DATA-DEPENDENT KERNELS IN SVM CLASSIFICATION OF SPEECH PATTERNS

Nathan Smith[†], *Mahesan Niranjan*[‡]

[†]Cambridge University Engineering Dept, Trumpington Street, Cambridge, CB2 1PZ, U.K.
[‡] Dept of Computer Science, Sheffield University, Portabello Street, Sheffield, S1 4DP, U.K.
nds1002@eng.cam.ac.uk M.Niranjan@dcs.shef.ac.uk

ABSTRACT

Support Vector Machines (SVMs) have recently proved to be powerful pattern classification tools with a strong connection to statistical learning theory. One of the hurdles to using SVMs in speech recognition, and a crucial aspect of SVM design in general, is the choice of the kernel function for non-separable data, and the setting of its parameters. This is often based on experience or a potentially costly search. This paper gives some experimental justification for the Fisher kernels proposed in [4]; kernels are obtained and their extra regularisation and use of labelled and unlabelled data discussed. Fisher kernels are derived from generative probability models of the data, and are a first-step to implementing kernels for variable length sequences.

1. Introduction

SVMs have recently been compared to Gaussian mixture classifiers for phonetic classification [2], and obstacles to extending SVMs to continuous speech recognition detailed; these include the problem of multi-class classification, the estimation of posterior probabilities, and the use of context-dependency and dynamics. This paper concentrates on the problem of choosing the kernel, and is a first step to implementing kernels for variable length sequences [4]. This paper is restricted to binary vowel classification.

Given a set of training data $(\boldsymbol{x}_1, \ldots \boldsymbol{x}_\ell)$ with corresponding targets $(y_1, \ldots y_\ell)$, where $\boldsymbol{x}_i \in \boldsymbol{R}^N$ and $y_i \in \{-1, +1\}$, a learning algorithm attempts to find a decision function $f_{\boldsymbol{w}}(\boldsymbol{x}_i) = y_i, \forall i$. A Support Vector Machine [3] is a classifier which jointly maximises the margin between the classes and minimises the misclassification error on the training data, the trade-off between the two being controlled by a regularisation parameter C. (The margin is a region of space surrounding the decision boundary which is vacant of data points in the linearly separable case). The decision boundary $\boldsymbol{w}$ is given by the minimisation of $\psi(\boldsymbol{w}, \boldsymbol{\xi})$, where $\boldsymbol{\xi} = (\xi_1 \ldots \xi_\ell)$ and ξ_i is the L1-norm for a misclassified $\boldsymbol{x}_i$; constraints have been omitted for brevity.

$$\psi(\boldsymbol{w}, \boldsymbol{\xi}) \quad = \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{\ell}\xi_i \qquad (1)$$

The standard solution method is via maximisation of the dual $W(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_1 \ldots \alpha_\ell)$ and α_i is the Lagrange multiplier associated with $\boldsymbol{x}_i$.

$$W(\boldsymbol{\alpha}) \quad = \quad \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j (\boldsymbol{x}_i^T \boldsymbol{x}_j) \qquad (2)$$

subject to

$$0 \leq \alpha_i \quad \leq \quad C,\ i = 1, \ldots \ell, \qquad \sum_{i=1}^{\ell}\alpha_i y_i = 0$$

Only data points which are misclassified or which lie within or on the margin have non-zero Lagrange multipliers - these are termed *support vectors* and are sufficient for defining the decision boundary $\boldsymbol{w}$.

For linearly non-separable data, the *input space* $\boldsymbol{x}_i$ is mapped to a meaningful, often high-dimensional, *feature space* $\phi(\boldsymbol{x}_i)$ where it is hoped that the data points will become linearly separable; the classifier is then built in this feature space. For computational tractability, all dot products in the feature space are replaced by kernel functions operating in the input space, $\phi(\boldsymbol{x}_i).\phi(\boldsymbol{x}_j) \mapsto K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Any function which satisfies Mercer's condition [3] represents a dot-product in a possibly unspecified feature space. Kernel selection is therefore equivalent to feature-space selection and will determine the capacity of the resulting classifier. The standard kernels include $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma^2\}$ and $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j)^d$ which give respectively Gaussian Radial Basis Function (`GRBF`) and homogeneous polynomial classifiers. These kernels should replace the dot product in Equation 2.

There is a need for a principled approach to choosing the kernel and tuning its parameters, as both the shape and scaling of the kernel will affect the shape of the decision boundary and hence the classifier performance.

