Gentle Guide to Support Vector Machines

Ming-Hsuan Yang

Linear Support Vector Machine

Given a set of points $\mathbf{x}_i \in \Re^n$ with $i = 1, 2, \ldots, N$. Each point $\mathbf{x}_i$ belongs to either of two classes with the label $y_i \in \{-1, 1\}$.

Definition 1 *The set S is linearly separable if there exist $\mathbf{w} \in \Re^n$ and $b \in \Re$ such that*

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N. \tag{9}$$

The pair $(\mathbf{w}, b)$ defines a hyperplane of equation $\mathbf{w} \cdot \mathbf{x}_i + b = 0$ named separating hyperplane. The signed distance d_i of a point $\mathbf{x}_i$ from the separating hyperplane $(\mathbf{w}, b)$ is given by

$$d_i = \frac{\mathbf{w} \cdot \mathbf{x}_i + b}{||\mathbf{w}||} \tag{10}$$

With (9) and (10), for all $\mathbf{x}_i \in S$, we have

$$y_i d_i \geq \frac{1}{||\mathbf{w}||} \tag{11}$$

Linear SVM (cont)

$$\forall \mathbf{x}_i \in S \quad y_i d_i \geq \frac{1}{||\mathbf{w}||}$$

Therefore, $\frac{1}{||\mathbf{w}||}$ is the lower bound on the distance between the points $\mathbf{x}_i$ and the separating hyperplane $(\mathbf{w}, b)$.

Definition 2 *The canonical representation of the separating hyperplane is obtained by rescaling the pair* $(\mathbf{w}, b)$ *into the pair* $(\mathbf{w}', b')$ *such that the distance of the closest point, say* $\mathbf{x}_j$ *equals* $\frac{1}{||\mathbf{w}'||}$.

Definition 3 *Given a linearly separable set S, the optimal separating hyperplane (OSH) is the separating hyperplane for which the distance of the closest point of S is maximum (i.e., maximize* $\frac{1}{||\mathbf{w}'||}$*).*

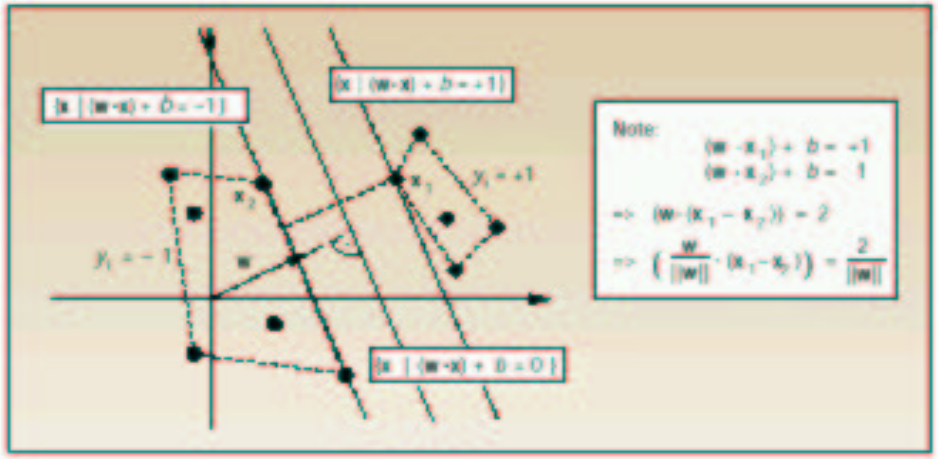


Figure 3: Linear separating hyperplane for the separable case [HSD+98]. Note that usually only a few points are on the margins.

Constrained Quadratic Programming

Problem 1

Minimize $\frac{1}{2}\mathbf{w} \cdot \mathbf{w}$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \qquad i = 1, 2, \ldots, N$

Let $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_N$ be the N nonnegative Lagrange multipliers associated with the constraints in (1), the solution to Problem 1 is equivalent to determining the saddle point of the function

$$L_P = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

with $L_P = L(\mathbf{w}, b, \alpha)$

Solving Constrained QP

At saddle point, L_P has minimum for $\mathbf{w} = \overline{\mathbf{w}}$ and $b = \bar{b}$ requiring

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} y_i \alpha_i = 0 \tag{12}$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{13}$$

with

$$\frac{\partial L}{\partial \mathbf{w}} = (\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \cdots, \frac{\partial L}{\partial w_n})$$

