Machine Learning

Amirkabir University of Technology
(Tehran Polytechnic)

Lecture 5.
Supervised learning
Support Vector Machine (SVM)

Alireza Rezvanian

Fall 2022

Last update: Nov. 09, 2022

Amirkabir University of Technology (Tehran Polytechnic)

Outline

- SVM Intuitions
- Maximizing the Margin
- Soft Margin Hyperplane
- Slack variables
- Kernel function
- Multiclass SVMs

Support Vector Machine (SVM)

- **S**upport **V**ector **M**achines are systems for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalisation theory and exploiting optimisation theory.
  - Cristianini & Shawe-Taylor (2000)
- Also called Sparse kernel machines
  - Kernel methods predict based on linear combinations of a kernel function evaluated at the training points,
  - Sparse because not all pairs of training points need be used
- Also called Maximum margin classifiers
- Widest street approach

SVM is easy

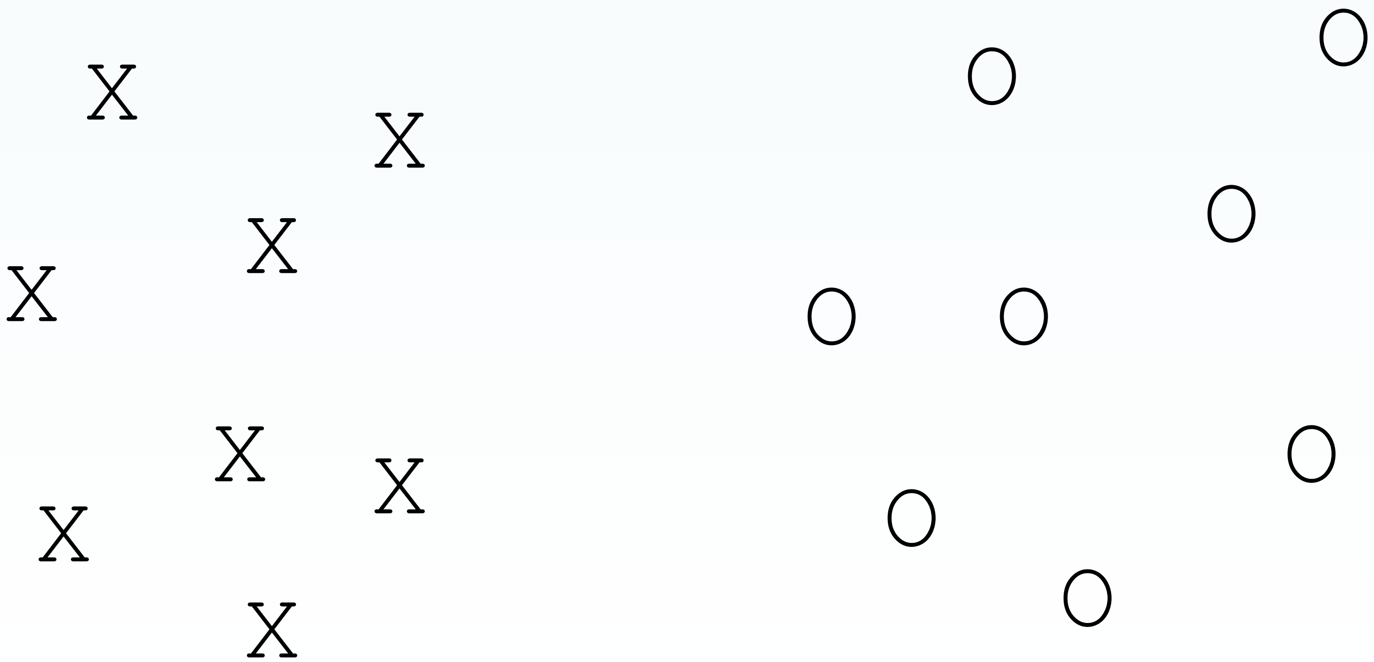
- Input: x
- Model: w
- Score: $w^T x$
- Prediction: $\text{sgn}(w^T x)$
- But how do we learn w?

