DISCRIMINATIVE CLASSIFIERS WITH ADAPTIVE KERNELS FOR NOISE ROBUST SPEECH RECOGNITION

M.J.F. Gales, F. Flego *

Cambridge University Engineering Department,
Trumpington Street, Cambridge,
CB2 1PZ United Kingdom

Abstract

Discriminative classifiers are a popular approach to solving classification problems. However one of the problems with these approaches, in particular kernel based classifiers such as Support Vector Machines (SVMs), is that they are hard to adapt to mismatches between the training and test data. This paper describes a scheme for overcoming this problem for speech recognition in noise by adapting the kernel rather than the SVM decision boundary. Generative kernels, defined using generative models, are one type of kernel that allows SVMs to handle sequence data. By compensating the parameters of the generative models for each noise condition noise-specific generative kernels can be obtained. These can be used to train a noise-independent SVM on a range of noise conditions, which can then be used with a test-set noise kernel for classification. The noise-specific kernels used in this paper are based on Vector Taylor Series (VTS) model-based compensation. VTS allows all the model parameters to be compensated and the background noise to be estimated in a maximum likelihood fashion. A brief discussion of VTS, and the optimisation of the mismatch function representing the impact of noise on the clean speech, is also included. Experiments using these VTS-based test-set noise kernels were run on the AURORA 2 continuous digit task. The proposed SVM rescoring scheme yields large gains in performance over the VTS compensated models.

Key words: speech recognition, noise robustness, support vector machines, generative kernels

* Corresponding author. Tel: (+44) 1223 332815, Fax: (+44) 1223 332662
Email addresses: `mjfg@eng.cam.ac.uk` (M.J.F. Gales), `ff257@eng.cam.ac.uk` (F. Flego).
URL: `http://mi.eng.cam.ac.uk/~mjfg` (M.J.F. Gales).

Preprint submitted to Computer Speech and Language *22 September 2009*

1 Introduction

Speech recognition is normally based on generative models, in the form of Hidden Markov Models (HMMs), and class priors, the language model. These are then combined using Bayes' rule to yield the class posteriors. Alternative approaches are to use discriminative models (Kuo and Gao, 2006; Gunawardana et al., 2005; Gales, 2007), which directly model the class posteriors, or discriminative functions such as Support Vector Machines (SVMs) (Vapnik, 1998), where the decision boundary is directly modelled. One of the problems with using these discriminative models and functions is that it is normally hard to adapt them to changing speakers or acoustic environments. This is particularly true of kernel based approaches, such as SVMs, where individual training examples are used to determine the decision boundaries. There has been some previous work on adapting margin classifier parameters to address this problem, see for example (Li and Bilmes, 2006; Huang et al., 2007) [1]. These approaches either use the existing SVM parameters as a form of prior for the samples from the new domain (Li and Bilmes, 2006), or define a resampling weight for the training samples to "match" the particular target domain (Huang et al., 2007). However for some speech recognition problems, for example noise-robust speech recognition and rapid speaker adaptation, it is not possible to ensure that the target speaker or environment is well covered by the training data. Furthermore the amount of data in the target domain may be very limited, for example a single utterance. For these forms of problem these approaches are not feasible.

An obvious application area where there are large mismatches between the training and test sets is speech recognition in noise. Handling changing acoustic conditions has been an active area of research for many years (Huang et al., 2001). Model-based compensation schemes (Gales, 1995; Moreno, 1996; Li et al., 2007) are a powerful approach to handling these mismatches. Well implemented model-based compensation schemes tend to out-perform feature-based compensation schemes as it is possible to more accurately model situations where speech is, for example, masked by the noise. This paper examines an approach that allows discriminative classifiers to be combined with model-based compensation schemes to improve the noise-robustness.

In this work, rather than attempting to modify the SVM itself, the form of the kernel is altered to reflect the changing acoustic conditions. For the class of kernels that make use of generative models (Jaakkola and Hausser, 1999; Smith and Gales, 2001), such as HMMs, the parameters associated with the kernel are simply the parameters of the generative model. Thus adapting

[1] In the machine learning literature this is sometimes referred to as sample selection bias or covariant shift.

generative kernels to a particular noise condition involves performing model-based compensation. This noise-specific generative kernel can then used by a noise-independent SVM for classification in the test noise condition. Provided the form of adapted kernel compensates for the effects of the environment changes, it should be possible to train (and classify with) a noise-independent SVM on a range of noise conditions with the appropriate noise dependent kernels.

Though this work is primarily interested in the feasibility of using noise-independent SVMs, it is important that a good model-based compensation scheme is used. In this work Vector Taylor Series (VTS) (Moreno, 1996) compensation using noise model parameters estimated on the test data (Liao and Gales, 2006) is implemented. In addition to a brief description of VTS model-based compensation, a discussion of recent research for improved performance by optimising the mismatch function is also included (Li et al., 2008).

This paper is organised as follows. The next section briefly reviews model-based compensation schemes and VTS model-based compensation. In addition recent approaches for improving VTS compensation by tuning the mismatch function are described. This is followed by a discussion of SVMs with dynamic kernels, based on generative models, for speech recognition. These dynamic, or sequence kernels, allow SVMs to be applied to sequence data such as speech. Section 4 then describes the complete scheme for using noise-independent SVMs. Results on the AURORA 2 database are given in section 5.

2 Model-Based Noise Compensation

There are a number of possible approaches to reduce the impact of changing background noise conditions and convolutional distortion on the performance of speech recognition systems. For a review of a number of approaches see (Huang et al., 2001). The approach examined in this work is often referred to as model-based compensation. Here the parameters of "clean" acoustic models, $\boldsymbol{\lambda}_{\mathsf{x}}$, are transformed to be representative of acoustic models trained in the target test condition, $\boldsymbol{\lambda}_{\mathsf{y}}$.

The first stage in producing a noise compensation scheme is to define the impact of the acoustic environment and channel on the clean speech data, the *mismatch function*. In the mel-cepstral domain [2] used in this work the following approximation between the static clean speech, noise and noise corrupted

[2] For this work the cepstral parameters are assumed to be based on "magnitude" filter-bin outputs. For some schemes, such as the default used in HTK (Young et al., 2006) and the ETSI features, where the filter-bin analysis is by default based on magnitude FFT values, the output may be directly used. For other work the square

speech observations is often used (log(.) and exp(.) indicate element-wise logarithm or exponential functions) (Acero, 1993)

$$\begin{aligned}\boldsymbol{y}_t^{\mathsf{s}} &= \boldsymbol{x}_t^{\mathsf{s}} + \boldsymbol{h} + \frac{1}{2}\mathbf{C}\log\left(1 + \exp(2\mathbf{C}^{\text{-}1}(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h}))\right)\\ &= \boldsymbol{x}_t^{\mathsf{s}} + \boldsymbol{h} + f(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h}) \qquad (1)\end{aligned}$$

where $\mathbf{C}$ is the Discrete Cosine Transform (DCT) matrix and $\boldsymbol{h}$ is the convolutional distortion or noise. For a given set of noise conditions, the observed (static) noise-corrupted speech vector at time t, $\boldsymbol{y}_t^{\mathsf{s}}$, is a highly non-linear function of the underlying clean (static) speech signal $\boldsymbol{x}_t^{\mathsf{s}}$, noise $\boldsymbol{n}_t^{\mathsf{s}}$ and convolutional noise $\boldsymbol{h}$. Noise compensation schemes are further complicated by the addition of dynamic parameters. The observation vector $\boldsymbol{y}_t$ is often formed of the static parameters appended by the delta and delta-delta parameters. Thus $\boldsymbol{y}_t^{\mathsf{T}} = \left[\boldsymbol{y}_t^{\mathsf{sT}}\ \Delta\boldsymbol{y}_t^{\mathsf{sT}}\ \Delta^2\boldsymbol{y}_t^{\mathsf{sT}}\right]$. Mismatch functions for all the parameters can be obtained (Gales, 1995; Gopinath et al., 1995).