Since these are equality constraints in the dual formulation, we can substitute them into L_P to give

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i D_{ij} \alpha_j \tag{14}$$

Solving Constrained QP using Dual

Problem 2

$$\begin{aligned} &\textit{Maximize} && -\tfrac{1}{2}\alpha^T D\alpha + \sum_{i=1}^{N} \alpha_i \\ &\textit{subject to} && \sum_{i=1}^{N} y_i\alpha_i = 0 \\ & && \alpha \geq 0 \end{aligned}$$

where D is an $N \times N$ matrix such that

$$D_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{15}$$

For the solution at the saddle point, $(\overline{\mathbf{w}}, \overline{b})$, it follows that from Problem 2 that

$$\overline{\mathbf{w}} = \sum_{i}^{N} \overline{\alpha}_i y_i \mathbf{x}_i \tag{16}$$

Solving Constrained QP Using Dual

b can be determined from $\overline{\alpha}$, which is a solution of the dual problem, and from the Kuhn-Tucker conditions

$$\overline{\alpha}_i(y_i(\overline{\mathbf{w}} \cdot \mathbf{x}_i + \overline{b}) - 1) = 0, \quad i = 1, \ldots, N \tag{17}$$

Recall equation (13) and constraints in Problem 1

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Note that the only $\overline{\alpha}_i$ that can be nonzero in (17) are those for which the constraints (9) are satisfied with the equality sign.

Support Vectors

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N. \;\; \overline{\mathbf{w}} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

Most of the constraints in (9) are satisfied with inequality signs i.e., most $\overline{\alpha}_i$ solved from the dual are null.

$\Rightarrow$ the vectors $\overline{\mathbf{w}}$ is a linear combination of a relative small percentage of the points $\mathbf{x}_i$.

$\Rightarrow$ these points are termed *support vectors* because they are the closest points from the OSH and the only points of S needed to determine the OSH (name of the game).

The problem of classify a new data point $\mathbf{x}$ is now simply solved by looking at the sign of

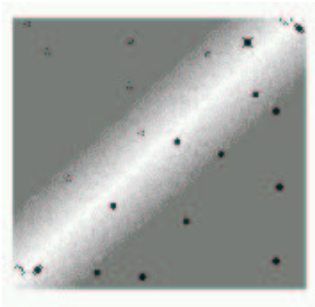
$$sgn(\overline{\mathbf{w}} \cdot \mathbf{x} + \overline{b})$$

Soft Margin Classifier

In the case that the set S is not linearly separable or one simply ignore whether or not the set S is linearly separable, the previous analysis can be generalized by introducing N nonnegative variable $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$ such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \tag{18}$$

Purpose: to allow for a small number of misclassified points for better generalization or computational efficiency.



Generalized OSH

The generalized OSH is then regarded as the solution to

Problem 3

$$\begin{aligned} &\textit{Minimize} && \tfrac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ &\textit{subject to} && y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1,\ldots,N \\ & && \xi \geq \mathbf{0} \end{aligned}$$

Role of C:

- as a regularization parameter (cf. Radial Basis Function, fitting).
- large $C \Rightarrow$ minimize the number of misclassified points.
- small $C \Rightarrow$ maximize the minimum distance $\frac{1}{||\mathbf{w}||}$.

Dual Problem

Problem 4

Maximize $\quad -\frac{1}{2}\alpha^T D\alpha + \sum_{i=1}^N \alpha_i$

subject to $\quad \sum_{i=1}^N y_i\alpha_i = 0 \quad 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, N$

As before,

$$\overline{\mathbf{w}} = \sum_{i=1}^{N} \overline{\alpha}_i y_i \mathbf{x}_i$$

and $\overline{b}$ can be determined from $\overline{\alpha}$, solution of the dual Problem 4 and from the new Kuhn-Tucker conditions

$$\overline{\alpha}_i(y_i(\overline{w} \cdot \mathbf{x}_i + \overline{b}) - 1 + \overline{\xi}) = 0 \tag{19}$$

$$(C - \overline{\alpha}_i)\overline{\xi} = 0 \tag{20}$$

where $\overline{\xi}$ are the values of the ξ at the saddle point. Similar to separable case, the points $\mathbf{x}_i$ for which $\overline{\alpha}_i > 0$ are termed *support vectors*.

