the associated score-space has the greatest dimensionality. It is therefore important that the kernel function includes some form of dynamic range normalisation. One option is Spherical Normalisation [13] where each feature vector is mapped onto the surface of a unit sphere. An alternative approach is to perform normalisation at the kernel level. This may be achieved in a variety of ways, for example by replacing $k_k(\boldsymbol{O}_i, \boldsymbol{O}_j) \rightarrow \frac{1}{M} k_k(\boldsymbol{O}_i, \boldsymbol{O}_j)$ where M is the dimensionality of $\boldsymbol{\phi}_k(\boldsymbol{O}; \boldsymbol{\lambda})$. Alternatively, the score-space features of each kernel may simply be duplicated so all kernels have the same dimensionality. This is the method used in this work.

The `maxMargin` MKL criterion used in this work is defined by the following objective function.

$$\begin{aligned} \min \quad & \sum_{s=1}^{S} \left(\frac{1}{2} \sum_{k=1}^{K} \frac{1}{\beta_k} ||\boldsymbol{w}_k^{(s)}||_2^2 + C \sum_{i=1}^{N} \xi_i^{(s)} \right) + \varphi \sum_{k=1}^{K} \beta_k^2 \\ \text{w.r.t.} \quad & \boldsymbol{\beta}, \boldsymbol{w}_k, \boldsymbol{b}, \boldsymbol{\xi} \\ \text{s.t.} \quad & y_i^{(s)} \left(\sum_{k=1}^{K} \boldsymbol{w}_k^{(s)\mathrm{T}} \boldsymbol{\phi}_k(\boldsymbol{O}_i^{(s)}; \boldsymbol{\lambda}) + b^{(s)} \right) \geq 1 - \xi_i^{(s)} \ \forall i \ \forall s \\ & \xi_i^{(s)} \geq 0 \ \forall i \ \forall s, \quad \beta_k \geq 0 \ \forall k, \quad \sum_{k=1}^{K} \beta_k = 1 \end{aligned} \tag{26}$$

Where the speaker s ranges from $1 \ldots S$ and samples i range from $1 \ldots N^{(s)}$, $\boldsymbol{w}_k = \{\boldsymbol{w}_k^{(1)}, \ldots, \boldsymbol{w}_k^{(S)}\}$, $\boldsymbol{b} = \{b^{(1)}, \ldots, b^{(S)}\}$ and $\boldsymbol{\xi} = \{\xi^{(1)}, \ldots, \xi^{(s)}\}$.

Equation 26 may be efficiently optimised by a similar approach to that used in [10]. First, an equivalent constrained optimisation problem is defined.

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \textstyle\sum_{s=1}^{S} J(s, \boldsymbol{\beta}) + \varphi \sum_{k=1}^{K} \beta_k^2 \\ \text{s.t.} \quad & \beta_k \geq 0 \ \forall k, \quad \textstyle\sum_{k=1}^{K} \beta_k = 1 \end{aligned} \tag{27}$$

where $J(s, \boldsymbol{\beta})$ is the optimal value of the objective function associated with an SVM with kernel (22) and fixed kernel weights $\boldsymbol{\beta}$ after training on data associated with speaker s.

$$\begin{aligned} J(s, \boldsymbol{\beta}) = \max \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{O}_i, \boldsymbol{O}_j, \boldsymbol{\beta}) \\ \text{w.r.t.} \quad & \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \ \forall i \end{aligned} \tag{28}$$

A projected-gradient scheme can then be used to optimise (27). At each iteration $J(s, \boldsymbol{\beta})$ can be estimated using a standard efficient SVM implementation such as [24]. An expression for the derivatives of $J(s, \boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}$ follows from the form in [10].

