Support Vector Machines - an Introduction

Ron Meir

Department of Electrical Engineering
Technion, Israel

Sources of Information

Web http://www.kernel-machines.org/

Tutorial C.J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition (download from above site)

Books

- ⋆ V. Vapnik, *Statistical Learning Theory*, 1998.
- ⋆ N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, 2000.
- ⋆ A. Smola and B. Schölkopf, *Learning with Kernels*, 2002.

Optimization book D. Bertsekas, *Nonlinear Programming*, Second Edition 1999.

Application Domains

Supervised Learning

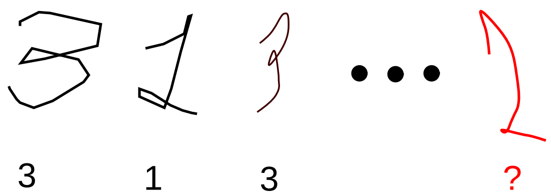
- ⋆ Pattern Recognition - state of the art results for OCR, text classification, Biological sequencing
- ⋆ Regression and time series - good results

Unsupervised Learning

- ⋆ Dimensionality Reduction - Non linear principal component analysis
- ⋆ Clustering
- ⋆ Novelty detection

Reinforcement Learning: Some preliminary results

Classification I



The problem:

Input:	$\mathbf{x}$ feature vector
Label:	$y \in \{1, 2, \ldots, k\}$
Data:	$\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
Unknown source:	$\mathbf{x} \sim p(\mathbf{x})$
Target:	$y = f(x)$

Objective: Given **new** x, predict y so that **probability of error is minimal**

CLASSIFICATION II

The 'Model'

Hypothesis class: $\mathcal{H} : \mathbb{R}^d \mapsto \{\pm 1\}$

Loss: $\ell(y, h(x)) = I[y \neq h(x)])$

Generalization: $L(h) = \mathbf{E}\{\ell(Y, h(X))\}$

Objective: Find $h \in \mathcal{H}$ which minimizes $L(h)$

Caveat: Only have **data** at our disposal

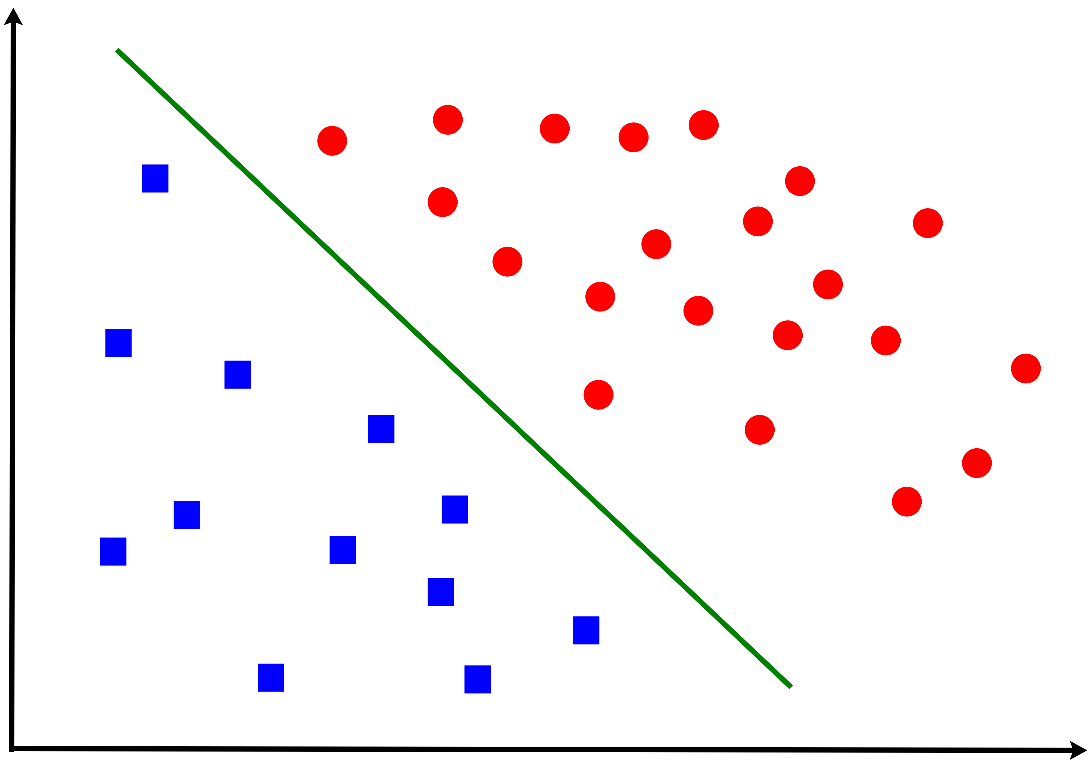
'Solution': Form empirical estimator which 'generalizes well'

Question: How can we efficiently construct complex hypotheses with good generalization?

