ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 121-126

On-line Support Vector Machines and Optimization Strategies

J.M. Górriz, C.G. Puntonet, M. Salmerón,
R. Martin-Clemente and S. Hornillo-Mellado and J. Ortega
E.S.I., Informática, Universidad de Granada
C/ Periodista Daniel Saucedo, 18071 Granada, Spain

Abstract. In this work we derive a new on-line parametric model for time series forecasting based on Vapnik-Chervonenkis (VC) theory. Using the strong connection between *support vector machines* (SVM) and *Regularization theory* (RT), we propose a regularization operator in order to obtain a suitable expansion of radial basis functions (RBFs) with the corresponding expressions for updating neural parameters. This operator seeks for the "flattest" function in a feature space, minimizing the risk functional. Finally we mention some modifications and extensions that can be applied to control neural resources and select relevant input space in order to avoid high computational effort (batch learning).

1 Introduction

The purpose of this work is twofold. It introduces the foundations of SVM [1] and its connection with RT [2] in order to show the new on-line algorithm for time series forecasting. SVMs are learning algorithms based on the structural risk minimization principle [1] (SRM) characterized by the use of the expansion of SV "admissible" kernels and the sparsity of the solution. They have been proposed as a technique in time series forecasting [3] and have faced the overfitting problem, presented in classical neural networks, thanks to their high capacity for generalization. The solution for SVM prediction is achieved solving the constrained quadratic programming problem thus SV machines are nonparametric techniques, i.e. the number of basis functions are unknown before hand. The solution of this complex problem in *real-time applications* can be extremely uncomfortable because of high computational time demand.

SVM are essentially Regularization Networks (RN) with the kernels being Green´s function of the corresponding regularization operators [4]. Using this connection, with a clever choice of regularization operator (based on SVM philosophy), we should obtain a parametric model being very resistant to the overfitting problem. Our parametric model is a *Resource allocating Network* [5] characterized by the control of neural resources and by the use of matrix decompositions, i.e. Singular Value Decomposition (SVD) and QR Decomposition to input selection and neural pruning [6].

We organize the essay as follows. SV algorithm and its connection to RT Theory will be presented in sections 2 and 3. The new on-line algorithm will be compare to a previous version of it and the standard SVM in section 4. Finally we state some conclusions in section 5.

2 SVM and Regularization Theory

The SV algorithm is a nonlinear generalization of the generalized portrait developed in the sixties by Vapnik and Lerner in [1]. The basic idea in SVM for regression and function estimation, is to use a mapping function Φ from the input space $\mathcal{X}$ into a high dimensional feature space $\mathcal{F}$ and then to apply a linear regression. Thus the standard linear regression transforms into:

$$f(x) = \langle \omega \cdot \Phi(x) \rangle + b. \tag{1}$$

where $\Phi : \mathcal{X} \to \mathcal{F}$, b is a bias or threshold and $\omega \in \mathcal{F}$ is a vector defining the function class. The target is to determinate ω, i.e. the set of parameters in the neural network, minimizing the regularizated risk expressed as:

$$R_{reg}[f] = R_{emp}[f] + \lambda ||\omega||^2. \tag{2}$$

thus we are enforcing "flatness" in feature space, that is we seek small ω. Note that equation 2 is very common in RN with a certain second term. SVM algorithm is a way of solving the minimization of equation 2, that can be expressed as a quadratic programming problem using the formulation stated in [1]. The optimization problem is solve constructing a Lagrange function by introducing dual variables, and the selected loss function. Once it is uniquely solved, we can write the vector ω in terms of the data points as follows:

$$\omega = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \Phi(x_i). \tag{3}$$

where α_i, α_i^* are the solutions of the mentioned quadratic problem. Once this problem, with high computational demand [1], is solved we take equation 3 into 1 and obtain the solution in terms of dot products:

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \langle \Phi(x_i) \cdot \Phi(x) \rangle + b. \tag{4}$$

At this point we use a trick to avoid computing the dot product in high dimensional feature space in equation 4, replacing it by a kernel function that

[1]This calculation must be compute several times during the process

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 121-126

satisfies Mercer´s condition. Mercer´s Theorem [7] guarantees the existence of this kernel function:

$$f(x) = \sum_{i=1}^{\ell} h_i \cdot k(x_i, x) + b. \tag{5}$$

where $h_i \equiv (\alpha_i - \alpha_i^*)$ and $k(x_i, x) = \langle \Phi(x_i) \cdot \Phi(x) \rangle$.

