CAP 5610: Machine Learning

Lecture 6:
Support Vector Machines I

Instructor: Dr. Gita Sukthankar
Email: gitars@eecs.ucf.edu

Reference Materials From: GuoJun Qi, Ray Mooney, Henry Kautz

Reading

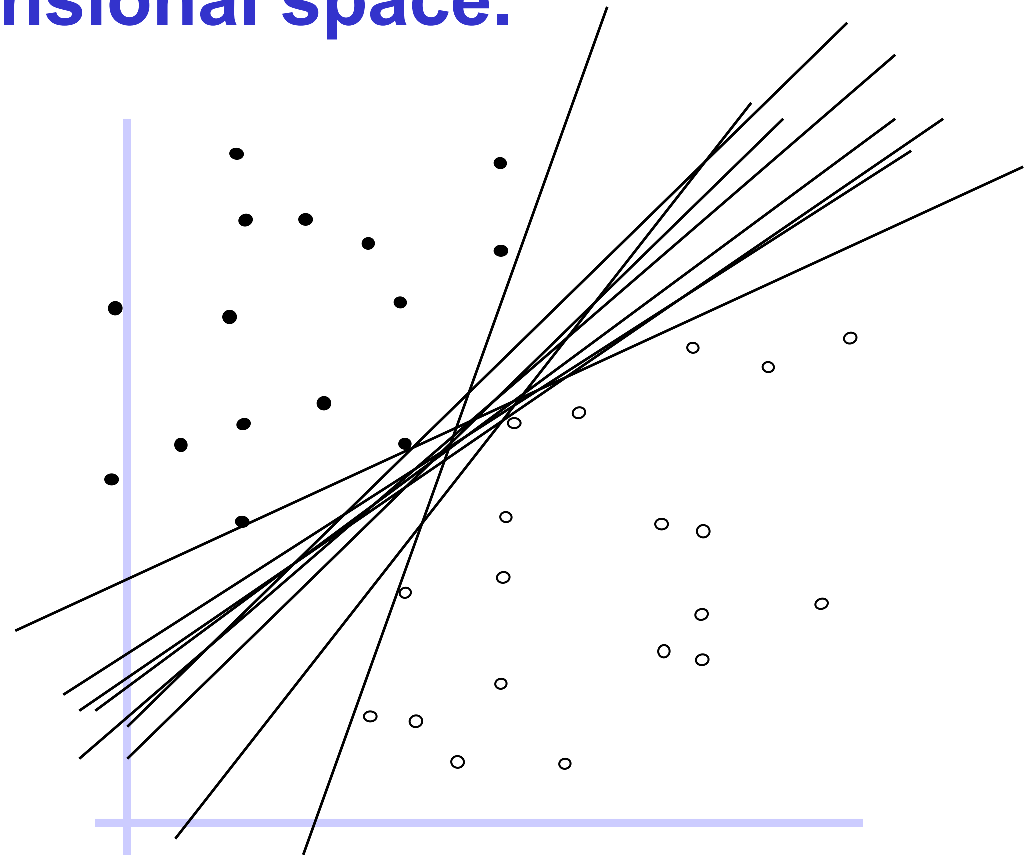
- ML: Probabilistic Perspective (Murphy): Chapter 14
- PRML (Bishop): Chapters 6 and 7
- ML: Algorithmic Perspective (Marsland): Chapter 8

Linear Classifier

- Naive Bayes
  - Assume each attribute is drawn from Gaussian distribution with the same variance
  - Generative model: estimate the mean and variance with closed-form solution
- Logistic regression
  - Directly maximizing the log likelihood to fit the model into the training data
  - Discriminative model: no closed-form solution, a gradient ascent method is used.

Drawback

- Lacking of a geometric intuition to explain what's a good linear classifier in high dimensional space.



SVM vs. logistic regression

- Logistic Regression optimizes the log likelihood function, with probabilities modeled by the sigmoid function.
  - Tends to be more sensitive to outliers than SVM
- SVM extends by using kernel tricks, transforming datasets into rich feature space, so that complex problems can be still dealt with in the same “linear” fashion in the lifted hyper space.
  - More sophisticated optimization procedure
  - Use of hinge loss vs. log loss
- Depending on the number of features vs. the number of datapoints it may be better to use one or the other

G. Drakos, Towards Data Science

SVM

- Supervised learning methods used for
  - Classification
  - Regression
- A special property: simultaneously
  - minimize the classification error
  - maximize the geometric margin
  - maximum margin classifier
- Excellent theory and good performance
- Dominant method in machine learning for many years

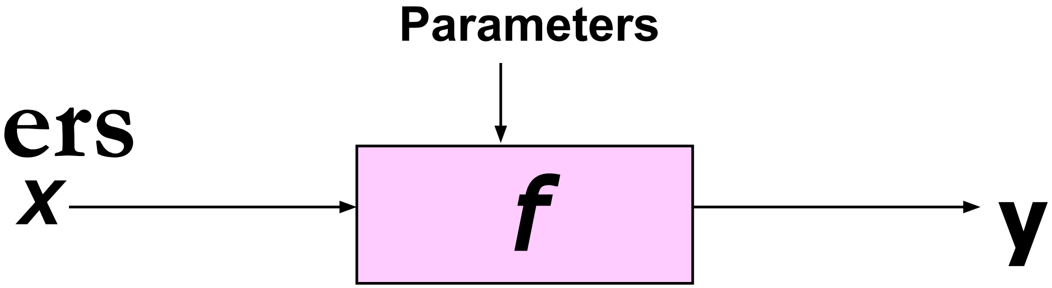
Outline

- Linear SVM – hard margin
- Linear SVM – soft margin
- Non-linear SVM
- Application

Outline

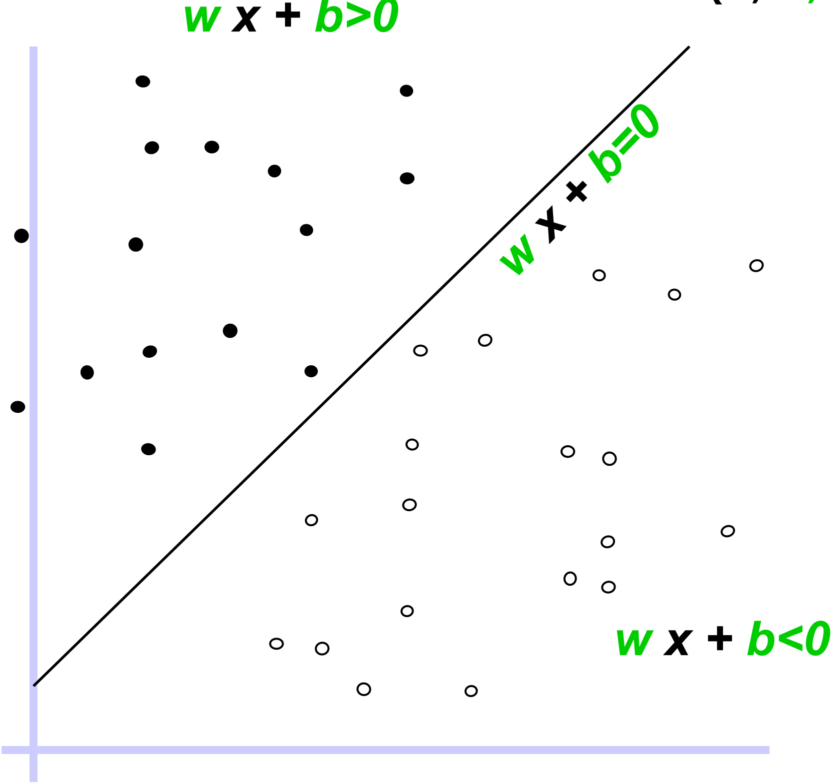
- Linear SVM – hard margin
- Linear SVM – soft margin
- Non-linear SVM
- Application

Linear Classifiers



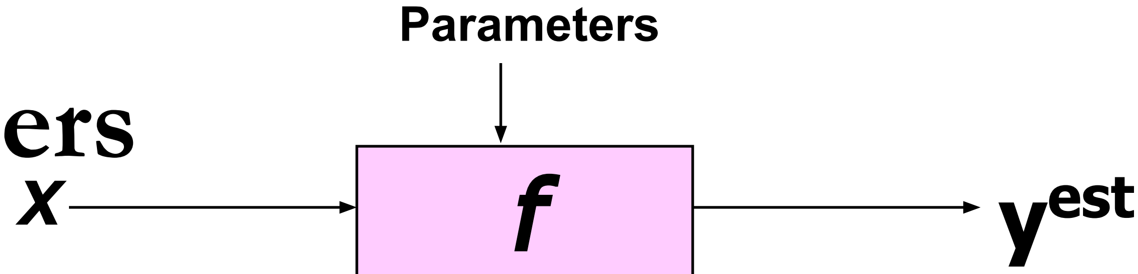
Label *y*:
- **denotes +1**
- **denotes -1**

