Exploring the Set of Sparse, Optimal Classifiers

Martin Brown
Dept Computing and Mathematics, Manchester Metropolitan University, Manchester, UK
m.brown@mmu.ac.uk

Abstract

Feature selection is an important part of classifier design. However, determining an appropriate structure for the empirical classifier is a difficult process that often requires extensive exploration and evaluation. This paper describes a novel approach that represents the feature selection process as a regularization/optimization problem, which has a single global minimum. In addition, a new algorithm is described that allows the designer to efficiently construct the complete family of sparse kernel-based classifiers, and therefore their structures can be explored interactively. This allows the designer to investigate the parameters' trajectories as the regularization parameter is altered and look for effects such as Simpson's paradox that occurs in many multivariate data analysis problems. The approach is demonstrated on the well-known Australian Credit data set.

1. Introduction

Feature/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. This is becoming even more important in recent years, as a large range of different kernel/transformations have been proposed as potential features. 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.

In many cases, the feature selection procedure is not integrated with parameter estimation and may be done in a sub-optimal, approximate fashion. For instance, some form of forwards selection stepwise search may be used to identify the most important features in a computationally efficient, but non-optimal manner. Also, when the selection procedure is performed prior to the parameter estimation, the prediction confidence limits are often under-estimated as the "irrelevant" terms have already been excluded from the model. Alternatively, input relevance hyperparameters have been widely used with neural network models [8,11] and often these are thresholded at some arbitrary level that is difficult to interpret, especially as the learning problem is often ill-conditioned [7]. Finally, designers are not always aware of the subtle interactions, such as the existence of Simpson's Paradox, that occur along the parameters’ trajectories as a classifier becomes increasing complex. Simpson's Paradox occurs when the direction of an association between two variables is reversed when a third variable is controlled [10]. In the context of variable selection, a previously significant variable may become zero, or its sign may be reversed when a new variable enters the model. This obviously limits the qualitative understanding (knowledge extraction) of the sparse model that is one of the primary reasons for constructing sparse models.

This paper describes a novel approach to sparse classification problems. Classification is posed within a regularization framework, where a 2-norm loss function is used to measure the classification accuracy and a 1-norm parameter complexity function is used to control the classifier's sparseness [1,2]. As this regularization function is piecewise differentiable, there exists a single global minimum which can be found using standard Quadratic Programming (QP) algorithms. In addition, the 1-norm parameter complexity function ensures that many of the classifier’s parameters will be zero, hence feature selection is implicitly performed as part of the parameter estimation process. One novel contribution of this work is the derivation of the normal forms for basis pursuit classification. Another novel contribution is the construction of an iterative procedure that estimates the complete set of sparse classifiers in an efficient manner. By starting with the simplest possible classifier where all the parameters are zero, parameters are iteratively introduced in a manner that generates every optimal sparse classifier (and sometimes features are removed) as the regularization parameter is reduced, and thus the classifier’s complexity increases, [2]. The designer can then explore this family of sparse, optimal classifiers in order to assess the sensitivity of choosing a particular structure and looking for effects such as Simpson’s paradox where parameters switch signs when other features are introduced into the model. This procedure is applied to the Australian Credit data set, [9], and the
sparse initial models and Simpson’s paradox are both demonstrated.

2. Sparse classifier design

This section describes a sparse regularization approach for classification problems. A regularization framework is introduced and a simple but naïve method for calculating the model’s parameters is described. This basic framework is then used in the subsequent sections to derive the normal form and then show how the complete family of sparse, optimal classifiers can be calculated in a computationally efficient manner.

