ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 213-218

Generalized Support Vector Machines

Davide Mattera [*], Francesco Palmieri[*], and Simon Haykin[**]

[*] Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni
Università degli Studi di Napoli Federico II
Via Claudio, 21, 80125, Napoli, Italy

[**] Communications Research Laboratory. McMaster University,
1280 Main Street W., Hamilton, Ontario, Canada, L8S 4K1

Abstract. Most Support Vector (SV) methods proposed in the recent literature can be viewed in a unified framework with great flexibility in terms of the choice of the basis functions. We show that all these problems can be solved within a unique approach if we are equipped with a robust method for finding a sparse solution of a linear system. Moreover, for such a purpose, we propose an iterative algorithm that can be simply implemented. This allows us to generalize the classical SV method to a generic choice of the basis functions.

1. Introduction

In many engineering problems, one is interested in finding a solution of a linear system of equations:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b} \tag{1}$$

where A is a $\ell \times M$ real matrix, $\boldsymbol{u}$ an M-dimensional vector and $\boldsymbol{b}$ an ℓ-dimensional vector. We consider here the case in which one is not simply interested in obtaining an accurate solution, but in obtaining a solution which is both accurate and sparse, i.e. an accurate solution such that a large part of its components are null. Three settings of the problem may be of interest: 1) look for the most accurate solution such that the number of its nonzero components is smaller than a fixed value:

$$S1 : \begin{cases} \min\limits_{\boldsymbol{u}} \mathcal{E}(\boldsymbol{u}) \\ \\ \mathcal{N}(\boldsymbol{u}) \leq d_1 \end{cases} , \tag{2}$$

where $\mathcal{E}(\boldsymbol{u})$ is a measure of the accuracy of $\boldsymbol{u}$ as solution of the linear system and $\mathcal{N}(\boldsymbol{u})$ is the number of nonzero components of $\boldsymbol{u}$; 2) look for the sparsest point in a given surrounding of the solutions:

$$S2 : \begin{cases} \min\limits_{\boldsymbol{u}} \mathcal{N}(\boldsymbol{u}) \\ \\ \mathcal{E}(\boldsymbol{u}) \leq E \end{cases} ; \tag{3}$$

3) minimize the following quantity:

$$S3: \ \min_{\boldsymbol{u}} \mathcal{E}(\boldsymbol{u}) + \varepsilon \mathcal{N}(\boldsymbol{u}) \ , \tag{4}$$

where ε is a positive constant which controls the trading of sparsity and accuracy. Moreover, the following constraint should be imposed in all the three settings to avoid an unstable solution:

$$|u_j| \leq C_j, \qquad\qquad j \in \{1, \ldots, M\}, \tag{5}$$

where C_j are constants (usually set to the same value C).

Brute force solutions of these problems would be time-consuming. We review, with some refinements, the SV method which allows one to obtain a robust solution in a reasonable time.

2. An approach for a positive semidefinite matrix

This type of approach was introduced for particular applicative scenarios in [7, 9, 1, 2] with the name of SV method or Basis Pursuit De-Noising. The matrix of the linear system is assumed to be a square positive semidefinite matrix. (We will see in the following how to utilize this method when this assumption is not valid). In this approach, the idea is to solve problems $S1$, $S2$ and $S3$ by utilizing the following approximations of $\mathcal{E}(\boldsymbol{u})$ and $\mathcal{N}(\boldsymbol{u})$:

$$\mathcal{E}(\boldsymbol{u}) \simeq \mathcal{E}_1(\boldsymbol{u}) \ \stackrel{\triangle}{=} \ \frac{1}{2}\boldsymbol{u}^T \boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}^T \boldsymbol{u} \ , \tag{6}$$

$$\mathcal{N}(\boldsymbol{u}) \simeq \mathcal{N}_1(\boldsymbol{u}) \ \stackrel{\triangle}{=} \ ||\boldsymbol{u}||_{L_1} = \sum_{j \in \mathcal{B}} |u_j|, \tag{7}$$