$f(x,w,b) = sign(w\ x + b)$



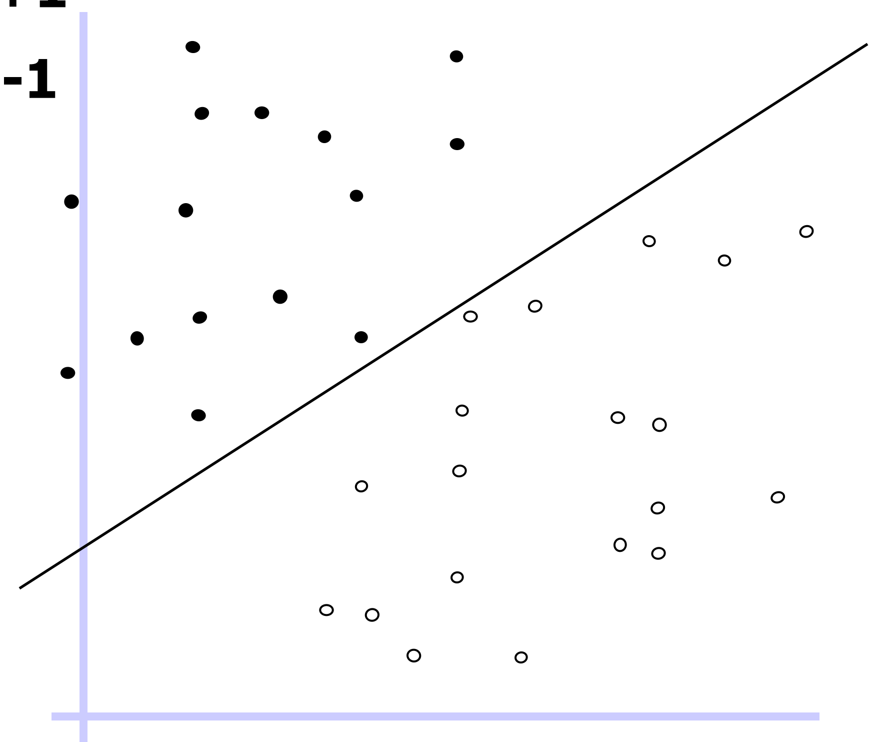
How would you classify this data?

Linear Classifiers



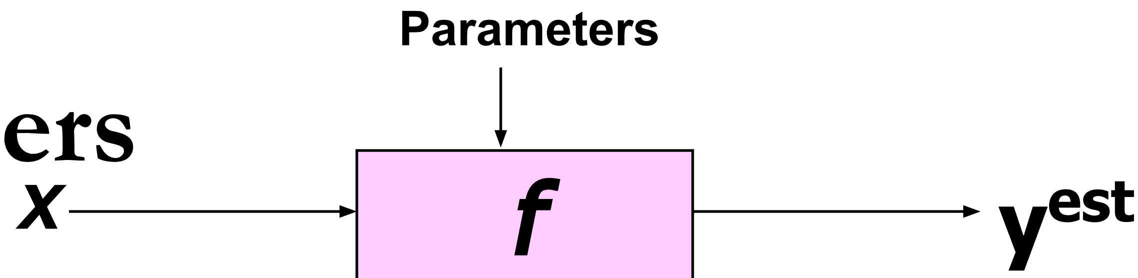
$$f(x,w,b) = sign(w\ x + b)$$

- denotes +1
- ∘ denotes -1



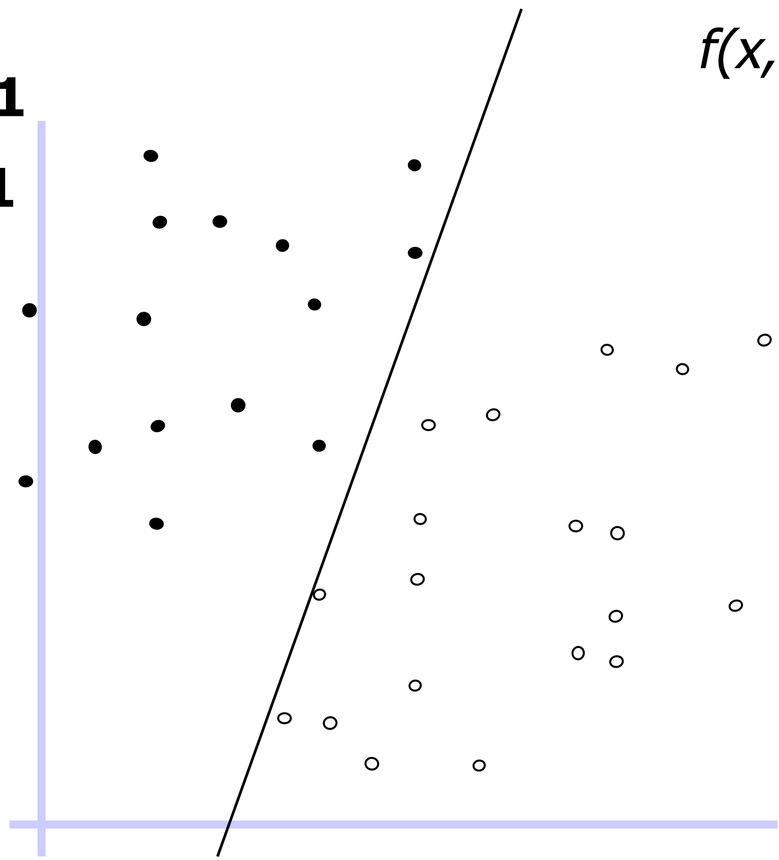
How would you classify this data?

Linear Classifiers



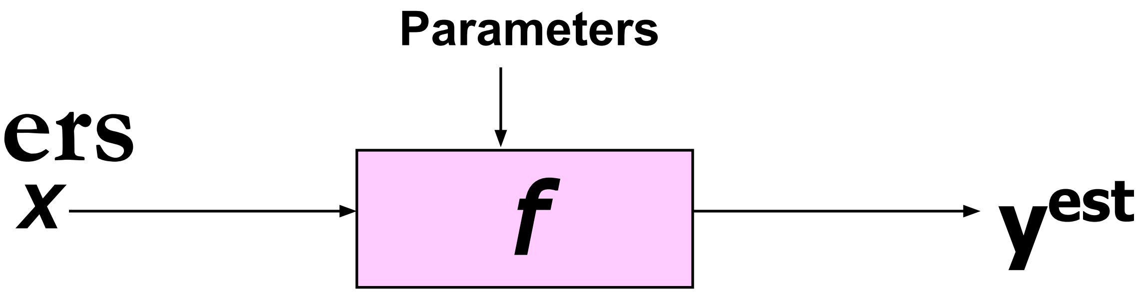
$f(x,w,b) = sign(w\ x + b)$

- • denotes +1
- ◦ denotes -1



How would you classify this data?

