ELSEVIER Computational Statistics & Data Analysis 38 (2002) 421–432

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Regularization and statistical learning theory for data analysis

Theodoros Evgeniou[a,*], Tomaso Poggio[b], Massimiliano Pontil[b], Alessandro Verri[c]

[a] *Technology Management, INSEAD, Fointainebleau 77305, France*
[b] *Center for Biological and Computational Learning, MIT, 45 Carleton St., Cambridge, MA, 02142, USA*
[c] *INFM-DISI, Università di Genova, Via Dodecaneso 35, 16146 Genova, Italy*

Abstract

Problems of data analysis, like classification and regression, can be studied in the framework of Regularization Theory as ill-posed problems, or through Statistical Learning Theory in the learning-from-example paradigm. In this paper we highlight the connections between these two approaches and discuss techniques, like support vector machines and regularization networks, which can be justified in this theoretical framework and proved to be useful in a number of image analysis applications.

Keywords: Statistical learning theory; Regularization theory; Support vector machine; Regularization networks; Image analysis applications

1. Introduction

The goal of this paper is to provide a brief introduction to the study of *supervised learning* within the framework of Regularization Theory and Statistical Learning Theory. For a detailed review of the theoretical aspects of this subject see Evgeniou et al. (1999). In supervised learning or *learning-from-examples* a machine is trained, instead of programmed, to perform a given task on a number of input–output pairs. According to this paradigm, training means choosing a function which best describes the relation between the inputs and the outputs. In functional analysis, the choice of

* Corresponding author.

0167-9473/02/$ - see front matter
PII: S 0167-9473(01)00069-X

the optimal function is an example of an ill-posed problem which can be addressed with the machinery of Regularization Theory. In a probabilistic setting, a second fundamental problem, studied by Statistical Learning Theory, is how well the chosen function generalizes, or how well it estimates the output for new inputs.

This paper is organized as follows. We first outline the key concepts of Regularization and Statistical Learning Theory in Sections 2 and 3, respectively. We then present in Section 4 Regularization Networks and Support Vector Machines (SVMs), two important learning techniques which can be theoretically justified within the proposed framework. In Section 5 we discuss implementation issues and a few applications of SVMs which recently gained much attention from the image analysis community. Finally, we draw our conclusions in Section 6.

2. Regularization theory

We consider techniques which lead to solutions of the form

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i),$$

where the $\mathbf{x}_i$, $i = 1, \ldots, \ell$ are the input examples, K a certain symmetric positive definite function named kernel, and c_i a set of ℓ parameters to be determined from the examples. The function $\hat{f}$ is found by minimizing functionals of the type

$$\Psi[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda ||f||_K^2,$$

where f belongs to some suitable Hilbert space $\mathscr{H}$, V is a *loss function* which measures the goodness of the predicted output $f(\mathbf{x}_i)$ with respect to the given output y_i, $||f||_K^2$ a *smoothness* term which can be thought of as a norm in the Reproducing Kernel Hilbert Space defined by the kernel K and λ a positive parameter which controls the relative weight between the data and the smoothness terms. The choice of the loss function determines different learning techniques, each leading to a different learning algorithm for computing the coefficients c_i.

The inclusion of the $\lambda ||f||_K^2$ factor above is central in Regularization theory and, as we discuss below, it is also central in Statistical Learning Theory.

