# A Generalised Derivative Kernel for Speaker Verification

*C. Longworth and M.J.F. Gales*

Engineering Department, Cambridge University
Trumpington St, Cambridge, CB2 1PZ
{cl336,mjfg}@eng.cam.ac.uk

## Abstract

An important aspect of SVM-based speaker verification systems is the choice of dynamic kernel. For the GLDS kernel, a static kernel is used to map each observation into a higher order feature space. Features are then obtained by taking a simple average over all frames. Derivative kernels, such as the Fisher kernel, use a generative model as a principled way of extracting a fixed set of features from each utterance. However, the model and features are defined using the original observations. Here, a dynamic kernel is described that combines these two approaches. In general, it is not possible to explicitly train a model in the feature space associated with a static kernel. However, by using a suitable metric with approximate component posteriors, this form of dynamic kernel can be computed. This kernel generalises the GLDS and derivative kernel as special cases and is also closely related to parametric kernels such as the GMM-supervector kernel. Preliminary results using this kernel are presented on the 2002 NIST SRE dataset.

**Index Terms**: Speaker Verification, Support Vector Machines, Dynamic kernels.

## 1. Introduction

Speaker verification (SV) is a binary classification task in which the objective is to determine whether or not a speech utterance was spoken by a specific claimed speaker. There has been considerable interest and success in applying Support Vector Machines (SVMs) to this task. SVMs are general purpose classifiers that have been found to perform well on a wide range of classification tasks. Many kernels used with SVMs only handle data of fixed dimensionality allowing the inner-product, or associated *static kernel*, to be simply computed. In contrast, speech utterances are typically parameterised as variable length sequences of observation vectors. This has led to the use of *dynamic kernels*, also known as sequence kernels. These dynamic kernels map variable length sequences into a fixed dimensionality in which the inner product, or static kernel, can be computed.

One early form of dynamic kernel that was found to be effective for the SV task is the Generalised Linear Discriminant Sequence (GLDS) kernel [1, 8]. Under this kernel, a static kernel mapping is applied to each observation vector. A fixed dimensional set of features is then obtained by taking the sum of the expanded observations over all frames. However, this has an averaging effect and useful information may be lost. More recent approaches [2] [3] have examined dynamic kernels based upon generative models. Many kernels, such as the Fisher kernel [4], belong to the family of derivative kernels [5]. Given a generative model, such as a Gaussian Mixture Model (GMM), the features extracted are the derivatives of the utterance log-likelihood with respect to the model parameters. Unlike the GLDS kernel, for each feature the observations are weighted by the component occupancy. This effectively uses the generative model to extract structure from the utterance.

This paper introduces a kernel that combines static and dynamic kernel approaches. The generalised derivative kernel (GDK) is a form of derivative kernel where, instead of calculating derivatives in the original observation space, the derivatives are calculated in the feature space associated with a static kernel defined between pairs of observations. Verification may be based on higher-order observation level features while exploiting the nature of the generative model to obtain a fixed set of features. This kernel generalises both the standard derivative kernel and also the GLDS kernel, when the model is a standard Gaussian distribution, providing a theoretical link between the two forms of kernel. In general, it is not possible to explicitly train a generative model in the feature space associated with a static kernel. However, such a model can be approximated. In this paper, a kernel metric is selected that normalises the variance term from the features, avoiding explicit calculation of these parameters. Furthermore, the component posteriors in the feature space are approximated by posteriors derived in the observation space. This paper is organised as follows. In the next section, SVM-based speaker verification is described and two forms of dynamic kernel, the GLDS kernel and the family of generative kernels, are briefly reviewed. Next the GDK is described and various required approximations are discussed. In section 4 results are presented for the GDK on the NIST 2002 SRE dataset. Finally, conclusions are drawn.

## 2. SVM-based Speaker Verification

SVMs have been successfully applied to a wide range of machine learning problems. One reason for this is that they can be kernelised. In SVM training and inference all references to data are in the form of inner-products between data examples. It is then possible to define 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*. Standard forms of kernel, such as the polynomial or Gaussian kernel, have been found to provide gains over linear kernels on a range of tasks. One issue when applying SVMs to speech processing tasks is that most standard forms of kernel only operate on data of fixed dimensionality. However, speech utterances are typically variable length sequences $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$. This has led to the development of dynamic kernels. These kernels operate on sequences and have the form

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = <\boldsymbol{\phi}(\boldsymbol{O}_i), \boldsymbol{\phi}(\boldsymbol{O}_j)> \quad (1)$$

Chris Longworth was funded by the Schiff Foundation.