$$h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$$

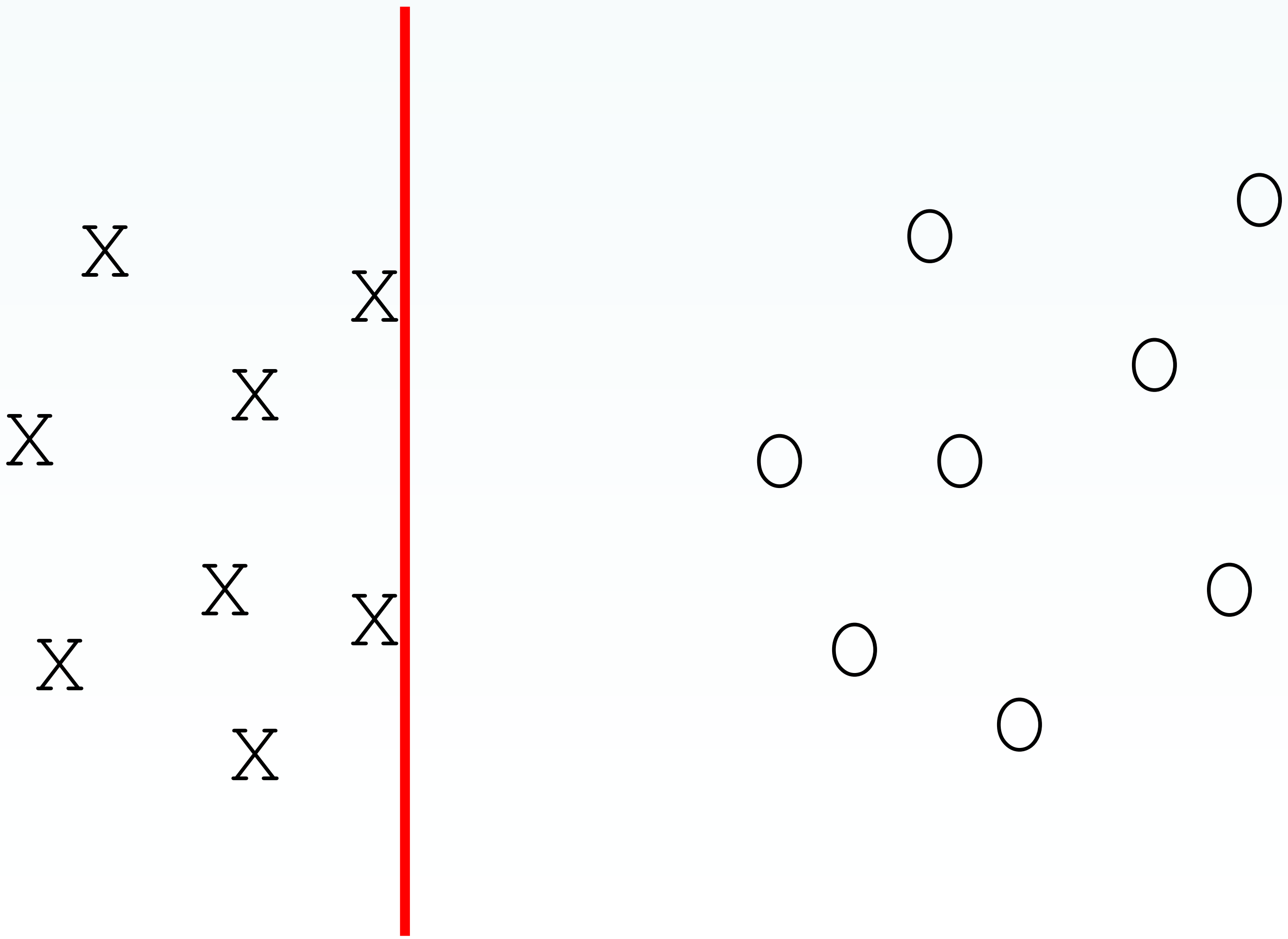
$$\min_{w,b} \frac{1}{2} w^T w + c \sum_i \xi_i$$