The empirical models considered in this paper are linear in their adjustable parameters. This includes a wide range of linear, polynomial and kernel [4] machines:

\[ y = \phi^\top \theta + b \]

where \( \theta \) and \( \phi \) are the \( n \)-dimensional parameter and feature vectors, respectively and \( b \) represents the bias term. In addition, the classifier’s output, \( y \), is thresholded at \(-1\) and \(+1\). [4]. The input features determine the classifier’s capabilities and generalization abilities, and some techniques, such as support vector machines [4], assume flexible kernels are available and use regularization methods to control complexity. A training set of the form \( \{ \Phi, t \} \) is assumed to exist where the exemplar input data, \( \Phi \), is \( l \times n \), and each feature is typically scaled to be zero mean and unit variance (see Section 3.4) and the classification target vector, \( t \), is an \( l \)-dimensional bipolar vector as each element is either \(-1\) or \(+1\).

2.1 A regularization approach to sparse classification

A lot of research/application in statistical modelling has taken place in recent years, motivated by Vapnik’s work on Statistical Learning Theory [4]. Using this approach, one way of formulating a classification regularization function is given by:

\[ \min_{\theta, \zeta, \lambda} f(\theta, b, \zeta, \lambda) = \frac{1}{2} \| \Phi \theta + b \|_2^2 + \lambda \| \zeta \|_1 \]

(1)

where \( \zeta \) and \( \lambda \) are the residual vector and the regularization parameter, respectively. The non-negative output residual vector is defined by:

\[ \text{diag}(t)(\Phi \theta + b) + \zeta \geq 1 \]

\[ \zeta \geq 0 \]

(2)

and the set of active data points corresponds to those output residuals that are strictly positive and hence lie in or on the wrong side of the classification margin. By optimizing the classifier’s parameters, the aim is to maximize the so-called classification margin and thus ensure that the classifier provides a graded fit to the training data. In addition, it should also be noted that this is a piecewise QP problem in \( n + l + 1 \) parameters as the output residuals are a thresholded linear function of the classifier’s active parameters.

In this regularization approach to machine learning, the loss function, \( l(\xi) = \frac{1}{2} \| \xi \|_2^2 \), which measures the data fitting error is balanced against a model complexity function, \( p(\theta) = \| \theta \|_1 \), and the optimal parameters are determined by the regularization parameter, \( \lambda \). A small value of \( \lambda \) will ensure that greater weight is given to minimizing the output errors, at the expense of a model with a larger parameter vector. A large value of \( \lambda \) ensures that more importance is given to smaller parameter vectors although the output errors are larger.

It is important to note that the model complexity term is a form of Bayesian prior. Normal ridge regression techniques and support vector machines usually assume that the model complexity is measured with respect to a 2-norm [4]. However, basis pursuit classification, or robust classification, [1,7] refers to the fact that a 1-norm is used to measure the parameter vector’s size. Most \( p \)-norms will produce parameter values that are small, but non-zero. It can be shown, [1], that using a 1-norm produces sparse solutions while still producing a continuous optimization problem, as will be demonstrated in the next section. In fact, \( p=1 \) is the only \( p \)-norm (\( p \geq 1 \)) which produces a truly sparse solution without some form of thresholding procedure and has been developed by Chen and Efron et al for regression algorithms [3,5].

2.2 Direct calculation of a sparse classifier

Before focusing on the problem of producing an algorithm that iteratively constructs the complete family of optimal, sparse classifiers, it is worthwhile considering how this problem can be solved for a particular value of the regularization parameter \( \lambda \). The sparse regularization optimality criterion is a piecewise QP problem. A well-known method [3] to overcome this non-linearity is to introduce a set of \( 2n \) parameters, \( z \), where the first \( n \) values are the corresponding values of \( \theta \) when \( \theta \) is positive (and zero otherwise) and the second \( n \) values are \(-\theta\) when \( \theta \) is negative (and zero otherwise). This allows the original cost function to be expressed in terms of \( z \):

\[ \min_{\lambda, \theta, \zeta, \lambda} f(z, b, \zeta, \lambda) = \frac{1}{2} \sum z_i^2 + \lambda \sum \zeta_i \]

\[ \zeta \geq 0 \]

\[ \text{diag}(t)(\Phi \theta + b) + \zeta \geq 1 \]

\[ \zeta \geq 0 \]