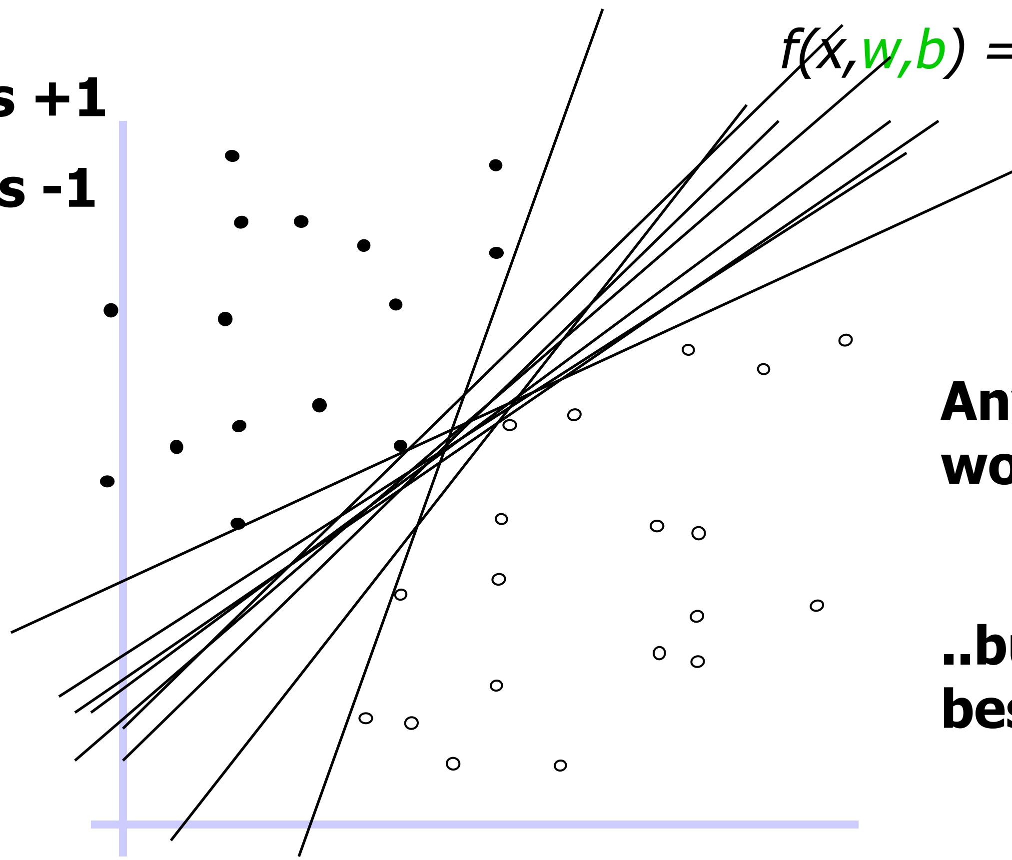
Linear Classifiers



$$f(x,w,b) = sign(w\ x + b)$$

• denotes +1

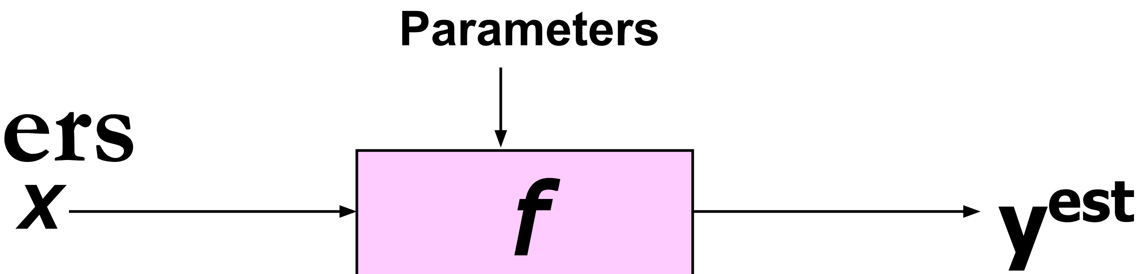
∘ denotes -1



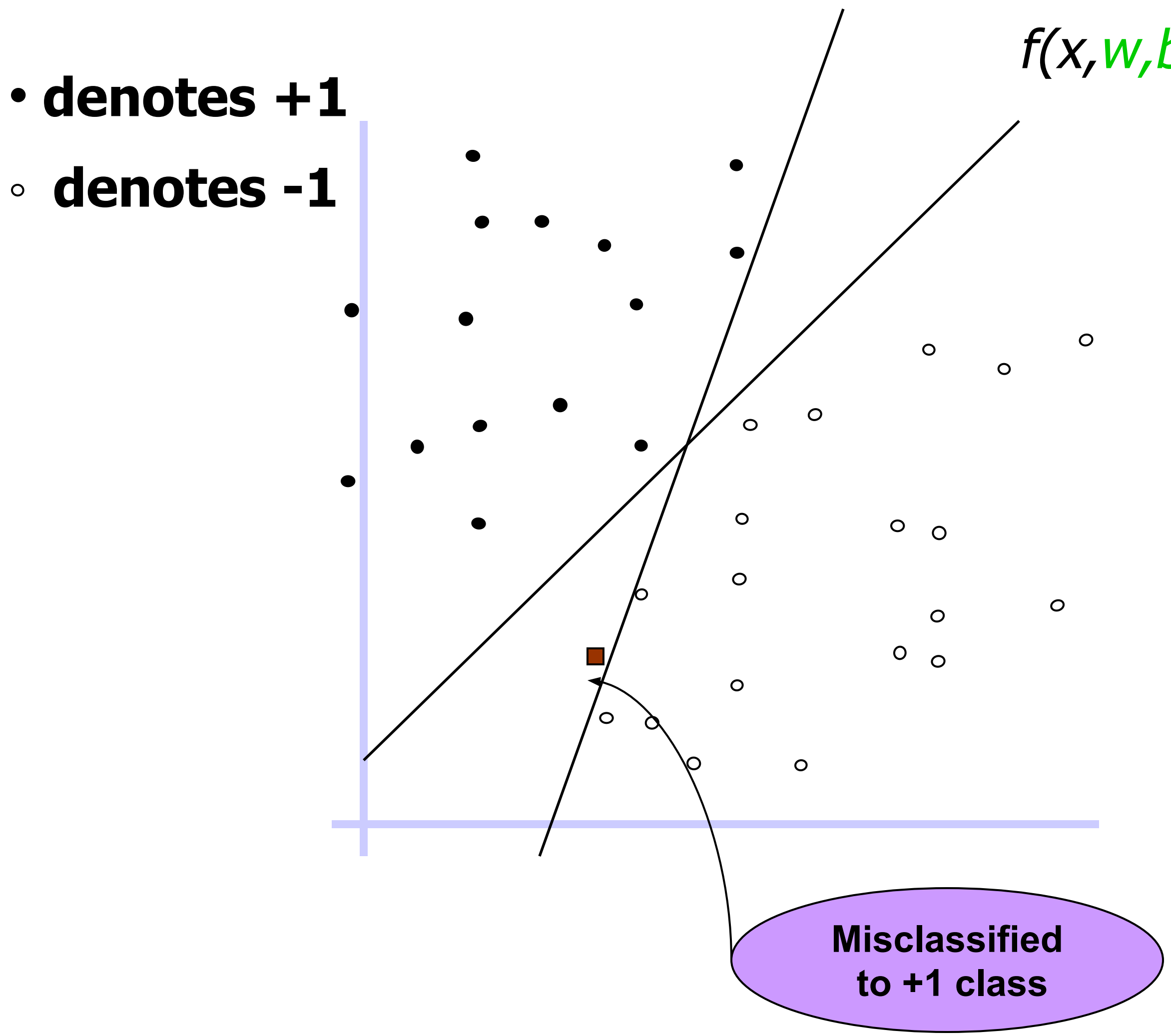
Any of these would be fine..

..but which is best?

Linear Classifiers

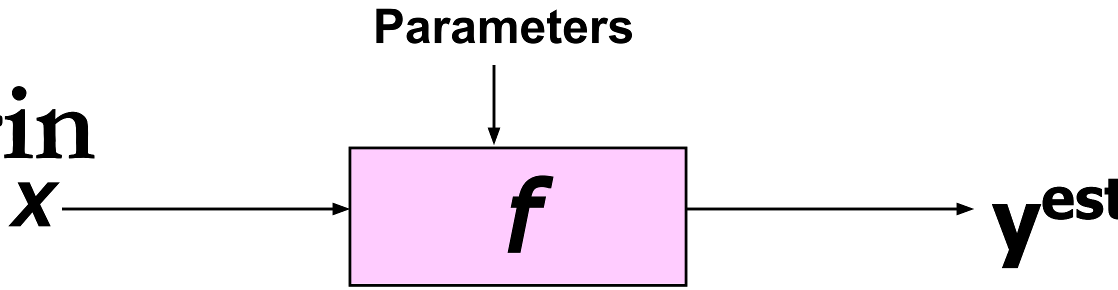


$f(x, w, b) = sign(w\ x + b)$

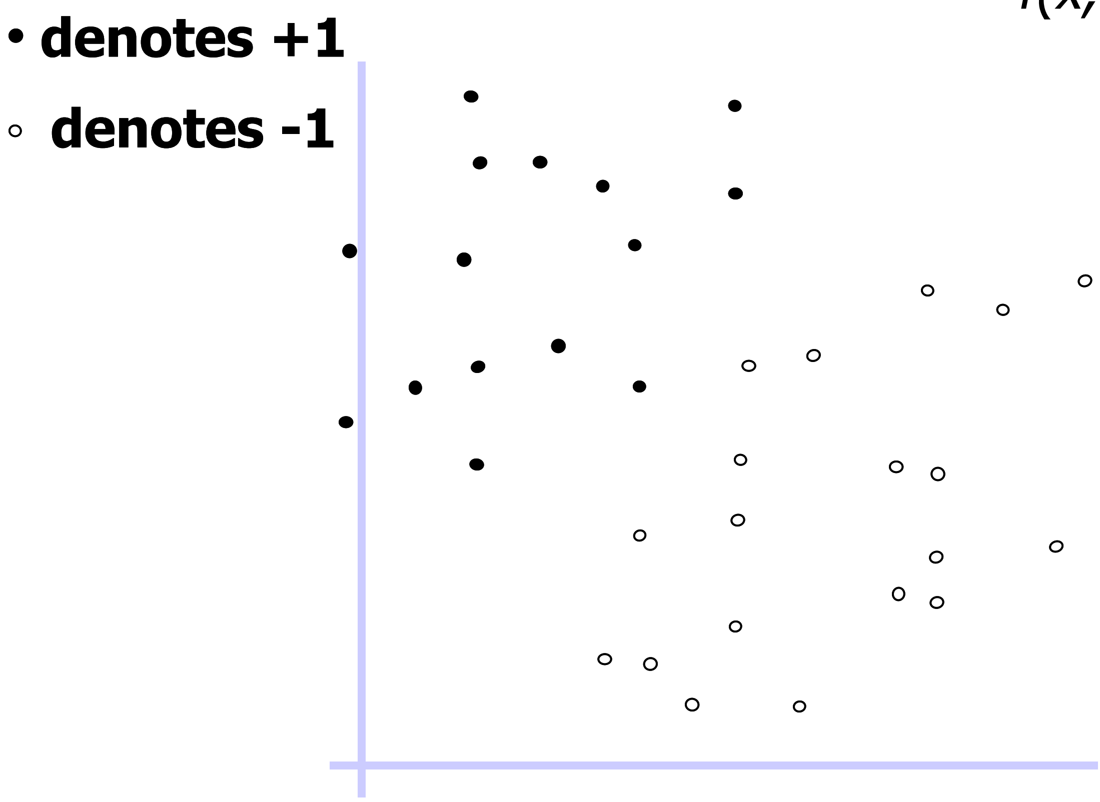


How would you classify this data?

Classifier Margin

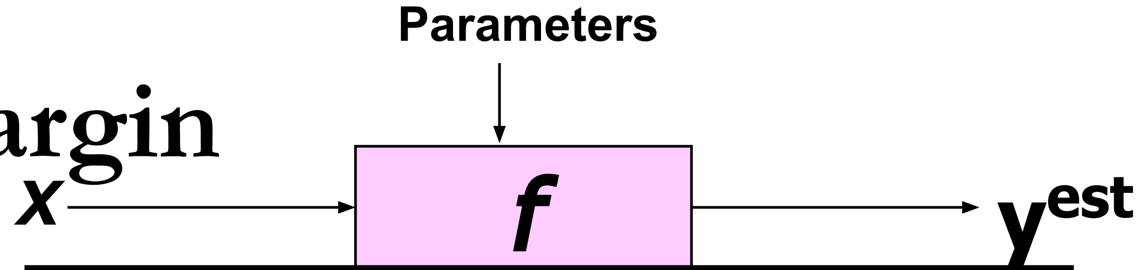


$f(x,w,b) = sign(w\ x + b)$



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a data point.

