FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 16, Dec. 2003, 305-316

# Learning from Data Using Support Vector Machines

**Srdjan Stanković, Miloš Stanković, Maja Stanković, and Milan Milosavljević**

**Abstract:** Learning from data is discussed from the point of view of support vector machines. A specific algorithm solving the polychotomy problem is described. The methodology is illustrated on two complex examples taken from practice.

**Keywords:** Signal processing, learning, support vector machines.

## 1 Introduction

Science and engineering are still dominantly based on the *first-principle modeling*: one starts with basic scientific models (e.g. Newton's law, Maxwell's equations), and measurement data are used to verify the models and estimate some unknown model parameters. Similarly, *Fisher's paradigm* in statistics assumes that a priori knowledge encompasses the desired dependency up to the value of a finite number of parameters. *Digital signal processing methodology* is typically based on strict a priori assumptions concerning signal properties.

However, in many applications first-principle models are unknown and a priori assumptions are hard to verify. On the other hand, modern information technology provides often large amounts of data. The efforts are now oriented towards data mining and developing models *directly from data*. Many of the approaches to this problem are inspired by learning capabilities

Manuscript received June 22, 2003. An earlier version of this paper was presented at the 10th Telecommunications Forum TELFOR 2002, November 26-28, 2002, Belgrade, Serbia.

The authors are with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, Serbia and Montenegro (e-mail: stankovic@etf.bg.ac.yu).

of biological systems. *Learning from data* has become a popular term, covering, in particular, *adaptive methods* from signal processing , and, in general, all methods aimed at extracting information contained in the *training data* and using it to answer to *questions about new samples* (prediction). Specific learning tasks include typically the following: (1) classification (pattern recognition), (2) regression, or estimation of unknown functions from noisy samples, (3) probability density estimation.

Vapnik and Chervonenkis have proposed a new approach to the problem of learning from data, which has exerted a strong impact on diverse areas of science and engineering, including signal processing and statistical estimation, in general. This paper is aimed at presenting a very condensed outline of the main ideas of this theory, focused on the *support vector machines* applied to the classification problem, and to provide some practical illustrations obtained by solving complex problems in real life.

## 2 Learning from data

Learning from data samples consists of two main steps:

1. learning (estimating) unknown dependencies from samples;
2. predicting outputs for new inputs using dependencies obtained within (1).

These two steps correspond to two classical types of inference: induction, in the sense of going from particular samples to general models, and deduction (going from general models to new particular cases). In some cases it is not necessary to have models valid everywhere: it is sufficient to estimate outputs for a few particular inputs. Such an approach is called transduction, and can provide better estimates than the classical induction/deduction pair (see Fig. 1). The existing approaches to the problem of learning are based exclusively on induction, i.e. estimation of a function modeling the relationship between the given inputs and outputs. It subassumes generalization, and can be a difficult task, especially when the data set is small.

All learning methods use a priori knowledge in the form of a set of approximating functions $\boldsymbol{f}(\boldsymbol{x}, \omega)$, $\omega \in \Omega$, where $\Omega$ is a set of indexing parameters, and $\boldsymbol{x}$ the input vector. Approximation quality is measured by a loss function $L(y, \boldsymbol{f}(\boldsymbol{x}, \omega))$, where $y$ is the output. Its expectation using density $\boldsymbol{p}(\boldsymbol{x}, y)$ provides the expected risk

$$R(\omega) = \int L(y, \boldsymbol{f}(\boldsymbol{x}, \omega)) \boldsymbol{p}(\boldsymbol{x}, y) \mathrm{d}\boldsymbol{x} \mathrm{d}y$$

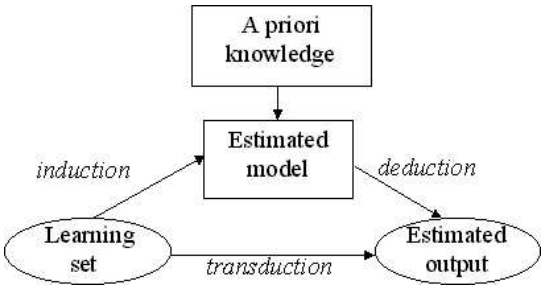


Fig. 1. Two inference principles: induction/deduction and transduction.

