Non-Linear Feature Selection for Classification

M. Brown
University of Manchester Institute of Science and Technology
Department of Electrical Engineering and Electronics,
Main Building, Sackville Street,
Manchester M60 1QD, U.K.

N. P. Costen
The Manchester Metropolitan University
Department of Computing and Mathematics,
John Dalton Building, Chester Street,
Manchester M1 5GD, U.K.

Abstract

This paper addresses the issues associated with performing feature or parameter selection for non-linear classifiers using a basis pursuit regularization framework. New results on representing the feature selection problem as a primal/dual calculation for both hard and soft margin classification problems are derived, and it is shown that optimal feature selection can be posed, in dual form, as a set of $2n$ linear inequality constraints. While this is efficient, it does limit the technique to non-linear kernels that have a finite expansion, such as polynomials. The issues associated with both efficiently calculating a polynomial basis pursuit classifier are then addressed and the technique is shown to improve discrimination performance on the MNIST digit set.

1 Introduction to Feature Selection

Feature and parameter selection is an important part of many machine learning problems. It can be used to identify important terms when the problem is poorly structured and thus learn more about the variables' information content. Similarly, it can be used to build more robust classifiers, rejecting variables that do not contain significant information. Examples of application in this area include gene selection from microarray data and text categorization [7]. This has become even more important in recent years, as a large range of different kernel transformations have been proposed as potential features [9]. These dual goals of knowledge extraction and data fitting are common in most statistical classification problems. However, optimal feature selection is an NP-complete problem.

A typical method of stepwise forwards selection and backwards elimination algorithms, allow developers to perform a locally optimal search through the space of sparse models. Also, because this information is rarely represented in the final model, estimates of the prediction accuracy may be overly optimistic. Basis pursuit regularization [1, 3, 11] provides an alternative framework for developing features selection algorithms, where instead of a combinatorial search being performed, the aim is to minimize a regularization

BMVC 2004 doi:10.5244/C.18.63

function such as

$$f(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{t} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1 \tag{1}$$

where $\boldsymbol{t}, \boldsymbol{y}$ and $\boldsymbol{\theta}$ are the target training data, classifier predictions and hyperplane parameters respectively. The 1-norm on the parameter vector introduces derivative discontinuities when $\theta_i = 0$ into the regularization function, which produces pressure to remain at zero [4, 13]. Sparse model searching is therefore a globally optimal continuous optimization procedure that can be solved in polynomial time. The use of a 1-norm to measure parameter sparseness is similar to using a 2-norm on the output errors to approximate minimizing the total number of classification errors. Sparseness and classification error minimization are both NP-complete problems, however using continuous, soft approximations makes the problem tractable and transform combinatorial, discrete search problems into a continuous optimization problem.

Previous work has investigated how to generate the complete, optimal parameter locus, $\boldsymbol{\theta}(\lambda)$, in order to explore the set of optimal sparse models as the regularization parameter varies between 0 (maximum likelihood solution) and ∞ (prior model), for both regression and classification scenarios [4]. This allows model developers to investigate whether effects such as Simpson's paradox [10], are present in the set of sparse models. In addition, an on-line version, for sequentially training optimal, sparse classification and regression models has been proposed [2]. However, all of this work has been performed for linear models. In this paper, the theory is extended by considering how the calculations can be obtained in the dual space (data space) as this is often the starting point in standard Support Vector Machines for considering the use of non-linear kernels. In addition, the efficient implementation of polynomial kernels is discussed. The approach of using non-linear basis pursuit classification is validated on the MNIST [8] digit set and is shown to improve classification performance over comparable linear bases.