The aim of model-based compensation schemes is to obtain the parameters of the noise-corrupted speech model from the clean speech and noise models. Most model-based compensation methods assume that if the speech and noise models are Gaussian then the combined noise-corrupted speech model will also be Gaussian. Thus to compute the expected value of the "observations" for each corrupted speech component m (assuming a single noise component) the following must be computed

$$\boldsymbol{\mu}_{\mathsf{y}}^{(m)} = \mathcal{E}\left\{\boldsymbol{y}|m\right\}; \quad \boldsymbol{\Sigma}_{\mathsf{y}}^{(m)} = \text{diag}\left(\mathcal{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}|m\right\} - \boldsymbol{\mu}_{\mathsf{y}}^{(m)}\boldsymbol{\mu}_{\mathsf{y}}^{(m)\mathsf{T}}\right) \qquad (2)$$

There is no simple closed-form solution to these equations so various approximations have been proposed. These include Parallel Model Combination (Gales, 1995) and Vector Taylor Series (VTS) (Moreno, 1996). As noise models are not normally available, these must also be estimated from the observed data. Schemes that allow all the model parameters to be estimated have been proposed (Moreno, 1996; Liao and Gales, 2007; Li et al., 2007).

In previous work on combining SVMs with model-based noise compensated generative kernels (Gales and Longworth, 2008), an idealised version of these model-based compensation schemes was used, i.e. Single-Pass Retraining (SPR) (Gales, 1995).

In this paper a more practical compensation scheme based on VTS is used.

root of the filter-bin outputs must be used. Of course this complicates the process as the front-end processing must be clearly defined so that the appropriate mismatch function can be applied.

This is briefly described in the next section and followed by a discussion of recent work on optimising the mismatch function for improved recognition.

2.1 Vector Taylor Series Compensation

Vector Taylor series model-based compensation is a popular approach for model-based compensation (Moreno, 1996; Acero et al., 2000; Liao and Gales, 2007; Li et al., 2007). A number of possible forms have been examined in the literature. In this work the first-order VTS scheme described in (Liao and Gales, 2006) is used. A brief summary of the scheme is given here. The first-order VTS expansion of the corrupted speech in equation 1 may be expressed as

$$\begin{aligned}\boldsymbol{y}_t^{\mathsf{s}} \approx\; & \overline{\boldsymbol{x}}^{\mathsf{s}} + \overline{\boldsymbol{h}} + f(\overline{\boldsymbol{n}}^{\mathsf{s}} - \overline{\boldsymbol{x}}^{\mathsf{s}} - \overline{\boldsymbol{h}}) \\ & + (\boldsymbol{x}_t^{\mathsf{s}} - \overline{\boldsymbol{x}}^{\mathsf{s}})\frac{\partial f}{\partial \boldsymbol{x}^{\mathsf{s}}} + (\boldsymbol{n}_t^{\mathsf{s}} - \overline{\boldsymbol{n}}^{\mathsf{s}})\frac{\partial f}{\partial \boldsymbol{n}^{\mathsf{s}}} + (\boldsymbol{h} - \overline{\boldsymbol{h}})\frac{\partial f}{\partial \boldsymbol{h}}\end{aligned} \tag{3}$$

where the partial derivatives are evaluated at the expansion point $\{\overline{\boldsymbol{x}}^{\mathsf{s}}, \overline{\boldsymbol{n}}^{\mathsf{s}}, \overline{\boldsymbol{h}}\}$. This approximation can be used to obtain the noise corrupted model parameters. To compute the mean for a particular corrupted speech component, the expansion point is set to the means of the speech, additive and convolutional noise distributions for that component. Thus the static mean of the corrupted speech distribution (Acero et al., 2000), $\boldsymbol{\mu}_{\mathsf{y}}^{\mathsf{s}}$, is given by[3]

$$\begin{aligned}\boldsymbol{\mu}_{\mathsf{y}}^{\mathsf{s}} = \mathcal{E}\Big\{ & \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}} + \boldsymbol{\mu}_{\mathsf{h}} + f(\boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{h}}) \\ & + (\boldsymbol{x}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}})\frac{\partial f}{\partial \boldsymbol{x}^{\mathsf{s}}} + (\boldsymbol{n}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{s}})\frac{\partial f}{\partial \boldsymbol{n}^{\mathsf{s}}} + (\boldsymbol{h} - \boldsymbol{\mu}_{\mathsf{h}})\frac{\partial f}{\partial \boldsymbol{h}}\Big\}\end{aligned} \tag{4}$$

where the expectation is over the clean speech and noise distributions which have mean and variances of: $\boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}}$, $\boldsymbol{\Sigma}^{\mathsf{s}}{}_{\mathsf{x}}$ for the clean speech model; $\boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{s}}$, $\boldsymbol{\Sigma}_{\mathsf{n}}^{\mathsf{s}}$ for the additive noise model; and $\boldsymbol{\mu}_{\mathsf{h}}, \boldsymbol{\Sigma}_{\mathsf{h}}$ for the convolutional distortion. The partial derivatives above can be expressed in terms of partial derivatives of $\boldsymbol{y}^{\mathsf{s}}$ with respect to $\boldsymbol{x}^{\mathsf{s}}$, $\boldsymbol{n}^{\mathsf{s}}$ and $\boldsymbol{h}$ evaluated at $\boldsymbol{\mu} = \boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{h}}$. These have the form

$$\partial \boldsymbol{y}^{\mathsf{s}}/\partial \boldsymbol{x}^{\mathsf{s}} = \partial \boldsymbol{y}^{\mathsf{s}}/\partial \boldsymbol{h} = \mathbf{A} \tag{5}$$

$$\partial \boldsymbol{y}^{\mathsf{s}}/\partial \boldsymbol{n}^{\mathsf{s}} = \mathbf{I} - \mathbf{A} \tag{6}$$

where $\mathbf{A} = \mathbf{CFC}^{\text{-1}}$ and $\mathbf{F}$ is a diagonal matrix with elements given by $1/(1 + \exp(2\mathbf{C}^{\text{-1}}\boldsymbol{\mu}))$. It follows that the means and variances of the noisy speech are given by

[3] The dependence of the noise corrupted speech mean and clean speech mean on the component have been dropped for clarity.

$$\boldsymbol{\mu}_{\mathsf{y}}^{\mathsf{s}} = \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}} + \boldsymbol{\mu}_{\mathsf{h}} + f(\boldsymbol{\mu}_{\mathsf{n}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{x}}^{\mathsf{s}} - \boldsymbol{\mu}_{\mathsf{h}}) \tag{7}$$

$$\boldsymbol{\Sigma}_{\mathsf{y}}^{\mathsf{s}} = \text{diag}\left(\mathbf{A}\boldsymbol{\Sigma}_{\mathsf{x}}^{\mathsf{s}}\mathbf{A}^{\mathsf{T}} + (\mathbf{I} - \mathbf{A})\boldsymbol{\Sigma}_{\mathsf{n}}^{\mathsf{s}}(\mathbf{I} - \mathbf{A})^{\mathsf{T}}\right) \tag{8}$$

Note the convolutional distortion is assumed to have zero variance, $\boldsymbol{\Sigma}_{\mathsf{h}} = \mathbf{0}$.

The VTS compensation based on equation 3 only allows the static parameters of the acoustic models to be compensated. For best performance the dynamic, delta and delta-delta, parameters must also be compensated (Gales, 1995). A commonly used approximation for this is the *continuous time approximation* (Gopinath et al., 1995). Here the dynamic parameters, which are a discrete time estimate of the gradient, are approximated by the instantaneous derivative with respect to time

$$\Delta \boldsymbol{y}_t^{\mathsf{s}} = \frac{\sum_{i=1}^{n} w_i \left(\boldsymbol{y}_{t+i}^{\mathsf{s}} - \boldsymbol{y}_{t-i}^{\mathsf{s}}\right)}{\sum_{i=1}^{n} w_i^2} \approx \frac{\partial \boldsymbol{y}_t^{\mathsf{s}}}{\partial t} \tag{9}$$

It is then possible to show that, for example, the mean of the delta parameters $\boldsymbol{\mu}_{\mathsf{y}}^{\Delta}$ can be expressed as

$$\boldsymbol{\mu}_{\mathsf{y}}^{\Delta} = \mathbf{A}\boldsymbol{\mu}_{\mathsf{x}}^{\Delta}. \tag{10}$$

The variances and delta-delta parameters can also be compensated in this fashion. It is possible to improve on the continuous-time dynamic compensation mismatch function in equation 9. A form based on simple-difference rather than regression-based dynamic parameters can be used (Gales, 1995). Alternatively, regression-based parameters based on an explicit transformation of a window of static features have been shown to outperform the continuous-time approximation (van Dalen and Gales, 2008). These forms of improved dynamic parameter approximation are not investigated in this work, but can be combined with the approaches described.