$$\xi_i = \max(0, 1 - y_i(w^T x_i + b))$$

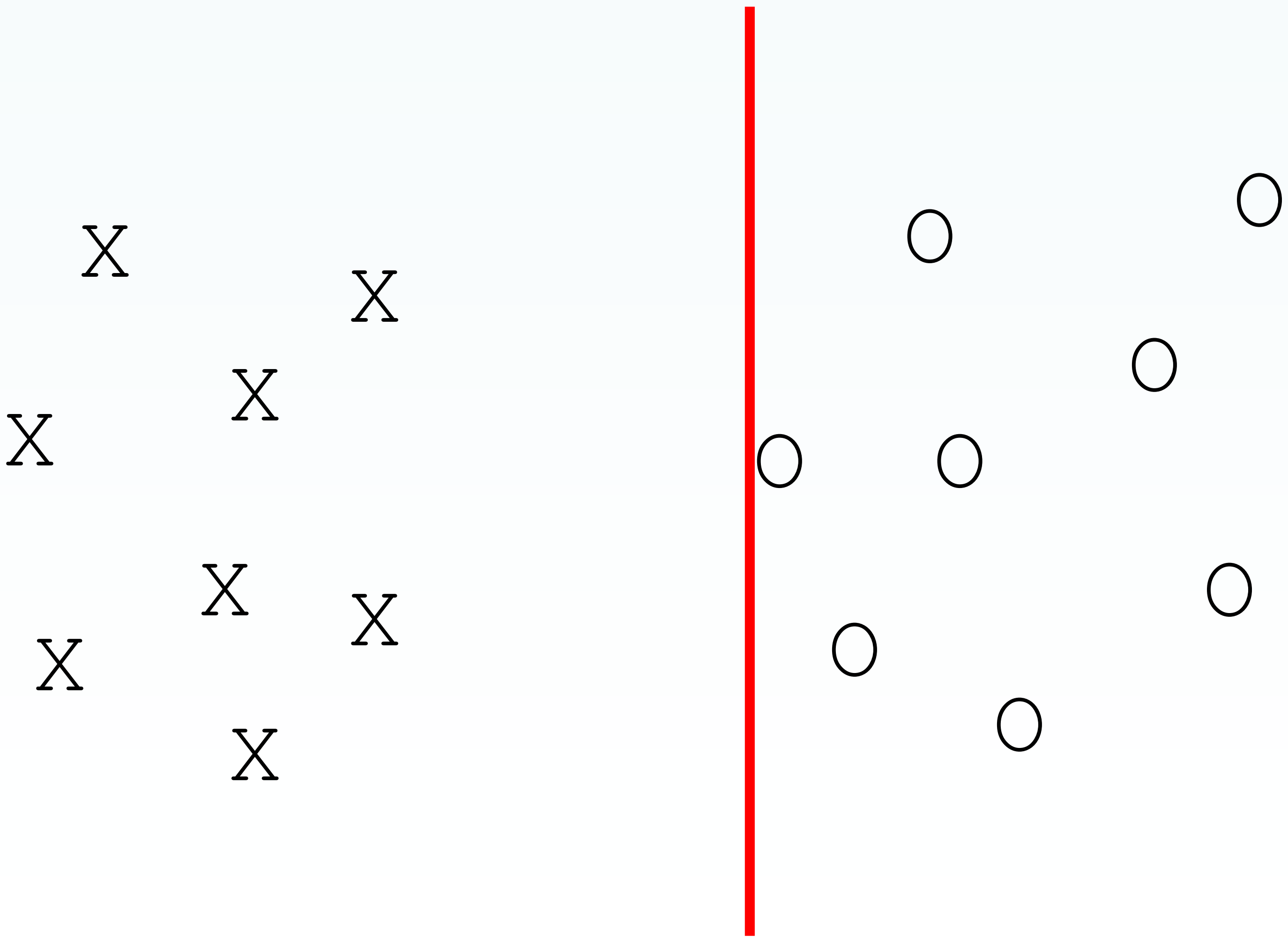
Intuitions



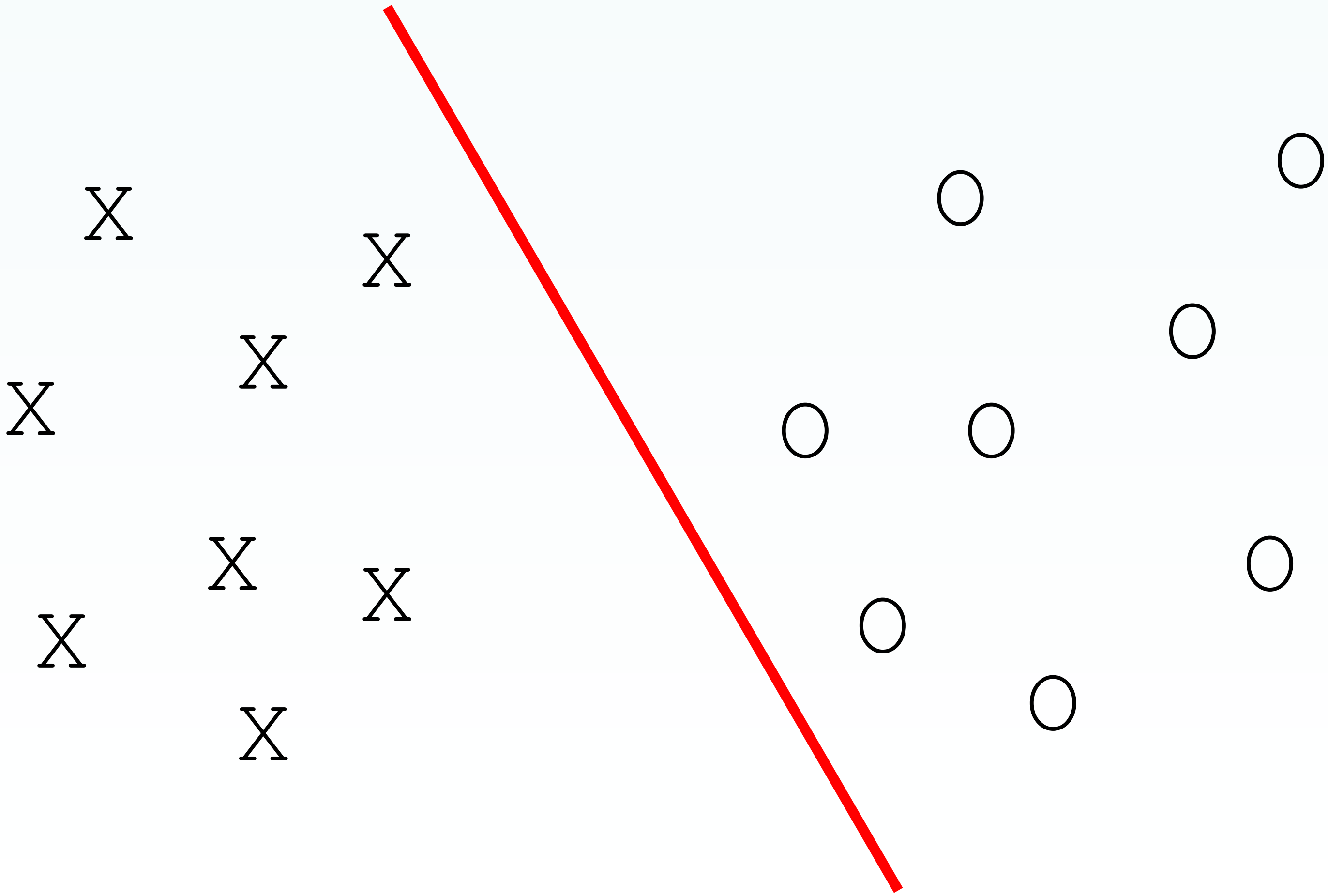