which is in practice known only at the learning data points. *Parametric methods utilize*, typically, small sets of approximating functions, while *flexible methods* utilize much broader sets of functions, but require additional constraints on a potential of a function. Each learning method encompasses also an *inductive principle* (or *inference method*), specifying what needs to be done, and a *learning method*, representing its implementation for a given set of approximating functions. For example, in the case of the inductive principle of *empirical risk minimization (ERM)*, the aim is to find a function minimizing the empirical risk (defined on the learning set), instead of the expected risk (true error). Depending on the loss function and the class of approximating functions, this induction principle can be implemented in diverse ways (e.g. linear regression, polynomial methods, neural networks). Parametric methods start from a fixed parametric model (e.g. polynomials of a given degree), while in the case of flexible methods it is necessary to estimate both the model complexity and the parameters. Parametric methods are often characterized by a high model error, or *bias*. Flexible methods attempt to reduce the bias by adapting the model complexity to the data.

## 3 Support Vector Machines

### 3.1 Separable case.

Suppose that the learning set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, $\boldsymbol{x} \in R^d$, $y \in \{+1, -1\}$ is linearly separable by hyperplanes $D(\boldsymbol{x}) = (\boldsymbol{w}\boldsymbol{x}) + \boldsymbol{w}_0$. Minimal distance between a hyperplane and the closest input vector is called *margin*, and is denoted by $C$. A separating hyperplane is optimal if $C$ is maximal. Supposing that $C$ exists, all learning samples satisfy the inequality

$$\frac{y_k D(\boldsymbol{x}_k)}{\| \boldsymbol{w} \|} \geq C, \quad k = 1, \ldots, n$$

where $y_k \in \{+1, -1\}$. The problem is to find the parameter vector $\boldsymbol{w}$ maximizing the margin C under the condition $C\boldsymbol{w} = 1$, which limits solutions. In such a way we obtain *canonical* hyperplanes satisfying the separation conditions

$$y_i[\boldsymbol{w}\boldsymbol{x}_i + w_0] \geq 1, \quad i = 1, \ldots, n$$

Maximization of $C$ reduces to minimization of $\eta(\boldsymbol{w}) = \| \boldsymbol{w} \|^2$ with respect to $\boldsymbol{w}$ and $w_0$. Samples inside the margin, i.e. samples for which the above inequality reduces to equality, are called *support vectors*.. Generalization capability of an optimal hyperplane is directly connected to the number of support vectors. If an optimal hyperplane can be constructed by using a small number of support vectors, its good generalization can be expected. It has been demonstrated that the complexity of the canonical hyperplanes (VC-dimension) can be controlled irrespective of the dimension of samples. The optimal hyperplane can be found by solving a quadratic optimization problem with linear inequality constraints. However, dimensionality of the problem can be reduced from the dimensionality of the input vector to the dimensionality of the sample set. Namely, one can define the dual problem leading to the following formulation: Find parameters $\alpha_i$, $i = 1, \ldots, n$, which maximize the function

$$Q(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \boldsymbol{x}_j) + \sum_{i=1}^{n} \alpha_i$$

under the constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \ldots, n$$

for the learning set $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$. The optimal hyperplane is then given by

$$D(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^* y_i (\boldsymbol{x}\boldsymbol{x}_i) + w_0^*$$

where $\alpha_i^*$ are the solutions of the dual problem, and $w_0^*$ is given by

$$w_0^* = y_s - \sum_{i=1}^{n} \alpha_i^* y_i (\boldsymbol{x}_i \boldsymbol{x}_s).$$

The last two expressions contain only scalar products $(\boldsymbol{x}\boldsymbol{x}')$ of the input vectors. Samples for which $\alpha_i^*$ are different from zero are support vectors. In practice, support vectors represent a small percentage of the input vectors. Notice that the last optimization problem can be solved by using standard quadratic programming methods.

### 3.2 Nonseparable case

Inthe nonseparable case, the aim is to make discrimination with the minimal number of errors, and, at the same time, to find the optimal hyperplane for the vectors that are correctly separated (vectors are called nonseparable if they fall within the margin, but they can be, at the same time, correctly separated). In order to solve this problem, nonnegative slack variables $\xi_i$ are introduced

$$y_i[(\boldsymbol{w}\boldsymbol{x}_i) + w_0] \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

