Support Vector Machines

Stephan Dreiseitl
University of Applied Sciences
Upper Austria at Hagenberg

Harvard-MIT Division of Health Sciences and Technology
HST.951J: Medical Decision Support

Overview

- Motivation
- Statistical learning theory
- VC dimension
- Optimal separating hyperplanes
- Kernel functions
- Performance evaluation

Motivation

Given data $D = \{(x_i, t_i)\}$ distributed according to $P(x,t)$, which is better performance on test set?

0.7	0.5	0
-0.5	0.9	1
-0.2	-1.2	1
0.3	0.6	1
-0.2	0.5	0
0.8	-0.2	0

0.7	0.5	0
-0.5	0.9	1
-0.2	-1.2	1
0.3	0.6	1
-0.2	0.5	0
0.8	-0.2	1

0.7	0.5	0
-0.5	0.9	1
-0.2	-1.2	1
0.3	0.6	1
-0.2	0.5	1
0.8	-0.2	0

0.7	0.5	0
-0.5	0.9	1
-0.2	-1.2	1
0.3	0.6	1
-0.2	0.5	1
0.8	-0.2	1

Motivation

- Neural networks model $p(t|x)$ by
  - Topology restriction
  - Early stopping
  - Weight decay
  - Bayesian approach
- SVM: *capacity control*
- Based on statistical learning theory

Statistical learning theory

- Given data $D = \{(x_i, t_i)\}$, model output $y(\alpha, x_i) \in \{+1, -1\}$, class labels $t_i \in \{+1, -1\}$
- Fundamental question: Is learning *consistent*?
- Can we infer performance on test set (generalization error) from performance on training set?

Statistical learning theory

Average error on a data set *D* for model with parameter α:

$$R_{\text{emp}}(\alpha) = \frac{1}{2n} \sum_{i=1}^{n} | \, y(\alpha, x_i) - t_i \, |$$

Expected error of same model given unseen data distributed like *D*:

$$R(\alpha) = \frac{1}{2} \int | \, y(\alpha, x) - t \, | \, dP(x, t)$$

Statistical learning theory

- How can we relate R_{emp} and R?
- Generalization error $R(\alpha)$ depends on empirical error $R_{emp}(\alpha)$ and capacity h of model
- With probability 1-η :

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

h is VC dimension (*Vapnik-Chervonenkis*)

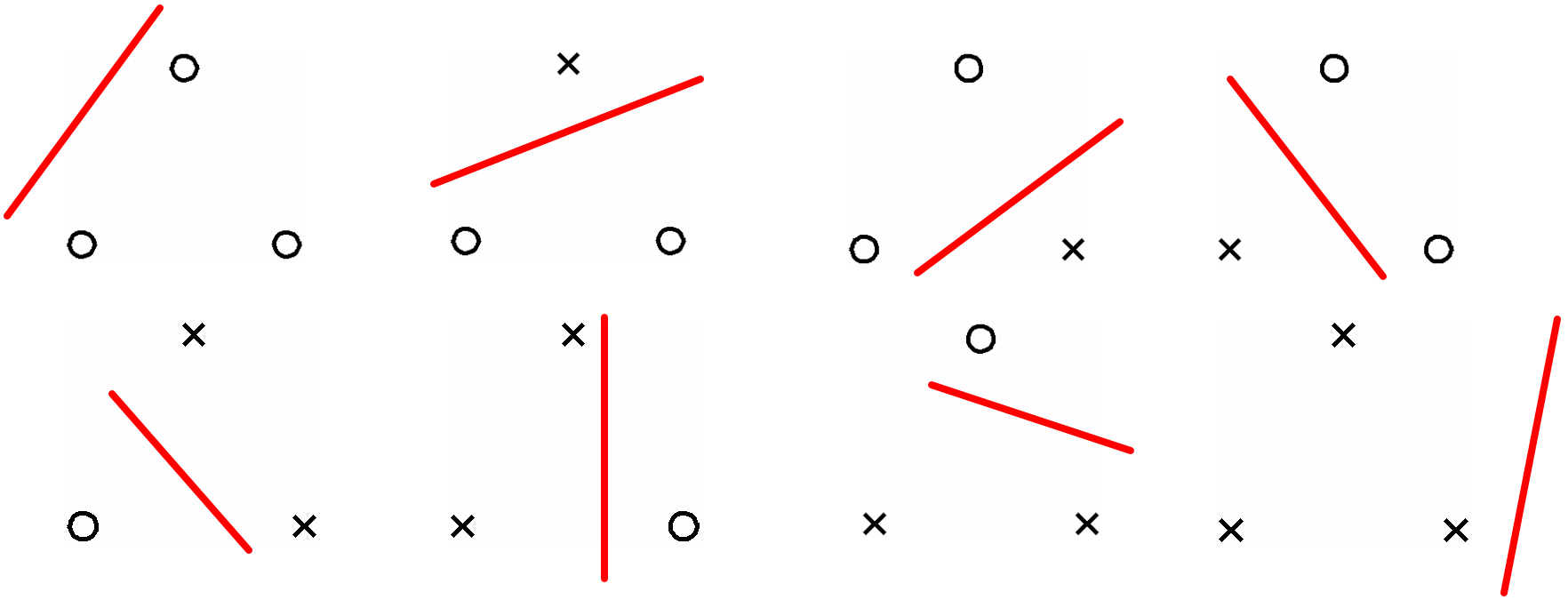
VC dimension

- Capacity measure for classification models
- Largest number of points that can be shattered by model
- Classifier *shatters* data points if for any labeling, points can be separated
- Not the same as number of parameters in model!

Shattering

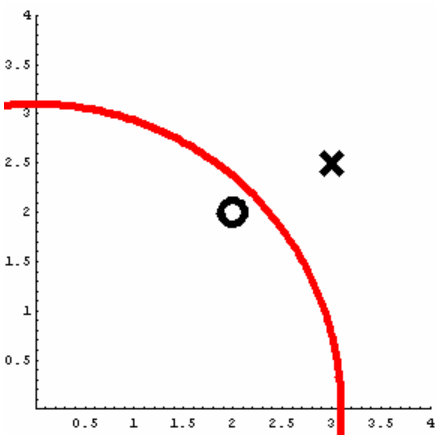
Straight lines can shatter 3 points in 2-space

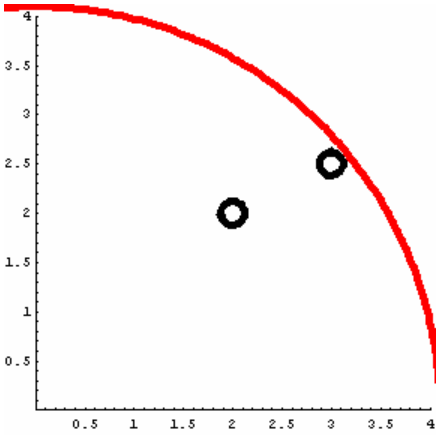
Model: sign($\alpha \cdot x$)



