INVITED PAPER

Training Augmented Models using SVMs

M.J.F. GALES[†] ***and*** **M.I. LAYTON**[†], ***Nonmembers***

SUMMARY There has been significant interest in developing new forms of acoustic model, in particular models which allow additional dependencies to be represented than those contained within a standard hidden Markov model (HMM). This paper discusses one such class of models, augmented statistical models. Here, a local exponential approximation is made about some point on a base model. This allows additional dependencies within the data to be modelled than are represented in the base distribution. Augmented models based on Gaussian mixture models (GMMs) and HMMs are briefly described. These augmented models are then related to generative kernels, one approach used for allowing support vector machines (SVMs) to be applied to variable length data. The training of augmented statistical models within an SVM, generative kernel, framework is then discussed. This may be viewed as using maximum margin training to estimate statistical models. Augmented Gaussian mixture models are then evaluated using rescoring on a large vocabulary speech recognition task.

key words: *speech recognition, hidden Markov models, support vector machines, augmented statistical models*

1. Introduction

There have been a wide-range of acoustic models applied to the speech recognition task. These range from the standard hidden Markov model (HMMs), to segmental models [1], switching linear dynamical systems (LDSs) [2], buried Markov models (BMMs) [3] and mixed memory models [4]. Many of these models can be viewed as state-space models and graphical models [2]. The underlying aspect of all these models is how to appropriately model the dependencies (and complexities) of the speech signal. For example, forms of dependencies include observation independence given the current state, as in an HMM, and independence given a continuous latent state-space variable, as in an LDS. The fundamental questions that must be answered when looking at these models is which latent variables should be included, what dependencies should be modelled, and how the distributions of the observations are altered by the dependencies. To the authors' knowledge, there are as yet no systematic approaches that allow all these questions to be answered.

In this paper a structured approach is described to obtain the statistics that determine the dependencies to be modelled in the observation sequence. The approach adopted is to use a base statistical model, and then for each point on that distribution to construct a local exponential model. The base statistical model determines the latent variables; the sufficient statistics are determined using higher order derivatives of the log-likelihood. This is similar to a constrained exponential model [5]. However here all the parameters of the model, including the parameters of the base distribution, may be trained. This will be referred to as an augmented statistical model [6].

Using this form of model, the number of model parameters rapidly increases. Though with sufficient training data, large numbers of model parameters can be trained, the use of robust training criteria that allow for good generalisation are useful. In this work maximum margin training, as used to train support vector machines (SVMs), is used. The paper is organised as follows. The next section describes the general theory of augmented statistical models and the forms that they take for Gaussian mixture models (GMMs) and HMMs. SVMs and generative kernels are then described. This is followed by a description of how maximum margin training can be used to train augmented models. The issues of applying these models to large vocabulary systems is then described followed by results on an large vocabulary task.

2. Augmented Statistical Models

2.1 The Exponential Family

Many standard forms of statistical model are based on the exponential family. The general form for the exponential family with parameters $\boldsymbol{\alpha}$ can be expressed as,

$$p(\mathbf{o};\boldsymbol{\alpha}) = \frac{1}{\tau} h(\mathbf{o}) \exp\left(\boldsymbol{\alpha}'\mathbf{T}(\mathbf{o})\right) \quad (1)$$

where $h(\mathbf{o})$ is the *reference distribution*, $\boldsymbol{\alpha}$ are the *natural parameters*, τ is the normalisation term (a function of both the reference distribution and the natural parameters), and the function $\mathbf{T}(\mathbf{o})$ is a *sufficient statistic*. There are a number of standard examples, including the exponential distribution and the Gaussian (Normal) distribution. The reason for the interest

Manuscript received January 1, 2005.
Manuscript revised January 1, 2005.
Final manuscript received January 1, 2005.

[†]The authors are with the Department of Engineering, University of Cambridge, Trumpington St., Cambridge, CB2 1PZ, U.K.

in members of the exponential family is that the sufficient statistics are of a fixed dimensionality and that conjugate priors can be defined simply for all members of this family. When dynamic data is being considered, such that each example is a series of observations, $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, $\mathbf{o}_t \in \mathbb{R}^d$, the range of possible statistics becomes very large. Dependencies between observations (as in BMMs [3]), as well as within the feature vector, can now be modelled.

An interesting subset of the set of all possible members of the exponential family is the *constrained exponential family* [5]. Here rather than allowing any form of statistics, a local exponential approximation to the reference distribution is used as the statistical model, where the local approximation replicates some of the properties of the reference distribution. In this paper a slightly more general form of statistical model than the constrained exponential family is used. In addition to the values of the local exponential model, the reference distribution parameters may be learnt from the data. These models are referred to as *augmented statistical models* [6].