Variables $\xi_i$ greater than zero correspond to nonseparable vectors, and those greater than 1 to the incorrectly classified ones. Then, we minimize

$$Q(\boldsymbol{w}) = \sum_{i=1}^{n} I(\xi_i > 0)$$

where $I(.)$ is the indicator function. This problem represents a complicated combinatorial optimization problem. However, if we take the following criterion as an approximation

$$\frac{C}{n} \sum_{i=0}^{n} \xi_i + \frac{1}{2} \parallel \boldsymbol{w} \parallel^2,$$

the coefficient $C$ allows achieving a trade off between complexity and the number of nonseparable samples. After defining the dual problem as above, we come to the following formulation: Find parameters $\alpha_i, i = 1, \ldots, n$, maximizing the function

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \boldsymbol{x}_j)$$

under the conditions

$$\sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \frac{C}{n}, \quad i = 1, \ldots, n$$

for a given learning set $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, and a regularization parameter C. The decision function has the same form as in the separable case. Coefficients $\alpha_i^*$ different from zero correspond to support vectors (which can be here inside the margin).

### 3.3 High-dimensional mapping and inner product kernels

Let $\boldsymbol{g}_j(\boldsymbol{x})$, $j = 1, \ldots, m$, represent a set of nonlinear transformation functions, mapping the inputs to the feature space. One may now construct hyperplanes in the *feature space*, when the discriminant function has the form

$$D(\boldsymbol{x}) = \sum_{j=1}^{m} w_j \boldsymbol{g}_j(\boldsymbol{x})$$

in which the number of terms at the right hand side depends on the feature space dimension, or

$$D(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i H(\boldsymbol{x}_i \boldsymbol{x})$$

where $H(\boldsymbol{x}, \boldsymbol{x}')$ is the inner product kernel. For given transformation functions, it has the form

$$H(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{m} \boldsymbol{g}_j(\boldsymbol{x}) \boldsymbol{g}_j(\boldsymbol{x}')$$

where m can be even infinite. In practice, typically, we have polynomials of $q$-th degree

$$H(\boldsymbol{x}, \boldsymbol{x}') = [(\boldsymbol{x}\boldsymbol{x}') + 1]^q$$

radial basis functions

$$H(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{\sigma^2}},$$

or neural networks

$$H(\boldsymbol{x}, \boldsymbol{x}') = \tanh[v(\boldsymbol{x}\boldsymbol{x}' + a)]$$

### 3.4 General strategy

General strategy of support vector machines used for classification are based on the above given main principles. One starts from the inner product kernel for high-dimensional mapping which replaces the inner product in the linear case, and, on the basis of the given procedures, obtains the optimal decision boundaries

$$D(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i H(\boldsymbol{x}_i \boldsymbol{x})$$

Parameters $\alpha_i^*$, $i = 1, \ldots, n$, result from maximization of

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i, \boldsymbol{x}_j)$$

under the constraints

$$\sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \frac{C}{n}, \quad i = 1, \ldots, n,$$

Support vector machines have several unique properties among the other statistical learning methods: *(1) estimation of probability density is not needed; (2) learning from data is done directly; (3) complexity does not depend on dimensionality; (4) nonlinear feature selection.*

## 4 Polychotomy Problem

The above discussion has been concerned with the learning algorithms for binary classifiers (dichotomy). Generalization to the polychotomy problem can be done in different ways. Two approaches have attracted the main attention of researchers:

1. Training of binary classifiers on the basis of the principle *one class versus the rest.* These classifiers are trained on the entire learning set, but with modified output values, i.e. for a given set of classes $A_1, A_2, \ldots, A_k$ (k is the number of classes) one constructs two classes $-A_i$ and $\bigcup_{j \neq i} A_j$. Since $i = 1, \ldots, k$, we define in such a way $k$ binary classifiers. Using these classifiers, one determines probabilities $p(A_i) i = 1, \ldots, k$, for a test vector to belong to the class $A_i$ (e.g. by applying the sigmoid function to the normalized distance from the decision boundary). We say that a given vector belongs to the class for which $p(A_i)$ is maximal.

2. Training of binary classifiers on the pairwise principle, i.e. for each pair of classes one estimates the probability for a vector to belong to one of these two classes. A convenient combination of these probabilities provides the final decision.

In the following we shall elaborate a specific algorithm belonging to the second approach. Consider a multiclass problem with K classes and n learning samples. Since we have basic two-class classifiers, we shall approach the

problem in the following way: for a given set of classes $A_1, A_2, \ldots, A_k$ and the given test vector we have pairs of probabilities $r_{ij} = P(A_i \mid A_i \quad or \quad A_j)$ i.e. probabilities that the test vector belongs to the class $A_i$ given that only two classes $A_i$ and $A_j$ exist. Based on these probabilities one has to determine the unconditional probabilities $p_i = P(A_i)$; maximal pi indicates the class choice.

