??, ??, 1–6 (??)

Statistical Learning Theory: a Primer

THEODOROS EVGENIOU AND MASSIMILIANO PONTIL*

Center for Biological and Computational Learning, MIT

Received ??. Revised ??.

Abstract. In this paper we first overview the main concepts of Statistical Learning Theory, a framework in which learning from examples can be studied in a principled way. We then briefly discuss well known as well emerging learning techniques such as Regularization Networks and Support Vector Machines which can be justified in term of the same induction principle.

Keywords: VC-dimension, Structural Risk Minimization, Regularization Networks, Support Vector Machines

1. Introduction

The goal of this paper is to provide a short introduction to Statistical Learning Theory (SLT) which studies problems and techniques of *supervised learning*. For a more detailed review of SLT see [5]. 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. The central question of SLT is how well the the chosen function generalizes, or how well it estimates the output for previously unseen inputs.

We will consider techniques which lead to solution of the form

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

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

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

where 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 term. The choice of the loss function determines different learning techniques, each leading to a different learning algorithm for computing the coefficients c_i.

The rest of the paper is organized as follows. Section 2 presents the main idea and concepts in the theory. Section 3 discusses Regularization Networks and Support Vector Machines, two important techniques which produce outputs of the form of equation (1).

2. Statistical Learning Theory

We consider two sets of random variables $\mathbf{x} \in X \subseteq R^d$ and $y \in Y \subseteq R$ related by a probabilistic relationship. The relationship is probabilistic because generally an element of X does not

* This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research is sponsored by grants from the National Science Foundation, ONR and Darpa. Additional support is provided by Eastman Kodak Company, Daimler-Chrysler, Siemens, ATR, AT&T, Compaq, Honda R&D Co., Ltd., Merrill-Lynch, NTT and Central Research Institute of Electric Power Industry.

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 $f : X \to Y$, that can be used, given any value of $\mathbf{x} \in X$, to predict a value y. For example X could be the set of all possible images, Y the set $\{-1, 1\}$, and $f(\mathbf{x})$ an *indicator function* which specifies whether image $\mathbf{x}$ contains a certain object ($y = 1$), or not ($y = -1$) (see for example [12]). Another example is the case where $\mathbf{x}$ is a set of parameters, such as pose or facial expressions, y is a motion field relative to a particular reference image of a face, and $f(\mathbf{x})$ is a regression function which maps parameters to motion (see for example [6]).

In SLT, the standard way to solve the learning 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:

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

We assume that the expected risk is defined on a "large" class of functions $\mathcal{F}$ and we will denote by f_0 the function which minimizes the expected risk in $\mathcal{F}$. The function f_0 is our ideal estimator, and it 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. SLT, as developed by Vapnik [15], 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

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

Straight minimization of the empirical risk in $\mathcal{F}$ can be problematic. First, it is usually an *ill-posed* problem [14], in the sense that there might be many, possibly infinitely many, functions minimizing the empirical risk. Second, it can lead to *overfitting*, meaning that although the minimum of the empirical risk can be very close zero, the expected risk – which is what we are really interested in – can be very large.

SLT 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 [16] or scale sensitive versions of it [9], [1]. The bounds have the following general form: with probability at least η

$$I[f] < I_{emp}[f] + \Phi(\sqrt{\frac{h}{\ell}}, \eta). \tag{2}$$

where h is the capacity, and Φ an increasing function of $\frac{h}{\ell}$ and η. For more information and for exact forms of function Φ we refer the reader to [16], [15], [1]. 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.

Since the space $\mathcal{F}$ is usually very large (i.e. $\mathcal{F}$ could be the space of square integrable functions), one typically considers smaller hypothesis spaces $\mathcal{H}$. Moreover, inequality (2) 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 (2). This observation leads to the method of *Structural Risk Minimization (SRM).*

The idea of SRM is to define a nested sequence of hypothesis spaces $H_1 \subset H_2 \subset \ldots \subset H_{\text{M}}$, where each hypothesis space H_m has finite capacity h_m and larger than that of all previous sets, that is: $h_1 \leq h_2, \ldots, \leq h_{\text{M}}$. 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 more and 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 (2), is minimized. Further information about the statistical properties of SRM can be found in [3], [15].

To summarize, in SLT the problem of learning form examples is solved in three steps: (a) we 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) we define a nested sequence of hypothesis spaces $H_m, m = 1, \ldots, M$ whose capacity is an increasing function of m; (c) we 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 (2).

3. Learning machines

3.1. Learning as functional minimization

We now consider hypothesis spaces which are subsets of a Reproducing Kernel Hilbert Space (RKHS) [17]. A RKHS is a Hilbert space of functions f of the form $f(\mathbf{x}) = \sum_{n=1}^{\text{N}} a_n \phi_n(\mathbf{x})$, where $\{\phi_n(\mathbf{x})\}_{n=1}^{\text{N}}$ 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:

$$\|f\|_{\text{K}}^2 = \sum_{n=1}^{\text{N}} \frac{a_n^2}{\lambda_n},$$

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

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

A nested sequence of spaces of functions in the RKHS can be constructed by bounding the RKHS norm of functions in the space. This can be done by defining a set of constants $A_1 < A_2 < \ldots < A_{\text{M}}$ and considering spaces of the form:

$$H_m = \{f \in RKHS : \|f\|_{\text{K}} \leq A_m\}$$

It can be shown that the capacity of the hypothesis spaces H_m is an increasing function of A_m (see for example [5]). According to the scheme given at the end of section 2, the solution of the learning problem is found by solving, for each A_m, the following optimization problem:

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

and choosing, among the solutions found for each A_m, the one with the best trade off between empirical risk and capacity, i.e. the one which minimizes the bound on the structural risk as given by inequality (2).