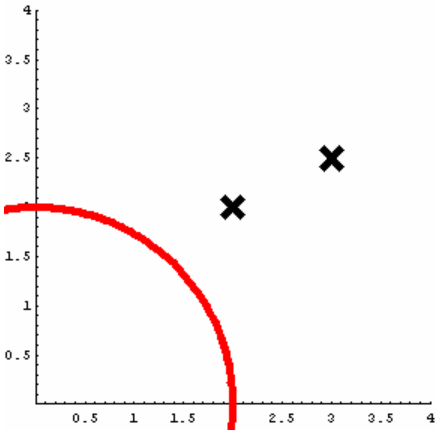
Shattering

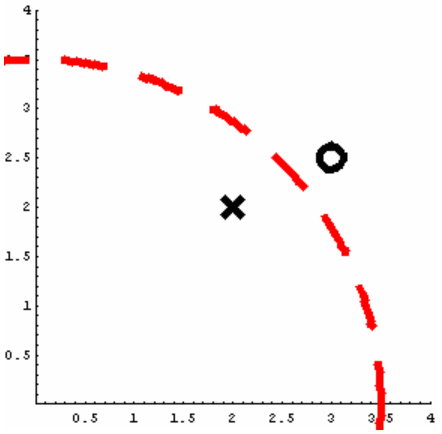
Other model:
sign($x \bullet x - \alpha$)





Still other model:
sign($\beta x \bullet x - \alpha$)





VC dimension

- Largest number of points for which there is arrangement that can be shattered
- For straight lines in 2-space, VC dim. is 3
- For hyperplanes in *n*-space, VC dim. is $n + 1$
- There is model with one parameter and infinite VC dimension

Structural risk minimization

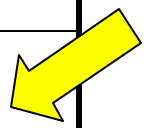
$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

- Fix data set and order classifiers according to their VC dimension
- For each classifier, train and calculate right-hand side
- Best classifier minimizes right-hand side

Structural risk minimization

$$R(\alpha) \le R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

Model	R_{emp}	VC conf.	Upper bound
$f_1(\alpha)$			
$f_2(\alpha)$			
$f_3(\alpha)$			
$f_4(\alpha)$			
$f_5(\alpha)$			



best

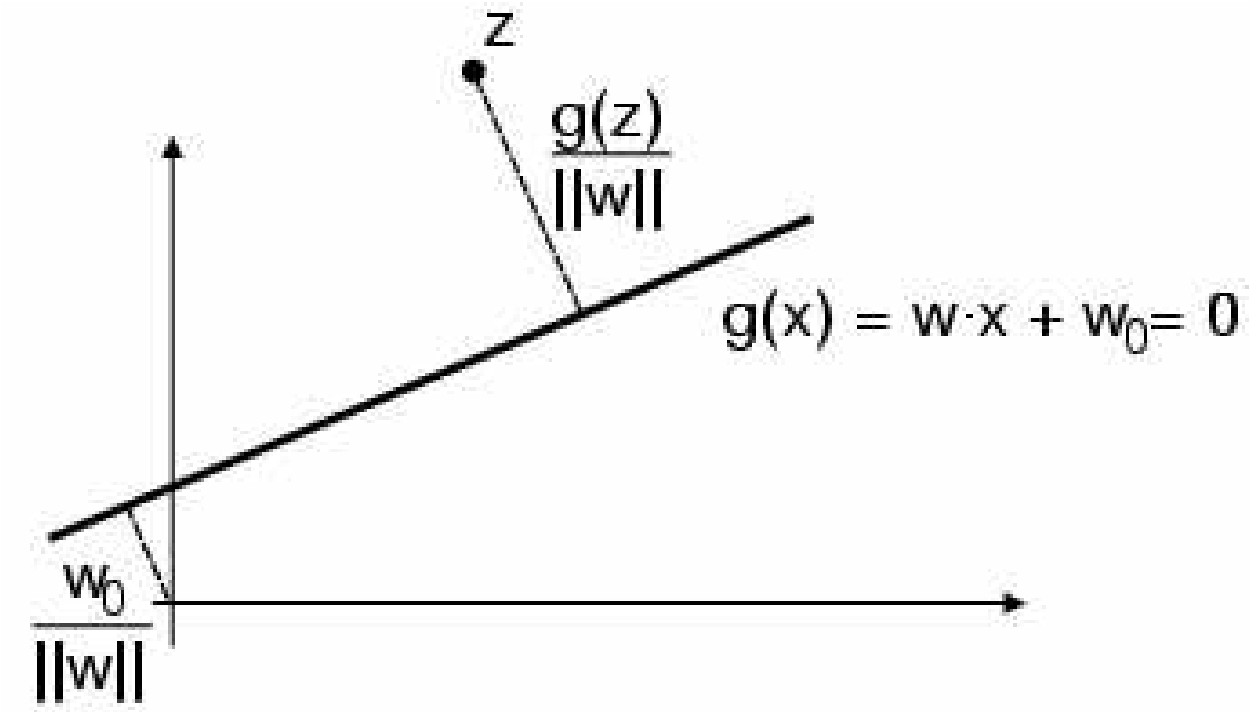
Model selection

- Cross-validation: use test sets to estimate error
- Penalize model complexity:
  - Akaike information criterion (AIC)
  - Bayesian information criterion
- Structural risk minimization

Support Vector Machines

- Implement hyperplanes, so know VC-dimension
- Algorithmic representation of concepts from statistical learning theory
- Determine hyperplanes that maximize *margin* between classes

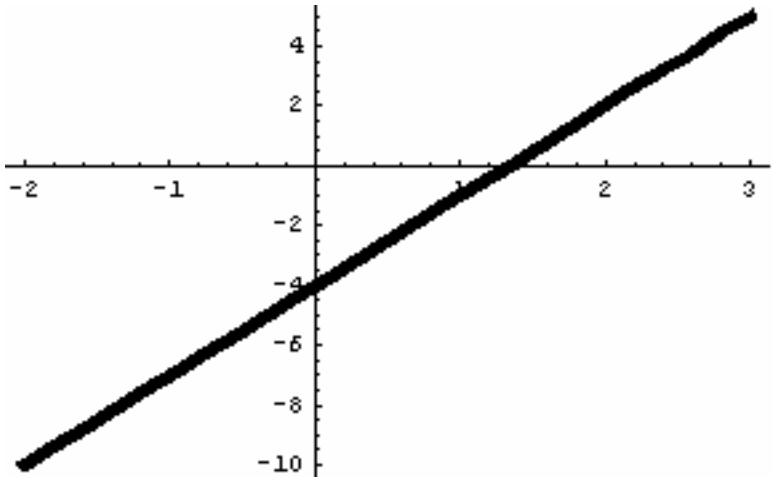
Geometry of hyperplanes



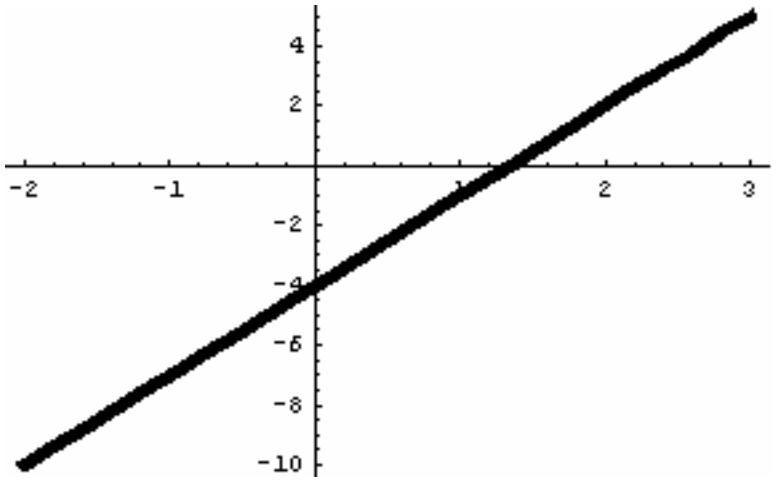
Geometry of hyperplanes

Hyperplanes invariant to scaling of parameters:

