SUPPORT VECTOR MACHINES FOR PATTERN RECOGNITION

A. Fazekas

SSIP'01

Szeged, HUNGARY

July 12-21 2001

This work was supported by the European Union Research Training Network "Multi-modal Human-Computer Interaction (HPRN-CT-2000-00111)".

Department of Informatics, Aristotle University of Thessaloniki, Greece

Roadmap

- Introduction
- Learning from examples
- Support vector classification
- Support vector machine
- SVM in face detection
- SVM with Walsh kernel
- Experimental results

Model of learning from examples

- A generator of random vectors x, drawn independently from a fixed, but unknown distribution $P(x)$.
- A supervisor that returns an output vector y for every input vector x, according to a conditional distribution function $P(y|x)$, also fixed but unknown.
- A learning machine capable of implementing a set of functions $f(x, \alpha), \alpha \in \Lambda$.

The problem of learning is that of choosing from the given set of functions, the one which predicts the supervisor's response in the best possible way. The selection is based on a training set of l random independent identically distributed observations drawn according to $P(x, y) = P(x)P(y|x)$.

Problem of risk minimization

In order to choose the best available approximation to the supervisor's response, one measures the loss $L(y, f(x, \alpha))$ between the response y of the supervisor to a given input x and the response $f(x, \alpha)$ provided by the learning machine. Consider the expected value of the loss, given by the risk functional

$$R(\alpha) = \int L(y, f(x, \alpha)) dP(x, y).$$

The goal is to find the function $f(x, \alpha_0)$ which minimizes the risk functional $R(\alpha)$ in the situation where the joint probability distribution $P(x, y)$ is unknown and the only available information is contained in the training set.

The problem of pattern recognition

- Let the supervisor's output y take on only two values $y = \{0, 1\}$.
- Let $f(x, \alpha)$, $\alpha \in \Lambda$ be a set of indicator functions (functions which take on only two values zero and one).
- Consider the following loss-function

$$L(y, f(x, \alpha)) = \begin{cases} 0, & \text{if } y = f(x, \alpha), \\ 1, & \text{if } y \neq f(x, \alpha). \end{cases}$$

The problem is to find the function which minimizes the probability of classification errors when probability measure $P(x, y)$ is unknown, but the data are given.

The importance of the set of functions

- What about allowing *all* functions from $\mathbb{R}^N$ to $\{\pm 1\}$?
- Training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \in \mathbb{R}^N \times \{\pm 1\}$.
- Test patterns $\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_l \in \mathbb{R}^N$, such that the elements of the training set is not elements of the test set.
- Based on the training set alone, there is no means of choosing which one is better, because for any f there exits f^*, where
  - $f^*(\mathbf{x}_i) = f(\mathbf{x})$, for all i,
  - $f^*(\overline{\mathbf{x}}_j) \neq f(\overline{\mathbf{x}}_j)$, for all j.
- There is "no free lunch". The restriction must be placed on the functions that we allow.

Restricting of the class of functions

- Statistical Learning (VC) Theory: take into account the capacity of the class of functions that the learning machine can implement.
- The Bayesian Way: place prior distributions $P(f)$ over the class of functions.

ERM induction principle

- In order to minimize the risk functional, for an unknown probability measure $P(z)$ the following induction principle is usually used.
- The expected risk functional $R(\alpha)$ is replaced by the empirical risk functional constructed on the basis of the training set.

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^{l} Q(z, \alpha)$$

- The principle is to approximate the function $Q(z, \alpha_0)$ which minimizes risk by the function $Q(z, \alpha_l)$ which minimizes empirical risk.
- The ERM principle is quite general.
- The classical methods for solving a specific learning problem are realizations of the ERM principle for the specific loss functions considered above.

Four parts of learning theory

- What are the conditions for consistency of the ERM principle?
- How fast does the sequence of smallest empirical risk values convergence to the smallest actual risk?
- How can one control the rate of convergence (the rate of generalization) of the learning machine?
- How can one construct algorithms that can control the rate of generalization?