Here $\phi(\boldsymbol{O})$ is a function that maps a sequence of observations to a fixed dimensional vector. To distinguish between the two forms of kernel, kernel functions that operate on fixed dimensional vectors will be referred to as *static kernels* in this paper. Static kernel functions are denoted $k(\boldsymbol{o}_i, \boldsymbol{o}_j)$ and dynamic kernel functions are denoted $K(\boldsymbol{O}_i, \boldsymbol{O}_j)$. For both static and dynamic kernels, the form of kernel function also defines the distance metric between two feature vectors. One such metric that is maximally non-committal is

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = \phi(\boldsymbol{O}_i)^{\mathsf{T}} \boldsymbol{Q}^{-1} \phi(\boldsymbol{O}_j) \quad (2)$$

$$\boldsymbol{Q} = \mathcal{E}\left\{(\phi(\boldsymbol{O}) - \boldsymbol{\mu}_\phi)(\phi(\boldsymbol{O}) - \boldsymbol{\mu}_\phi)^{\mathsf{T}}\right\} \quad (3)$$

$$\boldsymbol{\mu}_\phi = \mathcal{E}\{\phi(\boldsymbol{O}) \quad (4)$$

where $\mathcal{E}\{\}$ is the expectation with respect to $\boldsymbol{O}$. Alternatively an identity metric, $\boldsymbol{Q} = \boldsymbol{I}$, may be used. A number of dynamic kernels of this form have been proposed for speaker verification. The GLDS kernel and the family of generative kernels are briefly reviewed in the following sections.

### 2.1. GLDS Kernel

The Generalised Linear Discriminant Sequence (GLDS) kernel [1] was one of the earliest forms of dynamic kernels successfully applied to speaker verification. The GLDS kernel effectively maps each observation $\boldsymbol{o}_t$ into a feature-space $\Psi(\boldsymbol{o}_t)$. A duration-independent fixed-dimensional vector is then obtained by taking the mean of the expanded observations $\phi(\boldsymbol{O}) = \frac{1}{T}\sum_{t=1}^{T}\Psi(\boldsymbol{o}_t)$. Typically an identity metric is used. The kernel function is defined by taking the inner product of the means

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = \frac{1}{T_i T_j}\sum_{t=1}^{T_i}\sum_{s=1}^{T_j} k(\boldsymbol{o}_{it}, \boldsymbol{o}_{js}) \quad (5)$$

Standard forms of static kernel such as polynomial or Gaussian kernels may be applied. One disadvantage to using this form of kernel is that the static kernel function must be calculated between all pairs of observations. For longer utterances this can be computationally expensive. When $k(\boldsymbol{o}_i, \boldsymbol{o}_j)$ is linear the GLDS kernel may be simplified to

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) =< \boldsymbol{\mu}_i, \boldsymbol{\mu}_j > \quad (6)$$

where $\boldsymbol{\mu}_i$ is the mean of the observations in $\boldsymbol{O}_i$.

### 2.2. Generative kernels

Generative kernels are a form of dynamic kernel where a generative model is used to map utterances into a fixed dimensional space. For parametric kernels, such as the GMM-supervector kernel [3], the features $\phi_{\boldsymbol{\lambda}}(\boldsymbol{O})$ are the parameters $\boldsymbol{\lambda}$ of a generative model trained to represent $\boldsymbol{O}$. Thus

$$\phi_{\boldsymbol{\lambda}}(\boldsymbol{O}) = \left[\hat{\boldsymbol{\lambda}}\right], \quad \hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}}\{\log p(\boldsymbol{O}; \boldsymbol{\lambda})\} \quad (7)$$

Parametric kernels are related to the GLDS kernel. When $\boldsymbol{\lambda}$ represents the means associated with a single-component GMM the kernel has a form equivalent to Equation 6.