The minimization of the functional

$$\frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)),$$

for $f \in \mathscr{H}$, which might seem a more straightforward approach, is an ill-posed problem,[1] because it admits an infinite number of solutions. Regularization theory (see Tikhonov and Arsenin, 1977; Morozov, 1984, for example) provides a framework

[1] A well-posed problem (in the sense of Hadamard, (Tikhonov and Arsenin, 1977)), is a problem for which a solution (a) exists, (b) is unique, and (c) depends continuously on the data. A problem for which at least one of the above conditions does not hold is *ill-posed.*

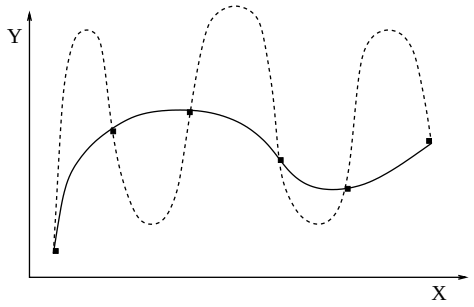

Fig. 1. Both the dashed and solid curves interpolate the data (denoted by the filled squares), but with different degree of smoothness.

for restoring well-posedness by adding appropriate constraints on the solution. The smoothness term above is a classical example of regularizing constraint which enforces uniqueness by penalizing functions with wild oscillation (see Fig. 1) and effectively restricting the space of admissible solutions. This ensures that the regularized solution has good predictive capabilities. This issue, however, needs a probabilistic treatment that is not studied with Regularization Theory. A well-founded theoretical framework within which the generalization capabilities of data analysis and supervised learning methods can be studied is Statistical learning theory (Vapnik, 1998), that we now briefly overview.

3. Statistical learning theory

We first formulate the problem of supervised learning in a statistical setting distinguishing between empirical and structural risk minimization and introducing the key concept of capacity control.

3.1. Empirical risk minimization

We consider two sets of random variables $\mathbf{x} \in X \subseteq \mathbb{R}^d$ and $y \in Y \subseteq \mathbb{R}$ related by a probabilistic relationship. The relationship is probabilistic because generally an element of X does not determine uniquely an element of Y, but rather a probability distribution on Y. This can be formalized assuming that an unknown probability distribution $p(\mathbf{x}, y)$ is defined over the set $X \times Y$. We are provided with *examples* of this probabilistic relationship, that is with a data set $D_\ell \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{\ell}$ called *training set*, obtained by sampling ℓ times the set $X \times Y$ according to $p(\mathbf{x}, y)$. The "problem of learning" consists in, given the data set D_ℓ, providing an *estimator*, that is a function $f : X \to Y$ able to predict a value y from any value of $\mathbf{x} \in X$.

In Statistical Learning Theory, the standard way to solve this problem consists in defining a *risk functional*, which measures the average amount of error or risk associated with an estimator, and then looking for the estimator with the lowest risk. If $V(y, f(\mathbf{x}))$ is the loss function measuring the error we make when we predict y

by $f(\mathbf{x})$, then the average error, the so called *expected risk*, is

$$I[f] \equiv \int_{X,Y} V(y, f(\mathbf{x}))\, p(\mathbf{x}, y)\, \mathrm{d}\mathbf{x}\, \mathrm{d}y.$$

We assume that the expected risk is defined on a "large" class of functions $\mathscr{H}$ and we will denote by f_0 the function which minimizes the expected risk in $\mathscr{H}$. The function f_0, our ideal estimator, is often called the *target* function. This function cannot be found in practice, because the probability distribution $p(\mathbf{x}, y)$ that defines the expected risk is unknown, and only a sample of it, the data set D_ℓ, is available. To overcome this shortcoming we need an *induction principle* that we can use to "learn" from the limited number of training data we have. Statistical Learning Theory, as developed by Vapnik (1998), builds on the so-called *empirical risk minimization* (ERM) induction principle. The ERM method consists in using the data set D_ℓ to build a stochastic approximation of the expected risk, which is usually called the *empirical risk*, defined as

$$I_{\mathrm{emp}}[f; \ell] = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)).$$

Straightforward minimization of the empirical risk in $\mathscr{H}$ can be problematic. First, as we have already discussed in the previous section, it is an *ill-posed* problem. Second, it can lead to *overfitting*, meaning that although the minimum of the empirical risk can be very close to zero, the expected risk which is what we are really interested in can be very large.

