Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

On the reduction of datasets dimensionality in a classification method based on generalized eigenvalue problems

Mario R. Guarracino

RT-ICAR-NA-2005-15 **Novembre 2005**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: napoli@icar.cnr.it, URL: *www.na.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

On the reduction of datasets dimensionality in a classification method based on generalized eigenvalue problems

Mario R. Guarracino[1]

Rapporto Tecnico N.:
RT-ICAR-NA-2005-15

Data:
Novembre 2005

[1] ICAR-CNR

On the reduction of datasets dimensionality in a classification method based on generalized eigenvalue problems

Mario R. Guarracino[1]

[1] High Performance Computing and Networking Institute,
National Research Council, Italy

Abstract

Scientific experiments continuously produce huge amount of data that are used as input for analysis.The task of keeping those data coming from new experiments updated can become a cumbersome task. Various techniques have been devised in the field of machine learning to solve this problem. In the present work we propose an algorithm to determine a subset of a training set which can be used to compute the classification function using ReGEC, a generalized eigenvalue based technique. It has the advantage it can analyze new data of the training set and update the classification in a fast and reliable way. Numerical experiments show that this technique makes ReGEC competitive with respect to state of the art methods in terms of accuracy and execution time.

1 Introduction

Supervised learning refers to the capability of a system to learn from a set of examples, which is a set of input/output couples. This set is called the *training set*. The trained system is able to provide an answer (output) for a new question (input). The term *supervised* originates from the fact that the desired output for the training set of points is provided by an external teacher.

Supervised learning systems can find applications in many fields. A bank prefers to classify customer loan requests as “good” or“bad” depending on their ability to pay back. The Internal Revenue Service tries to discover tax evaders starting from the characteristics of known ones. As another example, a built-in system in a car could detect if a walking pedestrian is going to cross the street. There are many applications in biology and medicine. The tissues that are prone to cancer can be detected with high accuracy, or the new DNA sequences or proteins can be tracked down to their origins. Given its amino acids sequence, finding how a protein folds provides important information on its expression level. More examples related to numerical interpolation, handwriting recognition and Montecarlo methods for numerical integration can be found, for example, in [4, 6].
Support Vector Machine (SVMs) algorithms [23] are the state-of-the-art for the existing classification methods. These methods classify the points from two linearly separable sets in two classes by solving a quadratic optimization problem in

order to find the optimal separating hyperplane between these two classes. This hyperplane maximizes the distance from the convex hulls of each class. These techniques can be extended to the nonlinear cases by embedding the data in a nonlinear space using *kernel functions* [20].

SVMs have been one of the most successful methods in supervised learning with applications in a wide spectrum of research areas, ranging from pattern recognition [11] and text categorization [9] to biomedicine [12, 3, 14], brain-computer interface [22, 7], and financial applications [25, 21]. The robustness of SVMs originates from the strong fundamentals of statistical learning theory [23]. The training part relies on optimization of a quadratic convex cost function. Quadratic programming (QP) is an extensively studied field of mathematics and there are many general purpose methods to solve QP problems such as quasi-newton, primal-dual, and interior-point methods. The general purpose methods are suitable for small size problems, whereas for large problems chunkingsubset selection [15] and decomposition [17] methods use subsets of points to optimize SVMs. SVM-Lite [10] and LIBSVM [5] are among the most preferred implementations that use chunking-subset selection and decomposition methods efficiently. There are also efficient algorithms that exploit the special structure of the optimization problem such as Generalized Proximal SVMs (GEPSVM) [13].

The binary classification problem can be formulated as a generalized eigenvalue problem [13]. This formulation differs from SVMs since, instead of finding one hyperplane that separates the two classes, it finds two hyperplanes that approximate the two classes. The prior study requires the solution of two different eigenvalue problems. The aim of this work is to present a subset selection technique to be used in conjunction with *Regularized General Eigenvalue Classifier* (ReGEC)[8], a classification method based on generalized eigenvalue problem. This new method, which we will call CReGEC greatly reduces execution times, while providing comparable accuracy results. The execution times are now competitive with the fastest methods available.

The notation used in the paper is as follows. All vectors are column vectors, unless transposed to row vectors by a prime $'$. Scalar product of two vectors x and y in $\mathbb{R}^n$ will be denoted by $x'y$, 2-norm of x will be denoted by $\|x\|$ and the unit vector will be denoted by e.