The compensation schemes described above have assumed that the noise model parameters, $\boldsymbol{\mu}_{\mathsf{n}}$, $\boldsymbol{\Sigma}_{\mathsf{n}}$ and $\boldsymbol{\mu}_{\mathsf{h}}$, are known. In practice these are seldom known in advance so must be estimated from the test data. In this work the noise estimation is based on the Maximum Likelihood (ML) noise estimation scheme described in (Liao and Gales, 2007). In addition, the second-order approach for the noise variance in (Li et al., 2007) was implemented. This was found to have no effect on recognition performance, but improved the speed of noise model parameter estimation.

2.2 Mismatch Function Optimisation

There are a number of assumptions required to obtain the form of the mismatch function given in equation 1, see for example (Gales, 1995). Recently there has been interest in modifying the form of the mismatch function to reduce the impact of these approximations, or more generally to yield improved recognition performance.

One of the assumptions that has been investigated for VTS is the use of a phase-sensitive mismatch function (Li et al., 2008). By relaxing the assumption that there is sufficient smoothing to remove all cross-terms, a function of the following form can be obtained

$$\begin{aligned} \boldsymbol{y}_t^{\mathsf{s}} = \boldsymbol{x}_t^{\mathsf{s}} + \boldsymbol{h} + \frac{1}{2}\mathbf{C}\log\Big(1 + \exp(2\mathbf{C}^{\text{-}1}(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h})) \\ +2\alpha\exp(\mathbf{C}^{\text{-}1}(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h}))\Big) \end{aligned} \tag{11}$$

where α is related to the level of smoothing. Note in theory α should be a function of the mel-bin value (Deng et al., 2004). In (Li et al., 2008) the value of α was tuned to minimise the error rate on the AURORA 2 task (Hirsch and Pearce, 2000). Significant gains over the baseline $\alpha = 0$ configuration were obtained. However the best performance was obtained using $\alpha = 2.5$, contradicting the phase-sensitive theory where $-1 \leq \alpha \leq 1$. Thus this form of compensation may be considered as a generalisation of the mismatch function where α is optimised, rather than compensating for limitations in the mel-bin smoothing.

An alternative form of mismatch function generalisation was proposed in (Gales, 1995). This has the form

$$\boldsymbol{y}_t^{\mathsf{s}} = \boldsymbol{x}_t^{\mathsf{s}} + \boldsymbol{h} + \frac{1}{\gamma}\mathbf{C}\log\Big(1 + \exp(\gamma\mathbf{C}^{\text{-}1}(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h}))\Big) \tag{12}$$

By tuning γ various forms of mismatch function can be obtained. Interestingly equations 11 and 12 are the same when $\gamma = 1$ and $\alpha = 1$ (magnitude combination) and when $\gamma = 2$ and $\alpha = 0$ (power combination). For the magnitude case the mismatch function is

$$\boldsymbol{y}_t^{\mathsf{s}} = \boldsymbol{x}_t^{\mathsf{s}} + \boldsymbol{h} + \mathbf{C}\log\Big(1 + \exp(\mathbf{C}^{\text{-}1}(\boldsymbol{n}_t^{\mathsf{s}} - \boldsymbol{x}_t^{\mathsf{s}} - \boldsymbol{h}))\Big) \tag{13}$$

This is the form used in (Liao and Gales, 2006; Liao, 2007) as it was found to outperform $\gamma = 2$. It is also consistent with the results in (Li et al., 2007) where $\alpha = 1$ yielded the majority of the gains over the baseline $\alpha = 0$, though additional gains were obtained using $\alpha = 2.5$. In this work $\gamma = 1$ ($\alpha = 1$) will be used as the baseline static compensation mismatch function. This was found to give consistently better VTS performance than $\gamma = 2$. For the final ETSI experiments in section 5.2, γ was optimised to minimise the Word Error Rate (WER). This will be referred to as a γ-optimised (γ-opt) system.

Though the two forms of mismatch optimisation have been described in the

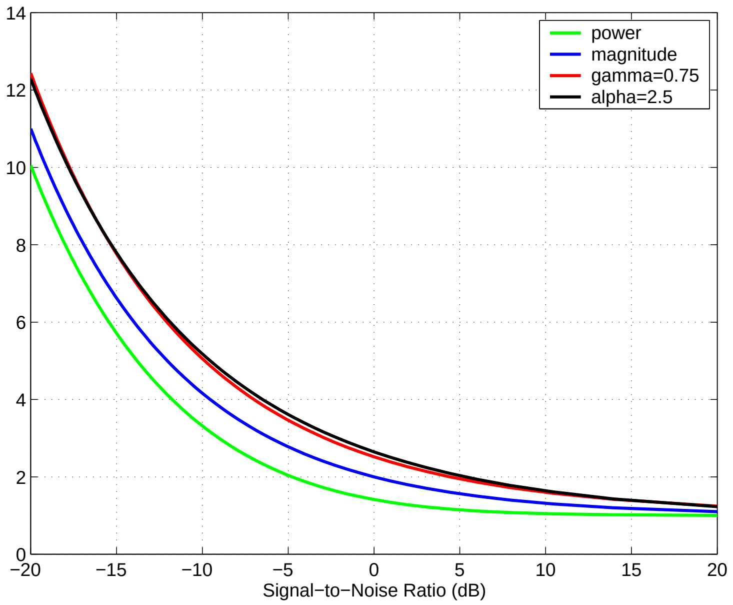

Fig. 1. $\mathcal{F}_\gamma(x,n)$ and $\mathcal{F}_\alpha(x,n)$ for power ($\alpha = 0, \gamma = 2$), magnitude ($\alpha = 1, \gamma = 1$), $\gamma = 0.75$ and $\alpha = 2.5$ against SNR in dB.

speech literature, there has been no systematic analysis of the relationship between the two. Here the two forms of mismatch function generalisation in equations 11 and 12 are compared. To simplify the comparison, normalised forms of the two equations were used based on the ratio of the corrupted speech magnitude to the clean speech magnitude in the linear spectral domain (convolutional distortion is ignored). Thus the functions for γ-optimisation, $\mathcal{F}_\gamma(x,n)$, and phase-sensitive, $\mathcal{F}_\alpha(x,n)$, mismatch functions are given by

$$\mathcal{F}_\gamma(x,n) = \left(1 + \left(\frac{n}{x}\right)^\gamma\right)^{1/\gamma} \tag{14}$$

$$\mathcal{F}_\alpha(x,n) = \left(1 + \left(\frac{n}{x}\right)^2 + 2\alpha\left(\frac{n}{x}\right)\right)^{1/2} \tag{15}$$

where n is the noise and x is the clean speech in the linear spectral domain. Figure 1 shows the γ-optimisation function for $\gamma = 0.75$ (the optimal value found in section 5.2) and phase-sensitive function for $\alpha = 2.5$ (the optimal value in (Li et al., 2008)) as well as the magnitude ($\alpha = 1, \gamma = 1$) and power ($\alpha = 0, \gamma = 2$) functions against signal-to-noise ratio (SNR). It is interesting that the best γ-optimised function and the best phase-sensitive α function show very similar trends as the SNR varies for these values of α and γ.

Given the close relationship between the two forms of mismatch function, and the more solid theoretical grounding, the γ-optimisation compensation scheme will be used in this paper.

