Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_299-3

Support Vector Machine

Mathias M. Adankon*[a] and Mohamed Cheriet[b]
[a]ML-Consulting, Laval, QC, Canada
[b]University of Montreal, Montreal, QC, Canada

Synonyms

Margin classifier; Maximum margin classifier; Optimal hyperplane SVM;

Definition

Support vector machines (SVMs) are particular linear classifiers which are based on the margin maximization principle. They perform structural risk minimization, which improves the complexity of the classifier with the aim of achieving excellent generalization performance. The SVM accomplishes the classification task by constructing, in a higher dimensional space, the hyperplane that optimally separates the data into two categories.

Introduction

Considering a two-category classification problem, a linear classifier separates the space, with a hyperplane, into two regions, each of which is also called a class. Before the creation of SVMs, the popular algorithm for determining the parameters of a linear classifier was a single-neuron perceptron. The perceptron algorithm uses an updating rule to generate a separating surface for a two-class problem. The procedure is guaranteed to converge when the training data are linearly separable; however, there exist an infinite number of hyperplanes that correctly classify these data (see Fig. 1).

The idea behind the SVM is to select the hyperplane that provides the best generalization capacity. Then, the SVM algorithm attempts to find the maximum margin between the two data categories and then determines the hyperplane that is in middle of the maximum margin. Thus, the points nearest the decision boundary are located at the same distance from the optimal hyperplane. In machine learning theory, it is demonstrated that the margin maximization principle provides the SVM with a good generalization capacity, because it minimizes the structural risk related to the complexity of the SVM [1].

SVM Formulation

Let us consider a dataset $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ with $x_i \in R^d$ and $y_i \in \{-1, 1\}$. SVM training attempts to find the parameters w and b of the linear decision function $f(x) = w.x + b$ defining the optimal hyperplane. The points nearest the decision boundary define the margin.

*E-mail: mathias.adankon@ml-consulting.ca

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_299-3

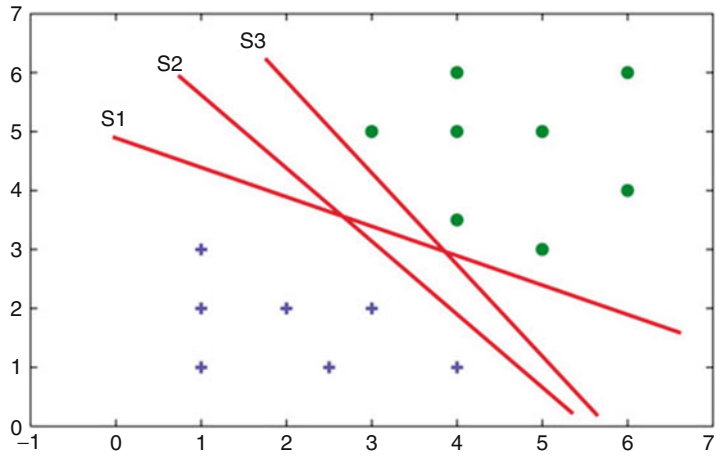

Fig. 1 Linear classifier: in this case, there exists an infinite number of solutions. Which is the best?

Considering two points x_1 and x_2 on opposite sides of the margin with $f(x_1) = 1$ and $f(x_2) = -1$, the margin equals $[f(x_1) - f(x_2)/||w|| = 2/||w||]$. Thus, maximizing the margin is equivalent to minimizing $||w||/2$ or $||w||^2/2$. Then, to find the optimal hyperplane, the SVM solves the following optimization problem :

$$\begin{aligned} &\min_{w,b} \tfrac{1}{2} w'w \\ &\text{s.t.} \quad y_i(w' .x_i + b) \geq 1 \quad \forall i = 1, \ldots, \ell \end{aligned} \tag{1}$$

The transformation of this optimization problem into its corresponding dual problem gives the following quadratic problem:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{\ell} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j (x_i . x_j) \\ &\text{s.t.} \sum_{i=1}^{\ell} y_i \alpha_i = 0; \quad \alpha \geq 0 \quad \forall i = 1, \ldots, \ell \end{aligned} \tag{2}$$

