AN OPTIMAL BASIS FOR FEATURE EXTRACTION WITH SUPPORT VECTOR MACHINE CLASSIFICATION USING THE RADIUS–MARGIN BOUND

J. Fortuna, D. Capson

Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada, L8S 4K1
email: jeff@grads.ece.mcmaster.ca

ABSTRACT

A method is presented for deriving an optimal basis for features classified with a support vector machine. The method is based on minimizing the leave-one-out error which is approximated by the radius–margin bound. A gradient descent method provides a learning rule for the basis in an outer loop of an iteration. The inner loop performs support vector machine training and provides support vector coefficients on which the gradient descent depends. In this way, the derivation of a basis for feature extraction and the support vector machine are jointly optimized. The efficacy of the method is illustrated with examples from multi-dimensional synthetic data sets.

1. INTRODUCTION

When data from multi-dimensional measurements is represented as a feature vector, the feature space of the raw data often has a very large dimension. It is usually prudent to re-represent the original measurements in more compact form (a shorter feature vector). The process of selecting features from the raw measurements is one of feature selection or feature extraction. It is, in general, a problem of dimensionality reduction. The ultimate goal of feature selection/extraction is to find the minimum number of features required to capture the essential structure in the raw data. This minimum number of features is termed the intrinsic dimensionality of the data. This dimensionality reduction is accomplished by applying a transformation (linear or non-linear) to the the input data.

Linear statistical techniques, such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) have been used to reduce the dimensionality of the data [1]. The underlying hypothesis is that the transformed feature vectors have a dimensionality approaching the intrinsic dimensionality of the data. The unsupervised feature extraction process is often sub-optimal with respect to any specific problem, since it only uses statistical information about the input data and there is often much more information available. In the context of pattern recognition, information about the a-priori classification of some input data examples is often available. As such, some form of supervised learning can be employed to jointly design the feature extractor and the classifier.

Recently, there has been considerable effort made in selecting features (performing dimensionality reduction) which are most relevant to a support vector machine (SVM). In [2] and [3] a wrapper model [4] is used, where a SVM is used as an inner loop of a feature selection problem. Irrelevant features are selected based on the fact that the support vectors are the only data points that are relevant for the determination of the separating hyperplane. Another related possibility is to scale the features by ranking their relevance to the SVM. This process was detailed in [5] and used in [6] to scale expression masks for facial expression recognition. In a Bayesian framework, joint feature selection and classification was performed by an expectation maximization algorithm in [7]. Both feature selection and feature scaling assumes that some features have been extracted a-priori by some other technique, or the raw input data was used. In [8] an iterative algorithm was employed to move support vectors towards the class mean and learn a new basis from regression on the modified features.

The method presented herein jointly extracts features and performs SVM classification using a wrapper methodology inspired by techniques to choose SVM parameters. In [5] and [9], kernel parameters such as C for a soft margin classifier and σ for a Gaussian radial basis function are optimized based on bounds which approximate the leave-one-out error. In this paper, the kernel is assumed to depend on the basis in which the data is represented. The feature extraction model is a linear one, where the basis is decomposed into a unitary transformation for the purposes of dimensionality reduction, and a non-orthogonal transformation, which is to be learned from the gradient method in [5]. The radius–margin bound is used to provide an upper bound on the leave-one-out error. The effectiveness of the learned basis is shown experimentally in the classification of some synthetic data sets. It is shown that the learned basis provides much lower error rates and increased margin over a PCA basis.

2. AN OPTIMAL BASIS REPRESENTATION

Linear subspace representations of a data set are defined by the model:

$$\mathbf{y} = \mathbf{W}^T\mathbf{x} \tag{1}$$

with n-dimensional data expressed as a random vector $\mathbf{x} \in \mathbb{R}^n$. In this model, a set of features $\mathbf{y}$ are extracted from the data by a linear transform $\mathbf{W}$. For the purposes of dimensionality reduction, $\mathbf{W} \in \mathbb{R}^{n\times m}$ with $m << n$. This is most often performed with PCA, which provides an orthogonal $\mathbf{W}$. This representation can be extended to allow for the representation of non-orthogonal directions in the data by applying another transformation $\mathbf{S} \in \mathbb{R}^{m\times m}$ so that:

$$\mathbf{z} = \mathbf{S}^T\mathbf{W}^T\mathbf{x}. \tag{2}$$

$\mathbf{S}$ can be found from minimizing the radius–margin bound of a support vector machine, as will be shown below.