(3)

which is now a standard QP problem in \( 2n+l+1 \) variables. While this approach is conceptually simple, it is not necessarily very efficient, as it does not directly exploit any of the problem’s particular features. More importantly, in order to explore neighbouring classifiers in order to explore how sensitive the calculated parameters are with respect to the regularization.
parameters, the QP algorithm would need to be recalculated (possibly starting from the previous solution). This would not exploit any knowledge about the problem’s structure and may miss out important parts of the solution where the active sets of data and parameters are altered.

The remainder of this paper is aimed at understanding the properties of the sparse classifier associated with basis pursuit classification and investigating how they can be exploited to produce efficient algorithms for feature selection.

3. Geometric analysis of optimality conditions

In this section, the properties of the optimal, sparse basis pursuit classifiers are examined and their normal form is derived. This shows that basis pursuit regularization introduces a dead-zone around inner product between the feature vectors and residual signal of width $\lambda$. In addition, an important interpretation of this normal form is that only the active parameters lie on the boundary of this dead-zone, while the inactive parameters lie strictly inside it. It also shows that along the optimal parameter curve, their values are a piecewise linear function of the regularization parameter $\lambda$. Finally, a set of necessary and sufficient constraints are derived that determine when both parameters and data points become active and inactive. These are fundamental insights for formulating the iterative classifier design algorithms in Section 4.

3.1. Global properties of optimal classifiers

A number of properties can be easily established about the classification regularization function $f(\theta)$ in (1). It is a continuous function of all its parameters and is piecewise differentiable with respect to the model’s parameters, due to the presence of the 1-norm model complexity function. The regularization function and the model’s parameters are parameterized by the regularization parameter $\lambda$, where $\theta = \arg \min_\theta f(0, b, \xi, \infty)$ represents the first boundary condition and the maximum likelihood solution, $\theta = \arg \min_\theta f(0, b, \xi, 0)$, represents the second. Therefore, the “path” of optimal sparse classifiers traces out a continuous curve between these two points.

These comments hold for any $p$-norm where $p \geq 1$, therefore the only difference is the actual path traced through parameter space as $\lambda$ varies between $\infty$ and 0. When $p=2$, the curve is a weighted quadratic where every parameter is (nearly) always non-zero, hence the classifiers are not sparse. In this section, conditions are derived which explain why sparse classifiers are produced when $p=1$ as many of the classifier’s parameters will be zero.

3.2. Basis pursuit normal form

In the following sections, an active parameter or data point refers to those values that are actively considered in the calculation. Therefore, the active parameter set is the set of parameters whose values are non-zero. For the data points, it refers to those data points lying in or on the wrong side of the classification margin and have a positive output error, $\xi$.

Therefore, assuming that the active set of parameters and data do not locally change, by differentiating (1) with respect to the active parameters and setting equal to zero to find the minima, the gradients of the loss and model complexity functions are related as:

$$\Phi^T(t - \Phi \theta - l_b) = \lambda \operatorname{sgn}(0)$$

(4)

For the non-active parameters, the derivative is zero and they do not become non-zero as the regularization parameter $\lambda$ is altered. This provides the bound across both active and in-active parameters:

$$|\phi|_1^T \leq \lambda$$

(5)

This can be regarded as the “normal form” for basis pursuit regression and classification. This bound is tight for the set of active parameters and loose for the inactive parameters. An important interpretation of basis pursuit is that it maximally reduces an infinity norm on the transformed residual signal by introducing enough non-zero parameters into the model so that the largest transformed residuals can all be reduced equally/controlled within the active parameter sub-space.