Many generative kernels, such as the Fisher kernel [4] and log-likelihood ratio kernel [6], belong to the family of derivative kernels [5]. For an utterance, $\boldsymbol{O}$, the features associated with a derivative kernel are the partial derivatives of the log-likelihood of $\boldsymbol{O}$ with respect to the parameters of a generative model $\boldsymbol{\lambda}$. The feature space has the form

$$\phi_{\nabla}(\boldsymbol{O}; \boldsymbol{\lambda}) = \left[\nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{O}; \boldsymbol{\lambda})\Big|_{\hat{\boldsymbol{\lambda}}}\right] \quad (8)$$

where $\hat{\boldsymbol{\lambda}}$ is the model parameter value at which the derivative is evaluated. Derivatives with respect to the means of a GMM can be used [7]. Here

$$\nabla_{\boldsymbol{\mu}_m} \log p(\boldsymbol{O}; \boldsymbol{\lambda})\Big|_{\boldsymbol{\lambda}} = \frac{1}{\rho^m}\sum_{t=1}^{T}\gamma_m(t)\boldsymbol{\Sigma}_m^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_m) \quad (9)$$

where $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m^{-1}$ are the mean and variance parameters associated with component $m$. Equation 9 includes an optional term $\rho^m$ to provide duration normalisation. This may be set to the number of frames $T$ in $\boldsymbol{O}$, or to the component occupancy, $\rho^m = \sum_{t=1}^{T}\gamma_m(t)$, where $\gamma_m(t)$ is the posterior probability of $\boldsymbol{o}_t$ being emitted by component $m$. This is important if the utterances in the dataset vary greatly in duration. Derivative kernels are also related to the GLDS kernel. When the model associated with the derivative kernel is a single component GMM with zero-mean and unit variance the kernel has a form equivalent to Equation 6. However, generally these kernels operate in very different ways.

## 3. Generalised derivative kernel

The GLDS kernel exploits a static kernel to explicitly map each observation into a more separable feature space. However, by taking a direct sum over all observations useful information may be 'averaged out'. Also, the resultant features may lack robustness to intermittent noise or long regions of silence. In contrast, generative kernels, such as the derivative kernel, provide a well-motivated mapping to a fixed dimensional set of features. However typically only a simple inner-product is calculated in the dynamic feature space. Combining static and dynamic approaches may therefore yield gains. This may be achieved in two ways. A dynamic function $\phi(\boldsymbol{O})$ may be used to map utterances into a fixed dimensional space, a static kernel can then be applied in this space. Alternatively, like the GLDS kernel, a static kernel may be defined at the level of individual observations. A model $\tilde{\boldsymbol{\lambda}}$ can then be defined in the associated static feature space.

The *generalised derivative kernel* (GDK) follows this second approach. Here, the features are derivatives with respect to the parameters of a model defined in the feature space. For GMMs, derivatives with respect to each feature space component mean are defined by

$$\nabla_{\tilde{\boldsymbol{\mu}}_m} \log p(\boldsymbol{O}; \tilde{\boldsymbol{\lambda}})\Big|_{\tilde{\boldsymbol{\lambda}}} = \frac{1}{\rho^m}\sum_{t=1}^{T}\tilde{\boldsymbol{\Sigma}}_m^{-1}\tilde{\gamma}_m(t)(\Psi(\boldsymbol{o}_t) - \tilde{\boldsymbol{\mu}}_m) \quad (10)$$

where $\tilde{\boldsymbol{\mu}}_m$ and $\tilde{\boldsymbol{\Sigma}}$ are the mean and covariance matrix respectively associated with component $m$ of the feature space GMM. $\tilde{\gamma}_m(t)$ is the posterior probability that $\Psi(\boldsymbol{o}_t)$ was emitted by component $m$ of this GMM. When the feature space consists of only mean derivatives, the kernel function has the form

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) = \sum_{m=1}^{M}\frac{1}{\rho_i^m\rho_j^m}\sum_{t=1}^{T_i}\sum_{s=1}^{T_j}\tilde{\gamma}_m(t)\tilde{\gamma}_m(s)f_m(\boldsymbol{o}_{it}, \boldsymbol{o}_{js}) \quad (11)$$

$$f_m(\boldsymbol{o}_i, \boldsymbol{o}_j) = [\Psi(\boldsymbol{o}_i) - \tilde{\boldsymbol{\mu}}_m]^{\mathsf{T}}\tilde{\boldsymbol{\Sigma}}_m^{-1}\tilde{\boldsymbol{Q}}_m^{-1}\tilde{\boldsymbol{\Sigma}}_m^{-1}[\Psi(\boldsymbol{o}_j) - \tilde{\boldsymbol{\mu}}_m] \quad (12)$$

When the static kernel function is linear, the GDK has the form of the standard derivative kernel in Equation 8. In the case when the model is a single-component GMM with zero mean and unit variance, the GDK has the form of the GLDS kernel in Equation 5. Thus both the GLDS and derivative kernels are special cases of the GDK.