Focus: Two-class problem, $y \in \{-1, +1\}$

Linearly Separable Classes

$$Y = \mathrm{sgn}[\mathbf{w}^\top \mathbf{x} + b > 0] = \begin{cases} +1 & \mathbf{w}^\top \mathbf{x} + b > 0 \\ -1 & \mathbf{w}^\top \mathbf{x} + b \leq 0 \end{cases}$$



Problem: Many solutions! Some are very poor

Task: Based on data, select hyper-plane which works well 'in general'

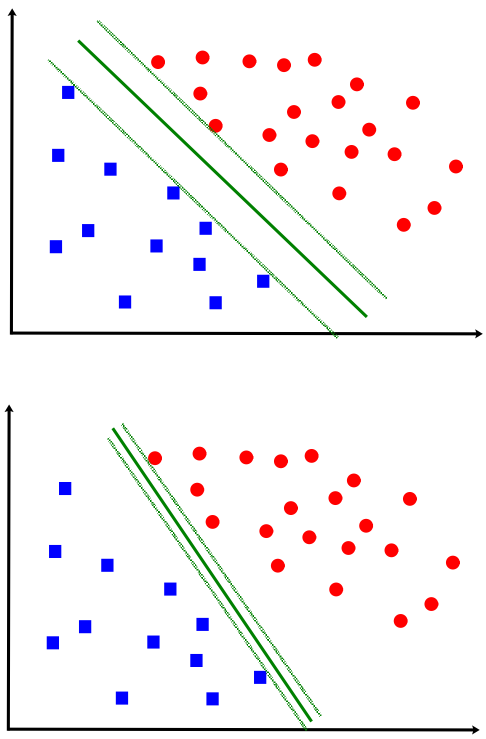
Selection of a Good Hyper-Plane

Objective: Select a 'good' hyper-plane using only the data!

Intuition: (Vapnik 1965) - assuming linear separability

(i) Separate the data

(ii) Place hyper-plane 'far' from data



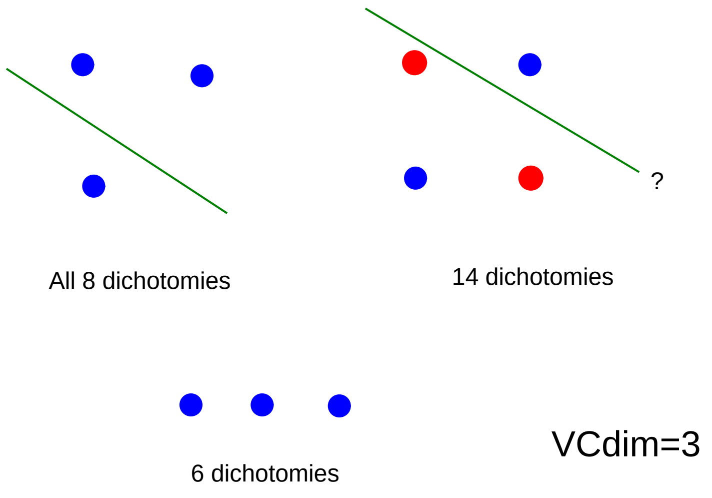
VC DIMENSION

Given: $\mathcal{H} = \{h : \mathbb{R}^d \mapsto \{-1, +1\}\}$

Question: How complex is the class?

§

Shattering: $\mathcal{H}$ shatters a set X if $\mathcal{H}$ achieves all dichotomies on X



VC-dimension The size of the largest shattered subset of X

Hyper-planes $\text{VCdim}(\mathcal{H}) = d + 1$

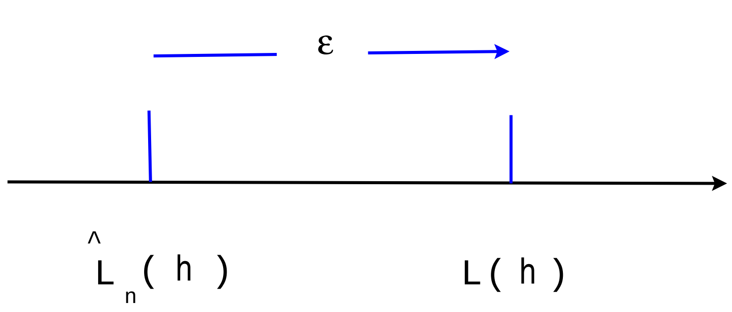
WHAT IS THE TRUE PERFORMANCE?

For $h \in \mathcal{H}$

$L(h)$	Probability of miss-classification
$\hat{L}_n(h)$	Empirical fraction of miss-classifications

Vapnik and Chervonenkis 1971: For **any** distribution with prob. $1 - \delta$, $\forall h \in \mathcal{H}$,

$$L(h) < \underbrace{\hat{L}_n(h)}_{\text{emp. error}} + \underbrace{c\sqrt{\frac{\mathrm{VCdim}(\mathcal{H}) \log n + \log \frac{1}{\delta}}{n}}}_{\text{complexity penalty}}$$

