CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

USING SVMS TO CLASSIFY VARIABLE LENGTH SPEECH PATTERNS

N.D. Smith [1] and M.J.F. Gales [2]

CUED/F-INFENG/TR.412

April 2002
(updated April 25, 2002)

[1] Cambridge University Engineering Dept.
Trumpington Street, Cambridge, CB2 1PZ, U.K.
E-mail: nds1002@eng.cam.ac.uk
http://svr-www.eng.cam.ac.uk/~nds1002

[2] Cambridge University Engineering Dept.
Trumpington Street, Cambridge, CB2 1PZ, U.K.
E-mail: mjfg@eng.cam.ac.uk
http://svr-www.eng.cam.ac.uk/~mjfg

Abstract

Support Vector Machines (SVMs) have shown good performance in a range of classification tasks. This report investigates the application of SVMs to the classification of variable length speech patterns. SVMs are inherently static, binary classifiers but speech recognition is a dynamic, multi-class task. There are therefore two main issues to resolve in applying SVMs to speech recognition; mapping variable length sequences to vectors of fixed dimension, and handling multiple classes. This report only concentrates on the first of these two issues. For a binary classification problem, variable length speech sequences can be mapped to vectors of fixed dimension using a mapping called the *Fisher score*. The Fisher score was originally defined using a log likelihood model for the data belonging to both competing classes, and was taken as the gradient of the log likelihood with respect to the model parameters. In this report, Fisher scores are extended to a range of mappings called score-spaces. Score-spaces are defined on the log likelihood or log posterior of model estimates. Under various conditions, score-spaces can be used to recover optimal minimum Bayes error classifiers with SVMs as the natural choice of classifier for these score-spaces. In more realistic problems, score-spaces can be regarded as techniques which extract extra discriminative information from generative models. The different score-spaces are tested on a speaker independent, isolated letter, speech recognition task. Relevant normalisation issues and feature selection techniques are addressed.

Note: Any ammendments or revisions to this report will be made available on N.Smith's personal webpage with the address given on the front cover; please download the latest version if you are able

1 Introduction

Speech recognition is a complicated, dynamic classification task. State-of-the-art recognition systems often use Hidden Markov Models (HMMs) trained to maximise likelihood or discriminative cost functions. One of the reasons for the popularity of HMMs is their ability to classify variable length sequences which occur naturally in speech due to variations in speaker, speaking rate and phonetic context. HMMs also readily handle the multi-class aspects of speech recognition since they are used to output likelihoods which can be incorporated into Bayesian decision rules. Recently, good classification performance on static classification tasks has been obtained by a technique called Support Vector Machines (SVMs) [23] [2]. These results encourage the application of SVMs to speech recognition. Unfortunately, SVMs are inherently static, binary classifiers. There are therefore two main problems to overcome in applying SVMs to speech recognition. First, how to map variable length sequences to vectors of fixed dimension which can then be classified using static SVMs. Second, how to obtain a multi-class decision from these binary classifiers. This report only concentrates on the first of these two problems. For the second problem, a simple majority voting scheme among a full set of binary classifiers is used (some alternative techniques are described in [15], [16] and [9]).

This report extends a technique called the *Fisher-score* [10] which can be used to map variable length sequences to vectors of fixed dimension. For a two-class problem, the mapping was originally defined using a generative model estimate for the data belonging to both classes. The Fisher-score is the vector of first-order derivatives of the log likelihood estimate with respect to each model parameter. Fisher-scores were later extended to more general *score-spaces* in [19], and in particular, score-spaces defined on the log ratio between likelihood estimates of competing classes[1]. Any static classifier can be used to yield decision rules in these score-spaces. However since score-spaces are typically high-dimensional, SVMs are a good choice of classifier for two-class problems since they have been shown to generalise well when the numbers of training samples are few relative to the size of the input space. This report introduces 'derivative score-spaces'. First an output is obtained, for example the log likelihood or log posterior model estimate for a class, or the log ratio between the likelihood and posterior model estimates for competing classes. Next, the zeroth, first and higher-order derivatives of the output are obtained with respect to the parameters of the relevant model estimates. The resulting vector-space will be called a 'derivative score-space'. A justification for the use of such score-spaces is presented. Under certain conditions, these score-spaces may be used to recover optimal minimum Bayes error classifiers. Unfortunately, these conditions are unlikely to be fulfilled for score-spaces defined on the HMM model estimates which are standard in speech recognition, mainly because of the conditional independence assumptions implicit in these model estimates. However, these score-spaces can still be used to extract extra discriminative information from the model estimates, and to obtain better approximations to optimal minimum Bayes error classifiers.

This report is organised as follows. Section 2 introduces the framework for score-spaces and gives a brief review of recent literature. Section 3 then presents a formal justification for derivative score-spaces by relating linear classifiers built in these score-space to optimal minimum Bayes error classifiers. Under certain conditions, SVMs are shown to be a natural choice of classifier for such score-spaces. Section 4 then details the mathematical formulation for the particular score-spaces used in the experiments in this report. Some normalisation issues are also highlighted. Section 5 gives some insight into class discrimination in score-spaces. It compares the discrimination performance between two of these derivative score-spaces, the likelihood-ratio and likelihood (1 class) score-spaces. It also discusses the effect of tying parameters across the models which define score-spaces. Section 6 then introduces feature selection techniques which can be used to decrease the size of the score-space and eliminate noisy dimensions. Section 7 then describes a set of experiments on a speaker-independent, isolated letter task. These experiments compare the performances of the different score-spaces and standard HMM schemes, and the effects of the normalisation and feature selection techniques. Finally, conclusions are drawn in Section 8[2].

2 Introducing score-spaces

This section presents the framework used to define score-spaces and summarises some recent research. Score-spaces were primarily developed to map variable length sequences to vectors of fixed-dimension [10]. This immediately allows static classifiers to be used in dynamic classification tasks. To describe the framework for

[1]This report develops our earlier research in [19] and is summarised in [18].

[2]In this report, normal font is used for scalars and scalar functions, and bold font for vectors, matrices and vector functions.

score-spaces, first consider an observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2 \dots \boldsymbol{o}_t \dots \boldsymbol{o}_T)$ where $\boldsymbol{o}_t \in \mathbb{R}^D$. Also consider a set of probability models for this observation sequence $\mathcal{P} = \{p_q(\boldsymbol{O}|\omega_q)\}$ (ω_q may be used to represent any subset of the data, and is later used to represent the qth class). Following the arguments and assumptions in [10], each observation sequence $\boldsymbol{O}$ can be mapped into a vector of fixed dimension $\boldsymbol{\varphi}^f_{\hat{F}}(\boldsymbol{O})$,

$$\boldsymbol{\varphi}^f_{\hat{F}}(\boldsymbol{O}) \quad = \quad \boldsymbol{\varphi}_{\hat{F}} f\big(\{p_q(\boldsymbol{O}|\omega_q)\}\big) \tag{1}$$

For convenience, the following terminology is used in this report.

- The probability models in the set $\mathcal{P}$ are called *score-models*. Score-models should reflect the nature of the classification problem. For example, HMMs are used in speech recognition to capture the dynamic information in the speech signal. For other classification tasks, different probabilistic models may be used such as Gaussian Mixture Models (GMMs) or discrete density HMMs. The score-models may be trained generatively to reflect data density, or may be trained in a discriminative manner [20].
- A particular function of the score-models $f(\cdot)$ is called the *score-argument*. The score-argument should be a principled description of how the score-models are used, in this case, for classification. The score-argument may be specified as the output from a single score-model, the ratio of outputs from two score-models, or a function of the outputs from a larger subset of score-models. For example, consider the original definition of the *Fisher-score* [10]. For a two-class problem, the score-model was simply the generative model for both classes, and the corresponding score-argument was the log of its output, otherwise known as its log likelihood. Similarly, [5] used the same score-argument but defined the score-model as the generative model for just one of the classes rather than both. For a two-class problem, [19] trained a generative model for each competing class and defined the score-argument as the log ratio between the two model outputs. [19] also suggested the use of a posterior probability function as the score-argument. The suitability of a particular score-argument for class discrimination is affected by the training criterion used for the score-models.
- The score-argument is mapped by a *score-operator*, $\hat{F}$. The score-operator should extract useful discriminative information from the score-models. One source of information is the sensitivity of the model outputs to their parameter settings. This was the original motivation behind the gradient score-operator in the Fisher-score [10], where the gradient or first-order derivative of the log likelihood function was taken with respect to the score-model parameters. This was also used as the score-operator in [19]. However the score-operator may also include other sources of additional, ideally complimentary, information. For example, it is sensible to include the outputs taken directly from the score-models themselves. It may also be useful to take other higher-order derivatives with respect to score-model parameters.
- The score-operator maps into a fixed-dimensional space $\boldsymbol{\varphi}^f_{\hat{F}}(\cdot)$ called the *score-space*. The different vectors which occupy this score-space, $\boldsymbol{\varphi}^f_{\hat{F}}(\boldsymbol{O})$ are simply called *scores*. A score-space is therefore defined by its score-models and their training criterion, and its score-argument and score-operator.

This report proposes the use of standard continuous density HMMs as score-models. Despite the conditional independence assumptions implicit in these models, HMMs have been successfully used in speech recognition (as explained later, this conditional independence assumption is one of the main factors which limit the performance of classifiers constructed in score-spaces for speech recognition). These score-models are used as class-conditional generative model estimates, and hence are used to yield class likelihood estimates. In this report, various score-arguments are used. These include the log output from the score-model for a single class, the log ratio between outputs from score-models for competing classes, and the log posterior for a class. The score-operators used are the zeroth-order, first-order and higher-order derivatives with respect to the parameters of the score-models. The resulting score-spaces can therefore be interpreted as 'derivative-based' score-spaces. A brief justification for these score-spaces follows. For purposes of generality, the explanation is set in the context of multi-class classification. However the experiments in this report only use score-spaces to tackle two-class problems. Binary decisions are then mapped to multi-class decisions using a post-processing majority voting scheme.

3 Score-spaces based on derivatives

Given a set of score-models and a score-argument, derivative operators may be used to map utterances into a score-space. The derivatives should be taken with respect to the model parameters. The resulting score-spaces are termed *derivative score-spaces*. This section presents a brief justification for derivative score-spaces, and illustrates how they may be used to recover better classifiers. SVMs are proposed as a sensible, and under certain conditions, the most appropriate classifier to use in these derivative score-spaces. First, some basic decision theory is reviewed to set the context for the analysis.

For context, the general Q-class problem is presented. Consider a set of training observation sequences, or utterances, $\{\boldsymbol{O}^1, \boldsymbol{O}^2 \dots \boldsymbol{O}^r \dots \boldsymbol{O}^R\}$ such that $\boldsymbol{O}^r = (\boldsymbol{o}_1^r, \boldsymbol{o}_2^r \dots \boldsymbol{o}_t^r \dots \boldsymbol{o}_{T_r}^r)$ and $\boldsymbol{o}_t^r \in \mathbb{R}^D$. Each utterance $\boldsymbol{O}^r$ belongs to one of Q classes and hence has class label $y_r = \omega_i$ where $\omega_i \in \{\omega_1 \dots \omega_Q\}$. Each class ω_i has R_i training utterances so that $R = \sum_{i=1}^{Q} R_i$. Each class has its own generative model. For an utterance $\boldsymbol{O}$, the generative model for class ω_i yields a class-conditional likelihood $p(\boldsymbol{O}|\omega_i)$. Each class ω_i also has a class prior $P(\omega_i)$ (in this report class priors are assumed fixed and independent of the parameters of generative models, but not necessarily known). The full set of class-conditional likelihoods and class priors may be used to define the posteriors for each class $P(\omega_i|\boldsymbol{O})$,

$$P(\omega_i|\boldsymbol{O}) = \frac{p(\boldsymbol{O}|\omega_i)P(\omega_i)}{\sum_{q=1}^{Q} p(\boldsymbol{O}|\omega_q)P(\omega_q)} \tag{2}$$

The class priors and class posteriors are by definition valid probability mass functions. The generative models and prior information are correct, so that the likelihoods, priors and posteriors are termed *true class likelihoods*, *true class priors* and *true class posteriors*. Next consider how these likelihoods and probability mass functions may be used to classify an unlabelled utterance $\boldsymbol{O}$. A popular classifier is the *Maximum A-Posteriori (MAP) decoder*. It assigns the utterance to the class $\omega_{\text{MAP}}(\boldsymbol{O})$ with maximum class posterior. The decision rule can be expressed in terms of class posteriors, or class likelihoods and class priors, as following,

$$\begin{aligned} \omega_{\text{MAP}}(\boldsymbol{O}) &= \operatorname*{argmax}_{\omega_i} \quad \ln P(\omega_i|\boldsymbol{O}) && (3) \\ &= \operatorname*{argmax}_{\omega_i} \quad \ln \frac{p(\boldsymbol{O}|\omega_i)P(\omega_i)}{\sum_{q=1}^{Q} p(\boldsymbol{O}|\omega_q)P(\omega_q)} && (4) \end{aligned}$$

A classifier which implements this decision rule is also known as an optimal *minimum Bayes error classifier* since it minimises the probability of error, or error rate[3]. A special case of the Q-class problem is the binary problem where $Q = 2$. This is of particular interest since SVMs are inherently binary classifiers. The log ratio of class likelihoods $\rho(\boldsymbol{O})$ and class posteriors $\rho_{\text{pos}}(\boldsymbol{O})$ may be expressed as follows,

$$\begin{aligned} \rho(\boldsymbol{O}) &= \ln \frac{p(\boldsymbol{O}|\omega_1)}{p(\boldsymbol{O}|\omega_2)} && (5) \\ \rho_{\text{pos}}(\boldsymbol{O}) &= \ln \frac{P(\omega_1|\boldsymbol{O})}{P(\omega_2|\boldsymbol{O})} \\ &= \ln \frac{p(\boldsymbol{O}|\omega_1)P(\omega_1)}{p(\boldsymbol{O}|\omega_2)P(\omega_2)} \\ &= \rho(\boldsymbol{O}) + \ln \frac{P(\omega_1)}{P(\omega_2)} && (6) \end{aligned}$$

The MAP decision rule may therefore be written in terms of $\rho_{\text{pos}}(\boldsymbol{O})$,

$$\omega_{\text{MAP}}(\boldsymbol{O}) = \begin{cases} \omega_1 & \text{if} \quad \rho_{\text{pos}}(\boldsymbol{O}) > 0 \\ \omega_2 & \text{otherwise} \end{cases} \tag{7}$$

The rule may also be written in terms of $\rho(\boldsymbol{O})$,

$$\omega_{\text{MAP}}(\boldsymbol{O}) = \begin{cases} \omega_1 & \text{if} \quad \rho(\boldsymbol{O}) + \ln P(\omega_1) - P(\omega_2) > 0 \\ \omega_2 & \text{otherwise} \end{cases} \tag{8}$$

Since these expressions for a MAP decoder are based on true generative models, the resulting decoder is optimal. Optimality is used in the sense that such a classifier attains the lowest possible error rate, since the true generative models are by definition the most discriminative models for the classification problem. In this report, the term 'optimal MAP decoder' is strictly reserved for such a classifier. In real problems, such as

[3] Test error rates derived from experiments are dependent on the size and content of the test set, and are only guaranteed to be equal to the error rate for infinite test sets.

the classification of speech patterns, the true generative models are rarely known. Therefore, there is often no means of evaluating the true class likelihoods $p(\boldsymbol{O}|\omega_i)$ or true class posteriors $P(\omega_i|\boldsymbol{O})$. Instead, the class likelihoods and class posteriors must be estimated by proposing various functional forms for the generative models for each class, and then estimating their parameters. In real-world problems, the functional forms of the proposed models are usually much simpler than the forms of the true models. These models are used to output estimates for class likelihoods and class posteriors. For an utterance $\boldsymbol{O}$, let the class likelihood estimate be $\hat{p}(\boldsymbol{O}|\omega_i)$ and the class posterior estimate be $\hat{P}(\omega_i|\boldsymbol{O})$. For clarity, the log ratios of class likelihood estimates and class posterior estimates are denoted by $\hat{\rho}(\boldsymbol{O})$ and $\hat{\rho}_{\text{pos}}(\boldsymbol{O})$ where,

$$\hat{\rho}(\boldsymbol{O}) = \ln \frac{\hat{p}(\boldsymbol{O}|\omega_1)}{\hat{p}(\boldsymbol{O}|\omega_2))} \tag{9}$$

$$\hat{\rho}_{\text{pos}}(\boldsymbol{O}) = \ln \frac{\hat{P}(\omega_1|\boldsymbol{O})}{\hat{P}(\omega_2|\boldsymbol{O})} \tag{10}$$

When the ratios $\hat{\rho}_{\text{pos}}(\boldsymbol{O})$ and $\hat{\rho}(\boldsymbol{O})$ are substituted for $\rho_{\text{pos}}(\boldsymbol{O})$ and $\rho(\boldsymbol{O})$ in Equations 7 and 8, the error rates of the resulting classifiers ordinarily increase. For a classification problem, the lowest possible error rate, or lower bound on error rate, is only guaranteed when the estimated likelihoods are identical to the true likelihoods over the whole input space of utterances. This occurs when the estimated generative models are identical to the true generative models, that is under conditions of *model correctness*. If the models are not correct, there is no guarantee that the classifier is optimal, and the classifier is simply a MAP decoder or minimum Bayes error classifier subject to the given model estimates. The error rate obtained using the correct models must be the lowest possible for the classification task since correct models are by definition the most discriminative for the classification task. In real-world classification problems, the model estimates are usually incorrect, and the resulting classifiers are therefore usually suboptimal. It would be valuable to investigate techniques which aim to use incorrect model estimates to obtain better approximations to optimal MAP decoders. This section investigates how score-spaces are one such technique. First, a simple case is presented where the estimated models have the correct functional form, but their parameters differ from those of the true models. In this case and with sufficient training data, the optimal MAP decoder can be recovered using score-spaces. Next, a more complex case is presented where the estimated models form a subset in the parameter space of the corresponding true models. In this case, optimal MAP decoders can only be recovered using score-spaces under certain conditions. SVMs are shown to be a sensible choice of classifier for two-class problems in such score-spaces. Finally, the practicality of using score-spaces for real-world problems such as speech classification is addressed, where the true models are much more complex than the model estimates. In these real-world problems, it is unlikely that score-spaces can be used to yield optimal MAP decoders, but they may nevertheless yield good approximations to them.

3.1 Estimated models of the correct functional form

To introduce the reasoning behind score-spaces based on derivatives, consider the simple case where the functional forms of the true class-conditional generative models are known. For class ω_i, let the functional form of the true generative model be $p_i(\boldsymbol{O}|\boldsymbol{a})$, where $\boldsymbol{a} \in \boldsymbol{\Theta}_i$ and $\boldsymbol{\Theta}_i$ is a vector-space of parameters. Let the true generative model be parameterised by $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}_i$. In many cases, this parameterisation is unknown and various criteria may be used to provide an estimate $\hat{\boldsymbol{\theta}}_i \in \boldsymbol{\Theta}_i$. The estimate may differ from $\boldsymbol{\theta}_i$ due to a lack of data or to local minima/maxima in the cost function. The true and estimated class likelihoods $p(\boldsymbol{O}|\omega_i)$ and $\hat{p}(\boldsymbol{O}|\omega_i)$ may be written as following,

$$p_i(\boldsymbol{O}|\boldsymbol{\theta}_i) \equiv p(\boldsymbol{O}|\omega_i) \tag{11}$$

$$p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \equiv \hat{p}(\boldsymbol{O}|\omega_i) \tag{12}$$

Note that the functional form of the class-conditional generative model for class ω_i is strictly represented by $p_i(\boldsymbol{O}|\boldsymbol{a})$ for any $\boldsymbol{a} \in \boldsymbol{\Theta}_i$ (the functional forms of different classes may differ, though this option is never investigated in the experiments in this report). The output from this generative model under a distinct parameterisation is the class likelihood under that parameterisation. So then, the *true class likelihood* is given by $p_i(\boldsymbol{O}|\boldsymbol{\theta}_i)$, while the *estimated class likelihood* is given by $p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$.