When $\lambda$ is large, all the parameters are zero and the classification margin is infinite and the bias term is simply the classification priors. Then the first parameter that becomes active occurs when $\lambda$ is reduced to:

$$\max |\phi^T(t - l_b)| = \lambda$$

i.e. the absolute value of the inner product between the feature vector and the residual signal is equal to the regularization parameter. It is this interpretation that provides a set of linear inequality constraints that determine when features/parameters become active.

3.3. Piecewise linear parameter trajectories

Rearranging (4) in the active parameter sub-space gives:

$$\theta = H_1^{-1} \Phi^T t_1 - \lambda \operatorname{sgn}(0)$$

(6)

where $H_1 = \Phi^T (\Phi^T \Phi)^{-1} \Phi$ is the local, active Hessian which has had the bias term subtracted and $t_1 = (I-11^T/l_\theta) \Phi$ represents the target vector minus the bias. This expression is “nearly” a direct, closed form solution for the optimal sparse parameter vector. When the sign of the optimal parameter vector has been identified, the solution is direct, however estimating the sign of the
parameters which lie in \{-1, 0, 1\} is equivalent to deciding which variables are included in a classifier’s design, which in general, is a non-trivial problem. However, it does show that, within the active parameter sub-space, the optimal parameters are a linear function of \(\lambda\) and therefore lie on a unique line. It also shows that the local optimal parameters are oriented towards the maximum likelihood solution, \(\theta = H^{-1}_1 \Phi^T t_1\), in the local active sub-space. The orientation also depends on the inverse Hessian (as with conventional ridge regression) but is the identity matrix if the kernels are orthonormal. A very simple example is shown in Figure 1 to illustrate the piecewise nature of the parameter trajectory for a 2-dimensional model without a bias term. Note the two line segments that occur along the axes correspond to sets of sparse classifiers.

**Figure 1.** An illustration of the piecewise linear optimal parameter curve as the regularization parameter \(\lambda\) is altered from \(\infty\) to 0. The contours of the quadratic loss function are shown.

To show how the bias term is removed from (6), it should be noted that the bias term is not included in the model complexity function, therefore, it can be calculated separately as the value that minimises the squared output residual across the active data set as:

\[
b = \mu_r(\zeta) = \mathbf{1}^T (\mathbf{t} - \Phi \theta)/(\sigma^2)
\]  

(7)

This is why the Hessian matrix and target vector are modified in (6).

### 3.4. Constraints on the parameters

The optimality conditions described in Sections 3.2 and 3.3 must be satisfied in both the active parameter and data sub-spaces. When \(\lambda\) is large, all the data points are active and none of the parameters are active and the bias term is simply the average value of the target values. As \(\lambda\) is reduced, some of the parameters become active and some of the data points become inactive, as they become correctly classified, although this is not necessarily a monotonic phenomenon. This is illustrated in Figure 1, where the curve of optimal, sparse models starts at \(\theta = \mathbf{0}\), where both parameters are inactive. As \(\lambda\) is reduced, \(\theta_i\) initially becomes active, then \(\theta\) becomes active, then \(\theta_2\) becomes inactive and finally both are active when \(\lambda \rightarrow 0\), close to the maximum likelihood solution.

These constraints on whether parameters and data points are active or inactive define the “knots” in the piecewise linear parameter paths, illustrated in Figure 1, as they determine which rows and columns of \(\Phi\) are involved in (6).

Initially, just consider the constraints on the active and inactive parameters, as the constraints on the data points will be considered next. There are two cases to consider:

- An active parameter becoming inactive.
- An inactive parameter becoming active.

For most regression and classification problems, both sets of constraints have to be considered.

Consider the first case when the optimal active set has already been identified and the regularization parameter is altered slightly (either increased or decreased). Then (6) is valid as long as:

\[
\text{sgn}(\theta) = \text{sgn}(\theta(0))
\]

(8)

where \(\theta(0)\) is the existing optimal parameter values within the active sub-space. This provides a set of \(n_l\) necessary range constraints on the active parameters. When any of these constraints are violated by one of the active parameters becoming zero, that parameter becomes inactive and will be dropped from the calculation. This is illustrated in Figure 1 when \(\theta_2\) is initially negative and then becomes zero.