$$\{x|w \bullet x + w_0 = 0\} = \{x|c\, w \bullet x + c\, w_0 = 0\}$$

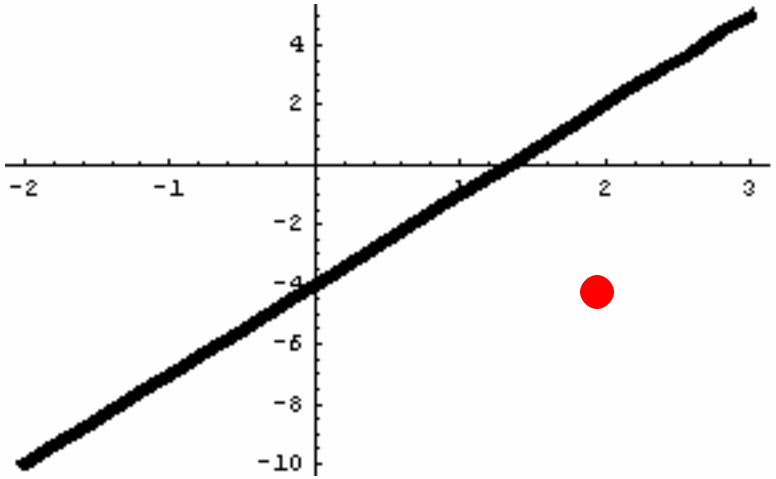


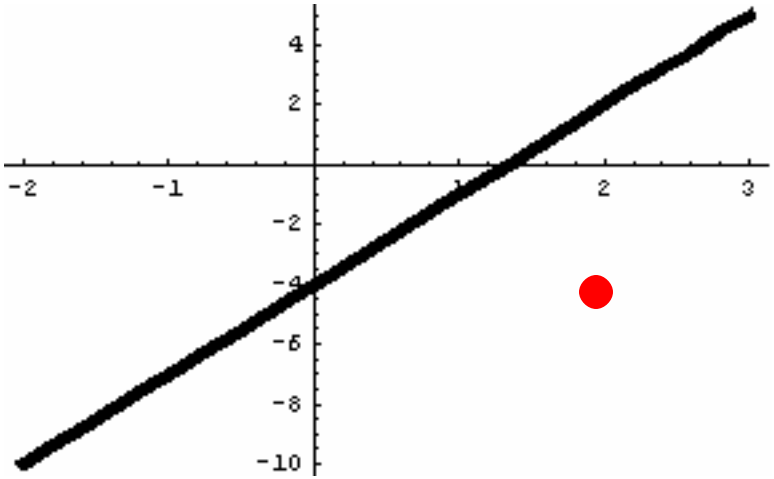
$3x - y - 4 = 0$



$6x - 2y - 8 = 0$

Geometry of hyperplanes





$3x - y - 4 = 0$

$6x - 2y - 8 = 0$

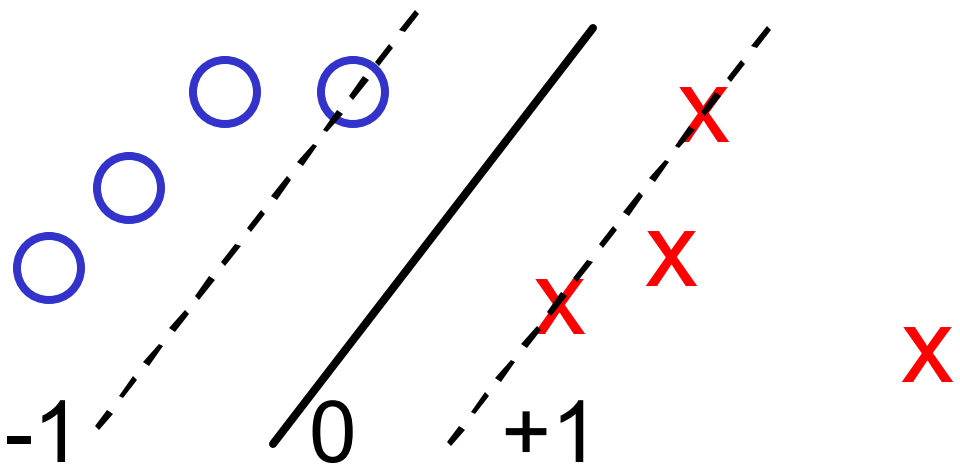
$3 * 2 - (-4) - 4 = 6$

$6 * 2 - 2*(-4) - 8 = 12$

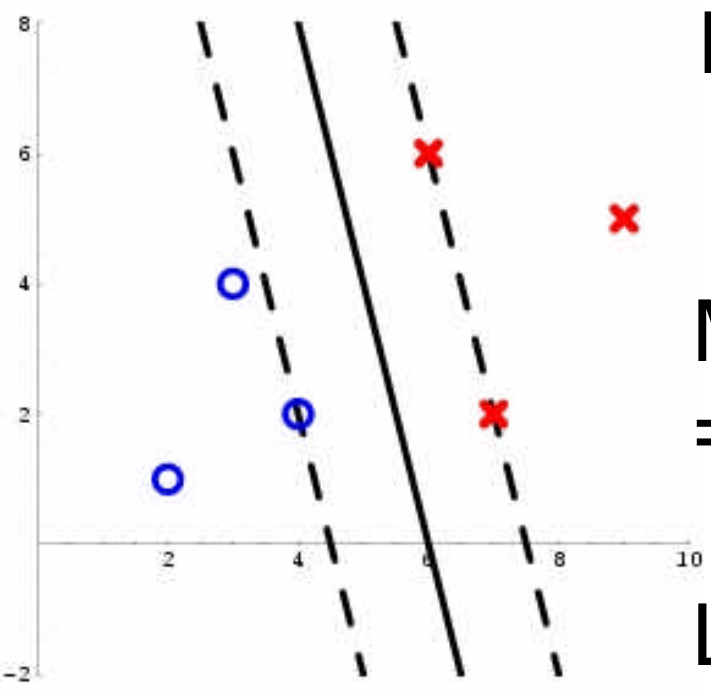
Optimal hyperplanes

We want

$w \bullet x_i + w_0 \geq 1$ for all x_i in class 1 (t_i =+1)
$w \bullet x_i + w_0 \leq -1$ for all x_i in class 2 (t =-1)



Optimal hyperplanes



Points on dashed lines satisfy
$g(x) = +1$ resp. $g(o) = -1$

Margin $= |g(x)|/||w|| + |g(o)|/||w|| = 2/||w||$

Largest (optimal) margin:
maximize $2/||w||$ equiv. to
minimize $||w||^2$
subject to $t_i(w \bullet x_i + w_0) - 1 \geq 0$