C. Relationship between kernel-combination and score-fusion

For SVM classifiers two general combination strategies are available, score-fusion and kernel combination. It is interesting to briefly compare these two approaches. During score-fusion the final score is determined by combining the scores obtained from K individual classifiers. For the unweighted case the final score $S(\boldsymbol{O}^{\mathrm{v}}, s)$ assigned to test utterance $\boldsymbol{O}^{\mathrm{v}}$ is given by

$$S(\boldsymbol{O}^{\mathrm{v}}, s) = \sum_{k=1}^{K} \boldsymbol{w}_k^{(s)\mathrm{T}} \boldsymbol{\phi}_k(\boldsymbol{O}^{\mathrm{v}}; \boldsymbol{\lambda}) + b_k^{(s)} \tag{29}$$

where $\boldsymbol{w}_k^{(s)}$ and $\boldsymbol{b}_k^{(s)}$ are the weights and bias associated with SVM classifier k and $\boldsymbol{\phi}_k(\boldsymbol{O}; \boldsymbol{\lambda})$ defines the associated feature space. The parameters of the K individual classifiers are optimised independently. If all of the component classifiers are SVMs then the combined optimisation can be expressed as

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} \left(\frac{1}{2} ||\boldsymbol{w}_k^{(s)}||_2^2 + C \sum_{i=1}^{N} \xi_{ik}^{(s)} \right) \\ \text{w.r.t.} \quad & \boldsymbol{w}_1^{(s)}, \ldots, \boldsymbol{w}_K^{(s)}, b_1^{(s)}, \ldots, b_K^{(s)}, \boldsymbol{\xi}_1^{(s)}, \ldots, \boldsymbol{\xi}_K^{(s)} \\ \text{s.t.} \quad & y_i \left(\boldsymbol{w}_k^{(s)\mathrm{T}} \boldsymbol{\phi}_k(\boldsymbol{O}_i; \boldsymbol{\lambda}) + b_k^{(s)} \right) \geq 1 - \xi_{ik}^{(s)} \ \forall i \ \forall k \\ & \xi_{ik}^{(s)} \geq 0 \ \forall i \ \forall k \end{aligned} \tag{30}$$

where $\boldsymbol{\xi}_k^{(s)}$ is the training error associated with classifier k.

An alternative approach to score-fusion is to to combine systems at the kernel level. Again considering the unweighted case, a combined kernel function may be defined as the sum of K individual kernels.

$$\boldsymbol{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{K} \boldsymbol{k}_k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{31}$$

where function $\boldsymbol{k}_k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, associated with kernel k, is defined by equation 6 for some function $\boldsymbol{\phi}_k(\boldsymbol{O}; \boldsymbol{\lambda})$. The primal form optimisation problem that corresponds to an SVM with this kernel is given by

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{k=1}^{K} ||\boldsymbol{w}_k^{(s)}||_2^2 + C \sum_{i=1}^{N} \xi_i^{(s)} \\ \text{w.r.t.} \quad & \boldsymbol{w}_1^{(s)}, \ldots, \boldsymbol{w}_K^{(s)}, b^{(s)}, \boldsymbol{\xi}^{(s)} \\ \text{s.t.} \quad & y_i \sum_{k=1}^{K} \boldsymbol{w}_k^{(s)\mathrm{T}} \boldsymbol{\phi}_k(\boldsymbol{O}_i; \boldsymbol{\lambda}) + b^{(s)} \geq 1 - \xi_i^{(s)} \ \forall i \\ & \xi_i^{(s)} \geq 0 \ \forall i \end{aligned} \tag{32}$$

By introducing new variables $b_k^{(s)}$ and $\xi_{1k}^{(s)}, \ldots, \xi_{Nk}^{(s)}$ for each kernel k such that $b^{(s)} = \sum_{k=1}^{K} b_k^{(s)}$ and $\xi_i^{(s)} = \sum_{k=1}^{K} \xi_{ik}^{(s)}$ this

optimisation can be expressed in a similar form to equation 30

$$\begin{aligned} \text{min} \quad & \sum_{k=1}^{K}\left(\frac{1}{2}||\boldsymbol{w}_k^{(s)}||_2^2 + C\sum_{i=1}^{N}\xi_{ik}^{(s)}\right) \\ \text{w.r.t.} \quad & \boldsymbol{w}_1^{(s)},\ldots,\boldsymbol{w}_K^{(s)},b_1^{(s)},\ldots,b_K^{(s)},\boldsymbol{\xi}_1^{(s)},\ldots,\boldsymbol{\xi}_K^{(s)} \\ \text{s.t.} \quad & y_i\sum_{k=1}^{K}\left(\boldsymbol{w}_k^{(s)\mathrm{T}}\boldsymbol{\phi}_k(\boldsymbol{O}_i;\boldsymbol{\lambda}) + b_k^{(s)}\right) \geq 1 - \sum_{k=1}^{K}\xi_{ik}^{(s)} \ \forall i \\ & \sum_{k=1}^{K}\xi_{ik}^{(s)} \geq 0 \ \forall i \end{aligned} \tag{33}$$