2.2 Augmented Statistical Models

Augmented statistical models are an attractive approach to building class-conditional probability density functions, since they yield a mathematically consistent formulation to add higher order dependencies into the model. First a base statistical model, $\check{p}(\mathbf{o};\boldsymbol{\lambda})$, is defined. A member of the exponential family that locally approximates this base model for a particular set of parameters $\boldsymbol{\lambda}$ is then used as the final statistical model. The general form of augmented statistical model for a base statistical model, $\check{p}(\mathbf{o};\boldsymbol{\lambda})$, can be expressed as,

$$\begin{aligned} p(\mathbf{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) &= \frac{1}{\tau}\check{p}(\mathbf{o};\boldsymbol{\lambda}) \exp\left(\boldsymbol{\alpha}' \left\{\boldsymbol{\nabla}_{\boldsymbol{\lambda}}^{(\rho)} \ln \check{p}(\mathbf{o};\boldsymbol{\lambda})\right\}\right) \\ &= \frac{1}{\tau}\overline{p}(\mathbf{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) \end{aligned} \tag{2}$$

where $\boldsymbol{\nabla}_{\boldsymbol{\lambda}}^{(\rho)} \ln(\check{p}(\mathbf{o};\boldsymbol{\lambda}))$ is the vector form of all the derivatives[†] up to order ρ,

$$\boldsymbol{\nabla}_{\boldsymbol{\lambda}}^{(\rho)} \ln \check{p}(\mathbf{o};\boldsymbol{\lambda}) = \begin{bmatrix} \boldsymbol{\nabla}_{\boldsymbol{\lambda}} \ln \check{p}(\mathbf{o};\boldsymbol{\lambda}) \\ \frac{1}{2!}\mathrm{vec}\left(\boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \ln \check{p}(\mathbf{o};\boldsymbol{\lambda})\right) \\ \vdots \\ \frac{1}{\rho!}\mathrm{vec}\left(\boldsymbol{\nabla}_{\boldsymbol{\lambda}}^{\rho} \ln \check{p}(\mathbf{o};\boldsymbol{\lambda})\right) \end{bmatrix} \tag{3}$$

τ is the appropriate normalisation term, thus

$$\tau = \int_{\mathbb{R}^d} \overline{p}(\mathbf{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) d\mathbf{o} \tag{4}$$

[†]For simplicity, in this work the *natural basis* and higher order derivatives are assumed to yield an orthogonal basis. Given this assumption it is not necessary to distinguish between covariant and contravariant components and bases [5].

For this augmented model to be a valid PDF, $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ must be chosen such that the integral is bounded.

If the base statistical model is itself a member of the exponential family, the augmented statistical model will also be a member of the exponential family, though not necessarily of the same form as the base distribution. This is not true for situations where the base statistical model is not a member of the exponential family, for example the Gaussian mixture model discussed in Sect. 2.3.

It is interesting to contrast the nature of the dependencies that are incorporated into the augmented model compared to those of the base model. Independence assumptions in the base statistical model are reflected in the independence assumptions in the augmented model. However this is not the case for the conditional independence assumptions. For example, for augmented GMMs (A-GMMs) and HMMs (A-HMMs), discussed below, the observations are not conditionally independent given the base component that generated them – the posterior, $P(n|\mathbf{o};\boldsymbol{\lambda})$ in Eq.(7), causes the likelihood to be a function of all the components. The augmented statistical models can be related to performing a Taylor series expansion on the statistical model [6], [7].

2.3 Augmented Gaussian Mixture Models

One of the standard forms of model used in statistical pattern processing are mixture models, in particular GMMs. The base statistical model, a GMM, has the form

$$\check{p}(\mathbf{o};\boldsymbol{\lambda}) = \sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \tag{5}$$

Considering just the first order derivatives of the mean as an element of the augmented model

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}_m} \ln \check{p}(\mathbf{o};\boldsymbol{\lambda}) = P(m|\mathbf{o};\boldsymbol{\lambda})\boldsymbol{\Sigma}_m^{-1}(\mathbf{o}-\boldsymbol{\mu}_m) \tag{6}$$

where $P(m|\mathbf{o};\boldsymbol{\lambda}) = c_m\mathcal{N}(\mathbf{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)/\check{p}(\mathbf{o};\boldsymbol{\lambda})$, and c_m, $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ are the prior, mean and covariance matrix for component m. The associated A-GMM is then given by

$$\begin{aligned} p(\mathbf{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{1}{\tau}\sum_{m=1}^{M} c_m \Bigg\{ &\mathcal{N}(\mathbf{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \times \\ &\exp\left(\sum_{n=1}^{M} P(n|\mathbf{o};\boldsymbol{\lambda})\boldsymbol{\alpha}_n'\boldsymbol{\Sigma}_n^{-1}(\mathbf{o}-\boldsymbol{\mu}_n)\right)\Bigg\} \end{aligned} \tag{7}$$

The parameters $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M\}$ are additional model parameters for the A-GMM.

Figure 1 shows two distributions trained using data generated from a symmetric log-normal distribution. ML training was used to obtain a two-component GMM for the training data. In addition an A-GMM, using only first-order derivatives of the mean, was

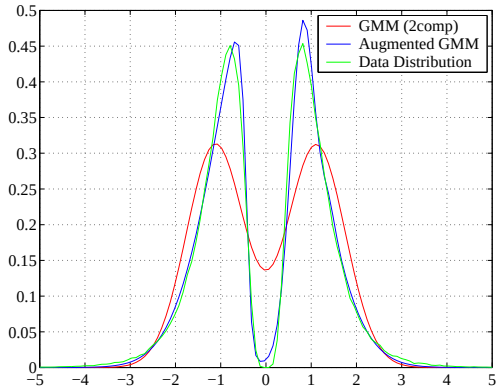


Fig. 1 Modelling of "symmetric" log-normal distribution

trained. From the diagram it is clear that the additional power of the A-GMM is able to model the distribution better than the GMM, though using an additional 2 model parameters. This is reflected in the average log-likelihoods, -1.59 for the GMM and -1.45 for the A-GMM. Interestingly even using a 4-component GMM the log-likelihood was only -1.46, still less than the two component A-GMM.