Evaluating Equation 11 requires training a generative model in the static feature space. In general, this is not possible and approximations must be used. Two key issues are how to obtain suitable component posteriors and how to estimate $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$. These are discussed in the following sections.

### 3.1. Component posterior estimation

To estimate Equation 11, a kernel function must be computed between each pair of observations. For long utterances this may be infeasible. A more efficient approach is to perform a Viterbi alignment of each observation to a component. Here the kernel function is approximated by

$$K(\boldsymbol{O}_i, \boldsymbol{O}_j) \approx \sum_{m=1}^{M} \frac{1}{\rho_i^m \rho_j^m} \sum_{t \in \tilde{S}_i^{(m)}} \sum_{s \in \tilde{S}_j^{(m)}} f_m(\boldsymbol{o}_{it}, \boldsymbol{o}_{js}) \quad (13)$$

where $t \in \tilde{S}_i^{(\hat{m})}$ if $\hat{m} = \arg\max_m \tilde{\gamma}_m(t)$. GMMs tend towards hard component alignments as the dimensionality of the space increases, therefore this approximation will be more robust when the dimensionality of $\Psi(\boldsymbol{o})$ is large. This is related to an approach used in [8], where a kernel function is only computed between a frame and its 'closest' frame in the other sequence. Here the generative model is used as a principled approach to identify sets of close frames.

An important issue is how to obtain $\tilde{\gamma}_m(t)$ when a generative model can not be trained in the feature space. One approach is to use the posteriors from a model trained in the observation space, $\boldsymbol{\lambda}$. Thus

$$\tilde{S}_j^{(m)} \approx S_j^{(m)} \text{ where } t \in S_i^{(\hat{m})} \text{ if } \hat{m} = \arg\max_m \gamma_m(t) \quad (14)$$

where $\gamma_m(t)$ are the posteriors associated with $\boldsymbol{\lambda}$. This may yield poor estimates when relative distances between observations differ greatly between the feature and input spaces.

### 3.2. Static kernel estimation

In Equation 12, $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$ is a function of the feature space variance $\tilde{\boldsymbol{\Sigma}}_m$ and the metric $\tilde{\boldsymbol{Q}}_m$ applied to the derivatives associated with component $m$. Generally a generative model cannot be trained in the feature space, nor the feature-space $\Psi(\boldsymbol{o})$ explicitly generated. Thus it is not possible to directly obtain these parameters. Schemes such as in [9] can be used when $\Psi(\boldsymbol{o})$ does not have an explicit representation. Here $\tilde{\boldsymbol{\Sigma}}_m^{-1}\tilde{\boldsymbol{Q}}_m^{-1}\tilde{\boldsymbol{\Sigma}}_m^{-1}$ is approximated by the identity matrix. This allows standard forms of static kernel to be used and avoids explicitly estimating $\tilde{\boldsymbol{\Sigma}}_m^{-1}$ and $\tilde{\boldsymbol{Q}}_m$. This approximation is likely to be more robust when observations are globally whitened. For the case when $\tilde{\boldsymbol{\mu}}_m$ represents the ML estimate over a set of background observations $\{\boldsymbol{o}_{\mathtt{B}1}, \dots, \boldsymbol{o}_{\mathtt{B}T_B}\}$, $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$ is approximated by

$$f_m(\boldsymbol{o}_i, \boldsymbol{o}_j) \approx k(\boldsymbol{o}_i, \boldsymbol{o}_j) - k_\mu^m(\boldsymbol{o}_i) - k_\mu^m(\boldsymbol{o}_j) + k_{\mu\mu}^m \quad (15)$$

$$k_\mu^m(\boldsymbol{o}) = \frac{1}{C_{\mathtt{B}}^{(m)}} \sum_{t \in S_{\mathtt{B}}^{(m)}} k(\boldsymbol{o}, \boldsymbol{o}_{\mathtt{B}t}) \quad (16)$$

$$k_{\mu\mu}^m = \frac{1}{\left(C_{\mathtt{B}}^{(m)}\right)^2} \sum_{s,t \in S_{\mathtt{B}}^{(m)}} k(\boldsymbol{o}_{\mathtt{B}s}, \boldsymbol{o}_{\mathtt{B}t}) \quad (17)$$