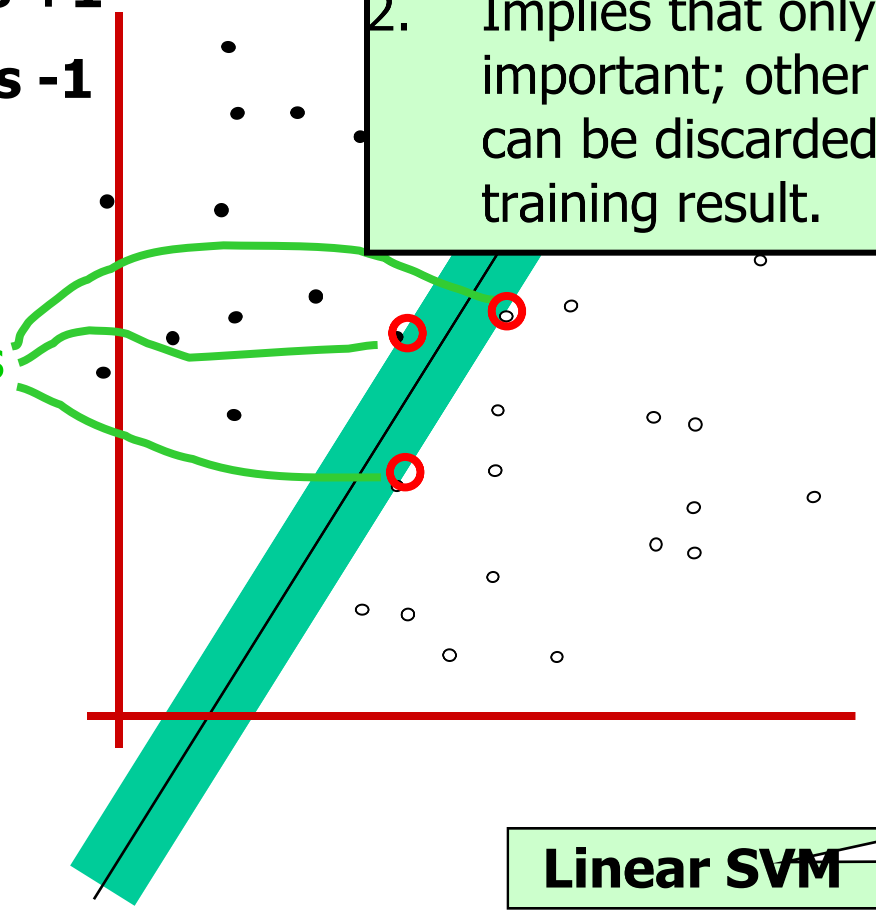
Maximum Margin



1. Maximizing the margin makes sense according to intuition
2. Implies that only support vectors are important; other training examples can be discarded without affecting the training result.

• denotes +1

∘ denotes -1

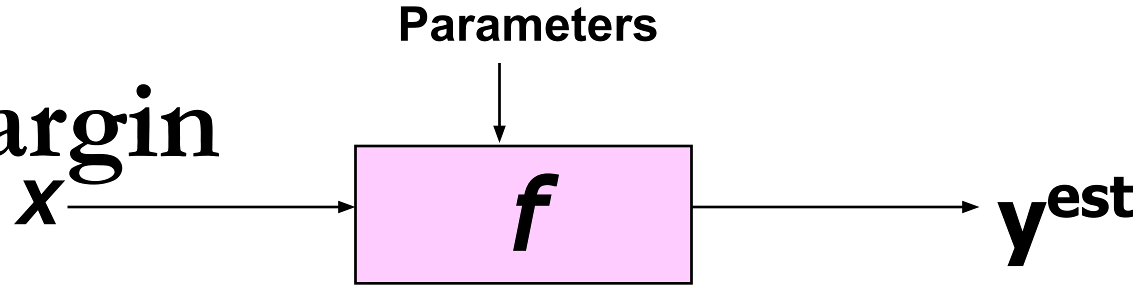


Support Vectors are those datapoints that the margin pushes up against

linear classifier with the maximum margin.

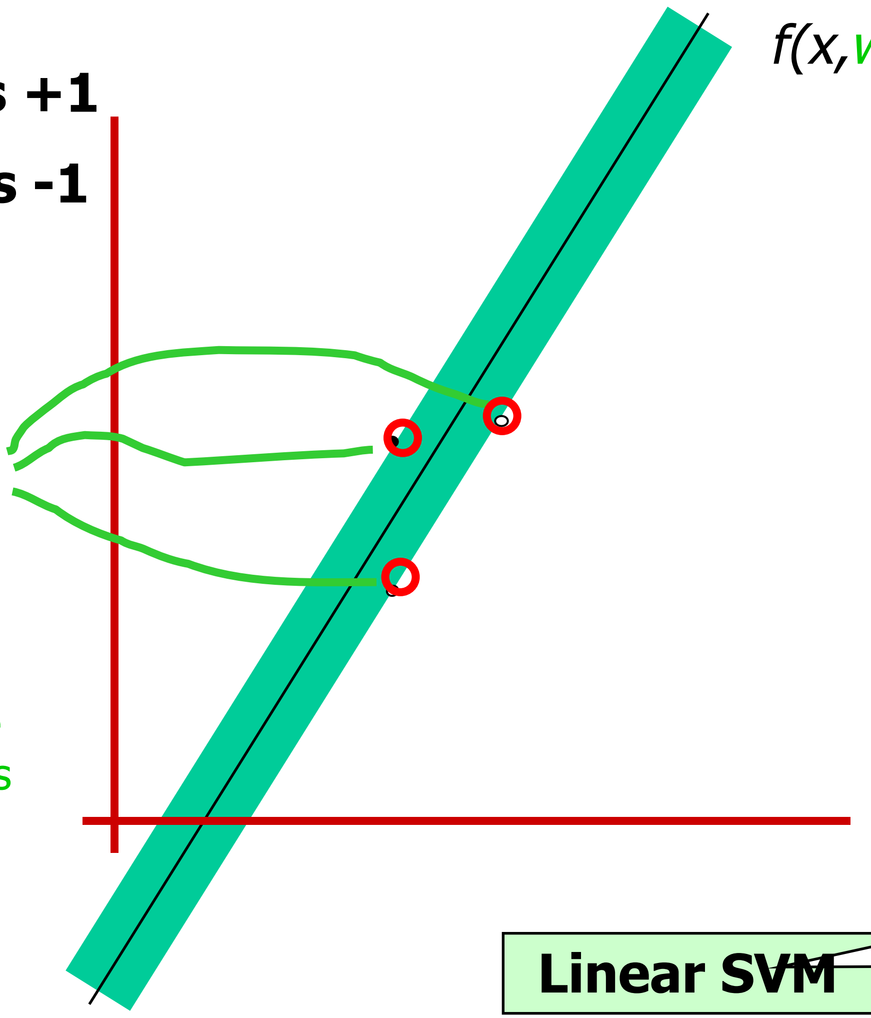
This is the simplest kind of SVM (Called an LSVM)

Maximum Margin



$f(x,w,b) = sign(w\ x + b)$

- • denotes +1
- ° denotes -1



Keeping only support vectors will not change the maximum margin classifier.

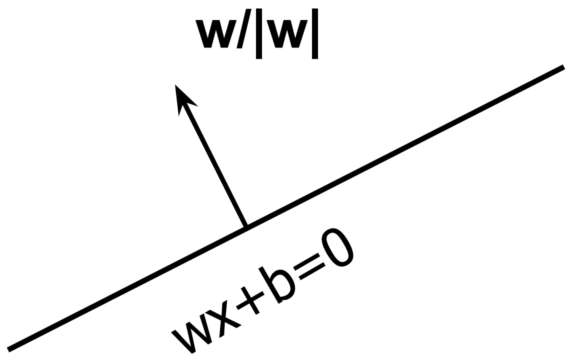
- ❑ Robust to the small changes (noises) in non-support vectors

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Basics

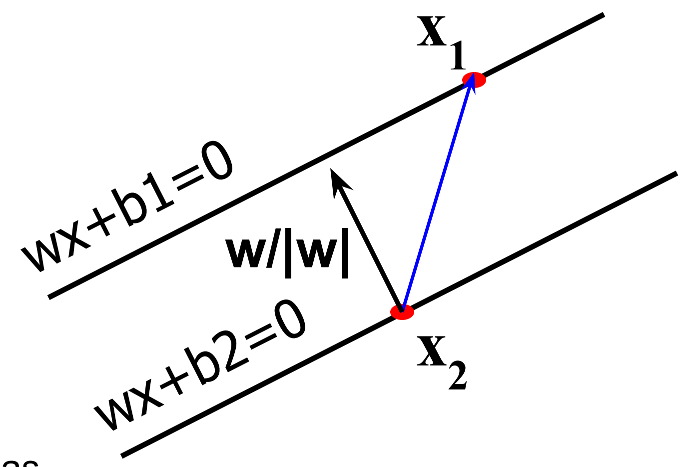
- **w/|w|:**
  - **Perpendicular to line wx+b=0**
  - **Unit length**