where u_j is the jth component of $\boldsymbol{u}$ and $\mathcal{B} \ \stackrel{\triangle}{=} \ \{1, \ldots, M\}$. If some indices are removed from $\mathcal{B}$, then the corresponding components of $\boldsymbol{u}$ are included in the obtained solution. Approximation (6) is motivated by the fact that, for a positive definite $\boldsymbol{A}$, $\mathcal{E}_1(\boldsymbol{u})$ is a quadratic function of $\boldsymbol{u}$ which assumes its minimum at the true solution of the linear system. Approximation (7) comes from the fact that $\mathcal{N}_1(\boldsymbol{u})$ is one of the best convex approximation of $\mathcal{N}(\boldsymbol{u})$. This assures that the considered optimization problem has a unique solution. Some non-convex approximation of $\mathcal{N}(\boldsymbol{u})$ were proposed (see [2]) but the resulting optimization problem presents many local minima. Perhaps, one could try to use the solution of the convex approximation as starting point for the non-convex problem. The main drawback of the SV approach is that it requires the solution of a nonlinear optimization problem. Many methods exist for solving it but they are generally difficult to implement and have consistent memory requirements (apart from the memory required to store $\boldsymbol{A}$ and $\boldsymbol{b}$).

We propose here a simple learning algorithm for solving the resulting optimization problem which is very simple to implement and does not require any additional storage.

2.1. A simple method for the sparse and robust solution of linear system

In this section we consider the problem to be solved for the setting $S3$; similar algorithms can be derived in the setting $S1$ and $S2$. We have to solve the problem $S3$ (4) with the approximations (6) and (7) subject to the constraint (5). We propose the application of a coordinate-descent method for its solution. We consider an index $j \in \{1, \ldots, M\}$ and minimize with reference to the variable u_j, maintaining fixed all the other variables; we simply need to apply the following updating rule [5]:

$$u_j = \min \left[C_j \cdot \Theta\left(|d_j - \hat{f}_j| - \varepsilon \right), \frac{\left| |d_j - \hat{f}_j| - \varepsilon \right|}{A_{jj}} \right] sign(d_j - \hat{f}_j). \qquad (8)$$

where $\hat{f}_j \overset{\triangle}{=} \sum_{i=1, i \neq j}^{M} A_{ij} u_i$, A_{ij} is the generic element of $\boldsymbol{A}$ and $\Theta(\xi)$ is the classic step function, i.e. it is equal to zero if $\xi \leq 0$ and it is equal to one if $\xi > 0$. Our simple approach consists in starting from a point which satisfies the bound (5) and in cycling on all the variables until the variations in the components of $\boldsymbol{u}$ become negligible for an entire cycle or for some cycles. If the value of j does not belong to $\mathcal{B}$ and we do not want to limit its value with the constraint (5), then the up-dating rule (8) reduces to $u_j = \dfrac{d_j - \hat{f}_j}{A_{jj}}$.

The main advantage of the proposed method lies in the simplicity of its implementation, that allows us to utilize it also for very large M. Another advantage lies in the fact that its complexity mainly depends on the number of the variables such that $0 < |u_j| < C$ at the optimum solution; therefore, it is better suited for sparse problems and for the case of small value of C (strong regularization). Some practical tricks, useful for reducing execution time, have been described in [5].

3. SV methods for a general matrix

In the case in which the matrix $\boldsymbol{A}$ is not square or not positive semidefinite, one can consider the following linear problem

$$(\boldsymbol{A}^T \boldsymbol{A})\boldsymbol{u} = \boldsymbol{A}^T \boldsymbol{b} \; . \qquad (9)$$

A sparse and accurate solution of (9) can be obtained with the method considered in the Section 2 since the matrix $\boldsymbol{A}^T\boldsymbol{A}$ is square and positive semidefinite.