2.4 Augmented Hidden Markov Models

For speech recognition the most commonly used acoustic model is the HMM. In contrast to GMMs, HMMs do not assume that the observations are independent of one another. For an example, consisting of T observations, $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, the HMM base statistical model can be written as,

$$\check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ \prod_{t=1}^{T} a_{\theta_t\theta_{t-1}} \left(\sum_{m\in\theta_t} c_m \mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \right) \right\} \quad (8)$$

where the state distributions are modelled using GMMs, θ denotes a particular state path through the HMM, and $\boldsymbol{\Theta}$ represents all valid state paths of length T through the HMM. The first order derivatives of the log-likelihood have a similar form to the GMM and can be written as [7],

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}_{jm}} \ln \check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{t=1}^{T} \gamma_{jm}(t) \boldsymbol{\Sigma}_{jm}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jm}) \quad (9)$$

where the posterior, $\gamma_{jm}(t) = P(\theta_t = \{s_j, m\}|\mathbf{O};\boldsymbol{\lambda})$. In addition to the state assignment s_j, the latent variable, θ_t, has been extended to include the mixture-component, m, From a segment model perspective, when each state has a single mixture-component, the derivative is related to the weighted segment mean. The inclusion of the posterior, $P(\theta_t = \{s_j, m\}|\mathbf{O};\boldsymbol{\lambda})$, in this expression ensures that observations generated by an A-HMM are dependent on all other observations in the utterance $\boldsymbol{O}$. Thus, conditional independence of observations given the current state is broken.

It is interesting to examine the higher-order derivatives of HMMs. For example, the second derivative with respect to the means is given by,

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}_{in}} \boldsymbol{\nabla}'_{\boldsymbol{\mu}_{jm}} \ln \check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{t=1}^{T}\sum_{\tau=1}^{T} \Big\{ \left(\gamma_{\{jm,in\}}(t,\tau) - \gamma_{jm}(t)\gamma_{in}(\tau)\right) \times \boldsymbol{\Sigma}_{in}^{-1} (\mathbf{o}_\tau - \boldsymbol{\mu}_{in}) (\mathbf{o}_t - \boldsymbol{\mu}_{jm})' \boldsymbol{\Sigma}_{jm}^{-1} \Big\} \quad (10)$$

where $\gamma_{\{jm,in\}}(t,\tau)$ is the joint state/component posterior,

$$\gamma_{\{jm,in\}}(t,\tau) = P(\theta_\tau = \{s_i, n\}, \theta_t = \{s_j, m\}|\boldsymbol{\lambda}, \mathbf{O}) \quad (11)$$

From Eq. (10), it is clear that second derivatives of the base statistical model are dependent on all observations within an utterance and all possible pairs of (discontiguous) states. Thus second-order A-HMMs overcome the first-order Markov assumption of the base model, allowing additional temporal dependencies to be modelled.

Calculation of the joint state/component posterior requires the use of a double forward-backward style algorithm [8]. Alternatively, continuous rational kernels can be used. These offer an attractive framework where standard and efficient finite-state transducer (FST) algorithms can be used. Within this framework, calculation of higher-order derivatives is reduced to the task of selecting appropriate finite-state transducers (FSTs). Examples of such transducers are given in [9].

Unfortunately, for both first- and second-order augmented models, the final log-likelihood cannot be computed until the end of the sentence, impacting standard pruning techniques.

3. Support Vector Machines

Support Vector Machines (SVMs) [10] are an approximate implementation of structural risk minimisation. They have been found to yield good performance on a wide of range tasks. This section briefly reviews SVMs and the use of generative kernels which are one approach to handling the dynamic nature of speech.

3.1 Maximum Margin Training

SVMs are based upon the intuitive concept of maximising the margin between the decision hyperplane and the closest training examples. This has been shown to be related to minimising an upper bound on the generalisation error [10]. However, in general it is not possible to construct a separating hyperplane between classes

with no classification errors. In these situations, an optimal hyperplane is found to minimise the probability of error, averaged over the training set. This is accomplished by allowing *soft margins*. The margin between classes is said to be soft if there exist training examples, $\mathbf{o}_i$ (with labels $y_i \in \{-1, 1\}$), that violate the constraint $y_i(\langle \mathbf{w}, \mathbf{o}_i \rangle + w_0) \geq 1$, where $\mathbf{w}$ is the weight vector, w_0 is the bias of the optimal hyperplane and $\langle \cdot, \cdot \rangle$ indicates the inner product between the two vectors using an appropriate metric[†]. Slack variables, $\epsilon_i \geq 0$, are introduced to measure the deviation of these examples from the ideal condition of pattern separability. For a set of n training examples, the objective function and constraint become,

$$\{\hat{\mathbf{w}}, \hat{w}_0\} = \arg\min_{\mathbf{w}, w_0} \left\{ \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{n} \epsilon_i \right\} \tag{12}$$

subject to $y_i(\langle \mathbf{w}, \mathbf{o}_i \rangle + w_0) \geq 1 - \epsilon_i$. C acts as a regularisation parameter and controls the trade-off between the margin and the number of misclassified points. For non-linearly separable data, Cover's theorem [11] states that examples may be made linearly separable with a high probability given a non-linear transformation, $\boldsymbol{\phi}(\mathbf{o}; \boldsymbol{\lambda})$, from input-space, $\mathbf{o}$, to a *feature-space* of sufficient dimensionality. Using this mapping, the kernelised form of the dual objective function is defined for the Lagrange multipliers, α^{svm},