Optimal hyperplanes

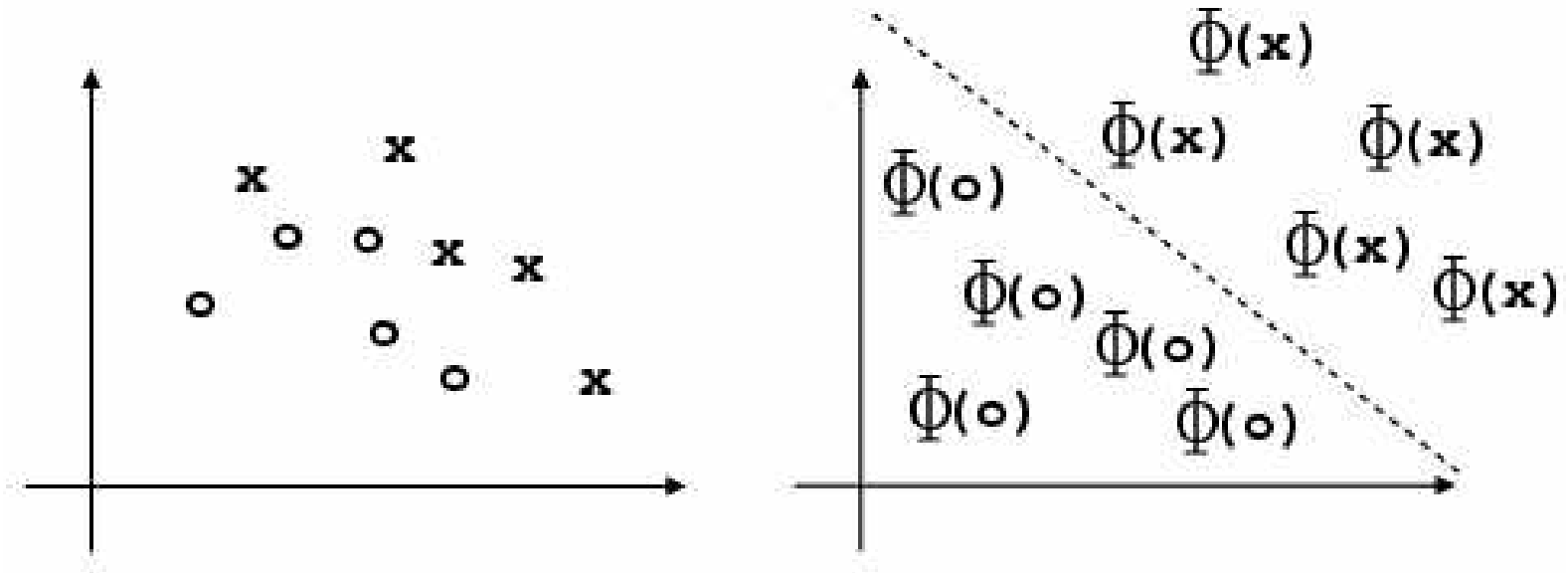
- Optimal hyperplane has largest margin ("large margin classifiers")
- Parameter estimation problem turned into constrained optimization problem
- Unique solution $w = \sum \alpha_i x_i$ over all inputs x_i on the margin ("support vectors")
- Decision function $g(x) = \text{sign}(\sum \alpha_i x_i \cdot x + w_0)$
- All other cases x_j irrelevant to solution!

Nonseparable data sets

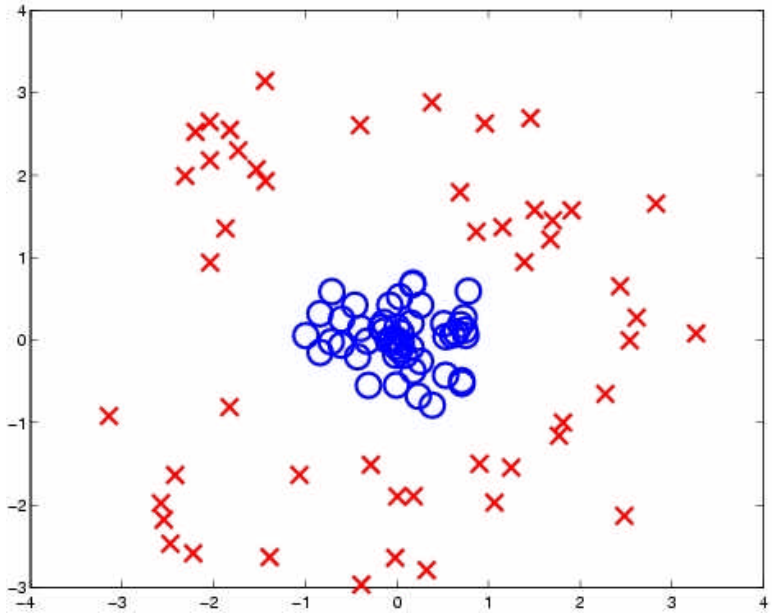
- Introduce slack variables $\xi_i \geq 0$
- Constraints are then
  $w \bullet x_i + w_0 \geq +1 - \xi_i$ for all x_i in class 1
  $w \bullet x_i + w_0 \leq -1 + \xi_i$ for all x_i in class 2
- minimize $||w||^2 + C\Sigma\xi_i$

Nonlinear SVM

- Idea: Nonlinearly project data into higher dimensional space with Φ: $R^m \rightarrow H$
- Apply optimal hyperplane algorithm in *H*



Nonlinear SVM example



Idea: Project $R^2 \rightarrow R^3$ via

$$\Phi(x_1, x_2) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$

Nonlinear SVM example

Do the math:

$$\Phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \Phi\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \cdot \begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix}$$

$$= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2$$

$$= (x_1y_1 + x_2y_2)^2$$

$$= \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)^2$$

Nonlinear SVM

- Recall: Input data x_i enters calculation only via dot products $x_i \cdot x_j$ or $\Phi(x_i) \cdot \Phi(x_j)$
- Kernel trick:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

- Advantage: no need to calculate Φ
- Advantage: no need to know H
- What are admissible kernel functions?

Kernel functions

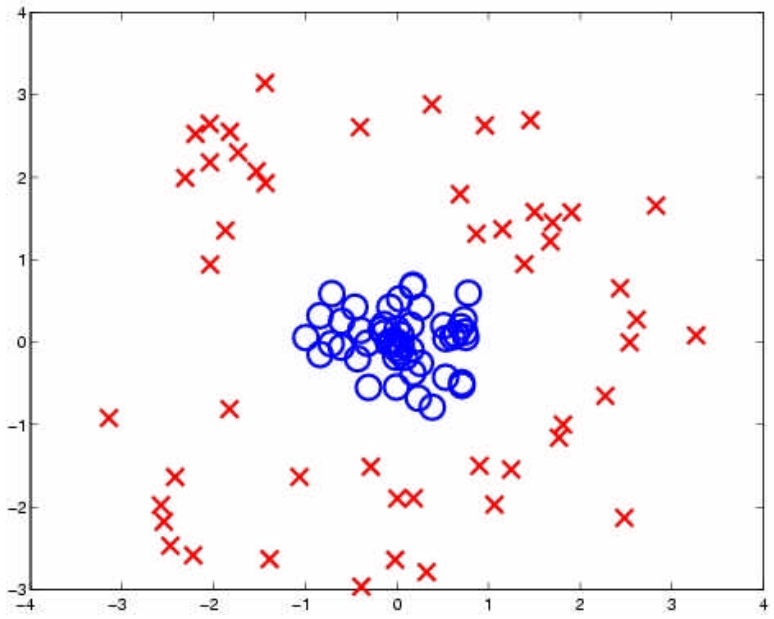
- Satisfy *Mercer's condition*
- Most widely used (+parameters):
  - Polynomials (degree)
  - Gaussians (variance)
- For given kernel function *K*, projection Φ and projection space *H* not unique