The solution of the previous problem gives the parameter $w = \sum_{i=1}^{\ell} y_i \alpha_i x_i$ of the optimal hyperplane. Thus, the decision function becomes $f(x) = \sum_{i=1}^{\ell} \alpha_i y_i (x_i . x) + b$ in dual space. Note that the value of the bias b does not appear in the dual problem. Using the constraints of the primal problem, the bias is given by $b = -1/2[\max_{y=-1}(w.x_i) + \min_{y=1}(w.x_i)]$. It is demonstrated with the Karush-Kuhn-Tucker conditions that only the examples x_i that satisfy $y_i(w.x_i + b) = 1$ are the corresponding α_i nonzero. These examples are called *support vectors* (see Fig. 2).

SVM in Practice

In real-world problems, the data are not linearly separable, and so a more sophisticated SVM is used to solve them. First, the slack variable is introduced in order to relax the margin (this is called a soft margin optimization). Second, the kernel trick is used to produce nonlinear boundaries [2].

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_299-3

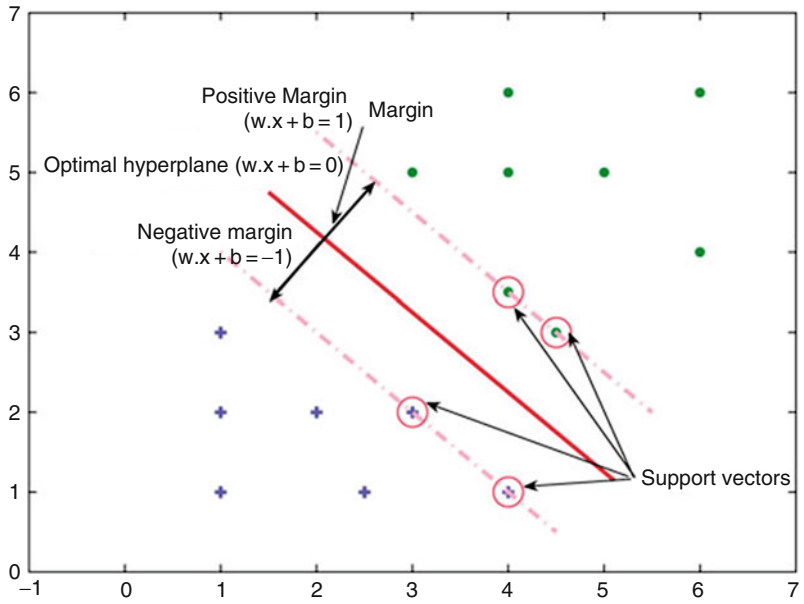

Fig. 2 SVM principle: illustration of the unique and optimal hyperplane in a two-dimensional input space based on margin maximization

The idea behind kernels is to map training data nonlinearly into a higher-dimensional feature space via a mapping function Φ and to construct a separating hyperplane which maximizes the margin (see Fig. 3). The construction of the linear decision surface in this feature space only requires the evaluation of dot products $\varphi(x_i).\varphi(x_j) = k(x_i, x_i)$, where the application $k : \mathcal{R}^d . \times \mathcal{R}^d \rightarrow \mathcal{R}$ is called the kernel function [3, 4].

The decision function given by an SVM is

$$y(x) = \text{sign}[w'\varphi(x) + b], \tag{3}$$

where w and b are found by resolving the following optimization problem that expresses the maximization of the margin $1/||w||$ and the minimization of training error:

$$\min_{w,b,\xi} \frac{1}{2} w'w + C \sum_{i=1}^{\ell} \xi_i \quad \text{(L1-SVM)} \quad \text{or} \quad \min_{w,b,\xi} \frac{1}{2} w'w + C \sum_{i=1}^{\ell} \xi_i^2 \quad \text{(L2-SVM)} \tag{4}$$

$$\text{subject to} : y_i[w'\varphi(x_i) + b] \geq 1 - \xi_i \quad \forall i = 1, \ldots, \ell \tag{5}$$

$$\xi_i \geq 0 \quad \forall i = 1, \ldots, \ell. \tag{6}$$

By applying the Lagrangian differentiation theorem to the corresponding dual problem, the following decision function is obtained:

$$y(x) = \text{sign}\left[\sum_{i=1}^{\ell} \alpha_i y_i k(x_i, x) + b\right], \tag{7}$$

with α as the solution of the dual problem.