Next, a simple score-space is derived for a Q-class problem. Consider again that the true and estimated class likelihoods are respectively $\{p(\boldsymbol{O}|\omega_i)\}_{i=1\ldots Q}$, and $\{\hat{p}(\boldsymbol{O}|\omega_i)\}_{i=1\ldots Q}$. The parameters of the true generative models may be summarised by $\boldsymbol{\xi} = (\boldsymbol{\theta}_1^\top, \ldots \boldsymbol{\theta}_i^\top, \ldots \boldsymbol{\theta}_Q^\top)^\top$ and the parameters of the estimated generative models by $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\theta}}_1^\top, \ldots \hat{\boldsymbol{\theta}}_i^\top \ldots \hat{\boldsymbol{\theta}}_Q^\top)^\top$. For each class ω_i, the true class log likelihood for utterance $\boldsymbol{O}$, $\ln p(\boldsymbol{O}|\omega_i)$, can be expanded about its parameter estimate $\hat{\boldsymbol{\theta}}_i$ using a Taylor series expansion. Here and elsewhere in the report, vec$(\cdot)$ is the vec operator which maps a matrix to a column vector, $\otimes$ is the Kronecker matrix product, ∇ is the derivative operator which returns a column vector, and $\nabla^2_{\hat{\boldsymbol{\theta}}_i}$ denotes $\nabla_{\hat{\boldsymbol{\theta}}_i}\nabla^\top_{\hat{\boldsymbol{\theta}}_i}$ [19]. For simplicity of notation, $\nabla_{\boldsymbol{\theta}_i}$ is an abbreviation for the derivative operator which evaluates the gradient of its argument at the parameterisation $\boldsymbol{\theta}_i$. So for example, for any $\boldsymbol{a} \in \boldsymbol{\Theta}_i$,

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}_i} \ln p_i(\boldsymbol{O}|\boldsymbol{\theta}_i) &\equiv \left(\nabla_{\boldsymbol{a}} \ln p_i(\boldsymbol{O}|\boldsymbol{\theta}_i)\right)\big|_{\boldsymbol{a}=\boldsymbol{\theta}_i} \\
\nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) &\equiv \left(\nabla_{\boldsymbol{a}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)\right)\big|_{\boldsymbol{a}=\hat{\boldsymbol{\theta}}_i}
\end{aligned}$$

Continuing the expansion of $\ln p(\boldsymbol{O}|\omega_i)$ about the parameter estimate $\hat{\boldsymbol{\theta}}_i$, and summarising terms with order higher than two by $O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3)$,

$$\begin{aligned}
\ln p(\boldsymbol{O}|\omega_i) &= \ln p_i(\boldsymbol{O}|\boldsymbol{\theta}_i) \\
&= \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) + (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^\top \nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) + \frac{1}{2}(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^\top [\nabla^2_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)](\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) + \\
&\quad O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3) \\
&= \begin{bmatrix} 1 \\ (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) \\ \frac{1}{2}[(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) \otimes (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)] \end{bmatrix}^\top \begin{bmatrix} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)] \end{bmatrix} + O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3) \\
&= \boldsymbol{w}^\top_{\mathrm{lk(i)}} \boldsymbol{\varphi}^{\mathrm{lk(i)}}(\boldsymbol{O}) + O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3) \qquad (13)
\end{aligned}$$

The Taylor series may be expanded to an arbitrarily high number of derivatives n so that the remainder $O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^{n+1})$ gradually diminishes[4]. As $n \to \infty$, the term $\boldsymbol{w}^\top_{\mathrm{lk(i)}} \boldsymbol{\varphi}^{\mathrm{lk(i)}}(\boldsymbol{O})$ represents an increasingly accurate approximation to the true log likelihood $\ln p(\boldsymbol{O}|\omega_i)$. In the limit, $\boldsymbol{\varphi}^{\mathrm{lk(i)}}(\boldsymbol{O})$ is a score in an infinite-dimensional score-space formed by an infinite number of derivatives with respect to the parameter estimates. The score-space is termed the *likelihood (1 class) score-space* for class ω_i. In this limit, the true log likelihood is the projection of its score onto a hyperplane in likelihood (1 class) score-space. The weight vector of the hyperplane is $\boldsymbol{w}_{\mathrm{lk(i)}}$. Since its elements are formed from Taylor series terms dependent on the true parameters $\boldsymbol{\theta}_i$ which are unknown, the elements of the weight vector $\boldsymbol{w}_{\mathrm{lk(i)}}$ must be estimated.

A linear classifier in such an infinite-dimensional score-space can be related to an optimal *likelihood threshold classifier*. For example, for an utterance $\boldsymbol{O}$, a threshold classifier takes the true class log likelihood $\ln p(\boldsymbol{O}|\omega_i)$ and sets a threshold $b_{\mathrm{lk(i)}}$. The utterance is classified as belonging to class ω_i if the log likelihood is above the threshold, else it is classified to the set of competing classes. The equivalent decision boundary in score-space is as follows, where the score-space has infinite dimension,

$$\begin{aligned}
\ln p(\boldsymbol{O}|\omega_i) &= b_{\mathrm{lk(i)}} \\
\boldsymbol{w}^\top_{\mathrm{lk(i)}} \boldsymbol{\varphi}^{\mathrm{lk(i)}}(\boldsymbol{O}) &= b_{\mathrm{lk(i)}} \\
\boldsymbol{w}^\top_{\mathrm{lk(i)}} \boldsymbol{\varphi}^{\mathrm{lk(i)}}(\boldsymbol{O}) - b_{\mathrm{lk(i)}} &= 0 \qquad (14)
\end{aligned}$$

With a sufficient quantity of labelled scores, it may be possible to specify a threshold $b_{\mathrm{lk(i)}}$ and train the optimal threshold classifier represented by the hyperplane $\boldsymbol{w}_{\mathrm{lk(i)}}$. The ability to train such an optimal classifier depends on the training algorithm for the linear classifier in score-space. Ideally an algorithm should yield a single solution which coincides with the optimal solution. Perceptrons are unsuitable since they give an infinite number of solutions when the classes are non-overlapping. However, an SVM has a unimodal cost function even when classes overlap, and hence returns a single solution[5]. Therefore an SVM is a suitable classifier for score-space, though as with all classifiers, coincidence of its solution with the optimal solution is difficult to predict or guarantee.

A disadvantage of likelihood threshold classifiers is that they take no account of the likelihood of competing classes in classifying utterances (although if a data-dependent threshold $b_{\mathrm{lk(i)}}$ is trained, then there is some

[4]There are many repeated elements in the vectors associated with the second and higher-order derivative terms. The inclusion or omission of these elements, while not affecting the information-content in the score-space, may affect the position of linear decision boundaries trained in score-space, depending on the form of score-space normalisation and particular training algorithm used (see Section 4.3 for more details).

[5]In practice, the solution returned by SVM training algorithms depend on convergence properties and the termination criterion employed.

limited accountability of competing classes). These threshold classifiers are naive and suboptimal. Better classifiers result from considering the ratios of likelihoods between competing classes. Consider the two-class problem where $Q = 2$. A likelihood-ratio classifier can be recovered through the use of score-spaces. In this case, denote the log likelihood-ratio between the true class log likelihoods as $\rho(\boldsymbol{O})$, and the corresponding ratio between the class log likelihood estimates as $\hat{\rho}(\boldsymbol{O})$. The log ratio $\rho(\boldsymbol{O})$ may be expanded using a Taylor series as previously. For convenience, the terms with order higher than two are summarised by $O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^3, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^3)$. So then,

$$\begin{aligned}
\rho(\boldsymbol{O}) &= \ln \frac{p(\boldsymbol{O}|\omega_1)}{p(\boldsymbol{O}|\omega_2)} = \ln \frac{p_1(\boldsymbol{O}|\boldsymbol{\theta}_1)}{p_2(\boldsymbol{O}|\boldsymbol{\theta}_2)} \\
&= \begin{bmatrix} 1 \\ (\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1) \\ \frac{1}{2}[(\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1) \otimes (\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)] \\ (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2) \\ \frac{1}{2}[(\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2) \otimes (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)] \end{bmatrix}^\top \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_1} \hat{\rho}(\boldsymbol{O})] \\ \nabla_{\hat{\boldsymbol{\theta}}_2} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_2} \hat{\rho}(\boldsymbol{O})] \end{bmatrix} + O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^3, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^3) \\
&= \boldsymbol{w}_{\mathrm{lr}}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) + O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^3, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^3) \qquad (15)
\end{aligned}$$

Again the Taylor series may be expanded to an arbitrary high number of derivatives n, so that the remainder $O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^{n+1}, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^{n+1})$ gradually diminishes. Again, as $n \to \infty$, $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$ becomes a score in an infinite-dimensional score-space. This score-space is termed the *likelihood-ratio score-space*. $\rho(\boldsymbol{O})$ is the projection of a score onto the hyperplane defined by the weight vector $\boldsymbol{w}_{\mathrm{lr}}$. Since the weight vector is unknown, it must be estimated in some manner. For an utterance $\boldsymbol{O}$, the ratio $\rho(\boldsymbol{O})$ may be incorporated into an optimal likelihood-ratio classifier using a threshold b_{lr}. The utterance is assigned to class ω_1 if $\rho(\boldsymbol{O})$ exceeds the threshold b_{lr}, else it is assigned to class ω_2. This decision boundary may be expressed in terms of a hyperplane in an infinite-dimensional score-space,

$$\begin{aligned}
\rho(\boldsymbol{O}) &= b_{\mathrm{lr}} \\
\ln p(\boldsymbol{O}|\omega_1) - \ln p(\boldsymbol{O}|\omega_2) &= b_{\mathrm{lr}} \\
\boldsymbol{w}_{\mathrm{lr}}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) &= b_{\mathrm{lr}} \\
\boldsymbol{w}_{\mathrm{lr}}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) - b_{\mathrm{lr}} &= 0 \qquad (16)
\end{aligned}$$

The threshold b_{lr} is an offset or bias to the hyperplane in likelihood-ratio score-space. It may be trained or pre-determined and encodes the required confidence in the class decision. A sensible choice for b_{lr} is the ratio of class priors, $b_{\mathrm{lr}} = -\ln P(\omega_1) + \ln P(\omega_2)$. In this case, the optimal likelihood-ratio threshold classifier becomes an optimal MAP decoder as described by Equation 8. Given a MAP decoder, it is useful to obtain a measure of confidence in the class decision. One such measure is the *posterior margin*. The true posterior margin is the log ratio between the true class posteriors. This can be recovered from the likelihood-ratio score-space. For example, for an utterance $\boldsymbol{O}$, the true posterior margin $\gamma_{\mathrm{pos}}(\boldsymbol{O})$ is the perpendicular distance from the decision hyperplane to the score $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$ in the infinite-dimensional score-space,

$$\begin{aligned}
\gamma_{\mathrm{pos}}(\boldsymbol{O}) &= \ln P(\omega_1|\boldsymbol{O}) - \ln P(\omega_2|\boldsymbol{O}) \\
&= \boldsymbol{w}_{\mathrm{lr}}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) - b_{\mathrm{lr}} \qquad (17)
\end{aligned}$$

In this case the true posterior margin is identical to the standard margin term used in SVM literature.

The present case where the correct functional forms of class-conditional generative models are known, is simply to introduce the reasoning behind derivative score-spaces. However, it is still valuable to consider the usefulness of score-spaces in this simple case. Given a quantity of training data in the form of labelled utterances, classifiers may be constructed in two different ways,

- The labelled utterances are used to train score-models which have the correct functional form. The outputs of the score-models are then used to make decisions in the standard minimum Bayes error framework.
- The labelled utterances are used to train score-models of the correct functional form. These score-models are then used to produce score-spaces. A classifier is then trained in a score-space and used to make decisions.

It is sensible to suppose that a classifier in score-space can give no better decision than a minimum Bayes error classifier constructed directly on the outputs of score-models. Furthermore, the classifier in score-space requires the estimation of an infinite number of parameters. A practical solution to this, as discussed later, is to truncate the score-space. With truncation, however, there is no longer any guarantee that the optimal classifier can be obtained even when there is sufficient training data. Truncated score-spaces are therefore expected to give some degradation in classification performance. Despite this, there are two possible scenarios where classifiers in score-spaces, even when truncated, may yield better performance than classifiers constructed directly on the model outputs.

- Score-models are trained according to some criterion. Training criteria are often multi-modal with local minima/maxima. As a result, the score-models trained may poorly match the true generative models, despite them having the same functional form. Since score-spaces use extra information in estimating their decision boundaries, it is possible that the resulting classifiers are more optimal. SVMs may be suitable training algorithms for linear classifiers in score-space since their cost functions are unimodal.
- If there is a lack of training data, then the parameters of the score-models may be poor estimates of the parameters of the true generative models. A simple classifier based on score-model outputs may be biased. However a linear classifier in score-space may be trained in a discriminative manner, for example through using SVMs. SVMs are expected to make better use of the limited data because of their regularisation properties.

In more realistic cases where the true generative models are much more complex than the estimated models, building classifiers in score-space may be even more advantageous (this is discussed in more detail in Section 3.3). However at this stage, it is important to highlight a potential weakness of classifiers constructed using score-spaces based on generative models. In order to maximise the usage from a set of training data, it is sensible to use the same training data to both estimate the generative models and to train the linear classifier in score-space. If this data is limited, then there is a risk of over-training, particularly if the score-space has many more dimensions than the number of parameters for the score-models. For this reason, it is imperative that the training algorithm for the linear classifier in score-space has proper regularisation, such as that provided by SVMs. However, in cases of few training utterances, it may be argued that greater performance gains could be achieved by using the data to discriminatively train score-models, rather than leave the discriminative training to the score-space. Although, the score-models would no longer be generative in nature, they should still yield useful score-spaces [20].

To return to considering classifiers defined in score-spaces; so far, the only derivative score-spaces described are those based on likelihood or likelihood-ratio score-arguments. However, the same approach can be used in deriving score-spaces based on posterior and posterior-ratio score-arguments. For utterance $\boldsymbol{O}$ and class ω_i, let $P(\omega_i|\boldsymbol{O})$ denote the *true class posterior* (i.e. the class posterior defined by the true class likelihoods), and let $\hat{P}(\omega_i|\boldsymbol{O})$ denote the *estimated class posterior* (i.e. the class posterior defined by the estimated class likelihoods). Using a Taylor series expansion, $\ln P(\omega_i|\boldsymbol{O})$ may be expressed in terms of a score in the posterior score-space for class ω_i. Derivatives must be taken with respect to the full set of parameter estimates $\hat{\boldsymbol{\xi}}$ since the class posterior is a function of the likelihoods for all classes. So then,

$$
\begin{aligned}
\ln P(\omega_i|\boldsymbol{O}) &= \ln P(\omega_i|\boldsymbol{O}) + (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^\top \nabla_{\hat{\boldsymbol{\xi}}} \ln P(\omega_i|\boldsymbol{O}) + \frac{1}{2}(\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^\top [\nabla^2_{\hat{\boldsymbol{\xi}}} \ln P(\omega_i|\boldsymbol{O})](\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}) \\
&\quad + O((\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^3) \\
&= \boldsymbol{w}^\top_{\mathrm{pos(i)}} \boldsymbol{\varphi}^{\mathrm{pos(i)}}(\boldsymbol{O}) + O((\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^3) \qquad (18)
\end{aligned}
$$

When these score-spaces are expanded to an infinite number of derivatives, linear classifiers trained in such score-spaces may be related to optimal posterior threshold classifiers (a detailed derivation of posterior score-spaces is given in Appendix B). For a two-class problem, where the sum of the two class posteriors is unity, the log ratio between true class posteriors is a useful measure of class discrimination. The score-space derived from this measure is called the posterior-ratio score-space. Since, in the present analysis, class priors are fixed and independent of the parameters of class-conditional generative models, the posterior-ratio score-space is closely related to the likelihood-ratio score-space. Consider the log posterior-ratio defined on true class posteriors, $\rho_{\mathrm{pos}}(\boldsymbol{O})$, and expand it using a Taylor series expansion about the parameter estimates $\hat{\boldsymbol{\xi}}$, then,

$$
\begin{aligned}
\rho_{\mathrm{pos}}(\boldsymbol{O}) &= \ln \frac{P(\omega_1|\boldsymbol{O})}{P(\omega_2|\boldsymbol{O})} \\
&= \boldsymbol{w}^\top_{\mathrm{pr}} \boldsymbol{\varphi}^{\mathrm{pr}}(\boldsymbol{O}) + O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^3, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^3) \qquad (19)
\end{aligned}
$$

The derivatives of the posterior-ratio score-space are identical to the derivatives of the likelihood-ratio score-space to an infinite order, with the exception of the zeroth-order derivative where $\hat{\rho}(\boldsymbol{O})$ is replaced by $\hat{\rho}_{\rm pos}(\boldsymbol{O})$. When the score-space is expanded to an infinite number of derivatives, a linear decision boundary may be trained in this score-space. If so, the perpendicular distance from the decision hyperplane to a score is again the posterior margin for the corresponding utterance. In fact, a linear classifier is sufficient to recover the optimal MAP decoder for the two-class problem. Therefore, for an utterance $\boldsymbol{O}$, the optimal MAP decision rule can be written in terms of either the likelihood-ratio score-space or posterior-ratio score-space as following,

$$\begin{aligned}\rho_{\rm pos}(\boldsymbol{O}) &= 0\\ \boldsymbol{w}_{\rm pr}^{\top}\boldsymbol{\varphi}^{\rm pr}(\boldsymbol{O}) &= 0\\ \boldsymbol{w}_{\rm lr}^{\top}\boldsymbol{\varphi}^{\rm lr}(\boldsymbol{O}) - b_{\rm lr} &= 0\end{aligned}$$

For the posterior-ratio score-space, the linear classifier should be unbiased so that it passes through the origin, while for the likelihood-ratio score-space the linear classifier should be allowed to have a bias. Both score-spaces should be defined on the same likelihood estimates, and both should be expanded to an infinite number of derivatives. The bias in the likelihood-ratio score-space is determined by the ratio of class priors. In the special case where the two class priors are equal, then $b_{\rm lr} = 0$ and $\hat{\rho}_{\rm pos}(\boldsymbol{O}) = \hat{\rho}(\boldsymbol{O})$. The posterior-ratio score-space $\boldsymbol{\varphi}^{\rm pr}(\boldsymbol{O})$ and likelihood-ratio score-space $\boldsymbol{\varphi}^{\rm lr}(\boldsymbol{O})$ are then identical. Various non-Bayesian approaches can be used to incorporate cost factors. If the class cost factors are independent of the generative model parameters, similarly to the class priors in this analysis, then they may simply be regarded as multiplying factors for the class priors.

Having considered how derivative score-spaces are used to recover optimal MAP decoders, some practical issues in using these score-spaces should be addressed. So far, in order to recover an optimal MAP decoder, score-spaces have been expanded to infinite orders of derivatives. There are two problems with this; (a) the analytical intractability and computational expense of deriving and calculating an infinite number of derivatives, and (b) a lack of sufficient data to estimate an infinite number of elements in the hyperplane weight vector. A practical solution to this is to use truncated score-spaces. However, various issues must be addressed to do this successfully.

- A score-space may be truncated, for example, after the first-order derivatives. However, if a linear classifier is then trained in a truncated score-space, the resulting classifiers can at best only approximate the optimal MAP decoder. Of course, this is under the assumption that the hyperplane weights and bias are well-estimated.
- If a truncated score-space is to be used to approximate a MAP decoder, then the approximation is expected to be better when the higher-order derivatives which are omitted diminish and are small in comparison to the low-order derivatives which are retained in the score-space. This is best achieved by ensuring that the zeroth-order derivative is a close approximation to the true value being estimated. This immediately suggests a dependency between the choice of training criterion for the score-models and the choice of score-space (i.e. the choice of score-argument).

For example, if the likelihood (1 class) or likelihood-ratio score-spaces are truncated after their first-order derivatives, then the class log likelihood estimates should be good approximations to the true class log likelihoods. A maximum likelihood training criterion may therefore be suitable for training the score-models. Similarly, if a truncated posterior score-space is used, then the class log posterior estimates should approximate the true class log posteriors. Posterior-based discriminative training criteria may be useful (e.g. Maximum Mutual Information Estimation (MMIE) [26] which seeks to maximise the sum of utterance log posteriors across the training set). However, extra factors may affect the choice of training criterion and score-argument. Besides matching the training criterion to the score-argument to achieve an accurate zeroth-order derivative estimate, a careful choice of training criterion may help promote class discrimination in the chosen score-space (see Section 5.1).

3.2 Estimated models as subsets of correct models in parameter space

Having considered how derivative score-spaces can be used to recover optimal MAP decoders, and having addressed some practical issues involved in using these score-spaces, it is important to consider whether these observations hold when the constraints between the estimated and true generative models are relaxed. For this reason, mismatches between the functional forms of the true and estimated generative models are considered, but with the significant constraint that the parameter space of each estimated model forms a subset of the parameter space of the corresponding true generative model. In this special case, SVMs are shown to be a sensible choice of classifier for derivative score-spaces. Although this section only addresses the likelihood-ratio score-space, similar analyses can be applied to the likelihood (1 class) and posterior-based score-spaces.