In addition, consider the constraints on the inactive parameters. From (5), the inactive transformed residuals must be less than \(\lambda\) in magnitude. When the transformed residual is equal to \(\lambda\) in magnitude, the parameter will become a member of the active set and its value can be directly calculated from (6). Therefore, there is a set of \(2n_l\) functional linear constraints on the inactive parameter set of the form:

\[
-\lambda I \leq \Phi_j^T \zeta \leq \lambda I
\]

(9)

where \(\Phi_j\) is the matrix of inactive features. When a constraint is violated, the previously inactive parameter becomes active.

Therefore, there is a set of \(n+n_l\) linear inequality constraints on the parameter vector, all of which must be satisfied when the optimal parameter values are calculated using (6).

The continuity constraint, which occurs when a parameter becomes active or inactive:

\[
\lambda = -\frac{\Phi_j^T \zeta}{\text{sgn}(\theta)}
\]

can be interpreted as selecting that parameter into the model (becoming non-zero) whose feature is maximally correlated with the current residual signal. This is the condition that \(\lambda\) must satisfy when the first parameter is
introduced into the classifier and also determines the sign of the parameter when it first becomes active. It illustrates why the features should all have a common scaling (unity) when they are of equal prior importance otherwise their order of selection will be influenced by the magnitudes of the features (although this can be used to specify priors for the individual features).

In summary, each parameter in the classifier has an associated linear inequality constraint that determines when it either becomes or ceases being active. As long as the parameter constraints are satisfied, it is possible to search the space of sparse classifiers in an optimal manner using the direct, linear calculation in (6).

3.5. Constraints on the data points

Just as an active parameter has a non-zero value, an active data point has a positive residual defined by (2). Therefore the active data points either lie in the classification margin or on the wrong side of it. In addition, an active data point can become inactive or an inactive data point can become active, and in this section a set of linear inequality constraints for the \( l \) data points will be derived. When a data point becomes active or inactive, the effect on the calculation of the optimal parameters is a “knot” in the piecewise linear curve. This is because the feature matrix, \( \Phi \), will either have a column dropped or included and this will also change the inverse Hessian matrix in (6).

For basis pursuit classification, an active data point lies inside, or on the wrong side of, the margin. Therefore, a data point no longer affects the solution when:

\[
t_{j}(\Phi^{T} \theta + b) > 1
\]

as the corresponding residual, \( \zeta_j \), is zero. Initially, when \( \theta = \infty \), all the parameters are identically zero, the bias term represents the class priors and all the data points are active because the margin is infinite.

Using (10), the set of active data points must therefore satisfy:

\[
diag(t) \Phi \theta \leq 1 - t \mu_a(t)
\]

where \( \mu_a(t) \) represents the mean of the target vector for the current active data set and the transformed features are being used to remove the effect of the bias term. This gives a set of \( l_a \) constraints, one for each active data point.

A similar set of constraints occurs for the inactive data points. The active data points do not become inactive in a monotonic fashion. A point may become inactive and then return to the active set. The linear inequality constraints on the inactive data points are:

\[
diag(t) \Phi \theta \leq -1 + t \mu_i(t)
\]

which are simply the negated coefficients for the active data points. This gives a set of \( l_i \) constraints, one for each inactive data point.

Therefore, there is a set of \( l \) inequality constraints, one for each data point and overall, the parameter and data constraints specify necessary and sufficient conditions for an optimal basis pursuit solution.

4. Calculating the complete set of sparse classifiers

In this section, a new procedure is described for calculating the complete set of optimal, sparse classifiers. It allows a designer to view the complete parameter trajectories as the regularization parameter is changed and understand how features get introduced into classifier as well as exploring their interaction. This provides the designer with important knowledge about the stability of the feature selection process. An equivalent procedure for regression has also recently been proposed [5].