Statistical Learning Theory provides probabilistic bounds on the distance between the empirical and expected risk of any function (therefore including the minimizer of the empirical risk in a function space that can be used to control overfitting). The bounds involve the number of examples ℓ and the *capacity* h of the function space, a quantity measuring the "complexity" of the space. Appropriate capacity quantities are defined in the theory, the most popular one being the VC-dimension (Vapnik and Chervonenkis, 1971) or scale sensitive versions of it (Kearns and Shapire, 1994; Alon et al., 1993). The bounds have the following general form: with probability at least η

$$I[f] < I_{emp}[f] + \varphi\left(\sqrt{\frac{h}{\ell}}, \eta\right), \tag{1}$$

where h is the capacity and φ an increasing function of h/ℓ and η. For more information and the exact forms of the function φ we refer the reader to Vapnik and Chervonenkis (1971), Vapnik (1998) and Alon et al. (1993). Intuitively, if the capacity of the function space in which we perform empirical risk minimization is very large and the number of examples is small, then the distance between the empirical and expected risk can be large and overfitting is very likely to occur.

3.2. Structural risk minimization

Since the space $\mathscr{H}$ is usually very large (e.g. $\mathscr{H}$ could be the space of square integrable functions), one typically considers a smaller hypothesis space H. Moreover, inequality (1) suggests an alternative method for achieving good generalization: instead of minimizing the empirical risk, find the best trade off between the empirical risk and the *complexity of the hypothesis space* measured by the second term in the r.h.s. of inequality (1). This observation leads to the method of *Structural Risk Minimization* (*SRM*).

The idea of SRM is to define a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \cdots \subset H_M$, where each hypothesis space H_m has finite capacity h_m and larger than that of all previous sets, that is: $h_1 \leqslant h_2, \ldots, \leqslant h_M$. For example H_m could be the set of polynomials of degree m, or a set of splines with m nodes, or some more complicated nonlinear parameterization. Using such a nested sequence of increasingly more complex hypothesis spaces, SRM consists of choosing the minimizer of the empirical risk in the space H_{m^*} for which the bound on the *structural risk*, as measured by the right hand side of inequality (1), is minimized. Further information about the statistical properties of SRM can be found in Devroye et al. (1996) and Vapnik (1998).

To summarize, the problem of learning from examples can be solved in three steps: (a) define a loss function $V(y, f(\mathbf{x}))$ measuring the error of predicting the output of input $\mathbf{x}$ with $f(\mathbf{x})$ when the actual output is y; (b) define a nested sequence of hypothesis spaces H_m, $m = 1, \ldots, M$ whose capacity is an increasing function of m; (c) minimize the empirical risk in each of H_m and choose, among the solutions found, the one with the best trade off between the empirical risk and the capacity as given by the right hand side of inequality (1).

3.3. Capacity control in reproducing Kernel Hilbert spaces

Insight about the connection between Regularization Theory and Statistical Learning Theory in the problem of learning from examples can be gained through the concept of Reproducing Kernel Hilbert Space (RKHS) (Wahba, 1990). A RKHS is a Hilbert space of functions f of the form $f(\mathbf{x}) = \sum_{n=1}^{N} a_n \phi_n(\mathbf{x})$, where $\{\phi_n(\mathbf{x})\}_{n=1}^{N}$ is a set of given, linearly independent basis functions and N can be possibly infinite. A RKHS is equipped with a norm which is defined as

$$||f||_K^2 = \sum_{n-1}^{N} \frac{a_n^2}{\lambda_n},$$

where $\{\lambda_n\}_{n=1}^{N}$ is a decreasing, positive sequence of real values whose sum is finite. The constants λ_n and the basis functions $\{\phi_n\}_{n=1}^{N}$ define the symmetric positive definite kernel function

$$K(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{N} \lambda_n \phi_n(\mathbf{x}) \phi_n(\mathbf{y}).$$