Intuitions



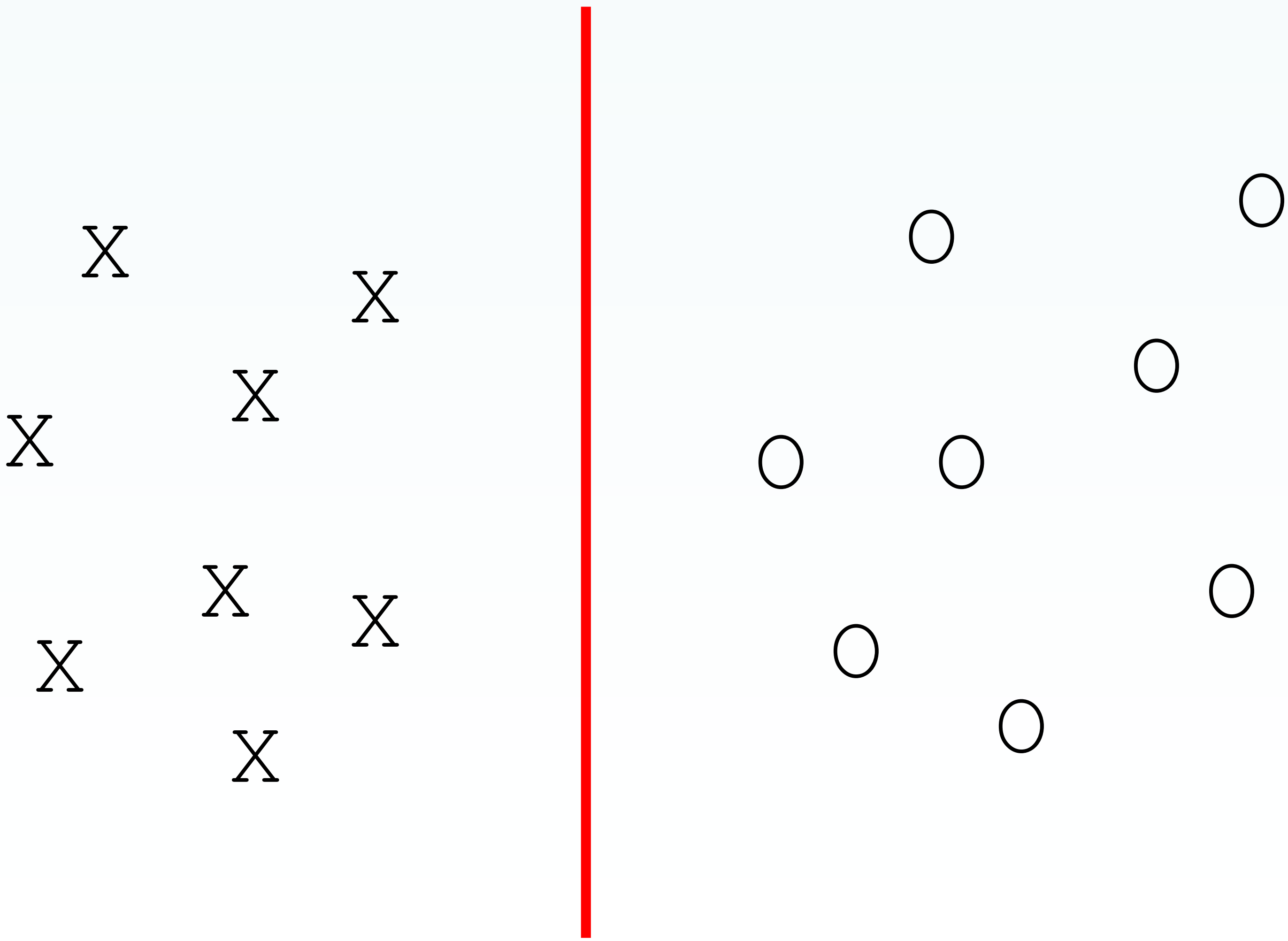
Intuitions



Intuitions



Intuitions



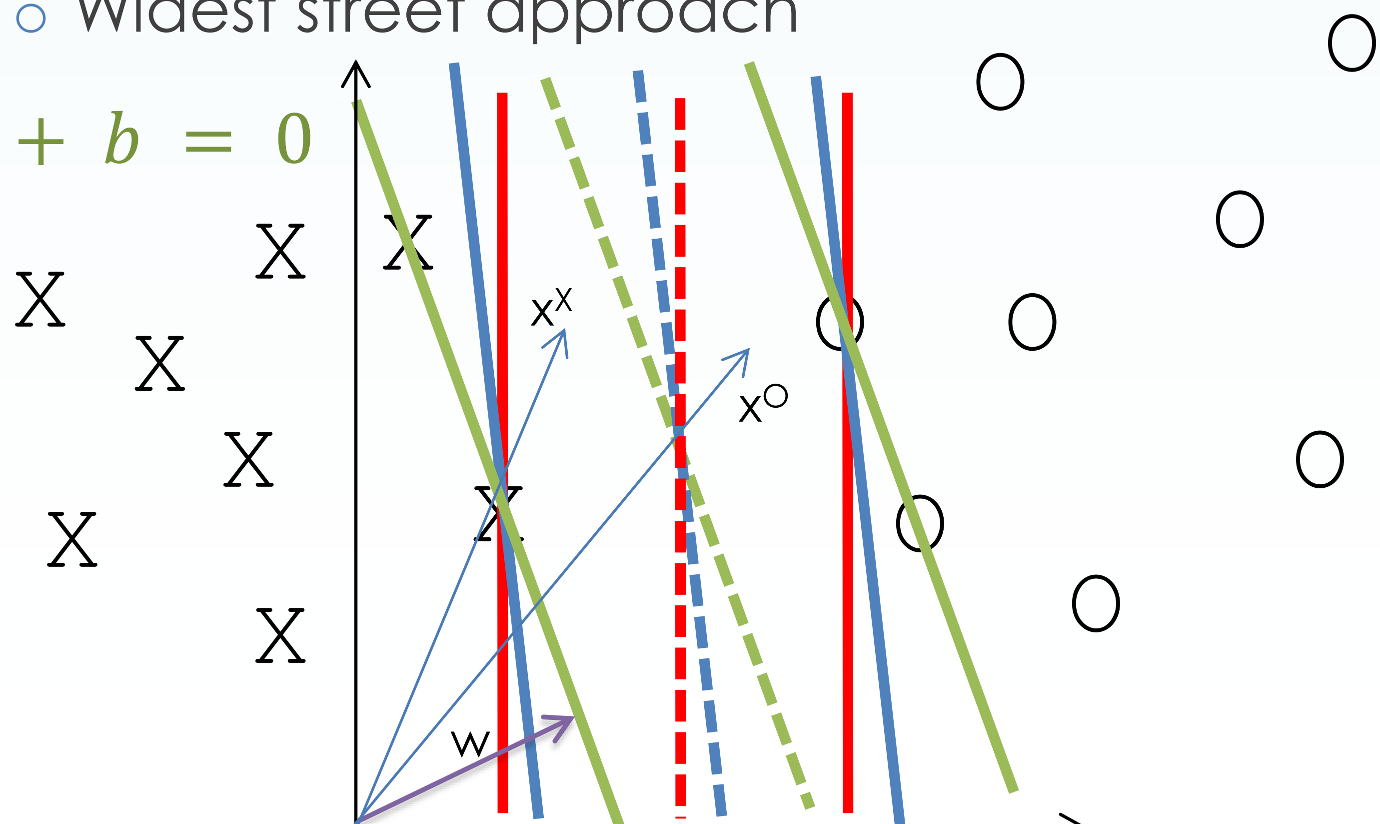
SVM Intuitions

- Vladimir Vapnik
  - h: X→{-1, +1}
  - Maximum margin classifiers
  - Widest street approach