Three milestones in learning theory

- The finiteness of the VC-dimension of the set of indicator functions implemented by the learning machine forms the necessary and sufficient condition for consistency of the ERM method independent of probability measure.
- The finiteness of VC-dimension also implies fast convergence.

SRM induction principle

- The ERM principle is intended for dealing with a large sample size.
- Indeed, the ERM principle can be justified by considering the inequalitie

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{B_\epsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{B_\epsilon}}\right)$$

- However, if $\frac{l}{h}$ is small, then even a small $R_{emp}(\alpha_l)$ does not guarantee a small value of risk. In this case the minimization for $R(\alpha)$ requires a new principle.
- This principle is based on the simultaneous minimization of two terms in the inequalitie.
- To minimize risk in this case it is necessary to find a method which, along with minimizing the value of empirical risk, controls the VC-dimension of the learning machine.
- For any distribution function the SRM method provides convergence to the best possible solution with probability one.

The optimal separating hyperplanes I.

- Suppose the training data $(x_1, y_1), \ldots, (x_l, y_l)$, $x \in \mathbb{R}^n$, $y \in \{+1, -1\}$ can be separated by a hyperplane

$$(w \cdot x) - b = 0.$$

- We say that this set of vectors is separated by the optimal hyperplane if it is separated without error and the distance between the closest vector and the hyperplane is maximal.
- To describe the separating hyperplane let us use the following form:

$$\begin{aligned} (\mathbf{w} \cdot \mathbf{x}_i) - b \geq 1, \quad & \text{if } y_i = +1, \\ (\mathbf{w} \cdot \mathbf{x}_i) - b \leq -1, \quad & \text{if } y_i = -1. \end{aligned}$$

The optimal separating hyperplanes II.

- In the following we use a compact notation for these inequalities:

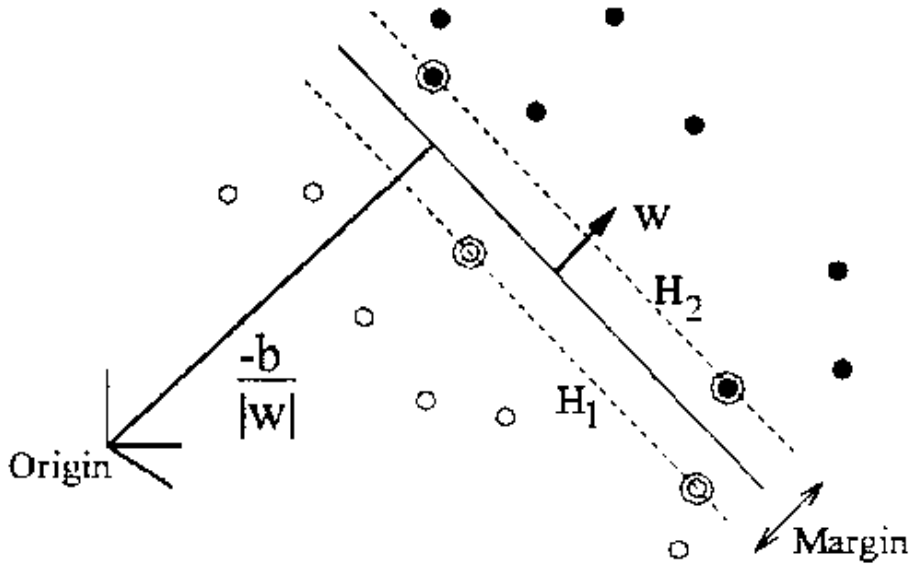
$$y_i \left((\mathbf{w} \cdot \mathbf{x}_i) - b \right) \geq 1, \quad i = 1, \ldots, l.$$

- It is easy to check that the optimal hyperplane is the one that satisfies the condition and minimizes functional

$$\Phi(w) = \frac{1}{2} \cdot \|w\|^2.$$

- The solution to this optimization problem is given by the saddle point of a Lagrange functional.

The optimal separating hyperplanes III.