3.1. A particular case

Let us consider the case in which $M > \ell$ and the matrix $\boldsymbol{A}$ can be written as

$$\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2], \qquad (10)$$

where $\boldsymbol{A}_1$ is a $\ell \times \ell$ symmetric positive semidefinite matrix. Let us write the vector $\boldsymbol{u}$ as $[\boldsymbol{u}_1^T, \boldsymbol{u}_2^T]^T$, where $\boldsymbol{u}_1$ is an ℓ-dimensional vector. The SV approach suggests [8, 7] finding $\boldsymbol{u}_1$ by passing to the previous case with the square positive semidefinite matrix equal to $\boldsymbol{A}_1$ and by adding to the problem the following additional constraint:

$$\boldsymbol{A}_2^T \boldsymbol{u}_1 = \boldsymbol{0} \ . \tag{11}$$

The advantage consists in solving an ℓ dimensional problem rather than an M dimensional one. However, we pay this reduction with the additional constraint (11). Moreover, it is not always easy to complete the solution determining the vector $\boldsymbol{u}_2$. We remark that this approach is not general since it can be applied only if the matrix $\boldsymbol{A}$ can be written in the form (10); moreover, it is not equivalent to (9).

4. Generalized Support Vector Machine

Let us consider the classical problem of regression estimation: we are given ℓ noisy examples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell)\}$ obtained from a unknown target function $\overline{f}(\boldsymbol{x})$ and a class of functions

$$\hat{f}(\boldsymbol{x}) = u_1 \phi_1(\boldsymbol{x}) + u_2 \phi_2(\boldsymbol{x}) + \ldots + u_M \phi_M(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x}) \boldsymbol{u} \tag{12}$$

where $\boldsymbol{u} \triangleq [u_1, \ldots, u_M]^T$ and $\boldsymbol{\phi}(\boldsymbol{x}) \triangleq [\phi_1(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})]$ is a vector of known functions; M can be also chosen dependent on ℓ.

Both the theory of regularization and the Statistical Learning Theory [9, 6, 3] suggests finding an approximation of the target function by solving the following linear system:

$$\left(\boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{R}\right) \boldsymbol{u} = \boldsymbol{G}^T \boldsymbol{y} \tag{13}$$

where $\boldsymbol{y} \triangleq [y_1, \ldots, y_\ell]$, $\boldsymbol{G} \triangleq [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_\ell)]^T$. The matrix $\boldsymbol{R}$ is a regularizing matrix weighted by a small value of λ and it depends on the knowledge available *a priori*; when no a priori knowledge is available, a typical choice is $\boldsymbol{R} = \boldsymbol{I}$.

By applying to the linear system (13) the methods described in Sections 2 and 3 for deriving a sparse solution, we obtain an original improvement on the existing techniques for regression estimation; we call this approach "Generalized SVM". The reason for which one is interested in sparsity lies in the reduced computational complexity for evaluating $\hat{f}$, in the greater accuracy obtainable and in the possibility to give a physical meaning to the obtained model, i.e. in trying an explanation of the given data in the shortest and most accurate form.

The parameters λ and C, as well as the parameters eventually introduced in the definition of the function $\boldsymbol{\phi}(\cdot)$, need to be chosen by utilizing some cross-validation technique or the bounds provided by the VC-theory [9].

We show now that other already proposed methods for regression estimation, developed in the context of SV Machine (SVM), can be seen as particular cases of our method. Let us first consider the case in which $M = \ell$ and $\phi_i(\boldsymbol{x}) = \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i)$ where $\mathcal{K}(\cdot, \cdot)$ is a symmetric positive definite kernel. In this case, choosing $\boldsymbol{R} = \boldsymbol{G}$, we obtain that the solution $\boldsymbol{u}$ should satisfy the following linear system:

$$(\boldsymbol{G} + \lambda \boldsymbol{I})\, \boldsymbol{u} = \boldsymbol{y} \ . \tag{14}$$

This equation is well-known as *strict interpolation* condition [3]. If we apply the method described in Section 2 to make sparse the solution of (14), we obtain the classical method of SVM when the threshold is fixed to zero. Actually, in the classical SVM λ is set to zero; it becomes different from zero if a quadratic-linear ε-insensitive loss function, rather than simply a linear ε-insensitive one, is chosen. Moreover, for a symmetric positive semidefinite kernel $\mathcal{K}(\cdot, \cdot)$, robust solutions of (14) and (13) are very similar in practice; SVM still refers to (14).