$$\hat{\boldsymbol{\alpha}}^{\text{svm}} = \arg\max_{\alpha^{\text{svm}}} \left\{ \sum_{i=1}^{n} \alpha_i^{\text{svm}} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{\text{svm}} \alpha_j^{\text{svm}} y_i y_j K(\boldsymbol{\lambda}, \mathbf{o}_i, \mathbf{o}_j) \right\} \tag{13}$$

subject to $\sum_{i=1}^{n} \alpha_i^{\text{svm}} y_i = 0$ and $0 \leq \alpha_i^{\text{svm}} \leq C$. Here

$$K(\boldsymbol{\lambda}, \mathbf{o}_i, \mathbf{o}_j) = \langle \boldsymbol{\phi}(\mathbf{o}_i; \boldsymbol{\lambda}), \boldsymbol{\phi}(\mathbf{o}_j; \boldsymbol{\lambda}) \rangle \tag{14}$$

The upper limit on the Lagrange multipliers, $\boldsymbol{\alpha}^{\text{svm}}$, limits the influence of individual examples (which may be outliers). At optimality, the Karush-Kuhn-Tucker (KKT) conditions ensure that only examples that lie on the margin ($\langle \mathbf{w}, \mathbf{o}_i \rangle + w_0 = 1 - \epsilon_i$), or that violate the margin constraint, have $\alpha_i^{\text{svm}} > 0$. These examples are known as the *support vectors* [10].

3.2 Generative Kernels

One of the issues with applying SVMs to time varying data, such as speech data, is that the SVM is inherently static in nature. To handle this problem Fisher Kernels [12] and generative kernels [7] have been proposed. Here a generative model that can handle dynamic data is used. An example first-order form of a generative kernel for the set of T observations, $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, may be written as,

$$\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) = \frac{1}{T} \begin{bmatrix} \ln \check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(1)}) - \ln \check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(2)}) \\ \boldsymbol{\nabla}_{\boldsymbol{\lambda}^{(1)}} \ln \check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(1)}) \\ -\boldsymbol{\nabla}_{\boldsymbol{\lambda}^{(2)}} \ln \check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(2)}) \end{bmatrix} \tag{15}$$

where $\check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(1)})$ and $\check{p}(\mathbf{O}; \boldsymbol{\lambda}^{(2)})$ are the generative models associated with classes ω_1 and ω_2 respectively. The term $1/T$ provides length normalisation for the score-space. When the base model parameters are constrained to be equal, the generative score-space reduces to the Fisher score-space.

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 *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,

$$K(\boldsymbol{\lambda}, \mathbf{O}_i, \mathbf{O}_j) = \boldsymbol{\phi}(\mathbf{O}_i; \boldsymbol{\lambda})' \mathbf{G}^{-1} \boldsymbol{\phi}(\mathbf{O}_j; \boldsymbol{\lambda}) \tag{16}$$

where $\mathbf{O}_i$ and $\mathbf{O}_j$ are two sequences and,

$$\mathbf{G} = \mathcal{E}\left\{ (\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi) (\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi)' \right\} \tag{17}$$

where $\boldsymbol{\mu}_\phi = \mathcal{E}\{\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda})\}$. This will be the form of generative kernel used in this work.

In contrast to other forms of kernel there may be many parameters associated with the generative model. It is therefore sensible to investigate maximum margin training of the generative kernels [8]. Here,

$$\{\hat{\boldsymbol{\alpha}}^{\text{svm}}, \hat{\boldsymbol{\lambda}}\} = \arg\max_{\alpha^{\text{svm}}} \min_{\lambda} \left\{ \sum_{i=1}^{n} \alpha_i^{\text{svm}} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^{\text{svm}} \alpha_j^{\text{svm}} y_i y_j K(\boldsymbol{\lambda}, \mathbf{O}_i, \mathbf{O}_j) \right\} \tag{18}$$

Unfortunately there is no simple method for optimising the values. A simple iterative process can be used where the support vectors are estimated and then the generative kernel parameters are updated using gradient descent.

4. Maximum Margin Statistical Models

From the previous section, maximising the margin is a good training criterion when dealing with large dimensional feature-spaces, or little training data. Furthermore as maximum margin training attempts to correctly classify all training examples, it is inherently discriminatory in nature and is thus an obvious alternative criterion to discriminative criteria such as maximum mutual information (MMI) [13]. The disadvantage of

[†]Where a Euclidean metric is used this simply becomes the scalar product of the two vectors.

the approach is that it is inherently a binary classification approach. For this work, the binary case will be considered, although schemes have been proposed to handle the multiclass case [14]. For the binary case, there are two sets of augmented model parameters to train, $\{\boldsymbol{\lambda}^{(1)}, \boldsymbol{\alpha}^{(1)}\}$ and $\{\boldsymbol{\lambda}^{(2)}, \boldsymbol{\alpha}^{(2)}\}$. For a binary classification problem, the Bayes' decision is based on,

$$\frac{P(\omega_1)\tau^{(2)}\overline{p}(\mathbf{o};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{P(\omega_2)\tau^{(1)}\overline{p}(\mathbf{o};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})} \begin{array}{c}\omega_1\\ >\\ <\\ \omega_2\end{array} 1 \tag{19}$$

where $P(\omega_1)$ and $P(\omega_2)$ are priors for the two classes. Taking logs of both sides, this may be expressed as,

$$\ln\left(\frac{\overline{p}(\mathbf{o};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{\overline{p}(\mathbf{o};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})}\right) + b \begin{array}{c}\omega_1\\ >\\ <\\ \omega_2\end{array} 0 \tag{20}$$

where the class priors and normalisation terms are combined as,

$$b = \ln\left(\frac{P(\omega_1)\tau^{(2)}}{P(\omega_2)\tau^{(1)}}\right) \tag{21}$$