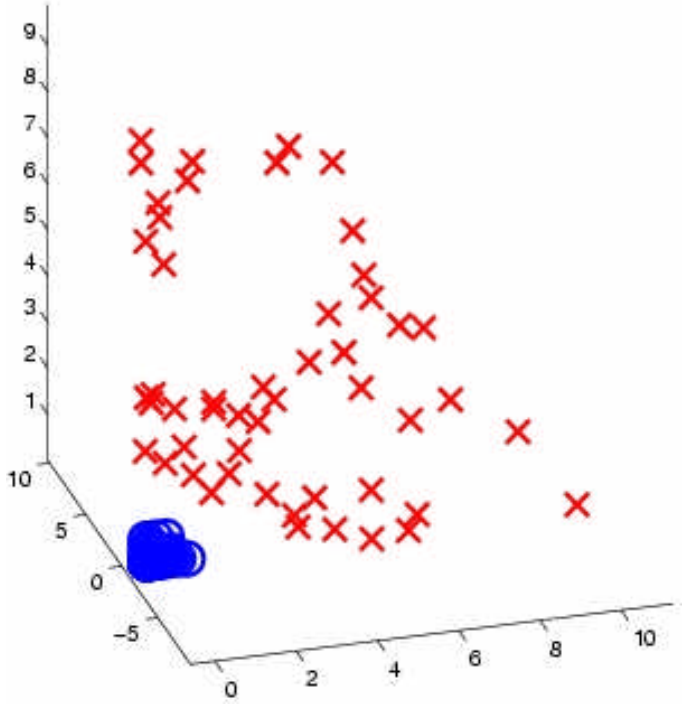
Kernel function example

- Data space R^2; $x, y \in R^2$
- $K(x,y) = (x \cdot y)^2$
- Possible $H=R^3$
- Possible $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$