3 SVMs and Generative Kernels

Support Vector Machines (SVMs) (Vapnik, 1998) are an approximate implementation of structural risk minimisation. SVMs are binary classifiers, where the decision boundary is estimated to maximise the margin, the distance from the decision boundary to the closest points from each of the classes. They have been found to yield good performance on a wide range of tasks and are suitable for use with data in high dimensional spaces. The theory behind SVMs has been extensively described in many papers, for example see (Vapnik, 1998; Steinwart and Christmann, 2008), and is not discussed here. This section concentrates on how SVMs can be applied to tasks where there is sequence data, for example speech recognition.

One of the issues with applying SVMs to sequence data, such as speech, is that the SVM is inherently static in nature; "observations" (or sequences) are all required to be of the same dimension. A range of *dynamic kernels* have been proposed that handle this problem, for an overview see (Layton, 2006). Of particular interest in this work are those kernels that are based on generative models, such as Fisher kernels (Jaakkola and Hausser, 1999) and generative kernels (Smith and Gales, 2001). In these approaches a generative model is used to determine the feature-space for the kernel. For example a first-order feature-space for a generative kernel with observation sequence $\mathbf{Y}$ may be written as

$$\boldsymbol{\phi}(\mathbf{Y};\boldsymbol{\lambda}) = \frac{1}{T}\begin{bmatrix} \log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_1)})\right) - \log\left(p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_2)})\right) \\ \boldsymbol{\nabla}_{\lambda^{(\omega_1)}} \log p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_1)}) \\ \boldsymbol{\nabla}_{\lambda^{(\omega_2)}} \log p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_2)}) \end{bmatrix} \tag{16}$$

where $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^{(\omega_1)}, \boldsymbol{\lambda}^{(\omega_2)}\}$ are the set of acoustic model parameters, and $p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_1)})$ and $p(\mathbf{Y};\boldsymbol{\lambda}^{(\omega_2)})$ are the likelihood of the data sequence $\mathbf{Y}$ using generative models associated with classes ω_1 and ω_2 respectively. The normalisation term $1/T$ is used to help normalise the feature-space for sequences of different lengths.

A range of generative models can be used to determine the kernel. HMMs are the form selected in this paper as they are the standard model used in speech recognition. Considering the derivative with respect to the means and variances, elements of the feature-space will have the form (Smith and Gales, 2001)

$$\frac{\partial}{\partial \boldsymbol{\mu}^{(m)}} \log p(\mathbf{Y};\boldsymbol{\lambda}) = \sum_{t=1}^{T} \gamma_m(t) \boldsymbol{\Sigma}^{(m)\text{-}1} \left(\boldsymbol{y}_t - \boldsymbol{\mu}^{(m)}\right) \quad (17)$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}^{(m)}} \log p(\mathbf{Y};\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t=1}^{T} \gamma_m(t) \left(-\boldsymbol{\Sigma}^{(m)\text{-}1} + \boldsymbol{\Sigma}^{(m)\text{-}1}(\boldsymbol{y}_t - \boldsymbol{\mu}^{(m)})(\boldsymbol{y}_t - \boldsymbol{\mu}^{(m)})^{\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\right) \quad (18)$$

where $\gamma_m(t)$ is the posterior probability that component m generated the observation at time t given the complete observation sequence $\mathbf{Y} = \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$. Only the derivatives with respect to the means are used in this work, though it is possible to use other, and higher-order, derivatives.

As SVM training is a distance based learning scheme it is necessary to define an appropriate metric for the distance between two points. The simplest approach is to use a *Euclidean* metric. However, in the same fashion as using the *Mahalanobis*, rather than Euclidean, distances for nearest-neighbour training, an appropriately weighted distance measure may be better. One such metric which is maximally non-committal is given by (Smith and Gales, 2001)

$$K(\mathbf{Y}_i, \mathbf{Y}_j; \boldsymbol{\lambda}) = \boldsymbol{\phi}(\mathbf{Y}_i; \boldsymbol{\lambda})^{\mathsf{T}} \mathbf{G}^{\text{-}1} \boldsymbol{\phi}(\mathbf{Y}_j; \boldsymbol{\lambda}) \quad (19)$$

where $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are two observation sequences and $\mathbf{G}$ is related to the Fisher Information matrix (the log-likelihood ratio is also normalised in this work). In common with other work in this area (Smith and Gales, 2001; Layton and Gales, 2006), $\mathbf{G}$ is approximated by the diagonalised empirical covariance matrix of the training data. Thus

$$\mathbf{G} = \operatorname{diag}\left(\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\phi}(\mathbf{Y}_i;\boldsymbol{\lambda}) - \boldsymbol{\mu}_{\phi}\right)\left(\boldsymbol{\phi}(\mathbf{Y}_i;\boldsymbol{\lambda}) - \boldsymbol{\mu}_{\phi}\right)^{\mathsf{T}}\right) \quad (20)$$

$$\boldsymbol{\mu}_{\phi} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\phi}(\mathbf{Y}_i;\boldsymbol{\lambda}) \quad (21)$$

where there are n training data sequences $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$.

Classification of observation sequence $\mathbf{Y}$ with this form of generative kernel is based on the SVM score $\mathcal{S}_{\texttt{svm}}(\mathbf{Y};\boldsymbol{\lambda})$

$$\mathcal{S}_{\texttt{svm}}(\mathbf{Y};\boldsymbol{\lambda}) = \sum_{i=1}^{n} \alpha_i^{\texttt{svm}} z_i K(\mathbf{Y}_i, \mathbf{Y}; \boldsymbol{\lambda}) + b \quad (22)$$

and the classification rule

$$\hat{\omega} = \begin{cases} \omega_1, \, \mathcal{S}_{\texttt{svm}}(\mathbf{Y}; \boldsymbol{\lambda}) \geq 0 \\ \omega_2, \, \mathcal{S}_{\texttt{svm}}(\mathbf{Y}; \boldsymbol{\lambda}) < 0 \end{cases} \tag{23}$$

where $\alpha_i^{\texttt{svm}}$ is the Lagrange multiplier for observation sequence $\mathbf{Y}_i$ obtained from the SVM maximum margin training, b is the bias and $z_i \in \{1, -1\}$ indicates whether the sequence was a positive (ω_1) or negative (ω_2) example.

4 SVMs for Noise Robustness

The previous two sections have described model-based compensation and support vector machines with generative kernels. This section describes how these schemes can be combined together to allow noise-specific generative kernels to be used with a noise-independent SVM for speech recognition.

One of the problems with using SVMs for speech recognition is that standard SVMs are binary classifiers whereas speech is a multi-class task; for large vocabulary systems there are a vast number of classes. One approach to handling this problem is acoustic code-breaking (Venkataramani et al., 2003). During recognition an initial decoding of the test data is run and a set of pairs of highly confusable words in the test utterance obtained. These confusable pairs can then be rescored using the appropriate binary classifier. To train these classifiers, confusable pairs in the training data are obtained by finding the most confusable word to each of the reference words. This provides a set of training examples for each binary classifier.

As the vocabulary for the task considered in this work, continuous digit recognition, is small, a modified version of acoustic code-breaking is used. It is possible to consider all possible pairs of words to train a set of SVMs that cover all possibilities. As in acoustic code-breaking an initial decoding is run to obtain word boundaries. During rescoring all possible word pairs, rather than just a restricted pair, are scored between the word boundaries. Note acoustic code-breaking as described in (Venkataramani et al., 2003) can be used to allow this approach to be applied to larger vocabulary tasks.

The SVM training and sequence rescoring are described in detail in the next two sections.

The first stage in training the SVM is to segment the data into homogeneous blocks which have the same background noise and channel distortion. The clean acoustic model is then adapted to each of the training data noise conditions using VTS. This adapted corrupted speech model can be used to segment the training data into words and also derive the feature-space to train the SVM. During SVM training rather than just selecting the data from the specific confusable pairs all the data from each of the words is used during training. This yields a far larger number of training examples for the SVM.