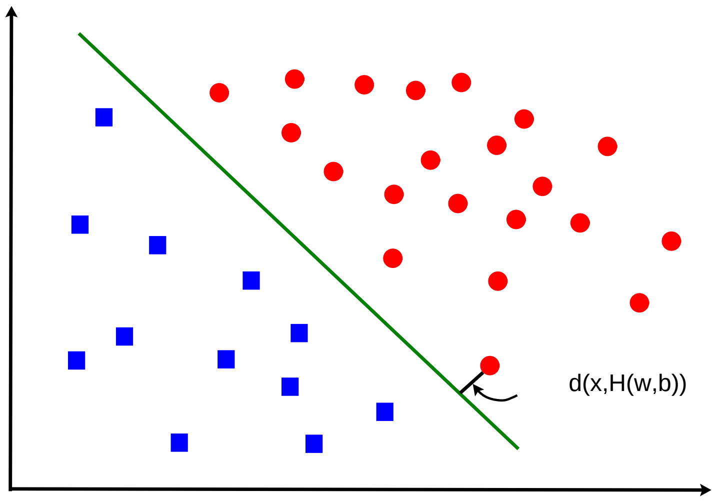


An Improved VC Bound I

Hyper-plane: $H(\mathbf{w}, b) = \{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + b = 0\}$

Distance of a point from a hyper-plane:

$$d(\mathbf{x}, H(\mathbf{w}, b)) = \frac{\mathbf{w}^\top \mathbf{x} + b}{\|\mathbf{w}\|}$$



Optimal hyper-plane (linearly separable case)

$$\boxed{\max_{\mathbf{w}, b} \min_{1 \leq i \leq n} d(\mathbf{x}_i, H(\mathbf{w}, b))}$$

An Improved VC Bound II

Canonical hyper-plane:

$$\min_{1 \leq i \leq n} |\mathbf{w}^\top \mathbf{x}_i + b| = 1$$

(No loss of generality)

Improved VC Bound (Vapnik 95) VC dimension of set of canonical hyper-planes such that

$$\|\mathbf{w}\| \leq A$$
$$\mathbf{x}_i \in \text{Ball of radius } L$$

is

$$\boxed{\text{VCdim} \leq \min(A^2 L^2, d) + 1}$$

Observe: Constraints reduce VC-dim bound

Canonical hyper-planes with **minimal norm** yields best bound

Suggestion: Use hyper-plane with minimal norm

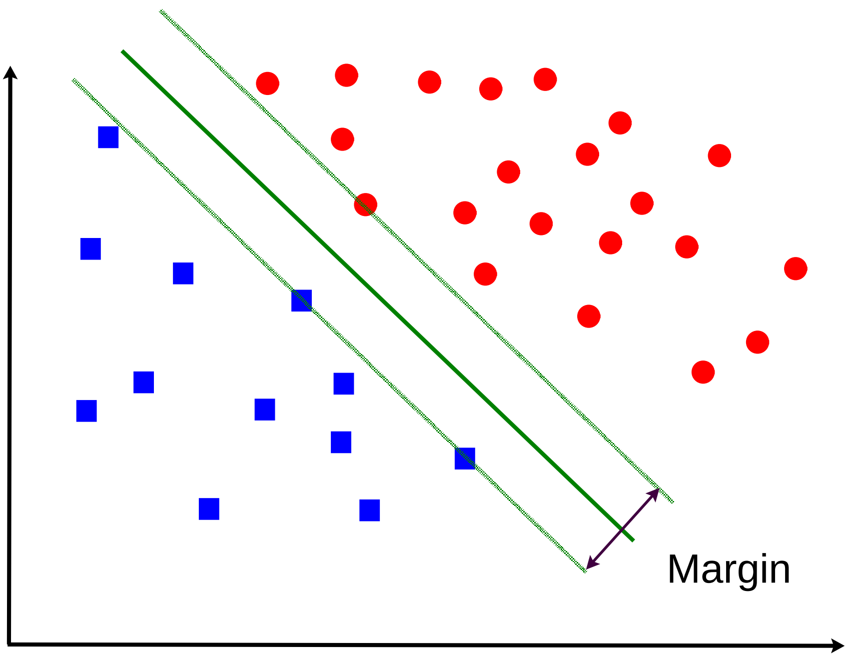
The Optimization Problem I

Canonical hyper-planes: $(\mathbf{x}_i, \mathbf{w} \in \mathbb{R}^d)$

$$\min_{1 \le i \le n} |\mathbf{w}^\top \mathbf{x}_i + b| \ge 1$$

Support vectors:

$$\{\mathbf{x}_i : |\mathbf{w}^\top \mathbf{x}_i + b| = 1\}$$



Margin Distance between hyper-planes defined by support vectors

The Optimization Problem II

Distance from support vector to $H(\mathbf{w}, b)$

$$\frac{\mathbf{w}^\top \mathbf{x}_i + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|}$$

$$\textbf{Margin} = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

$$\begin{array}{lll} \text{minimize} & \frac{1}{2}\mathbf{w}^\top \mathbf{w} & \\ \text{subject to} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 & i = 1, 2, \ldots, n \end{array}$$

1. Convex quadratic program
2. Linear inequality constraints (many!)
3. $d + 1$ parameters, n constraints

Convex Optimization

Problem:

$$\begin{array}{rll} \text{minimize} & f(\mathbf{x}) & \\ \text{subject to} & h_i(\mathbf{x}) = 0, & i = 1, \ldots, m, \\ & g_j(\mathbf{x}) \leq 0, & j = 1, \ldots, r \end{array}$$

Active constraints: $A(\mathbf{x}) = \{j : g_j(\mathbf{x}) = 0\}$

Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{r} \mu_j h_j(\mathbf{x})$$

Sufficient conditions for minimum: (KKT)