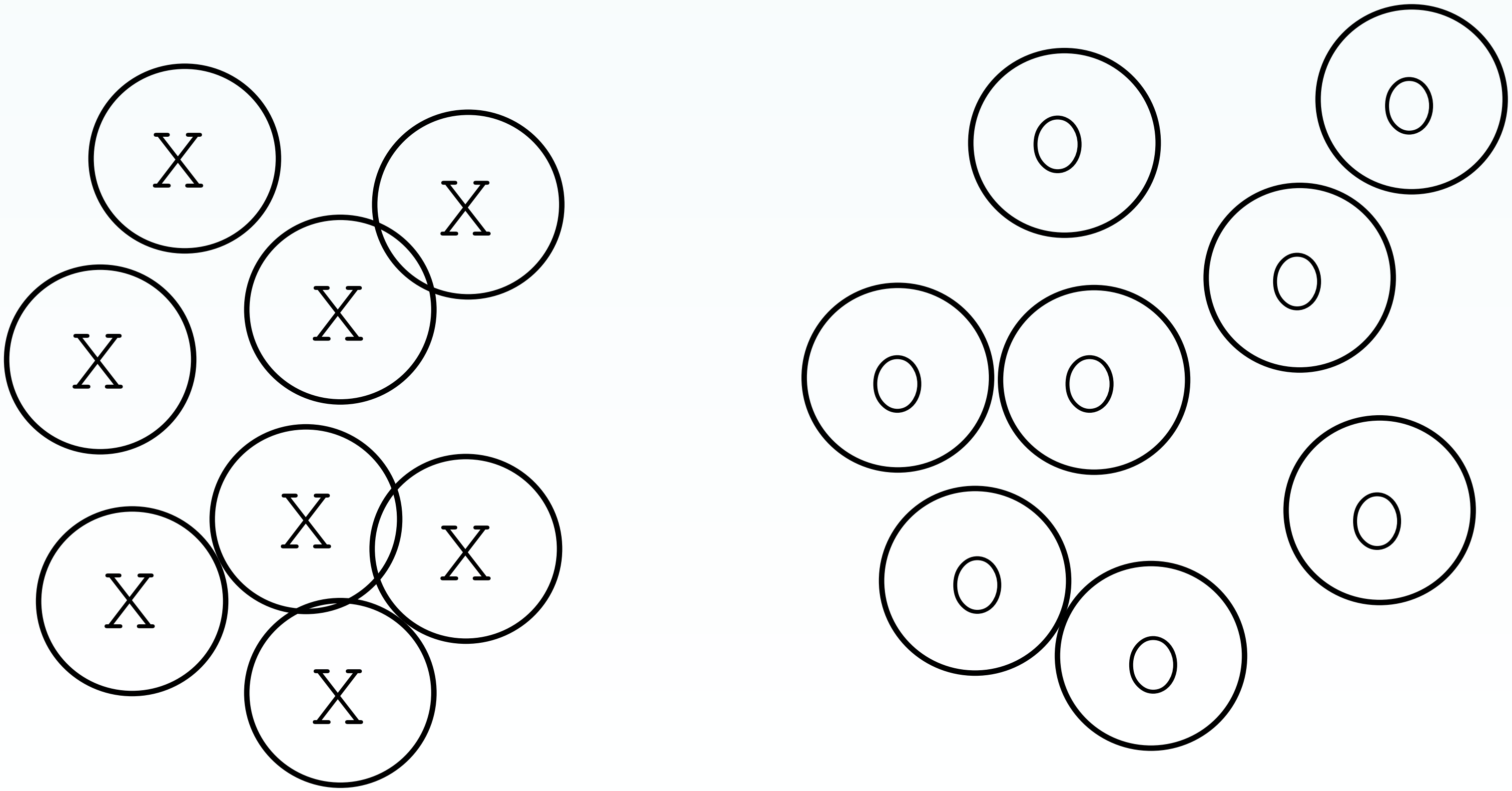
If $(w.x_t \geq 0)$ then class is O