In this work only the log-likelihood ratio and derivatives with respect to the means are used. However in contrast to the standard use of the generative kernel, there is an issue with using equation 17 when different noise-specific generative kernels are used for different blocks of data, each characterised by the same background noise conditions. Examining equation 17 shows that the derivative, as expected, is not *dimensionless*[4] . Thus if the dynamic range of the data is dramatically altered by the addition of noise, then the dynamic range of features will vary from noise condition to noise condition. To keep the dynamic ranges of each block of features consistent, it is necessary to ensure that all the features are "dimensionless". For the single dimension-case, this can be simply achieved by using the standard-deviation rather than the variance when computing the derivative in equation 17. When a multi-dimensional feature vector is used, then the "square-root", the Choleski-factorisation, of the inverse covariance matrix must be used. This will be written as $\boldsymbol{\Sigma}^{(m)-1/2}$ for simplicity of notation. For this work only diagonal covariance matrices are used so it is only necessary to take the square-root of the elements on the leading diagonal of the inverse covariance matrix.

Note in the general case, when the same covariance matrices are used for all data sequences, dynamic range differences can arise only between different elements of the vector in equation 16. But this is not generally a problem, since these differences are handled by the metric $\mathbf{G}$. Thus the feature-space used for the SVMs between classes ω_l and ω_j is of the form

[4] This is using the term dimensionless consistent with the use in dimensional analysis, rather than the size of the feature vector.

$$\boldsymbol{\phi}(\mathbf{Y};\boldsymbol{\lambda}) = \frac{1}{T}\begin{bmatrix} \log\left(p(\mathbf{Y};\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)})\right) - \log\left(p(\mathbf{Y};\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)})\right) \\ \sum_{t=1}^{T}\gamma_m(t)\boldsymbol{\Sigma}_{\mathsf{y}}^{(\omega_l 1)\text{-}1/2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_{\mathsf{y}}^{(\omega_l 1)}\right) \\ \vdots \\ \sum_{t=1}^{T}\gamma_m(t)\boldsymbol{\Sigma}_{\mathsf{y}}^{(\omega_l M)\text{-}1/2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_{\mathsf{y}}^{(\omega_l M)}\right) \\ \sum_{t=1}^{T}\gamma_m(t)\boldsymbol{\Sigma}_{\mathsf{y}}^{(\omega_j 1)\text{-}1/2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_{\mathsf{y}}^{(\omega_j 1)}\right) \\ \vdots \\ \sum_{t=1}^{T}\gamma_m(t)\boldsymbol{\Sigma}_{\mathsf{y}}^{(\omega_j M)\text{-}1/2}\left(\boldsymbol{y}_t - \boldsymbol{\mu}_{\mathsf{y}}^{(\omega_j M)}\right) \end{bmatrix} \tag{24}$$

where the clean models have been compensated for the test condition $\mathbf{Y}$, $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)}, \boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)}\}$, and there are a total of M components in each of the two word models $\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)}$ and $\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)}$

The complete procedure for training the noise-independent SVMs is:

(1) For each training noise condition perform model-based compensation to map the clean model parameters to noise corrupted model parameters: $\boldsymbol{\lambda}_{\mathsf{x}} \rightarrow \boldsymbol{\lambda}_{\mathsf{y}}$
(2) Align each training utterance $\mathbf{Y}$ using the reference, $\mathbf{r} = r_1, \dots, r_K$, and $\boldsymbol{\lambda}_{\mathsf{y}}$ to give the word-segmented data sequence $\tilde{\mathbf{Y}}_1, \dots, \tilde{\mathbf{Y}}_K$
(3) For all word pairs (ω_l, ω_j) and for each segment $\tilde{\mathbf{Y}}_i$ set $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)}, \boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)}\}$
   (a) obtain $\boldsymbol{\phi}(\tilde{\mathbf{Y}}_i;\boldsymbol{\lambda})$ for all training examples of ω_l using the appropriate noise compensated acoustic models $\boldsymbol{\lambda}$
   (b) obtain $\boldsymbol{\phi}(\tilde{\mathbf{Y}}_i;\boldsymbol{\lambda})$ for all training examples of ω_j using the appropriate noise compensated acoustic models $\boldsymbol{\lambda}$
   (c) train a noise-independent SVM for pair (ω_l, ω_j) using all positive (a) and negative (b) examples.

This process yields a set of one-versus-one SVMs, one for each word pair (ω_l, ω_j). Both linear and non-linear SVMs may be trained. In this work only linear SVMs were used as the generative kernels typically yield a high dimensional feature without using, for example, polynomial kernels.

4.2 SVM Rescoring

The same set of one-versus-one SVMs trained as above are used for all test noise conditions. For each test noise conditions VTS is used to adapt the clean models. These noise-compensated models are then used to recognise and segment the data. A standard majority voting approach is used to label each segment.

Thus during SVM rescoring the following complete procedure is used:

(1) Compensate the acoustic models for the test noise condition: $\boldsymbol{\lambda}_{\mathsf{x}} \rightarrow \boldsymbol{\lambda}_{\mathsf{y}}$

(2) Recognise the test utterance $\mathbf{Y}$ using $\boldsymbol{\lambda}_{\mathsf{y}}$ to obtain 1-best hypothesis, $\mathbf{r} = r_1, \ldots, r_K$ and align to give the word-segmented data sequence $\tilde{\mathbf{Y}}_1, \ldots, \tilde{\mathbf{Y}}_K$

(3) For each segment, $\tilde{\mathbf{Y}}_i$:

   (a) for each word pair (ω_l, ω_j) set $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)}, \boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)}\}$

$$\hat{\omega} = \begin{cases} \omega_l; & \text{if } \mathcal{S}_{\texttt{svm}}^{(lj)}(\tilde{\mathbf{Y}}_i; \boldsymbol{\lambda}) + \epsilon \frac{1}{\sqrt{g_{11}}} \left(\log \left(p(\tilde{\mathbf{Y}}_i; \boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_l)}) \right) \right. \\ & \qquad \left. - \log \left(p(\tilde{\mathbf{Y}}_i; \boldsymbol{\lambda}_{\mathsf{y}}^{(\omega_j)}) \right) \right) \geq 0 \\ \omega_j; & \text{otherwise} \end{cases} \tag{25}$$

$$\text{set } : \text{count}[\hat{\omega}] = \text{count}[\hat{\omega}] + 1 \tag{26}$$

   (b) classification, $\hat{r}_i$, is given by:

      (i) if no ties in voting: $\hat{r}_i = \text{argmax}_{\omega} \{\text{count}[\omega]\}$

      (ii) if only two words (w_l, w_j) tie then $\hat{r}_i$ determined using equation 25

      (iii) if more than two words tie $\hat{r}_i = r_i$

Note that $\mathcal{S}_{\texttt{svm}}^{(lj)}(\tilde{\mathbf{Y}}_i; \boldsymbol{\lambda})$ in step (3) indicates that the SVM associated with word pair (ω_l, ω_j) is used for scoring $\tilde{\mathbf{Y}}_i$.

In equation 25 a tunable parameter ϵ is used. This is an empirically set value which is used to scale the contribution of the log-likelihood ratio to the SVM score. The log-likelihood ratio is known to be the most discriminatory of the dimensions of the score-space. It is the single feature used for standard HMM-based speech recognition. However using a maximally non-committal metric, $\mathbf{G}$, all dimensions are treated equally. Thus ϵ is used to reflect the additional usefulness of the log-likelihood ratio. As $\epsilon \rightarrow \infty$ the performance of the system will tend to the HMM performance. In equation 25 ϵ is applied to a normalised log-likelihood ratio where the square-root of the first term of the metric $\mathbf{G}$, g_{11}, is used as the normalisation term. Though there was little difference in performance between the normalised and un-normalised versions, it was felt to generalise better between tasks and different noise conditions. For the initial experiments in this work $\epsilon = 2$ was used, this is the value used for the SPR experiments described in (Gales and Longworth, 2008).

The routine above is known to be suboptimal in a number of ways. A simple scheme is used to combine classifier outputs. More complex versions of binary classifier combination, or multi-class SVM may be used, for example (Gales et al., 2009; Wu et al., 2004). The alignment associated with each word-segment is not updated if the hypothesis sequence changes. It is possible to repeat the alignment if the hypothesis changes. However in initial experiments this gave no change in performance. The computational load associated with this scheme increases approximately linearly with the number of word pairs. However as the vocabulary size V increases the computational cost will increase at $\mathcal{O}(V^2)$. Thus the scheme is currently only suited for small vocabulary tasks, such as digit string recognition. It is possible to use acoustic code-breaking to apply these approaches to larger tasks, but this is not investigated in this work. Furthermore it is possible to use the features associated with the generative kernels with other forms of classifier, for example multi-layer perceptrons. Again this is a possible future direction.