Kernel combination and score fusion are closely related and the optimisation functions 30 and 33 differ only in the constraints. The constraints for score-fusion are more restrictive since each example must satisfy a separate marginal constraint for each kernel. For kernel combination, individual terms associated with each kernel in the constraint may be violated if on average the example lies outside the margin. One consequence of this is that kernel combination may generalise more effectively than score-fusion schemes when training errors are made by individual classifiers.

IV. Experimental Results

The performance of various dynamic kernels was evaluated on the 2002 NIST SRE one-speaker detection task [25][1]. This task consists of classifying individual channels of conversational speech recorded over a cellular telephone channel. The dataset consists of 330 target speakers (139 male and 191 female) each with a single utterance of enrollment data of up to 120 seconds in duration. Test utterances are scored against 11 potential speaker identities of the same gender, one of which is usually the true speaker. Each utterance was parameterised into sequences of 31-dimensional observations using a bandwidth of 0-3.8 Khz and a framerate of 10ms. Observations were comprised of 15 static and 15 delta mel-PLP coefficients and the delta energy. Static energy coefficients were not included since previous works [26] have shown that they contain no speaker-discriminant information. To introduce additional robustness to noise, Cepstral Mean Subtraction was performed followed by Cepstral Feature Warping [27] using a three second window.

Systems were primarily evaluated using the EER metric. In the 2002 SRE the detection cost function (DCF) was also used to evaluate systems. To aid comparison with other work some minDCF scores are also quoted. The normalised DCF cost used in this paper takes the form

$$\text{DCF} = P_{\text{Miss}} + 9.9P_{\text{False Alarm}}. \tag{34}$$

minDCF is the minimum DCF score obtained a posteriori by adjusting the decision threshold.

Initially, gender-dependent UBMs were trained using ML for all SRE 2002 enrollment data. Each UBM consisted of a diagonal covariance GMM with a range of Gaussian components. For each enrolled speaker, a speaker-dependent GMM was constructed by MAP adapting the means of the appropriate gender-dependent UBM. The UBM was chosen using the provided gender information. Two iterations of static prior MAP were used with τ set at 25. GMM training and adaptation was implemented using the HTK toolkit [28]. An initial baseline classifier was formed by taking the log-likelihood ratio between the target speaker model and the UBM of the appropriate gender. 128 component models were used for the baseline classifier. The speaker-dependent models were also used as the generative models for a derivative kernel. For this kernel the score-space defined in equation 15 was used, equivalent to a standard Fisher kernell [3]. This consisted of first-order derivatives with respect to the GMM means only.

Parametric kernels were also used. Here utterance-dependent GMMs were obtained by adapting the appropriate UBM means using two iterations of static-prior MAP. The task does not permit cross-gender trials so the gender of the utterance was known in all cases. For the parametric kernels τ was set at 5. During preliminary experiments this yielded gains compared to larger values. The discrepancy between the optimal value of τ for the parametric and derivative kernels is a consequence of the mismatch between training and test utterance duration present in the 2002 SRE data. Only the test data is used to adapt the UBMs for the parametric kernel. Finally, for each utterance a parametric feature-vector was constructed by concatenating the GMM means to give the score-space defined in equation 14. This implementation of a parametric kernel is equivalent, under a maximally non-committal metric, to the standard GMM-supervector kernel [4].