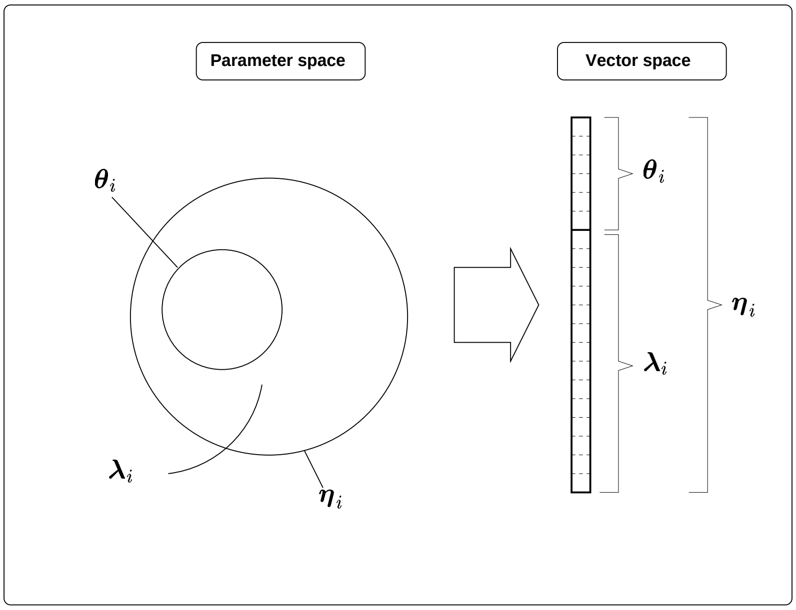


Figure 1: Relationship between the parameters of the true and estimated generative models

Consider that class ω_i has a true generative model defined by a parameter space $\boldsymbol{\eta}_i$. Consider also that a model estimate is proposed whose parameters correspond to a subset $\boldsymbol{\theta}_i$ in the space of $\boldsymbol{\eta}_i$ (see Figure 1). It is expected that the parameter space $\boldsymbol{\eta}_i$ is significantly larger than the parameter space $\boldsymbol{\theta}_i$ due to the expected greater complexity of the true generative model. The complimentary subset of $\boldsymbol{\theta}_i$ in $\boldsymbol{\eta}_i$ is $\boldsymbol{\lambda}_i$. The estimated model may be considered as having the same parameters as the true generative model, but with many parameters in the subspace corresponding to $\boldsymbol{\lambda}_i$ set to zero or other constant values. The parameter vectors for the true and estimated generative models may be respectively summarised by $\boldsymbol{\eta}_i$ and $\hat{\boldsymbol{\eta}}_i$, where $\tilde{\boldsymbol{\lambda}}_i$ is a vector of constants,

$$\begin{aligned}\boldsymbol{\eta}_i &= \begin{bmatrix}\boldsymbol{\theta}_i \\ \boldsymbol{\lambda}_i\end{bmatrix} \\ \hat{\boldsymbol{\eta}}_i &= \begin{bmatrix}\hat{\boldsymbol{\theta}}_i \\ \tilde{\boldsymbol{\lambda}}_i\end{bmatrix}\end{aligned}$$

For an utterance $\boldsymbol{O}$, the true class likelihood is $p(\boldsymbol{O}|\omega_i)$ and the estimated class likelihood is $\hat{p}(\boldsymbol{O}|\omega_i)$. The true likelihood may be written as $p_{\text{true(i)}}(\boldsymbol{O}|\boldsymbol{\eta}_i)$, and the estimated likelihood as $p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$. In fact, $p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$ is an approximation to $p_{\text{true(i)}}(\boldsymbol{O}|\boldsymbol{\eta}_i)$. This structure in parameter space permits an analysis similar to that in Section 3.1. First consider the true log likelihood $\ln p(\boldsymbol{O}|\omega_i)$ and express this in terms of a Taylor series expansion about the parameter estimate $\hat{\boldsymbol{\eta}}_i$ (in this expansion, the estimated likelihood $p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \equiv p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i)$ since $\tilde{\boldsymbol{\lambda}}_i$ is constant and may be discarded from the list of conditional dependencies). The parameters $\hat{\boldsymbol{\theta}}_i$ and $\tilde{\boldsymbol{\lambda}}_i$ are considered independent to an infinite order of derivative, so that there are no cross-terms in the second and higher-order derivatives). So then,

$$
\begin{aligned}
\ln p(\boldsymbol{O}|\omega_i) &= \ln p_{\mathrm{true(i)}}(\boldsymbol{O}|\boldsymbol{\eta}_i) \\
&= \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i) + (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^\top \nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i) + (\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^\top \nabla_{\tilde{\boldsymbol{\lambda}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i) \\
&\quad + \frac{1}{2}(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^\top [\nabla^2_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i)](\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) + \frac{1}{2}(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^\top [\nabla^2_{\tilde{\boldsymbol{\lambda}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\eta}}_i)](\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i) \\
&\quad + O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3) + O((\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^3) \\
&= \begin{bmatrix} 1 \\ (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) \\ \frac{1}{2}[(\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i) \otimes (\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)] \end{bmatrix}^\top \begin{bmatrix} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)] \end{bmatrix} + O((\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i)^3) \\
&\quad + \begin{bmatrix} 0 \\ (\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i) \\ \frac{1}{2}[(\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i) \otimes (\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)] \end{bmatrix}^\top \begin{bmatrix} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) \\ \mathrm{vec}[\nabla^2_{\tilde{\boldsymbol{\lambda}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)] \end{bmatrix} + O((\boldsymbol{\lambda}_i - \tilde{\boldsymbol{\lambda}}_i)^3) \qquad (20)
\end{aligned}
$$

The Taylor series has been expressed as a sum of two halves, one half dependent on the parameters $\boldsymbol{\theta}_i$ and their estimates $\hat{\boldsymbol{\theta}}_i$, and the other half dependent on the unknown parameters $\boldsymbol{\lambda}_i$ and their constant values $\tilde{\boldsymbol{\lambda}}_i$. The remainder may again be made arbitrarily small by expanding to to an arbitrarily high number of derivatives. A similar expansion can be made for the log likelihood-ratio between the two true class likelihoods. Consider two classes ω_1 and ω_2 whose true and estimated models are related by subset structures in parameter space as described in Figure 1 (though the true generative models for each class need not be related in any way). The true likelihoods are $p_{\mathrm{true(1)}}(\boldsymbol{O}|\boldsymbol{\eta}_1)$ and $p_{\mathrm{true(2)}}(\boldsymbol{O}|\boldsymbol{\eta}_2)$, and the corresponding likelihood estimates are $p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)$ and $p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)$. With the same independence assumptions between $\hat{\boldsymbol{\theta}}_1$ and $\tilde{\boldsymbol{\lambda}}_1$ and between $\hat{\boldsymbol{\theta}}_2$ and $\tilde{\boldsymbol{\lambda}}_2$, the log ratio between the true likelihoods can be expressed as follows,

$$
\begin{aligned}
\rho(\boldsymbol{O}) &= \ln \frac{p_{\mathrm{true(1)}}(\boldsymbol{O}|\boldsymbol{\eta}_1)}{p_{\mathrm{true(2)}}(\boldsymbol{O}|\boldsymbol{\eta}_2)} \\
&= \boldsymbol{u}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) + \boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\mathrm{lr}}(\boldsymbol{O}) + O((\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^3, (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)^3) + O((\boldsymbol{\lambda}_1 - \tilde{\boldsymbol{\lambda}}_1)^3, (\boldsymbol{\lambda}_2 - \tilde{\boldsymbol{\lambda}}_2)^3) \qquad (21)
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{u} &= \begin{bmatrix} 1 \\ (\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1) \\ \frac{1}{2}[(\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1) \otimes (\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)] \\ (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2) \\ \frac{1}{2}[(\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2) \otimes (\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)] \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} 0 \\ (\boldsymbol{\lambda}_1 - \hat{\boldsymbol{\lambda}}_1) \\ \frac{1}{2}[(\boldsymbol{\lambda}_1 - \hat{\boldsymbol{\lambda}}_1) \otimes (\boldsymbol{\lambda}_1 - \hat{\boldsymbol{\lambda}}_1)] \\ (\boldsymbol{\lambda}_2 - \hat{\boldsymbol{\lambda}}_2) \\ \frac{1}{2}[(\boldsymbol{\lambda}_2 - \hat{\boldsymbol{\lambda}}_2) \otimes (\boldsymbol{\lambda}_2 - \hat{\boldsymbol{\lambda}}_2)] \end{bmatrix} \\
\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) &= \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_1} \hat{\rho}(\boldsymbol{O})] \\ \nabla_{\hat{\boldsymbol{\theta}}_2} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_2} \hat{\rho}(\boldsymbol{O})] \end{bmatrix}, \tilde{\boldsymbol{\varphi}}^{\mathrm{lr}}(\boldsymbol{O}) = \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\lambda}}_1} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\lambda}}_1} \hat{\rho}(\boldsymbol{O})] \\ \nabla_{\hat{\boldsymbol{\lambda}}_2} \hat{\rho}(\boldsymbol{O}) \\ \mathrm{vec}[\nabla^2_{\hat{\boldsymbol{\lambda}}_2} \hat{\rho}(\boldsymbol{O})] \end{bmatrix}
\end{aligned}
$$

The remainder term can again be made arbitrarily small by expanding to an arbitrarily high number of derivatives n. Neglecting the remainder, the true log likelihood-ratio can simply be expressed as a sum of two terms. The first term is a simple inner product between a score $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$ in a score-subspace and the weight vector $\boldsymbol{u}$ of a hyperplane in that subspace. This score can be evaluated since the score-subspace is defined by $\hat{\rho}(\boldsymbol{O})$ and its derivatives with respect to the parameter estimates $(\hat{\boldsymbol{\theta}}_1^\top, \hat{\boldsymbol{\theta}}_2^\top)^\top$. The second term is a simple inner product between a score $\tilde{\boldsymbol{\varphi}}^{\mathrm{lr}}(\boldsymbol{O})$ in a second score-subspace, and the weight vector $\boldsymbol{v}$ of a hyperplane in that subspace. This score cannot be evaluated since the score-subspace is defined by $\hat{\rho}(\boldsymbol{O})$ and its derivatives with respect to the unknown constants $(\tilde{\boldsymbol{\lambda}}_1^\top, \tilde{\boldsymbol{\lambda}}_2^\top)^\top$. As $n \to \infty$, the score-subspaces become infinite in dimension, the remainder terms diminish, and a better approximation to the log ratio between the true class likelihoods is permitted. In this limit, an optimal MAP decoder can be recovered by training a linear classifier in the full likelihood-ratio score-space (i.e the likelihood-ratio score-space formed by concatenating the two score-subspaces $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$ and $\tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O})$). The decision boundary is given by $\rho_{\text{pos}}(\boldsymbol{O}) = 0$, where,

$$\begin{aligned}
\rho_{\text{pos}}(\boldsymbol{O}) &= \ln P(\omega_1|\boldsymbol{O}) - \ln P(\omega_2|\boldsymbol{O}) \\
&= \ln\big(p_{\text{true}(1)}(\boldsymbol{O}|\boldsymbol{\eta}_1)P(\omega_1)\big) - \ln\big(p_{\text{true}(2)}(\boldsymbol{O}|\boldsymbol{\eta}_2)P(\omega_2)\big) \\
&= \rho(\boldsymbol{O}) + \ln P(\omega_1) - \ln P(\omega_2) \\
&= \boldsymbol{u}^\top \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O}) + \boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O}) - b_{\text{lr}} \\
&= [\boldsymbol{u}^\top, \boldsymbol{v}^\top] \begin{bmatrix} \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O}) \\ \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O}) \end{bmatrix} - b_{\text{lr}}
\end{aligned} \tag{22}$$

and $b_{\text{lr}} = -\ln P(\omega_1) + \ln P(\omega_2)$. However, even if it was feasible to form the score-space $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$ using an infinite number of derivatives, and even if there was an infinite amount of training data for correct estimation of $\boldsymbol{u}$, there would be no guarantee that the optimal decision boundary could be obtained. This is because of the effect of the unknown term $\boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O})$. This scenario is closer to real-world examples, rather than the constrained scenario of Section 3.1. It is therefore valuable to consider the conditions under which an optimal MAP decoder can be recovered in this more realistic case. Inspecting Equation 22 and letting $y_i = +1$ for class ω_1 and $y_i = -1$ for class ω_2, a necessary condition for recovering the optimal MAP decoder is,

$$y_i(\boldsymbol{u}^\top \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O}) - b_{\text{lr}}) > -y_i(\boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O})) \tag{23}$$

In effect, this states that the contribution of $\boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}(\boldsymbol{O})$ to the log posterior ratio must not force a change in the choice of a MAP decoder which bases its decision solely on $\boldsymbol{u}^\top \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$. Alternatively, the margin obtained from considering $\boldsymbol{u}^\top \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O}) - b_{\text{lr}}$ must be large enough so that any subtraction of $\boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O})$ does not change the sign of the margin and hence the choice of the MAP decoder. If these conditions are fulfilled for all possible utterances, then a learning algorithm for $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$ may be able to recover the optimal decoder. Although any algorithm may be used to train the linear decision boundary or hyperplane, an SVM is a particularly suitable choice of algorithm for the following reasons,

- If the training scores are indeed linearly separable in the known score-subspace $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$, then the natural choice of decision boundary in this subspace is one that lies midway between the two competing classes. Intuitively, this boundary has the greatest chance of fulfilling the necessary condition stated in Equation 23. An SVM is a learning algorithm which seeks to place the boundary in such a position. This implies that 'hard margin' SVMs are a natural choice for learning the decision boundary in this case.
- In practice, the training scores are usually linearly inseparable in the subspace $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$. The overlap between classes may be due to the close proximity of the true generative models in the input space and the stochastic nature of the generation processes. It may also be due to the oversimplified complexity of the estimated models and significant effect of the unknown parameters. The existence of various noise sources also causes overlap. Two different forms of SVM can be used,
  - A linear SVM may still be used to select a suitable decision boundary in the score-subspace $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$. However, the 'soft margin' SVM formulation must be used which trades-off a minimisation of training errors with regularisation which seeks to maximise the 'margin of separability' between the two competing classes.
  - Alternatively, an SVM with non-linear kernel can be trained in the known score-subspace, $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$. The kernel may give the SVM sufficient power to learn the patterns relating $\boldsymbol{u}^\top \boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$ with $\boldsymbol{v}^\top \tilde{\boldsymbol{\varphi}}^{\text{lr}}(\boldsymbol{O})$ and the class label y_i, so that it is possible to recover a better approximation to the optimal MAP decoder. In practice, there must be a sufficiently large quantity of training scores and a sufficiently powerful kernel to learn these patterns successfully. If there is a lack of training scores, then an SVM with linear kernel may prove to be more robust.

Some further points must be noted if an SVM is used to recover the decision boundary in the known score-subspace. First, for an SVM, the quality of the decision boundary is affected by the quantity of data available to estimate the weight vector $\boldsymbol{u}$ and the bias term b_{lr}[6]. Even if a decision boundary exists which fulfils the conditions in Equation 23, it may not be possible to recover the boundary if there is insufficient or poor quality data. For example, an SVM is constrained to position its support vectors on existing training scores, the locations of which may be suboptimal. Second, the SVM margin in the known score-subspace corresponds

[6]Note that the value of b_{lr} estimated for the known score-subspace is not necessarily the same as that which would be required for the full score-space.

to $(\boldsymbol{u}^\top \boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) - b_{\mathrm{lr}})$. Hence, the SVM margin is a rough approximation to the posterior margin. However, it must be used cautiously as such because of the unknown effect of the unknown parameters.

As detailed in Section 3.1, good approximations to optimal MAP decoders may be possible using truncated score-spaces when the omitted derivatives vanish and are small in comparison to those retained. In practice, this means that the zeroth-order derivative must provide a good estimate to the true value being estimated. However in this case, even if the higher-order derivative terms in $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$ diminish and are small in comparison to their low-order derivative terms, there is no guarantee that the high-order derivative terms in $\tilde{\boldsymbol{\varphi}}^{\mathrm{lr}}(\boldsymbol{O})$ are likewise diminished in comparison to their low-order derivative terms. For this reason, it is difficult to check whether truncated score-spaces in this case are able to yield good approximations to optimal MAP decoders.

3.3 True and estimated models of different complexity

So far, two special cases have been considered. The first constrained the true and estimated models to have the same functional form. The second relaxed this constraint but still imposed a structure on their parameter spaces. These two cases served to illustrate the conditions under which optimal MAP decoders could be recovered, and why SVMs were a good choice of classifier in derivative score-spaces. However, for real-world problems the constraints between true and estimated generative models need to be relaxed even further. In real problems, the true models are significantly more complex than the models proposed for them. The frameworks considered in Sections 3.1 and 3.2 are too simple. The significant differences in complexity, or severe model incorrectness for the estimates, limit the classification performance of derivative score-spaces. Even if score-spaces were formed from an infinite number of derivatives, it would be unlikely that linear classifiers drawn in such score-spaces would recover optimal MAP decoders. Despite this, such score-spaces may still enable better approximations to optimal MAP decoders, particularly if non-linear decision boundaries are drawn in the score-spaces. For the classification of speech patterns, the true speech models may be considered infinite-order Markov processes, but the estimated models are usually much simpler first-order Markov processes. It is expected that this model incorrectness, in particular the conditional independence assumptions, prohibits the recovery of optimal MAP decoders using linear classifiers in derivative score-spaces for speech recognition. The self-same model incorrectness assumptions are in fact responsible for the degradation in performance of all classifiers based on standard HMMs in speech recognition.

When model incorrectness and a truncation of score-spaces diminish any hope of recovering optimal MAP decoders, score-spaces are still useful. They may simply be viewed as sets of features which contain useful information relating utterances to their model estimates. For example, first-order derivatives with respect to model parameters give information as to parameter sensitivity [10]. Classifiers which are trained in these score-spaces should still give informative decisions. In these cases, the use of generatively-trained models may be abandoned for discriminatively-trained models. Discriminatively-trained score-models make no assumptions about model correctness, and the class discriminative information contained in their score-spaces may be more useful than, or may at least be partly complimentary to, the class discriminative information contained in score-spaces defined on generatively-trained score-models.

4 Defining score-spaces

4.1 Derivative score-spaces

To use score-spaces, suitable choices must be made for the score-models, score-arguments, and score-operators. First, consider the score-models. Many state-of-the-art speech recognition systems use HMMs with continuous density output distributions modelled by Gaussian Mixture Models (GMMs). These HMMs are often referred to as *acoustic models*. In this report, left-to-right HMMs with no skips are used as score-models. Since the score-spaces used in the experiments in this report are only defined on two-class problems, the score-models for each two-class problem may be summarised as following; $p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)$ and $p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)$ are the respective likelihood estimates for classes ω_1 and ω_2, $p_g(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_g)$ is the likelihood estimate produced by a generative model for both classes, and the posterior estimates for classes ω_1 and ω_2 are $\hat{P}(\omega_1|\boldsymbol{O})$ and $\hat{P}(\omega_2|\boldsymbol{O})$ respectively. To recap on some abbreviations, $\hat{\rho}(\boldsymbol{O}) = (\ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) - \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2))$, $\hat{\rho}_{\mathrm{pos}}(\boldsymbol{O}) = (\ln \hat{P}(\omega_1|\boldsymbol{O}) - \ln \hat{P}(\omega_2|\boldsymbol{O}))$, and $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\theta}}_1^\top, \hat{\boldsymbol{\theta}}_2^\top)^\top$. A suitable training criterion must also be chosen for the score-models. In this report, score-models are trained to maximise likelihood. Alternatively, score-models may be trained discriminatively [20].