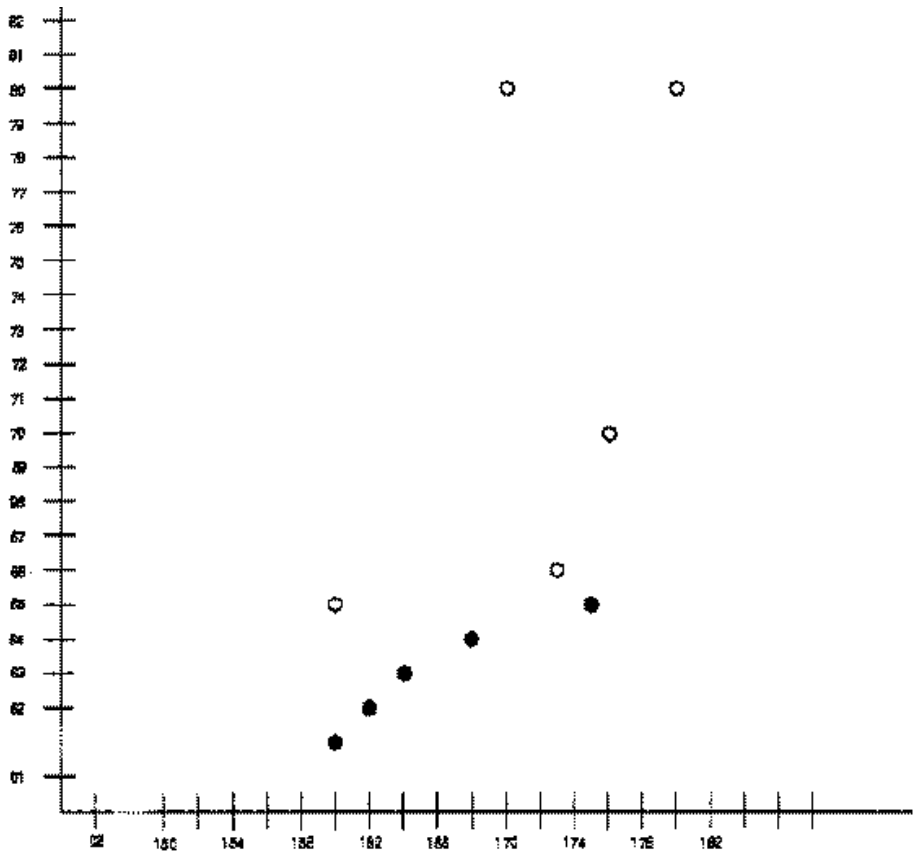
Basic example

- The problem we look at initially is the problem of finding binary classifiers.
- Let us consider the given weight and height of a person. We want to find a way of determining their gender.
- If we are given a set of examples with height, weight and gender, we can come up with a hypothesis which will enable us to determine a person's gender from their weight and height.
- The weights and heights in a two-dimensional coordinate system are points.
- Let us find the separating hyperplane which divides the points into two regions, one female, one male.