This work was supported by the Canadian Networks of Centers of Excellence Program (IRIS), the Natural Sciences and Engineering Research Council (NSERC) and the McMaster Manufacturing Research Institute (MMRI)

2.1. Support Vector Machines

With the SVM, a labeled set of feature vectors $\mathbf{x}_i, Y_i$ is constructed for all l features in the training data set. The class of feature $\mathbf{x}_i$ is defined by $Y = \{1, -1\}$. The optimal separating hyperplane $\mathbf{w}$ for the features can be found with:

$$\begin{aligned} \min \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.} \quad & Y_i(\mathbf{w}\Phi(\mathbf{x}) + b) \geq 1 \quad \forall i, \end{aligned} \tag{3}$$

with $\mathbf{\Phi}$ a non-linear function. The dual expression to maximize for a non-linear hard margin SVM can be written as [10]:

$$\begin{aligned} \max \quad & L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j Y_i Y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{l} \alpha_i Y_i = 0 \quad \text{and} \quad \alpha_i \geq 0 \quad \forall i. \end{aligned} \tag{4}$$

$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x},)\Phi(\mathbf{x}) \rangle$ is a kernel function satisfying Mercer's conditions. An example kernel function (the one used herein) is the Gaussian radial basis function:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2} \right) \tag{5}$$

where σ is the standard deviation of the kernel's exponential function.

2.2. Radius–Margin Bound

Vapnik [10] proposed an upper bound on the number of errors with the leave-one-out method of training:

$$T = R^2 ||\mathbf{w}||^2 \tag{6}$$

where $\frac{1}{||\mathbf{w}||}$ is the ℓ_2 distance from the hyperplane to the closest data point and the images of the training data fall within a sphere of radius R.

If the bound $T(\alpha, \theta)$ is dependent on parameters θ for which it is to be minimized, the total derivative with respect to each parameter θ_p is [5]:

$$\frac{\partial T}{\partial \theta_p} = \left. \frac{\partial T}{\partial \theta_p} \right|_{\alpha \text{ fixed}} + \frac{\partial T}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_p}. \tag{7}$$

In [5] it was noted that at optimality, $\frac{\partial T}{\partial \alpha} = 0$, so T can be differentiated directly with respect to the parameters.

The derivative of the radius–margin bound, with respect to the parameters is:

$$\frac{\partial T}{\partial \theta_p} = \frac{\partial R^2}{\partial \theta_p} ||\mathbf{w}||^2 + \frac{\partial ||\mathbf{w}||^2}{\partial \theta_p} R^2. \tag{8}$$

The evaluation of these partials was performed in [5] wherein:

$$\frac{\partial ||\mathbf{w}||^2}{\partial \theta_p} = -\sum_{i,j=1}^{l} \alpha_i \alpha_j Y_i Y_j \frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \theta_p} \tag{9}$$

and

$$\frac{\partial R^2}{\partial \theta_p} = \sum_{i,j=1}^{l} \beta_i \frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \theta_p} - \beta_i \beta_j \frac{\partial K(\mathbf{x}_i, \mathbf{x}_j)}{\partial \theta_p}. \tag{10}$$

2.3. Gradient Step for an Optimal Basis

Using the model in Equation (2), the radius–margin bound can be used to find a gradient step which will converge on an optimal $\mathbf{S}$ such that the leave-one-out training error will be minimized. To do this, the derivative of the margin and the radius must be found with respect to the parameters, which are the elements of the transformation matrix $\mathbf{S}$. Formally, for the derivative of the margin is:

$$\frac{\partial ||\mathbf{w}||^2}{\partial \mathbf{S}} = -\sum_{i,j=1}^{l} \alpha_i \alpha_j Y_i Y_j \frac{\partial K(\mathbf{S}^T \mathbf{y}_i, \mathbf{S}^T \mathbf{y}_i)}{\partial \mathbf{S}}, \tag{11}$$

where $\mathbf{y}$ is the data under the transform in Equation (1). For a Gaussian kernel and an ℓ_2 norm:

$$K = \exp^{-\left({\mathbf{y}_i}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_i - 2{\mathbf{y}_i}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_j + {\mathbf{y}_j}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_j \right)}. \tag{12}$$

This definition implies the condition that the matrix $\mathbf{S}\mathbf{S}^T$ is positive definite. This implication will be discussed further in Section 4. The derivative of the margin can be written as:

$$\frac{\partial ||\mathbf{w}||^2}{\partial \mathbf{S}} = -\sum_{i,j=1}^{l} \alpha_i \alpha_j Y_i Y_j f \frac{\partial g}{\partial \mathbf{S}}, \tag{13}$$

where f and g are defined as:

$$\begin{aligned} f &= -K(\mathbf{S}^T \mathbf{y}_i, \mathbf{S}^T \mathbf{y}_j) \\ g &= -({\mathbf{y}_i}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_i - 2{\mathbf{y}_i}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_j + {\mathbf{y}_j}^T \mathbf{S}\mathbf{S}^T \mathbf{y}_j). \end{aligned} \tag{14}$$