Let $\mathbf{x}^*$ be a **local** minimum. Then $\exists \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ s.t.

$$\begin{array}{ll} \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0 & \\ \mu_j^* \geq 0 & j = 1, 2, \ldots, r \\ \mu_j^* = 0 & \forall j \notin A(x^*) \end{array}$$

The Dual Problem I

Motivation:

- ⋆ Many **inequality** constraints
- ⋆ High (sometimes infinite) input dimension

Primal Problem

$$
\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & e_i^\top x = d_i, \qquad i = 1, \ldots, m, \\
& a_j^\top x \leq b_j, \qquad j = 1, \ldots, r
\end{aligned}
$$

Lagrangian

$$L_P(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i (e_i^\top x - d_i) + \sum_{j=1}^{r} \mu_j (a_j^\top \mathbf{x} - b_j)$$

The Dual Problem II

Dual Lagrangian

$$L_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{x}} L_P(\mathbf{x}, \lambda, \mu)$$

Dual Problem

$$\begin{array}{ll} \underset{\boldsymbol{\lambda},\boldsymbol{\mu}}{\text{maximize}} & L_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{subject to} & \boldsymbol{\mu} \geq \mathbf{0} \end{array}$$

Observation:

- ⋆ $L_P(x, \lambda, \mu)$ quadratic $\Rightarrow L_D(\lambda, \mu)$ quadratic
- ⋆ Constraints in Dual greatly simplified
- ⋆ $m + r$ variables, r constraints

Duality Theorem:
Optimal solutions of **P** and **D** coincide

SVM in the Primal Space I

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to} \quad y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \quad i = 1, \dots, n.$$

$$L_P(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1],$$

Solution:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$
$$0 = \sum_{i=1}^{n} \alpha_i y_i \qquad (\alpha_i \geq 0)$$

KKT condition:

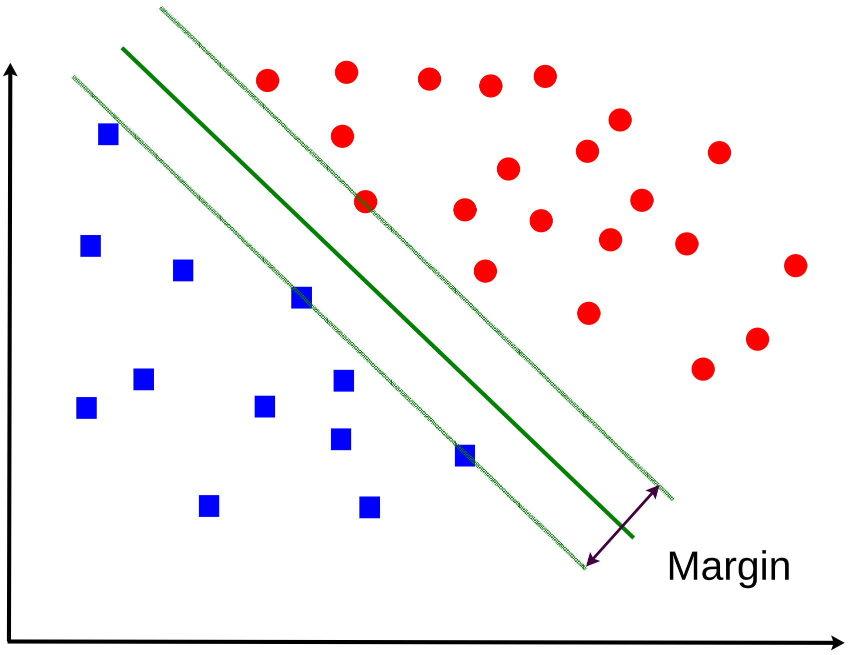
$$\alpha_i = 0 \text{ unless } y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$$

Sparsity: Often many α_i vanish!

SVM in the Primal Space II

Recall

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$



Support vectors: All $\mathbf{x}_i$ for which $\alpha_i > 0$

Occurs if constraint is obeyed with equality

Support Vectors in the Dual Space

$$\text{max.} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad \alpha_i \geq 0$$

Determination of b: For support vectors

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$$

Thus

$$b^* = -\frac{1}{2} \left(\min_{y_i=+1} \{\mathbf{w}^{*T} \mathbf{x}_i\} + \max_{y_i=-1} \{\mathbf{w}^{*T} \mathbf{x}_i\} \right)$$

Classifier: (Recall $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$)

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i^\top \mathbf{x} + b^* \right)$$

Non-Separable Case I

Objective: find a good separating hyper-plane for the non-separable case

Problem: Cannot satisfy $y_i[\mathbf{w}^\top \mathbf{x}_i + b] \geq 1$ for all i

Solution: *Slack* variables

$$
\begin{aligned}
\mathbf{w}^\top \mathbf{x}_i + b &\geq +1 - \xi_i && \text{for } y_i = +1, \\
\mathbf{w}^\top \mathbf{x}_i + b &\leq -1 + \xi_i && \text{for } y_i = -1, \\
\xi_i &\geq 0 && k = 1, 2, \ldots, n.
\end{aligned}
$$

An error occurs if $\xi_i > 1$. Thus,

$$\sum_{i=1}^{n} I(\xi_i > 1) = \#\ \text{errors}$$

Non-Separable Case II

Proposed solution: Minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} I(\xi_i > 1) \qquad \textbf{(non - convex!)}$$

Suggestion: Replace $I(\xi_i > 1)$ by ξ_i (upper bound)

$$\begin{array}{ll} \underset{w,b,\xi}{\text{minimize}} & L_P(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

Tradeoff: Large C - penalize errors, Small C penalize complexity

Dual Problem: Same as in separable case, except that $0 \leq \alpha_i \leq C$

Support vectors: $\alpha_i > 0$ - but lose geometric interpretation!