2. Kernels from generative probability models

A kernel is proposed in [4] which is derived from a generative probability model of the data and which attempts to find a natural comparison between examples induced by the model. Although the examples can be variable length data sequences, this paper only considers exam-

ples which are single vectors. For a generative probability model $p(\boldsymbol{x}_i|\boldsymbol{\theta})$, let $\boldsymbol{x}_i \in \boldsymbol{R}^N$ and $S = \{\boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ where S defines a Riemannian manifold. Let the Fisher kernel between two vectors be $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where I is the Fisher Information matrix, and $U_{\boldsymbol{x}_i} = \nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{x}_i|\boldsymbol{\theta})$,

$$\begin{aligned} K(\boldsymbol{x}_i, \boldsymbol{x}_j) &\propto \phi^T_{\boldsymbol{x}_i} I \phi_{\boldsymbol{x}_j} \\ &\propto U^T_{\boldsymbol{x}_i} I^{-1} U_{\boldsymbol{x}_j} \quad (3) \\ \text{where} \quad \phi_{\boldsymbol{x}_n} &= I^{-1} U_{\boldsymbol{x}_n} \quad (4) \end{aligned}$$

Because the kernel has been defined using a dot product, there is no need to check for Mercer's condition. The kernel implies a feature space $\phi(\boldsymbol{x}_i)$ which is the *natural gradient space* of the log likelihood model, and a distance metric which is the local Riemannian metric. In the experiments in this paper, the simpler kernel $K_U(\boldsymbol{x}_i, \boldsymbol{x}_j) \propto U^T_{\boldsymbol{x}_i} U_{\boldsymbol{x}_j}$ is used for derivational simplicity. The Fisher Information has been approximated as Identity which is equivalent to assuming that the co-ordinate system for $\boldsymbol{\theta}$ is orthonormal and the space S is Euclidean. This is a strong assumption, and should be investigated in any future work.

If examples are still linearly non-separable in the feature space $\phi(\boldsymbol{x}_i)$, then homogeneous polynomials giving higher-order decision boundaries can be drawn in this feature space using $\tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_U(\boldsymbol{x}_i, \boldsymbol{x}_j)^{d_{\tilde{K}}}$, where $d_{\tilde{K}}$ is the kernel degree. These kernels also satisfy Mercer's condition [3].

2.1. Deriving the Fisher kernels

Two generative probability models are investigated for a two-class problem; the *Two-Gaussian model* and the simpler *Single-Gaussian model* [5]. Reduction in computational complexity is achieved by diagonalising the covariance matrices. Let us define,

$$\mathcal{N}_{[n,k]}(\boldsymbol{\mu}_k, \Sigma_k) = \frac{(2\pi)^{-N/2}}{|\Sigma_k|^{1/2}} \exp^{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)} \quad (5)$$

Two-Gaussian model

$$p(\boldsymbol{x}_n|\boldsymbol{\theta}) = w_1 \mathcal{N}_{[n,1]}(\boldsymbol{\mu}_1, \Sigma_1) + w_2 \mathcal{N}_{[n,2]}(\boldsymbol{\mu}_2, \Sigma_2) \quad (6)$$

For unlabelled data, the parameters can be found by standard mixture modelling techniques; alternatively if labelling is provided for a two-class problem as in these experiments, then the parameters can refer to different classes, and ML estimation of the parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2, w_1, w_2\}$ may be performed, where the components are respectively the means, covariances and prior probabilities for classes 1 and 2. `TwoGauDiag` refers to the kernel based on a model with diagonal covariances; the corresponding feature space is $2(2N+1)$-dimensional. For simplicity in derivation, the constraint $w_1 + w_2 = 1$ has not been used. The formulation can readily be extended to a multiple mixture generative model. A matrix partition is denoted by $\vdots$, and the Kronecker matrix product by $\otimes$. The Fisher kernel $K_U(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is given by,

$$K_U(\boldsymbol{x}_i, \boldsymbol{x}_j) = [\nabla_{\boldsymbol{\theta}} \ln \boldsymbol{\alpha}_i]^T [\nabla_{\boldsymbol{\theta}} \ln \boldsymbol{\alpha}_j] \quad (7)$$