Note that the priors for the classes are not trained using ML, but rather using maximum margin. Using Eq. 2, Eq. 20 can be rewritten as the scalar product, $(\cdot,\cdot)$,

$$\left(\begin{bmatrix}\mathbf{w}\\ w_0\end{bmatrix}, \begin{bmatrix}\boldsymbol{\phi}(\mathbf{o};\boldsymbol{\lambda})\\ 1\end{bmatrix}\right) \begin{array}{c}\omega_1\\ >\\ <\\ \omega_2\end{array} 0 \tag{22}$$

This now has the form of a *score-space*, $\boldsymbol{\phi}(\mathbf{o};\boldsymbol{\lambda})$, which is a function of the base-statistical model parameters,

$$\boldsymbol{\phi}(\mathbf{o};\boldsymbol{\lambda}) = \begin{bmatrix} \ln\check{p}(\mathbf{o};\boldsymbol{\lambda}^{(1)}) - \ln\check{p}(\mathbf{o};\boldsymbol{\lambda}^{(2)}) \\ \boldsymbol{\nabla}^{(\rho)}_{\boldsymbol{\lambda}^{(1)}} \ln\check{p}(\mathbf{o};\boldsymbol{\lambda}) \\ -\boldsymbol{\nabla}^{(\rho)}_{\boldsymbol{\lambda}^{(2)}} \ln\check{p}(\mathbf{o};\boldsymbol{\lambda}) \end{bmatrix} \tag{23}$$

and a linear decision boundary which is determined by the augmented model parameters $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)\dagger}$,

$$\begin{bmatrix}\mathbf{w}\\ w_0\end{bmatrix} = \begin{bmatrix}1\\ \boldsymbol{\alpha}^{(1)}\\ \boldsymbol{\alpha}^{(2)}\\ b\end{bmatrix} \tag{24}$$

One candidate for estimating the decision boundary is the Support Vector Machine (SVM). This is suitable for these forms of models as the decision boundaries that are estimated achieve good generalisation performance even when the dimensionality of the feature-space (in this case the score-space) is very large. This may be viewed as maximum margin training of the statistical models. If the SVM is trained, the parameters of the augmented model are given by††,

$$\alpha_0 \begin{bmatrix}1\\ \boldsymbol{\alpha}^{(1)}\\ \boldsymbol{\alpha}^{(2)}\end{bmatrix} = \alpha_0\mathbf{w} = \sum_{i=1}^{n} \alpha_i^{\mathtt{svm}} y_i \mathbf{G}^{-1}\boldsymbol{\phi}(\mathbf{o}_i;\boldsymbol{\lambda}) \tag{25}$$

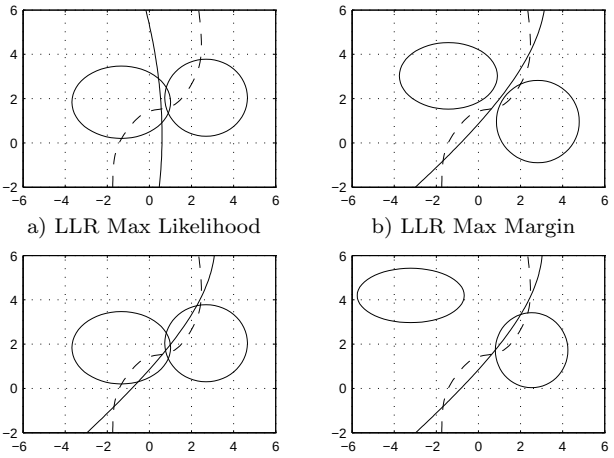
where $\mathbf{G}$ is given by Eq. 17. The additional scaling term α_0 has no affect on the decision boundary, but allows standard SVM training to be used††† (by avoiding the need to explicitly enforce the constraint, $w_1 = 1$).

One objection to the use of SVMs is that the distances from the decision boundaries do not have a statistical interpretation [15]. This has led to techniques that transform the output so that it is probabilistic in nature [16] and the use of the relevance vector machine [15]. However if generative kernels are used the distance from the decision boundary is directly interpretable as the log-posterior ratio of the two classes. This comes directly from Eq. 22. It is interesting to contrast this to MMI training [13]. In MMI training the average posterior of the correct label is optimised. In maximum margin training of this form, all correctly labelled points beyond the margin are ignored.

A simple example of maximum margin training of a statistical model is shown in Fig. 2 on artificial data generated using three-component GMMs per class. Here class-conditional single Gaussian component models are used as the base distribution. The ML estimate for this model is shown in Fig. 2(a) along with the SVM trained decision boundary from the one-dimension log-likelihood ratio (LLR) score-space,

$$\phi(\mathbf{o};\boldsymbol{\lambda}) = \left[\ln\check{p}(\mathbf{o};\boldsymbol{\lambda}^{(1)}) - \ln\check{p}(\mathbf{o};\boldsymbol{\lambda}^{(2)})\right] \tag{26}$$

and the Bayes' decision boundary (the dotted line).



a) LLR Max Likelihood b) LLR Max Margin

c) LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ Max Likelihood d) LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ Max Margin

Fig. 2 (a) Maximum Likelihood (ML) and (b) Maximum Margin (MM) distributions in a Log-Likelihood Ratio (LLR) score-space; and (c) ML base distribution and LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ score-space; (d) MM base distribution and a LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ score-space.