The proposed procedure works in a forwards selection-type manner. By starting off with the simplest possible classifier, features are introduced into the active set and data points are dropped from the active set as the classification margin shrinks because the regularization parameter is reduced. The actual path the parameters take is piecewise linear and the knot values occur when the sets of active parameters or data points change as the regularization parameter is reduced.

The remaining sparse models are then obtained by linearly interpolating between these values. As described in Section 3.3, the first knot in the optimal parameter set is \( \theta = 0 \), and the locally active parameter sub-space is given by the kernel that maximizes \( \lambda^{T} \Phi \).

Given that \( \theta(0) \) is a knot value in the current active parameter sub-space, then from Section 3.3, the local change in the (active) parameters is given by:

\[
\theta(\mu) = \theta(0) + \mu H^{-1} \text{sgn}(0)
\]

where \( \mu \) is the step size and the update direction is \( s = H^{-1} \text{sgn}(0) \). This is a linear function of \( \mu \) and the aim is to step along this path as long as all the parameter and data constraints, derived in Sections 3.4 and 3.5, are valid. As soon as one of the constraints becomes active, the path direction will change as a parameter or data point will be either introduced to or dropped from the calculation.

4.1. Classification design using iterative linear programming

This “shooting algorithm” generates a sequence of Linear Programming (LP) sub-problems where the aim is to iteratively step along the piecewise linear path. It will terminate when the regularization parameter is reduced to zero and the parameters correspond to their maximum likelihood values. In addition, when the problem is reasonably well conditioned, the number of iterations
will be approximately $n+l$. However, it should be noted that even when the problem is badly conditioned, the proposed procedure will find every optimal, sparse classifier.

Each knot value can now be iteratively calculated using the sequence of LP sub-problems in terms of the unknown step length $\mu$:

$$\min \ - \mu$$

$$\text{st} \ \mu \leq 1$$

$$\mu(-\text{sgn}(\theta(0)).*s) \leq \theta(0)$$

$$\mu((1\Theta_j \text{sgn}(\Theta_j) - \Phi_j^I) Ds) \leq (1\Theta_j \text{sgn}(\Theta_j) - \Phi_j^I)(t - \Phi(0))$$

$$\mu((1\Theta_j \text{sgn}(\Theta_j) + \Phi_j^I) Ds) \leq (1\Theta_j \text{sgn}(\Theta_j) + \Phi_j^I)(t - \Phi(0))$$

$$\mu(\text{diag}(t)s) \leq 1 - t\mu_a(t) - \text{diag}(t)\Phi(0)$$

The first constraint is a single bound of the form $\lambda \geq 0$. The next three constraints deal with the active/inactive parameter set. The last two constraints are concerned with the active/inactive data set. It should be noted that some subscripts that have been suppressed for simplicity. While this LP problem may appear quite complex, it is important to note that this is simply a set of $n+n+l+1$ constraints on a single variable of the form $\mu \leq b$, where only non-negative values of $\mu$ are acceptable. Therefore, once the coefficient vectors have been determined, calculating the next knot value is simply the case of finding the minimum valid value. In fact, although this is posed as a LP sub-problem because of its simple form there is no need to use a LP sub-routine to solve it and efficient, rank 1, inverse Hessian updates can be used to efficiently re-calculate the step direction.

5. Australian credit approval data set

The Australian Credit data set is a real-world data set that has been used by Quinlan and Statlog [9] to validate a wide number of machine learning, statistical and neural classification algorithms. In the Statlog study, the linear classification rules were ranked highly, thus demonstrating that there was little non-linearity in the Bayes discriminant. The data set consisted of 690 observations about whether or not credit should be granted and there were 14 potential attributes - 6 continuous and 8 categorical. The categorical variables ranged from simple binary indicators to 14 separate values. The categorical variables are encoded using a “1 of C-1” mapping which produces a potential set of 35 features. Note that the bias term is implicitly calculated, using (7), and not included in this variable count, although its trajectory is shown in Figure 3 and is the only non-zero parameter when $\lambda$ is large as it represents the class priors.