A nested sequence of spaces of functions in the RKHS can be constructed by bounding the norm of in that space. This can be done by defining a set of constants $A_1 < A_2 \cdots < A_M$ and considering spaces of the form

$$H_m = \{f \in \text{RKHS}: ||f||_K \leqslant A_m\}.$$

It can be shown that the capacity of the hypothesis spaces H_m is an increasing function of A_m (see for example Evgeniou et al., 1999). Therefore, the solution of the learning problem is found by solving, for each A_m, the following optimization problem

$$\begin{aligned} \min_f \quad & \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) \\ \text{subject to} \quad & ||f||_K \leqslant A_m, \end{aligned}$$

and choosing, among the solutions found for each A_m, the one minimizing the structural risk.

4. Learning machines

The implementation of the SRM method described above is not practical because it requires to look for the solution of a large number, in principle infinite, of constrained optimization problems. Before presenting two important learning techniques, which can be theoretically justified within the proposed framework, we show how this difficulty can be overcome.

4.1. Learning as functional minimization

Instead of looking for the solution of many optimization problems, we search for the minimum of

$$H[f] = \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda||f||_K^2. \tag{2}$$

The functional $H[f]$ contains both the empirical risk and the norm (complexity or smoothness) of f in the RKHS, similarly to functionals considered in Regularization Theory (Tikhonov and Arsenin, 1977). Within the Statistical Learning Theory framework, the *regularization parameter* λ can be seem as a penalty for functions with high capacity: the larger λ, the smaller the RKHS norm of the solution will be. This same factor also transforms an ill-posed problem into a well posed one, as discussed in Section 2.

When implementing SRM, the key issue is the choice of the hypothesis space, i.e. the parameter H_m where the structural risk is minimized. In the case of the functional of equation (2), the key issue becomes the choice of the regularization parameter λ. These two problems, as discussed in Evgeniou et al. (1999), are related, and the SRM method can in principle be used to choose λ (Vapnik, 1998). In practice, instead of using SRM other methods are used such as cross-validation (Wahba,

1990), Generalized Cross Validation, Finite Prediction Error and the MDL criteria (see Vapnik (1998) for a review and comparison).

An important feature of the minimizer of $H[f]$ is that, independently on the loss function V, the minimizer has the same general form (Wahba, 1990)

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i). \tag{3}$$

Notice that Eq. (3) establishes a representation of the function f as a linear combination of kernels centered in each data point. Using different kernels we get functions such as Gaussian radial basis functions ($K(\mathbf{x},\mathbf{y}) = \exp(-\beta||\mathbf{x}-\mathbf{y}||^2)$), or polynomials of degree d ($K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^d$) [5, 18].

We now turn to discuss a few learning techniques based on the minimization of functionals of the form (2) by specifying the loss function V. In particular, we will consider Regularization Networks and Support Vector Machines (SVM), a learning technique which has recently been proposed for both classification and regression problems (see Vapnik (1998) and references therein):

- Regularization Networks:

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2,$$

- SVM Classification:

$$V(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+, \tag{4}$$

where $|x|_+ = x$ if $x > 0$ and zero otherwise.

SVM Regression:

$$V(y_i, f(\mathbf{x}_i)) = |y_i - f(\mathbf{x}_i)|_\varepsilon, \tag{5}$$

where the function $|\cdot|_\varepsilon$, called ε-insensitive loss, is defined as:

$$|x|_\varepsilon \equiv \begin{cases} 0 & \text{if } |x| < \varepsilon \\ |x| - \varepsilon & \text{otherwise.} \end{cases}$$

We now briefly discuss each of these three techniques.