1.1 Basis Pursuit Regularization

As shown in Equation 1, basis pursuit regularization involves using a 2-norm loss function on the output errors with a 1-norm on the parameter values. For a classification scenario with the model

$$y = \boldsymbol{\phi}^T\boldsymbol{\theta} + b \tag{2}$$

where the qualitative decision is obtained by taking the sign of y, the primal, soft margin basis pursuit regularization function is expressed as

$$\begin{array}{ll} \min_{\boldsymbol{\theta},b,\boldsymbol{\xi}} & \|\boldsymbol{\xi}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1 \\ s.t. & \mathrm{diag}(\boldsymbol{t})(\boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{1}b) + \boldsymbol{\xi} \geq \mathbf{1} \end{array} \tag{3}$$

where the exemplar data $[\boldsymbol{\Phi}, \boldsymbol{t}]$ consists of l training samples, each feature vector is of length n and the class targets are bi-polar values $[-1, 1]$. This is a piecewise quadratic programming problem which can be transformed into a quadratic programming problem by mapping the parameter vector into a $2n$ dimensional space

$$\begin{array}{ll} \min_{\boldsymbol{z},b,\boldsymbol{\xi}} & \|\boldsymbol{\xi}\|_2^2 + \lambda\boldsymbol{z}^T\mathbf{1} \\ s.t. & \mathrm{diag}(\boldsymbol{t})([\boldsymbol{\Phi} \ \ -\boldsymbol{\Phi}]\boldsymbol{z} + \mathbf{1}b) + \boldsymbol{\xi} \geq \mathbf{1} \\ & \boldsymbol{z} \geq \mathbf{0} \end{array} \tag{4}$$

where, using Matlab notation, $\boldsymbol{z}(1:n) = \boldsymbol{\theta}(\theta > 0)$, $\boldsymbol{z}(n+1:2n) = -\boldsymbol{\theta}(\theta < 0)$ and $z = 0$ for the remaining values. Alternatively, iterative approaches can be developed that either generate the complete parameter locus from an initial value at $\boldsymbol{\theta} = \mathbf{0}$, or iteratively search the piecewise quadratic surface. The advantage of transforming the problem into a larger (parameter) space is that standard QP routines can be exploited and, as shown in Section 3, optimization theories employed. It should be noted that there are $2n+l$ linear inequality constraints in this primal form and $2n+1+l$ parameters in the optimization function.

For a particular value of λ, the set of parameters that are non-zero is known as the active parameter set and the set of data points such that $\mathrm{diag}(\boldsymbol{t})(\boldsymbol{\Phi\theta} + \mathbf{1}b) \leq \mathbf{1}$ is known as the active data set. The active data set is used to calculate the values of the active parameters for a particular classifier. The inactive data points play no role in this calculation, as with normal Support Vector Machines (SVM) [12]. The major difference with the standard SVM formulation is the use of a 1-norm on the parameter values, rather than a 2-norm. This has the effect of inducing a different distance norm when calculating the size of the margin, which calculates a margin that lies in the subspace of active features.

1.2 Linear and Polynomial Basis

The major limitation of the proposed approach is the assumption of a linear decision boundary in Equation 2. Standard SVMs extend this work by expressing Equation 3 in its dual form, as an expansion across the data points, rather than the parameters. This in turn allows the calculation to take place in data space, rather than parameter space, and thus a wide range of flexible, non-linear kernels can be employed. In this paper, it is shown that the piecewise differentiable nature of the 1-norm in Equation 3 means that a qualitative term of the form $\mathrm{sgn}(\boldsymbol{\theta})$, is always present in the dual calculation.

Therefore, the paper concentrates on how a polynomial expansion of the original features can be used to produce non-linear discriminant boundaries, while still providing a convenient framework for performing feature selection.

2 Primal and Dual Calculations

In this section, is it is shown how the feature selection, basis pursuit optimization algorithm can be analyzed and calculated in dual (data) space, rather than in the primal (parameter) space. The limitations associated with this approach are also discussed.

2.1 Hard Margin Linear Classifier

When hard margin classifiers are considered, it is presumed that the data is linearly separable, and the aim is to calculate the largest margin that separates the two classes, by minimizing the size of the classifier's parameter vector. The primal form of the hard margin linear classifier is given by

$$\begin{array}{l} \min_{\boldsymbol{\theta},b} \|\boldsymbol{\theta}\|_1 \\ s.t. \quad \mathrm{diag}(\boldsymbol{t})(\boldsymbol{\Phi\theta} + \mathbf{1}b) \geq \mathbf{1} \end{array} \tag{5}$$

which is a piecewise linear programming (LP) problem. Transforming to z-space, the primal problem can be expressed as