The partial of g is the derivative of a scalar function with respect to a matrix $\mathbf{S}$. As such, the derivative is a matrix where each element is the derivative with respect to each element of $\mathbf{S}$. Thus the derivative of the margin is:

$$\frac{\partial ||\mathbf{w}||^2}{\partial \mathbf{S}} = -\sum_{i,j=1}^{l} \alpha_i \alpha_j Y_i Y_j f \mathbf{A}\mathbf{S} \tag{15}$$

where the matrix $\mathbf{A} = \mathbf{y}_i {\mathbf{y}_i}^T - 2\mathbf{y}_i {\mathbf{y}_j}^T + \mathbf{y}_i {\mathbf{y}_i}^T$. In the same way, the expression for the derivative of the radius function is:

$$\frac{\partial R^2}{\partial \theta_p} = \sum_{i,j=1}^{l} \beta_i f \mathbf{A}\mathbf{S}. \tag{16}$$

The gradient update rule for $\mathbf{S}$ is:

$$\mathbf{S} \leftarrow \mathbf{S} - \epsilon \left[\frac{\partial R^2}{\partial \theta_p} ||\mathbf{w}||^2 + \frac{\partial ||\mathbf{w}||^2}{\partial \theta_p} R^2 \right], \tag{17}$$

where the derivatives are shown in Equation (15) and Equation (16) respectively. $||\mathbf{w}||^2$ and α are obtained from the SVM which is updated at each iteration. Thus the following iterative procedure [5] is employed:

1. Initialize $\mathbf{S}$ to a random matrix
2. Orthogonalize $\mathbf{S}$ with $\mathbf{S} \leftarrow (\mathbf{S}\mathbf{S}^T)^{-\frac{1}{2}}\mathbf{S}$
3. Employ a standard SVM to determine α and $||\mathbf{w}||^2$
4. Update $\mathbf{S}$ with the rule in Equation (17)
5. Return to step 3 or terminate if T in Equation (6) is minimum

To simplify the update rule, no constraint was placed on $\mathbf{S}$ although as mentioned previously, $\mathbf{S}\mathbf{S}^T$ must be positive definite. This

was dealt with in two ways. First, in Step 2 of the iteration, $\mathbf{SS}^T$ initialized to be positive definite. In fact, $\mathbf{SS}^T = \mathbf{I}$. However, at a point very near the minimum, $\mathbf{SS}^T$ will become singular. In the iteration, this is detected by an increase in the bound, and provides the termination condition. Alternatively, a zero eigenvalue in $\mathbf{SS}^T$ could be used as the termination condition. In the implementation chosen for this paper, a conjugate gradient method was used instead of the aforementioned update rule based on steepest descent, to speed up convergence.

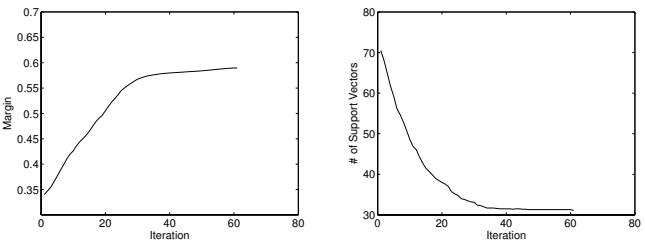
3. RESULTS

Four multi-dimensional datasets were used to test the algorithm. They are generated using PRTools4, a Matlab toolbox for pattern recognition [11]. Each dataset was used to randomly generate 50 multi-dimensional training points according to the dataset distribution. The Difficult dataset is defined as multi-dimensional and a dimensionality of 30 was used. The other 4 datasets are defined as two dimensional. To increase their dimensionality, the first 50 dimensions were created by random instances of the first dimension of the two-dimensional dataset. The second 50 dimensions were created by random instances of the second dimension. Although this is an artificial multi-dimensional distribution, provided that the variance of the distributions is large enough, the random instances generate sufficient complexity. For all 4 datasets, the dimensionality of the data was reduced to 10 with PCA before SVM classification. As such, the non-orthogonal matrix $\mathbf{S} \in \mathbb{R}^{10\times 10}$. For the RBF kernel, the standard deviation was 10, 100, 5 and 20 for the Difficult, Banana, Highleyman and Lithuanian datasets respectively. The SVM parameter C was set to 1 for all experiments. The kernel standard deviation and C parameters were chosen empirically for reasonable classification performance and were not optimized in any way.