4.2. Regularization networks

The approximation scheme that arises from the minimization of the quadratic functional

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - f(\mathbf{x}_i))^2 + \lambda||f||_K^2 \tag{6}$$

for a fixed λ is a special form of regularization. It is possible to show (see for example Girosi et al., 1995) that the coefficients c_i of the minimizer of (6) in Eq. (3) satisfy the following linear system of equations:

$$(G + \lambda I)\mathbf{c} = \mathbf{y},$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i, \quad (\mathbf{c})_i = c_i, \quad (G)_{ij} = K(\mathbf{x}_i, \mathbf{x}_j).$$

Since the coefficients c_i satisfy a linear system, Eq. (3) can be rewritten as:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i b_i(\mathbf{x}), \tag{7}$$

with $b_i(\mathbf{x}) = \sum_{j=1}^{\ell} (G + \lambda I)^{-1}_{ij} K(\mathbf{x}_i, \mathbf{x})$. Eq. (7) gives the dual representation of RN. Notice the difference between Eqs. (3) and (7): in the first one the coefficients c_i are learned from the data while in the second one the bases functions b_i are learned, the coefficient of the expansion being equal to the output of the examples. We refer to Girosi et al. (1995) for more information on the dual representation.

4.3. Support vector machines

We now discuss support vector machines (SVM) (Cortes and Vapnik, 1995; Vapnik, 1998). We distinguish between real output (regression) and binary output (classification) problems. The method of SVM regression corresponds to the following minimization:

$$\operatorname*{Min}_f \frac{1}{\ell} \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i)|_\varepsilon + \lambda ||f||_K^2,$$

while the method of SVM classification corresponds to:

$$\operatorname*{Min}_f \frac{1}{\ell} \sum_{i=1}^{\ell} |1 - y_i f(\mathbf{x}_i)|_+ + \lambda ||f||_K^2.$$

A remarkable property of SVMs is that loss functions (5) and (4) lead to *sparse* solutions. This means that, unlike in the case of Regularization Networks, typically only a small fraction of the coefficients c_i in Eq. (3) are nonzero. The data points $\mathbf{x}_i$ associated with the nonzero c_i are called *support vectors*. If all data points which are not support vectors were to be discarded from the training set the same solution would be found. In this context, an interesting perspective on SVM is to consider its information compression properties. The support vectors represent the most informative data points and compress the information contained in the training set: for the purpose of, say, classification only the support vectors need to be stored, while all other training examples can be discarded. This, along with some geometric properties of SVMs such as the interpretation of the RKHS norm of their solution as the inverse of the *margin* (Vapnik, 1998), is a key property of SVM and might explain why this technique works well in many practical applications.

5. Algorithms and applications

In this section we discuss some implementation issues and give a brief overview of applications of the learning techniques discussed in the previous section in the area

of image analysis. Since algorithms for regularization networks are well established (see for instance Trefethen and Bau, 1998), we concentrate on SVMs and, from the algorithmic viewpoint, we consider the case of classification.

5.1. A decomposition method

We begin by introducing the variables $\alpha_i = y_i c_i$. Eq. (3) can thus be rewritten as $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$. The α_i (Vapnik, 1998) are the solution of the following Quadratic Programming (QP) problem

Problem P1

$$\begin{aligned} &\text{Maximize} && W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to} && 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, \ell \end{aligned}$$

with $C = 1/(2\lambda)$. This problem is non trivial since the size of matrix of the quadratic form is $\ell \times \ell$ and the matrix is dense. When ℓ is not too large (up to a few hundred), one can solve problem P1 by using standard optimization algorithms (see Vanderbei (1997), for example). However, in many practical applications ℓ can be of the order of several thousands or more. In this case one needs to resort to more sophisticated techniques.