For derivative score-spaces, the score-operators should be derivative operators. In the experiments in this report, only zeroth and first-order derivative operators are used so that the resulting score-spaces are heavily truncated versions of infinite-dimensional score-spaces. Higher-order derivatives have mainly been ignored to reduce analytical and computational complexity, but also under the assumption that the score-models are well-trained and higher-order derivatives are small and diminish in comparison to lower-order derivatives. For two-class problems, various score-arguments are used. These are the log likelihood for a two-class model, the log likelihood for a single class, the log likelihood-ratio between two competing classes, the log posterior for a single class, and the log posterior-ratio between two competing classes. When these score-arguments are combined with derivative score-operators, the resulting score-spaces are respectively termed the likelihood (2 class) score-space $\boldsymbol{\varphi}^{\mathrm{lk2}}(\boldsymbol{O})$ (and also the Fisher-score $\boldsymbol{\varphi}^{\mathrm{fish}}(\boldsymbol{O})$), the likelihood (1 class) score-space $\boldsymbol{\varphi}^{\mathrm{lk}}(\boldsymbol{O})$, the likelihood-ratio score-space $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$, the posterior score-space $\boldsymbol{\varphi}^{\mathrm{pos}}(\boldsymbol{O})$, and the posterior-ratio score-space $\boldsymbol{\varphi}^{\mathrm{pr}}(\boldsymbol{O})$. These score-spaces are defined as follows, where for brevity the likelihood (1 class) and posterior score-spaces are defined only for class ω_1, and so are labelled as $\boldsymbol{\varphi}^{\mathrm{lk}(1)}(\boldsymbol{O})$ and $\boldsymbol{\varphi}^{\mathrm{pos}(1)}(\boldsymbol{O})$ (fuller derivations for the posterior and posterior-ratio score-spaces are given in Appendix B),

$$\boldsymbol{\varphi}^{\mathrm{fish}}(\boldsymbol{O}) = \nabla_{\hat{\boldsymbol{\theta}}_g} \ln p_g(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_g) \tag{24}$$

$$\boldsymbol{\varphi}^{\mathrm{lk2}}(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \nabla_{\hat{\boldsymbol{\theta}}_g} \end{bmatrix} \ln p_g(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_g) = \begin{bmatrix} \ln p_g(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_g) \\ \nabla_{\hat{\boldsymbol{\theta}}_g} \ln p_g(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_g) \end{bmatrix} \tag{25}$$

$$\boldsymbol{\varphi}^{\mathrm{lk}(1)}(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \end{bmatrix} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) = \begin{bmatrix} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \end{bmatrix} \tag{26}$$

$$\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \nabla_{\hat{\boldsymbol{\xi}}} \end{bmatrix} \hat{\rho}(\boldsymbol{O}) = \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1}\hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_2}\hat{\rho}(\boldsymbol{O}) \end{bmatrix} = \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \tag{27}$$

$$\boldsymbol{\varphi}^{\mathrm{pos}(1)}(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \nabla_{\hat{\boldsymbol{\xi}}} \end{bmatrix} \ln \hat{P}(\omega_1|\boldsymbol{O}) = \begin{bmatrix} \ln \hat{P}(\omega_1|\boldsymbol{O}) \\ \hat{P}(\omega_2|\boldsymbol{O})\nabla_{\hat{\boldsymbol{\xi}}}\hat{\rho}(\boldsymbol{O}) \end{bmatrix} = \begin{bmatrix} \ln \hat{P}(\omega_1|\boldsymbol{O}) \\ \hat{P}(\omega_2|\boldsymbol{O})\nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\hat{P}(\omega_2|\boldsymbol{O})\nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \tag{28}$$

$$\boldsymbol{\varphi}^{\mathrm{pr}}(\boldsymbol{O}) = \begin{bmatrix} 1 \\ \nabla_{\hat{\boldsymbol{\xi}}} \end{bmatrix} \hat{\rho}_{\mathrm{pos}}(\boldsymbol{O}) = \begin{bmatrix} \hat{\rho}_{\mathrm{pos}}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \tag{29}$$

Analytical expressions for these score-spaces may be formed from common building-blocks. The building-blocks are first-order derivatives of log probability models with respect to model parameters. The common building blocks can then be weighted and concatenated in different combinations to form the different score-spaces. The first-order derivatives with respect to model parameters are derived in [19] and are repeated below in summary format. For an HMM with N emitting states, let state $j \in \{1 \ldots N\}$ have an output distribution formed by a mixture of K Gaussian components, and let component $k \in \{1 \ldots K\}$ have a Gaussian distribution $\mathcal{N}_{[t,jk]}(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$ for observation $\boldsymbol{o}_t$, where,

$$\mathcal{N}_{[t,jk]}(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_{jk}|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})^\top \boldsymbol{\Sigma}_{jk}^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})\} \tag{30}$$

Each component has parameters of weight w_{jk}, mean $\boldsymbol{\mu}_{jk}$ and covariance $\boldsymbol{\Sigma}_{jk}$. For the data sequence $\boldsymbol{O}$ of duration T observations, the first-order derivatives of the log likelihood estimate with respect to these model parameters are,

$$\nabla_{\boldsymbol{\mu}_{jk}} \ln p(\boldsymbol{O}|\hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \gamma_{jk}(t) S_{[t,jk]}^\top \tag{31}$$

$$\nabla_{\mathrm{vec}(\boldsymbol{\Sigma}_{jk})} \ln p(\boldsymbol{O}|\hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \gamma_{jk}(t) \frac{1}{2}\left[-[\mathrm{vec}(\boldsymbol{\Sigma}_{jk}^{-1})]^\top + \{S_{[t,jk]} \otimes S_{[t,jk]}\}\right]^\top \tag{32}$$

$$\nabla_{w_{jk}} \ln p(\boldsymbol{O}|\hat{\boldsymbol{\theta}}) = \sum_{t=1, \gamma_j(t)\neq 0}^{T} \gamma_{jk}(t)\left[\frac{1}{w_{jk}} - \frac{\gamma_{j1}(t)}{w_{j1}\gamma_{jk}(t)}\right] \tag{33}$$

$$\text{where} \qquad S_{[t,jk]} = (\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})^\top \boldsymbol{\Sigma}_{jk}^{-1} \tag{34}$$

and $\gamma_{jk}(t)$ is the component posterior probability,

$$\begin{aligned}\gamma_{jk}(t) &= p\Big(m(t) = jk, s(t) = j|\boldsymbol{O}, \hat{\boldsymbol{\theta}}\Big) \\ &= \gamma_j(t) \times \frac{w_{jk}\mathcal{N}_{[t,jk]}(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})}{\sum_{i=1}^{K} w_{ji}\mathcal{N}_{[t,ji]}(\boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji})}\end{aligned} \tag{35}$$

$\gamma_j(t)$ is the state posterior probability (i.e. the posterior probability for observation $\boldsymbol{o}_t$ being generated by state j) and is typically supplied by an external system such as an HMM recogniser. Assuming the HMM is left-to-right with no skips and assuming that a state only appears once in the HMM (i.e. there is no state-tying along the HMM), then the 1st-order derivative for the self-transition probability for state j, a_{jj}, is,

$$\nabla_{a_{jj}} \ln p(\boldsymbol{O}|\hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T}\Big[\frac{\gamma_j(t)}{a_{jj}} - \frac{1}{Ta_{jj}(1-a_{jj})}\Big] \tag{36}$$

The first-order derivatives for the parameters of each Gaussian component in the HMM (Equations 31 to 33) and each self-transition probability (Equation 36) can be spliced together, and weighted if necessary, to form 'super-vectors'. These super-vectors are *scores*. Due to the sum to unity constraints, one of the weight parameters in each Gaussian mixture should be discarded from the definition of the super-vector or score-space, as should the forward transitions in the left-to-right HMM.

Section 3 implies improved classification performance using second-order derivatives. The introduction of second-order derivatives requires a large increase in the size of the score-space and an increase in computational requirements. However as an example, the second-order derivative of the log probability model with respect to a component mean, $\boldsymbol{\mu}_{jk}$, is defined as following,

$$\nabla^2_{\boldsymbol{\mu}_{jk}} \ln p(\boldsymbol{O}|\hat{\boldsymbol{\theta}}) \approx -\sum_{t=1}^{T} \gamma_{jk}(t)\boldsymbol{\Sigma}_{jk}^{-1} \tag{37}$$

The component posterior probability $\gamma_{jk}(t)$ has been assumed independent of $\boldsymbol{\mu}_{jk}$. This assumption is invalid and is only made for analytical simplicity and to give a rough indication of the performance of second-order derivatives. Valid second-order derivatives should be derived for future work.

4.2 Sequence length normalisation

Two different observation sequences with similar content but at different rates of variation (e.g. in speech recognition, two utterances spoken identically but at different speaking rates), project into two different regions of score-space. This is because the score for an observation sequence is the sum of scores for the individual observations (see Equations 31 to 33 and 36). While a non-linear classifier may be powerful enough to classify both patterns as belonging to the same class, the task of the classifier is eased by normalising this variation. This is called *sequence length normalisation*. One approach is to train generative models in a normalised manner. Rather than define score-spaces using generative models trained to maximise the standard log likelihoods $l(\hat{\boldsymbol{\theta}})$, score-spaces may be defined using generative models trained to maximise normalised log likelihoods $l_n(\hat{\boldsymbol{\theta}})$. For a particular utterance and particular generative model estimate, the normalised log likelihood is produced by dividing the log likelihood contribution from each state by the total occupancy for that state. Each state therefore contributes, in effect, one observation to the final normalised log likelihood. As the utterance length varies, the normalised log likelihood score should retain the same dynamic range. To explain, consider R utterances which are attributed to a class modelled by an HMM with N states. For each utterance $\boldsymbol{O}^r = (\boldsymbol{o}_1^r, \boldsymbol{o}_2^r \ldots \boldsymbol{o}_t^r \ldots \boldsymbol{o}_{T_r}^r), r \in \{1 \ldots R\}$, the recognition path enters state j at time τ_j^r and remains in the state for duration d_j^r (where time is measured in numbers of observations). The output probability in state j for observation $\boldsymbol{o}_t^r$ is $b_j(\boldsymbol{o}_t^r)$. So then,

$$l(\hat{\boldsymbol{\theta}}) = \sum_{r=1}^{R}\sum_{j=1}^{N}\Big(\sum_{t=\tau_j^r}^{\tau_j^r+d_j^r-1}\Big(\ln b_j(\boldsymbol{o}_t^r)\Big) + (d_j^r - 1)\ln a_{jj} + \ln a_{j(j+1)}\Big) \tag{38}$$

$$l_n(\hat{\boldsymbol{\theta}}) = \sum_{r=1}^{R}\sum_{j=1}^{N}\frac{1}{d_j^r}\Big(\sum_{t=\tau_j^r}^{\tau_j^r+d_j^r-1}\Big(\ln b_j(\boldsymbol{o}_t^r)\Big) + (d_j^r - 1)\ln a_{jj} + \ln a_{j(j+1)}\Big) \tag{39}$$

It is not possible to maximise $l_n(\hat{\boldsymbol{\theta}})$ using the EM algorithm. Hill-climbing techniques could be used but usually have a high computational cost. In this report, a simpler method of normalisation is used. Generative models are still trained to maximise standard log likelihoods $l(\hat{\boldsymbol{\theta}})$. However, rather than define each score-space using the standard state posteriors $\gamma_j(t)$, normalised state posteriors $\gamma'_j(t)$ are used, where each state posterior is normalised by the total state occupancy over the utterance,

$$\gamma'_j(t) = \frac{\gamma_j(t)}{\sum_{\tau=1}^{T} \gamma_j(\tau)} \tag{40}$$

For normalisation, the total state occupancy is preferred to the total number of observations T as proposed in [19], because the latter assumes that the duration of each state in the state sequence is lengthened or shortened by the same ratio, which is not necessarily the case. Normalisation by the total state occupancy allows different portions of the state sequence to be lengthened or shortened by different ratios. The normalised component posteriors, $\gamma'_{jk}(t)$, are given by,

$$\gamma'_{jk}(t) \quad = \quad \gamma'_j(t) \times \frac{w_{jk}\mathcal{N}_{[t,jk]}(\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})}{\sum_{i=1}^{K} w_{ji}\mathcal{N}_{[t,ji]}(\boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji})} \tag{41}$$

The normalised component posteriors $\gamma'_{jk}(t)$ should be substituted for $\gamma_{jk}(t)$ in the definitions of the derivatives in Equations 31 to 33 and 36.

4.3 Score-space normalisation

Having defined different score-spaces, it is necessary to consider the appropriate metric tensor for classifiers constructed in score-space. Many classifiers, including SVMs, use an inner product term during training and testing. As the different dimensions of score-space are non-orthogonal, the inner product term should be defined using the covariance matrix in score-space $\boldsymbol{\Sigma}_{\text{sc}}$, and the expectation in score-space $\boldsymbol{\mu}_{\text{sc}}$. In order to employ standard classifiers in score-space such as SVMs, the score-space should first be normalised before classifiers are trained and tested. This is called *score-space normalisation* and is a form of 'data whitening'. It is expressed by the following transformation from $\boldsymbol{\varphi}(\boldsymbol{O})$ to $\boldsymbol{\varphi}_N(\boldsymbol{O})$,

$$\boldsymbol{\varphi}_N(\boldsymbol{O}) \quad = \quad \boldsymbol{\Sigma}_{\text{sc}}^{-1/2}(\boldsymbol{\varphi}(\boldsymbol{O}) - \boldsymbol{\mu}_{\text{sc}}) \tag{42}$$

The score-space is shifted to have zero mean prior to transformation by $\boldsymbol{\Sigma}_{\text{sc}}^{-1/2}$. However, if an SVM trains a decision boundary in the resulting score-space, the zero-mean shift can be ignored without change in solution if the SVM uses a translational invariant kernel such as a Gaussian Radial Basis Function (GRBF) kernel. The zero-mean shift can also be ignored if an SVM uses a linear kernel though the kernel is not translational invariant, since the hyperplane bias encodes the zero-mean shift. For SVMs with polynomial kernels in score-space, subtraction of $\boldsymbol{\mu}_{\text{sc}}$ is required. Since the experiments in this report use SVMs with linear kernels, a simpler form of data whitening is employed,

$$\boldsymbol{\varphi}_N(\boldsymbol{O}) \quad = \quad \boldsymbol{\Sigma}_{\text{sc}}^{-1/2}\boldsymbol{\varphi}(\boldsymbol{O}) \tag{43}$$

If an SVM is trained in $\boldsymbol{\varphi}_N(\boldsymbol{O})$ using a standard linear kernel, as for the experiments in this report, then the SVM can be viewed as operating directly in the input-space of observation sequences. In this case, the SVM kernel is called the *Normalised kernel* [19] to distinguish it from the Fisher kernel. For two observation sequences $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$, the Normalised kernel is $k_{\text{N}}(\boldsymbol{O}_i, \boldsymbol{O}_j)$ where,

$$k_{\text{N}}(\boldsymbol{O}_i, \boldsymbol{O}_j) \quad = \quad \boldsymbol{\varphi}(\boldsymbol{O}_i)^{\top}\boldsymbol{\Sigma}_{\text{sc}}^{-1}\boldsymbol{\varphi}(\boldsymbol{O}_j) \tag{44}$$

Due to the naturally occurring overlap between classes, model incorrectness for score-models, and sequence length normalisation, a standard linear kernel operating in the normalised score-space may be insufficient to separate competing classes. Better classifiers may be yielded by training non-linear classifiers to learn the patterns relating scores to their labels in the normalised score-space.

Since the true $\boldsymbol{\Sigma}_{\text{sc}}$ is only exactly determined when the underlying data distribution is known, various approximations to this matrix must be made. The approximations may provide robustness, reduce computational load, or be applicable under certain conditions. In [19], a *scaling approximation* was shown to be an acceptable substitution for full whitening while avoiding the need for inverting large matrices (though [19] was for classification of single observations rather than sequences, on a different database). The experiments in this report use such a scaling approximation, replacing $\mathbf{\Sigma}_{\rm sc}^{-1/2}$ by its empirically estimated diagonal form. Failure to perform score-space normalisation for a linear kernel in score-space results in a kernel similar to the Plain kernel [17]. This has been shown in [19] to give poor performance most probably due to the variation in dynamic range for different dimensions of unnormalised score-space. This variation in dynamic range may cause the classification boundary to be undesirably biased to dimensions which have larger dynamic ranges but which are less class discriminative. The Fisher kernel in [10] uses the Fisher Information matrix rather than the covariance matrix in score-space normalisation. This is only an acceptable normalisation when the expectation in the score-space is zero. This is true for the Fisher score if the score-models are trained to maximise likelihood, and the set of data used to train the score-models and to train and test the SVM are 'matched' in the sense that they are drawn from the same probability distribution.

As noted in Section 3.1, there are often many repeated elements in the vectors associated with the second and higher-order derivatives of score-space. It is useful to investigate whether the inclusion or omission of these repeated elements affect the solution of classifiers constructed in the normalised score-space $\boldsymbol{\varphi}_N(\boldsymbol{O})$. As an example of repeated elements, the likelihood-ratio score-space $\boldsymbol{\varphi}^{\rm lr}(\boldsymbol{O})$ has a second-order term, $[\nabla^2_{\hat{\boldsymbol{\theta}}_i}\rho(\boldsymbol{O})]$. If the dimensionality of $\boldsymbol{\theta}_i$ is d, then since this second-order term is a symmetric matrix, the dimensionality of its vectorised form ${\rm vec}[\nabla^2_{\hat{\boldsymbol{\theta}}_i}\rho(\boldsymbol{O})]$ may be reduced without loss of information from d^2 to $d(d+1)/2$. This in turn reduces the dimensionality of the full score-space $\boldsymbol{\varphi}^{\rm lr}(\boldsymbol{O})$ by $d(d-1)/2$. These $d(d-1)/2$ elements are examples of what are called *repeated elements*. It is clear that the inclusion or omission of repeated elements does not alter the information content in score-space. Also, if full whitening is implemented, the inclusion or omission of repeated elements does not alter the form of the normalised score-space and hence of the solution returned by a classifier trained in the normalised score-space. However, this is not true for some approximations to full whitening. These two cases are explained below.

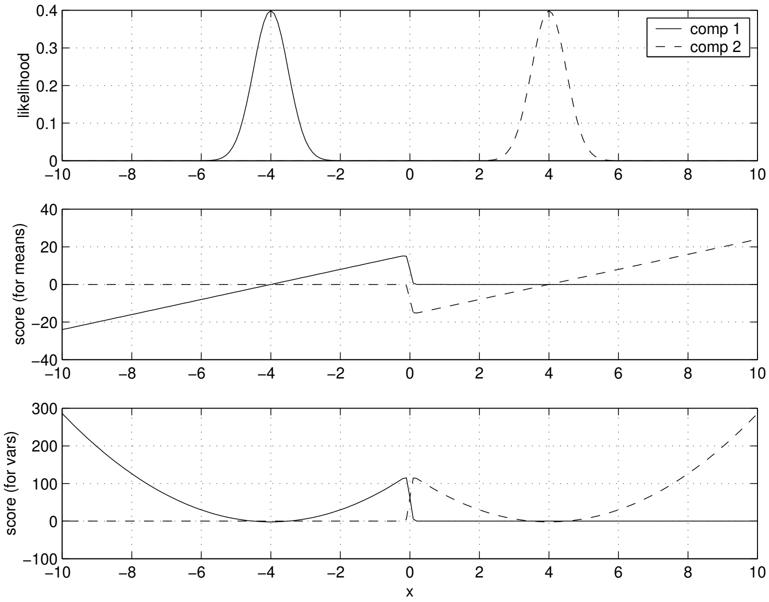
- Full whitening: First, if repeated elements are included in the definition of the score-space $\boldsymbol{\varphi}(\boldsymbol{O})$, then the true covariance matrix $\mathbf{\Sigma}_{\rm sc}$ is not of full rank. This implies that an inverse $\mathbf{\Sigma}_{\rm sc}^{-1}$ does not exist [8]. However if Singular Value Decomposition of $\mathbf{\Sigma}_{\rm sc}$ is performed and output dimensions with zero singular values are discarded, a whitening matrix may be obtained. The range-space of this non-square whitening matrix is the normalised score-space $\boldsymbol{\varphi}_N(\boldsymbol{O})$. Second, if repeated elements are omitted from the definition of the score-space $\boldsymbol{\varphi}(\boldsymbol{O})$, and numerical problems are ignored, then if the same process is repeated as above, a whitening matrix with the same range-space is obtained. The normalised score-space $\boldsymbol{\varphi}_N(\boldsymbol{O})$ is therefore invariant to the inclusion or omission of repeated elements. The solution of a classifier constructed in $\boldsymbol{\varphi}_N(\boldsymbol{O})$ is therefore also invariant to the inclusion or omission of repeated elements.
- Scaling approximation: In scaling, $\mathbf{\Sigma}_{\rm sc}$ is replaced by an empirically calculated diagonal approximation. The range-space of this diagonal matrix is the normalised score-space $\boldsymbol{\varphi}_N(\boldsymbol{O})$. Since a diagonal matrix has full rank and its range-space varies with the size of its input-space, the size and form of $\boldsymbol{\varphi}_N(\boldsymbol{O})$ vary with the inclusion or omission of repeated elements. This in turn may affect the solution returned by a classifier trained in $\boldsymbol{\varphi}_N(\boldsymbol{O})$. For example, if an SVM is used to train a decision boundary in $\boldsymbol{\varphi}_N(\boldsymbol{O})$, then the errors in each dimension of $\boldsymbol{\varphi}_N(\boldsymbol{O})$ are weighted and used to position the boundary. If repeated elements are included in $\boldsymbol{\varphi}(\boldsymbol{O})$ and hence in $\boldsymbol{\varphi}_N(\boldsymbol{O})$, the errors associated with these repeated elements are effectively given twice the weighting they would possess if the repeated elements were omitted. This difference in weighting may well alter the position of a decision boundary trained by an SVM in $\boldsymbol{\varphi}_N(\boldsymbol{O})$.

By similar arguments to 'full whitening', if any full matrix approximation to $\mathbf{\Sigma}_{\rm sc}$ is used for whitening, then a classifier trained in $\boldsymbol{\varphi}_N(\boldsymbol{O})$ is also expected to be invariant to the inclusion or omission of repeated elements. If there is no score-space normalisation, then this is equivalent to substituting the Identity matrix for $\mathbf{\Sigma}_{\rm sc}^{-1/2}$. By the same arguments as for the 'scaling approximation', the solution returned by a classifier in normalised score-space is expected to vary with the inclusion or omission of repeated elements.