$$\begin{array}{ll} \min_{\boldsymbol{z},b} \boldsymbol{z}^T \mathbf{1} & \\ s.t. & \mathrm{diag}(\boldsymbol{t})([\boldsymbol{\Phi} - \boldsymbol{\Phi}]\boldsymbol{z} + \mathbf{1}b) \geq \mathbf{1} \\ & \boldsymbol{z} \geq \mathbf{0} \end{array} \tag{6}$$

and creating a Lagrangian gives

$$\begin{array}{rcl} L(\boldsymbol{z}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) & = & \boldsymbol{z}^T \mathbf{1} - \boldsymbol{\alpha}^T (\mathrm{diag}(\boldsymbol{t})([\boldsymbol{\Phi} - \boldsymbol{\Phi}]\boldsymbol{z} + \mathbf{1}b) - \mathbf{1}) - \boldsymbol{\beta}^T \boldsymbol{z} \\ & s.t. & \boldsymbol{\alpha} \geq \mathbf{0} \\ & & \boldsymbol{\beta} \geq \mathbf{0} \end{array} \tag{7}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the vectors of Lagrange multipliers. Differentiating the Lagrangian with respect to $\boldsymbol{\theta}$ and b and equating to zero gives

$$\begin{array}{ll} \dfrac{\delta L}{\delta \boldsymbol{z}} = \mathbf{0} = & \mathbf{1} - \left[\begin{array}{c} \boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \\ -\boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \end{array} \right] \boldsymbol{\alpha} - \boldsymbol{\beta} \\ \dfrac{\delta L}{\delta b} = 0 = & \boldsymbol{\alpha}^T \boldsymbol{t} \end{array} \tag{8}$$

and back-substituting the constraints into the Lagrangian gives a formulation in a which is simply a Linear Programming algorithm (LP),

$$\begin{array}{ll} \min_{\boldsymbol{\alpha}} -\boldsymbol{\alpha}^T \mathbf{1} & \\ s.t. & \boldsymbol{\alpha} \geq \mathbf{0} \\ & \boldsymbol{\alpha}^T \boldsymbol{t} = 0 \\ & \left[\begin{array}{c} \boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \\ -\boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \end{array} \right] \boldsymbol{\alpha} \leq \mathbf{1}. \end{array} \tag{9}$$

Apart from the last set of constraints, this is a fairly trivial LP problem in l parameters. The last set of constraints determine which $z_i > 0$, or equivalently, which $\beta_i = 0$. They do not appear in the normal SVM formulation where a 2-norm on the parameter vector is used. Perhaps most importantly, they represent $2n$ constraints on the parameter values when the rest of the calculation is being performed in data space. If non-linear kernels that have an infinite expansion were used, they would represent an infinitely large constraint set. Therefore, the rest of the paper is focussed on non-linear kernels/features that have a finite expansion in parameter space. Polynomials are one such non-linear feature.

2.2 Soft Margin Linear Classifier

To express Equation 3 in its dual form, consider its corresponding Lagrangian,

$$\begin{array}{rcl} L(\boldsymbol{z}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) & = & \lambda \boldsymbol{z}^T \mathbf{1} + \frac{1}{2} \boldsymbol{\xi}^T \boldsymbol{\xi} - \boldsymbol{\alpha}^T (\mathrm{diag}(\boldsymbol{t})([\boldsymbol{\Phi} - \boldsymbol{\Phi}]\boldsymbol{z} + \mathbf{1}b) + \boldsymbol{\xi} - \mathbf{1}) - \boldsymbol{\beta}^T \boldsymbol{z} \\ & s.t. & \boldsymbol{\alpha} \geq \mathbf{0} \\ & & \boldsymbol{\beta} \geq \mathbf{0} \end{array} \tag{10}$$

and differentiating with respect to $\boldsymbol{\theta}$, $\boldsymbol{\xi}$ and b to find the minimum gives

$$\begin{aligned}
\frac{\delta L}{\delta \boldsymbol{z}} &= \boldsymbol{0} = & \lambda \boldsymbol{1} - \boldsymbol{\beta} - \begin{bmatrix} \boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \\ -\boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \end{bmatrix} \boldsymbol{\alpha} \\
\frac{\delta L}{\delta \boldsymbol{\xi}} &= \boldsymbol{0} = & \boldsymbol{\xi} - \boldsymbol{\alpha} \\
\frac{\delta L}{\delta b} &= 0 = & \boldsymbol{\alpha}^T \boldsymbol{t}.
\end{aligned} \tag{11}$$