Among the many possible strategies, we illustrate a method for solving problem P1 which was introduced in Osuna et al. (1997). The method finds the solution by solving a sequence of simpler problems derived from problem P1. In each subproblem, one maximizes $W(\alpha)$ with respect to a subset of components of $\boldsymbol{\alpha}$, while keeping the other components constant. More precisely, we partition the set of index $I = \{1, \ldots, \ell\}$ in two sets: B, the *working set*, and its complement N in I. Likewise, we decompose $\boldsymbol{\alpha}$ in the two vectors $\boldsymbol{\alpha}_B$ and $\boldsymbol{\alpha}_N$. We then look at the function $W(\boldsymbol{\alpha})$ as a function of $\boldsymbol{\alpha}_B$ only

$$W(\boldsymbol{\alpha}_B; \boldsymbol{\alpha}_N) = \sum_{i \in B}^{\ell} \alpha_i h_i - \frac{1}{2} \sum_{i,j \in B} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + a,$$

where

$$h_i = 1 - \sum_{j \in N}^{\ell} \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

and a is a constant. We consider the following subproblem

Problem P2

$$\begin{aligned} &\text{Maximize} && W(\boldsymbol{\alpha}_B; \boldsymbol{\alpha}_N) \\ &\text{subject to} && 0 \leqslant \alpha_i \leqslant C, \quad i \in B \end{aligned}$$

The method works in three steps:

1. Randomly select g index in I to form the set B and set $\boldsymbol{\alpha}_N = 0$.
2. Solve problem P2.
3. Define $W_j(\alpha) = \partial W/\partial \alpha_j$. Look for an index $j \in N$ such that one of the following is true:
   - $W_j < 1$ and $\alpha_j = 0$
   - $W_j = 1$ and $\alpha_j \in \{0, C\}$
   - $W_j > 1$ and $\alpha_j > 0$,

If such an index j exists, swap j with any index in B and go back to step 2. Otherwise stop. The vector $(\boldsymbol{\alpha}_B, \boldsymbol{\alpha}_N)$ is the solution of problem P1.

For a proof of the convergence of the algorithm see Osuna et al. (1997). Software implementing various versions of this algorithm are available on internet.[2] Among the other methods, it is worth mentioning the technique on sequential updates of the solution developed in Platt (1998). We conclude by observing that a very similar algorithm can be derived for the case of regression, see for instance Vapnik (1998).

5.2. Using SVMs for image analysis

SVMs have been used as the core classifiers of vision systems for example for identifying faces (Osuna et al., 1997), pedestrians (Oren et al., 1997), and objects (Papageorgiou et al., 1998), for appearance-based 3-D object recognition (Pontil and Verri, 1998), and for recognizing dynamic events in image sequences (Pittore et al., 2000). In all these cases the proposed vision systems were able to deal with *objects* difficult to model due to significant variety of geometry, color, texture, and viewing conditions.

In what follows we briefly review a real-world application in which SVMs have been used along with machine vision techniques for automatic fish grading by weight. The motivation comes from the increasing push of the fish retail market toward standard-weight packages with minimal waste implied by filleting and portioning, and by the fairly inaccurate grading which can be obtained by the mechanical graders currently in use. The key idea of the system (Odone et al., 1998) is to learn from examples the shape-weight relation for each specific batch of fish to grade.

The two major software components are a real-time vision module and a machine learning module which implements an SVM for regression. The vision module detects fish sliding through a transparent channel, and acquires side and top views simultaneously when the fish appears in the middle of the side image. The system then takes shape measurements from both views. In the training set the vision system is used to acquire measurements from a number of fish and produce the training set. The SVM module uses the training set to infer an optimal approximation function, which describes the relation shape-weight for that particular population of fish. When the whole fish pool is graded, this function is used for estimate the weight of all fish (3 fish per second).

[2] For example, a fast implementation, due to Ryan Rifkin, can be downloaded from http://five-percent-nation.mit.edu/PersonalPages/rif/SvmFu.

A SVM for regression is trained with linear and quadratic kernel using the 13 shape measurements made available by the vision module. Both kernels seem appropriate for the limited weight-length ranges envisaged for batch grading on fish farms (Odone et al., 2001). In off-line and more recent on-line experiments, the system prototype reached 95% accuracy in weight estimation of out of sample fish.