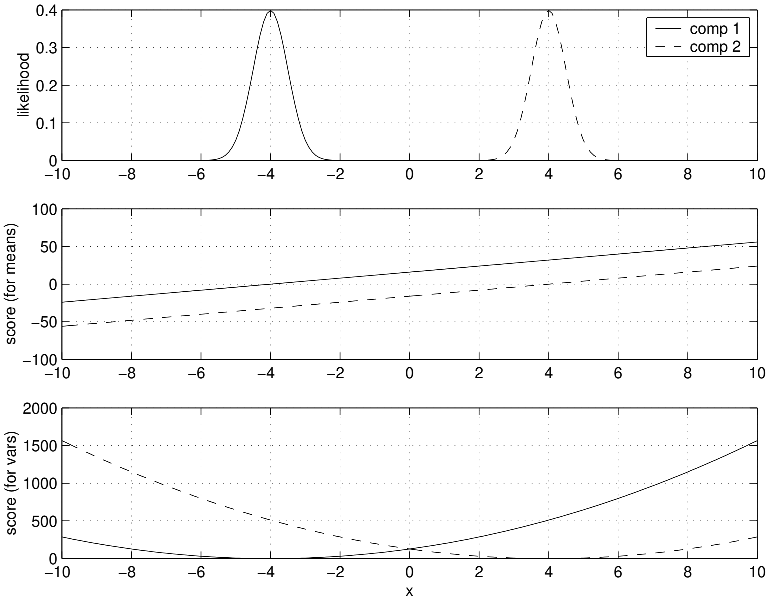
5 Class discrimination in score-space

5.1 Comparing discrimination in different score-spaces

The interaction between the score-argument and the training criterion for the score-models may be used to encourage class discrimination in the score-space. For example, consider the truncated likelihood (2 class)



(a) single generative model with two components (subspace in $\boldsymbol{\varphi}^{\mathrm{lk2}}(\boldsymbol{O})$)



(b) two class-conditional generative models (subspace in $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$)

Figure 2: Illustrating 'wrap-around'

and likelihood-ratio score-spaces defined by zeroth and first-order derivatives. First, compare the subspaces defined by the zeroth-order derivative operators. The likelihood (2 class) is expected to provide poorer class discrimination than the likelihood-ratio since it is more susceptible to variations in speaker and acoustic conditions. The likelihood-ratio normalises some of these factors.

Next, compare the subspaces defined by the first-order derivative operators. The subspace for the likelihood (2 class) score-space is expected to hinder class discrimination due to an effect which will be referred to as *wrap-around*. Wrap-around occurs when the acoustic space 'wraps' onto a score-subspace so that two distinct points in acoustic space map to a single point in the score-subspace. This can degrade class discrimination in the score-subspace. To illustrate this, consider the simple problem of a 1-dimensional input space populated by two competing classes of data. The two classes ω_1 and ω_2 are both modelled by single Gaussian components $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ with $\mu_1 = -4.0$, $\mu_2 = +4.0$ and $\sigma_1 = \sigma_2 = 0.5$ (the small overlap between these two Gaussians simulates the small overlap between the acoustic models in the 39-dimensional acoustic space used in this report). The two Gaussians may be linearly combined with equal weights to form a mixture score-model for the two classes (see Figure 2(a)). Alternatively they may be kept as two distinct class-conditional score-models (see Figure 2(b)). Both sets of score-models project into four-dimensional subspaces of first-order derivatives defined by the two component means and two component variances (the score-subspace for the mixture model should have an additional dimension due to the weight term, but this extra dimension is ignored in the current explanation). The mixture model projects into a likelihood (2 class) score-subspace, and the two class-conditional models project into a likelihood-ratio score-subspace.

The first plot in Figure 2(a) shows the positions of the two Gaussian components in the 1-dimensional acoustic space. The second plot shows the derivatives with respect to the means for these two components (the unbroken line is the derivative with respect to μ_1, the broken line is the derivative with respect to μ_2). The third plot shows the derivatives with respect to σ_1 and σ_2. Each derivative is in fact the product of a gradient term and the component posterior (see Equations 31 and 32). As the two component posteriors are sigmoidal pairs and tend to zero with distance from their relevant components, the resulting derivatives for the component means and variances also tend to zero with distance from their relevant components. For this reason the derivatives for each of the four dimensions are zero for virtually half of acoustic space. For the derivatives of the means, there are two distinct positions in acoustic space which map to zero score-subspace, these being located at the peaks of the two Gaussians. A similar observation is noted for the derivatives of the variances. Therefore if two observations belonging to different classes are 'generated' at the peaks of their respective Gaussians, then the two observations map to the same point in the four-dimensional score-subspace. It is then impossible to discriminate between these two observations through observing their scores in this subspace alone. The acoustic space *wraps* round onto the ambiguous zero in score-subspace. More generally, observations modelled by the two Gaussians but belonging to the two different classes would show significant confusion in the four-dimensional score-subspace. For this reason, the likelihood (2 class) score-subspace defined by the first-order derivatives is not expected to be very discriminative. This score-subspace is also known as the Fisher score.

However, if the corresponding graphs for the two distinct class-conditional models are considered in Figure 2(b), then the plots for the derivatives are not forced to zero. This is because the component posteriors for observations are forced to unity irrespective of the position of the observations in acoustic space. The derivatives are in fact the simple gradient plots (see Equations 31 and 32). There is therefore good class discrimination in the four-dimensional score-subspace, and wrap-around does not occur[7]. Therefore, for the score-subspaces defined by first-order derivatives, the likelihood-ratio score-subspace is expected to be more discriminative than the likelihood (2 class) score-subspace.

There is however a special case when the likelihood (2 class) score-subspace defined by first-order derivatives, as formulated above, gives fair discrimination. This is when each of the two classes represented by the single Gaussians in Figure 2(a) has a distinct bi-modal distribution which straddles the peak of its Gaussian. There are then very few observations which actually sit at the Gaussian peaks and which project into the ambiguous zero point in score-subspace. The score-subspace therefore still allows some degree of separation between the two classes. Although some confusibility is still expected, limited discrimination in this likelihood (2 class) score-subspace is possible.

When considering truncated score-spaces defined by zeroth and first-order derivative operators, the likelihood (2 class) score-space and Fisher score are expected to give poorer class discrimination than the likelihood-

[7] If a 'soft-margin' SVM classifier is trained in this score-subspace, a boundary is fitted to all four dimensions of the score-subspace which, while being regularised, trades-off the errors in each dimension. Adding new discriminative dimensions into the score-subspace is valuable because they may cause the decision boundary to shift to more discriminative positions.

ratio score-space due to 'wrap-around'. The likelihood (1 class) score-space contains approximately half the information in the likelihood-ratio score-space, and so is expected to have similar, though slightly poorer, class discrimination.

5.2 Discrimination for score-spaces based on tied states

In pattern recognition, the data belonging to competing classes sometimes possess sufficient similarity that the models for the classes may share parameters without serious degradation in classification performance. For example, GMMs may share components or HMMs may share states. Parameters are shared so that data belonging to contributing classes are pooled to provide robust estimates for the shared parameters. Improved robustness should give improved classification performance. However, parameter sharing is only advantageous if there is similarity between classes. For example, in the speech recognition of isolated letters, the E-like endings of letters `B` and `C` have significant acoustic similarity. When modelled by HMMs, the final states of the two HMMs may be shared. This is called *state-tying* [27]. It is interesting to examine whether score-subspaces defined by the parameters of tied states contain class discriminative information.

To continue with the example of letter recognition, let there be two classes of utterances corresponding to letters `B` and `C`. Each letter is modelled by an HMM score-model. Using the notation defined elsewhere in this report, an utterance $\boldsymbol{O}$ is mapped to a score in the likelihood-ratio score-space, $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$. This score is a function of,

1. the data in the utterance, $\boldsymbol{O}$,
2. the parameter estimates for the score-models, $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\theta}}_1^\top, \hat{\boldsymbol{\theta}}_2^\top)^\top$,
3. for each score-model, the state posteriors $\gamma_j(t)$ for $j = (1, \dots N), t = (1, \dots T)$.

Let states n to N of each model be tied such that $1 < n < N$. As shown in Figure 3, the score-subspace defined by the tied parameters of model `B` are denoted as $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$, and the score-subspace defined by the tied parameters of model `C` as $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(C)}}(\boldsymbol{O})$. Scores are therefore a function of utterance data, score-model parameters and state posteriors. Class discriminative information may originate from any one of these three sources. There are two important issues to consider.

- Is there significant discrimination between classes in the subspaces $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$ or $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(C)}}(\boldsymbol{O})$? If not, these subspaces may be discarded without significant loss in classification performance. This is a 'between-class consideration' and requires knowledge of the distribution of classes `B` and `C` in these score-subspaces. For example for $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$, the final segments of the utterances of each class have significant acoustic similarity and little class discrimination is expected from this source. Also by definition, the parameters of the score-model `B` are independent of the class. However, the state posteriors for states n to N of score-model `B` are affected by the history of each utterance for states 1 to $(n-1)$. Since the initial segments of utterances for each class are acoustically different, the state posteriors for states n to N are expected to be a source of discriminative information. The distributions of the scores for classes `B` and `C` are therefore expected to show some degree of separation in $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$. Hence limited class discrimination is expected in this score-subspace. By similar arguments, some limited class discrimination is expected in the score-subspace $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(C)}}(\boldsymbol{O})$. However the quantity of discriminative information, relative to the subspaces of untied states, is expected to be small. So in practice, these subspaces should be the first to be discarded from the full score-space $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$ if there is a need to reduce dimensionality to save computational expense.

- For a given utterance, is the class discriminative information in $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$ identical to that in $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(C)}}(\boldsymbol{O})$? If so, one of these subspaces may be discarded without loss in classification performance. This is a 'within-class consideration'. By definition, the utterance $\boldsymbol{O}$ is identical for both subspaces and is not a source of class discrimination. Also state parameters are tied and are identical for these two subspaces. However, since states 1 to $(n-1)$ of each score-model differ, the state posteriors for states 1 to $(n-1)$, and hence n to N, of each score-model are expected to differ. This provides a source of difference between the scores in $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(B)}}(\boldsymbol{O})$ and $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{lk(C)}}(\boldsymbol{O})$. Both score-subspaces should therefore be retained in the definition of the full score-space $\boldsymbol{\varphi}^{\mathrm{lr}}(\boldsymbol{O})$.

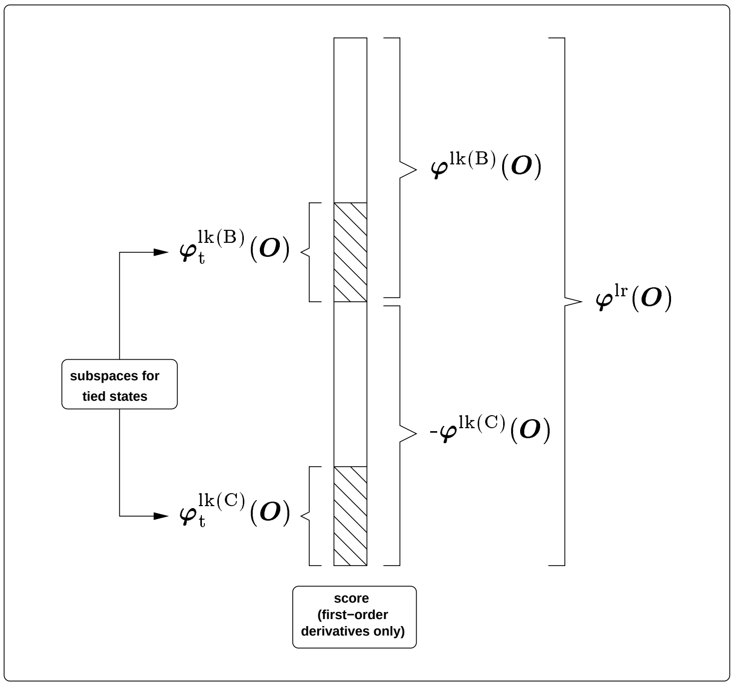


Figure 3: Content of the likelihood-ratio score-space $\varphi^{\mathrm{lr}}(\boldsymbol{O})$ (first-order derivatives only)

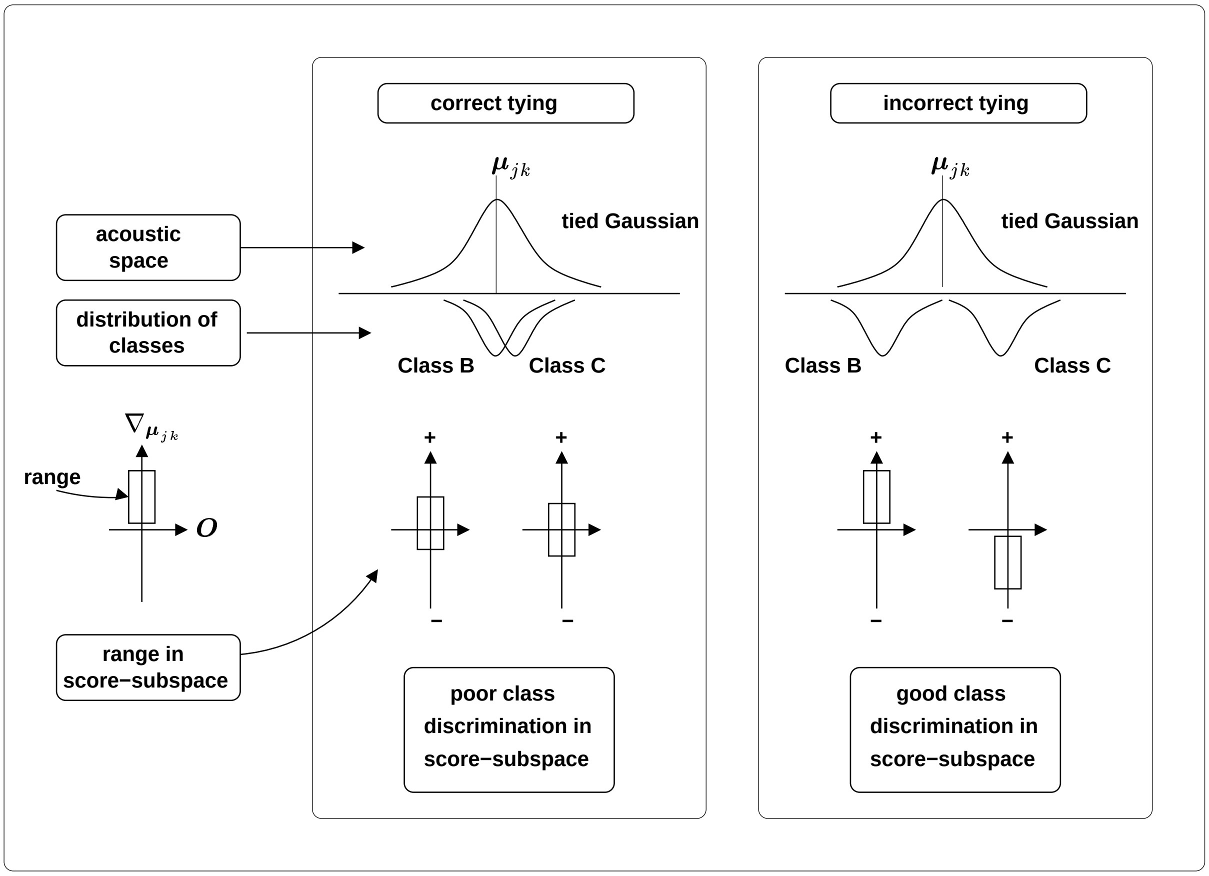


Figure 4: Discrimination in score-subspaces defined on the parameters of tied states

Finally, it is important to consider the consequence of erroneously tying states when the class data modelled by the tied states have distinct distributions. A simple illustration is provided in Figure 4. The left-most column shows the case of correct tying where the distributions for classes B and C overlap in acoustic-space. These distributions are modelled by a 'tied' Gaussian component with mean $\boldsymbol{\mu}_{jk}$. Unit component posteriors are assumed for simplicity. The scores for each class in the subspace $\nabla_{\boldsymbol{\mu}_{jk}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$ overlap and are difficult to separate (however as detailed previously, some separation may be possible if state posteriors for the classes differ). The right-most column shows the case of incorrect tying where the two classes have distinct distributions in acoustic space but a 'tied' Gaussian is used to model both distributions. In this case, good class discrimination is possible in the score-subspace $\nabla_{\boldsymbol{\mu}_{jk}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$. Therefore in cases of erroneous tying, the score-subspaces defined on tied state parameters contain useful class discriminative information. In practical tying, there is always some degree of erroneous tying if only because of model incorrectness, so it may always be useful to include subspaces defined on tied states in the definition of the full score-space. Finally, Figure 4 also gives an excellent illustration of the discriminative ability of score-spaces over standard likelihood estimates from score-models. The two classes in the left-most column cannot readily be separated by their likelihood estimates (i.e. the outputs from the Gaussian component). However, there is clear separation between the classes when the gradients with respect to the Gaussian mean $\nabla_{\boldsymbol{\mu}_{jk}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)$ are inspected.

6 Feature selection in score-space

The scores that populate a particular score-space contain information relating the utterances they represent to the score-models used to define the score-space. However, a classifier in score-space only requires the information which is relevant for class discrimination. Dimensions of score-space which provide little or no class discrimination may be discarded using techniques in feature extraction or feature selection. Feature extraction and feature selection are described in [21], and some relevant issues for feature selection for score-spaces presented in Appendix A. In this section, the term 'feature' is used synonymously with the term 'dimension of score-space'. *Feature extraction* techniques may be used to linearly or non-linearly transform a score-space into a smaller subspace. The aim of such a transformation is to produce a space which allows a classifier to be trained with improved generalisation performance. Alternatively, the transformation may reduce the computational load for the learning algorithm, or produce a set of features which better 'match' the learning algorithm. Examples of feature extraction techniques include Principle Component Analysis (PCA) [21]. *Feature selection* is simpler than feature extraction. The features in the original score-space are ranked according to a performance criterion, and then the best features are selected, according to the criterion, and used to form a new subspace. A classifier is then constructed in this new subspace. Such ranking methods assume that each feature is independent with respect to class discrimination. More sophisticated methods can be used to account for any dependencies between features. In this report, feature selection is preferred to feature extraction since it is computationally cheaper and more robust when there are few training samples, though less optimal. Rather than provide a method for designing score-subspaces and selecting the optimal size of subspace, this report aims to show that feature selection can be used to form a score-subspace which yields lower error rates, and to show which types of features are most important for class discrimination. Data-driven feature selection techniques are expected to be more beneficial for score-spaces than techniques based on prior or expert knowledge. As explained in Appendix A, this is simply because the features which are regarded as most useful through prior knowledge are not necessarily those found most useful by a particular algorithm. Three feature selection methods are used to rank dimensions of feature-space in the experiments in this report; (a) Fisher-ratios, (b) the elements of the weight vector of a linear classifier trained in the full feature-space, and (c) a variant of the Fisher-ratio. These methods are described in Appendix A. The variant of the Fisher-ratio is described and used since it illustrates the differences between the use of Fisher-ratios and the weight vector. The ranking methods may be used to select either a predetermined number of dimensions or all dimensions above a predetermined threshold.

7 Experimental Results

Experiments were conducted using the ISOLET database [4] which consists of single isolated letter utterances. Data was coded at a 10msec frame rate with a window-size of 25.6msec, and parameterised into 12 MFCCs and a log energy term with corresponding delta and acceleration parameters. This gave 39-dimensional feature vectors. Experiments were performed on the harder E-set {B C D E G P T V Z} and the full alphabet. There were 240 utterances per letter from isolet{1,2,3,4} for training and 60 utterances per letter from isolet5 for testing (a total of 2160 and 540 training and test utterances for the E-set, and 6240 and 1560 training and test utterances for the full alphabet).

The baseline HMM system modelled each letter with a 10-emitting state left-to-right continuous density HMM with no skips. Silence was modelled by a single emitting-state HMM with no skips. Each state output distribution was modelled by a GMM with the same number of components and with diagonal covariance matrices. HMMs were trained to maximise likelihood (the HMMs were first flat-started and then the number of components was incrementally increased by mixture-splitting [27]). The models were tested using a Viterbi recogniser constrained to a silence-letter-silence network. Multi-class performance was measured as the utterance error rate obtained by a standard minimum Bayes error classification scheme and also by a majority voting scheme among a full set of 1v1 HMM binary classifiers (for tied letters, the relevant 1v1 classifiers were inspected and then, if necessary, random selection performed [19]; this adds a little 'noise' to all the experimental results in this report, and hence an alternative approach would use the likelihood-ratio or posterior-ratio, when available, to resolve tied decisions). The baseline HMMs were used as the generative models for the SVM classifiers. SVMs were trained and tested using a modified version of SVM^{light} Version 3.02 [11]. Scaling was used as an acceptable form of score-space normalisation. Sequence length normalisation as described in Equations 40 and 41 was also used. A linear kernel was used in this score-space because it gave better performance than GRBF kernels of variable width and polynomial kernels of degree 2 (both homogeneous, inhomogeneous, and inhomogeneous with zero-mean score-space). The linear kernel did not require any tunable parameters and, in some preliminary tests, its performance was insensitive to variations in the SVM trade-off parameter C between 0.01 and 1000. In these experiments C was fixed at 100, and biased hyperplanes were permitted. Performance was assessed using multi-class utterance error rate as obtained by the majority voting scheme.