where $C_{\mathtt{B}}^{(m)}$ is the number of frames in $\boldsymbol{O}_{\mathtt{B}}$ aligned with component $m$. Calculating $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$ requires evaluating a static kernel function between each observation and the entire background dataset. When $T_B$ is large, this may not be feasible. Instead, the mean normalisation in the feature space, $\Psi(\boldsymbol{o}) - \tilde{\boldsymbol{\mu}}_m$, can be approximated by a normalisation evaluated in the observation space $\Psi(\boldsymbol{o} - \boldsymbol{\mu}_m)$. Again, when distances vary greatly between the observation and feature space, this approximation may not be robust. In this paper, $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$ is approximated by

$$f_m(\boldsymbol{o}_i, \boldsymbol{o}_j) \approx k([\boldsymbol{o}_i - \boldsymbol{\mu}_m], [\boldsymbol{o}_j - \boldsymbol{\mu}_m]) \quad (18)$$

### 3.3. Relationship to parametric kernels

Derivative and parametric kernels are known to be related and, under certain conditions discussed in [5], the functions obtained will be identical. A similar approach may be used to approximate a parametric kernel, where the model is defined in the feature space of a static kernel. If the feature space consists of GMM means only, an identity metric is used, and hard component alignments are estimated using a linear model $\boldsymbol{\lambda}$, the kernel will have the form

$$K_{\boldsymbol{\lambda}}(\boldsymbol{O}_i, \boldsymbol{O}_j) \approx \frac{1}{C_i^{(m)} C_j^{(m)}} \sum_{m=1}^{M} \sum_{t \in S_i^{(m)}} \sum_{s \in S_j^{(m)}} k(\boldsymbol{o}_{it}, \boldsymbol{o}_{js}) \quad (19)$$

This is similar to Equation 13 when component-occupancy normalisation is used. If the GDK is computed using the approximation in Equation 18, and $k(\boldsymbol{o}_i, \boldsymbol{o}_j)$ is stationary, the two forms of kernel will be identical.

## 4. Experimental results

Various forms of kernel were evaluated on the 2002 NIST SRE one-speaker detection task [10]. This task contains cellular data with speech from 139 male and 191 female speakers. Each utterance was parameterised using a frame rate of 10ms and a window size of 30ms. 31 features were extracted per frame, these consisted of 15 static, 15 delta Mel-PLP coefficients and the delta energy. Initially SVM classifiers were evaluated using the GLDS kernel defined in Equation 5. For each speaker, an imposter training set was created using all the non-speaker enrollment utterances. For this experiment, spectral-based normalisation techniques were not used since they typically normalise each utterance mean to zero. A maximally non-committal metric was also not used since over long utterances the mean of the ΔPLP coefficients tends to zero. Using a linear static kernel, the Equal Error Rate (EER) of the GLDS kernel was 23.74%. When a second-order polynomial kernel was used this dropped to 22.40%. This gain is partly due to the fact that the SVM is able to use information from the delta coefficients.

SVM classifiers using generalised derivative kernels were also evaluated. For certain limited forms of static kernel, such as linear and low-order polynomial kernels, it is possible to evaluate the GDK kernel defined in Equation 11 by explicitly training a generative model in the static feature space. Cepstral feature warping [11] was performed on each utterance using a three second window to introduce additional robustness to channel noise. Each speech observation was then explicitly mapped into the corresponding feature space. For each form of static kernel, gender-dependent, diagonal-covariance GMM, UBMs were trained by EM using all SRE 2002 enrollment utterances of the appropriate gender[1]. Speaker-dependent GMMs

[1]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.

were constructed by MAP adapting the means of the appropriate gender-dependent UBM. A diagonal approximation was used for $\boldsymbol{Q}$, estimated based on the covariance matrix features extracted from the enrollment data. Derivatives were normalised by the component occupancy.

| System | Linear | | Polynomial | |
|---|---|---|---|---|
| | 16 | 128 | 16 | 128 |
| `GMM-LLR` | 17.40 | 12.04 | 16.40 | 14.52 |
| `Explicit` | 12.14 | 8.38 | 10.32 | 10.94 |
| `Viterbi` | 12.30 | 9.15 | 10.38 | 11.11 |
| `Approx` | 12.21 | 10.20 | 9.25 | 10.29 |

Table 1: Performance of GDK systems based upon explicit models (`Explicit`), explicit models using Viterbi alignments (`Viterbi`) and using approximated models (`Approx`)