During preliminary experiments, this setup optimised individual performance of both parametric and derivative kernels. It was also designed to avoid the conditions given in section II-C under which derivative and parametric features are identical. The setup used was not intended to be state-of-the-art, but to demonstrate the potential gains that may be achieved by combining derivative and parametric SVM classifiers. For example, score-normalisation techniques such as T-Norm [29] and noise-robustness techniques such as NAP [4] or WCCN [30] were not used, however they are expected to provide additional gains in combination with the techniques evaluated here. Kernel-level normalisation, as described in Section III, outperformed spherical normalisation and was used in these experiments to normalise the magnitude of the feature vectors. When combining kernels of different score-space dimensionality, the features of the smaller kernels were duplicated until the dimensionality of all kernels was the same. This was not found to affect individual kernel performance. Including covariance terms into the feature space was also examined, but did not lead to to gains for either parametric or derivative systems. SVM^{light} [24] was used to train classifiers for each enrolled speaker.

The SVM regularisation term C was left at the SVM^{light} default. Imposter examples were obtained from the enrollment data associated with other speakers of the same gender[2]. To

[1]The 2002 SRE data was chosen as it is the most recent evaluation dataset to be made generally available through the LDC. Later datasets are currently only available to SRE participants. However the techniques discussed in this paper may be easily applied to more recent tasks.

[2]The setup used did not conform to the NIST SRE protocol, since enrollment data was used for both UBM training and imposter modelling. This was necessary due to the limited amount of development data available to the authors.

reduce classifier bias each true utterance was duplicated until the two training sets were equal. For each kernel, a diagonal approximation to the maximally non-committal distance metric shown in equation 6 was defined by normalising the global variance of each feature calculated over all speakers.

System	EER (%)	minDCF
GMM-LLR	12.10	0.4915
∇_{128}	8.62	0.3759
λ_{64}	9.55	0.3830
λ_{128}	8.61	0.3521
λ_{256}	8.58	0.3498
λ_{512}	8.83	0.3702
$\lambda_{128} + \nabla_{128}$	8.08	0.3440

TABLE I
COMPARISON OF EQUAL-WEIGHT KERNEL COMBINATION AGAINST DERIVATIVE (∇), PARAMETRIC (λ), AND BASELINE GMM-LLR SYSTEMS

The performance of these initial systems is shown in Table I. For 128-component models, derivative and parametric kernel performance was similar and both yielded significant gains compared to the GMM-LLR classifier. This is consistent with previous work such as [4] and [13] and demonstrates the powerful generalisation ability of the SVM classifier. For parametric kernels, optimal performance was obtained using 256 components. This contrasts with [4] where further gains were obtained using larger models. This is believed to be related to the limited amount of training data used for UBM training compared to other work.

Initially, pairwise combination of 128-component derivative and parametric kernels was examined. Equal weights were used for each kernel. The error associated with this classifier was 8.08% representing a 5% relative gain compared to the 128-component parametric kernel[3]. Although different values of τ were used to adapt the two forms of kernels, preliminary experiments where this constant was identical for both kernels also showed similar gains for combination, indicating that this was not a significant factor affecting how complementary the kernels were.

φ	Kernel Weights		EER (%)	minDCF
	∇_{128}	λ_{128}		
0	1.00	0.00	8.62	0.3759
0.008	0.80	0.20	8.19	0.3651
0.064	0.55	0.45	8.11	0.3474
∞	0.50	0.50	8.08	0.3440
`minEER`	0.62	0.38	8.04	0.3537

TABLE II
PERFORMANCE OF `maxMargin` MKL COMBINATION AS φ VARIES COMPARED TO OPTIMAL `minEER` WEIGHTING

Experiments were performed to identify whether individually weighting each kernel could yield gains compared to equal combination. Initially, combination using a `minEER` criterion was evaluated. A line-search was performed and the kernel weights selected that gave the lowest EER. Although infeasible for larger number of kernels, this criterion forms an upper bound on the gains obtainable using MKL. Next system combination was performed using the `maxMargin` criterion for MKL described in Section III. $\boldsymbol{\beta}$ was tied over all speakers. Table II compares the performances obtained using `maxMargin` for a range of values of φ against the optimal `minEER` weighting. When $\varphi = 0$ a sparse weighting is obtained that performs poorly compared to the baseline. This indicates that the standard level of sparsity associated with MKL as defined in [10] is not appropriate for this task. By increasing φ gains are observed. The case when $\varphi = \infty$ is equivalent to an equal-weighted combination. If a value for φ is selected that minimises the EER, MKL is guaranteed to not perform worse than equal-weight combination. Unlike using the `minEER` criterion this is feasible for large numbers of kernels as in all cases only a single parameter must be optimised.