- **Margin of two parallel lines is**

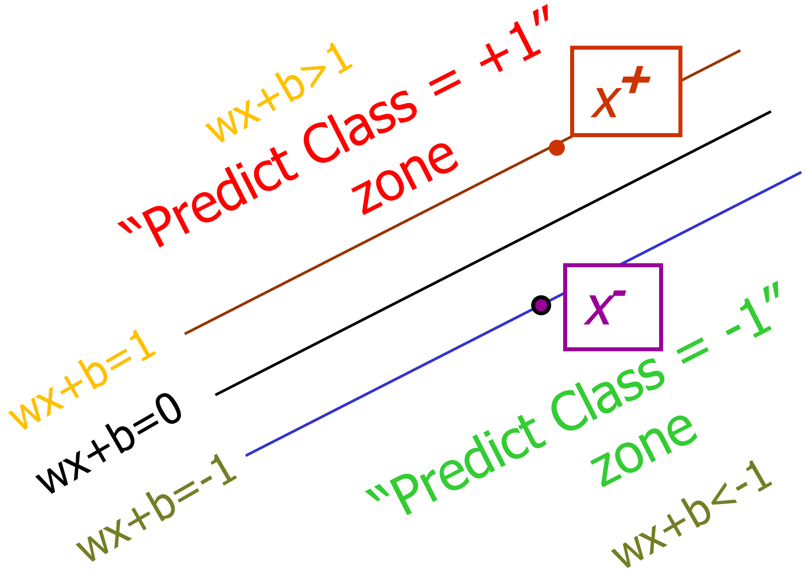
$$\frac{w \mid x_1 - x_2 \mid}{\mid w \mid} = \frac{\mid b_1 - b_2 \mid}{\mid w \mid}$$

$$wx_1 + b_1 = 0 \qquad w(x_1 - x_2) = b_2 - b_1$$
$$wx_2 + b_2 = 0$$



Intuition: you are projecting x onto w and shifting that output by a bias.
The isocontour (wx+b=0) will be perpendicular to w.

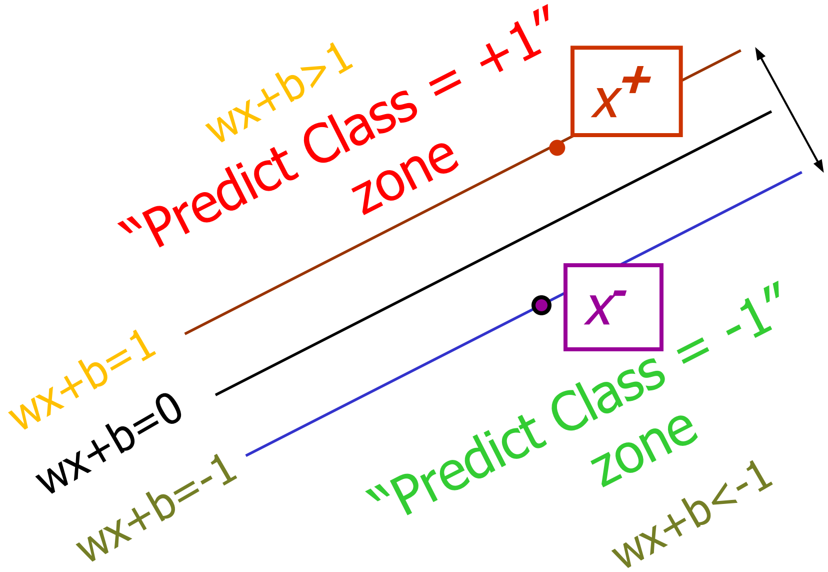
Linear SVM Mathematically



Decision rule:

- Positive examples: $w \,.\, x^+ + b > +1$
- Negative examples: $w \,.\, x^- + b < -1$
- Subtracting two equations: $w \,.\, (x^+ - x^-) = 2$

Linear SVM Mathematically



M=Margin Width

What we know:

- $w \,.\, x^+ + b = +1$
- $w \,.\, x^- + b = -1$
- $w \,.\, (x^+ - x^-) = 2$

$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

Linear SVM Mathematically

- **Goal: 1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \textit{\textbf{if } } y_i = +1$$

$$wx_i + b \leq -1 \quad \textit{\textbf{if } } y_i = -1$$

$$y_i(wx_i + b) \geq 1 \quad \textbf{for all i}$$

2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- **Minimize** $\Phi(w) = \frac{1}{2} w^t w$

  subject to $y_i(wx_i + b) \geq 1 \qquad \forall i$

Solving the Optimization Problem

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Use *Lagrange multiplier* α_i is associated with every constraint : $y_i(wx_i + b) \geq 1$, dual problem

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

Refer: Christopher J. C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

The Optimization Problem Solution

- The solution has the form:

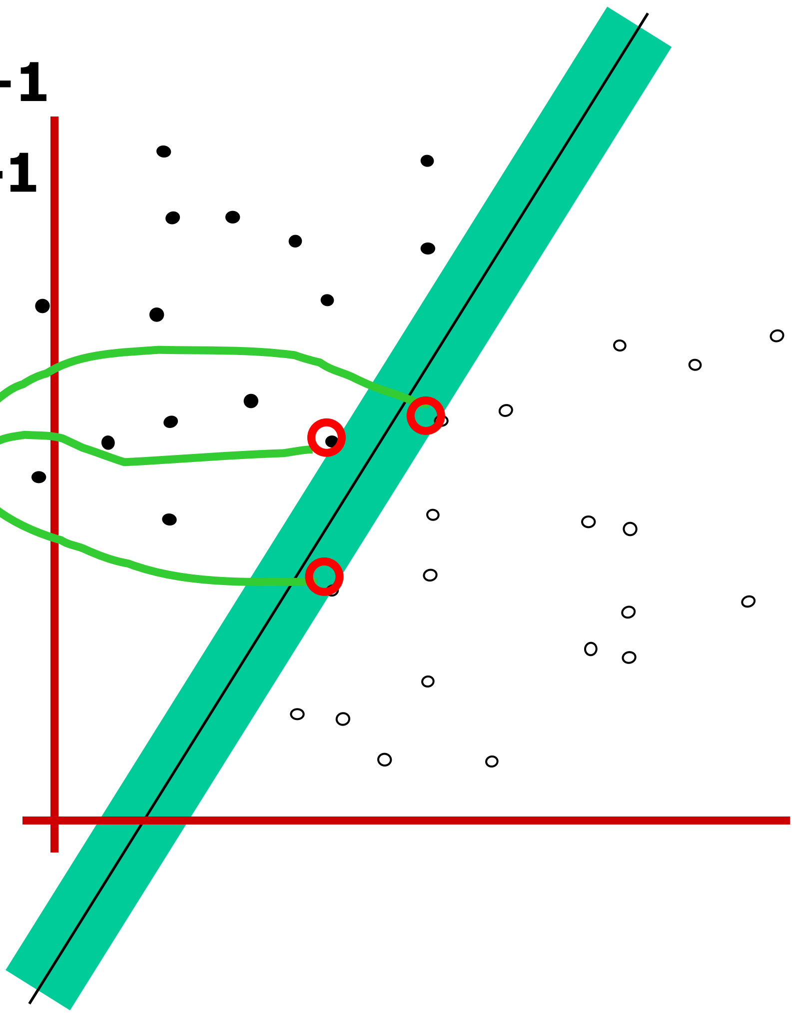
$$\mathrm{w} = \Sigma \alpha_i y_i \mathrm{x}_i \qquad b = y_k - \mathrm{w}^T\mathrm{x}_k \quad \text{for any } \mathrm{x}_k \text{ such that } \alpha_k \neq 0$$

- α_i must satisfy Karush-Kuhn-Tucker conditions:
  $\alpha_i\,[y_i(\mathrm{w}^T\mathrm{x}_i+b)-1]=0$, for any i
  - If $\alpha_i > 0$, $y_i(\mathrm{w}^T\mathrm{x}_i+b)-1=0$, x_i is on the margin
  - If $y_i(\mathrm{w}^T x_i+b)>1$, $\alpha_i = 0$
- Each non-zero α_i indicates that corresponding x_i is a **support vector**.

Maximum Margin

• **denotes +1**

° **denotes -1**



w, b depends only on Support Vectors via active constraints

$y_i(\mathrm{w}^T\mathrm{x}_k+b)-1=0$

The Optimization Problem Solution

- To classify the new test point x, we use

$$f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b = \Sigma\alpha_i y_i \boxed{\mathbf{x}_i^T\mathbf{x}} + b$$

Find $\alpha_1 \ldots \alpha_N$ **such that**

$\mathbf{Q}(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x}_i^T\mathbf{x}}_j$ **is maximized and**

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ **for all** α_i

Quadratic Programming

- Why is this reformulation a good thing?
- The problem

$$\text{Maximize} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

$$\text{subject to} \sum_i y_i \alpha_i = 0 \ \text{ and } \ \alpha_i \geq 0$$

is an instance of what is called a positive, semi-definite programming problem

- For a fixed real-number accuracy, can be solved in O(n log n) time
- CPLEX is the most commonly used software for doing this