Non-Separable Case III

Solution:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_i \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b\right)$$

KKT conditions:

$$0 = \sum_{i=1}^{n} \alpha_i y_i$$

$$0 = \alpha_i (y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) - 1 + \xi_i)$$

$$0 = (C - \alpha_i)\xi_i$$

Support vectors: characterized by $\alpha_i > 0$

Non-Separable Case IV

Two types of support vectors:

Recall

$$\alpha_i(y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) - 1 + \xi_i\right) = 0$$
$$(C - \alpha_i)\xi_i = 0$$

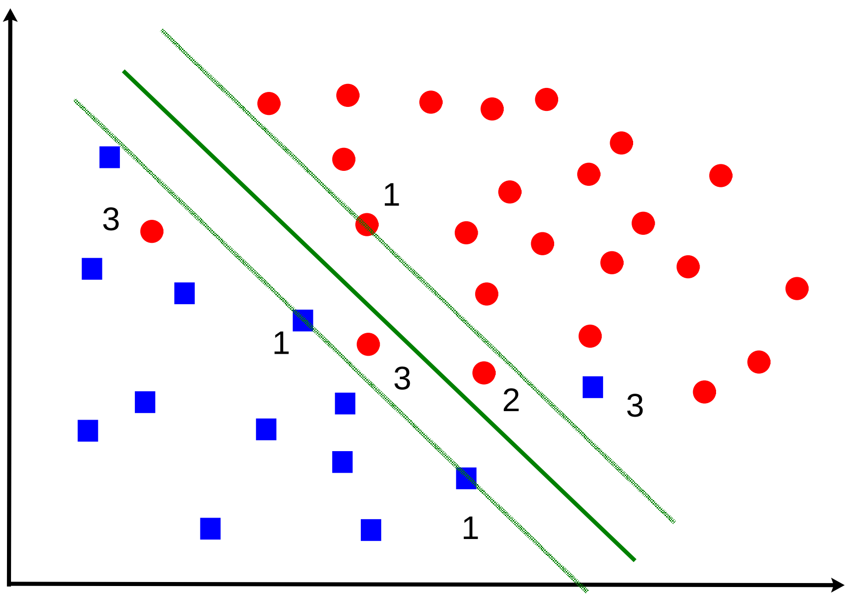
Margin vectors:

$$0 < \alpha_i < C \Rightarrow \xi_i = 0 \quad \Rightarrow \quad d(\mathbf{x}_i, H(\mathbf{w}, b)) = \frac{1}{\|\mathbf{w}\|}$$

Non-margin vectors: $\alpha_i = C$

- **Errors:** $\xi_i > 1$ Misclassified
- **Non-errors:** $0 \leq \xi_i \leq 1$ Correcty classified
  Within margin

Non-Separable Case V



Support Vectors:

1	margin s.v.	$\xi_i = 0$	Correct
2	non-margin s.v.	$\xi_i < 1$	Correct (in margin)
3	non-margin s.v.	$\xi_i > 1$	Error

Problem: Lose clear geometric intuition and sparsity

Non-linear SVM I

Linear Separability: More likely in high dimensions

Mapping: Map input into high-dimensional *feature space* $\mathbf{\Phi}$

Classifier: Construct *linear* classifier in $\mathbf{\Phi}$

Motivation: Appropriate choice of $\mathbf{\Phi}$ leads to linear separability.

Non-linearity and **high dimension** are essential (Cover '65)!

$$\mathbf{\Phi} : \mathbb{R}^d \mapsto \mathbb{R}^D \qquad (D \gg d)$$
$$\mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x})$$

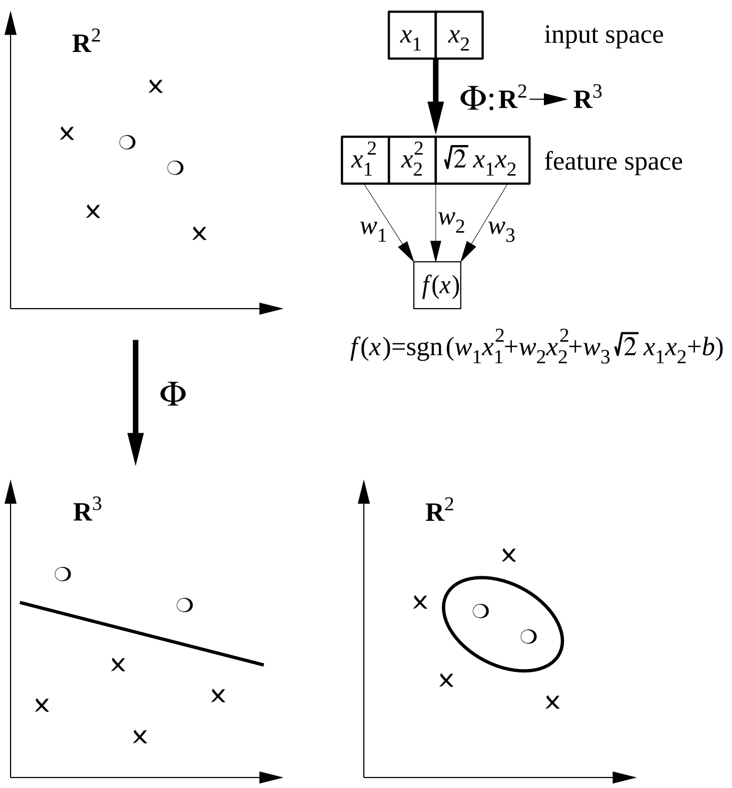
Hyper-plane condition: $\mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}) + b = 0$