DOI 10.1007/978-3-642-27733-7_299-3

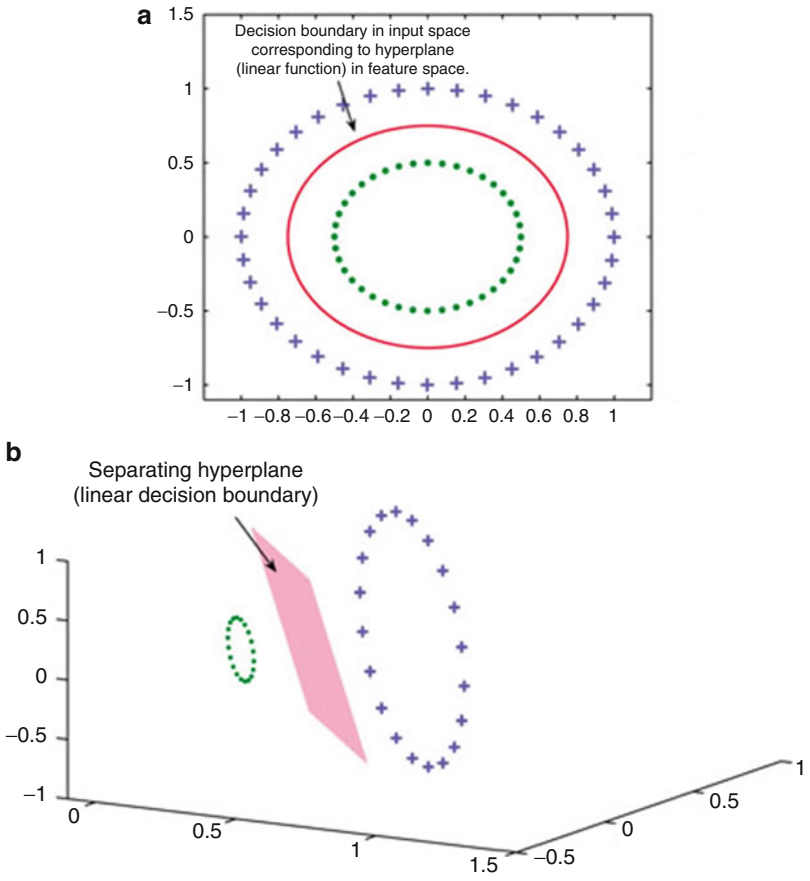

Fig. 3 Illustration of the kernel trick: the data are mapped into a higher-dimensional feature space, where a separating hyperplane is constructed using the margin maximization principle. The hyperplane is computed using the kernel function without the explicit expression of the mapping function: (**a**) nonlinearly separable data in the input space and (**b**) data in the higher-dimensional feature space

The dual problem for the L1-SVM is the following quadratic optimization problem:

$$\text{maximize}: \; W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{8}$$

$$\text{subject to}: \; \sum_{i=1}^{\ell} \alpha_i y_i = 0 \;\; \text{and} \;\; 0 \leq \alpha_i \leq C, \; i = 1, \ldots, \ell. \tag{9}$$

Using the L2-SVM, the dual problem becomes

$$\text{maximize}: \; W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \left(k(x_i, x_j) + \frac{1}{C} \delta_{ij} \right) \tag{10}$$

$$\text{subject to}: \; \sum_{i=1}^{\ell} \alpha_i y_i = 0 \;\; \text{and} \;\; 0 \leq \alpha_i, \; i = 1, \ldots, \ell. \tag{11}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

DOI 10.1007/978-3-642-27733-7_299-3

Table 1 Common kernel used with SVM

Gaussian (RBF)	$k(x, y) = \exp(-\|\|x - y\|\|/\sigma^2)$
Polynomial	$k(x, y) = (ax.y + b)^n$
Laplacian	$k(x, y) = \exp(-a\|\|x - y\|\| + b)$
Multi-quadratic	$k(x, y) = (a\|\|x - y\|\| + b)^{1/2}$
Inverse multi-quadratic	$k(x, y) = (a\|\|x - y\|\| + b)^{-1/2}$
KMOD	$k(x, y) = a\left[\exp\left(\frac{\gamma^2}{\|\|x-y\|\|^2+\sigma^2}\right) - 1\right]$

In practice, the L1-SVM is used most of the time, and its popular implementation developed by Joachims [5] is very fast and scales to large datasets. This implementation, called *SV Mlight*, is available at svmlight.joachims.org.