The error curve and parameter trajectory for a linear model are shown in Figures 2 and 3, respectively. Here the values are plotted against $\log(\lambda+1)$ as there is a lot of activity when $\lambda$ is close to zero. As can be clearly seen, for a large range of sparse models, the error rate is approximately 100 and there is a single dominant parameter. For small value of $\lambda$, the number of classification errors goes down from 100 to around 80, however this is at the expense of over 30 variables being introduced into the model and their values increasing significantly, so their relevance is questionable. When $\lambda$ was small, a number of parameters started to reverse the sign as the regularization parameter was reduced. In practice, approximately 10 of the 35 parameters were subject to this effect and the behaviour of one such parameter is shown in Figure 4. It was interesting to note that this example of Simpson’s paradox was largely restricted to the parameters associated with the two categorical input variables which a large number of values. In addition, it only occurred when the number of active data points was becoming small due to the margin decreasing in size. It will be instructive to identify how prevalent Simpson’s paradox is on other classification problems and whether it is largely restricted to categorical inputs. It is important because it limits the qualitative interpretation of the parameter values.

The advantage of using an approach such as this is that the properties for all the sparse models can be directly visualized and the behaviour of the parameters discussed. In addition, it was found that the approach was efficient, compared to re-training a single classifier, even though the approach generates every sparse classifier. However, in order to select a particular model, two critical points can be identified. When the first parameter is introduced into the classifier, the number of errors is substantially reduced to 100 and this is for a classifier that just contains one input. Alternatively, if a more complex classifier is required, when $\log(\lambda+1) = 1$, the number of errors have been reduced to approximately 80, and this is before the onset of the effects of Simpson’s paradox. The parameter values are reasonably stable. So this may be considered an alternative classifier.

![Figure 2. Number of classification errors verses the transformed regularization parameter, $\log(\lambda+1)$. As the model becomes more complex and $\lambda$ decreases, the number of active data points is becoming small due to the margin decreasing in size.](image)
the number of errors on the training set decreases, but not monotonically. There is a large reduction when a single parameter/feature enters the classifier.

Figure 3. A plot of the parameter trajectories verses the transformed regularization parameter, $\log(\lambda+1)$. Initially, a single feature/parameter dramatically decreases the number of classification errors, and the bias term reflects the non-zero class priors for the active data set.

Figure 4. A plot of $\theta_j$ verses $\log(\lambda+1)$. When the parameter enters the classifier, it is initially positive, however when other features entre the classifier, it becomes negative.

6. Conclusions

A novel and efficient iterative method for calculating the complete set of sparse classifiers has been proposed. It provides designers with more information about the sensitivity of the problem as they can immediately visualize how the parameters change as the regularization parameter is altered from infinity to zero. One advantage of using a basis pursuit classification framework is that there exists a global minimum when calculating the sparse feature set. In addition, when the regularization parameter is altered slightly, at most one of the classification parameter will change state. This fully supports a forwards selection (or alternatively backwards elimination) approach to optimal feature selection. The one major difference between using basis pursuit for regression or classification problems is the extra set of constraints imposed by the output thresholding for classification problems that determine the active set of data points. An important part of this work is to explore parameter interaction and the selection stability and an example of this was seen in the Australian credit data set when parameter values can change sign, and hence interpretation because the model complexity is altered slightly.

Future work will investigate robust and efficient implementations of the iterative LP sub-problems, estimating the number of segments/knots in the set of sparse models, specifying non-uniform priors as well as investigating how the technique can be used for feature selection with non-linear kernels and developing error bar estimates for the sparse regularization framework.

7. References