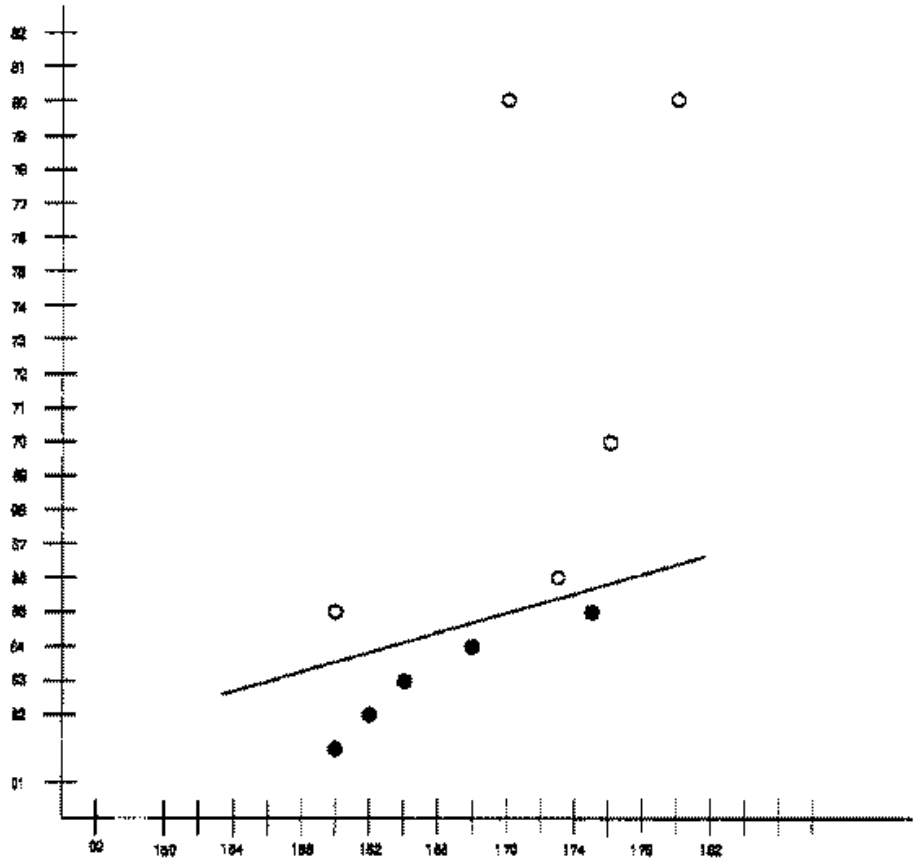
Data of the basic example

No.	Height	Weight	Gender
1	180	80	m
2	173	66	m
3	170	80	m
4	176	70	m
5	160	65	m
6	160	61	f
7	162	62	f
8	168	64	f
9	164	63	f
10	175	65	f

Representation of the data I.



Representation of the data II.



Support vector machine I.

- Map the input vectors into a very high-dimensional feature space through some nonlinear mapping choosen a priori.
- In this space construct an optimal separating hyperplane.
- To generalize well, we control (decrease) the VC dimension by constructing an optimal separating hyperplane (that maximizes the margin).
- To increase the margin we use very high dimensional spaces.

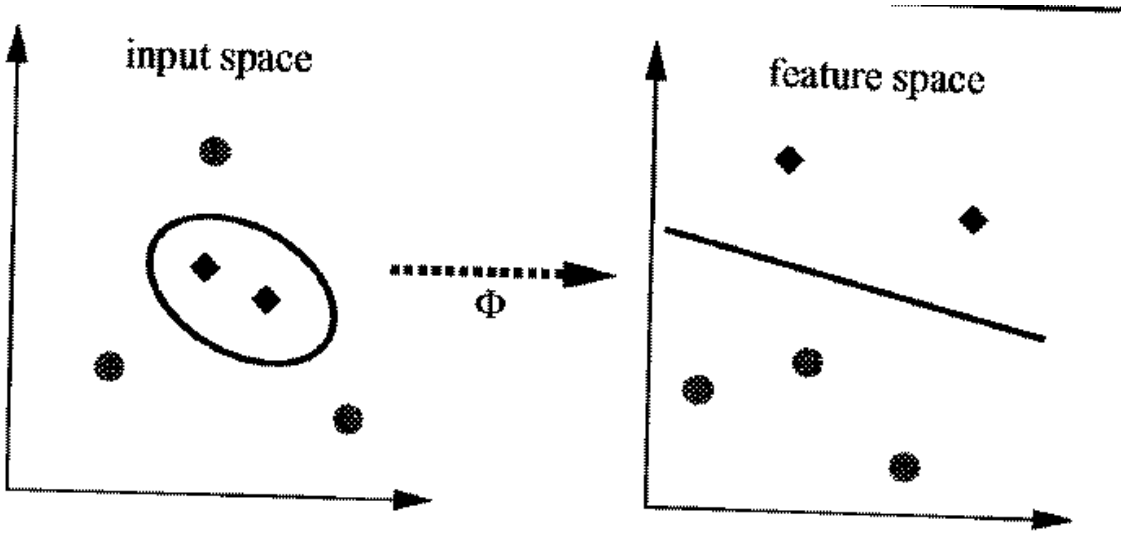
Support vector machine II.