Kernel function example

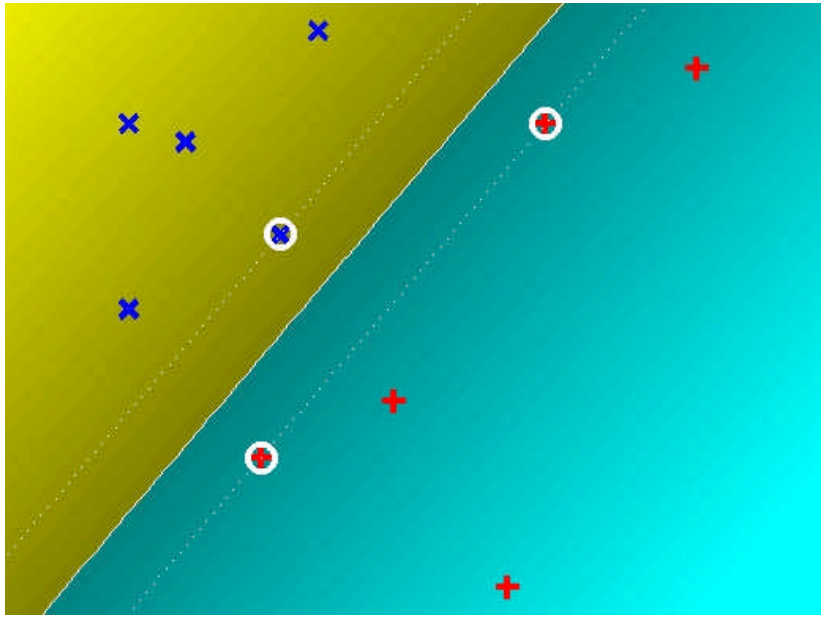
$K(x,y) = (x \cdot y)^2$



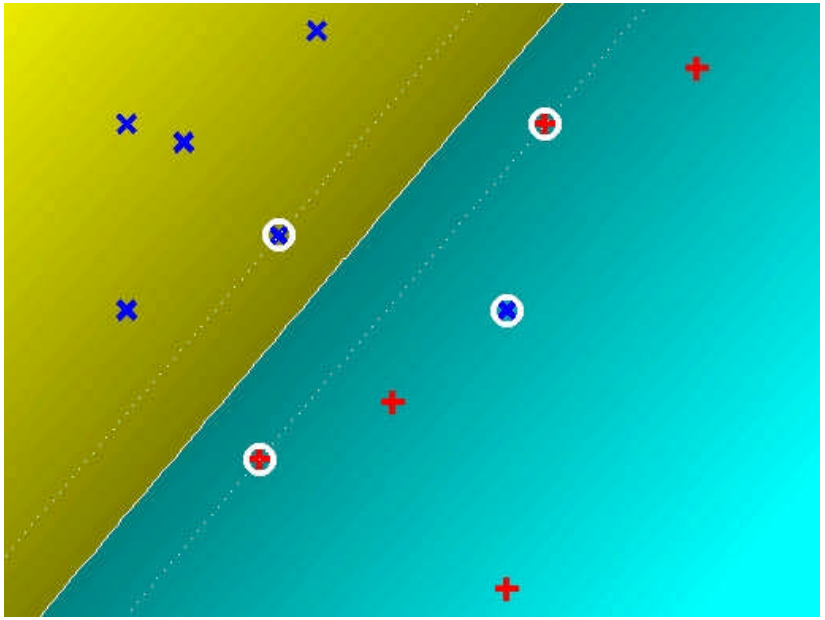


SVM examples

Linearly separable

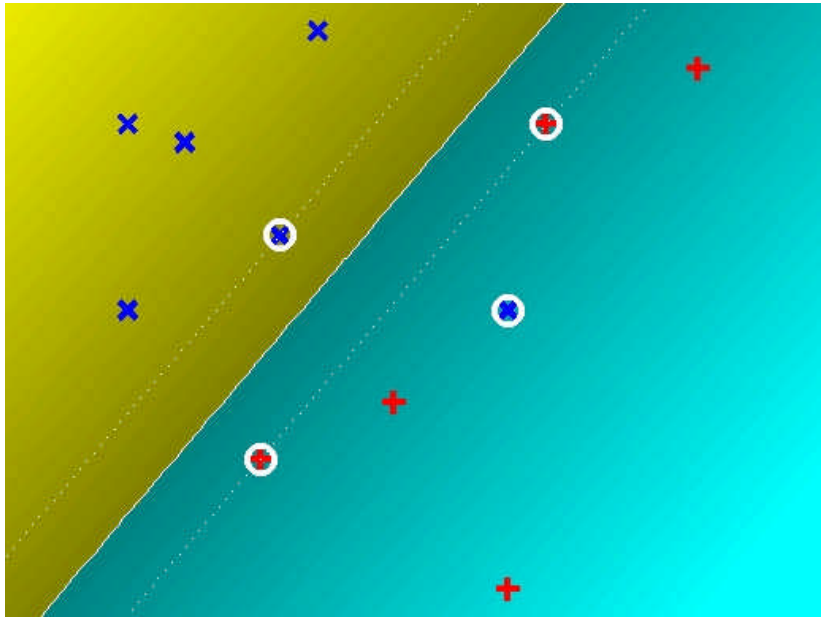


C=100

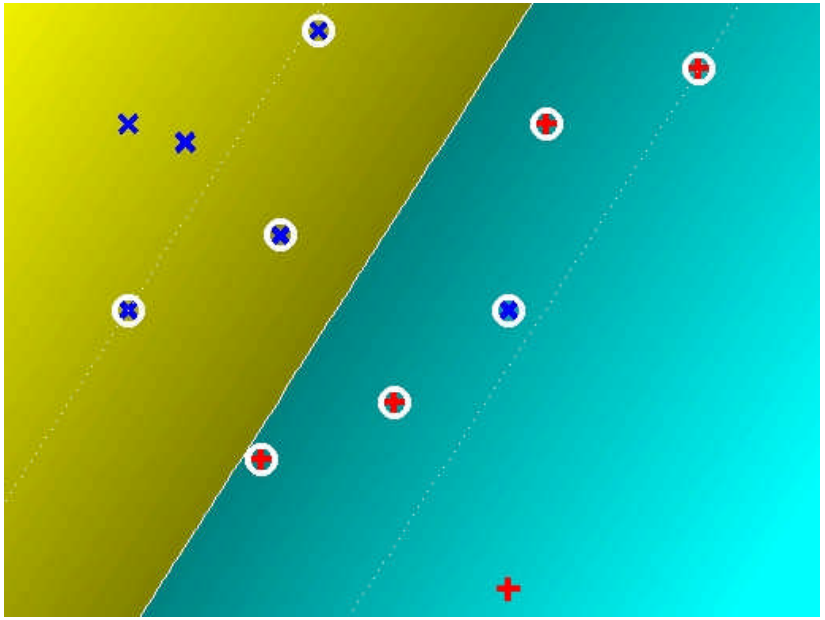


SVM examples

C=100

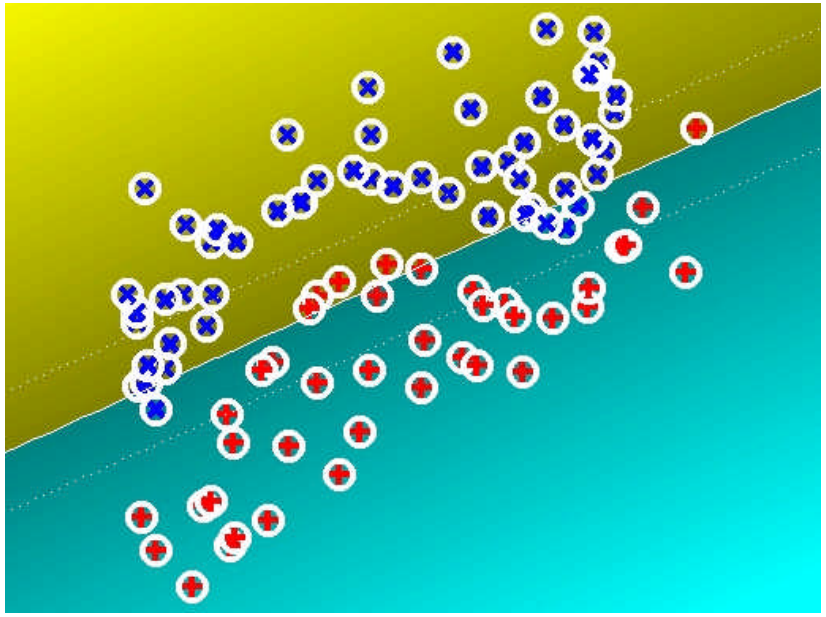


C=1

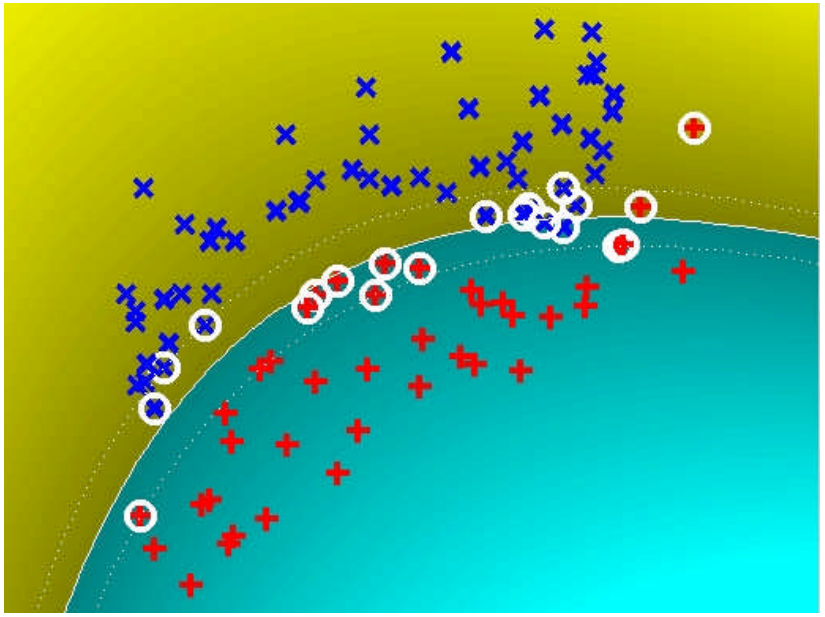


SVM examples

Linear function

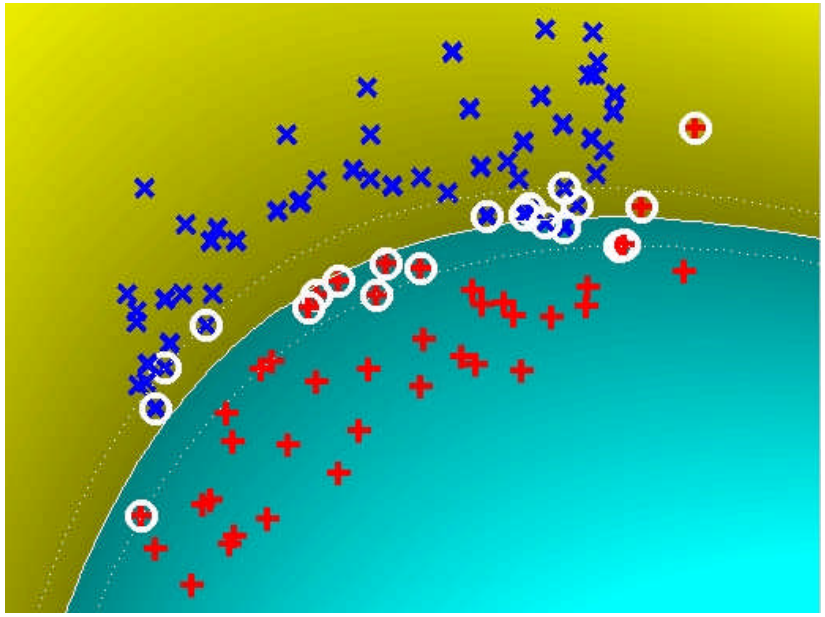


Quad. polynomial

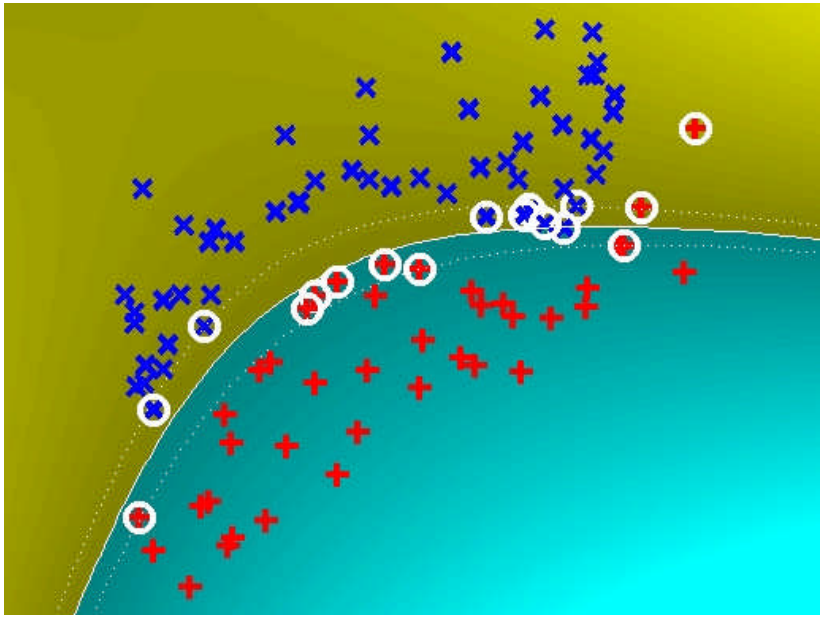


SVM examples

Quad. poly., C=10

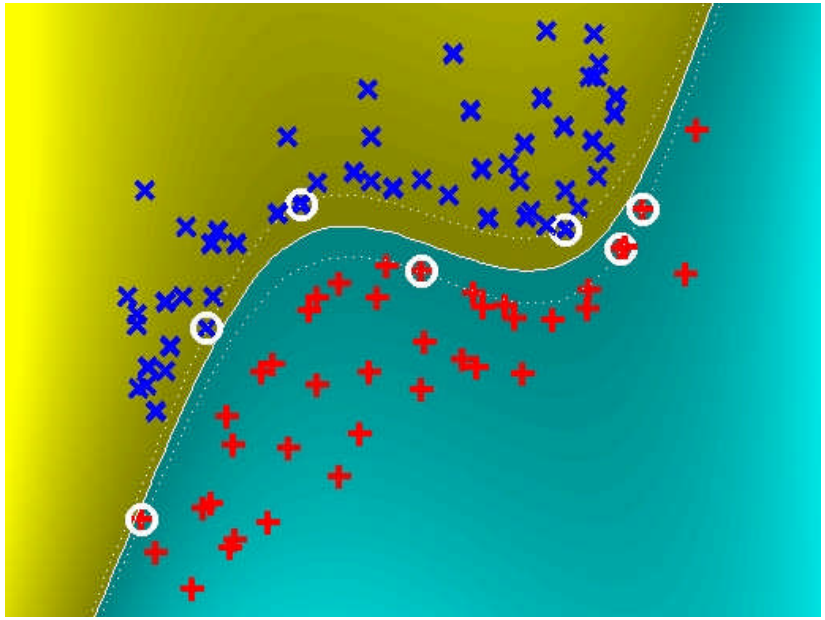


Quad. poly., C=100

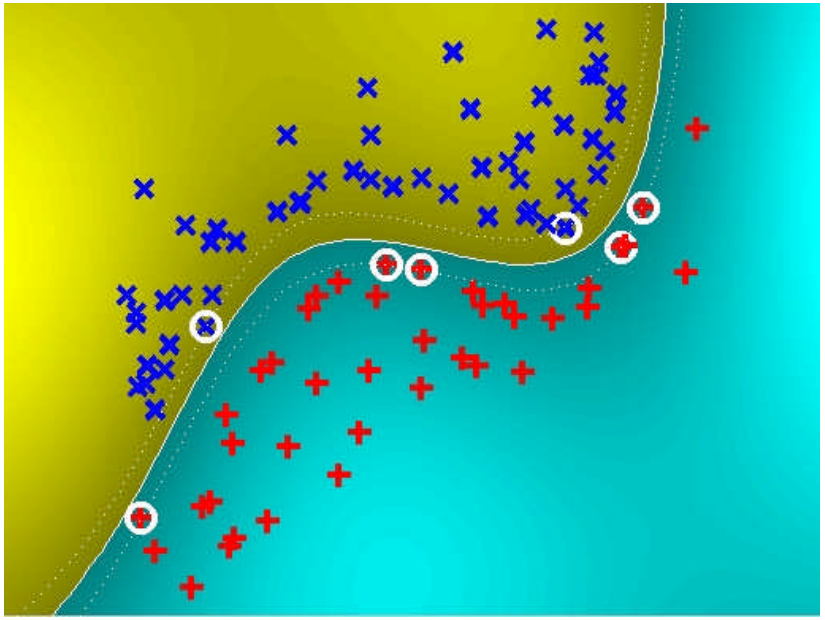


SVM examples

Cubic polynomial

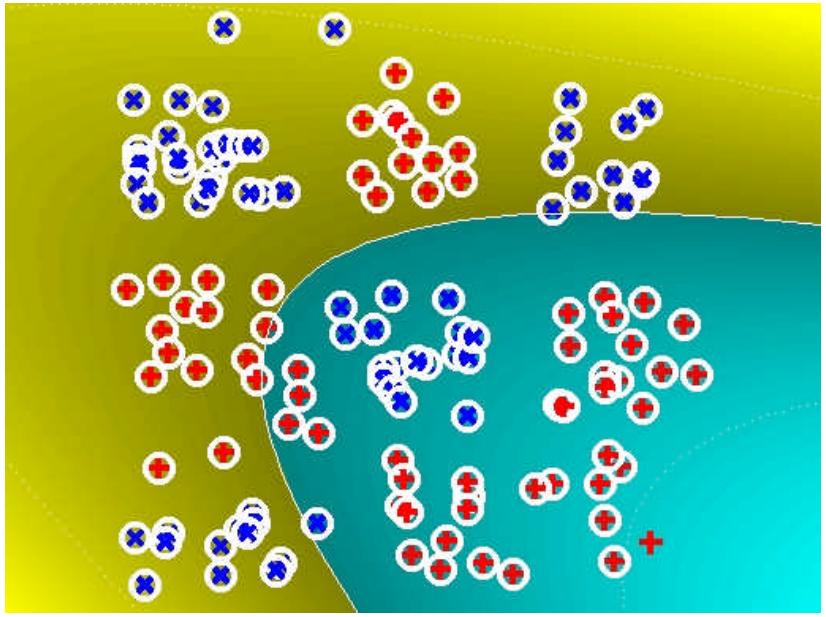


Gaussian, $\sigma = 1$

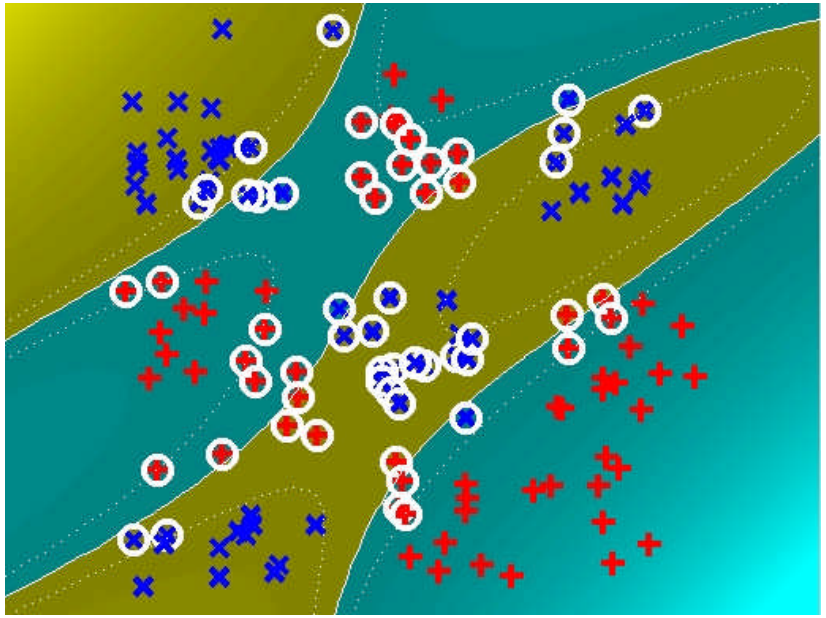