Two cases:

- $\overline{\alpha_i} < C$

  $\Rightarrow \quad \overline{\mathbf{x}}_i = 0$

  $\Rightarrow$ the support vectors lie at a distance $\frac{1}{||\mathbf{w}||}$ from the OSH

  $\Rightarrow$ called *margin vectors*

- $\overline{\alpha_i} = C$

  1. $\xi_i > 1$, misclassified points
  2. $0 < \xi_i \leq 1$, points correctly classified but closer than $\frac{1}{||\mathbf{w}||}$ from the OSH
  3. $\xi = 0$, margin vectors (rare case)

Neglecting the last rare case, we refer to all the support vectors for which $\alpha_i = C$ as errors. All the points that are not support vectors are correctly classified and lie outside the margin strip.

Nonlinear Support Vector Machine

- Note that the only way the data points appear in the training problem is in the form of dot products $\mathbf{x}_i \cdot \mathbf{x}_j$.
- In higher dimensional space (feature space), it is very likely that a linear separator (hyperplane) can be constructed.
- E.g.: we map the data points to input space $\mathcal{R}^n$ to some feature space of higher dimension, $\mathcal{R}^m$, $(m > n)$ using function Φ.

$$\Phi : \mathcal{R}^n \rightarrow \mathcal{R}^m$$

Example:

$$\begin{array}{lcl} \Phi : \mathcal{R}^2 & \rightarrow & \mathcal{R}^3 \\ \mathbf{x} = (x_1, x_2) & \rightarrow & \mathbf{x}^{'} = (x_1^2, x_2^2, x_1 x_2) \end{array}$$

- Then the training algorithm would only depend on the dot products in $\mathcal{H}$, i.e., on the functions of the form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$.

In other words,

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{N} y_i \alpha_i \mathbf{x}_i \cdot \mathbf{x} + b$$

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{N} y_i \alpha_i \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}) + b$$

- But the transformation operator, Φ, is computationally expensive.
- 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.
- One example, $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2}$.
- All the previous derivations in linear SVM hold (substituting dot product with kernel function), since we are still doing a linear separation, but in a different space.

- Map the training data nonlinearly into a higher-dimensional feature space via Φ, and construct a separating hyperplane with maximum margin there.
- This yields a nonlinear decision boundary in input space. By the use of kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space.

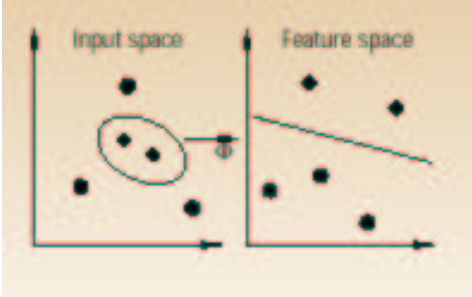


Figure 4: Mapping between input space and feature space [HSD+98].

- Question: Is a feature space always more expressive than input space? How do we determine kernel functions?

Mercer's Condition for Kernel Function

- The idea of constructing support vector networks comes form considering general forms of the dot product in a Hilbert space.

$$\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) \equiv K(\mathbf{u}, \mathbf{v}) \tag{21}$$

- Question: Which kernel does there exist a pair $\{\mathcal{H}, \Phi\}$ such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$
- Answer: Mercer's condition. It tells us whether or not a prospective kernel is actually a dot product in some space.
- According to the Hilbert-Schmidt Theory, any symmetric function $K(\mathbf{u}, \mathbf{v})$, with $K(\mathbf{u}, \mathbf{v}) \in L_2$, can be expanded in the form

$$K(\mathbf{u}, \mathbf{v}) = \sum_i \lambda_i \Phi_i(\mathbf{u}) \cdot \Phi(\mathbf{v}) \tag{22}$$

where $\lambda_i \in \mathcal{R}$ and Φ are eigenvalues and eigenfunctions

$$\int K(\mathbf{u}, \mathbf{v})\phi(\mathbf{u})d\mathbf{u} = \lambda_i \Phi_i(\mathbf{v})$$

of the integral operator defined by the kernel $K(\mathbf{u}, \mathbf{v})$.