Video

Example: https://youtu.be/5zRmhOUjjGY

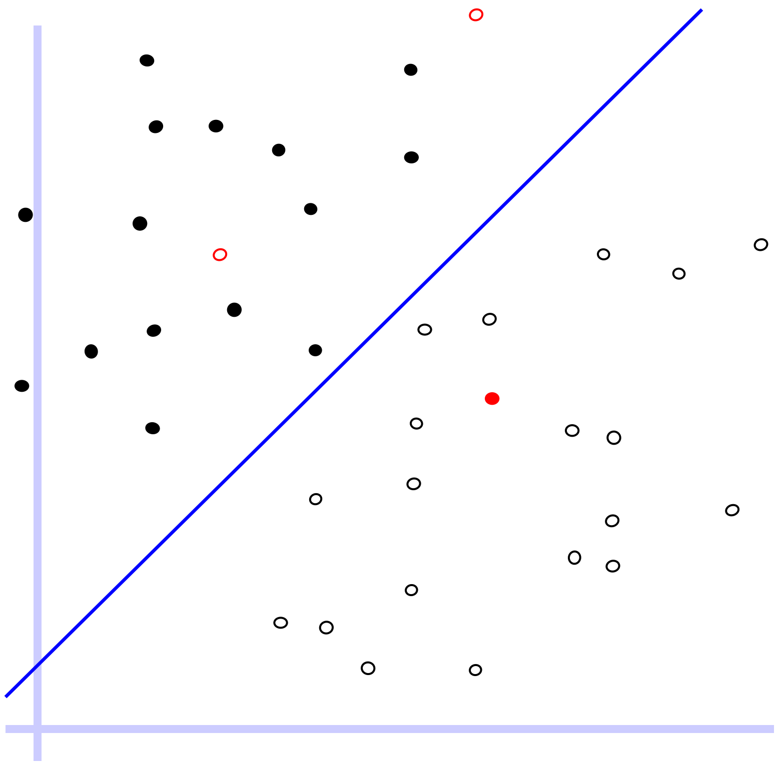
Outline

- Linear SVM – hard margin
- Linear SVM – soft margin
- Non-linear SVM
- Application

Dataset with noise

• denotes +1

◦ denotes -1



- **Hard Margin:** So far, all data points are classified correctly
  - No training error
- **What if the training set is noisy?**

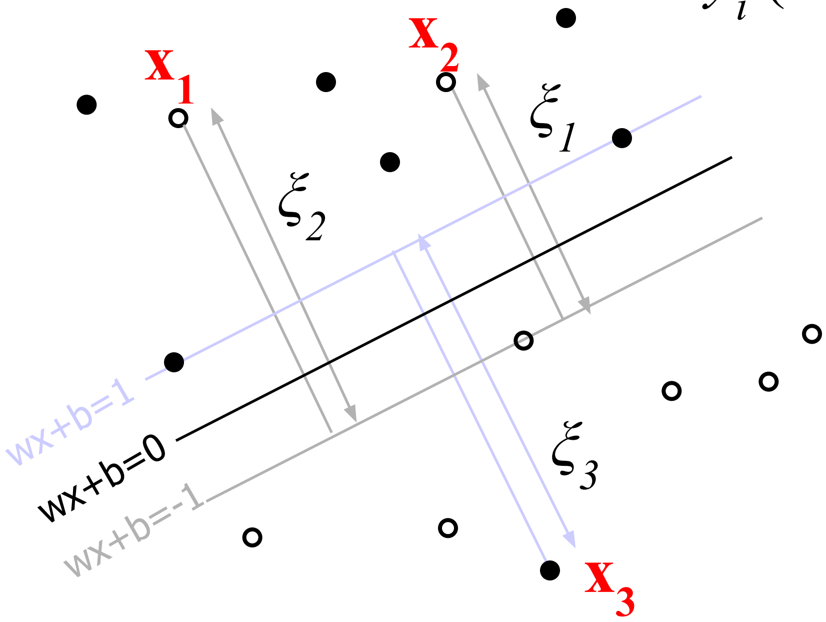
Soft Margin Classification

Previous constraints

$$y_i\,(\mathrm{w}^T\mathrm{x}_\mathrm{i} + b) \geq 1$$

Slack variables ξ_i to allow misclassification:

$$y_i\,(\mathrm{w}^T\mathrm{x}_\mathrm{i} + b) \geq 1 - \xi_i \qquad \xi_i \geq 0$$



What should our quadratic optimization criterion be?

We expect ξ_i to be small.

$$\Phi(\mathrm{w}) = \tfrac{1}{2}\,\mathrm{w}^T\mathrm{w} + C\Sigma\xi_i$$

Hard Margin vs. Soft Margin

- The old formulation:

Find w and b such that
$\Phi(\mathrm{w}) = \frac{1}{2}\,\mathrm{w}^T\mathrm{w}$ is minimized and for all $\{(\mathrm{x}_i, y_i)\}$
$y_i\,(\mathrm{w}^T\mathrm{x}_i + \mathrm{b}) \geq 1$

- The new formulation incorporating slack variables:

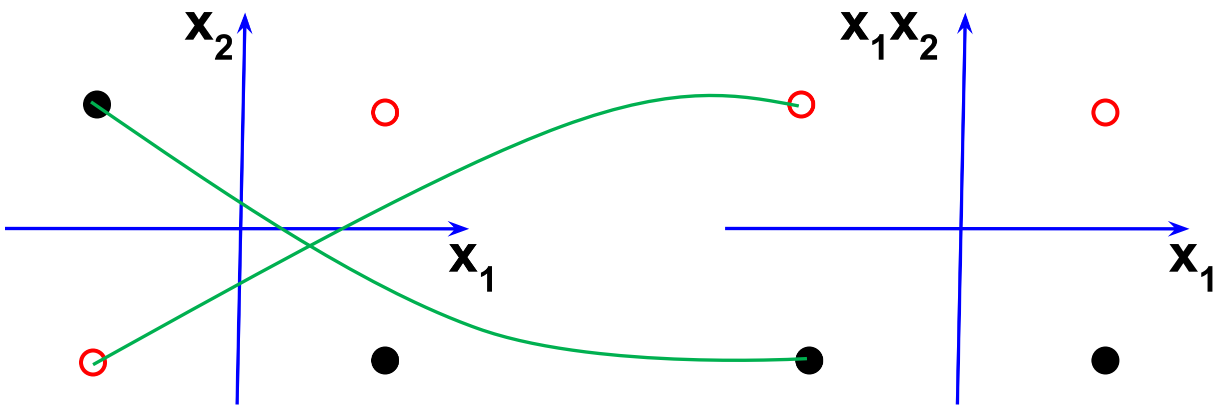
Find w and b such that
$\Phi(\mathrm{w}) = \frac{1}{2}\,\mathrm{w}^T\mathrm{w} + \boxed{C\Sigma\xi_i}$ is minimized and for all $\{(\mathrm{x}_i, y_i)\}$
$y_i\,(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 - \boxed{\xi_i}$ and $\xi_i \geq 0$ for all i

- Similar solution can be obtained to that of hard margin
- Parameter *C* can be viewed as a way to control overfitting.

Outline

- Linear SVM – hard margin
- Linear SVM – soft margin
- Non-linear SVM
- Application

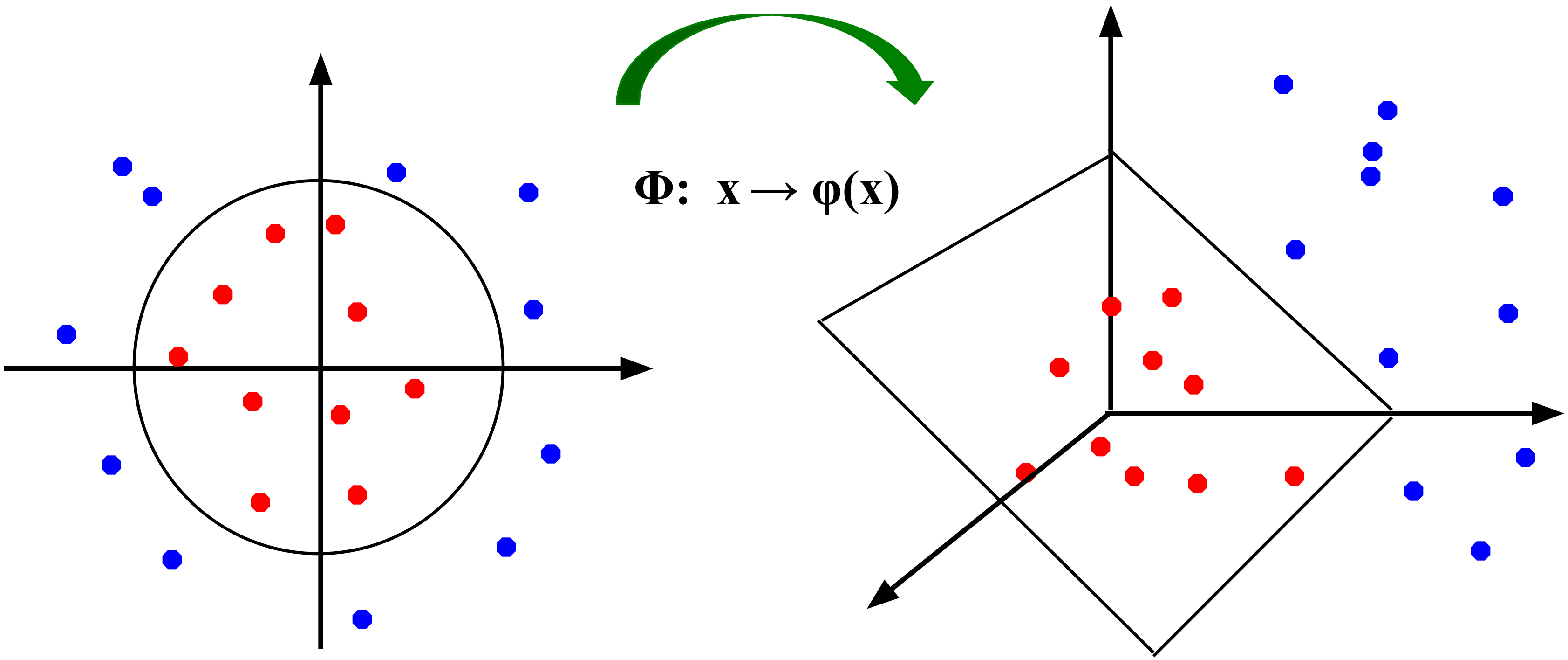
XOR problem



- XOR data are not linearly separable
- Mapping (x_1, x_2) to $(x_1, x_1 x_2)$

Non-linear SVMs: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Non-linear SVMs

- **If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \varphi(x)$, optimization problem is similar:**

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\varphi(x_i)^T\varphi(x_j)}$ is maximized
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- **Classifying function is:**

$$f(x) = \Sigma\alpha_i y_i \boxed{\varphi(x_i)^T\varphi(x)} + b$$

- **But relies on inner product $\varphi(x_i)^T\varphi(x)$**

The "Kernel Trick"

- SVM relies on
  - *Linear:* $K(\mathrm{x}_i,\mathrm{x}_j)=\mathrm{x}_i^T\mathrm{x}_j$
  - *Non-linear:* $K(\mathrm{x}_i,\mathrm{x}_j)=\varphi(\mathrm{x}_i)^T\varphi(\mathrm{x}_j)$
- Feature mapping is time-consuming.
- Use a kernel function that directly obtains the value of inner product
- Feature mapping φ is not necessary in this case.