System	EER (%)	
	Equal-Weight	MKL
$\lambda_{64}+\lambda_{128}$	9.02	8.55
$\lambda_{128}+\lambda_{256}$	8.32	8.32
$\lambda_{256}+\lambda_{512}$	8.52	8.52
$\lambda_{64}+\lambda_{128}+\lambda_{256}+\lambda_{512}$	8.42	8.22
$\lambda_{128}+\nabla_{128}$	8.08	8.04
$\lambda_{64}+\lambda_{128}+\lambda_{256}+\lambda_{512}+\nabla_{128}$	7.99	7.78

TABLE III
COMPARISON OF EQUAL-WEIGHT COMBINATION AGAINST `maxMargin` MKL FOR VARIOUS COMBINATIONS OF KERNELS

The `maxMargin` MKL scheme was then applied to other combinations of kernels where `minEER` is not always possible. For the optimisation problem defined in equation 26, the value of the objective function increases monotonically with φ. This means that an appropriate regularisation factor cannot be automatically selected by maximising the objective function. Instead, for each combination φ was adjusted a posteriori to reduce the EER[4]. Results are presented in Table III. Combinations of parametric kernels based upon different generative model structures were examined, using GMMs ranging from 64 to 512 components. As discussed in section II-D, using features extracted from a range of model structures may improve performance. Combining kernels based on different generative models also allows the user to avoid explicitly choosing an appropriate model structure. Although no gains were observed for equal-weight combination of 64 and 128 component models, combination of 128 and 256 component models did yield small gains compared to the individual kernels. By comparison the performance of a 512-component system was 8.83% indicating that these gains were not simply due to the increased complexity of the combined classifier.

[3]The results presented here differ from those previously reported in [20] and [31]. Since publication a flaw was discovered in the evaluation methodology that caused the reported gains from kernel combination to be severely optimistic. Updated versions of these papers are available from mi.eng.cam.ac.uk/∼cl336

[4]Selecting a value for φ that optimises EER on the test set, rather than development data, can introduce bias. However as only a single parameter is tuned this bias is expected to be small. In preliminary experiments φ was optimised independently over different subsets of speakers. The optimal φ was found to be independent of the subset chosen and the minimal EER was achieved over a wide range of φ.

For `maxMargin` MKL all examined pairwise combinations gave gains. When all four parametric kernels were combined a 0.22% reduction in EER was observed compared to equal-weight combination. Similar gains were observed in minDCF resulting in 0.3428 for four-way combination. The best overall performance for kernel-level combination was 7.78% (0.3089 minDCF) achieved when all kernels were combined. This represented a gain of 0.26% compared to an equal weight combination and 35% relative gain compared to the GMM-LLR system. From the DET curve in Figure 1 it can be seen that this system performed best over the majority of the operating range.

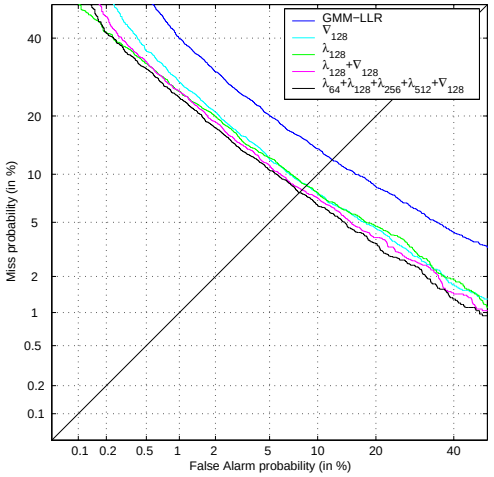

Fig. 1. DET graph comparing `maxMargin` MKL combination against individual systems

Combination	Criterion	EER (%)	
		$\nabla_{128}+\lambda_{128}$	$\nabla_{128}+\lambda_{64}+\lambda_{128}$ $+\lambda_{256}+\lambda_{512}$
Score	`Equal`	8.13	7.95
	`minEER`	8.08	-
	`LR`	8.15	7.85
Kernel	`Equal`	8.08	7.99
	`minEER`	8.04	-
	`maxMargin`	8.04	7.78