Inner products: $\mathbf{x}^\top \mathbf{x} \mapsto \mathbf{\Phi}^\top(\mathbf{x}) \mathbf{\Phi}(\mathbf{x})$

NON-LINEAR SVM II

Decisions in Input and Feature Space:

Problems becomes linearly separable in feature space (Fig. from Schölkopf & Smola 2002)



Recall for linear SVM

$$f(\mathbf{x}) = \sum_{i=1}^{d} w_i x_i + b \quad ; \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Non-Linear SVM III

Obtained

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b$$

In feature space

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{\Phi}(\mathbf{x}_i)^\top \mathbf{\Phi}(\mathbf{x}) + b$$

Kernel: A symmetric function $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$

Inner product kernels: In addition

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{\Phi}(\mathbf{x})^\top \mathbf{\Phi}(\mathbf{z})$$

Motivation: $\mathbf{\Phi} \in \mathbb{R}^D$, where D may be very large - inner products expensive

Non-Linear Support Vectors IV

Examples:

Linear mapping: $\mathbf{\Phi}(\mathbf{x}) = A\mathbf{x}$

$$K(\mathbf{x}, \mathbf{z}) = (A\mathbf{x})^\top (A\mathbf{z}) = \mathbf{x}^\top A^\top A\mathbf{z} = \mathbf{x}^\top B\mathbf{z}$$

Quadratic map: $\mathbf{\Phi}(\mathbf{x}) = \{(x_i x_j)\}_{i,j=1}^{d,d}$

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^\top \mathbf{z})^2 \\ &= \left(\sum_{i=1}^{d} x_i z_i \right)^2 \\ &= \sum_{i,j=(1,1)}^{d,d} (x_i x_j)(z_i z_j) \end{aligned}$$

Objective: Work directly with kernels, **avoiding** mapping $\mathbf{\Phi}$

Question: Under what conditions is $K(\mathbf{x}, \mathbf{z}) = \mathbf{\Phi}(\mathbf{x})^\top \mathbf{\Phi}(\mathbf{z})$?

Mercer Kernels I

Assumptions:

1. $K(\mathbf{x}, \mathbf{z})$ a continuous symmetric function
2. K is positive definite: for any $f \in L_2$ not identically zero

$$\int f(\mathbf{x}) K(\mathbf{x}, \mathbf{z}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} > 0$$

Mercer's Theorem:

$$K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{z})$$

$$\int K(\mathbf{x}, \mathbf{z}) \psi_j(\mathbf{z}) d\mathbf{z} = \lambda_j \psi_j(\mathbf{x})$$

Conclusion: Let $\phi_j(\mathbf{x}) = \sqrt{\lambda_j} \psi_j(\mathbf{x})$, then

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{\Phi}(\mathbf{x})^\top \mathbf{\Phi}(\mathbf{z})$$

Mercer Kernels II

Classifier:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{\Phi}(\mathbf{x}_i)^\top \boldsymbol{\Phi}(\mathbf{x}) + b\right)$$
$$= \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

The gain: Implement **infinite-dimensional** mapping, but do all calculations in **finite dimension**

maximize	$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$
subject to	$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad 0 \le \alpha_i \le C$

Observe: Only difference from linear case is in the kernel

Optimization task is unchanged!

KERNEL SELECTION I

Simpler Mercer conditions: for any finite set of points $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is positive-definite

$$\mathbf{v}^\top \mathbf{K} \mathbf{v} > 0$$

Classifier:

$$\boxed{f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right)}$$

Constructing kernels: Assume K_1 and K_2 kernels, f real-valued function

1. $K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z})$
2. $K_1(\mathbf{x}, \mathbf{z}) K_2(\mathbf{x}, \mathbf{z})$
3. $f(\mathbf{x}) f(\mathbf{z})$
4. $K_3(\boldsymbol{\Phi}(\mathbf{x}), \boldsymbol{\Phi}(\mathbf{z}))$

Kernel Selection II

Explicit construction:

1. $p(K_1(\mathbf{x}, \mathbf{y}))$ - polynomial with positive coefficients
2. $\exp[K(\mathbf{x}, \mathbf{z})]$
3. $\exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2)$

Some standard kernels:

Polynomial	$(\mathbf{x}^\top \mathbf{x} + 1)^p$
Gaussian	$\exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right)$
Splines	piece-wise polynomial between knots

Kernel Selection III

Adaptive kernels?