- Example:

  2-dimensional vectors $\mathrm{x}=[x_1\ \ x_2]$; let $K(\mathrm{x}_i,\mathrm{x}_j)=(1+\mathrm{x}_i^T\mathrm{x}_j)^2$,

  It is inner product of $\varphi(\mathrm{x}) = [1\ \ x_1^2\ \ \sqrt{2}\,x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]$

Verify: $K(\mathrm{x}_i,\mathrm{x}_j)=(1+\mathrm{x}_i^T\mathrm{x}_j)^2$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\,x_{i1}x_{j1}\,x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1\ \ x_{i1}^2\ \ \sqrt{2}\,x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T[1\ \ x_{j1}^2\ \ \sqrt{2}\,x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]$$

$$= \varphi(\mathrm{x}_i)^T\varphi(\mathrm{x}_j),$$

Examples of Kernel Functions

- **Linear: $K(x_i, x_j) = x_i^T x_j$**

- **Polynomial of power *p*: $K(x_i, x_j) = (1 + x_i^T x_j)^p$**

- **Gaussian (radial-basis function network):**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$

- **Sigmoid: $K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1)$**

Non-linear SVMs Mathematically

- **Dual problem formulation:**

$$\boxed{\begin{array}{l}\textbf{Find } \alpha_1 \ldots \alpha_N \textbf{ such that} \\ Q(\alpha) = \Sigma\alpha_i - \tfrac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j) \textbf{ is maximized and} \\ (1)\ \ \Sigma\alpha_i y_i = 0 \\ (2)\ \alpha_i \geq 0 \textbf{ for all } \alpha_i \end{array}}$$

- **The solution is:**

$$\boxed{f(x) = \Sigma\alpha_i y_i K(x_i, x_j) + b}$$

- **Optimization techniques for finding α_i's remain the same!**

Nonlinear SVM - Overview

- The feature is mapped to a high dimensional space where
  - training data are separable.
  - inner product is computed by kernel function.
- Optimization problem is similar to linear SVM

Overfitting

- It can be shown that: the portion, n, of unseen data that will be missclassified is bounded by:
  - n <= number of support vectors / number of training examples
- In SVM case: fewer support vectors mean a simpler representation of the hyperplane.

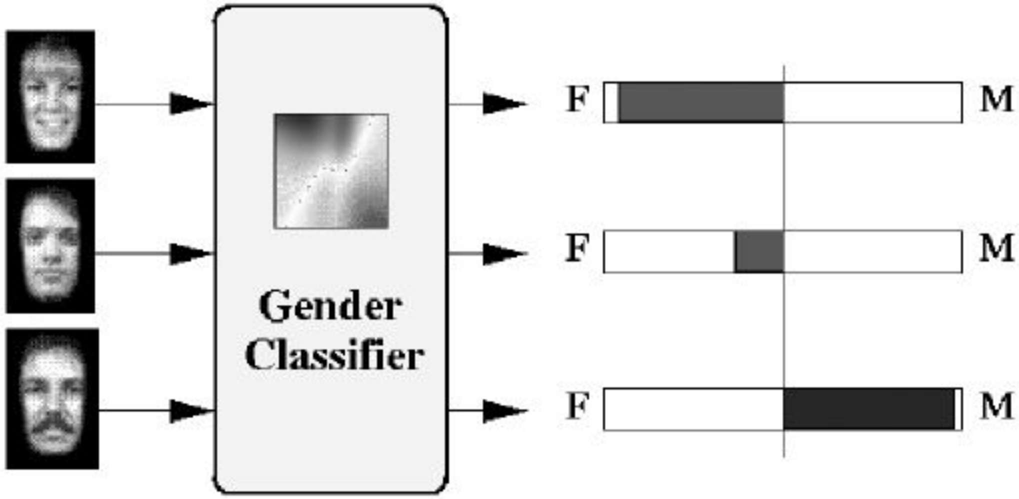
Large Amounts of Data

- In theory having a lot of data is great for improving classification.
- But it could easily mean that expensive methods like SVMs (train time) or kNN (test time) are quite impractical
- Naïve Bayes can come back into its own again!
- Or other advanced methods with linear training/test complexity like regularized logistic regression (though much more expensive to train).

Outline

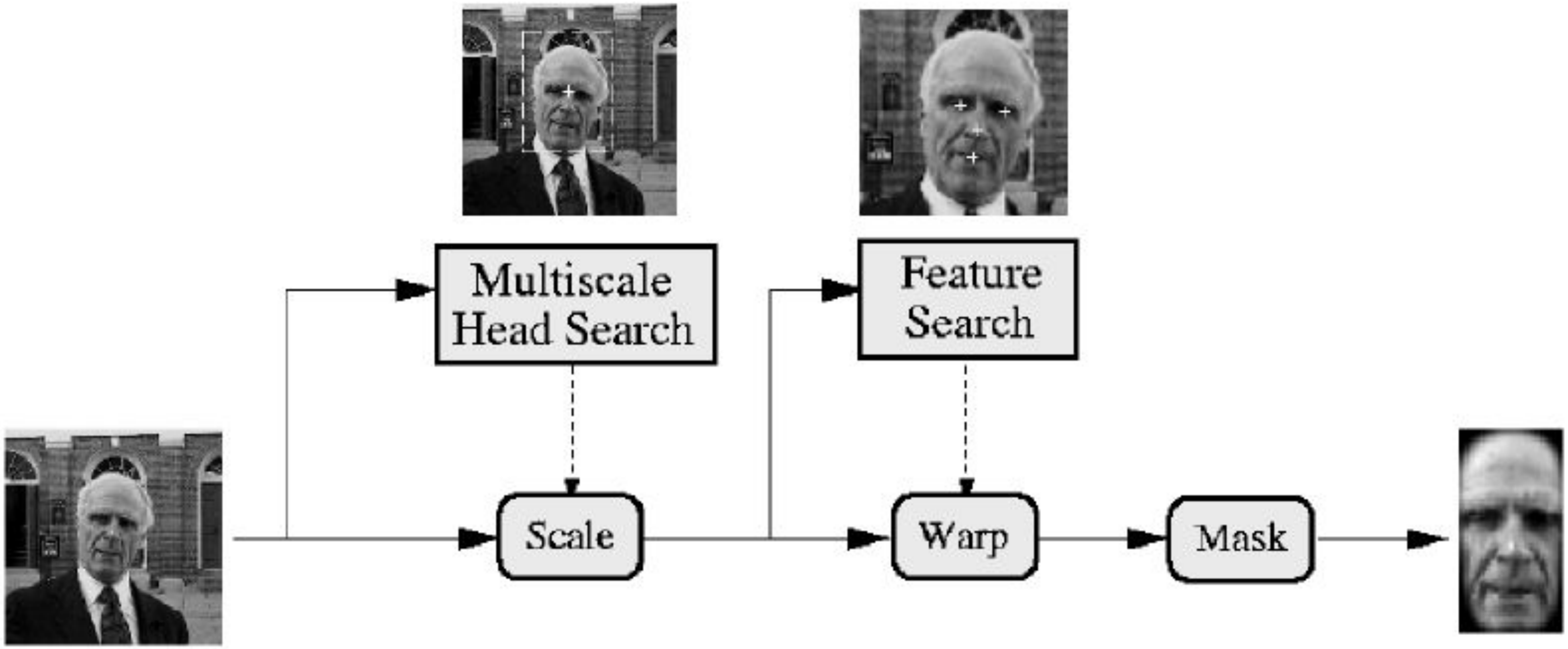
- Linear SVM – hard margin
- Linear SVM – soft margin
- Non-linear SVM
- Applications