SVM Model Selection

To achieve good SVM performance, optimum values for the kernel parameters and for the hyperparameter C must be chosen. The latter is a regularization parameter controlling the trade-off between the training error minimization and the margin maximization. The kernel parameters define the kernel function used to map data into a higher-dimensional feature space (see Table 1). Like kernel functions, there are the Gaussian kernel $k(x_i, x_j) = \exp(-||x_i - x_j||^2/\sigma^2)$ with parameter σ and the polynomial kernel $k(x_i, x_j) = (ax_i'x_j + b)^d$ with parameters a, b, and d. The task of selecting the hyperparameters that yield the best performance of the machine is called model selection [6–9].

As an illustration, Fig. 4a shows the variation of the error rate on a validation set versus the variation of the Gaussian kernel with a fixed value of C, and Fig. 4b shows the variation of the error rate on the validation set versus the variation of the hyperparameter C with a fixed value of the RBF kernel parameter. In each case, the binary problem described by the "Thyroid" data taken from the UCI benchmark is resolved. Clearly, the best performance is achieved with an optimum choice of the kernel parameter and of C.

With the SVM, as with other kernel classifiers, the choice of kernel corresponds to choosing a function space for learning. The kernel determines the functional form of all possible solutions. Thus, the choice of kernel is very important in the construction of a good machine. So, in order to obtain a good performance from the SVM classifier, one first needs to design or choose a type of kernel and then optimize the SVM's hyperparameters to improve the classifier's generalization capacity. Figure 4c illustrates the influence of the kernel choice, where the RBF and the polynomial kernels are compared to the datasets taken from the challenge website on model selection and prediction organized by Isabelle Guyon.

Resolution of Multiclass Problems with the SVM

The SVM is formulated for the binary classification problem. However, there are some techniques used to combine several binary SVMs in order to build a system for the multiclass problem (e.g., a 10-class digit recognition problem). Two popular methods are presented here:

DOI 10.1007/978-3-642-27733-7_299-3

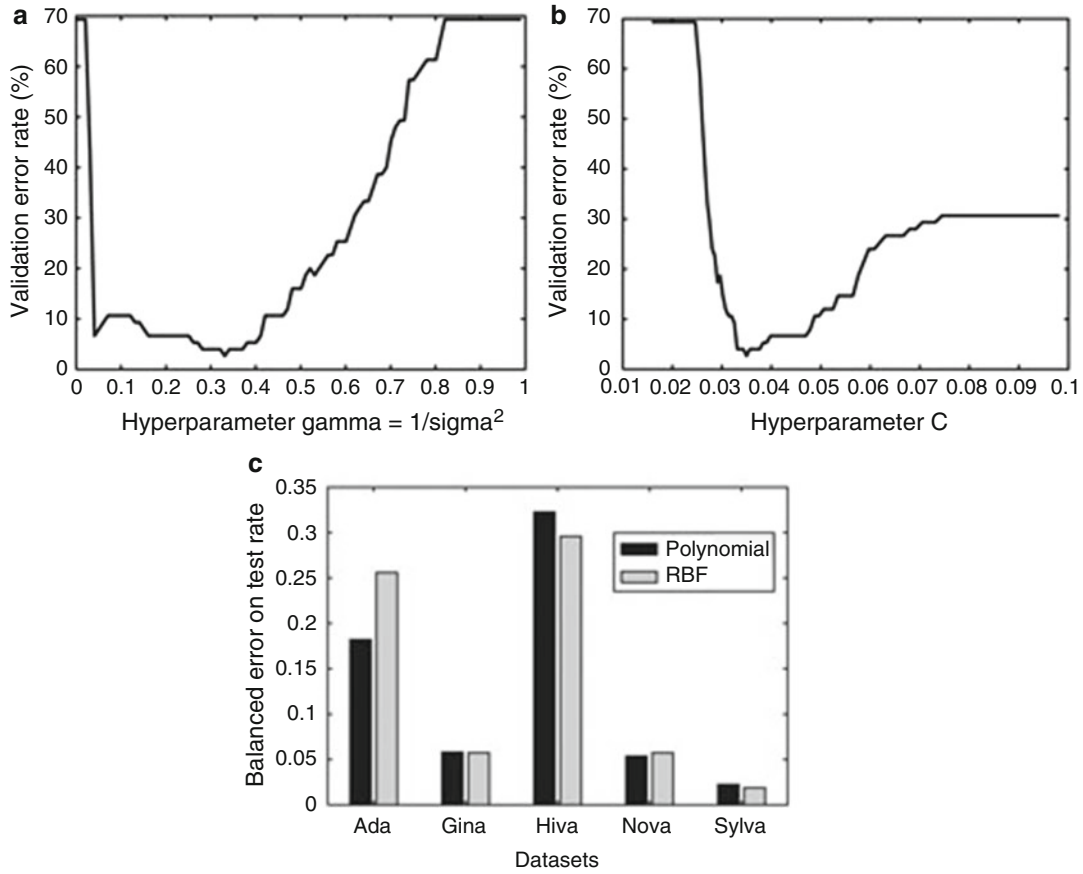

Fig. 4 (**a**) and (**b**) show the impact of SVM hyperparameters on classifier generalization, while (**c**) illustrates the influence of the choice of kernel function

One Versus the Rest: The idea is to construct as many SVMs as there are classes, where each SVM is trained to separate one class from the rest. Thus, for a c-class problem, c SVMs are built and combined to perform multiclass classification according to the maximal output. The ith SVM is trained with all the examples in the ith class with positive labels, and all the other examples with negative examples. This is also known as the *one-against-all* method.

Pairwise (or One-Against-One): The idea here is to construct $c(c-1)2$ SVMs for a c-class problem, each SVM being trained for every possible pair of classes. A common way to make a decision with the pairwise method is by voting. A rule for discriminating between every pair of classes is constructed, and the class with the largest vote is selected.

Least Squares SVM

The least squares SVM (LS-SVM) is a variant of the standard SVM and constitutes the response to the following question: *How much can the SVM formulation be simplified without losing any of its advantages?* Suykens and Vandewalle [10] proposed the LS-SVM where the training algorithm solves a convex problem like the SVM. In addition, the training algorithm of the LS-SVM is very simplified, since a system of linear equations is resolved instead of a quadratic problem in the SVM case. The formulation of the LS-SVM is

DOI 10.1007/978-3-642-27733-7_299-3

$$\min_{w,b,\xi} \frac{1}{2} w'w + \frac{1}{2} C \sum_{i=1}^{\ell} \xi_i^2 \tag{12}$$

$$\text{s.t} \quad \xi_i = y_i - [w'\varphi(x_i) + b] \quad \forall_i = 1, \ldots, \ell \tag{13}$$

And the corresponding dual problem gives the solution in matrix form as follows:

$$\begin{pmatrix} K + C^{-1} I & 1' \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ b \end{pmatrix} = \begin{pmatrix} Y \\ 0 \end{pmatrix} \tag{14}$$

where:

$K_{ij} = k(x_i, x_j)$
$Y = (y_1, \ldots, y_\ell)'$
$\alpha = (\alpha_1, \ldots, \alpha_\ell)'$
$1 = (1, \ldots, 1)$

Unlike SVM, LS-SVM is not sparse enough. However, the formulation of the LS-SVM allows to perform easily the model selection with leave-one-out procedure which involves a huge computational time for SVM. In fact, it is possible to compute the exact cross validation error without repeating the training step [11, 12].

Other SVM Variants

The transductive SVM (TSVM) is an interesting version of the SVM, which uses transductive inference. In this case, the TSVM attempts to find the hyperplane and the labels of the test data that maximize the margin with minimum error. Thus, the label of the test data is obtained in one step. Vapnik [1] proposed this formulation to reinforce the classifier on the test set by adding the minimization of the error on the test set during the training process. This formulation has been used elsewhere recently for training semi-supervised SVMs (S3VM).