- The training algorithm would only depend on the data through dot products in the feature space, i.e. on functions of the form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Now if there were a "kernel function" K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, we would only need to use K in the training algorithm, and would never need to explicitly even know what Φ is.
- Mercer's Condition
- Polynomial kernel $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$
- Gaussian radial kernel $(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$
- Two-layer sigmoidal neural network $(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta)$

Summary of some features of SVM I.

- SVM performs *Structural Risk Minimisation*.
- It creates a classifier with minimised *VC dimension*.
- If the VC dimension is low, the *expected probability of error* is low as well.
- SVM uses a *linear separating hyperplane* to create a classifier. **But** some problems can not be linearly separated in the original input space.
- SVM can non-linearly transform the original input space into a higher dimensional feature space.

Summary of some features of SVM II.



Neural networks as a solution

- Most neural networks are designed to find a separating hyperplane.
- This is not necessarily optimal.
- In fact many neural networks start with a random line and move it, until all training points are on the right side of the line.

Support vector machines as a solution

- Support Vector Machines use geometric properties to exactly calculate the optimal separating hyperplane directly from the training data.
- They also introduce methods to deal with non-linearly separable cases.

Face detection I.

Some tasks from the literature:

- face tracking
- face detection
- face recognition
- face verification

Face detection II.

- Equalization of the gray-level information
- Oval mask
- Scanning
- Extraction of the information to a pyramid.
- SVM

The Walsh system

- Let r be the function defined on $[0,1)$ by

$$r(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}), \\ -1, & x \in [\frac{1}{2}, 1) \end{cases}$$

extended to $\mathbb{R}$ periodically with period 1. The Rademacher system $R = \{r_n, n \in \mathbb{N}\}$ is defined by

$$r_n(x) = r(2^n x), \quad x \in \mathbb{R}, n \in \mathbb{N}.$$

- The Walsh system $W = \{w_n, n \in \mathbb{N}\}$ is product of Rademacher functions in the following way. If $n \in \mathbb{N}$ has binary coefficients $\{n_k, k \in \mathbb{N}\}$ then

$$w_n(x) = \prod_{k=0}^{\infty} r_k^{n_k}(x).$$

The Walsh kernel

- It is well-known if f is an $\mathbb{R}$-valued, integrable function on the interval $[0,1)$ then $f(x) = \sum_{k=0}^{\infty} a_k(f) w_k(x)$, where $a_k = \langle f, w_k \rangle$.
- Let N be a fixed element of the set $\mathbb{N}$, and