TABLE IV
COMPARISON OF KERNEL-LEVEL AND SCORE-LEVEL APPROACHES FOR CLASSIFIER COMBINATION

Finally, the kernel combination schemes were compared against standard score-fusion approaches. Several approaches were examined. Initially, combined scores were obtained by either equally weighting scores (`equal`) or by selecting the score a-posteriori that resulted in the lowest EER (`minEER`). Logistic regression (`LR`) was also applied as in [23]. In the absence of a suitable development dataset logistic regression was applied directly to the test scores using the correct label information. Like the `minEER` criterion, this breaks experimental protocol, however it does provide an upper bound on the gains that can be achieved using logistic regression. Unlike `LR` and `minEER` score-fusion, where a weight for each classifier was trained using the test data, for `maxMargin` MKL only φ, a single parameter, was optimised on the test set. This system is therefore likely be more robust to overtuning. Results for these experiments are given in Table IV. For pairwise combination `minEER` performance was best under kernel-combination indicating that this combination strategy is able to generalise more effectively. This may be related to the more relaxed marginal constraint associated with kernel combination. When all five kernels were combined `maxMargin` kernel combination outperformed `LR` score-fusion, despite the "optimal" logistic-regression scheme used.

V. CONCLUSION

This paper has discussed combination of two general forms of dynamic kernel to improve performance of an SVM-based speaker verification system. Many existing dynamic kernels can be placed into one of these two classes, *parametric kernels*, where the feature-space consists of parameters from the utterance-dependent model, and *derivative kernels*, where the derivatives of the utterance log-likelihood with respect to parameters of a generative model are used. The two sets of features produced have different properties and may be complementary. However, under certain conditions, discussed in Section II-C, the feature-spaces produced may be shown to be identical. Many systems combine multiple classifiers to improve system performance. By avoiding these conditions a complementary set of kernels may be obtained.

One option for system combination is to combine classifiers at the kernel level. This paper examined a number of refinements to a recently proposed maximum-margin based scheme for learning a suitable kernel weighting. The scheme has a known tendency towards sparse weightings, which may not be optimal for speaker verification. A regularisation term was proposed allowing the user to tune the sparsity by adjusting a single parameter. Tying of kernel weights over all speakers was also applied to increase the robustness of the parameter estimates.

Various combinations of dynamic kernels were evaluated using the NIST 2002 evaluation data. Both parametric and derivative kernels individually provided gains compared to the baseline GMM system. By combining multiple kernels based upon different generative model structures further gains were observed. The best performance achieved was 7.78% EER when all kernels were combined. This represented a gain of 0.21% compared to an equal-weight pairwise baseline. The focus of this paper has been to specifically examine combination of parametric and derivative kernels and to provide a general scheme for their combination, rather than present a state-of-the-art system. Incorporating score-normalisation and introducing further noise-robustness techniques is expected to provide additional gains. Combination using more diverse forms of kernel, such as MLLR or CAT kernels, is also expected to yield gains.

REFERENCES

[1] V. Wan and W. Campbell, "Support vector machines for speaker verification and identification," in *Proc. Neural Networks for Signal Processing X*, 2000.