Gender recognition



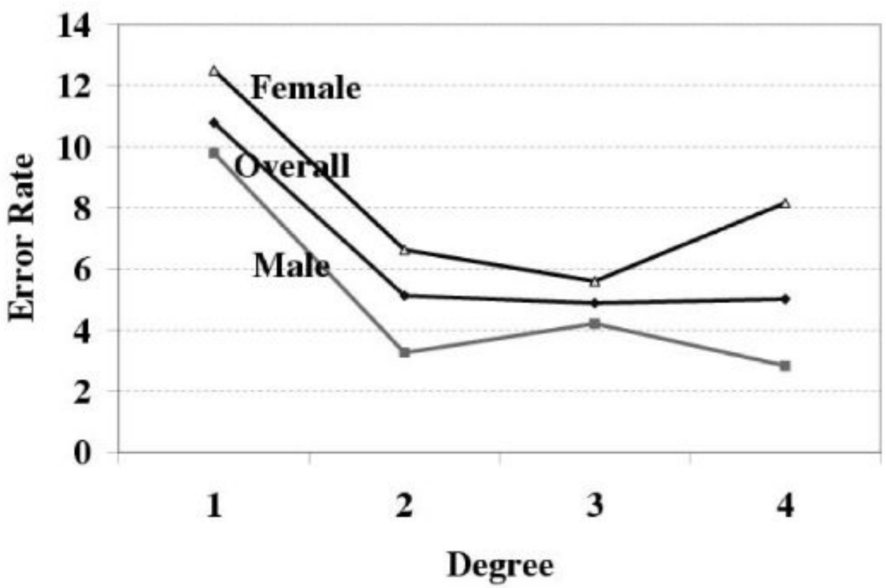
- **Application:**
  - **Adaptive advertisement**
  - **Collection of demographic information in shopping mall**

Face cropping and alignment

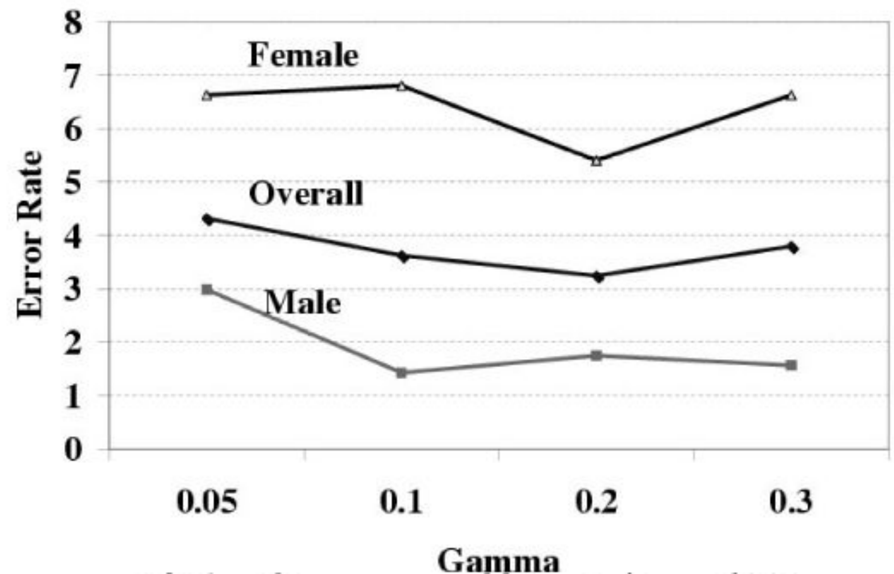


Recognition rate

- 1044 male image, 711 female
- 4/5 for training, 1/5 for testing
- Higher error rate in classifying female

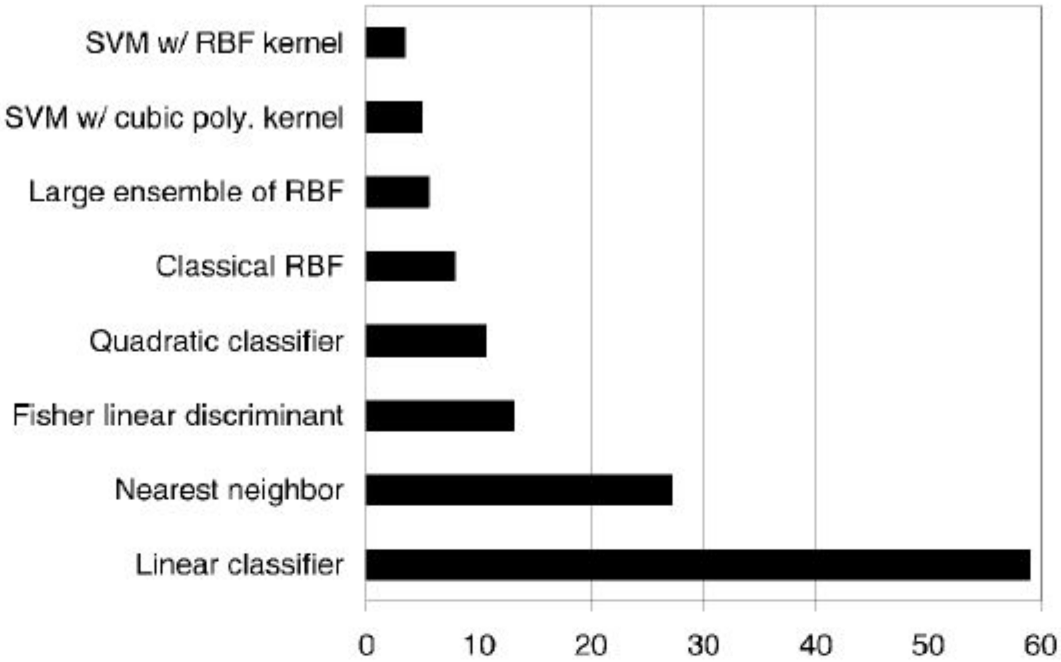


$$k(\mathbf{x}, \mathbf{x}_i) = ((\mathbf{x} \cdot \mathbf{x}_i) + 1)^d$$



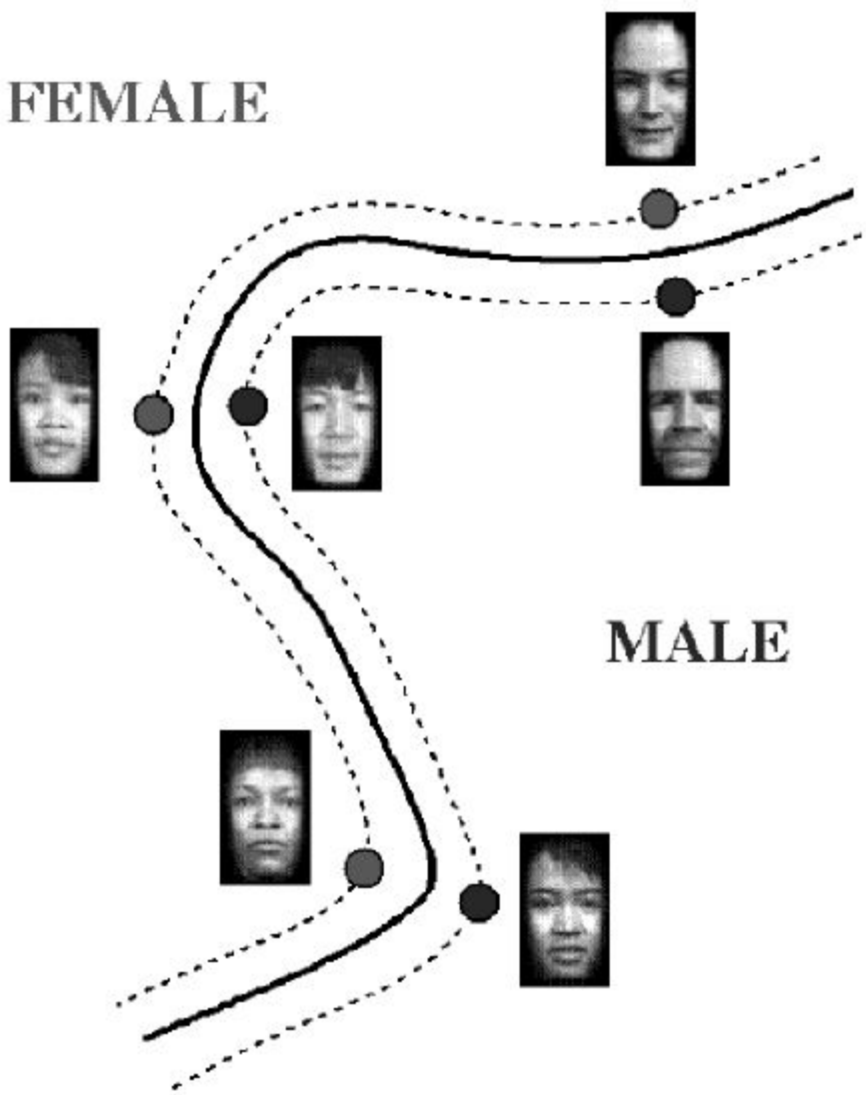
$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma|\mathbf{x} - \mathbf{x}_i|^2)$$

Comparison



SVM w/ RBF kernel
SVM w/ cubic poly. kernel
Large ensemble of RBF
Classical RBF
Quadratic classifier
Fisher linear discriminant
Nearest neighbor
Linear classifier
0
10
20
30
40
50
60

Support face

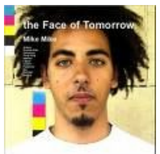


- Each pair are closest in the projected high dimensional space

Application - Object Recognition

Face **Bike**

Application - Object Recognition



x, y=1

x, y=1

x, y=1

x, y=-1

x, y=-1

x, y=-1

Train SVM: find w and b

x

$$f(\mathrm{x}) = \Sigma \alpha_i y_i K(\mathrm{x}_i, \mathrm{x}_j) + b$$

y=-1

Resources

- **References**
  - Vladimir Vapnik: The Nature of Statistical Learning Theory. Springer-Verlag, 1995.
  - Christopher J. C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998
  - Bernhard Scholkopf and A. J. Smola: Learning with Kernels. 2002
- **A useful website: www.kernel-machines.org**
- **Software:**
  - LIBSVM: www.csie.ntu.edu.tw/~cjlin/libsvm/
  - SVMLight: svmlight.joachims.org/