†Due to the definition of the bias b, there is some interaction between the base statistical model parameters and the augmented parameters, $\boldsymbol{\alpha}$

††The additional use of the metric $\mathbf{G}$ below is due to training the SVM using an inner product, whereas the augmented model is described in terms of a scalar product.

†††The values of the bias must also be scaled, hence $b = w_0/\alpha_0$.

The base acoustic model was then trained using maximum margin estimation using Eq. 18 and the score-space in Eq. 26. The decision boundary and positions of the Gaussians are shown in Fig. 2(b). The direction of the decision boundary is closer to that of the Bayes' decision boundary and is also reflected in the classification rate in this data. Figures (c) and (d) show the decision boundaries that result from using the LLR with derivatives of the mean and covariances. The decision boundaries that result from (c) and (d) are very similar to that in (b). This is because the class-conditional base distribution was a member of the exponential family (a Gaussian). Since the derivatives with respect to the means and the variances yield first and second order statistics [17], the final distribution should be the same as directly training class-conditional Gaussian distributions using maximum margin.

One of the issues with using maximum margin training with generative kernels is that the final distribution is not guaranteed to be a valid distribution (though a distance from the decision boundary will always be available). This is best illustrated by examining maximum margin training of a univariate Gaussian class-conditional distribution. Consider a mean and variance first derivative score-space. The resultant maximum margin variance is given by $\sigma^4/(\sigma^2 - \alpha)$ where α is associated with the variance derivative. Thus if $\alpha \geq \sigma^2$ the effective variance will not be positive definite.

For the training to directly correspond to estimating augmented model parameters a linear kernel in the score-space is required. However, non-linear kernels such as polynomial and Gaussian kernels are commonly used. Provided that the normalisation integral (the value of τ) is bounded, the distances from the decision may still be interpreted as a log-posterior ratio. However, this will not have the form of the augmented models described in Sect. 2.

5. LVCSR Decoding

SVMs are inherently binary classifiers. For Large Vocabulary Continuous Speech Recognition (LVCSR) there are a vast number of possible classes making one-versus-one binary classification impractical. In order to apply the maximum margin trained statistical models of the previous section to LVCSR it is necessary to map this highly complex classification problem into a set of binary classification problems.

To solve this problem an approach related to that described in [18] is used. Word-lattices are first generated by a standard, in this case HMM-based, LVCSR system. These lattices consist of nodes and arcs. The arcs are labelled with words, language and acoustic model likelihoods. The nodes are labelled with time stamps. The word-lattices are then converted to a *confusion network*. This consists of a series of nodes, with a linear graph. Each of the arcs is labelled with a word, a start and end time and a log-posterior, $\mathcal{F}(\omega_i)$. For details of this process see [19]. The confusion networks are then pruned so that at each node a maximum of two possible words occur. The pruning is achieved by simply selecting the words with the greatest posteriors.

Once a set of confusion pairs has been generated, it is possible to train a set of statistical models for each pair of data. One issue is the form of score-space to be used. From Eq. 19 only the unigram prior for the word is available (though the probabilities are trained in a maximum margin fashion). For LVCSR, trigram and higher order language models (LMs) are commonly used. This additional information can be incorporated into the binary classifier using the log-posteriors from the confusion networks. The log-posterior may be treated as an additional information source. Now the decision rule becomes,

$$\frac{1}{T}\ln\left(\frac{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})}\right)+b+\beta(\mathcal{F}(\omega_1)-\mathcal{F}(\omega_2)) \begin{array}{c}\omega_1 \\ \geq \\ < \\ \omega_2\end{array} 0$$

where β is an empirically set constant. This is a simple process as it only requires combining the log-posterior ratio with the distance from the SVM decision boundary. Alternatively the posterior may be combined into the score-space to give,

$$\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda}) = \begin{bmatrix} \mathcal{F}(\omega_1) - \mathcal{F}(\omega_2) \\ \frac{1}{T}\ln\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)}) - \ln\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)}) \\ \frac{1}{T}\boldsymbol{\nabla}^{(\rho)}_{\boldsymbol{\lambda}^{(1)}}\ln\check{p}(\mathbf{O};\boldsymbol{\lambda}) \\ -\frac{1}{T}\boldsymbol{\nabla}^{(\rho)}_{\boldsymbol{\lambda}^{(2)}}\ln\check{p}(\mathbf{O};\boldsymbol{\lambda}) \end{bmatrix} \quad (27)$$

This allows β to be set using maximum margin training. However empirically setting β has a number of advantages. The lattices used to generate the confusion networks are usually generated using HMMs that have been trained on all the acoustic data. This means that the posteriors obtained from the confusion networks are liable to be biased. Thus the β value will tend to be larger than when estimated using held out data.

6. Results

The database used for the LVCSR experiments was a 400 hour subset of the Fisher LDC data. This is the `fsh2004sub` data set used for initial system development [20]. The model set used was based on the standard front-end and models described in [20]. However for this work only ML, rather than discriminatively, trained acoustic models were used. The confusion networks were generated using a bigram language model on the same 400 hours as used to train the acoustic models. The held-out dataset for these experiments was the eval03 test set, which consists of 6 hours of data.