$$\mathbf{\Phi}_N(f) = (a_0(f), \ldots, a_{N-1}(f)) .$$

- Then the kernel function is

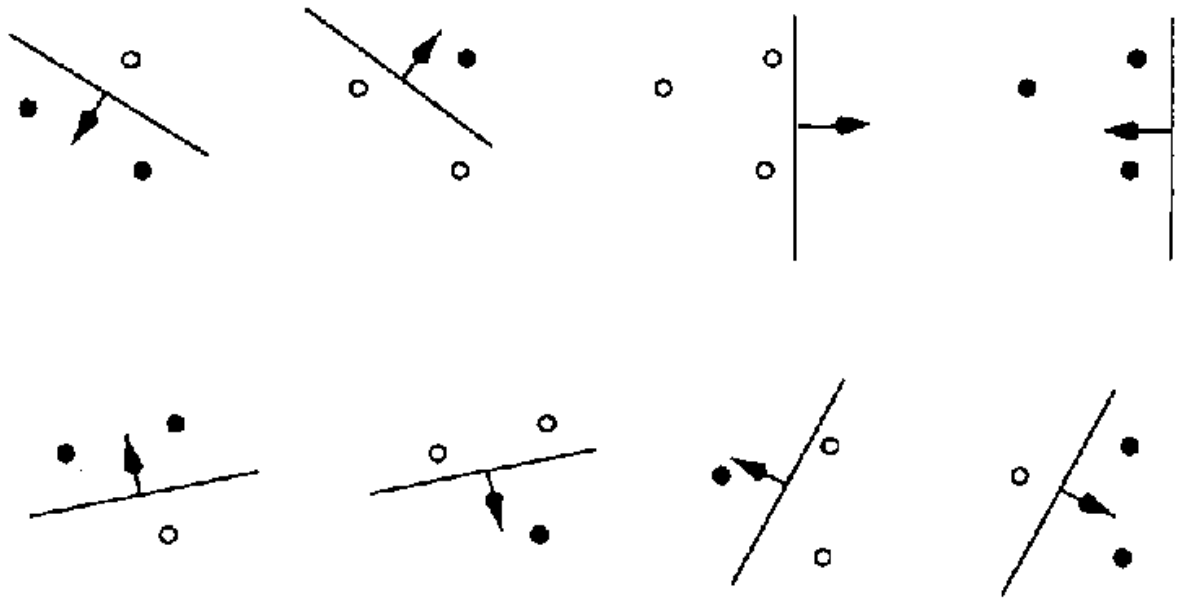
$$K_N(f,g) = \langle \mathbf{\Phi}_N(f), \mathbf{\Phi}_N(g) \rangle .$$

- It is easy to see, that $\mathbf{\Phi}(f)$ is a Walsh-transformation of the one-dimensional function f.
- To determine the value of the kernel function $K_N(f,g)$ is not a difficult task, because some fast Walsh transformation are well-known in the literature.
- We will get the higher dimensional kernel function as a tensor of one-dimensional kernel functions.

About the VC-dimension I.

- The Vapnik-Chervonenkis dimension has a very important role in the statistical learning.
  - It characterizes the learning capacity.
  - One can avoid the overfitting with its control.
  - One can minimize the expected value of the error with its control.
- The VC-dimension of a set of $\{+1, -1\}$-valued functions is equal to the largest number h of points of the domain of the functions that can be separated into two different classes in all the 2^h possible ways using the functions of this set of functions.
- The VC-dimension of the class of all Walsh functions is equal to ∞.
- The relations $N = 2^n$ and $n \geq 2$ are assumed in the following.
- The VC-dimension of the set $W_N = \{w_k \mid k = 0, \ldots, N-1\}$ equals n.

About the VC-dimension II.



Experimental results I.

- For all experiments the `Mathlab` SVM toolbox developed by Steve Gunn was used. For a complete test, several auxiliary routines have been added to the original toolbox.
  - Training set of 46 images (31 (IBERMATICA) face – 15 non-face)
  - IBERMATICA – several sources of degradation are modeled.
  - All images are recorded in 256 grey levels.
  - They are of dimensions 320×240.

Experimental results II.

- The procedure for collecting face patterns is as follows.
  - A rectangle part of dimensions 128×128 pixels has been manually determined that includes the actual face.
  - This area has been subsampled four times. At each subsampling, non-overlapping regions of 2×2 pixels are replaced by their average.
  - The training patterns of dimensions 8×8 are built.
  - The class label $+1$ has been appended to each pattern.
  - Similarly, 15 non-face patterns have been collected from images in the same way, and labeled by -1.

Experimental results III.