5 Results

The performance of the proposed scheme was evaluated on the AURORA 2 task (Hirsch and Pearce, 2000). AURORA 2 is a small vocabulary digit string recognition task. As the vocabulary size (excluding silence) is only eleven (one to nine, plus zero and oh) the number of word pairs is small (66 including silence) making it suitable for the proposed scheme. The utterances in this task are one to seven digits long based on the TIDIGITS database with noise artificially added. The clean training data was used to train the acoustic models. This comprises 8440 utterances from 55 male and 55 female speakers. Two forms of front-end were examined, one HTK-based, the other ETSI-based (Hirsch and Pearce, 2000). For both front-ends a 39 dimensional feature vector consisting of 12 MFCCs appended with the zeroth cepstrum, delta and delta-delta coefficients was used. These differ slightly from the standard parameterisations and perform slightly worse. However this form of front-end allows VTS compensation to be applied to compensate the acoustic models. The acoustic models are 16 emitting state whole word digit models, with 3 mixtures per state and silence and inter-word pause models. All three test sets, A,B and C, were used for evaluating the schemes. For sets A and B, there were a total of 8 noise conditions (4 in each) at 5 different SNRs, 0dB to 20dB. For test set C there were two additional noise conditions at the same range of SNRs. In addition to background additive noise convolutional distortion was added to test set C. Test set A was used as the development set for tuning parameters.

For all the VTS schemes the same procedure as in (Li et al., 2007) was used. An initial estimate of the background additive noise for each utterance was obtained using the first and last 20 frames of the utterance. This was then used as the noise model for VTS compensation and each utterance recognised. This hypothesis was used to estimate a per-utterance noise model in an ML-fashion. The final recognition output used this ML-estimated noise model for VTS compensation.

Where SVM rescoring was used, the SVMs were trained on a subset of the training data available for test set A. This is multi-style data characterized by different noise/SNR conditions with 422 sentences available for each condition (a subset of all the training data).

For the SVMs training only three of the four available noise conditions (N2-N4) and three of the five SNRs 10dB, 15dB and 20dB were used. This allows the generalisation of the SVM to unseen noise conditions to be evaluated on test set A as no data from noise condition N1 and SNRs 5dB and 0dB were used. Note this makes the SVM experiments hard to compare with other approaches where none of the multi-style training data was used. However the baseline VTS experiments are comparable. For all experiments the SVMs were built using the top 1500 dimensions of $\boldsymbol{\phi}(\tilde{\mathbf{Y}}_i; \boldsymbol{\lambda})$ ranked using the Fisher ratio.

5.1 HTK-based Front-End

SNR	Noise				Avg
(dB)	N1	N2	N3	N4	
20	1.78	1.87	1.55	1.54	1.69
15	2.67	2.63	2.00	2.13	2.36
10	5.13	4.20	3.37	4.91	4.39
05	12.25	12.03	8.20	12.40	11.20
00	32.18	37.70	21.92	26.47	29.55
Avg	10.80	11.69	7.41	9.49	9.84

Table 1
VTS WER (%) on AURORA 2 test set A using HTK features.

Initial investigations used the HTK-features (Hirsch and Pearce, 2000). As discussed in section 2.2 these initial results used a mismatch function with magnitude combination, $\gamma = 1$. Table 1 shows the performance of the VTS compensated clean system on each of the test set A noise conditions. As expected, as the SNR decreases the word error rate (WER) increases. Note the word error rate for the clean, uncompensated, model set on the 5dB SNR system was 66.75%. There is thus an 83% relative reduction in error rate by using VTS model-based compensation at 5dB.

Table 2 shows the performance of the SVM rescoring for test set A. For these initial experiments the value of ϵ was set to 2, which was the value used in (Gales and Longworth, 2008). The noise conditions where the multi-style training data was used to train the SVMs are marked with a † as there may be a slight bias for these numbers. For all noise conditions a reduction in the

SNR (dB)	Noise				Avg
	N1	N2	N3	N4	
20	1.38	†1.45	†1.25	†1.30	1.35
15	2.12	†2.00	†1.64	†1.54	1.82
10	3.62	†2.93	†2.54	†3.86	3.23
05	8.78	8.43	6.20	9.53	8.22
00	24.04	27.93	18.52	21.54	23.00
Avg	7.99	8.55	6.03	7.55	7.52

Table 2
SVM rescoring WER (%) on AURORA 2 test set A ($\epsilon = 2$), SVMs trained on N2-N4 10-20dB SNR indicated with †.

WER was observed. The performance gains for the unseen N1 noise condition were consistent with those of the seen N2-N4 conditions. In addition gains at the lower, unseen, SNRs 0dB and 5dB were seen. For example the gain at 5dB SNR was about 27% relative reduction. Overall a 23% relative reduction in error over test set A was obtained compared to the baseline VTS scheme.

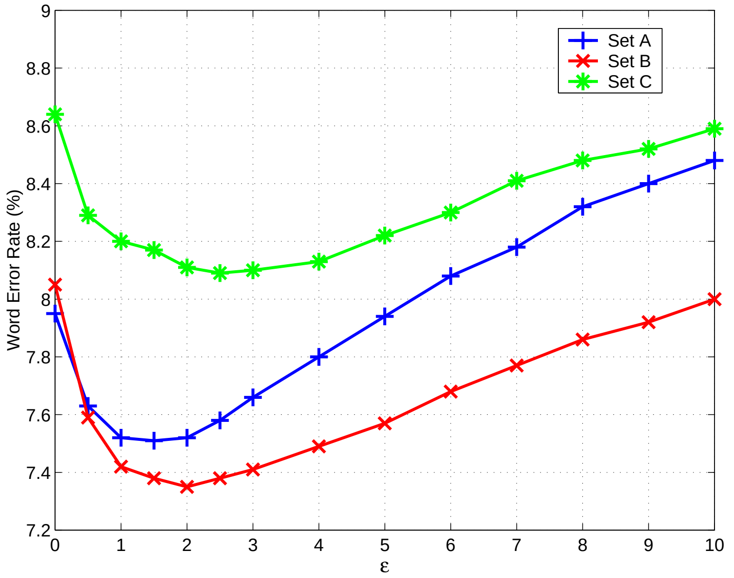

Fig. 2. SVM rescoring performance WER (%) against ϵ using HTK features, SVMs trained on test set A N2-N4 10-20dB SNR.

The performance of the system is expected to be sensitive to the value of ϵ selected. Figure 2 shows how the average performance for each of the three test sets varies as ϵ changes. The performance of the SVM by itself ($\epsilon = 0$) is better than the baseline VTS performance (see table 3). For example the average WER on test set A for the "pure" SVM performance ($\epsilon = 0$) was 7.95% compared to 9.84% for the VTS compensation. Additional gains were obtained by increasing ϵ. The minima for the three test sets occur in the range 1.5 to 3.0. Thus the previously used value of $\epsilon = 2$ is a reasonable value for all three test sets, so was left unaltered for the later experiments.

SNR	Set A		Set B		Set C	
(dB)	VTS	SVM	VTS	SVM	VTS	SVM
20	1.69	1.35	1.46	1.22	1.57	1.33
15	2.36	1.82	2.37	1.77	2.47	2.00
10	4.39	3.23	4.12	3.16	4.49	3.52
05	11.20	8.22	10.05	7.68	10.69	8.70
00	29.55	23.00	27.54	22.93	28.41	25.01
Avg	9.84	7.52	9.11	7.35	9.53	8.11

Table 3
VTS ($\gamma = 1$) and SVM rescoring performance WER (%) tests sets A, B, C ($\epsilon = 2$) using HTK features, SVMs trained on test set A N2-N4 10-20dB SNR.