6. Conclusions

Regularization and Statistical Learning theory provide a framework within which data analysis tools can be developed and analyzed. Both theories suggest that learning and data analysis methods should not focus on the minimization of an empirical error over existing data. Such a minimization is both ill-posed and not necessarily leading to models with good predictive capabilities. Instead, both theories suggest that one needs to minimize a combination of the empirical error over existing data *and* a penalty factor that penalizes solutions that are too complex: the smoothness or capacity of the functions considered needs to be controlled. Different choices of loss function (measurements of empirical error) and of the penalty factor lead to different learning (data analysis) methods. Two important methods developed with appropriate choices of these two terms are Regularization Networks and Support Vector Machines.

References

Alon, N., Ben-David, S., Cesa-Bianchi, N., Haussler, D., 1993. Scale-sensitive dimensions, uniform convergence, and learnability. Symposium on Foundations of Computer Science.

Cortes, C., Vapnik, V., 1995. Support vector networks. Mach. Learn. 20, 1, 25.

Devroye, L., Györfi, L., Lugosi, G., 1996. A Probabilistic Theory of Pattern Recognition. In: I. Karatzas and M. Yor (Eds.), Applications of mathematics, Vol. 31. Springer, New York.

Evgeniou, T., Pontil, M., Poggio, T., 1999. A unified framework for Regularization Networks and Support Vector Machines. A.I. Memo No. 1654, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Girosi, F., Jones, M., Poggio, T., 1995. Regularization theory and neural networks architectures. Neural Comput. 7, 219–269.

Kearns, M., Shapire, R., 1994. Efficient distribution-free learning of probabilisitic concepts. J. Comput. Syst. Sci. 48 (3), 464–497.

Morozov, V.A., 1984. Methods for solving incorrectly posed problems. Springer, Berlin.

Odone, F., Trucco, E., Verri, A., 1998. Visual Learning of Weight from Shape Using Support Vector Machines. Proceedings of the British Machine Visual Conference, Southampton, UK.

Odone, F., Trucco, E., Verri, A., 2001. A Trainable System for Grading Fish from Images. Appl. Artif. Intell., Special issue on Machine Learning in Computer Vision, 15 (8), p. 735–745.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T., 1997. Pedestrian Detection Using Wavelet Templates. Proceedings of CVPR'97, Puerto Rico, pp. 193–199.

Osuna, E., Freund, R., Girosi, F., 1997. An improved training algorithm for support vector machines. In: J. Principle, L. Gile, N. Morgan, and E. Wilson (Eds.), IEEE Workshop on Neural Networks and Signal Processing. Amelia Island, FL., p. 276–285.

Papageorgiou, C., Oren, M., Poggio, T., 1998. A General Framework for Object Detection. Proceedings of the International Conference on Computer Vision. Bombay, India.

Pittore, M., Campani, M., Verri, A., 2000. Learning to recognize visual dynamic events from examples. Int. J. Comput. Vis. 38, 35–44.

Platt, J.C., 1998. Sequential minimal imization: A fast algorithm for training support vector machines. Technical Report MST-TR-98-14, Microsoft Research, April.

Pontil, M., Verri, A., 1998. Support Vector Machines for 3-D Object Recognition. IEEE Trans. Patt. Anal. Machine Intell. 20, 637–646.

Tikhonov, A.N., Arsenin, V.Y., 1977. Solutions of Ill-posed Problems. Winston, W.H, Washington, D.C.

Trefethen, L.N., Bau, D., 1998. Numerical Linear Algebra. Philadelphia: Series in Applied Mathematics, Vol. 11. SIAM.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley, New York.

Vapnik, V.N., Chervonenkis, A.Y., 1971. On the Uniform Convergence of Relative Frequencies of events to their probabilities. Th. Prob. Appl. 17 (2), 264–280.

Wahba, G., 1990. Splines Models for Observational Data. Series in Applied Mathematics, Philadelphia, Vol. 59. SIAM.

Vanderbei, R.J., 1997. LOQO User's Manual. Princeton University.