If $(w.x_t + b \quad 0)$ then class is O, s.b. b=-c

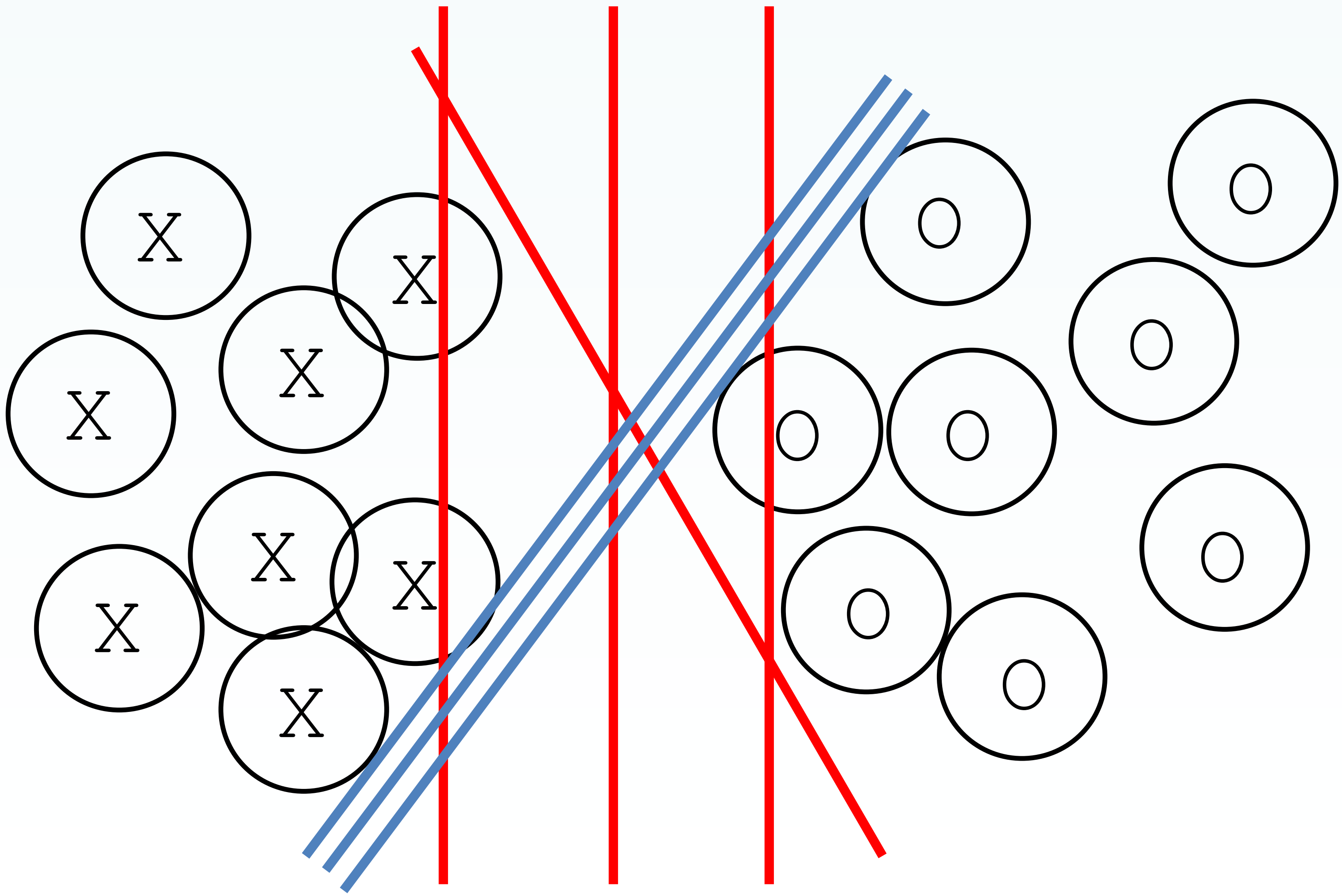