Table 3 summarises the results for VTS compensation and SVM rescoring for all three available test sets. Note that none of the noise conditions for test sets B and C were used to train the SVMs. For all noise conditions large reductions in WER were obtained using SVM rescoring compared to the baseline VTS compensation. The relative reductions in average WER were 23%, 19% and 14% for test sets A, B and C respectively. Though the relative gains for test sets B and C were slightly less than that for test set A, it still indicates that a good level of noise-independent classification can be obtained using these noise-specific generative kernels.

The results presented so far have applied both VTS compensation for kernel adaptation and SVM rescoring. In order to identify the gains from each of these stages an additional set of experiments were run. First a system was built using the same SVM training data, but where no adaptation of the kernel parameters in training or test was performed. To make the contrasts as fair as possible, initial decoding was run with VTS-compensated HMMs and ϵ used to scale the VTS-compensated log-likelihood ratios. Thus as $\epsilon \rightarrow \infty$ the performance will tend to the HMM VTS compensated performance, rather than the uncompensated performance. The results for this system on test set A are shown in figure 3, labelled *uncomp: N2-N4, 10-20dB.* The performance when $\epsilon = 0$ was very poor, just under 14%. As the value was increased the performance improved. However the performance was always worse than the system using VTS kernel adaptation, labelled *VTS: N2-N4, 10-20dB.* Some performance gain over the baseline VTS scheme, 9.84%, can be seen indicating that the discriminative nature of this “multi-style” SVM is still useful.

The second experiment involved restricting the range of noise SNRs seen by

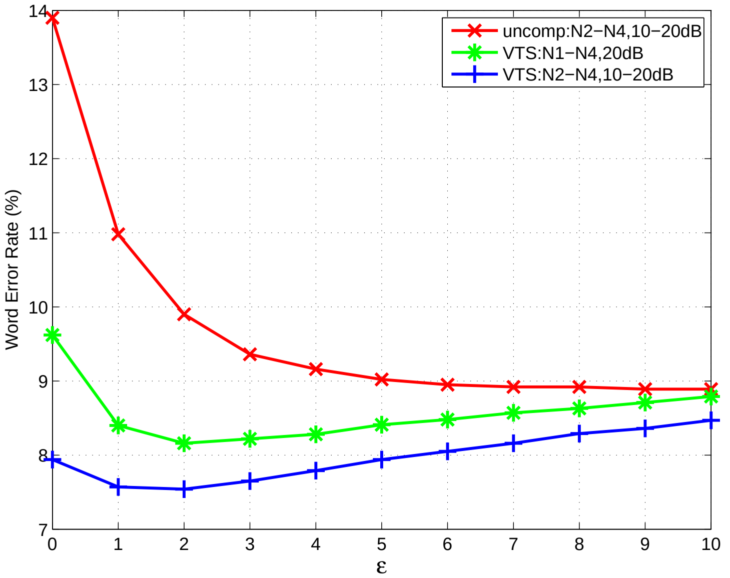

Fig. 3. SVM rescoring performance WER (%) against ϵ using HTK features, SVMs trained on test set A. Training using no compensation on N2-N4 10-20dB SNR noise conditions; VTS compensation on N1-N4 20dB SNR noise conditions, and VTS compensation on N2-N4 10-20dB SNR noise conditions.

the SVMs to assess to what extent the VTS kernel adaptation compensates for the varying noise conditions. The SVMs were trained on the 20dB SNR data [5] . Figure 3 shows the performance of this system, labelled *VTS: N1-N4, 20dB*, as the value of ϵ varies. This system was trained on all the noise conditions as the amount of data to train the SVM was less than when using a range of SNRs. The results show a similar trend to the system trained on a range of SNRs, but the performance is consistently worse. This indicates that the VTS kernel adaptation allows a degree of insensitivity to the SNR conditions, but data from a range of SNRs is better.

5.2 ETSI-based Front-End

The previous section has used the HTK-based features with the baseline $\gamma = 1$ noise compensation. The ETSI-based features are known to yield better per-

[5] An alternative experiment would be to train the SVMs on the clean data used to train the HMMs. However, on this, and other tasks, it has been observed that training the generative models and the SVM parameters on the same data produces “biased” SVM parameters. The reason is that, given that the generative models’ parameters “match” the training data, the scores obtained from them tends to be too closely linked to the training data and are not representative of unseen data. It is interesting that the addition of noise to the data (even when VTS compensation is performed) does not suffer from the same problem.

formance than the HTK-based features (Hirsch and Pearce, 2000). The interaction of this improved front-end with VTS and the SVM rescoring gains was therefore investigated. It is also interesting to examine whether performance gains are possible when the VTS mismatch function is optimised. For this work the γ-optimisation in equation 12 was used.

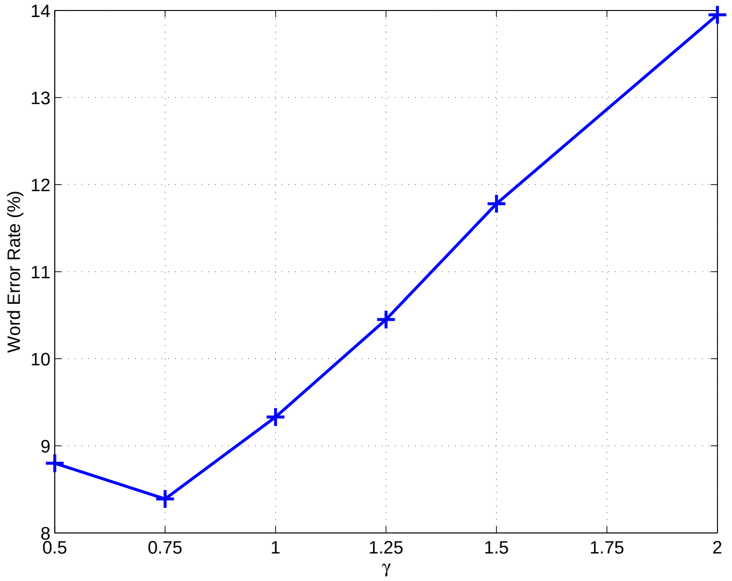

Fig. 4. Plot of VTS γ-optimised WER (%) performance on test set A against γ

Figure 4 shows the variations in WER for test set A as the value of γ is varied in equation 12. From the graph it is clear that the magnitude-based mismatch function ($\gamma = 1$) outperforms the more theoretically correct power-based mismatch function ($\gamma = 2$). The best performance on test set A is with $\gamma = 0.75$. As discussed in section 2.2 the impact of this γ-optimised system is very similar to the α-optimised phase-sensitive system in (Li et al., 2008).

SNR	Set A			Set B			Set C		
(dB)	VTS	γ-opt	SVM	VTS	γ-opt	SVM	VTS	γ-opt	SVM
20	1.60	1.47	1.27	1.38	1.30	1.09	1.63	1.49	1.25
15	2.26	2.12	1.75	2.23	2.11	1.69	2.38	2.18	1.87
10	4.27	3.91	3.10	3.91	3.51	2.78	4.40	4.07	3.40
05	10.52	8.94	7.10	9.13	8.53	6.81	9.64	8.56	7.66
00	28.00	25.48	21.73	25.95	24.46	21.64	26.05	24.51	22.56
Avg	9.33	8.39	6.99	8.52	7.98	6.80	8.82	8.16	7.35

Table 4
VTS ($\gamma = 1$), VTS γ-optimised ($\gamma = 0.75$) and SVM rescoring($\epsilon = 2$) performance WER (%) tests Sets A, B, C using ETSI features, SVMs trained on set A N2-N4 10-20dB SNR.

Table 4 shows the performance for the baseline ($\gamma = 1$) and the γ-optimised VTS performance ($\gamma = 0.75$). The performance of the baseline system is consistently better than the HTK-based features when VTS is used [6] . By optimising γ further large gains in performance can be obtained over all test sets. Note that though γ was optimised on test set A, there are consistent gains on test sets B and C. For example on test set B a 7% relative reduction in average WER was obtained, with gains for all SNRs. This is consistent with the gains observed for phase-sensitive α optimised compensation (Li et al., 2008).