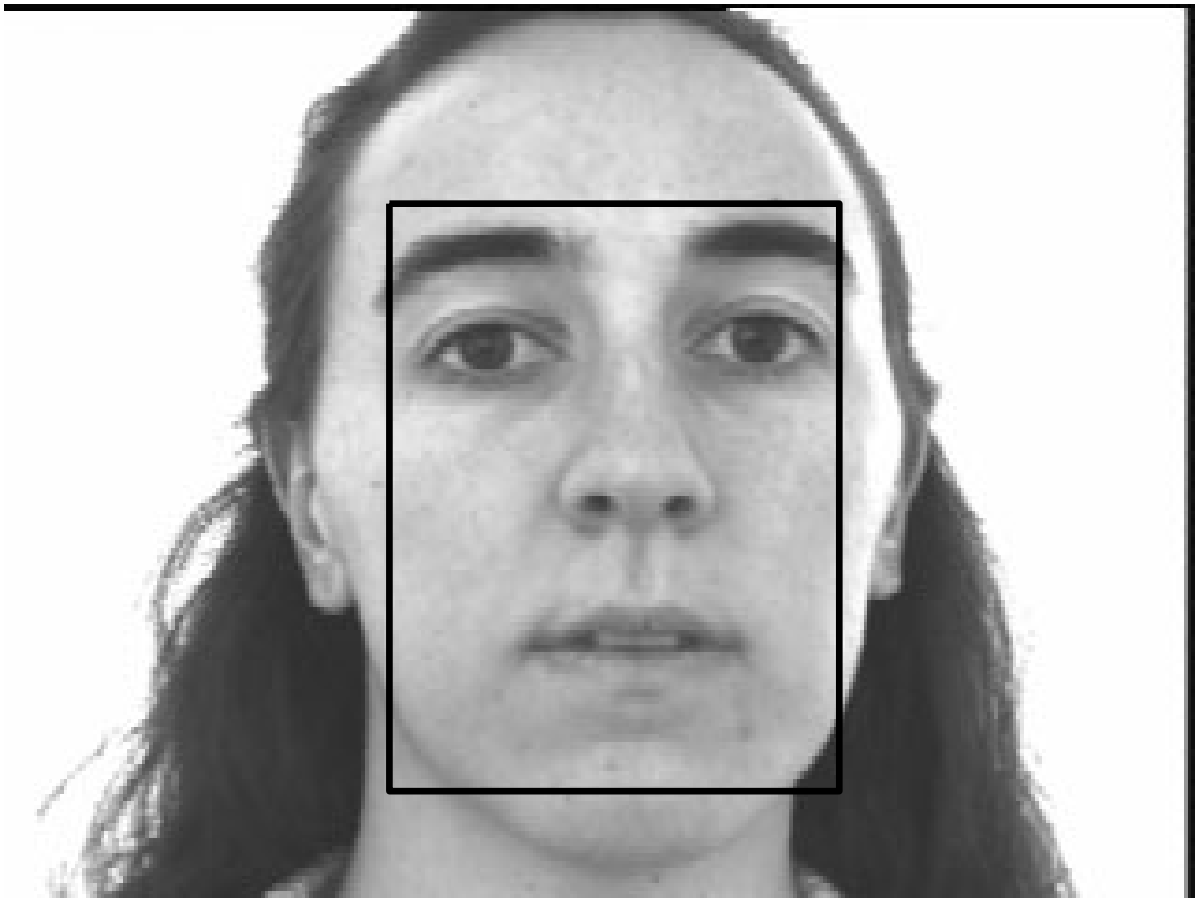
- We have trained the three different SVMs. The trained SVMs have been applied to 414 test examples (249 face and 165 non-face). The test images are classified as non-face ones or face ones. The following table gives the results on the test.

	Linear	Walsh	Polynomial
Time	2.3581	2.3432	2.5327
Errors	9	8	7
Margin	0.66	4.58	2.17
SVs	15	12	8

Experimental results IV.

Experimental results V.

Experimental results VI.



References

- V.N. Vapnik, "An Overview of Statistical Learning Theory", *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, 988–999, 1999.
- M.O. Stitson, J.A.E. Weston, A. Gammerman, V. Vovk, and V. Vapnik, "Theory of Support Vector Machines", *Technical Report CSD-TR-96-17*,1–28,1996.
- C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery* 2, 121–167, 1998.
- B. Schölkopf, "Support Vector Learning", DAGM, 1999.
- H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1., 23–38,1998.