To test the effect of score-space normalisation, the SVM classifier based on generative models with 2 components per state was used without the scaling approximation to full whitening. The error rate increased from 5.0% to 11.1%. This confirmed the importance of normalising the dynamic ranges of different dimensions of score-space ([19] details experiments comparing the different approximations to full whitening but these were not repeated here due to computational expense and the limited improvement expected over scaling).

abbreviation	score-subspace
m	$\nabla_{\boldsymbol{\mu}_{jk}} \ln p_i(\boldsymbol{O}\vert\boldsymbol{\theta}_i)$
v	$\nabla_{\mathrm{vec}(\boldsymbol{\Sigma}_{jk})} \ln p_i(\boldsymbol{O}\vert\boldsymbol{\theta}_i)$
w	$\nabla_{w_{jk}} \ln p_i(\boldsymbol{O}\vert\boldsymbol{\theta}_i)$
t	$\nabla_{a_{jj}} \ln p_i(\boldsymbol{O}\vert\boldsymbol{\theta}_i)$
l	$\ln p_i(\boldsymbol{O}\vert\boldsymbol{\theta}_i)$
r	$\ln(p_2(\boldsymbol{O}\vert\boldsymbol{\theta}_2)/p_1(\boldsymbol{O}\vert\boldsymbol{\theta}_1))$
p	$\ln P(\omega_i\vert\boldsymbol{O})$

Table 1: Abbreviations used for describing score-subspaces

In this report, the abbreviations detailed in Table 1 are used to describe score-spaces (since diagonal covariance matrices are used in this report, the score-subspace v strictly refers to $\nabla_{\boldsymbol{v}_{jk}} \ln p_i(\boldsymbol{O}|\boldsymbol{\theta}_i)$ where $\boldsymbol{v}_{jk} \equiv \mathrm{diag}\{\boldsymbol{\Sigma}_{jk}\}$). The score-operator was fixed at the zeroth-order and first-order derivative operators, so the terms 'likelihood (2 class)', 'likelihood (1 class)', 'likelihood-ratio' and 'posterior' score-spaces are used to refer to the truncated score-spaces defined respectively on the log likelihood for a generative model for both classes, the log likelihood for a generative model for only the first of the two classes, the log likelihood-ratio between the two competing classes, and the class posterior for the first of the two classes in each constituent binary problem. The exception is the original Fisher-score which is identical to the 'likelihood (2 class)' score-space except that the zeroth-order derivative operator is not included in the score-operator.

7.1 Comparing different score-spaces

Table 2 shows how error rates for E-set classification vary with the complexity of generative score-models. The number of components per class per state was varied from 1 to 6. The likelihood (2-class) and likelihood (1-class) score-spaces are ml score-spaces, and the likelihood-ratio score-space is a mr score-space. The Fisher-score is a m score-space and contains no log likelihood information. The table also details the statistical significance between the error rates obtained using the likelihood-ratio score-space and the standard minimum Bayes error HMM scheme using the same models. McNemar's Test [6] was used to yield the percentage probability, P, that the two algorithms had the same error rate and the differences were simply due to random error. A confidence level was then given as $100 - P$. A statistically significant result may be taken at a 95% confidence level (for the statistical significance tests, any SVM decision for tied letters by random selection was assigned to a separate 'undecided' class).

Num comp. per class per state	HMM		SVM score-space			
	min. Bayes error	majority voting	Fisher	lik (2-class)	lik (1-class)	lik-ratio (stat.sign.)
1	11.3	11.3	6.5	6.1	7.6	6.9 (99.8%)
2	8.7	8.7	10.2	9.3	6.3	5.0 (98.9%)
4	6.7	6.7	23.5	23.2	8.0	5.7 (13.6%)
6	7.2	7.2	31.5	30.6	7.8	6.1 (59.5%)

Table 2: Error rates for different SVM and HMM systems (ISOLET E-set)

The baseline HMMs were comparable to state-of-the-art results on the ISOLET E-set [14] (see also [12] and [22] for full alphabet results). The majority voting scheme among binary HMM classifiers gave the same performance as the minimum Bayes error classification scheme, indicating that the majority voting scheme was an acceptable multi-class technique for the E-set task. The lowest error rates were obtained for HMMs with four components per state, possibly reflecting the gender and class divisions. The original Fisher-score and likelihood (2 class) score-space performed more poorly than the baseline HMM systems except for score-models with 1 component per class per state. The Fisher-score did not contain likelihood information and so performed slightly worse than the likelihood (2-class) score-space. The poor performance of these score-spaces was probably due to the confusion between classes caused by wrap-around. An improvement in performance was obtained using the likelihood (1 class) and likelihood-ratio score-spaces since they were not susceptible to wrap-around. The likelihood (1 class) score-space performed worse than the likelihood-ratio score-space since it only contained approximately half the information. The SVM systems built in the likelihood-ratio score-space consistently outperformed standard HMM systems built using the same score-models. Although the decrease in error rate was only statistically significant for the models with 1 and 2 components per state, the trend in results was consistent from 1 to 6 components per state. The lowest error rate for the likelihood-ratio score-space was 5.0% (statistically significant at a 72.0% confidence level relative to the best HMM system at 4 components per state). The corresponding score-models had 2 components per state, possibly reflecting the gender division in the speakers. Overall, the results show that the best SVM system outperformed the best HMM system, indicating score-spaces successfully extracted extra discriminative information from the score-models and utterances.

There is surprisingly good performance for the Fisher-score and likelihood (2 class) score-space defined using score-models with 1 component per class per state (in total the models had 2 components per state because they modelled both classes). To test whether these components were modelling the gender or class division in the training set, the frame-state alignments were fixed and single Baum-Welch iterations performed separately on the female data, the male data, and the data belonging to the two classes. The updated weights were inspected to discern the division being modelled by the two Gaussians for each state. The statistics revealed that the two Gaussians for the first emitting-state of the HMM, one of the states which is of importance in modelling the E-set, in fact modelled the class division, not the gender division as at first supposed. The good classification performance for this special case was probably due to bi-modal class distributions straddling each of the two Gaussian peaks (see Section 5.1). This would result in a score-space with some discriminative ability.

It is instructive to view the confusion matrices generated by the HMM and SVM classifiers. The SVM classifier using class-conditional generative models with 2 components per state produced the confusion matrix in Table 3. This is compared to the confusion matrix for the best HMM system with class-conditional HMMs of 4 components per state in Table 4. The SVM discriminated the `B` and `E` letters much better than the HMM (compare 1 error with 8 errors). Previous research in [5] and [7] observed a decorrelation in error structure for HMM and SVM classifiers. While this is supported by the `B-E` results and the misclassification of `P` as `B`, for which the SVM made 3 errors and the HMM made zero errors, broadly speaking these results show a correlation in the letter pairs classified wrongly by the SVM and HMM classifiers. The similarity in error structure may be due to the use of the HMM generative models in the definitions of the score-spaces.

	B	C	D	E	G	P	T	V	Z	tot
B	55	0	0	1	0	0	0	4	0	60
C	0	60	0	0	0	0	0	0	0	60
D	0	0	53	2	1	0	0	4	0	60
E	0	0	1	59	0	0	0	0	0	60
G	0	0	0	0	59	0	1	0	0	60
P	3	0	0	1	0	55	0	1	0	60
T	0	0	0	0	1	2	57	0	0	60
V	3	0	0	0	0	0	0	57	0	60
Z	0	2	0	0	0	0	0	0	58	60
tot	61	62	54	63	61	57	58	66	58	540

Table 3: Confusion matrix for SVM classifier (`mr` likelihood-ratio score-space based on class-conditional HMMs with 2 components per state): error rate = 5.0%

	B	C	D	E	G	P	T	V	Z	tot
B	54	0	0	4	0	0	0	2	0	60
C	0	58	0	0	0	0	0	0	2	60
D	0	0	55	1	1	0	1	2	0	60
E	4	0	1	54	0	1	0	0	0	60
G	0	0	0	0	57	0	3	0	0	60
P	0	0	0	0	0	59	0	1	0	60
T	0	1	0	0	2	2	55	0	0	60
V	3	0	0	1	0	0	0	56	0	60
Z	0	3	0	0	0	0	0	1	56	60
tot	61	62	56	60	60	62	59	62	58	540

Table 4: Confusion matrix for HMM classifier (using class-conditional HMMs with 4 components per state): error rate = 6.7%

However, larger and harder tasks are required to investigate this properly. The superior performance of the SVM classifier was mainly due to better classification of the most confusible letter pairs.

Num comp per state	Error rate, %
1	5.7
2	4.7
4	3.4
6	3.7
10	3.9
20	3.9

Table 5: Error rates for full ISOLET task using HMM classifiers of varying complexity

A classifier was constructed in the `mp` posterior score-space based on score-models with 2 components per state (`p` represents $\ln P(\omega_1|\boldsymbol{O})$ where ω_1 is the model for the first of the two classes in each constituent binary problem). The error rate was 11.7%, which represented an increase in error rate from the likelihood (1 class) score-space at 6.3%[8]. Generally, the posterior estimates for the utterances tended either to unity or zero. This was probably caused by poor likelihood estimates due to model incorrectness. Posterior-based training schemes may produce better posterior estimates since they make no assumptions about model correctness. Posterior-based training schemes may also be used to train score-models for posterior and posterior-ratio score-spaces. For example, experiments were performed in [20] using `mr` likelihood-ratio score-spaces defined on score-models trained discriminatively using MMIE (these score-spaces were identical to posterior-ratio score-spaces to zeroth and first-order derivatives since class priors were assumed equal). For score-models with 2 components per state, the resulting classifier yielded an error-rate of 4.3%. This compared favourably with the 5.0% error-rate obtained in this report for an identical classifier but with score-models trained to maximise likelihood.

[8]For reference, the `DvT` classifier did not converge under the baseline SVM parameters, so C was changed to 1000 instead of 100 for this classifier only. The increase in error rate from 6.3% did not appear to be solely due to an over-trained `DvT` classifier.

Experiments were then performed on the full alphabet task. Table 5 details the variation in error rate as the complexity of HMMs was varied for a standard minimum Bayes error classification scheme and the majority voting scheme. Both schemes gave the same error rates. The best HMM classifier, at four components per state, achieved an error rate of 3.4% which compared favourably with results in [14], [12] and [22].

SVM classifiers were tested on the `mr` and full `wmvtr` score-spaces and various of their score-subspaces. For computational reasons, experiments were limited to the likelihood-ratio score-space, and to score-models with 2 components per state. The error rate for the `mr` score-space was 3.0%, and for the full `wmvtr` score-space was 2.4% (statistically significant at 48.8% and 96.4% confidence levels respectively relative to the best HMM system at 3.4% error rate). Therefore, a lower error rate was obtained by including the derivatives of all the HMM parameters, rather than just the means. Important duration information was contained in the derivatives of the transition probabilities. Limited information can be deduced from a comparison of confusion matrices because of the relatively low error rates for the full alphabet task.

7.2 Feature selection in score-space

Experiments were performed which selected subsets of the `wmvtr` likelihood-ratio score-space and then built classifiers in those score-subspaces. The score-models were class-conditional HMMs with two components per state. First, the score-subspaces were selected by parameter type. The error rates for SVM classifiers built in these score-subspaces are detailed in Table 6 together with the dimensionality of the score-subspaces. The SVM classifiers were otherwise identical to the baseline SVM classifiers. As expected, the log likelihood-ratio was a powerful discriminating feature (the error rate for the classifier built in this 1-dimensional score-subspace was not the same as that for the majority voting scheme using HMMs since the SVM decision boundaries were not constrained to have zero bias). Generally, increasing the number of dimensions in the score-subspaces increased the discriminative power of these score-subspaces. Adding new features was beneficial. The results also suggest that there was more discriminative power in the derivatives of the component means than in the derivatives of the component variances. The derivatives of the transitions were also useful as might be expected for a dynamic classification task.

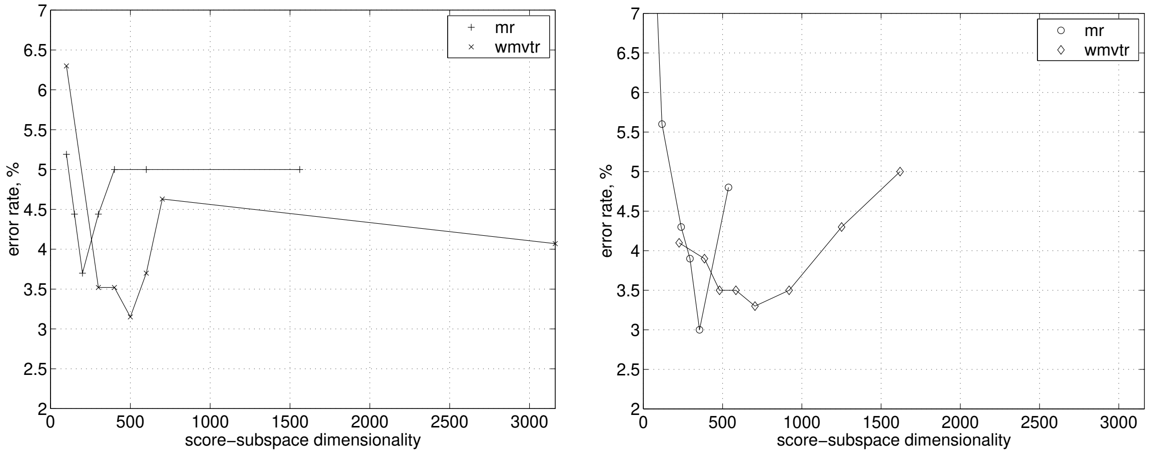
score-space	error rate, %	score-space dimensionality
`r`	8.5	1
`v`	7.2	1560
`m`	5.2	1560
`mr`	5.0	1561
`mv`	5.0	3120
`wmv`	4.4	3140
`mvt`	4.4	3140
`wmvt`	4.1	3160
`wmvtr`	4.1	3161

Table 6: Error rates for subspaces of the likelihood-ratio score-space (based on generative HMMs with two components per state): ISOLET E-set

Simply selecting score-subspaces based on parameter type may be suboptimal because not all members of each parameter type are necessarily equally class discriminative. Experiments were performed comparing the performance of classifiers constructed in score-subspaces selected using other feature selection criteria. First, features from the baseline `mr` and `wmvtr` score-spaces were selected using highest Fisher-ratios. The same fixed number of dimensions were selected in the score-spaces of each constituent binary classifier. Despite a small test set of 540 utterances and its associated 'noise'[9], the graphs in Figure 5(a) illustrate that error rates decreased as discriminative dimensions were added and then increased as less discriminative and noisy dimensions were added. The position of each SVM decision boundary is influenced by all dimensions of score-subspace, and the minimisation of errors in the least discriminative dimensions may cause the decision boundary to shift to less optimal positions. For this reason, adding the less discriminative dimensions of score-space can harm performance. The lowest error rate for the `mr` score-space was 3.7% at 200 dimensions,

[9]Note also that in all experiments, including the feature selection experiments, there is some added 'noise' when random selection is used to decide between unresolved tied letters [19]

and for the **wmvtr** score-space was 3.2% at 500 dimensions. These are respectively statistically significant at 99.1% and 99.7% confidence levels relative to the best HMM system with 4 components per state.



(a) fixed dimensionality per classifier

(b) threshold in Fisher-ratios

Figure 5: ISOLET E-set error rates as dimensions of the **mr** and **wmvtr** score-spaces were varied (based on HMMs with 2 components per state)

Next, score-subspaces were selected using dimensions with Fisher-ratios above a pre-determined threshold. Despite the problems of a small test set as detailed above, the plots for various thresholds in the E-set are given in Figure 5(b). The dimensions plotted are the mean number of dimensions for the constituent binary classifiers for each threshold. Again, the score-models were restricted to HMMs with 2 components per state. As expected, the score-subspaces decreased in size as the threshold in Fisher-ratios was raised. The results indicate that there were optimal thresholds which minimised error rate. The lowest error rate for **mr** was 3.0% at a threshold of 0.05 and a mean number of dimensions per constituent binary classifier of 354 (significant at a 99.7% confidence level relative to the best HMM system with 4 components per state). This compared favourably with the lowest error rate obtained at 3.7% from considering a fixed number of 200 dimensions per classifier. Only one threshold was selected for all the constituent binary classifiers and this may be suboptimal. However, this is expected to be better than selecting a fixed number of dimensions per binary classifier, since Fisher-ratios give some measure of useful class discrimination. Similarly, the lowest error rate for the **wmvtr** score-space was 3.3% at a threshold of 0.035, corresponding to a mean dimension-size of 704 dimensions per binary classifier (significant at a 99.8% confidence level relative to the best HMM system at 4 components per state). This was fractionally poorer than the lowest error rate obtained using 500 dimensions per classifier at 3.2%. The results indicate that both techniques were useful; score-subspaces may either be selected using a threshold in Fisher-ratio values, or by selecting a predetermined number of dimensions per constituent binary classifier.

The feature selection experiments were repeated for the full alphabet task. Subsets of the **wmvtr** score-space were selected according to highest Fisher-ratios. A fixed number of dimensions were selected for each constituent binary classifier. An approximately optimal number of dimensions was found at 500. The resulting error rate was 2.1% (statistically significant at a 99.0% confidence level relative to the HMM system with 4 components per state at 3.4%).

Different feature selection schemes are suitable for different feature spaces. Some simple experiments were performed comparing three different methods for feature selection in score-spaces. These methods are the use of Fisher-ratios, the use of a variant of Fisher-ratios, and the use of the weight vector of a linear classifier trained in the full score-space. These methods are described in Appendix A. All classifiers used the baseline SVM parameters and their score-spaces were defined on HMMs with 2 components per state.

Table 7 shows that the worst feature selection technique for this task was that based on choosing features using elements of the weight vector. Since each score-subspace was selected using a linear SVM, and a linear SVM was then trained in that score-subspace using the same SVM parameters, there was a risk of over-training and poor generalisation. This probably explains the poor performance. Feature selection using the Fisher-ratio and its variant gave classifiers of comparable performance. However, curves were drawn showing the variation in error rate as the score-subspaces were reduced in size. These were more noisy for the variant

task	score-space	feature selection method	score-subspace size	error rate %
E-set	mr	Fisher-ratio	200	3.7
E-set	mr	variant	200	3.5
E-set	mr	weight vector	200	12.0
E-set	mr	Fisher-ratio	300	4.4
E-set	mr	variant	300	4.3
E-set	mr	weight vector	300	9.3
full	wmvtr	Fisher-ratio	500	2.1
full	wmvtr	variant	500	1.9
full	wmvtr	weight vector	500	6.0

Table 7: Comparing different feature selection schemes for score-space

of the Fisher-ratio than for the Fisher-ratio itself. This was probably a consequence of the lack of proper normalisation between different dimensions of score-space for the variant of the Fisher-ratio.

7.3 Analysing discrimination in score-space

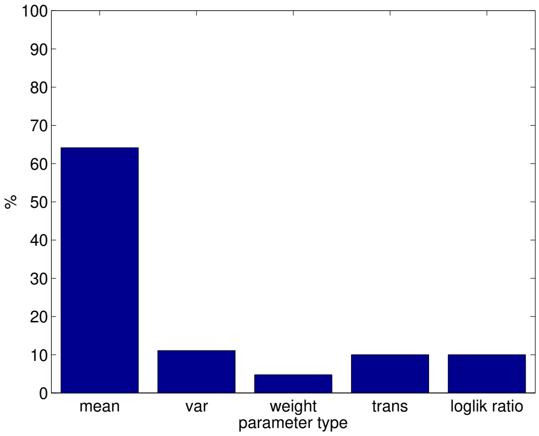
It is generally known that static low-order MFCCs are the most discriminative and useful features for speech recognition. To confirm this, SVM classifiers were analysed for the E-set and full alphabet tasks. Both of the classifiers selected used full `wmvtr` score-spaces based on generative models with 2 components per state. For the E-set task, each of the 36 constituent binary classifiers were analysed and the 10 most discriminative dimensions of score-space extracted from each. This gave a set of 360 discriminative E-set features. Similarly for the full alphabet task, each of the 325 constituent binary classifiers were analysed and the 10 most discriminative dimensions selected from each. This gave a set of 3250 discriminative features, with multiple references to the same feature possible. Fisher-ratios were calculated on the training data. For the E-set, the most discriminative feature was not the log likelihood-ratio in only 2 out of 36 cases, and for the full alphabet in the same 2 out of 325 cases. While the log likelihood-ratio was measured as the 2nd and 3rd most discriminative features in these two cases, the most discriminative features in these cases were both dependent on mean vectors from state 2.