Before back-substituting into Equation 10, it is worthwhile considering these optimality constraints in a bit more detail. From the second constraint, the Lagrange multipliers $\boldsymbol{\alpha}$ are simply equal to the non-negative residuals $\boldsymbol{\xi}$. Therefore, in the active data space, they are related to the optimal parameter values via $\boldsymbol{\alpha} = \boldsymbol{t} - (\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{1}b)$. This gives a set of linear equations on the active data points from which to determine the optimal active parameters ($\boldsymbol{\theta}$, or $\boldsymbol{z}$ and b).

Substituting from Equation 11 into Equation 10 gives

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{1} \\
s.t. \quad & \boldsymbol{\alpha} \geq \boldsymbol{0} \\
& \boldsymbol{\alpha}^T \boldsymbol{t} = 0 \\
& \begin{bmatrix} \boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \\ -\boldsymbol{\Phi}^T \mathrm{diag}(\boldsymbol{t}) \end{bmatrix} \boldsymbol{\alpha} \leq \lambda \boldsymbol{1}
\end{aligned} \tag{12}$$

which is a QP problem in data space with a constraint set that again includes the $2n$ parameter constraints. Using a 2-norm on the active data residuals, transforms the previous LP problem into one with a quadratic objective. The constraint set is the same. In both of these cases, it is interesting to note that the relationship between the Lagrange multipliers and the optimal parameters enters as linear inequality constraints.

2.3 Examples

The effectiveness of the algorithms is demonstrated by a pair of artificial problems, the results of which are shown in Figures 1 and 2. The former is a hard-margin case, and as can be seen generates exactly the same classification model as a standard SVM. The latter is a soft-margin case, and demonstrates that reducing the margin of the classifier will increase the number of parameters included in calculating the hyperplane.

2.4 Size of Active Data and Parameter Spaces

While the dual calculation may initially appear attractive because the LP/QP problems are solved in data space (l parameters and $2n+l+1$ constraints), when compared with the primal problems, it should be noted that in basis pursuit regularization there are always more active data points than parameters. This result highlights the fact that reducing λ does not always increase the number of active parameters, especially when it is very small. Indeed, as the margin decreases in size and the active data set decreases, this can form a reducing, strict upper bound on the active feature set which must also decrease in size.

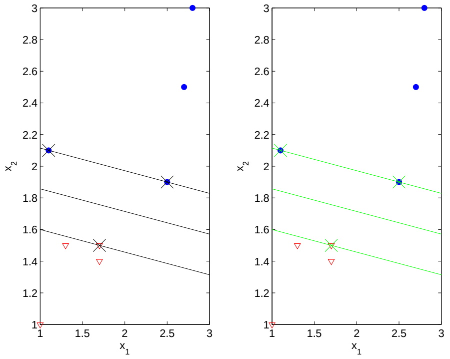

Figure 1: Classifier calculated for a hard-margin case, on the left the 1-norm classifier and on the right a 2-norm SVM.

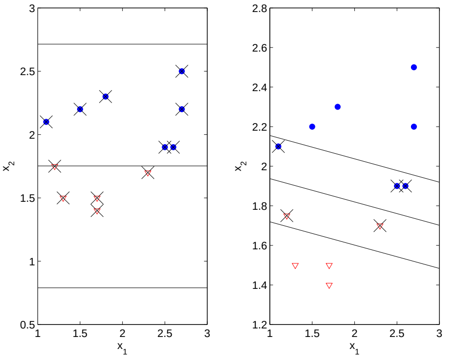

Figure 2: Classifier calculated for a soft-margin case, on the left using 1 parameter and on the right using 2.