For initial assessment, 8-fold cross-validation experiments were carried out on the training data. For

Word Pair (examples)	Training	CN post.	# Components 1	2	4
A/THE (8533)	ML	79.8	58.3	58.4	56.2
	SVM $\boldsymbol{\phi}^{\mathtt{ll}}()$		61.1	63.0	64.7
	$+\beta$CN		79.8	80.0	80.3
	SVM $\boldsymbol{\phi}^{\mathtt{cn}}()$		80.4	80.1	80.6
CAN/CAN'T (3761)	ML	78.5	81.7	86.0	88.2
	SVM $\boldsymbol{\phi}^{\mathtt{ll}}()$		84.8	89.4	90.5
	$+\beta$CN		88.5	91.2	91.9
	SVM $\boldsymbol{\phi}^{\mathtt{cn}}()$		89.0	91.4	91.6
KNOW/NO (4475)	ML	83.1	68.4	69.4	70.8
	SVM $\boldsymbol{\phi}^{\mathtt{ll}}()$		72.1	73.6	76.6
	$+\beta$CN		84.3	84.5	85.2
	SVM $\boldsymbol{\phi}^{\mathtt{cn}}()$		85.7	86.2	86.2

Table 1 8-Fold cross-validation results (% correct) using variable number of components, ML training of the base model (ML) and SVM training with LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ ($\boldsymbol{\phi}^{\mathtt{ll}}()$) and LLR+$\boldsymbol{\nabla}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$+CN posterior ($\boldsymbol{\phi}^{\mathtt{cn}}()$) score-spaces with the ML model.

all experiments diagonal covariance matrix GMMs were trained using the longest time-stamps from the confusion networks[†] for the two confusable words. The number of examples for each word pairing were sampled so that the number of positive examples for each word is the same. This means that random selection will yield 50% correct. The GMMs were trained using ML. SVMs were then trained using a first order mean and covariance matrix score-space. A range of SVMs were trained and a few examples are shown in detail in table 1. For all cases using SVMs trained in the likelihood ratio plus derivative score-space gave performance gains over the ML trained base model. For the "CAN/CAN'T" pairing the ML GMM and SVM systems were better than the confusion network score used as a baseline. However in general this was not the case. For the "A/THE" pairing the performance was less than 60% for the ML GMM. Then, using the two forms of combining the confusion network posterior scores from Sect. 5, gains in performance were obtained for most cases. Though schemes where the ML GMM performance was poor, such as "A/THE", the gains were negligible. Using the score-space including the confusion network posterior gave consistent gains over simply interpolating the information sources.

To examine performance on held-out data the eval03 test set was used. This has a total of 76,157 words in the reference transcription. The baseline results using a bigram language model were 34.4% and 33.9% using Viterbi and Confusion Network (CN) decoding respectively, and the baseline numbers using a trigram were 30.8% and 30.1% respectively. As expected the use of CN decoding consistently decreased the error rate. These CN decoding results were used as the baseline for SVM rescoring. Table 2 shows the results of rescoring with 9 and 15 SVMs trained on confusion pairs from the 400 hour training set. All SVM rescoring was based on $\boldsymbol{\phi}^{\mathtt{ll}}() + \beta$CN, with β roughly tuned to the task[†]. As expected from table 1 there was a range of performances depending on the word pair. The best performance for the bigram LM was obtained using the "CAN/CAN'T" pairing. This reduced the number of errors by 17 in a total of 165 pairs (10.0% reduction). The use of the $\boldsymbol{\phi}^{\mathtt{cn}}()$ score-space to find β was worse than the standard CN decoding. This illustrates the dependence of the posterior scores on the exact acoustic/language models used. Overall, though the number of errors reduced was small, the percentage of pairs corrected, 3.1%, for 9 pairs indicates that the general approach may be useful. Even using SVMs based on 15 commonly confused pairs less than 3% of the hypothesised words were rescored.

SVM Rescoring	#corrected/#pairs (% corrected) bigram LM	trigram LM
9 SVMs	44/1401 (3.1%)	41/1310 (3.1%)
15 SVMs	55/2116 (2.6%)	43/1954 (2.2%)

Table 2 SVM rescoring giving change in number of errors compared to the CN decoding and total pairs rescored using $\boldsymbol{\phi}^{\mathtt{ll}}() + \beta$CN on eval03.

Second and higher order score-spaces yielded no improvement in performance. Since, for speech recognition, observations are sampled from a high-dimensional space, posterior probabilities of latent states are polarised to be either one or zero. Second derivatives (which are a function of the derivatives of the posteriors) are therefore negligible[††] and can be ignored.

7. Conclusion

This paper has described the general form of augmented statistical models. These models are specified by a base distribution and a local exponential family approximation to that distribution at a particular point. The

[†]By the longest time-stamps the earliest time of the two words and the latest time of the two words is used. This is required as the confusion network times are generated by taking the earliest and latest times that contribute to an arc.

[†]The performance was relatively insensitive to differences in the value of β.

[††]For example, consider the second derivative of a GMM with respect to $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_k$, $k \neq j$,

$$\nabla_{\boldsymbol{\mu}_k}\nabla^T_{\boldsymbol{\mu}_j} \ln p(\boldsymbol{o};\boldsymbol{\lambda}) = -P(j|\boldsymbol{o};\boldsymbol{\lambda})P(k|\boldsymbol{o};\boldsymbol{\lambda})\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_k)(\boldsymbol{o}-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}$$

When posteriors are one or zero, at least one of $P(j|\boldsymbol{o};\boldsymbol{\lambda})$ and $P(k|\boldsymbol{o};\boldsymbol{\lambda})$ must be zero (since observations cannot be generated by two components simultaneously). The second derivative is zero.