The implementation of the SRM method described above is not practical because it requires to look for the solution of a large number constrained optimization problems. This difficulty is overcome by searching for the minimum of:

$$H[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\text{K}}^2. \quad (3)$$

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 [14]. The *regularization parameter* λ penalizes functions with high capacity: the larger λ, the smaller the RKHS norm of the solution will be.

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 (3), the key issue becomes the choice of the regularization parameter λ. These two problems, as discussed in [5], are related, and the SRM method can in principle be used to choose λ [15]. In practice, instead of using SRM other methods are used such

as cross-validation ([17]), Generalized Cross Validation, Finite Prediction Error and the MDL criteria (see [15] 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 ([17])

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i), \tag{4}$$

Notice that equation (4) 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 ($K(\mathbf{x}, \mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^d$) [7], [15].

We now turn to discuss a few learning techniques based on the minimization of functionals of the form (3) 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 [15] and references therein):

- Regularization Networks

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2, \tag{5}$$

- SVM Classification

$$V(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+, \tag{6}$$

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

$$V(y_i, f(\mathbf{x}_i)) = |y_i - f(\mathbf{x}_i)|_\epsilon, \tag{7}$$

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

$$|x|_\epsilon \equiv \begin{cases} 0 & \text{if } |x| < \epsilon \\ |x| - \epsilon & \text{otherwise.} \end{cases} \tag{8}$$

We now briefly discuss each of these three techniques.

3.2. Regularization Networks

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

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - f(\mathbf{x}_i))^2 + \lambda\|f\|_{\mathrm{K}}^2 \tag{9}$$

for a fixed λ is a special form of regularization. It is possible to show (see for example [7]) that the coefficients c_i of the minimizer of (9) in equation (4) satisfy the following linear system of equations:

$$(G + \lambda I)\mathbf{c} = \mathbf{y}, \tag{10}$$

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, equation (4) can be rewritten as:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i b_i(\mathbf{x}), \tag{11}$$

with $b_i(\mathbf{x}) = \sum_{j=1}^{\ell}(G + \lambda I)^{-1}_{ij} K(\mathbf{x}_i, \mathbf{x})$. Equation (11) gives the dual representation of RN. Notice the difference between equation (4) and (11): 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 [7] for more information on the dual representation.

3.3. Support Vector Machines

We now discuss Support Vector Machines (SVM) [2], [15]. We distinguish between real output (regression) and binary output (classification) problems. The method of SVM regression corresponds to the following minimization:

$$\mathrm{Min}_f \ \frac{1}{\ell}\sum_{i=1}^{\ell}|y_i - f(\mathbf{x}_i)|_\epsilon + \lambda\|f\|_{\mathrm{K}}^2 \tag{12}$$

while the method of SVM classification corresponds to:

$$\text{Min}_f \ \frac{1}{\ell}\sum_{i=1}^{\ell}|1-y_i f(\mathbf{x}_i)|_+ + \lambda\|f\|_K^2, \quad (13)$$

It turns out that for both problems (12) and (13) the coefficients c_i in equation (4) can be found by solving a Quadratic Programming (QP) problem with linear constraints. The size of the box is inversely proportional to the regularization parameter λ. The QP problem is non trivial since the size of matrix of the quadratic form is equal to $\ell \times \ell$ and the matrix is dense. A number of algorithms for training SVM have been proposed: some are based on a decomposition approach where the QP problem is attacked by solving a sequence of smaller QP problems [11], others on sequential updates of the solution [13].

A remarkable property of SVMs is that loss functions (7) and (6) 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 equation (4) 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* [15], is a key property of SVM and might explain why this technique works well in many practical applications.

3.4. Kernels and data representations

We conclude this short review with a short discussion on kernels and data representations. A key issue when using the learning techniques discussed above is the choice of the kernel K in equation (4). The kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ defines a dot product between the projections of the two inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, in the feature space (the features being $\{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots \phi_N(\mathbf{x})\}$ with N the dimensionality of the RKHS). Therefore its choice is closely related to the choice of the "effective" representation of the data, i.e. the image representation in a vision application.

The problem of choosing the kernel for the machines discussed here, and more generally the issue of finding appropriate data representations for learning, is an important and open one. The theory does not provide a general method for finding "good" data representations, but suggests representations that lead to "simple" solutions. Although there is not a general solution to this problem, a number of recent experimental and theoretical works provide insights for specific applications [4], [8], [10], [15].

References

1. N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Symposium on Foundations of Computer Science*, 1993.
2. C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.
3. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of mathematics. Springer, New York, 1996.
4. T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations for object detection using kernel classifiers. In *Proceedings ACCV*, page (to appear), Taiwan, January 2000.
5. T. Evgeniou, M. Pontil, and T. Poggio. A unified framework for regularization networks and support vector machines. A.I. Memo No. 1654, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1999.
6. T. Ezzat and T. Poggio. Facial analysis and synthesis using image-based models. In *Face and Gesture Recognition*, pages 116–121, 1996.
7. F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.
8. T. Jaakkola and D. Haussler. Probabilistic kernel regression models. In *Proc. of Neural Information Processing Conference*, 1998.
9. M. Kearns and R.E. Shapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and Systems Sciences*, 48(3):464–497, 1994.
10. A. Mohan. Robust object detection in images by components. Master's thesis, Massachusetts Institute of Technology, May 1999.
11. E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks and Signal Processing*, Amelia Island, FL, September 1997.

12. C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998.
13. J. C. Platt. Sequential minimal imization: A fast algorithm for training support vector machines. Technical Report MST-TR-98-14, Microsoft Research, April 1998.
14. A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.
15. V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
16. V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequences of events to their probabilities. *Th. Prob. and its Applications*, 17(2):264–280, 1971.
17. G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.