This result can be proven by contradiction. Consider when a parameter vector, $\boldsymbol{\theta}$, is optimal and there are more active parameters than data. Then the active Hessian matrix in Equation 3 is singular because its rank is determined by the minimum number of active data and parameters. Let $\boldsymbol{n}$ be a member of the corresponding null space and consider the local update $\boldsymbol{\theta}+\rho\boldsymbol{n}$ where $\rho > 0$. Without loss of generality, this can be assumed to reduce $\|\boldsymbol{\theta}\|_1$ (if not, then the corresponding alternative $\boldsymbol{\theta}+\boldsymbol{n}$ will). However, because $\boldsymbol{n}$ lies in the Hessian's null space, the model's output is unchanged as are the corresponding errors. Therefore, a new parameter vector has been found with a smaller regularization function value, which contradicts the previous assumption of optimality. The only other possibility is that $\boldsymbol{n}$ lies along a contour of $\|\boldsymbol{\theta}\|_1$ and in that case, the subspace can be search to find a new parameter value such that at least one active parameter becomes inactive. This has the same effect because the number of active parameters can be reduced in a stepwise fashion.

Therefore, even though the number of variables in the dual problem is l (rather than $2n+l$ for the primal problem), there may be more efficient implementations that operate directly on the space of active parameters.

2.5 Calculating the Parameter Locus

One motivation for considering the primal/dual formulation is to obtain a more efficient implementation in the dual space, where the QP problem is solved over l data points and there exists an efficient method for performing the inner product between the basis functions, which results in the kernel formulation for parameter estimation and prediction. In this paper it has been shown that for the basis pursuit problem, these two representations are almost equivalent, and the introduction of the $2n$ linear inequality constraints in the dual problem is a major difference from the standard SVM formulation. Each parameter has 2 “soft” linear inequality constraints which determine whether it will be a member of the active set. While this is an added complexity, compared to the standard SVM formulation, it should be noted that this is linear in n and reasonably efficient to calculate and is a significant reduction on performing a combinatorial search across all feature subsets, when performing optimal, hard feature selection.

In Section 2.4, it was shown that the number of active data and parameters are equivalent, so there is little to be gained by calculating the solution in the dual/data space, though the insights provided may motivate new learning algorithms. In the optimal parameter locus calculation is performed in data, $\boldsymbol{\alpha}$, space, it is easy to establish that the Lagrange multipliers, $\boldsymbol{\alpha}$, are a piecewise linear function of λ. This is because they are equal to the residuals on the active data points, and zero otherwise. Thus, because the parameters are piecewise linear functions and the residual is just a linear mapping of the parameters, it follows that the residual path must also be piecewise linear. The linear segments can be calculated using

$$\boldsymbol{\alpha}(\mu) = \boldsymbol{\alpha}_0 - \mu\boldsymbol{\Phi}\boldsymbol{H}^{-1}\mathrm{sgn}(\boldsymbol{\theta}). \tag{13}$$

where μ is the free non-negative parameter that determines the linear segment for the Lagrange multipliers, and $\boldsymbol{\alpha}_0$ is the initial value at a knot on the locus.

3 Efficient Implementation

In many SVM approaches, the rational for performing the calculation in the dual space is that

- The number of data points is typically much less than the number of parameters in feature space.
- The inner product calculation, over the features, which produces the gram matrix, can be efficiently calculated using another representation.

In this paper, it has been shown that the size of the active feature space is generally equal to the size of the active data space, so a matrix of the same size must be inverted, irrespective of whether the primal or dual problem is solved. Similarly, the inner product calculation over feature space does not involve all of the potential features, so there is little opportunity to use a transformed calculation in performing the inner product. This is due to the use of a 1-norm on the parameter vector in the primal problem, rather than a 2-norm.

The main focus for an implementation, is how the set of $2n+l$ linear inequality constraints that determine whether the parameters and data are active or inactive. If either the potential parameter or data space is too large, evaluating this constraint set will take a considerable time. However, it should be noted that the size of both the active data and parameter spaces is bounded above by $\min[n,l]$.

4 Examples

The algorithm was studied via a pair of data sets. The first was an entirely artificial two class problem. Existing in $\mathbf{R}^3$, the classes form a pair of concentric ovals in the first two dimensions. The third dimension is a random Gaussian distribution, included to demonstrate the feature selection; the data were displaced from the origin by adding the vector $[3,4,5]$ to each observation. These data cannot be separated via a linear classifier; the plane derived depends almost entirely on jitter in the third dimension.