The remainder of the the paper is organized as follows. Section 2 describes how the generalized eigenvalue classifier differs from the generic SVM methods. In Section 3 the subset selection technique is presented. In Section 4, numerical experiments are reported, and finally, in Section 5, conclusions are drawn and future work is proposed.

2 Related work

SVM algorithm for classification consists of finding a hyperplane that separates the elements belonging to two different classes. The separating hyperplane is usually chosen to maximize the margin between the two classes. The margin can be defined as the minimum distance between the separating hyperplane and the points of either class. The optimum hyperplane is the one that maximizes the margin. The points that are closest to the hyperplane are called *support vectors*, and are the only points needed to train the classifier. Consider two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{k\times m}$, that represent the two classes, each row being a point in the feature space. The quadratic linearly constrained problem to obtain the optimal hyperplane (w, b) is:

$$\begin{aligned} & \min f(w) = \frac{w'w}{2} \\ s.t. \quad & (Aw + b) \geq e \\ & (Bw + b) \leq -e. \end{aligned} \tag{1}$$

Mangasarian et al. [13] proposes to classify these two sets of points A and B using two hyperplanes, each closest to one set of points, and furthest from the other. Let $x'w - \gamma = 0$ be a hyperplane in $\mathbb{R}^m$. In order to satisfy the previous condition for the points in A, the hyperplanes can be obtained by solving the following optimization problem:

$$\min_{w,\gamma\neq 0} \frac{\|Aw - e\gamma\|^2}{\|Bw - e\gamma\|^2}. \tag{2}$$

The hyperplane for the B can be obtained by minimizing the inverse of the objective function in (3). Now, let

$$G = [A \quad -e]'[A \quad -e], \quad H = [B \quad -e]'[B \quad -e], \quad z = [w' \quad \gamma]', \tag{3}$$

then equation (2), becomes:

$$\min_{z\in\mathbb{R}^m} \frac{z'Gz}{z'Hz}. \tag{4}$$

The expression in (4) is the Raleigh quotient of the generalized eigenvalue problem $Gx = \lambda Hx$. The stationary points are obtained at and only at the eigenvectors of (4), where the value of the objective function is given by the eigenvalues. When H is positive definite, the Raleigh quotient is bounded and it ranges over the interval determined by minimum and maximum eigenvalues [16]. H is positive definite under the assumption that the columns of $[B \quad -e]$ are linearly independent. The inverse of the objective function in (4) has the same eigenvectors and reciprocal eigenvalues. Let $z_{min} = [w_1 \quad \gamma_1]$ and $z_{max} = [w_2 \quad \gamma_2]$ be the eigenvectors related to the eigenvalues of smallest and largest modulo, respectively. Then $x'w_1 - \gamma_1 = 0$ is the closest hyperplane to the set of points in A and the furthest from those in B and $x'w_2 - \gamma_2 = 0$ is the closest hyperplane to the set of points in B and the furthest from those in A.

Mangasarian et al. proposes to use Tikhonov regularization applied to a two-fold problem:

$$\min_{w,\gamma\neq 0} \frac{\|Aw - e\gamma\|^2 + \delta\|z\|^2}{\|Bw - e\gamma\|^2}, \tag{5}$$

and

$$\min_{w,\gamma\neq 0} \frac{\|Bw - e\gamma\|^2 + \delta\|z\|^2}{\|Aw - e\gamma\|^2}, \tag{6}$$

where δ is the regularization parameter and the new problems are still convex. The minimum eigenvalues-eigenvectors of these problems are approximations of the minimum and the maximum eigenvalues-eigenvectors of equation (4). The solutions $(w_i, \gamma_i), i = 1, 2$ to (5) and (6) represent the two hyperplanes approximating the two classes of training points.

In practice, if $\beta G - \alpha H$ is nonsingular for every α and β, it is possible to transform the problem into another problem that is nonsingular and that has the same eigenvectors of the initial one, as proved by Y. Saad ([19], p. 288). In the linear case, the regularized problem becomes

$$\min_{w,\gamma\neq 0} \frac{\|Aw - e\gamma\|^2 + \hat{\delta_1}\|Bw - e\gamma\|^2}{\|Bw - e\gamma\|^2 + \hat{\delta_2}\|Aw - e\gamma\|^2}. \tag{7}$$

The spectrum is now shifted and inverted so that the minimum eigenvalue of the original problem becomes the maximum of the regularized one, and the maximum becomes the minimum eigenvalue. Choosing the eigenvectors related to the new minimum and maximum eigenvalue, we still obtain the same ones of the original problem.