- A sufficient condition to ensure that (21) defines a dot product in a feature space is that all the eigenvalues in the expansion (22) are positive. To guarantee that these coefficients are positive, it is necessary and sufficient (Mercer's theorem) that the condition

$$\int\int K(\mathbf{u}, \mathbf{v})g(\mathbf{u})g(\mathbf{v})d\mathbf{u}d\mathbf{v} > 0$$

is satisfied for all g such that

$$\int g^2(\mathbf{u})d\mathbf{u} < \infty$$

Some Kernel Functions in SVM

Simple dot product:

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$$

Vovk's polynomial:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$

Radial basis function (RBF):

$$K(\mathbf{x}, \mathbf{y}) = e^{-||\mathbf{x}-\mathbf{y}||^2/2\sigma^2}$$

Two layer neural network:

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta)$$

Some Properties of SVM

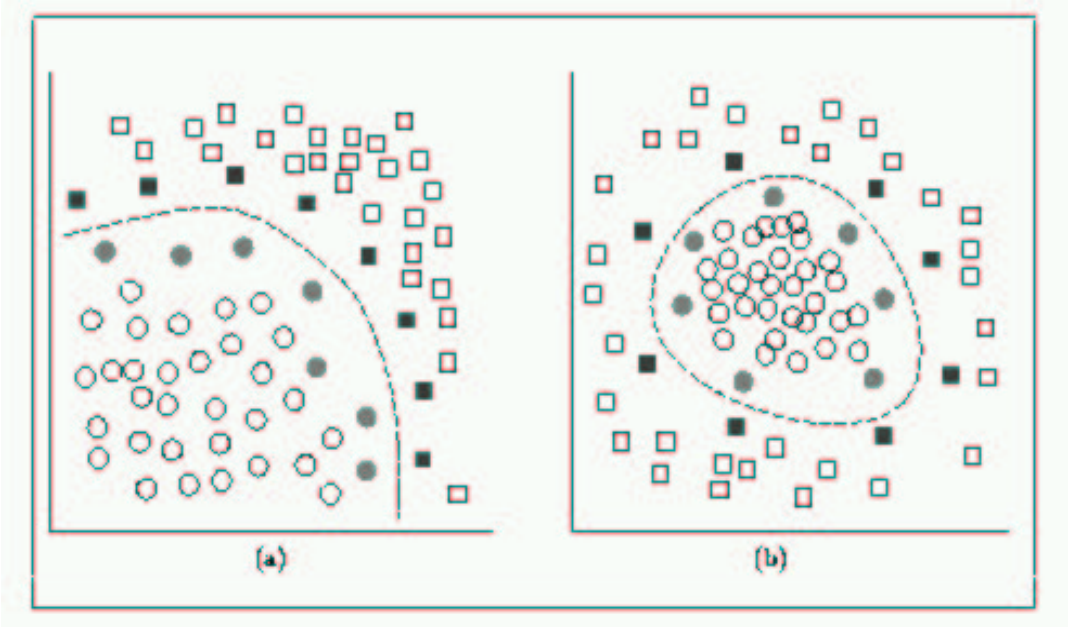


Figure 5: Decision surface in (a) by a polynomial classifier, and in (b) by a RBF where the support vectors are indicated in dark fill. Note the reduced number of them and their position close to the boundary. In (b), the support vectors are the RBF centers [OFG97a].

Architecture for General Support Vector Machine

- Linear SVM: Kernel function is just a dot product in input space
- Nonlinear SVM: Need to choose an appropriate kernel function

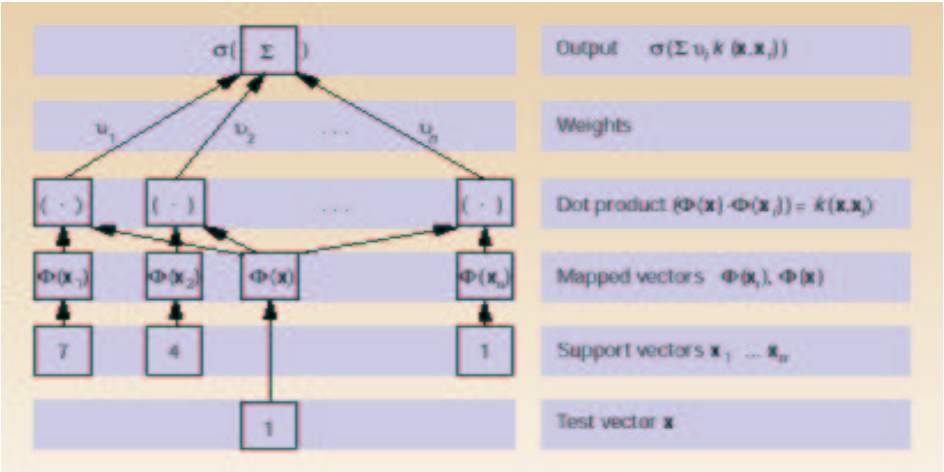


Figure 6: Architecture of SVM methods [HSD+98].