However, when a polynomial kernel, $k\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = (\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 + 1)^2$ is applied to the data, the classifier extracts the correct dimensions. As can be seen from Figure 4, evolution of the parameters is not a simple scaling as the parameter λ is decreased. Initially only the pure terms relating to the first dimension are used, below a value of approximately 50, those relating to the second enter and the bilinear term (x_1, x_2) reduces in importance, demonstrating Simpson's paradox.

The boundaries shown in Figure 3 depend upon the squared terms for the first two parameters, their bilinear term and their individual linear ones. There is also a very small (four orders of magnitude smaller) linear term for the third dimension. It is worth noting that there a number of active points (those on or within the ±1 contours). This reflects the use of automatic relevance detection [5, 6] for selecting the most appropriate model. The model selected as having the highest probability is the 710th; the final, 995th, model has eight non-zero parameters.

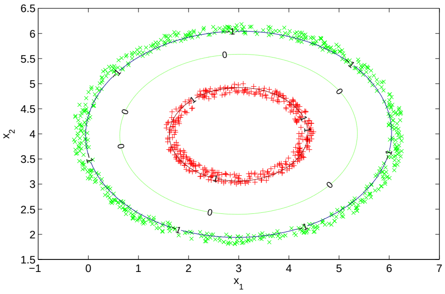

Figure 3: The classification boundary as found by the polynomial classifier.

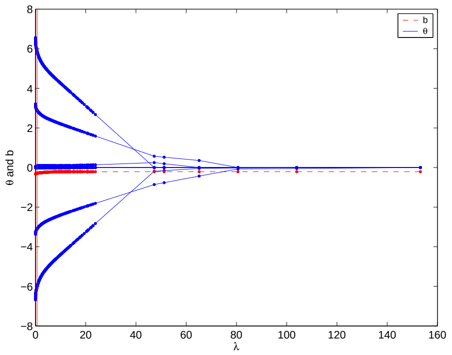

Figure 4: Variation in the parameters used by the polynomial classifier.

The second data set was the MNIST digits. This is a large, widely used, set of hand-written digits [8], some examples of which are shown in Figure 5. This comprises a training set of 60,000 digits, each centred in a 28 × 28 image, and a 10,000 digit test set. Both the set have approximately, but not exactly, even distributions of digits. For testing purposes, a sub-ensemble was formed of 8000 randomly-selected images, with an even distribution of digits. To both stabilize the data and also exclude zero-variance samples, a PCA was performed on this ensemble and the most variable 20 eigenvectors retained (this accounted for 64.3% of the variation in the ensemble).

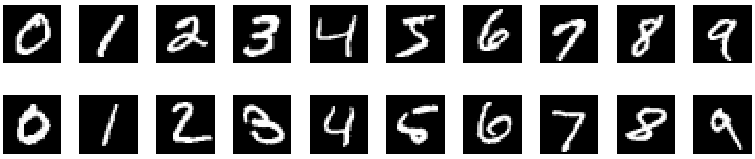

Figure 5: Samples from the MNIST training (upper line) and test sets (lower line).

The ensemble were then encoded on the PCA and a set of 10 EBP classifiers built, each with one target digit labelled '1' and all others labelled '-1'; this was performed with both linear and power-2 polynomial kernels. Suitability of the individual nodes of the models was assessed by automatic relevance detection [6, 5]. Performance was then measured by assessing the accuracy with which the test set could be classified as the target-digit or another; these values were combined by calculating the area under the receiver operating characteristic curves of each classifier. Although the classifier is bias-free for the training set, this need not be true for test samples. Calculating the hit-rate at equal errors would introduce the notion of a calibration set which is otherwise absent. Hence there is a need to calculate a combined measure. The results are shown in Figure 6; clearly the polynomial classifiers are significantly more accurate than the linear classifiers (means are 0.9669 and 0.8777 respectively). It should be noted that there is no correlation between the accuracies for individual digits for the linear and polynomial classifiers and that the mean training AUCs are 0.9659 and 0.9967 respectively. It should also be noted that this advantage for the polynomial classifier is not solely due to the increased number of dimensions available; the images were re-classified with 230 linear dimensions available (this accounted for 97.5% of the variation in the ensemble). The mean AUC was reduced to 0.9225 for the test data and 0.9824 for the training data, with 214 dimensions extracted.