This regularization works for the linear case if we suppose that in each class of the training set there is a number of linearly independent rows that is at least equal to the number of the features. This is often the case and, since the number of points in the training set is much greater than the number of features, $Ker(G)$ and $Ker(H)$ have both dimension 1. In this case, the probability of a nontrivial intersection is zero.

A standard technique in SVMs to obtain a greater separability between sets is to embed the points into a nonlinear space, via kernel functions. In this work we use the *Gaussian kernel*,

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}. \tag{8}$$

In (8), x_i and x_j denote two points in the feature space. This technique usually allows one to obtain better results, as shown in several applications. Results regarding nonlinearly separable problems [1, 2] still hold and a formulation for the eigenvalues problem can easily be derived.

In the nonlinear case the situation is different. Using the kernel function (8), each element of the kernel matrix is

$$K(A, B)_{i,j} = e^{-\frac{\|A_i - B_j\|^2}{\sigma}}. \tag{9}$$

Let

$$C = \begin{bmatrix} A \\ B \end{bmatrix},$$

then, problem (2) becomes:

$$\min_{w,\gamma\neq 0} \frac{\|K(A, C)u - e\gamma\|^2}{\|K(B, C)u - e\gamma\|^2}. \tag{10}$$

Now the associated eigenvalue problem has matrices of order $n + k + 1$ and rank at most m. This means a regularization technique is needed, since the problem can be singular.
We propose to generate the following two proximal surfaces:

$$K(x, C)u_1 - \gamma_1 = 0, \quad K(x, C)u_2 - \gamma_2 = 0 \tag{11}$$

by solving the following problem

$$\min_{w,\gamma \neq 0} \frac{\|K(A, C)u - e\gamma\|^2 + \delta\|\tilde{K_B}u - e\gamma\|^2}{\|K(B, C)u - e\gamma\|^2 + \delta\|\tilde{K_A}u - e\gamma\|^2} \tag{12}$$

where $\tilde{K_A}$ and $\tilde{K_B}$ are diagonal matrices with the diagonal entries from the matrices $K(A, C)$ and $K(B, C)$. The perturbation theory of eigenvalue problems [24] provides an estimation of the distance between the original and the regularized eigenvectors. If we call z an eigenvector of the initial problem and $z(\delta)$ the corresponding one in the regularized problem, then $|z - z(\delta)| = \mathcal{O}(\delta)$, which means their closeness is in the order of δ.

As mentioned before, the minimum and the maximum eigenvalues obtained from the solution of (12) provide the proximal planes P_i, $i = 1, 2$ to classify the new points. A point x is classified using the distance

$$dist(x, P_i) = \frac{\|K(x, C)u - \gamma\|^2}{\|u\|^2}. \tag{13}$$

and the class of a point x is determined as

$$class(x) = argmin_{i=1,2}\{dist(x, P_i)\}. \tag{14}$$

3 A subset selection technique

The idea behind the proposed subset selection technique is to find a subset of the training set which classifies the training set with sufficient accuracy. To evaluate classification we use ReGEC. Since the training set is a sample of the population, the subset should be able to retain the information needed to classify the whole population. The algorithm starts to evaluate the classification with one point from each class in the chunked training set. Each point is then tested with respect to the chunked set and is added if *i*) when added to the chunked set, it still classifies the previous chunked set with accuracy 1, and *ii*) the accuracy of classification of the new chunked set on the training set is greater of the previous chunked set.
In this way, a point is added iff the new set still correctly classifies all previously added points, and it is capable to provide more accurate classification of the training set.

Before starting this procedure, a sorting process is needed in order to feed the algorithm in such a way the overall classification is comparable with the initial one. We have empirically found that, if we sort the features of centers of gravity of each class in descending order, with respect to their absolute value, and we use

dataset	train	test	m
Diabetis	468	300	8
German	700	300	20
Flare-solar	666	400	9
Titanic	150	2051	3

Table 1: Datasets

dataset	train	σ	ReGEC	chunked	σ	CReGEC	SVM
Diabetis	468	500.	74.56	12	492.	73.85	76.21
German	700	500.	70.26	24	10.	73.65	75.66
Flare-solar	666	3.	58.23	11	400.	63.19	65.80
Titanic	150	150.	75.29	4	50.	75.04	77.36

Table 2: Classification accuracy using gaussian kernel

this ordering to sort the training set, we obtain classification accuracy that are, as it will be shown in the next paragraph, comparable with the one obtained with the whole training set.