where

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \ln \boldsymbol{\alpha}_n = \Big[& \tau_{[n,1]} S_{[n,1]} \vdots \tau_{[n,2]} S_{[n,2]} \vdots \tau_{[n,1]} Q_{[n,1]} \vdots \\ & \tau_{[n,2]} Q_{[n,2]} \vdots \frac{\tau_{[n,1]}}{w_1} \vdots \frac{\tau_{[n,2]}}{w_2} \Big]^T \quad (8) \\ \boldsymbol{\alpha}_n = \; & p(\boldsymbol{x}_n|\boldsymbol{\theta}) \quad (9) \\ \tau_{[n,k]} = \; & \frac{w_k \mathcal{N}_{[n,k]}(\boldsymbol{\mu}_k, \Sigma_k)}{\boldsymbol{\alpha}_n} \quad (10) \\ S_{[n,k]} = \; & (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \quad (11) \\ Q_{[n,k]} = \; & \frac{1}{2} \Big[-[\mathrm{vec}(\Sigma_k^{-1})]^T + S_{[n,k]} \otimes S_{[n,k]} \Big] \quad (12) \\ \mathrm{vec}(F) = \; & [f_{11}, f_{12}, \ldots f_{mn}]^T \text{ (F is } m \times n) \quad (13) \end{aligned}$$

Single-Gaussian model

$$p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \mathcal{N}_{[n,g]}(\boldsymbol{\mu}_g, \Sigma_g) \quad (14)$$

The model assumes all data points are generated by a single multivariate Gaussian with global mean and covariance, $\boldsymbol{\theta} = \{\boldsymbol{\mu}_g, \Sigma_g\}$. ML estimation is here used. `SinGauDiag` refers to the kernel derived from a probability model with a diagonal covariance, and so the feature space is $2N$-dimensional. The Single-Gaussian kernel is a special case of the Two-Gaussian kernel but with $\tau_{[n,1]} = 1$ and all the partitions due to Class 2 omitted. The partition for w_1 is also omitted because the weight is already known at unity.

3. Experimental results

The experiments classify static speech patterns in 2 and 13 dimensions, taken from steady-state vowel data in the Peterson-Barney (PB) and TIMIT databases respectively. PB data consists of the frequencies of the 1st and 2nd formants in Hz. TIMIT data consists of the central frames of vowels using a parameterisation for input space of 12 MFCCs and a log energy term; there is a $10msec$ frame rate with a $20msec$ window. The vowel data is only extracted from the TIMIT training set, not its test set. The TIMIT vowel classes are also given the PB labelling; vowels `iy`, `uh`, `uw` and `er` are labelled as classes 1,8, 9 and 10 respectively. The experimental scope only includes binary vowel classification.

In these experiments a hard-split is made; for PB, the same 150 points are used to train the generative model and the SVM, and there are 150 test points; for TIMIT there are 300 points to train the generative model, a different 300 data points to train the SVM, and 300 test points. The over-training on PB data is permitted because of a lack of available data. In this paper, *corr* refers to percentage correctness and *nsv* to the number of support vectors. For the `GRBF` kernel, the width is set as $\sigma = \left(\prod_{d=1}^{N} (x(d)_{\max} - x(d)_{\min})\right)^{1/N}$, where for example, $x(d)_{\max}$ is the maximum value of the dth component of the $\boldsymbol{x}_i$ vectors.

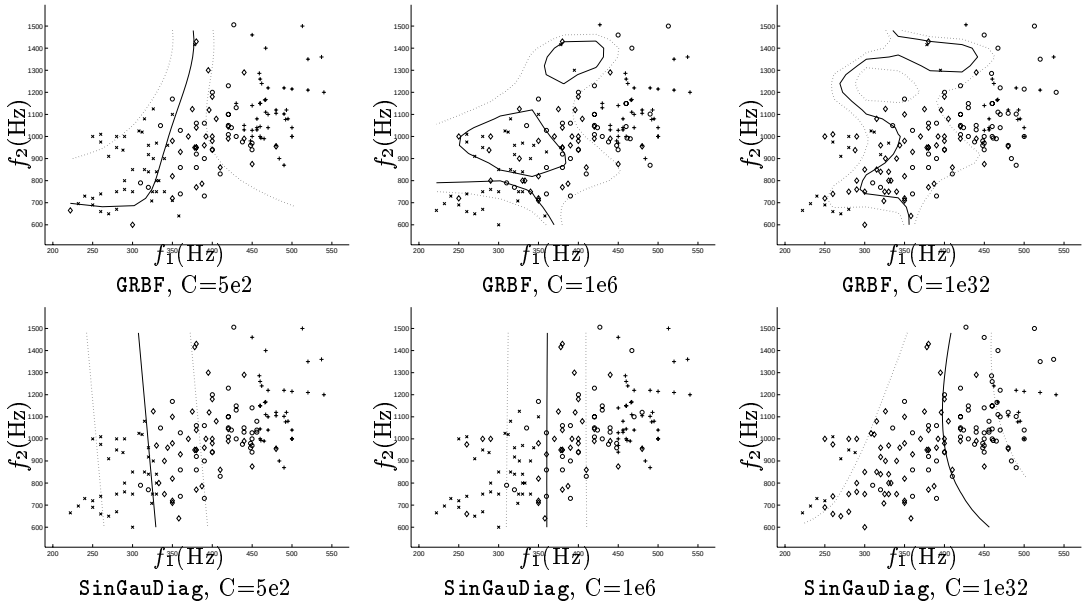


Figure 1: PB data (8v9): The boundary for the Fisher kernel appears less sensitive to the C-parameter