Since $P(A_i \mid A_i \quad \text{or} \quad A_j) = p_i/(p_i + p_j)$ and $\sum p_i = 1$ we are looking for $K - 1$ free parameters satisfying $K(K-1)/2$ conditions; this problem does not have solution, in general. We shall be looking, therefore, for given $r_{ij}$, for the best approximation $\hat{\mu} = \hat{p}_i/(\hat{p}_i + \hat{p}_j)$ starting from the criterion of minimizing the average scaled *Kullback-Leibler distance (relative entropy)*:

$$l(p_1, p_2, \ldots, p_k) = \sum_{i<j} n_{ij}[r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 + r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}}]$$

where $n_{ij}$ is the number of sample from classes $A_i$ and $A_j$. Coefficients $n_{ij}$ have to increase the estimation efficiency, especially in the case of disproportions in class volumes. This function has to be minimized with respect to the probabilities $p_1, p_2, \ldots, p_k$. In such a way, one obtains:

$$\sum_{i \neq j} n_{ij}\mu_{ij} = \sum_{i \neq j} n_{ij} r_{ij}, \quad i = 1, \ldots, K$$

under the condition that $\sum p_i = 1$. We shall, therefore construct the following iterative algorithm for obtaining the probabilities $\hat{p}_i$:

1. Start from the estimates of $\hat{p}_i$ and the corresponding $\hat{\mu}_{ij}$

2. Repeat until convergence the following procedure:

$$\hat{p}_i \leftarrow \hat{p}_i \frac{\sum\limits_{i \neq j} n_{ij}\mu_{ij}}{\sum\limits_{i \neq j} n_{ij} r_{ij}}$$

   In each step one again normalizes $\hat{p}_i$ and calculates $\hat{\mu}_{ij}$

3. $$\hat{\boldsymbol{p}} \leftarrow \hat{\boldsymbol{p}} / \sum \hat{p}_i$$

   Having the set of probabilities $p_1, p_2, \ldots, p_K$, we shall say that a test vector belongs to the class $A_i$ if $p_i = \max p_1, p_2, \ldots, p_k$.

## 5 Experimental Results

### 5.1 Speech recognition

In this experiment SVM is applied to the recognition of spoken vowels. Utilized data belong to the public domain and are used for testing nonlinear classifiers, in general. Learning data consist of 528 samples obtained from 8 speakers, while the test set contains 462 samples obtained from 7 speakers (different from those generating the learning set). The speech signal is filtered through a low pass filter with the cutoff frequency of 4.7 kHz, and then discretized, the sampling frequency being 10 kHz. Then, in order to generate features, a time window function of 50 ms is applied. The features are obtained by using the LPC (Linear Predictive Coding) method. It has been assumed that the $10^{th}$ order LPC filter provides a good approximation; therefore, the inputs are 10- dimensional, consisting of 10 reflexion coefficients.

SVM machines are applied to the classification of 11 vowels (11 classes); the obtained results are presented in Tables 1, 2 and 3. Both radial basis and polynomial functions have been applied. The polychotomy method ≫one versus the rest≪ has been applied.

Table 1. Classification of vowels using radial basis functions (variable $\sigma^2$ and fixed $C = 1$).

| $\sigma^2$ (radial basis functions) | Test error (for fixed $C = 1$) |
|---|---|
| 0.1 | 44.16 % |
| 0.3 | 39.18 % |
| 0.5 | 40.26 % |
| 0.7 | 37.88 % |
| 0.9 | 37.23 % |
| 1 | 36.36 % |
| 3 | 36.36 % |
| 8 | 39.18 % |

Table 2. Classification of vowels using radial basis functions (variable regularization parameter C and fixed $\sigma^2 = 1$)

| $C$ (regularization parameter) | Test error (for fixed $\sigma^2 = 1$) |
|---|---|
| 0.01 | 35.71 % |
| 0.05 | 35.71 % |
| 0.1 | 35.28 % |
| 1 | 36.36 % |
| 2 | 36.66 % |
| 10 | 37.02 % |
| 100 | 36.80 % |
| 10000 | 36.80 % |

It is interesting to observe the influence of the regularization parameter C and the parameter 2 of the radial basis functions. Remark for comparison that the neural networks (multilayer perceptrons) provide the error of 44 %.