Table 1 shows the performance for linear and second-order homogeneous polynomial kernels, using an GDK-based SVM classifier for 16 and 128-component models. Baseline results, using a LLR classifier with the same models, are also presented. In all cases, the SVM outperformed the LLR classifier. In the linear case, LLR and SVM performance improved for larger model sizes. During preliminary experiments further small gains were observed up to 1024 components. By contrast, for the polynomial SVM classifier best performance was obtained using 16-component models. This difference is due to the significantly larger feature space associated with the polynomial kernel, 527 versus 31 features per component. For the given dataset, 128-component models were found to suffer from over-training. Although the best polynomial system did not outperform the 128-component linear system, further small gains were achieved when these two systems were combined. Applying a maximum-margin based Multiple Kernel Learning scheme [12] gave a performance of 8.08%.

To examine the effect on performance of using Viterbi component alignments, GDK systems (`Viterbi`) were trained as defined in Equation 13. Using hard alignments degraded the performance of all systems. The effect was less severe for the polynomial kernels, since for GMMs trained in a high-dimensional space, the component posteriors already tend towards hard alignments. Lastly, GDK systems (`Approx`) using approximated models were trained for these two forms of kernel. Here the component posteriors were obtained using the linear models, and $f_m(\boldsymbol{o}_i, \boldsymbol{o}_j)$ was approximated using Equation 18. For the linear kernels, performance was worse for this system, this is due to the approximation used for $\boldsymbol{Q}$. For the polynomial systems the performance improved indicating that the posteriors associated with the linear models were more robust.

| Kernel | 16 | 128 |
|---|---|---|
| Linear | 12.21 | 10.20 |
| Polynomial order 2 | 9.25 | 10.29 |
| Polynomial order 3 | 9.96 | 13.17 |
| Gaussian | 14.29 | 12.94 |

Table 2: GDK performance using 16 and 128-component models for various static kernels

Finally, the generalised derivative kernel was evaluated using other forms of static kernel. A third-order homogeneous polynomial kernel and a Gaussian kernel were used. For the Gaussian kernel $\sigma^2$ was set to 31, the dimensionality of the observations. For 128-components, no gains were observed using non-linear kernels due to the limited amount of available training data. When 16-component models were used both polynomial kernels gave gains over the linear case and also outperformed the 128-component linear system. Overall best performance was obtained using a second-order polynomial kernel with a 16-component model.

## 5. Conclusion

This paper has introduced a new form of dynamic kernel that combines a static kernel applied over individual observations with a generative model based approach to obtain a fixed dimensional representation of each utterance. By choosing a suitable metric, and using approximate component posteriors, this form of kernel may be computed. This dynamic kernel generalises both the derivative kernel and the GLDS kernel as special cases. The form of kernel obtained is also closely related to a parametric kernel where the models are defined in the static kernel feature-space. In preliminary experiments, using non-linear static kernels gave gains. For tasks where more training speech is available, further gains may be obtained using larger model sizes.

## 6. References


[1] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP*, 2002.

[2] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A.Venkataramam, "MLLR transforms as features in speaker recognition," in *Interspeech*, 2005.

[3] 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.

[4] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *NIPS*, 1999.

[5] C. Longworth and M. Gales, "Derivative and parametric kernels for speaker verification," in *Proc. ICSLP*, 2007.

[6] M. J. F. Gales and M. Layton, "Training augmented models using SVMs," *IEICE Special Issue on Statistical Models for Speech Recognition*, 2006.

[7] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," *IEEE Transactions Speech and Audio Processing*, 2004.

[8] J. Mariéthoz and S. Bengio, "A kernel trick for sequences applied to text-independent speaker verification systems," *Pattern Recognition*, vol. 40, pp. 2315–2324, 2007.

[9] J. Louradour, K. Daoudi, and F. Bach, "Feature space mahalanobis sequence kernels: Application to SVM speaker verification," *IEEE Transactions on Audio,Speech and Language Processing*, vol. 15, pp. 2465–2475, 2007.

[10] A. Martin, "The NIST year 2002 speaker recognition evaluation plan," 2002, available from http://www.nist.gov/speech/tests/spk/2002/doc.

[11] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. ISCA Workshop on Speaker Recognition - 2001: A Speaker Odyssey*, 2001.

[12] C. Longworth and M. Gales, "Multiple kernel learning for speaker verification," in *Proc. ICASSP*, 2008.