RT appeared in the methods for solving *ill posed problems* [2]. In RN we minimize a expression similar to equation 2. However, the search criterium is enforcing smoothness (instead of flatness) for the function in input space (instead of feature space). Thus we get:

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} ||\hat{P}f||^2. \tag{6}$$

where $\hat{P}$ denotes a regularization operator in the sense of [2], mapping from the Hilbert Space H of functions to a dot product Space D such as $\langle f, g \rangle \quad \forall f, g \in H$ is well defined. Applying Fréchet´s differential[2] to equation 6 and the concept of Green´s function of $\hat{P}^* \hat{P}$[3], we get [6]:

$$f(x) = \lambda \sum_{i=1}^{\ell} [y_i - f(x_i)]_\epsilon \cdot G(x, x_i). \tag{7}$$

The correspondence between SVM and RN (equations 5 and 7) is proved if and only if the Green´s function G is an "admissible" kernel in the terms of Mercer´s theorem [7],i.e. we can write G as:

$$G(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \quad with \quad \Phi : x_i \to (\hat{P}G)(x_i, .). \tag{8}$$

Prove: Minimizing $||\mathbf{P}f||^2$ can be expressed as:

$$||\mathbf{P}f||^2 = \int dx (\mathbf{P}f)^2 = \int dx f(x) \mathbf{P}^* \mathbf{P} f(x) \tag{9}$$

we can expand f in terms of green's function associated to $\mathbf{P}$, thus we get:

$$\begin{aligned} ||\mathbf{P}f||^2 &= \textstyle\sum_{i,j}^N h_i h_j \int dx G(x, x_i) \mathbf{P}^* \mathbf{P} G(x, x_j) \\ &= \textstyle\sum_{i,j}^N h_i h_j \int dx G(x, x_i) \delta(x - x_j) = \sum_{i,j}^N h_i h_j G(x_j, x_i) \end{aligned} \tag{10}$$

then only if G is Mercer Kernel it correspond to a dot product in some feature space. Then minimizing 6 is equivalent to minimize 2†. A similar prove of this connection can be found in [4]. Hence given a regularization operator, we can find an admissible kernel such that SV machine using it will enforce flatness in feature space and minimize the equation 6. Moreover, given a SV kernel we can find a regularization operator such that the SVM can be seen as a RN.

[2]Generalizated differentiation of a function: $dR[f] = \left[\frac{d}{d\rho} R[f + \rho h]\right]$, where $h \in H$.

[3]$\hat{P}^* \hat{P} \cdot G(x_i, x_j) = \delta(x_i - x_j)$, (here δ denotes the Dirac´s δ, that is $\langle f, \delta(x_i) \rangle = f(x_i)$)

3 On-line Algorithm Using Regularization Operators

In this section we show a new on-line RN based on “Resource Allocating Network” algorithms (RAN) [4] [5] which consist of a network using RBFs, a strategy for allocating new units (RBFs), using two part novelty condition [5]; input space selection and neural pruning using matrix decompositions such as SVD and QR with pivoting [6]; and a learning rule based on SRM [1]. Our network has 1 layer as is stated in equation 5. In terms of RBFs the latter equation can be expressed as:

$$f(x) = \sum_{i=1}^{N(t)} h_i \cdot \exp\left(-\frac{||x(t) - x_i(t)||^2}{2\sigma_i^2(t)}\right) + b. \tag{11}$$

where $N(t)$ is the number of neurons, $x_i(t)$ is the center of neurons and $\sigma_i(t)$ the radius of neurons, at time “t”. In order to minimize equation 6 we propose a regularization operator based on SVM philosophy. We enforce flatness in feature space, as described in section 2, using the regularization operator $||\hat{P}f||^2 \equiv ||\omega||^2$, thus we get:

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} \sum_{i,j=1}^{N(t)} h_i h_j k(x_i, x_j). \tag{12}$$

We assume that $R_{emp} = (y - f(x))^2$ we minimize equation 12 adjusting the centers and radius (gradient descend method $\Delta\chi = -\eta \frac{\partial R[f]}{\partial \chi}$, using Chernoff bounds):

$$\Delta x_i = -2\frac{\eta}{\sigma_i}(x - x_i)h_i(f(x) - y)k(x, x_i) + \alpha \sum_{i,j=1}^{N(t)} h_i h_j k(x_i, x_j)(x_i - x_j). \tag{13}$$

and

$$\Delta h_i = \tilde{\alpha}(t) f(x_i) - \eta(f(x) - y)k(x, x_i). \tag{14}$$

where $\alpha(t), \tilde{\alpha}(t)$ are scalar-valued “adaptation gain”, related to a similar gain used in the stochastic approximation processes [6], as in these methods, it should decrease in time. The second summand in equation 13 can be evaluated in several regions inspired by the so called “divide-and-conquer” principle and used in unsupervised learning, i.e. competitive learning in self organizing maps [8] (SOM) or in SVMs experts [9]. This is necessary because of volatile nature of time series, i.e. stock returns, switch their dynamics among different regions, leading to gradual changes in the dependency between the input and output variables [6]. Thus the super-index in the latter equation is redefined as:

$$N_c(t) = \{s_i(t) : ||x(t) - x_i(t)|| \leq \rho\}. \tag{15}$$

the set of neurons close to current input.