In the next section we present comparisons of accuracy and speed of the proposed method to the original generalized proximal classifier as well as the widely used SVMs implementations.

4 Numerical results

The aforementioned method has been tested on benchmark data sets publicly available. Results regard their performance in terms of classification accuracy and execution time when using a non linear kernel. We used data from IDA repository [18]. That repository is widely used to compare the performance of new algorithms to the existing methods. For each data set, it offers 100 predefined random splits into training and test sets. For several algorithms, results obtained from each trial, including SVMs, are recorded. The accuracy results for the non linear kernel from [18]. Execution times and the other accuracy results have been calculated using an Intel Centrino CPU 1.6GHz, 512MB RAM running Windows XP, Matlab 6. Matlab function *eig* for the solution of the generalized eigenvalue problem has been used for ReGEC. The latest releases for LIBSVM [5] and SVMlight [10] have been used to compare these methods with SVMs.

In Table 1, for each dataset, name, dimension of the train and test sets, and number of features are reported. In Table 2, classification accuracy using gaussian kernels has been evaluated for CReGEC, ReGEC and SVM. For the first two methods the dimension of the train and σ have been given. As it can be seen, the dimension of the training set is dramatically reduced, while accuracy is almost the same.
In Table 3, elapsed time is reported. In all cases, CReGEC outperforms ReGEC. To better understand the execution times of all considered methods, in Table 4,

```
Let $A \in \mathbb{R}^{m \times s}$ and $B \in \mathbb{R}^{n \times s}$ the training points in each class.
Choose appropriate $\delta$ and $\sigma \in \mathbb{R}$
Set $acc = 1$, $acc\_glob = 0$ and $acc\_glob\_old = 0$.

% Compute centers of gravity of A and B
  $barA = sum(A, 1)/m;$
  $barB = sum(B, 1)/n;$

% Sort A and B wrt features of centers of gravity
  $[orderA, IorderA] = sort(abs(barA));$
  $[orderB, IorderB] = sort(abs(barB));$
  $A = sortrows(A, (sort(IorderA.^{-1})).^{-1});$
  $B = sortrows(B, (sort(IorderB.^{-1})).^{-1});$

  $kA = 2;$  % index of A point to add
  $kB = 2;$  % index of B point to add
  $ttr = [A(1, :); B(1, :)];$
  $ttr\_l = [1; -1];$
  $while(acc\_glob \sim= 1 \&\& kA <= m \&\& kB <= n)$

% Check whether the next point to A has to be added
     $if(kA \sim= m \&\& acc\_glob \sim= 1)$
       $temp = [ttr; A(kA, :)];$
       $temp\_l = [ttr\_l; 1];$
    % Compute classification accuracy of chunked dataset
       $acc = regec(temp, temp\_l, ttr, ttr\_l, \sigma, \delta);$
       $if(acc == 1)$
    % Compute accuracy of classification on the trainset
         $acc_glob = regec(temp, temp\_l, train, train\_l, \sigma, \delta);$
         $if(acc\_glob > acc\_glob\_old)$
           $acc\_glob\_old = acc\_glob;$
           $ttr = temp;$
           $ttr\_l = temp\_l;$
         end
       end
       $kA = kA + 1;$
     end
    % Check whether the next point to B has to be added
     ...
  end
```

Figure 1: ReGEC algorithm

Dataset	**CReGEC**	**ReGEC**
Diabetis	0.0499	2.5437
German	0.0824	8.1464
Flare-solar	0.0641	4.6579
Titanic	0.3019	1.8936

Table 3: Elapsed time in seconds using gaussian kernel

Dataset	**CReGEC**	**ReGEC**	**LIBSVM**
Diabetis	1	50.9759	5.8777
German	1	98.8640	7.3934
Flare-solar	1	72.6661	5.5419
Titanic	1	6,2722	0.7461

Table 4: Relative execution time

we set to 1 the execution time of CReGEC, and we evaluate the execution time ratio between CReGEC and ReGEC. In order to compare those results, with the ones reported in [8] and obtained on a processor with a different speed, we have evaluated the ratio between ReGEC and libSVM execution times. From the quotient of those two quantities we can evaluate the relative execution time of each method with respect to the others. For example, we can see that on Diabetis dataset ReGEC takes 50.9759 times longer than CReGEC to evaluate the classification, and that SVM is only 5.8777 times slower than CReGEC. Only in the case of Titanic dataset, SVM outperforms CReGEC. We note that the gain grows with the number of points in the training set.