In [13], the Bayesian approach is used with one and two levels of inference to model the semi-supervised learning problem and its application to SVM, and LS-SVM is proposed. This framework established the Bayesian interpretation of the S3VM introduced first time as a TSVM and gave the root for developing other semi-supervised training algorithms.

Concerning regression problem where the goal is to find an approximation for unknown function with output $y_i \in R$, we use the support vector regression (SVR) [1]. In this model, the generalization term represented by the margin maximization is conserved as in the original SVM, and the loss function penalizes linearly only the points outside the margin.

Applications

The SVM is a powerful classifier which has been used successfully in many pattern recognition problems, and it has also been shown to perform well in biometrics recognition applications. For example, in [14], an iris recognition system for human identification has been proposed, in which the extracted iris features are fed into an SVM for classification. The experimental results show that

DOI 10.1007/978-3-642-27733-7_299-3

the performance of the SVM as a classifier is far better than the performance of a classifier based on the artificial neural network. In another example, Yao et al. [15], in a fingerprint classification application, used recursive neural networks to extract a set of distributed features of the fingerprint which can be integrated into the SVM. Many other SVM applications, like handwriting recognition [8, 16], can be found at www.clopinet.com/isabelle/Projects/SVM/applist.html.

Related Entries

- Biometric Applications, Overview
- Biometrics, Overview
- Support Vector Machine

References

1. V. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998)
2. B. E. Boser, I. Guyon, V. Vapnik, A Training Algorithm for Optimal Margin Classifiers, in *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, ed. by D. Haussler, (ACM Press, Pittsburgh, PA, USA, 1992), pp. 144–152
3. B. Scholkopf, A. Smola, *Learning with Kernels* (MIT, Cambridge, 2002)
4. N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines* (Cambridge University Press, Cambridge/New York, 2000)
5. T. Joachims, Making large-scale support vector machine learning practical, in *Advances in Kernel Methods: Support Vector Machines*, ed. by B. Scholkopf, C.J.C. Burges, A.J. Smola (MIT, Cambridge, 1998)
6. O. Chapelle, V. Vapnik, Model selection for support vector machines, in *Advances in Neural Information Processing Systems*, Denver, 1999
7. N.E. Ayat, M. Cheriet, C. Suen, Automatic model selection for the optimization of the SVM kernels. Pattern Recognit. **38**(9), 1733–1745 (2005)
8. M.M. Adankon, M. Cheriet, Optimizing resources in model selection for support vector machines. Pattern Recognit. **40**(3), 953–963 (2007)
9. M.M. Adankon, M. Cheriet, New formulation of SVM for model selection, in *International Joint Conference in Neural Networks 2006*, Vancouver (IEEE, 2006), pp. 3566–3573
10. J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines* (World Scientific, Singapore, 2002)
11. G.C. Cawley, N.L.C. Talbot, Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. Neural Netw. **17**, 1467–1475 (2004)
12. M.M. Adankon, M. Cheriet, Model selection for the LS-SVM. Application to handwriting recognition. Pattern Recognit. **42**(11), 3264–3270 (2009)
13. M.M. Adankon, M. Cheriet, A. Biem, Semisupervised learning using Bayesian interpretation: application to LS-SVM. IEEE Trans. Neural Netw. **22**(4), 513–524 (2011)
14. K. Roy, P. Bhattacharya, Iris recognition using support vector machine, in *APR International Conference on Biometric Authentication (ICBA)*, Hong Kong, Jan 2006. Springer Lecture Note Series in Computer Science (LNCS), vol. 3882, 2006, pp. 486–492

DOI 10.1007/978-3-642-27733-7_299-3

15. Y. Yao, G. Luca Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines. Pattern Recognit. Comput. Sci. **36**(2), 397–406 (2003)
16. N. Matic, I. Guyon, J. Denker, V. Vapnik, Writer adaptation for on-line handwritten character recognition, in *Second International Conference on Pattern Recognition and Document Analysis*, Tsukuba (IEEE, 1993), pp. 187–191