It should be noted that the distribution of features in each discriminative set is heavily biased towards the features derived from Gaussian means and variances, since such features form the majority of score-space. For example, each constituent score-space in this analysis is formed from 3161 features of which only 10 are selected to contribute to the discriminative set. The 3161 features comprised 1560 derivatives of means, 1560 derivatives of variances, 20 derivatives of weights, 20 derivatives of transitions, and 1 log likelihood-ratio. Despite this bias, the distribution of features in each discriminative set still gives important information as to which features are most important for classifiers constructed in score-space. The distributions of the discriminative sets by parameter type are given in Figure 6(a). For the E-set, 288 from 360 features were derived from state parameters (means, variances and weights), a further 36 from transition probabilities, and 36 were log likelihood ratios. For the full alphabet task, 2168 from 3250 features were derived from state parameters, 757 from transition probabilities, and 325 were log likelihood-ratios. For both these tasks, the most important HMM parameters for class discrimination were the Gaussian means. For the full alphabet task, the derivatives of the transitions also showed good discriminative ability. The transition probability for state 2 was particularly important. This highlights the value of the transition probabilities in dynamic classification tasks, particularly when the duration of the class is highly variable as for the full alphabet task. To give an indication of the importance of each HMM state for class discrimination in the E-set and full alphabet tasks, the distributions across states for the features derived from state parameters are given in Figure 6(b). For the E-set, all features dependent on state parameters were derived from states 1, 2, 3 and 4. This is as expected since most discrimination between E-set letters comes from the initial letter segments and not from the E-like endings. For the full alphabet task where a more diverse range of sounds were modelled, the discriminative set contained features dependent on all states. However, states 1, 2 and 3 were the most important states, and states 9 and 10 were also useful. This indicates that the most discriminative states were the initial and end states[10]. Next, Figure 6(c) shows the distributions of the features dependent on state

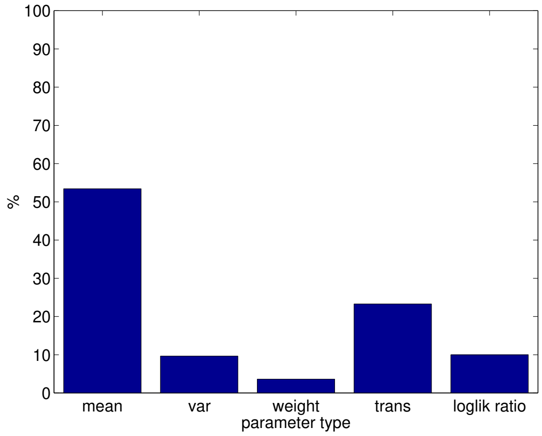
[10] State 2 was more useful than state 1 probably because the first state had to model within-class variation at the silence-letter

(a) distribution of all features by parameter type

E-set (360 features)



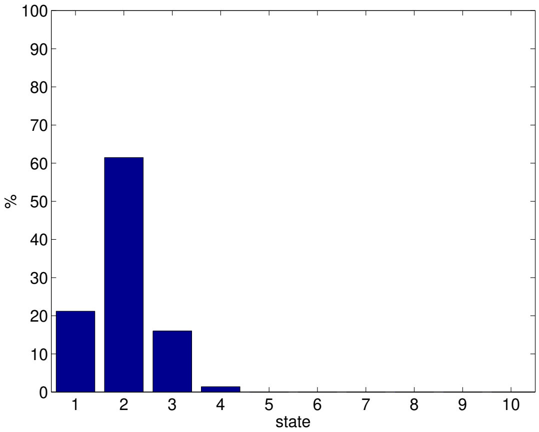
Full alphabet (3250 features)



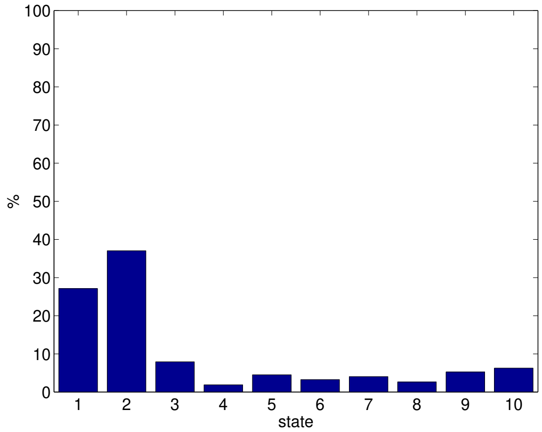
var ≡ variance, trans ≡ transition probability, loglik ratio ≡ log likelihood-ratio

(b) distribution of state features across states

E-set (288 features)

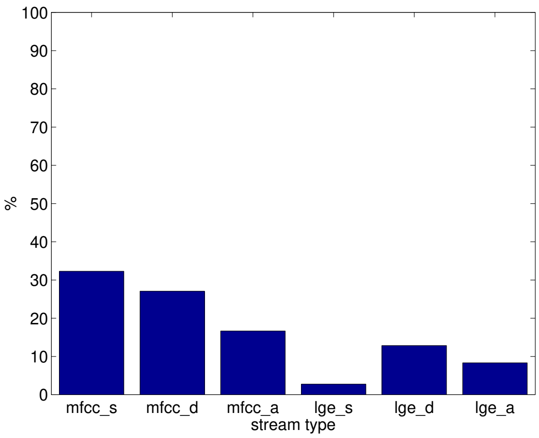


Full alphabet (2168 features)

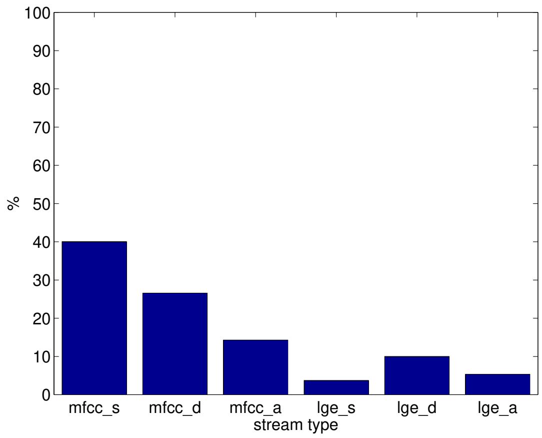


(c) distribution of state features by stream type

E-set (288 features)



Full alphabet (2168 features)



(mfcc ≡ MFCC, lge ≡ log energy, s ≡ static, d ≡ delta, a ≡ acceleration)

Figure 6: Percentage distribution of most discriminative features

parameters across the static, delta and acceleration streams of acoustic-space. The first three columns of each histogram denote streams for MFCCs, the final three columns denote streams for the log energy term. For both the E-set and full alphabet tasks, the static MFCC stream was more important than the delta MFCC stream which was more important than the acceleration MFCC stream. For the log energy terms, the delta term was more important than the acceleration term which was more important than the static term. The rate of variation of the log energy term was a more reliable feature for classification than its static value, possibly because of the sensitivity of energy to varying acoustic conditions. Finally, as expected but not illustrated, the low-order MFCCs and the log energy term were the most useful coefficients in acoustic-space.

The results for the E-set encourage the use of state-tying across E-set letters for states 5 to 10. As explained in Section 5.2, this state-tying produces more robust estimates of the parameters of tied states. If state-tying is valid, such that the class distributions for a particular state overlap and may safely be represented by a tied state, then only limited class discrimination is expected in the score-subspaces defined on the tied state parameters. However, if state-tying is invalid, then significant class discrimination may be learnt in these score-subspaces.

7.4 Sequence length normalisation

Some simple experiments investigated the merit of sequence length normalisation as described in Equations 40 and 41. Results of experiments performed on the E-set and full alphabet tasks are given in Table 8. All SVM classifiers were identical apart from the size of their score-spaces and the presence or absence of sequence length normalisation. The score-spaces were all based on HMMs with 2 components per state, and the SVMs were otherwise identical to the baseline SVM system.

task	score-space	sequence length normalisation?	score-subspace size	error rate %
E-set	`mr`	no	1561 (full)	5.0
E-set	`mr`	yes	1561 (full)	5.0
E-set	`wmvtr`	no	3161 (full)	5.6
E-set	`wmvtr`	yes	3161 (full)	4.1
E-set	`wmvtr`	no	500	3.9
E-set	`wmvtr`	yes	500	3.2
full	`mr`	no	1561 (full)	3.1
full	`mr`	yes	1561 (full)	3.0
full	`wmvtr`	no	3161 (full)	3.1
full	`wmvtr`	yes	3161 (full)	2.4
full	`wmvtr`	no	500	2.2
full	`wmvtr`	yes	500	2.1

Table 8: Investigating sequence length normalisation for different score-spaces

The table shows that sequence length normalisation was beneficial, particularly when the derivatives of the transitions were included in the score-space[11]. When the derivatives of the transitions are omitted from a score-space, there is no duration information contained in the score-space. Any gain in performance due to a reduction of within-class variation caused by different speaking rates can be offset by a loss in performance caused by the omission of duration information from the state posteriors. The importance of the derivatives of transitions for class discrimination was reflected in the set of most discriminative features in Section 7.3. These derivatives are even more important when sequence length normalisation is used. Sequence length normalisation is expected to be helpful when the test utterances are much longer or shorter in duration than the training utterances. Generally, sequence length normalisation eases the task of the classifier in score-space, and hence can improve classification performance.

boundary.

[11]Although the table includes a comparison for 500-dimensional `wmvtr` score-spaces with and without sequence length normalisation, there is no guarantee that the two regimes would require score-subspaces of the same size for optimal classification performance.

7.5 Higher-order derivatives

The baseline mr score-space was augmented with second-order derivatives $\nabla^2_{\boldsymbol{\mu}_{jk}} \ln p_i(\boldsymbol{O}|\boldsymbol{\theta}_i)$. These were considered the most important second-order derivatives. The size of the resulting score-spaces precluded the use of further second-order derivatives. To ease analytical derivation, these second-order derivatives assumed that component posteriors were independent of component means (see Section 4.1). The complexity of the score-models was varied and SVM classifiers built in the score-spaces. The SVM classifiers were otherwise identical to the baseline SVM system. The error rates for the classifiers are given in Table 9 where they are compared to the baseline SVM results. The results show that the inclusion of these higher-order derivatives was not beneficial. This may be due to the false independence assumption for component posteriors and component means. The scores in the subspace of second-order derivatives may therefore be invalid and may skew the SVM decision boundaries to less class discriminative positions. The loss in performance may also be due to a lack of training data for robust estimation of the parameters of the SVM hyperplanes, now that there are significantly more dimensions in score-space. Experiments on larger and more difficult tasks are required. Experiments using feature selection should also be tried.

num comp	score-space	
per state	mr	mr+2nd
1	6.9	7.0
2	5.0	6.3
4	5.7	6.5

Table 9: ISOLET E-set error rates for SVM classifiers built in the mr likelihood-ratio score-space with and without second-order derivatives with respect to the means

8 Conclusions

This report has extended Fisher scores and proposed a set of derivative score-spaces. These score-spaces are defined using derivatives of log likelihood or log posterior estimates taken with respect to the parameters of the model estimates. Various conclusions and remarks have been made concerning these score-spaces which are summarised below,

- When the estimated models are of the correct functional form and score-spaces are defined on an infinite number of derivatives, then linear classifiers trained in certain derivative score-spaces are sufficient for recovering optimal MAP decoders. Optimal MAP decoders are minimum Bayes error classifiers based on correct generative models.
- If the constraints between the estimated and true generative models are relaxed so that each estimated model forms a subset in the parameter space of its corresponding true model, then optimal MAP decoders can be recovered by linear classifiers in score-space, but only under certain conditions. If these conditions are fulfilled, then SVMs are a sensible choice of learning algorithm for such linear classifiers.
- For real problems, true generative models are significantly more complex than the estimated models proposed for them. In such cases, score-spaces are still useful and may be considered as techniques to extract extra discriminative information from the model estimates and the utterances they represent.
- Experiments were conducted on a speaker-independent isolated letter speech recognition task. Classifiers were built and compared in different score-spaces. Score-spaces were defined on log likelihood and log posterior estimates, and were mainly defined on zeroth and first-order derivatives. Classifiers in the likelihood-ratio score-space were shown to outperform classifiers built in the Fisher-score, and also standard HMM classifiers. Truncated derivative score-spaces enabled state-of-the-art classification results for this task.
- The report has also discussed some normalisation issues important for score-spaces, including sequence length normalisation. Some techniques for feature selection in score-space have also been compared.

This report has concentrated on using SVMs for speech recognition where the boundary information for the different classes is known. The application of this research to continuous speech recognition is still an open issue.

Acknowledgements

For funding, N.Smith would like to thank EPSRC and his CASE sponsor, the Speech Group at IBM U.K. Laboratories. He would also like to thank Thorsten Joachims, University of Dortmund, for his program SVM^{light}.

References

[1] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

[2] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[3] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley and Sons, 1973.

[4] M. Fanty and R. Cole. Spoken Letter Recognition. In R.P. Lippmann, J.E. Moody, and D.S. Touretzky, editors, *Neural Information Processing Systems 3*, pages 220–226. Morgan Kaufmann Publishers, 1991.

[5] S. Fine, J. Navrátil, and R. Gopinath. A hybrid GMM/SVM approach to speaker identification. In *Proceedings*, volume 1. International Conference on Acoustics, Speech, and Signal Processing, May 2001. Utah, USA.

[6] L. Gillick and S.J. Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *Proceedings*, volume 1, pages 532–535. International Conference on Acoustics, Speech, and Signal Processing, 1989.

[7] S.E. Golowich and D.X. Sun. A Support Vector/Hidden Markov Model Approach to Phoneme Recognition. In *ASA Proceedings of the Statistical Computing Section*, pages 125–130, 1998.

[8] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 3rd edition, 1996.

[9] T. Hastie and R. Tibshirani. Classification by Pairwise Coupling. Technical report, Stanford University and University of Toronto, November 1996.

[10] T. Jaakkola and D. Haussler. Exploiting Generative Models in Discriminative Classifiers. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[11] T. Joachims. Making Large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.

[12] S. Kapadia. *Discriminative Training of Hidden Markov Models*. PhD thesis, Cambridge University Engineering Department, March 1998.

[13] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[14] P.C. Loizou and A.S. Spanias. High-Performance Alphabet Recognition. *IEEE Transactions on Speech and Audio Processing*, 4(6):430–445, Nov. 1996.

[15] E. Mayoraz and M. Moreira. On the Decomposition of Polychotomies into Dichotomies. In *Proceedings*, pages 219–226. Fourteenth International Conference on Machine Learning (ICML'), 1997. Nashville,USA.

[16] M. Moreira and E. Mayoraz. Improved Pairwise Coupling Classification With Correcting Classifiers. In *Machine Learning: ECML-98 (Lecture Notes in Artificial Intelligence)*, pages 160–171. Springer, 1998. (In the Tenth European Conference on Machine Learning, Chemnitz, Germany, April 1998).

[17] N. Oliver, B. Schölkopf, and A. Smola. *Advances in Large-Margin Classifiers*, chapter Natural Regularization from Generative Models. MIT Press, 2000.

[18] N. Smith and M. Gales. Speech Recognition using SVMs. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

[19] N. Smith, M. Gales, and M. Niranjan. Data-dependent kernels in SVM classification of speech patterns. Technical Report CUED/F-INFENG/TR.387, Cambridge University Eng.Dept., April 2001.

[20] N.D. Smith and M.J.F. Gales. Using SVMs and discriminative models for speech recognition. In *Proceedings*. International Conference on Acoustics, Speech, and Signal Processing, 2002.

[21] C.W. Therrien. *Decision Estimation and Classification*. John Wiley & Sons, 1989.

[22] V. Valtchev. *Discriminative Methods in HMM-based Speech Recognition*. PhD thesis, Cambridge University Engineering Department, March 1995.

[23] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[24] J. Weston, A. Elisseeff, and B. Schölkopf. Use of the ℓ_0-norm with linear models and kernel methods. Technical report, BIOwulf Technologies, Savannah, Georgia, USA, May 2001.

[25] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature Selection for SVMs. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, 2001.

[26] P.C. Woodland and D. Povey. Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition. *To appear in Computer Speech and Language*, 2002.

[27] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. The HTK Book (for HTK Version 3.0), July 2000.

A Feature selection in score-space

In this section, the term 'feature' is used synonymously with the term 'dimension of score-space'. The section describes how feature selection techniques can be used to produce score-subspaces with the aim of increasing classification performance.

As explained in [13], not all features are relevant for a classification problem. In other words, not all dimensions of score-space are useful for discriminating between classes. With expert knowledge, it may be possible to deduce the set of relevant features a-priori. Relevance is tied to the underlying data distributions and optimal classifiers. However, in real-world problems, classifiers are suboptimal since their hypothesis spaces are restricted. For this reason, the 'optimal' set of features selected by a particular classifier may not coincide with the set of relevant features (optimality is here defined as the set of features which produce the lowest error rate for this particular classifier on an unknown test set). Therefore, data-driven feature selection techniques often produce better subspaces for particular classifiers than techniques based on expert or prior knowledge. Despite this, significant overlap is expected between the relevant feature set and the different classifier-specific 'optimal' feature sets. It is also important to consider the effect of finite training data. A finite set of training data may not provide enough information for feature selection techniques to recover the 'optimal' feature set for the particular classifier. This is a form of over-training and may be counter-acted with proper regularisation in the selection process.

Data-driven feature selection techniques can be divided into two broad classes; *filter* and *wrapper* methods [1]. Simply put, filter methods use a local criterion for feature selection and make no use of a final performance metric such as classification error rate. However, wrapper methods feed back a final performance metric and use it in the feature selection process itself. Although such methods usually produce better systems, the methods are computationally more expensive. For this reason, filter methods are used for score-spaces in this report.

A.1 Ranking methods for feature selection

A subset of data-driven filter methods is ranking methods. These methods assume different dimensions of feature space are independent with respect to class discrimination, and hence may select features which are not useful and which are simply highly correlated with classifier-specific 'optimal' features. However, such methods may be more robust than more sophisticated techniques which account for any possible dependence between features. Two ranking methods are; (a) ranking dimensions of score-space using Fisher-ratios, or (b) ranking dimensions according to the magnitude of the corresponding elements of the weight vector of a linear classifier trained in the full feature-space. These methods are discussed later in this section, together with a variant of the Fisher-ratio. The variant is useful for highlighting the differences between the other two feature selection techniques. Ranking methods may be used to select either a predetermined number of features or all features above a predetermined threshold in the selection criterion.

Consider a set of training utterances $\{\boldsymbol{O}_1 \dots \boldsymbol{O}_r \dots \boldsymbol{O}_N\}$ with corresponding scores $\{\boldsymbol{\varphi}(\boldsymbol{O}_1) \dots \boldsymbol{\varphi}(\boldsymbol{O}_r) \dots \boldsymbol{\varphi}(\boldsymbol{O}_N)\}$ and corresponding labels $\{y_1 \dots y_r \dots y_N\}$, where $y_r = +1$ for class ω_1 and $y_r = -1$ for class ω_2. Let the scores occupy a δ-dimensional score-space. The scores for classes ω_1 and ω_2 are distributed in score-space with probability density functions $p(\boldsymbol{\varphi}(\boldsymbol{O})|\omega_1)$ and $p(\boldsymbol{\varphi}(\boldsymbol{O})|\omega_2)$, and estimates $\hat{p}(\boldsymbol{\varphi}(\boldsymbol{O})|\omega_1)$ and $\hat{p}(\boldsymbol{\varphi}(\boldsymbol{O})|\omega_2)$. Let the two classes have respective class means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and respective class covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ in score-space. The diagonal forms of the covariance matrices are the variance vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, where $\boldsymbol{v}_i = \mathrm{diag}\{\boldsymbol{\Sigma}_i\}$. The dth dimension of a vector is denoted without bold font and with the dimension index d in brackets, e.g. the dth dimension of $\boldsymbol{\mu}_i$ is $\mu_i(d)$. The number of training utterances for classes ω_1 and ω_2 are respectively N_1 and N_2, such that $N = N_1 + N_2$.

A.1.1 Fisher-ratio and its variant

One of the forms of Fisher Discriminant Analysis [3] [21] calculates the within-class covariance matrix $\boldsymbol{W}$ and between-class covariance matrix $\boldsymbol{B}$ for a set of training samples, and then selects the dimensions in score-space corresponding to the eigenvectors of $\boldsymbol{G}$ with largest eigenvalues, where $\boldsymbol{G} = \boldsymbol{W}^{-1}\boldsymbol{B}$. For a two-class problem,

$$\boldsymbol{W} = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 \tag{45}$$

$$\boldsymbol{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \tag{46}$$

If all off-diagonal elements of $\boldsymbol{B}$ and $\boldsymbol{W}$ are assumed zero and hence all eigenvectors assumed to lie along the original dimension-axes, then a simpler form of Fisher Discriminant Analysis results. Each dimension of score-space is treated independently and the within-class and between-class relationship between dimensions is ignored. For the dth dimension of score-space where $d \in \{1 \ldots \delta\}$, the Fisher-ratio is given as $g(d)$ where,

$$g(d) = \frac{(\mu_1(d) - \mu_2(d))^2}{v_1(d) + v_2(d)} \tag{47}$$

Simply viewed, the Fisher-ratio $g(d)$ is a measure of class discrimination for each dimension in score-space. A variant of the Fisher-ratio, $g'(d)$ may also be used (see also [25]),

$$g'(d) = \frac{|(\mu_1(d) - \mu_2(d))|}{v_1(d) + v_2(d)} \tag{48}$$

From a numerical viewpoint, $g'(d)$ is not a dimensionless ratio and so is not properly normalised across the different dimensions of score-space. As described in Section A.1.3, it is equivalent to the discriminant measure based on the weight vector of a linear classifier under certain conditions.