SVM examples

Quad. polynomial

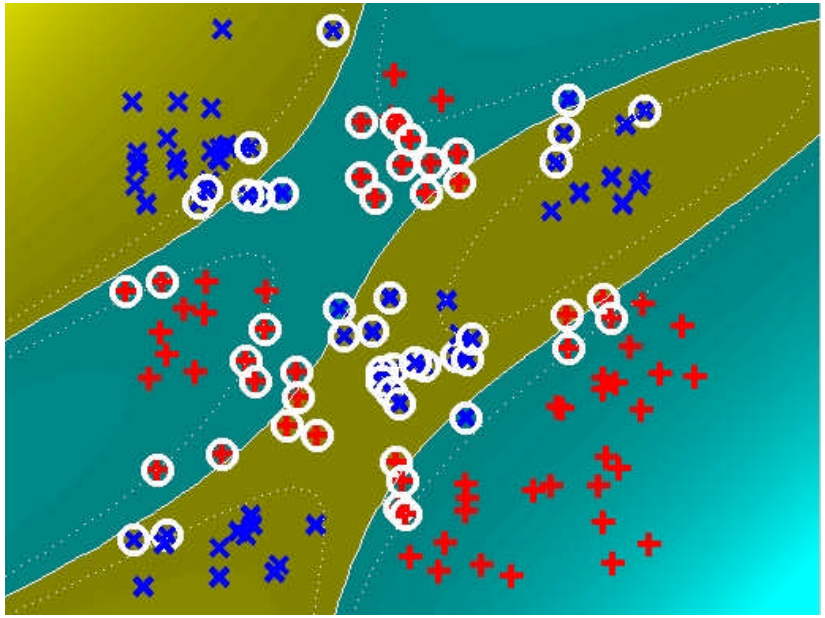


Cubic polynomial

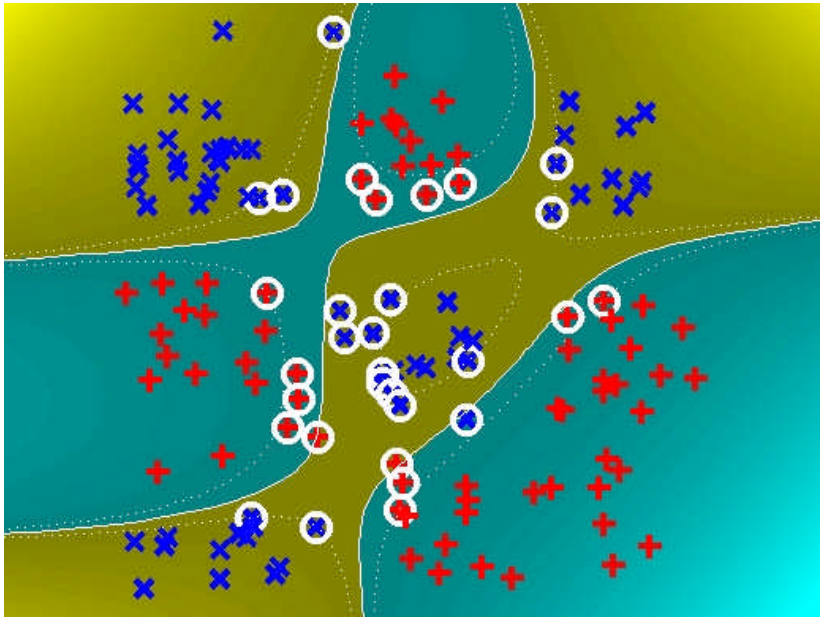


SVM examples

Cubic polynomial

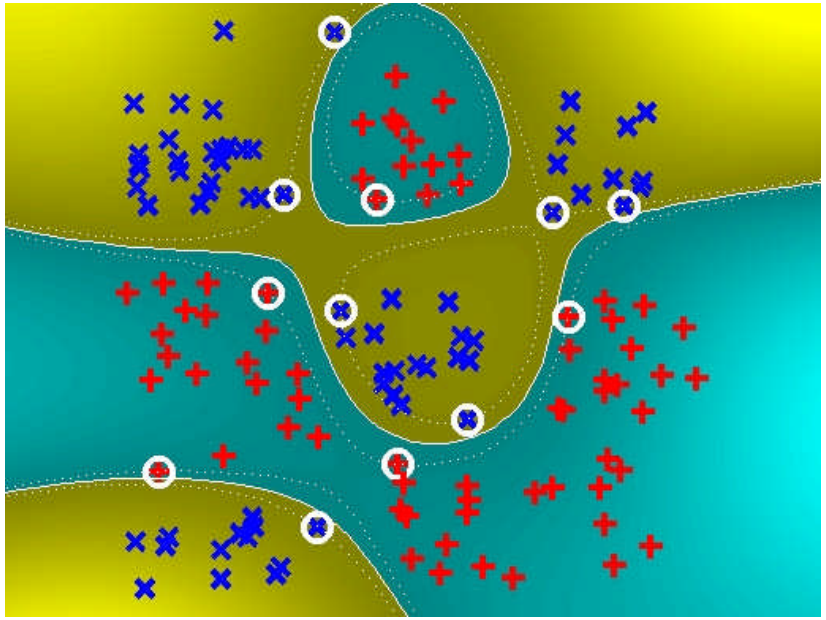


Degree 4 poly.

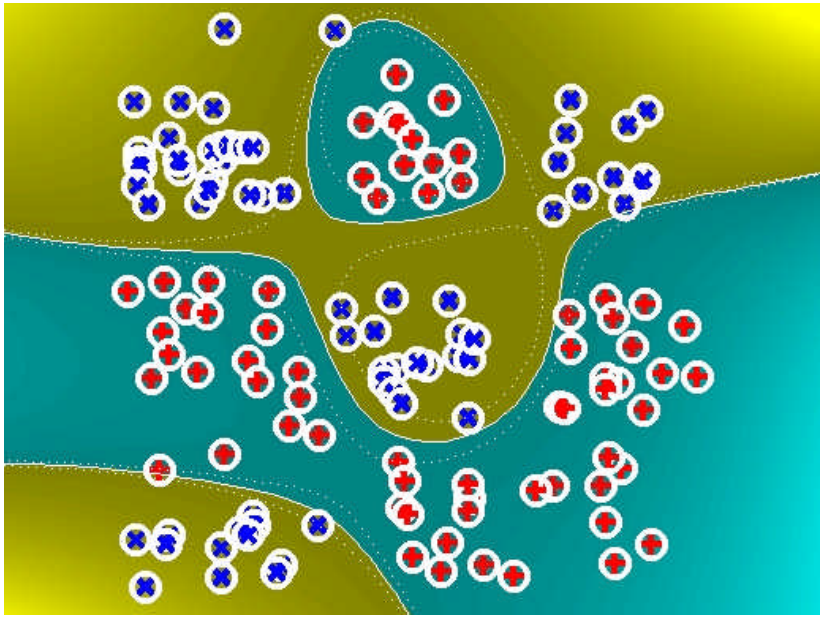


SVM examples

Gaussian, $\sigma = 1$



Gaussian, $\sigma = 3$



Performance comparison

- Log. regression $\Leftrightarrow$ ANN $\Leftrightarrow$ SVM
- Real-world data set
- 1619 lesion images
- 107 morphometric features:
  - Global (size, shape)
    - size
    - shape
  - Local (color distributions)
- Use ROC analysis

CN $\Leftrightarrow$ DN + MM

- Logistic regression: 0.829
- Artificial neural networks: 0.826
- Support vector machines (polynomial kernel): 0.738 to 0.813
- Support vector machines (Gaussian kernel): 0.786 to 0.831

MM $\Leftrightarrow$ CN + DN

- Logistic regression: 0.968
- Artificial neural networks: 0.968
- Support vector machines (polynomial kernel): 0.854 to 0.918
- Support vector machines (Gaussian kernel): 0.947 to 0.970

Summary

- SVM based on statistical learning theory
- Bounds on generalization performance
- Optimal separating hyperplanes
- Kernel trick (projection)
- Performs comparable to log. regression and neural networks

Pointers to the literature

- Burges C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery. 1998; 2(2):121-167.
- Christianini N, Shawe-Taylor J. An introduction to support vector machines. Cambridge University Press 2000.
- Vapnik V. Statistical learning theory. Wiley Interscience 1998.