Multiple Classes

For K-class pattern recognition problem, several approaches have been proposed

- One-against-the-rest: construct a hyperplane between class k and the $K-1$ other classes $\Rightarrow K$ SVM's.
- One-against-one: construct a hyper plane for any two classes $\Rightarrow \frac{K(K-1)}{2}$ SVM's.
- K-class SVM: by Watkins.
- John Platt's DAG method

Applications

- Pattern Recognition: hand digit recognition, 3D object recognition, face detection, face detection, pedestrian detection, gender classification, expression recognition, speaker identification, text classification.
- Regression Estimation: time series prediction.
- Signal Processing: seismic signal classification, density estimation, DNA sequence classification.

Life Beyond SVM

- Mistake-Bound On-Line Learning: Winnow, SNoW
- Ensemble of Homogeneous Classifiers: Boosting, Bagging
- Ensemble of Heterogeneous Classifiers: Kittler's method
- Random Subspace Method: Monte Carlo approach
- Kernel methods: Kernel PCA, Kernel Fisher Linear Discriminant
- Generative Models, Graphical Models, Nonlinear PCA, Probabilistic PCA, Mixture of Probabilistic PCA, etc.
- Maximum entropy approach
- Feature selection
- Gaussian process

SVM v.s. SNoW on Face Detection

A benchmark on SVM and SNoW based on 5732 training samples and 500 testing samples using a Sun Ultra Sparc 10. Each sample is an 20×20 image.

	Nonlinear SVM	Linear SVM	SNoW
Training Accuracy	100%	100%	100%
Testing Accuracy	100%	96%	97%
Memory Requirement	83MB	24MB	7MB
World Clock Time	5.8hr	3.7hr	0.6hr

Concluding Remarks

Pros

- Optimal hyperplane.
- Some kernels have infinite VC dimension.
- Can deal with high dimensional data.

Cons

- Numerical stability problems in solving constrained QP.
- Usually require positive/negative examples.
- Need to select a good kernel function.
- Require lots of memory and CPU time.

References

Introductory articles: [HSD+98] [SS98] [PV98] [OFG97b] [Bur98] [CV95]
Book: [Hay98] [Vap95] [Vap98] [SBS98] [CST00]
Ph.D. Thesis: [Cor95] [Sch97] [Smo98]
Vision papers: [PV98] [OFG97b] [OPS+97] [POP98]
Comparison with RBF: [SSB+97]
Kernel Machines web site: http://www.kernel-machines.org
Boosting web site: http://www.boosting.org/

References

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[Cor95] C Cortes. *Prediction of Generalization Ability in Learning Machines*. PhD thesis, Department of Computer Science, University of Rochester, 1995.

[CST00] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20, 1995.

[Hay98] S. Haykin. *Neural Networks : A Comprehensive Foundation*. Prentice Hall, 1998.

[HSD+98] M. A. Heasrt, B. Scholkopf, S. Dumais, E. Osuna, and J. Platt. Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, 1998.

[OFG97a] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report

AI MEMO 1602, MIT AI Lab, 1997.

[OFG97b] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[OPS+97] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[POP98] Constantine Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 555–562, 1998.

[PV98] Massimiliano Pontil and Alessandro Verri. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[SBS98] B. Scholkopf, C. Burges, and A. Smola, editors. *Advances in Kernel Methods*. MIT Press, 1998.

[Sch97] Bernhard Scholkopf. *Support Vector Learning*. PhD thesis, Informatik der Technischen Universitat Berlin, 1997.

[Smo98] A. Smola. *Learning with Kernels*. PhD thesis, GMD, 1998.

[SS98] A. J. Smola and B. Scholkopf. A tutorial on support vector regression. Technical Report TR-1998-030, Neuro COLT, GMD First, 1998.

[SSB+97] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi,

and T. Poggio. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765, 1997.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communications, and Control)*. John Wiley & Sons, 1998.