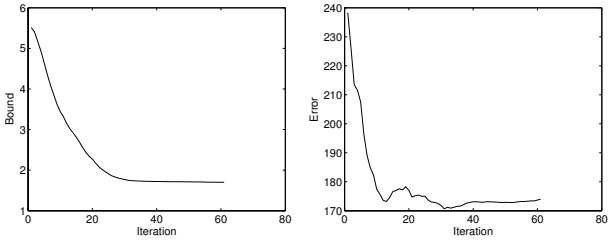
The average (over 10 independent runs of the experiment) bound and error is shown at each iteration for each of the datasets in Figs. 1 – 4. The average margin and number of support vectors for the Difficult dataset is also shown in Fig. 1. Although the algorithm terminated after a different number of iterations for each experimental run, the termination value was extended to the longest of the 10 runs for the purpose of averaging. Fig. 5 illustrates an example experimental run on the Lithuanian dataset without the termination condition. It is important to note that the bound begins to grow after the smallest eigenvalue reaches a zero value. This illustrates the behavior of the gradient descent after the constraint on the positive definiteness of $\mathbf{SS}^T$ is violated.

4. DISCUSSION

Table 1 shows that an increased margin, a decreased number of support vectors, a decreased Radius–Margin bound and a decreased error can be expected from the minimization of the bound. At the 95% confidence level, only one of the results missed significance (error for the Banana dataset). This is similar to results reported by Chapelle et. al. in [5] when feature vectors were scaled directly for a handwritten digit recognition experiment. Additionally, the evolution of the test error and bound reported herein in Figs. 1 – 4 were similar to those reported in [5]. The experiment in this paper has the additional advantage of a significant reduction in dimensionality from the linear transform of the data. Convergence occurred before approximately 40 iterations in all cases. From Fig. 5 it is clear that it is unnecessary to include the positive definite constraint on $\mathbf{SS}^T$ since the solution is part of the feasible set until a value close to the minimum bound is reached. The Difficult dataset, as was expected, was the most difficult to classify, followed closely by the Banana and the Lithuanian datasets. The algorithm, however, performed equally well on all of the datasets, providing anywhere from a 20% to 90% increase in margin, a 25% to 50% decrease in the number of support vectors, a 50% to 70% decrease in the Radius–Margin bound, and up to a 30% decrease in error.

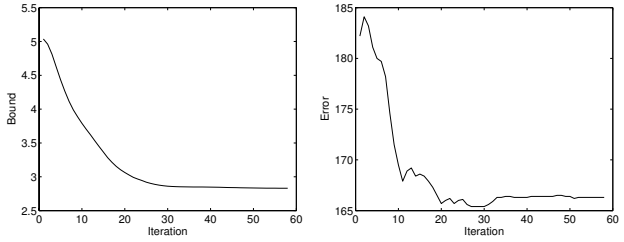


(a) Avg. Margin (b) Avg. # of SVs



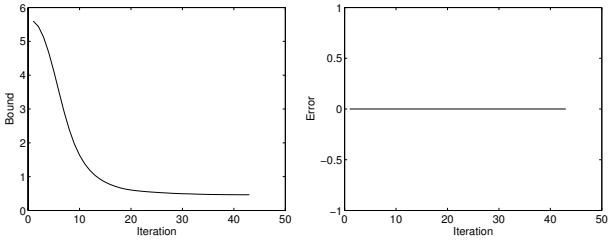
(c) Avg. Bound (d) Avg. Error

Fig. 1. Difficult Dataset Iteration.



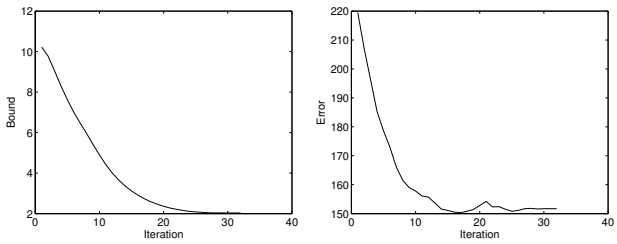
(a) Avg. Bound (b) Avg. Error

Fig. 2. Banana Dataset Iteration.