3.1. Sensitivity to the C-parameter

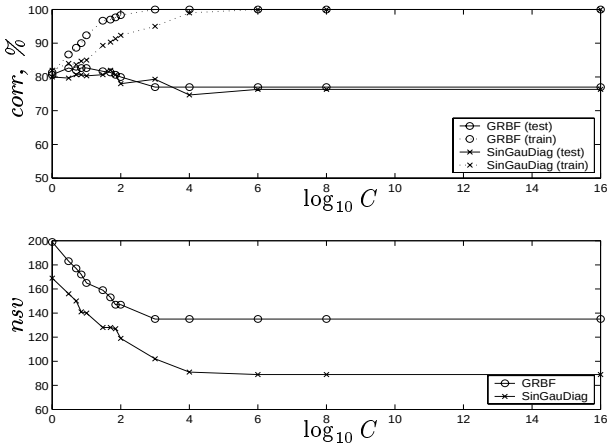


Figure 2: TIMIT data (8v9): Effect of the C-parameter on the correctness and number of support vectors

If the feature space does not permit linear separability, C will affect the position of the boundary. Different values of C are used to produce the plots for PB data in Figure 1; the plots suggest that the boundary position is less sensitive to C when using `SinGauDiag` than when using `GRBF`. This may be due to (a) the linear-like shape of the kernel (see Section 3.4); a perfectly linear kernel without any numerical regularisation would be constrained to give linear boundaries in the input space, and/or (b) the low-dimensionality of the feature space may add some regularisation because it limits the capacity of the classifier. A wider `GRBF` kernel may also give smoother boundaries, but then a method for optimising the `GRBF` width would be required.

Figure 2 compares the performance of `SinGauDiag` and `GRBF` on the TIMIT data; although `SinGauDiag`'s test correctness is marginally lower, it is achieved with fewer support vectors, thereby decreasing computation in test phase. However, for `SinGauDiag` the training correctness increases more slowly as C is increased than for `GRBF`; this may indicate a reluctance for the Fisher kernel to over-train. As C is increased, the classifiers tend towards minimum error classifiers.

Therefore, there is some extra regularisation from using the Fisher kernel, and the shape of its boundary appears to be less sensitive to C. Inspection of performance curves show that the optimum C-values depend on the combination of dataset and kernel. However, for comparison purposes in these experiments, C is kept fixed for a given dataset, and a 'good', but not necessarily optimum, value for TIMIT data is chosen at 30.

3.2. The effect of increasing the dimension of the feature space

Instead of drawing linear boundaries in the feature space, the capacity of the classifier may be increased by drawing polynomials. Table 1 details such experiments; α_{max} is the maximum Lagrange multiplier value. For $8v9$, higher order polynomials give greater training correctness, indicating that the polynomials are better fitting the training data. However, particularly with `SinGauDiag`, the test correctness degrades indicating over-training. For $1v10$, the α_{max} values are less than the upper limit of 30, and the training correctness is always 100%; this indicates significant separability between the two classes. Therefore, it is likely that the higher order polynomials are copying the order-1 polynomial's linear boundary in the feature space, and so there will be no over-fitting; this is demonstrated by the lack of degradation in the test correctnesses for $1v10$.

kernel (prob.)	$d_{\tilde{K}}$	α_{max}	*corr*		*nsv*	final dim
			test	train		
G(8v9)	1	30.0	81.67	96.67	159	∞
G(8v9)	2	30.0	78.67	99.00	169	∞
G(8v9)	3	30.0	78.33	99.67	191	∞
G(8v9)	4	30.0	79.00	100.00	214	∞
G(8v9)	5	26.6	78.33	100.00	231	∞
S(8v9)	1	30.0	80.67	89.33	128	26
S(8v9)	2	1.21	70.33	100.00	120	351
S(8v9)	3	6e-04	69.67	100.00	132	3276
G(1v10)	1	7.21	78.33	100.00	17	∞
G(1v10)	2	3.69	78.67	100.00	38	∞
G(1v10)	3	2.62	78.67	100.00	58	∞
G(1v10)	4	2.14	79.00	100.00	86	∞
G(1v10)	5	1.87	78.00	100.00	111	∞
S(1v10)	1	8.18	70.33	100.00	23	26
S(1v10)	2	5e-03	74.00	100.00	26	351
S(1v10)	3	5e-05	72.00	100.00	40	3276