The classical SVM machine is obtained for $M = \ell + 1$, $\phi_i(\boldsymbol{x}) = \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i)$ $(i = 1, \dots, \ell\ )$ as before and $\phi_{\ell+1}(\boldsymbol{x}) \equiv 1$. This choice was introduced in [9] since it does not modify the VC-dimension of the considered class of functions. In such a case, since the matrix $\boldsymbol{G}$ satisfies (10), one has to solve the ℓ-dimensional optimization problem (for the case without threshold) over the first ℓ components $\boldsymbol{u}_1$ of $\boldsymbol{u}$ under the constraint

$$\boldsymbol{1}^T \boldsymbol{u}_1 = 0. \tag{15}$$

In general, one can consider $M > l$ and define ϕ_i as before for $i \in \{1, \dots, \ell\}$ and other generic functions ϕ_i for $i \in \{\ell + 1, M\}$. Then, one can get the first ℓ components of the solution $\boldsymbol{u}$ by replacing the constraint (15) with (11) where $\boldsymbol{A}_2 = [\boldsymbol{\phi}_{\ell+1} \dots \boldsymbol{\phi}_M]$ with $\boldsymbol{\phi}_i = [\phi_i(\boldsymbol{x}_1) \dots \phi_i(\boldsymbol{x}_\ell)]^T$ $(i \in \{\ell + 1, \dots, M\})$. This choice implies that we are not interested in rendering null the last $M - \ell$ components of $\boldsymbol{u}$. This may come from a strong *a priori* knowledge about the opportunity of including the functions ϕ_i $(i = \ell + 1, \dots, d)$ in the estimated function. A simple choice would be that of including the linear term in last $M - \ell$ components of the expansion. This is very useful in some engineering problems where the nonlinear target function is known to be slightly nonlinear. The general SVM method with the constraint (11) has been independently discovered in [8]; a similar approach for classification problems has been introduced in [4].

5. Conclusions

We stated the general problem of finding a sparse solution of a linear system and reviewed the SV approach for solving it. Then, we note that the problem of learning from examples a function in the form (12) requires the solution of a linear system. By applying the SV approach for solving it sparsely, we have obtained as particular cases many variations of the SVM. We conclude with a simple example in which SVM cannot work well while Generalized SVM can.

Let us assume that the number of examples is $\ell = 50,000$; let us suppose we do not want to utilize more than $M = 3000$ Radial Basis Functions (RBF) in the expansion to be made sparse, because we cannot deal with a numerical problem with more than 3000 variables. With the classical SVM, we are forced to consider a class of function with RBF centered on $3,000$ out of the $50,000$ examples. Then, SVM utilizes only the chosen $3,000$ examples to get a good approximation inside the considered class of functions. With the Generalized SVM one considers the $M \times M$ matrix $\boldsymbol{A} = \boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}$ and, therefore, utilize all the $50,000$ to find a good approximation inside the same class of functions. This is, however, only an example in which one can utilize the freedom of choosing the vector $\boldsymbol{\phi}(\boldsymbol{x})$ and its dimension M to improve on the existing techniques. A good choice of the vector $\boldsymbol{\phi}(\boldsymbol{x})$ is an important issue for obtaining a good and sparse approximation in a specific problem. In our general approach this choice constitutes the counterpart of the kernel choice in the classical SVM approach.

References

[1] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. Technical Report 479, Department of Statistics, Stanford University, 1995.

[2] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, pages 223–244, 1998.

[3] S. Haykin. *Neural Networks : A Comprehensive Foundation.* Macmillan, New York, 1994.

[4] O. L. Mangasarian. Generalized support vector machines. Technical Report 98-14, Computer Sciences Department, University of Wisconsin, USA, Oct 1998. available at http://www.cs.wisc.edu/~ olvi/olvi.html.

[5] D. Mattera, F. Palmieri, and S. Haykin. An explicit algorithm for training support vector machines. submitted to IEEE Signal Processing Letters, August 1998.

[6] A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211 – 231, 1998.

[7] A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report TR-1998-030, NEURO-COLT, Royal Holloway College, 1998.

[8] A. J. Smola, T. Friess, and B. Schölkopf. Semiparametric support vector and linear programming machines. In *Advances in Neural Information Processing Systems.* MIT Press, 1998.

[9] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.