This means that, given a full score-space (with derivatives with respect to all parameters), maximum margin training of base model parameters yields no improvement in performance [21]. Note that, for the single component case, the SVM trained augmented model is the equivalent of a maximum margin trained model.

statistics used for the exponential model are based on first, and higher order, derivatives of the base distribution. These models are difficult to train because of the potentially large numbers of model parameters and issues in determining the normalisation term. This paper shows that these augmented models can be trained for a two-class problem using maximum margin training. This is directly related to the use of generative kernels within an SVM framework. Initial experiments using SVM training on a large vocabulary speech recognition task indicate that this form of modelling and training may be useful for speech recognition.

Acknowledgement

The authors would like to thank Nathan Smith for many helpful discussions in preparing this paper. Bennett Rogers helped with initial work on the LVCSR task. Martin Layton would like to thank the Schiff Foundation for funding. Extensive use was made of equipment supplied to the Speech Group at Cambridge University by IBM under an SUR award.

References

[1] M. Ostendorff, V.V. Digalakis, and O.A. Kimball, "From HMMs to segment models: A unified view of stochastic modelling for speech recognition," IEEE Transactions Speech and Audio Processing, vol.4, pp.360–378, 1996.

[2] A.V.I. Rosti and M.J.F. Gales, "Switching linear dynamical systems for speech recogntion," Tech. Rep. CUED/F-INFENG/TR461, Cambridge University, 2003. Available from: svr-www.eng.cam.ac.uk/˜mjfg.

[3] J. Bilmes, "Buried Markov models: A graphical-modelling approach to automatic speech recognition," Computer Speech and Language, vol.2-3, 2003.

[4] H. Nock, Techniques for modelling Phonological Processes in Automatic Speech Recognition, Ph.D. thesis, Cambridge University, 2001.

[5] S. Amari and H. Nagaoka, Methods of Information Geometry, Oxford University Press, 2000. (Translations of Mathematical Monographs, Volume 191, American Mathematical Society).

[6] N. Smith, Using Augmented Statistical Models and Score Spaces for Classification, Ph.D. thesis, University of Cambridge, September 2003.

[7] N.D. Smith and M.J.F. Gales, "Speech recognition using SVMs," Advances in Neural Information Processing Systems, 2001.

[8] M. Layton and M.J.F. Gales, "Maximum margin training of generative kernels," Tech. Rep. CUED/F-INFENG/TR.484, Department of Engineering, University of Cambridge, June 2004.

[9] M. Layton and M. Gales, "Acoustic modelling using continuous rational kernels," IEEE International Workshop on Machine Learning for Signal Processing, September 2005. To appear.

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

[11] T. Cover, "Geometric and statistical properties of systems of linear inequalities with applications in pattern recognition," IEEE Transactions on Electronic Computers, EC14(3), pp.326–334, June 1965.

[12] T. Jaakkola and D. Hausser, "Exploiting generative models in disciminative classifiers," Advances in Neural Information Processing Systems 11, ed. S. Solla and D. Cohn, pp.487–493, MIT Press, 1999.

[13] P. Brown, The Acoustic-Modelling Problem in Automatic Speech Recognition, Ph.D. thesis, IBM T.J. Watson Research Center, 1987.

[14] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, Royal Holloway, University of London, May 1998.

[15] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," Journal of Machine Learning Research, pp.211–244, 2001.

[16] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," Advances in Largin Margin Classifiers, 2000.

[17] N. Smith, M. Gales, and M. Niranjan, "Data dependent kernels in SVM classification of speech patterns," Tech. Rep. CUED/F-INFENG/TR.387, Department of Engineering, University of Cambridge, April 2001.

[18] V. Venkataramani, S. Chakrabartty, and W. Byrne, "Support vector machines for segmental minimum Bayes risk decoding of continuous speech," ASRU 2003, 2003.

[19] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," Proc. Eurospeech, 1999.

[20] G. Evermann, H. Chan, M. Gales, B. Jia, D. Mrva, P. Woodland, and K. Yu, "Training LVCSR systems on thousands of hours of data," Proc. ICASSP, 2005.

[21] M. Layton and M. Gales, "Augmented statistical models for ASR," Submitted to NIPS 2005. Available from http://mi.eng.cam.ac.uk/˜ml362.

Mark Gales Mark Gales studied for the B.A. in Electrical and Information Sciences at the University of Cambridge from 1985-88. 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 in the Engineering Department at Cambridge University. In 1995 he completed his doctoral thesis: Model-Based Techniques for Robust Speech Recognition supervised by Professor Steve Young. From 1995-1997 he was a Research Fellow at Emmanuel College Cambridge. He was then a Research Staff Member in the Speech group at the IBM T.J.Watson Research Center until 1999 when he returned to Cambridge University Engineering Department as a University Lecturer. He is currently a Reader in Information Engineering and a Fellow of Emmanuel College. Mark Gales is a member of the IEEE and was a member of the Speech Technical Committee from 2001-2004.

Martin Layton Martin Layton studied for the B.Sc. in Mathematics and Physics at the University of Bristol from 1999-2002. After graduation he worked for a year at Dresdner Kleinwork Wasserstein. In 2003 he began working towards the degree of Doctor of Philosophy with a working title of "Kernel Methods for Classifying Variable Length Data", supervised by Dr Mark Gales.