- ⋆ Determine kernel parameters (e.g. width) using cross-validation
- ⋆ Improved bounds - take into account properties of kernels
- ⋆ Use invariances to constrain kernels

Applications: State of the art in many domains. Can deal with huge data sets

Hand-writing recognition

Text classification

Bioinformatics

Many more

The Kernel Trick - Summary

- Can be thought of as non-linear similarity measure
- Can be used with any algorithm that uses only inner products
- Allows constructing non-linear classifiers using only linear algorithms
- Can be applied to vectorial and non-vectorial data (e.g. tree and string structures)

SVM and Penalization I

Recall the linear problem

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}} \quad & L_P(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

Let $[u]_+ = \max(0, u)$

One can show **equivalence** to

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \left\{ \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda\|\mathbf{w}\|^2 \right\} \\ \text{subject to} \quad & f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + b \end{aligned}$$

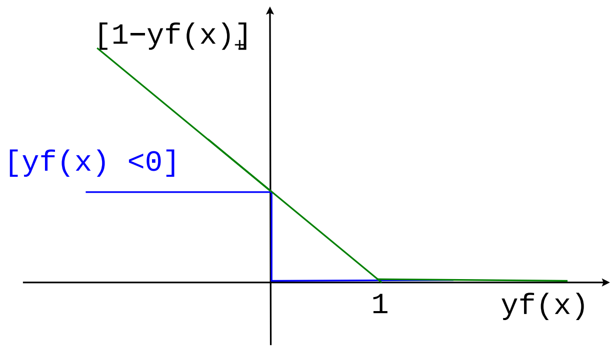
This has the classic form of **Regularization Theory**

empirical error + regularization term

SVM and Penalization II

$$\underset{w,b}{\text{minimize}} \left\{ \sum_{i=1}^{n} [1 - y_i f(\mathbf{x})]_+ + \lambda \|\mathbf{w}\|^2 \right\}$$

- ⋆ Maximize distance of points from the hyper-plane
- ⋆ Correctly classified points are also penalized



Suggestion: Consider a general formulation

$$\underset{w,b}{\text{minimize}} \quad \left\{ \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) + \lambda \|\mathbf{w}\|^2 \right\}$$

$$\text{subject to} \quad f(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x})^\top \mathbf{w} + b$$

SVM and Penalization III

Recall

$$\underset{w,b}{\text{minimize}} \quad \left\{ \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) + \lambda \|\mathbf{w}\|^2 \right\} \qquad (*)$$

$$\text{subject to} \quad f(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x})^\top \mathbf{w} + b$$

Representer Theorem: (special case) For any loss function $\ell(y, f(\mathbf{x}))$, the solution of $(*)$ is of the form

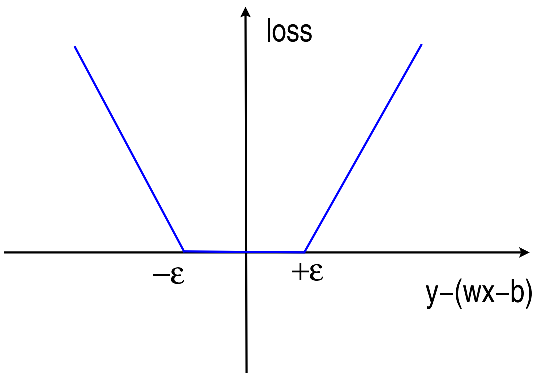
$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Compression bounds: Sparsity improves generalization! (A version of **Occam's razor**)

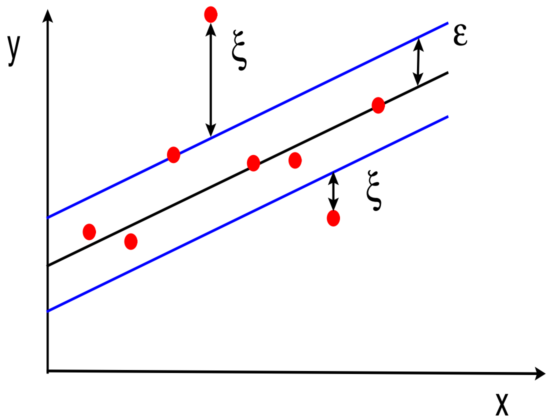
SVM Regression I

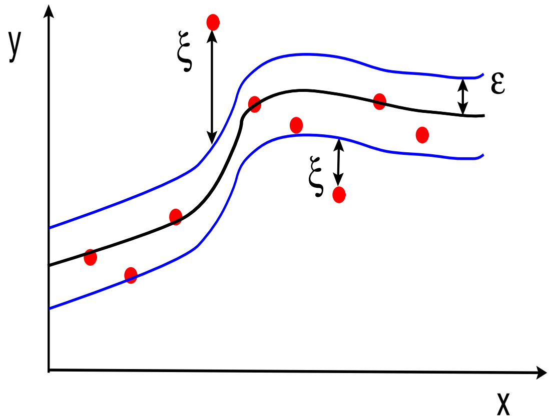
Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, $y_i \in \mathbb{R}$

Loss: Attempt to achieve **sparsity**



Achieve: Only badly predicted points contribute





SVM Regression II

Dual problem: Very similar to the case of classification

Solution: Solve a **quadratic optimization** problem in $2n$ variables α_i, α_i^*, $i = 1, 2, \ldots, n$.