Table 3. Classification of vowels using polynomial functions.

| Polynomial degree | Test error (for fixed $C = 1$) |
|---|---|
| 2 | 46.32 % |
| 3 | 50.00 % |
| 4 | 59.96 % |

### 5.2 Automatic diagnosis using gene expressions

According to the theory of gene expression, irregularities in the genetic material can indicate different kinds of malign diseases. In practice, the aim is to design a system for gene recognition which would classify, in a short time and accurately, some diseases that are not clearly distinguishable and require different treatments.

The system is based on the measurements of morphologic and protein gene expression. Data about gene expression are obtained from hybrid micronized DNA mollecules. This technology enables measuring expression levels of thousands of genes in only one experiment. We shall take as input values expressions of a large number of genes represented in the form of fraction, in which the denominator contains the the gene expression in nominal conditions, and the numerator gene expression in different, variable conditions of interest. Results of experiments with n genes represent a series of n expression level ratios. Data obtained from a series of m such experiments can be represented as a matrix of n rows containing m elements, each element being related to one given gene.

In the experiments, we have used original data obtained in some hospitals in U.S.A. which have a great experience in the diagnosis of malign diseases in children. In fact, we have on our disposal data obtained from 63 patients with 4 diseases which cannot be distinguished easily (23 with Ewing sarcoma, 20 with rhabdomyo sarcoma, 12 with neuroblasthoma and 8 with Burkitt lymphoma). Expression levels are obtained from 2308 genes in each patient. The input matrix (63×2308) obtained in such a way is used throughout all our experimenting. Since we do not have an isolated test set, evaluation of the results is done by cross-validation based on the principle ≫leave-one-out≪. Namely, we make a learning set by using 62 vectors, and we use the remaining one for test. We can make in such a way 63 different data sets, and 63 different classifiers. The total error is obtained by summing up errors of all 63 classifiers.

Input space dimensionality reduction based on the principal component

analysis (PCA) is done for the sake of comparison with the SVM method which does not require dimensionality reduction. Both described polychotomy methods have been applied (pairwise and one versus the rest).

The obtained results are presented in Table 4. As it can be seen, SVM perform well in the case of a very large number of inputs; computation time is negligible compared, for example, to the backpropagation methodology. As it can be seen, the linear model provides the best performance, i.e. the best generalization. This means that the increase of dimensionality of the feature space (paragraph 3.3) cannot make the upper bound for VC-dimension smaller.

Table 4. Classification of genes.

| Type of classifier | Number of input features | Inner product kernel | | | |
|---|---|---|---|---|---|
| | | Linear | 2 degree | 4 degree | Radial basis functions ($\sigma^2 = 1$) |
| SVM – | 10 | 23.80 | 33.33 | 57.14 | 63.49 |
| pairwise | 63 | 6.34 | 26.98 | 42.86 | 60.31 |
| coupling | 2308 | 7.94 | 23.81 | 42.86 | 63.49 |
| SVM – | 10 | 15.07 | 23.81 | 53.96 | 63.49 |
| on versus | 63 | 7.94 | 22.22 | 39.68 | 63.49 |
| all others | 2308 | 7.94 | 22.22 | 39.68 | 63.49 |

## 6 Conclusion

In this paper the problem of learning from data is treated from the point of view of the Vapnik-Chervonenkis approach and the methodology of support vector machines. Particular attention is paid to the multiclass classification problem, where a specific algorithm is described. The presented methodology is illustrated by using two examples: one is related to speech recognition, and the other to automatic dignosis by using gene expressions.

Implications of the presented methodology to the area of digital signal processing are tremendous. It should only be mentioned that regression by SVM is a field not yet well understood, offering numerous possibilities for robust and computationally efficient modeling from data. Extensions to dynamic signal models could bring new efficient tools to system identification.

## References

[1] V. Vapnik, *The Nature of Statistical Learning Theory.* N.Y.: Springer Verlag, 1995.

[2] T. Hastie and R. Tibshirani, *The Elements of Statistical Learning.* N.Y.: Springer Verlag, 2001.

[3] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition.* Moscow: Nauka, 1974 (in Russian).

[4] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems 10* (S. A. S. M. I. Jordan, M. J. Kearns, ed.), MTT Press, 1998.