[2] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP*, 2002.
[3] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *NIPS*, 1999.
[4] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, 2006.
[5] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A.Venkataramam, "MLLR transforms as features in speaker recognition," in *Interspeech*, 2005.
[6] H. Yang, Y. Dong, X. Zhao, L. Lu, and H.Wang, "Cluster adaptive training weights as features in SVM-based speaker verification," in *Interspeech*, 2007.
[7] W. Campbell, D. Sturim, W. Shen, D. Reynolds, and J. Navrtil, "The MIT-LL/IBM 2006 speaker recognition system: High-performance reduced-complexity recognition," in *Proc. ICASSP*, 2007.
[8] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the nist'99 1-speaker submissions," *Digital Signal Processing*, pp. 237–248, 2000.
[9] S. Sonnenburg, G. Rätsch, and C. Schäfer, "A general and efficient multiple kernel learning algorithm," *Advances in Neural Information Processing Systems*, 2005.
[10] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "More efficiency in multiple kernel learning," in *Proc. ICML*, 2007.
[11] D. Reynolds, "Speaker identification and verification using Gaussian mixture models," *Speech Communication*, vol. 17, pp. 91–105, 1995.
[12] ——, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.
[13] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," *IEEE Transactions Speech and Audio Processing*, 2004.
[14] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carraquillo, "Support vector machines for speaker and language recognition," *Computer Speech Language*, vol. 20, pp. 210–229, 2005.
[15] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama, "Dynamic time-alignment kernel in support vector machine," in *Advances in Neural Information Processing Systems*, 2001.
[16] M. Ferras, C. Leung, C. Barras, and J. L. Gauvain, "Constrained MLLR for speaker recognition," in *Proc. ICASSP*, 2007.
[17] M. J. F. Gales, "Cluster adaptive training of hidden markov models," *IEEE Transactions on Speech and Audio Processing*, 2000.
[18] M. Layton, "Augmented statistical models for classifying sequence data," Ph.D. dissertation, Cambridge University, 2006.
[19] M. J. F. Gales and M. Layton, "Training augmented models using SVMs," *IEICE Special Issue on Statistical Models for Speech Recognition*, 2006.
[20] C. Longworth and M. Gales, "Derivative and parametric kernels for speaker verification," in *Proc. ICSLP*, 2007.
[21] P. Moreno, P. Ho, and B. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," *Advances in Neural Information Processing Systems*, 2004.
[22] F. R. Bach, R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality and the SMO algorithm," in *Proc. ICML*, 2004.
[23] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. V. Leeuwen, P. Matejka, P. Schwarz, and A. Straheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions of Audio, Speech and Language Processing*, vol. 15, no. 7, 2007.
[24] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf, C. Burges and A. Smola, Ed. MIT Press, 1999.
[25] A. Martin, "The NIST year 2002 speaker recognition evaluation plan," 2002, available from http://www.nist.gov/speech/tests/spk/2002/doc.
[26] C. Barras and J. Gauvain, "Feature and score normalization for speaker verification of cellular data," in *Proc. ICASSP*, 2003.
[27] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. ISCA Workshop on Speaker Recognition - 2001: A Speaker Oddyssey*, 2001.
[28] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. L. G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*, Cambridge, UK, 2006.
[29] M. Hebert and D. Boies, "T-norm for text-dependent commercial speaker verification applications: Effect of lexical mismatch," in *Proc. ICASSP*, 2005.
[30] A. O. Hatch, , S.Kajarekar, and A. Stolcke, "Within-class covariance normalisation for SVM-based speaker recognition," in *Interspeech*, 2006.
[31] C. Longworth and M. Gales, "Multiple kernel learning for speaker verification," in *Proc. ICASSP*, 2008.

PLACE PHOTO HERE

Chris Longworth (M'08) received the BSc degree in Computer Science with Artificial Intelligence from Royal Holloway, University of London, UK in 2004 and the M.Phil degree in Computer Speech, Text and Internet Technology from Cambridge University, Cambridge, U.K in 2005.

He is currently pursuing the Ph.D. degree at the University of Cambridge where he is a member of Christ's College. His research interests include kernel methods and their application to speech and speaker recognition.

PLACE PHOTO HERE

Mark J.F. Gales (M'01) received the B.A. degree in electrical and information sciences and the Ph.D degree from the University of Cambridge, Cambridge, UK, in 1988 and 1995, respectively.

Following graduation, he worked as a Consultant at Roke Manor Research, Ltd. In 1991, he took up a position as a Research Associate in the Speech Vision and Robotics Group, Engineering Department, Cambridge University. From 1995 to 1997, he was a Research Fellow at Emmanuel College, Cambridge. He was then a Research Staff Member in the Speech Group, IBM T.J. Watson Research Center, Yorktown Heights, NY until 1999 when he returned to the Engineering Department, Cambridge University, as a University Lecturer. He is currently a Reader in Information Engineering and a Fellow of Emmanuel College.

Dr. Gales was a member of the Speech Technical Committee from 2001 to 2004.