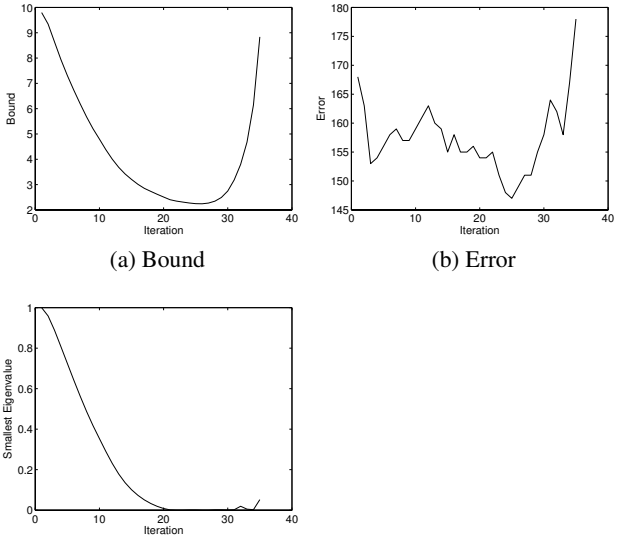
(a) Avg. Bound (b) Avg. Error

Fig. 3. Highleyman Dataset Iteration.



(a) Avg. Bound (b) Avg. Error

Fig. 4. Lithuanian Dataset Iteration.



(a) Bound (b) Error

(c) Smallest Eigenvalue

Fig. 5. Example of violation of positive definiteness.

5. CONCLUSION

When a linear subspace representation of data is applied to a SVM with a radial basis function kernel, a simple update rule can be derived to minimize the radius-margin bound. The basis update minimized the bound in a small number of iterations for the multiple dimension datasets tested. Given the proliferation of subspace based representations for a wide variety of classification tasks, the algorithm presented herein has general application.

6. REFERENCES

[1] B. Draper, K. Baek, M. Bartlett and R. Beveridge, "Recognizing faces with PCA and ICA", *Computer Vision and Image Understanding*, Vol. 91, pp. 115–137, 2003.

[2] P. Bradley and O. Mangasarian, "Feature selection via concave minimization and support vector machines", *Proceedings of the 15th International Conference on Machine Learning*, San Francisco, CA, pp. 82–90, 1998.

[3] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, "Feature selection for SVMs", *Advances in Neural Information Processing Systems*, Vol. 13, 2000.

		iter	*non-iter*	*p*
Difficult	Margin	0.426	0.337	1.81e-8
	NSV	48.6	71.7	3.86e-9
	Bound	3.44	5.51	5.32e-8
	Error	177.4	245.5	1.90e-3
Banana	Margin	0.447	0.373	7.28e-6
	NSV	44.8	60.3	7.14e-8
	Bound	3.79	5.04	6.34e-6
	Error	169.5	182.8	6.70e-2
Highleyman	Margin	0.777	0.413	1.59e-11
	NSV	23.3	47.3	5.92e-10
	Bound	1.64	5.60	2.39e-12
	Error	0	0	–
Lithuanian	Margin	0.457	0.306	2.00e-8
	NSV	46.9	72.8	5.59e-7
	Bound	4.90	10.22	3.74e-9
	Error	157.8	226.8	7.01e-4

Table 1. Means of margin, number of support vectors (NSV), bound, number of errors (Error) and p-value for the iterative (Iter) and non-iterative (Non-Iter) classification

[4] G. John and R. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem", *Proceedings of the 11th International Conference on Machine Learning*, San Mateo, CA, pp. 121–129, 1994.

[5] Chapelle, O., V. Vapnik, O. Bousquet and S. Mukherjee, *Choosing Multiple Parameters for Support Vector Machines*, Machine Learning, Vol. 46, No. 1, pp. 131-159, 2002.

[6] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs", *Neural Information Processing Systems 2002*, pp. 553-560, 2002.

[7] B. Krishnapuram, L. Carin, A. Hartemink and M. Figueiredo, "An EM algorithm for joint feature selection and classifier design",*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1105-1111, Sept. 2004.

[8] J. Fortuna and D. Capson, "*Improved Support Vector Classification Using PCA and ICA Feature Space Modification*", Pattern Recognition Journal, Vol. 37, No. 6, pp. 1117–1129, June 2004.

[9] S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms", *IEEE Transactions on Neural Networks*, Vol. 13, pp. 1225-1229, September 2002.

[10] V. Vapnik, "*Statistical Learning Theory*", John Wiley & Sons, 1998.

[11] R. Duin, *PRTOOLS, A Matlab Toolbox for Pattern Recognition*, Delft University of Technology, Delft, The Netherlands, 1997.