In addition, it is possible to consider the nature of the parameters included in the classifiers, by counting non-zero θ_j values in the maximum-probability classifier. While the linear classifiers have a mean of 19.9 out of 20 parameters used, and thus shows no-significant sparseness, the polynomial classifiers have a mean of 181 out of 230 parameters used. As is shown in Figure 7, although the square terms are almost always used, the bilinear terms are used significantly less frequently and linear least of all. Clearly in this case, the classifiers depend disproportionately the upon non-linear polynomial terms.

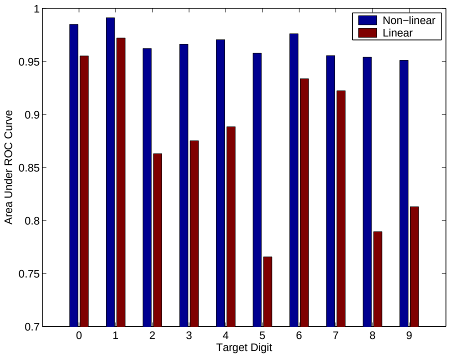

Figure 6: ROC area under the curves for linear and polynomial classifiers for the individual digits.

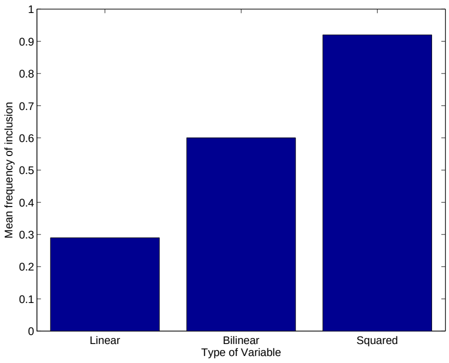

Figure 7: Frequency of inclusion in the non-linear classifiers of squared, bilinear and linear terms.

5 Conclusions

This paper has considered how to perform optimal feature selection for non-linear kernels, within a basis pursuit framework. This was achieved by transforming the piecewise

QP problem into a higher dimensional space, and then performing a primal/dual transformation for both hard and soft classification problems. In both cases, it was shown that feature selection is represented as a set of $2n$ linear inequality constraints. Thus, an NP-complete combinatorial search problem is transformed into a set of linear inequality constraints using basis pursuit. This optimal approach is limited to non-linear kernels with a finite expansion, such as polynomials. The approach was validated on both a test and a real-world data set. When the sparse, quadratic classifier was applied to the MNIST digit-set, the it discriminated target and non-target digits more accurately than linear classifiers with either equivalent numbers of dimensions or proportions of ensemble-variance.

References

[1] P. S. Bradley and O. L. Magasarian. Feature selection via concave minimization and Support Vector Machines. In *International Conference on Machine Learning*, pages 82–90, 1998.

[2] M. Brown, N. P. Costen, and S. Akamatsu. Efficient calculation of the optimal classification set. In *International Conference on Pattern Recognition*, 2004.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing*, 20(1):33–61, 1998.

[4] N. P. Costen and M. Brown. Exploratory sparse models for face classifcation. In *British Machine Vision Conference*, volume 1, pages 13–22, 2003.

[5] N. P. Costen, M. Brown, and S. Akamatsu. Sparse models for gender classification. In *International Conference on Automatic Face and Gesture Recognition*, pages 201–206, 2004.

[6] M. A. T. Figueiredo. Adaptive sparesness using Jeffreys prior. In *Advances in Neural Information Processing System 14*, pages 705–711, 2002.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, pages 1157–1182, March 2003.

[8] Y. LeCun, L. Bottou, Y. Benigo, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[9] S. Schölkopf, B. Mika, S. Burges, C. Knirsch, P. Müller, K. Rätsch, and G. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[10] E. H. Simpson. The interpretation of interaction of contingency tables. *Journal of the Royal Statistical Society, Series B*, 13:238–241, 1951.

[11] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[12] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[13] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm Support Vector Machines. In *Advances in Neural Information Processing Systems 16*. 2004.