In feature selection, dimensions of score-space should be selected with the largest $g(d)$ or $g'(d)$. Feature selection ignores the inter-dimensional relationships, and hence is often suboptimal in its selection of a sub-space. The dimensions selected maximally discriminate between classes according to their criteria. Either a predetermined number of dimensions with largest $g(d)$ or $g'(d)$ may be selected, or all dimensions with $g(d)$ or $g'(d)$ above a predetermined threshold. The score-subspaces formed through feature selection should allow improved classification performance if the discarded dimensions only represented 'noise'.

A.1.2 The weight vector of a linear classifier

For a two-class problem, a linear classifier may be trained to separate the two classes in the full score-space. The classifier may be described by the equation $\boldsymbol{w}^\top \boldsymbol{\varphi}(\boldsymbol{O}) - b = 0$, where $\boldsymbol{w}$ is a weight vector and b is a bias. Each element of the weight vector corresponds to a different dimension of score-space so that $\boldsymbol{w} = (w(1), w(2) \ldots w(d) \ldots w(\delta))^\top$. The magnitude of each element $|w(d)|$ can be used as a measure of class discrimination in the score-space. Feature selection selects the dimensions of score-space with largest $|w(d)|$, and uses these dimensions to form a new score-subspace. The score-subspace formed through feature selection should allow improved classification performance if the discarded dimensions only represented 'noise'. In [24], there is a discussion of how the L0 norm of the weight vector may be used in feature selection.

A.1.3 Equivalence of the feature selection criteria

Dimensions of score-space can therefore be ranked according to the Fisher-ratio $g(d)$, the variant of the Fisher-ratio $g'(d)$, or by using the magnitude of elements of the weight vector $|w(d)|$. Under certain conditions the ranking produced by the variant of the Fisher-ratio and the weight vector are identical. Assume the special case that the scores for each class are distributed normally in score-space so that their class-conditional probability density function estimates are Gaussian (for brevity $\boldsymbol{\varphi}(\boldsymbol{O}_r)$ is abbreviated to $\boldsymbol{\varphi}_r$),

$$\begin{aligned} \hat{p}(\boldsymbol{\varphi}_r|\omega_1) &= \mathcal{N}_{[r,1]}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \hat{p}(\boldsymbol{\varphi}_r)|\omega_2) &= \mathcal{N}_{[r,2]}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \\ \text{where} \quad \mathcal{N}_{[r,k]}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= \frac{1}{(2\pi)^{\delta/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{\varphi}_r - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\varphi}_r - \boldsymbol{\mu}_k)\} \end{aligned} \tag{49}$$

The parameters of the Gaussians may be estimated using a suitable estimation criterion. The decision boundary produced by a Maximum A-Posteriori (MAP) decoder in score-space can be expanded using Bayes Rule for an unknown score $\boldsymbol{\varphi}$, where $P(\omega_1)$ and $P(\omega_2)$ are the class priors for classes ω_1 and ω_2 respectively,

$$\begin{aligned} \ln \hat{P}(\omega_1|\boldsymbol{\varphi}) - \ln \hat{P}(\omega_2|\boldsymbol{\varphi}) &= 0 \\ \ln \hat{p}(\boldsymbol{\varphi}|\omega_1) - \ln \hat{p}(\boldsymbol{\varphi}|\omega_2) + \ln(P(\omega_1)/P(\omega_2)) &= 0 \end{aligned} \tag{50}$$

Substituting for the probability density estimate expressions, and tying the covariance matrices so that $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$,

$$
\begin{aligned}
-(\boldsymbol{\varphi} - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\varphi} - \boldsymbol{\mu}_1) + (\boldsymbol{\varphi} - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\varphi} - \boldsymbol{\mu}_2) + 2\ln(P(\omega_1)/P(\omega_2)) &= 0 \\
2\boldsymbol{\varphi}^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 + 2\ln(P(\omega_1)/P(\omega_2)) &= 0 \\
2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\varphi} - \left(\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - 2\ln(P(\omega_1)/P(\omega_2))\right) &= 0 \qquad (51) \\
\boldsymbol{w}^\top \boldsymbol{\varphi} - b &= 0 \qquad (52)
\end{aligned}
$$

The MAP decision rule can therefore be written in the form of a linear classifier where,

$$
\begin{aligned}
\boldsymbol{w} &= 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \qquad (53) \\
b &= -\boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 + \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - 2\ln(P(\omega_1)/P(\omega_2)) \qquad (54)
\end{aligned}
$$

Let $\boldsymbol{w} = (w(1), w(2) \dots w(d) \dots w(\delta))^\top$ so that the dth element of $\boldsymbol{w}$ is denoted by $w(d)$. Next assume that $\boldsymbol{\Sigma}$ is diagonal. The magnitude of $w(d)$ is then given by,

$$
\begin{aligned}
|w(d)| &= 2\frac{|\mu_1(d) - \mu_2(d)|}{v(d)} \\
&\propto \frac{|\mu_1(d) - \mu_2(d)|}{v(d)} \qquad (55)
\end{aligned}
$$

The variant of the Fisher-ratio is,

$$
\begin{aligned}
g'(d) &= \frac{|\mu_1(d) - \mu_2(d)|}{v_1(d) + v_2(d)} \\
&= \frac{|(\mu_1(d) - \mu_2(d))|}{2v(d)} \\
&\propto \frac{|(\mu_1(d) - \mu_2(d))|}{v(d)} \qquad (56)
\end{aligned}
$$

The two discriminant criteria, $|w(d)|$ and $g'(d)$ are identical within a scaling constant and so produce identical rankings among the dimensions of score-space. Therefore to summarise, the rankings produced by using the weight vector and the variant of the Fisher-ratio are identical under the following conditions,

1. the two classes have class-conditional Gaussian probability density estimates in score-space with tied, diagonal class covariance matrices,
2. the linear classifier is the MAP decision rule,
3. the same estimation criterion is used for calculating the means and variances used for $g'(d)$ and for the probability density estimates, e.g. a maximum likelihood criterion.

For real speech data, the assumptions of Gaussian probability density estimates in score-space are probably invalid. Furthermore, a linear classifier may be obtained using any learning algorithm and its solution may not coincide with the MAP decision rule. For this reason, the ranking produced by the Fisher-ratio variant are expected to differ from the ranking produced by examining the weight vector of a linear classifier trained in score-space. As a result in practice, the score-subspaces produced using these two feature selection criteria are expected to differ.

A.1.4 Comparing the different feature selection criteria

The Fisher-ratio and its variant are calculated using simple sufficient statistics, and the score-space can be reduced in size prior to training a linear or non-linear classifier in the subspace. However, if the weight vector of a linear classifier is used for feature selection, then the linear classifier must first be trained in the full score-space, then another linear or non-linear classifier trained in the resulting score-subspace. This ranking method is computationally more expensive than the use of Fisher-ratios or its variant unless the linear classifier can be readily computed from simple sufficient statistics. Also, there is the issue of over-training the initial linear classifier in the full score-space.

The suitability of each ranking criterion depends on the particular application and training set. For example, Fisher-ratios are calculated using all the training samples from both classes and hence their rankings should be robust, particularly when there are few training samples. However, as Fisher-ratios are defined using all the samples and reflect data density rather than the data near the decision boundary, a ranking based upon Fisher ratios may be suboptimal. Optimality is here defined in terms of the classification performance on unseen data of classifiers built in the selected score-subspaces. Similarly for the variant of the Fisher-ratio. However, a ranking based on the weight vector of a linear classifier couples the selection criterion to the learning algorithm for the linear classifier. For example, if a linear SVM is used to define the weight vector $\boldsymbol{w}$, then $\boldsymbol{w}$ is a linear combination of the support vectors in the full score-space. The selection criterion is then based on the training samples which define the decision boundary rather than all the training samples. When used to select a score-subspace, this selection criterion may perhaps give improved performance. However the selection may be more sensitive to variations in the training set, and there may be a lack of robustness for small training sets.

B Posterior and posterior-ratio score-spaces

Given the likelihood (1 class) score-spaces for a set of likelihood estimates, the posterior score-spaces defined on those estimates can readily be calculated up to the first-order derivatives. The relationship between these score-spaces for the zeroth and first-order derivatives is detailed below. A multi-class approach is maintained to illustrate how truncated posterior score-spaces may be used to map multiple classes into a common score-space. It may then be feasible to train a multi-class decision rule in this common score-space. The analysis is then constrained to the special case of a two-class problem.

Let there be Q classes $\{\omega_i\}|_{i=\{1\ldots Q\}}$. Each class has true class posterior $P(\omega_i|\boldsymbol{O})$, true class likelihood $p(\boldsymbol{O}|\omega_i)$ and true class prior $P(\omega_i)$. The class posteriors form a probability mass function so that $\sum_{i=1}^{Q} P(\omega_i|\boldsymbol{O}) = 1$. The class priors are fixed but not necessarily known, and also form a probability mass function so that $\sum_{i=1}^{Q} P(\omega_i) = 1$. The class likelihood estimates are $\hat{p}(\boldsymbol{O}|\omega_i)$ and the class posterior estimates are $\hat{P}(\omega_i|\boldsymbol{O})$. It is assumed that the functional forms of the true generative models are known so that the true and estimated class likelihoods are of the same functional form. The true and estimated class likelihoods may then be expressed as functions of their parameter vectors $\boldsymbol{\theta}_i$ and $\hat{\boldsymbol{\theta}}_i$ respectively,

$$\begin{aligned} p_i(\boldsymbol{O}|\boldsymbol{\theta}_i) &\equiv p(\boldsymbol{O}|\omega_i) \\ p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) &\equiv \hat{p}(\boldsymbol{O}|\omega_i) \end{aligned}$$

The set of parameter estimates for all classes is denoted by the vector $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\theta}}_1^\top, \ldots \hat{\boldsymbol{\theta}}_i^\top, \ldots \hat{\boldsymbol{\theta}}_Q^\top)^\top$. So then, the posterior score-space for class ω_i expanded to zeroth and first-order derivatives is,

$$\boldsymbol{\varphi}^{\mathrm{pos(i)}}(\boldsymbol{O}) = \begin{bmatrix} \ln \hat{P}(\omega_i|\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O}) \end{bmatrix} \tag{57}$$

Given the likelihood estimates, the zeroth-order derivative of the posterior score-space is,

$$\begin{aligned} \ln \hat{P}(\omega_i|\boldsymbol{O}) &= \ln\big(\hat{p}(\boldsymbol{O}|\omega_i)P(\omega_i)\big) - \ln \hat{p}(\boldsymbol{O}) \\ \text{where} \qquad \hat{p}(\boldsymbol{O}) &= \sum_{q=1}^{Q} \hat{p}(\boldsymbol{O}|\omega_q)P(\omega_q) \end{aligned} \tag{58}$$

Next, consider the first-order derivative of this posterior score-space, $\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O})$,

$$\begin{aligned} [\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O})]^\top \mathrm{d}\hat{\boldsymbol{\xi}} &= \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O}) \\ &= \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) + \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln P(\omega_i) - \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{p}(\boldsymbol{O}) \\ &= \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) - \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{p}(\boldsymbol{O}) \\ &= \mathrm{d}_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i) - \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{p}(\boldsymbol{O}) \\ &= [\nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_i - \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{p}(\boldsymbol{O}) \end{aligned} \tag{59}$$

Evaluating the derivative of $\ln \hat{p}(\boldsymbol{O})$ with respect to $\hat{\boldsymbol{\xi}}$, and assuming that $p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \neq 0, \forall q$,

$$\begin{aligned}
\mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{p}(\boldsymbol{O}) &= \frac{1}{\hat{p}(\boldsymbol{O})} \mathrm{d}_{\hat{\boldsymbol{\xi}}} \hat{p}(\boldsymbol{O}) \\
&= \frac{1}{\hat{p}(\boldsymbol{O})} \mathrm{d}_{\hat{\boldsymbol{\xi}}} \big(\sum_{q=1}^{Q} p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) P(\omega_q) \big) \\
&= \frac{1}{\hat{p}(\boldsymbol{O})} \big(\sum_{q=1}^{Q} \mathrm{d}_{\hat{\boldsymbol{\theta}}_q} p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) P(\omega_q) \big) \\
&= \sum_{q=1}^{Q} \frac{P(\omega_q)}{\hat{p}(\boldsymbol{O})} \mathrm{d}_{\hat{\boldsymbol{\theta}}_q} p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \\
&= \sum_{q=1}^{Q} \frac{P(\omega_q)}{\hat{p}(\boldsymbol{O})} \frac{p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q)}{p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q)} \mathrm{d}_{\hat{\boldsymbol{\theta}}_q} p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \\
&= \sum_{q=1}^{Q} \frac{p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) P(\omega_q)}{\hat{p}(\boldsymbol{O})} \mathrm{d}_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \\
&= \sum_{q=1}^{Q} \hat{P}(\omega_q|\boldsymbol{O}) \mathrm{d}_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \\
&= \sum_{q=1}^{Q} \hat{P}(\omega_q|\boldsymbol{O}) [\nabla_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_q
\end{aligned} \tag{60}$$

Returning to Equation 59,

$$\begin{aligned}
[\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O})]^\top \mathrm{d}\hat{\boldsymbol{\xi}} &= \mathrm{d}_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O}) \\
&= [\nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_i - \big(\sum_{q=1}^{Q} \hat{P}(\omega_q|\boldsymbol{O}) [\nabla_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_q \big) \\
&= \big(1 - \hat{P}(\omega_i|\boldsymbol{O})\big) [\nabla_{\hat{\boldsymbol{\theta}}_i} \ln p_i(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_i)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_i - \sum_{q=1, q\neq i}^{Q} \hat{P}(\omega_q|\boldsymbol{O}) [\nabla_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q)]^\top \mathrm{d}\hat{\boldsymbol{\theta}}_q
\end{aligned} \tag{61}$$

So then, the posterior score-space for model i expanded to zeroth and first-order derivatives is,

$$\begin{aligned}
\boldsymbol{\varphi}^{\mathrm{pos(i)}}(\boldsymbol{O}) &= \begin{bmatrix} \ln \hat{P}(\omega_i|\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O}) \end{bmatrix} \\
\text{where} \qquad \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_i|\boldsymbol{O}) &= \begin{bmatrix} \alpha_1 \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ \vdots \\ \alpha_q \nabla_{\hat{\boldsymbol{\theta}}_q} \ln p_q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_q) \\ \vdots \\ \alpha_Q \nabla_{\hat{\boldsymbol{\theta}}_Q} \ln p_Q(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_Q) \end{bmatrix} \\
\text{and} \qquad \alpha_q &= \begin{cases} 1 - \hat{P}(\omega_i|\boldsymbol{O}) & \text{if} \quad q = i \\ -\hat{P}(\omega_q|\boldsymbol{O}) & \text{if} \quad q \neq i \end{cases}
\end{aligned} \tag{62}$$

It may be feasible to use this truncated posterior score-space in a multi-class framework. For example, first train a set of Q class generative model estimates, one model for each class in the Q-class problem. Then process each class in turn. For class ω_1, each utterance is transformed into Q distinct first-order derivatives, one defined on each likelihood estimate. These Q first-order derivatives are then weighted and appended to the zeroth-order derivative using the structure in Equation 62. For each utterance, this gives a super-vector in the truncated posterior score-space for the model for class ω_1. This procedure is then repeated for the remaining classes, class ω_2 to ω_Q. At the end of this process, all the training data has been mapped into the common truncated posterior score-space. A multi-class classifier may then be trained in this score-space.

Next consider the special case of a binary classification problem where $Q = 2$. The first-order derivatives for the posterior score-space for class ω_1 is $\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_1|\boldsymbol{O})$ and for class ω_2 is $\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_2|\boldsymbol{O})$, where $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\theta}}_1^\top, \hat{\boldsymbol{\theta}}_2^\top)^\top$,

$$\begin{aligned}
\nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_1|\boldsymbol{O}) &= \begin{bmatrix} \left(1 - \hat{P}(\omega_1|\boldsymbol{O})\right) \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\hat{P}(\omega_2|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \\
&= \begin{bmatrix} \hat{P}(\omega_2|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\hat{P}(\omega_2|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \\
&= \hat{P}(\omega_2|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\xi}}} \ln \frac{p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)}{p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)} \qquad (63)
\end{aligned}$$

$$\begin{aligned}
\text{Similarly} \quad \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_2|\boldsymbol{O}) &= \begin{bmatrix} -\hat{P}(\omega_1|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ \hat{P}(\omega_1|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \\
&= -\hat{P}(\omega_1|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\xi}}} \ln \frac{p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)}{p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)} \qquad (64)
\end{aligned}$$

Therefore, the first-order derivatives in the posterior score-spaces for each class are weighted versions of their likelihood-ratio score-spaces. For each, the weight term is the class posterior of the competing class, and there is a negative multiplying factor if the posterior score-space is defined for the second of the two classes. A full definition of the posterior score-spaces for binary problems, truncated after their first-order derivatives, is given below, where $\hat{\rho}(\boldsymbol{O}) = \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) - \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)$,

$$\boldsymbol{\varphi}^{\text{pos}(1)}(\boldsymbol{O}) = \begin{bmatrix} \ln \hat{P}(\omega_1|\boldsymbol{O}) \\ \hat{P}(\omega_2|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\xi}}} \hat{\rho}(\boldsymbol{O}) \end{bmatrix} \qquad (65)$$

$$\boldsymbol{\varphi}^{\text{pos}(2)}(\boldsymbol{O}) = \begin{bmatrix} \ln \hat{P}(\omega_2|\boldsymbol{O}) \\ -\hat{P}(\omega_1|\boldsymbol{O}) \nabla_{\hat{\boldsymbol{\xi}}} \hat{\rho}(\boldsymbol{O}) \end{bmatrix} \qquad (66)$$

As detailed in Section 3, optimal MAP decoders can be recovered by training linear classifiers in the posterior score-spaces expanded to an infinite order of derivatives with respect to the parameter estimates. Truncated score-spaces may allow good approximations to optimal MAP decoders. Using the above expressions, the first-order derivative in the posterior-ratio score-space is obtained as,

$$\begin{aligned}
\nabla_{\hat{\boldsymbol{\xi}}} \ln \frac{\hat{P}(\omega_1|\boldsymbol{O})}{\hat{P}(\omega_2|\boldsymbol{O})} &= \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_1|\boldsymbol{O}) - \nabla_{\hat{\boldsymbol{\xi}}} \ln \hat{P}(\omega_2|\boldsymbol{O}) \\
&= \left(\hat{P}(\omega_2|\boldsymbol{O}) + \hat{P}(\omega_1|\boldsymbol{O})\right) \nabla_{\hat{\boldsymbol{\xi}}} \ln \frac{p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)}{p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)} \\
&= \nabla_{\hat{\boldsymbol{\xi}}} \ln \frac{p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1)}{p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)} \qquad (67)
\end{aligned}$$

As expected, the first-order derivative in the posterior-ratio score-space is identical to the first-order derivative in the likelihood-ratio score-space. Therefore when considering two-class problems and score-spaces truncated to the zeroth and first-order derivatives, the posterior-ratio score-space $\boldsymbol{\varphi}^{\text{pr}}(\boldsymbol{O})$ and likelihood-ratio score-space $\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O})$ are identical in all but their zeroth-order derivatives, where $\hat{\rho}_{\text{pos}}(\boldsymbol{O}) = \ln \hat{P}(\omega_1|\boldsymbol{O}) - \ln \hat{P}(\omega_2|\boldsymbol{O})$ and $\hat{\rho}(\boldsymbol{O}) = \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) - \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2)$,

$$\boldsymbol{\varphi}^{\text{pr}}(\boldsymbol{O}) = \begin{bmatrix} \hat{\rho}_{\text{pos}}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\xi}}} \hat{\rho}_{\text{pos}}(\boldsymbol{O}) \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{\text{pos}}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \qquad (68)$$

$$\boldsymbol{\varphi}^{\text{lr}}(\boldsymbol{O}) = \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\xi}}} \hat{\rho}(\boldsymbol{O}) \end{bmatrix} = \begin{bmatrix} \hat{\rho}(\boldsymbol{O}) \\ \nabla_{\hat{\boldsymbol{\theta}}_1} \ln p_1(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_1) \\ -\nabla_{\hat{\boldsymbol{\theta}}_2} \ln p_2(\boldsymbol{O}|\hat{\boldsymbol{\theta}}_2) \end{bmatrix} \qquad (69)$$