[4]The principal feature of these algorithms is sequential adaptation of neural resources.

Table 1: Evolution of NRMSE(Normalized Root Mean Square Error), number of neurons and input space dimension. Xth-S detones the Xth-step prediction NRMSE on the test set.Each cell: neurons / dimension

Method	1st-S	25th-S	50th-S	75-th	100th-S
NAPA_PRED	1.1982	0.98346	0.97866	0.91567	0.90985
Standard SVM	0.7005	0.7134	0.7106	0.7212	0.7216
SVM_online	0.7182	0.71525	0.71522	0.72094	0.7127
NAPA_PRED	12/6	19/8	16/8	17/11	15/11
SVM_online	7/5	8/5	9/6	12/6	9/7

4 Experiments

The application of our network is to predict complex time series. We choose the high-dimensional chaotic system generated by the Mackey-Glass delay differential equation:

$$\frac{dx(t)}{dt} = -b \cdot x(t) + a \cdot \frac{x(t-\tau)}{1 + x^{10}(t-\tau)} \quad . \tag{16}$$

with $b = 0.1$, $a = 0.2$ and delay $t_d = 17$. This equation was originally presented as a model of blood regulation [10] and became popular in modelling time series benchmark. We add two modifications to equation 16: Zero-mean gaussian noise with standard deviation equal to 1/4 of the standard deviation of the original series and dynamics changes randomly in terms of delay (between 100-300 time steps) $t_d = 17, 23, 30$. We integrated the chaotic model using MatLab software on a Pentium III at 850MHZ obtaining 2000 patterns. For our comparison we use 100 prediction results from SVM_online (presented in this paper), standard SVM (with ϵ-insensitive loss) and NAPA_PRED (RAN algorithm using matrix decompositions being one of the best on-line algorithms to date[6]). Clearly there´s a remarkable difference between previous on-line algorithm and SVM philosophy. Standard SVM and SVM_online achieve similar results for this set of data at the beginning of the process. In addition, there´s is noticeable improvement in the last iterations because of the volatile nature of the series. The change in time delay t_d, leads to gradual changes in the dependency between the input and output variables and, in general, it´s hard for a single model including SVMs to capture such a dynamic input-output relationship inherent in the data. Focussing our attention on the on-line algorithm, we observe the better performance of the new algorithm as is shown in Table 1.

5 Conclusions

Based on SRM and the principle of "divide and conquer" , a new online algorithm is developed by combining SVM and SOM using a resource allocating network and matrix decompositions. Minimizing the regularizated risk functional, using an operator the enforce flatness in feature space, we build a hybrid model that achieves high prediction performance, comparing with the previous on-line algorithms for time series forecasting. This performance is similar to the one achieve by SVM but with lower computational time demand, essential feature in real-time systems. The benefits of SVM for regression choice consist in solving a -uniquely solvable- quadratic optimization problem, unlike the general RBF networks, which requires suitable non-linear optimization with danger of getting stuck in local minima. Nevertheless the RBF networks used in this paper, with the help of various techniques obtain high performance, even under extremely volatile conditions, since the level of noise and the change of delay operation mode applied to the chaotic dynamics was rather high.

References

[1] Vapnik, V.: The Nature of Statistical Learning Theory. 2nd Edition. Springer (1999)

[2] Tikhonov, A.N., Arsenin, V.Y.:, Solutions of Ill-Posed Problems, Winston. Washington D.C., U.S.A. Berlin Heidelberg New York (1997) 415–438

[3] Muller, K.R, Smola A.J., Ratsch, G., Scholkopf, B., Kohlmorgen, J: Using Support Vector Machines for time series prediction. Advances in kernel Methods-Support Vector Learning, MIT Press,(1999) 243–254

[4] Smola, A.J., Scholkopf, B., Muller, K.R: The connection between regularization operators and support vector kernels. Neural Networks, 11, 637–649

[5] Platt, J.: A resource-allocating network for function interpolation. Neural Computation, 3, (1991) 213–225

[6] Górriz, J.M. : Algorítmos Híbridos para la Modelización de Series Temporales con Técnicas AR-ICA. Thesis, University of Cádiz (2003)

[7] Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London , num 209, vol A, 415–446 (1909)

[8] Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE, num 9, vol 78, 1464–1480 (1990)

[9] Cao, L.: Support Vector Machines Experts for Time Series Forecasting. Neurocomputing, vol 51, 321–339 (2003)

[10] Mackey, M.C., Glass, L., Science, 197 287–289 (1977)