Table 1: TIMIT data (C = 30): varying the degree of the homogeneous polynomial fitted to the feature space (`G` = `GRBF`, `S` = `SinGauDiag`)

An alternative view of these results is to realise that the `SinGauDiag` and polynomial together combine to give a single kernel transformation to a final feature space whose dimension increases with the polynomial degree. These results suggest that when the two classes are significantly non-separable, there is some beneficial regularisation in limiting the final feature space dimension (i.e. the capacity of the classifier); when the classes are more separable, this regularisation is not as greatly required. (The feature space for `GRBF` is already infinite, and the polynomial simply changes the subspace in which the classifier is built). Table 1 also gives the final feature space dimensions [1].

3.3. Using labelled/unlabelled data

There are several pattern classification problems in which labelling is expensive, but large amounts of unlabelled data is available. The Fisher kernel-SVM approach is applicable to such situations, where the generative model may be estimated from the unlabelled data and the discriminant function computed from the labelled data.

3.4. The kernel shapes

Figure 3 plots the Fisher, `GRBF` and polynomial kernels in the PB data-space, relative to a central datapoint. The Fisher kernels show a greater similarity to the polynomial than to the `GRBF`; future work should check whether `SinGauDiag` is strictly linear or just close to linear. `TwoGauDiag` has a much larger dynamic range than `SinGauDiag` (orders of magnitude 1 and 10^{-4} respectively).

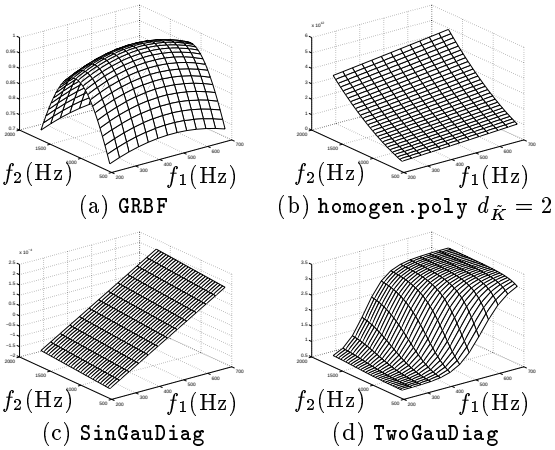
For TIMIT, `GRBF`, `SinGauDiag` and `TwoGauDiag` give respectively 81.67%, 80.67% and 78.00% on the $8v9$ problem, and 78.33%, 70.33% and 77.67% on the $1v10$ problem. This confirms the expectation that the Two-Gaussian model should be more suitable for the separable $1v10$ problem than the Single-Gaussian model, but vice-versa for the $8v9$ problem which is more inseparable.

4. Conclusions

These results show that the Fisher kernel is an alternative to the standard Gaussian RBF, and its classification boundary appears less sensitive to the C-value. The Fisher kernels limit the dimension of the feature space, and the results suggest that limiting the feature space dimension may give some beneficial regularisation, particularly when the two classes are very inseparable. When data is costly to label, Fisher kernels provide a means of using both the labelled and unlabelled data. Despite the greater computational cost in using Fisher kernels, these kernels provide a first step to using SVMs with variable length sequences, as required for continuous speech recognition.

5. Acknowledgements

For funding, N.Smith thanks EPSRC and his present CASE sponsor, the Speech Group at IBM U.K. Laboratories, and Steve Gunn, University of Southampton, for a starting point in the SVM code.



(a) `GRBF` (b) `homogen.poly` $d_{\tilde{K}} = 2$

(c) `SinGauDiag` (d) `TwoGauDiag`

Figure 3: Different kernels for PB data (8v9)

6. REFERENCES

1. C.J.C. Burges. A tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):955–974, 1998.
2. P. Clarkson and P.J. Moreno. On the use of Support Vector Machines for phonetic classification. In *Proceedings*. ICASSP, 1999.
3. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
4. 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.
5. N. Smith and M. Niranjan. Data-dependent kernels in SVM classification of speech patterns. Technical Report CUED/F-INFENG/TR.387, Cambridge University Eng.Dept., 2000.