5 Conclusions and future work

Research activities related to supervised learning have an important role in many scientific and engineering applications. In the present work a novel subset selection technique and its application has been proposed and tested against other methods on a number of datasets. Results show that the proposed method *i*) has a classification accuracy comparable to other methods, *ii*) has a computational performance comparable to most of the other methods, and *iii*) is much faster then the others.

In the last years there has been a wide effort devoted to the implementation of algorithms for the efficient computation of algorithms for the selection of features that influence classification. We will investigate techniques that can applied to the generalized eigenvalue classification methods.

References

[1] K. Bennet and C. Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, 2(2):1–13, 2000.

[2] K. Bennett and O. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[3] G. Patrizi C. Cifarelli. Solving large protein folding problem by a linear complementarity algorithmwith 0-1 variables. *Optimization Methods and Softwares*, 2005. Submitted for publication.

[4] F. Cucker and S. Smale. On the mathematical foundation of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2001.

[5] C.J. Lin C.W. Hsu, C.C. Chang. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf, 2004.

[6] F. Giannessi. Complementarity problems and their applications to structural engineering. In Pitagora, editor, *Methods and algorithms for optimization*, pages 507–514, Bologna, 1982.

[7] T. Ebrahimi G.N. Garcia and J.M. Vesin. Joint time-frequency-space classification of eeg in a brain-computer interface application. *Journal on Applied Signal Processing*, pages 713–729, 2003.

[8] M. R. Guarracino, C. Cifarelli, O. Seref, and P. M. Pardalos. A classification algorithm based on generalized eigenvalue problems. *Optimization Methods and Software*, page to appear, 2006.

[9] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Ndellec and Cline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pages 137–142, Berlin, 1998. Springer.

[10] T. Joachims. *Making large-Scale SVM Learning Practical.* Advances in Kernel Methods - Support Vector Learning. MIT-Press, 1999.

[11] S. Lee and A. Verri. Pattern recognition with support vector machines. In *SVM 2002*, Niagara Falls, Canada, 2002. Springer.

[12] D. Lin N. Cristianini C. Sugne T. Furey M. Ares M. Brown, W. Grundy and D. Haussler. Knowledge-base analysis of microarray gene expressiondata by using support vector machines. *PNAS*, 97(1):262–267, 2000.

[13] O. L. Mangasarian and E. W. Wild. Multisurface proximal support vector classification via generalized eigenvalues. Technical Report 04-03, Data Mining Institute, September 2004.

[14] W. S. Noble. *Kernel Methods in Computational Biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT Press, 2004.

[15] R. F. E. Osuna and F. Girosi. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 276–285, 1997.

[16] B. N. Parlett. *The Symmetric Eigenvalue Problem*, page 357. SIAM, Philadelphia,PA, 1998.

[17] J. Platt. *Advances in Kernel Methods: Support Vector Learning*, chapter Fast training of SVMs using sequential minimal optimization, pages 185–208. MIT press, Cambridge, MA, 1999.

[18] J.Weston B. Schlkopf S. Mika, G. Rtsch and K. R. Mller. Fisher discriminant analysis with kernels. *IEEE Neural Networks for Signal Processing*, IX:41–48, 1999.

[19] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halsted Press, New York, NY, 1992.

[20] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, Cambridge, UK, 2004.

[21] H. Ince T. B. Trafalis. Support vector machine for regression and applications to financial forecasting. In *International Joint Conference on Neural Networks (IJCNN'02)*, Como, Italy, 2002. IEEE-INNS-ENNS.

[22] T. Hinterberger J. Weston M. Bogdan N. Birbaumer T. N. Lal, M. Schroeder and B. Schlkopf. Support vector channel selection in bci. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

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

[24] J. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, 1965.

[25] C. J. Hsu W. H. Chenb S. Wuc Z. Huang, H. Chen. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37:543–558, 2004.