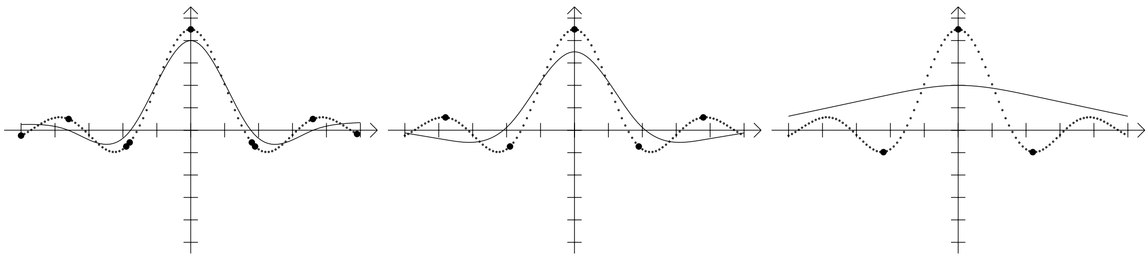
$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b$$

Sparsity: Only data for which $\alpha_i \neq \alpha_i^*$ contribute. This occurs only if

$$|f(\mathbf{x}_i) - y_i| \geq \epsilon \qquad \text{(outside tube)}$$

SVM REGRESSION III

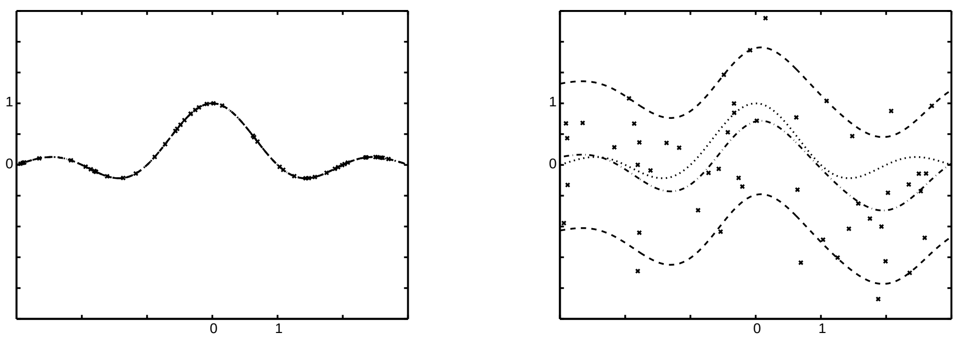
Effect of ϵ:



$$\epsilon = 0.1, 0.2, 0.5$$

Improved regression: ϵ-adaptive

Left - zero noise, right - $\sigma = 0.2$

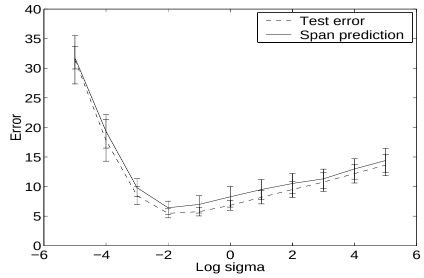


Figures: from Schölkopf and Smola 2002

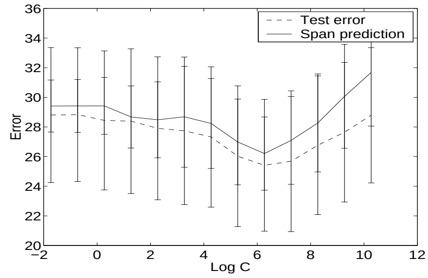
PRACTICALLY USEFUL BOUNDS?

Practically useful bound can be obtained using:

- ⋆ Data-dependent complexities
- ⋆ Specific learning algorithms
- ⋆ A great deal of ingenuity



(a) choice of σ in the postal database



(b) choice of C in the breast-cancer database

From: "Bounds on Error Expectations for Support Vector Machines", V. Vapnik and O. Chapelle, 2001 (200 examples in (b))

Summary I

Advantages

- ⋆ Systematic implementation through quadratic programming (very efficient implementations exist)
- ⋆ Excellent data-dependent generalization bounds exist
- ⋆ Regularization built into cost function
- ⋆ Statistical performance independent of dim. of feature space
- ⋆ Theoretically related to widely studied fields of regularization theory and sparse approximation
- ⋆ Fully adaptive procedures available for determining hyper-parameters

Summary II

Drawbacks

- ⋆ Treatment of non-separable case somewhat heuristic
- ⋆ Number of support vectors may depend strongly on the kernel type and and the hyper-parameters
- ⋆ Systematic choice of kernels is difficult (prior information) - some ideas exist
- ⋆ Optimization may require clever heuristics for large problems

SUMMARY III

Extensions

- ⋆ Online algorithms
- ⋆ Systematic choice of kernels using generative statistical models
- ⋆ Applications to
  - Clustering
  - Non-linear principal component analysis
  - Independent component analysis
- ⋆ Generalization bounds constantly improving (some even practically useful!)