The γ-optimised VTS models were then used in the SVM rescoring process. For these SVM rescoring experiments $\epsilon = 2$ was again used. These results are also shown in table 4. For all SNRs and test sets gains over the optimised VTS scheme were obtained. Relative gains of 17% for test set A, 15% for test set B and 10% for test set C were obtained. These gains are less than those obtained with the non-optimised HTK scheme, but still show that large gains using SVM rescoring are possible when the VTS compensation has been optimised.

6 Conclusions

This paper has described a new approach to noise-robust speech recognition. The scheme combines model-based noise compensation schemes with a discriminative classifier, in this case an SVM. Rather than adapting the discriminative classifier, changing noise conditions are handled by adapting the generative kernel. This is possible as generative models, such as HMMs, are used to determine the kernel feature-space. Thus model-based compensation can be used to adapt the generative models and obtain kernel features that are specific to the current noise environment. For this work VTS was used as the model-compensation scheme. In addition to the baseline VTS scheme, an approach using an optimised mismatch function was investigated. To handle the multi-class issue (the SVM is inherently binary) a combination of a modified version of acoustic code-breaking with majority voting was used. Thus the overall scheme allows a noise-independent SVM with noise-dependent generative kernels to be used to rescore the recognition output from a standard HMM-based speech recognition system.

Initial experiments on the AURORA 2 task are presented using VTS as the model-based compensation scheme. To train the SVMs a subset of the multi-style training data for test set A was used. To ensure that the SVMs could

[6] The performance of this baseline $\gamma = 1$ system is similar to the results in (Li et al., 2007). However it is unclear what domain was used for the compensation in (Li et al., 2007) as the default ETSI features are magnitude based.

handle unseen noise conditions and SNRs, no data from the N1 noise condition and the 0dB and 5dB SNRs were used to train the SVMs. In addition to test set A the performance was evaluated on test sets B and C with unseen noise conditions. Compared to the VTS trained system large reductions in WER were observed with SVM rescoring for all noise conditions, including the ones on which the SVMs were not trained. Consistent gains over the VTS baseline were obtained even when the VTS compensation scheme was optimised for the space in which the speech and noise are assumed to be additive, γ-optimisation.

The results presented in this paper are preliminary for a number of reasons. Though consistent gains on more complex tasks have been obtained using magnitude compensation ($\gamma = 1$) it is not clear whether further γ-optimisation will generalise from task to task. Discriminatively trained, or more complex, acoustic models can be used for this task. The proposed scheme can be applied using these improved generative models. The AURORA 2 task comprises artificially corrupted data. The standard issue of whether the large gains observed will map to "real" data remains to be evaluated. Finally SVMs were used as the discriminative classifier. For larger vocabulary tasks other discriminative classifiers, such as conditional augmented models (Layton and Gales, 2006), may be more appropriate. Despite these limitations, the results indicate that good levels of performance can be obtained using noise-independent discriminative classifiers with noise-specific kernels.

Acknowledgements

This work was partly funded by Toshiba Research Europe Ltd, Cambridge Research Laboratory. The authors would also like to thank Hank Liao for the initial implementation of VTS and Chris Longworth for software for generative kernel manipulation.

References

Acero, A., 1993. Acoustical and Environmental Robustness in Automatic Speech Recognition. Kluwer Academic Publishers.

Acero, A., Deng, L., Kristjansson, T., Zhang, J., 2000. HMM Adaptation using Vector Taylor Series for Noisy Speech Recognition. In: Proc. ICSLP. Beijing, China, pp. 869–872.

Deng, L., Droppo, J., Acero, A., 2004. Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise. IEEE Transactions on Speech and Audio Processing 12, 133–143.

Gales, M., 1995. Model-based techniques for noise robust speech recognition. Ph.D. thesis, Cambridge University.

Gales, M., Feb. 2007. Discriminative models for speech recognition. In: ITA Workshop. University San Diego, USA, pp. 170–176.

Gales, M., Longworth, C., 2008. Discriminative classifiers with generative kernels for noise robust ASR. In: Proc. Interspeech. Brisbane, Australia, pp. 1996–1999.

Gales, M., Ragni, A., AlDamarki, H., Gautier, C., Dec. 2009. Support vector machines for noise robust ASR. In: Proc. ASRU. Merano, Italy.

Gopinath, R., Gales, M., Gopalakrishnan, P., Balakrishnan-Aiyer, S., Picheny, M., 1995. Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task. In: Proc. ARPA Workshop on Spoken Language System Technology. Austin, Texas, pp. 127–130.

Gunawardana, A., Mahajan, M., Acero, A., Platt, J. C., 2005. Hidden conditional random fields for phone classification. In: Proc. Interspeech. pp. 1117–1120.

Hirsch, H.-G., Pearce, D., Sep. 2000. The AURORA experimental framework for the evaluation of speech recognition systems under noisy conditions. In: Proc. ASR. pp. 181–188.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., Schölkopf, B., Sep. 2007. Correcting sample selection bias by unlabeled data. In: Schölkopf B., J. Platt, T. H. (Ed.), Twentieth Annual Conference on Neural Information Processing Systems (NIPS 2006). MIT Press, Cambridge, MA, USA, pp. 601–608.

Huang, X. D., Acero, A., Hon, H. W., 2001. Spoken Language Processing. Prentice Hall.

Jaakkola, T., Hausser, D., 1999. Exploiting generative models in disciminative classifiers. In: Solla, S., Cohn, D. (Eds.), Advances in Neural Information Processing Systems 11. MIT Press, pp. 487–493.

Kuo, H.-K., Gao, Y., 2006. Maximum entropy direct models for speech recognition. IEEE Transactions Audio Speech and Language Processing.

Layton, M., Sep. 2006. Augmented statistical models for classifying sequence data. Ph.D. thesis, Cambridge University.

Layton, M., Gales, M., May 2006. Augmented statistical models for speech recognition. In: Proc. ICASSP. Vol. 1. Toulouse, pp. 129–132.

Li, J., Deng, L., Yu, D., Gong, Y., Acero, A., Dec. 2007. High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector taylor series. In: Proc. ASRU. Kyoto, Japan, pp. 65–70.

Li, J., Deng, L., Yu, D., Gong, Y., Acero, A., Apr. 2008. HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition. In: Proc. ICASSP. pp. 4069–4072.

Li, X., Bilmes, J., 2006. Regularized adaptation of discriminative classifiers.

In: Proc. ICASSP. Toulouse, France, pp. 237–240.

Liao, H., sep 2007. Uncertainty decoding for noise robust speech recognition. Ph.D. thesis, Cambridge University, Cambridge, UK.

Liao, H., Gales, M., 2006. Joint uncertainty decoding for robust large vocabulary speech recognition. Tech. Rep. CUED/F-INFENG/TR552, University of Cambridge, available from: `mi.eng.cam.ac.uk/~mjfg`.

Liao, H., Gales, M., Apr. 2007. Adaptive Training with Joint Uncertainty Decoding for Robust Recognition of Noise Data. In: Proc. ICASSP. Vol. 4. Honolulu, USA, pp. 389–392.

Moreno, P., 1996. Speech recognition in noisy environments. Ph.D. thesis, Carnegie Mellon University.

Smith, N., Gales, M., Dec. 2001. Speech recognition using SVMs. In: Advances in Neural Information Processing Systems. Vol. 14. pp. 1197–1204.

Steinwart, I., Christmann, A., Aug. 2008. Support Vector Machines (Information Science and Statistics). Springer.

van Dalen, R., Gales, M., 2008. Covariance modelling for noise-robust speech recognition. In: Proc. InterSpeech. Brisbane, Australia, pp. 2000–2003.

Vapnik, V., 1998. Statistical learning theory. John Wiley & Sons.

Venkataramani, V., Chakrabartty, S., Byrne, W., 2003. Support vector machines for segmental minimum Bayes risk decoding of continuous speech. In: Proc. ASRU. pp. 13–18.

Wu, T.-F., Lin, C.-J., Weng, R., 2004. Probability estimates for multi-class classification by pairwise coupling. Machine Learning Research 5, 975–1005.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., Dec. 2006. The HTK Book (for HTK Version 3.4). University of Cambridge, `http://htk.eng.cam.ac.uk`.