$$w.x + b = 0$$



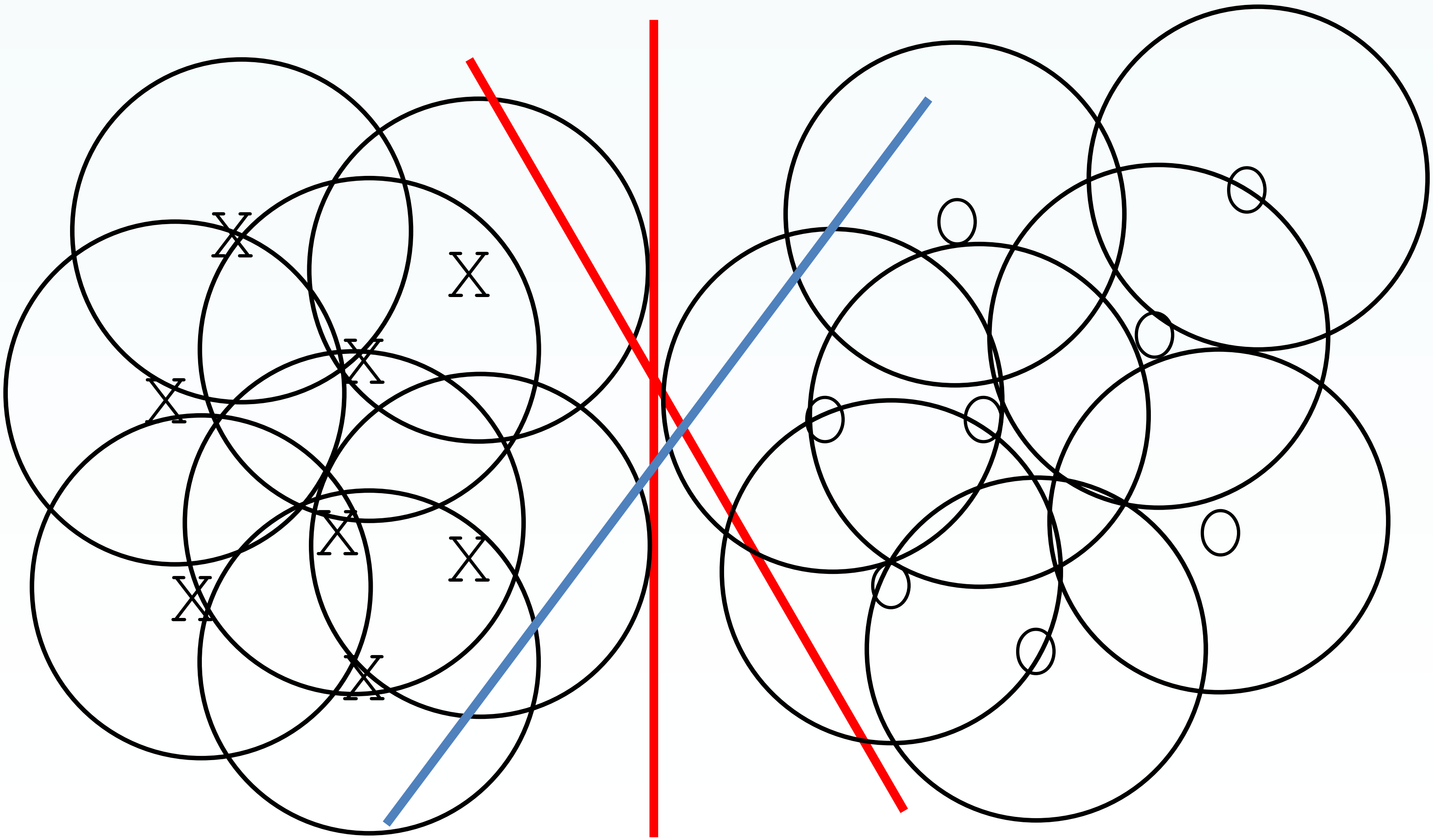
Noise in the Observations



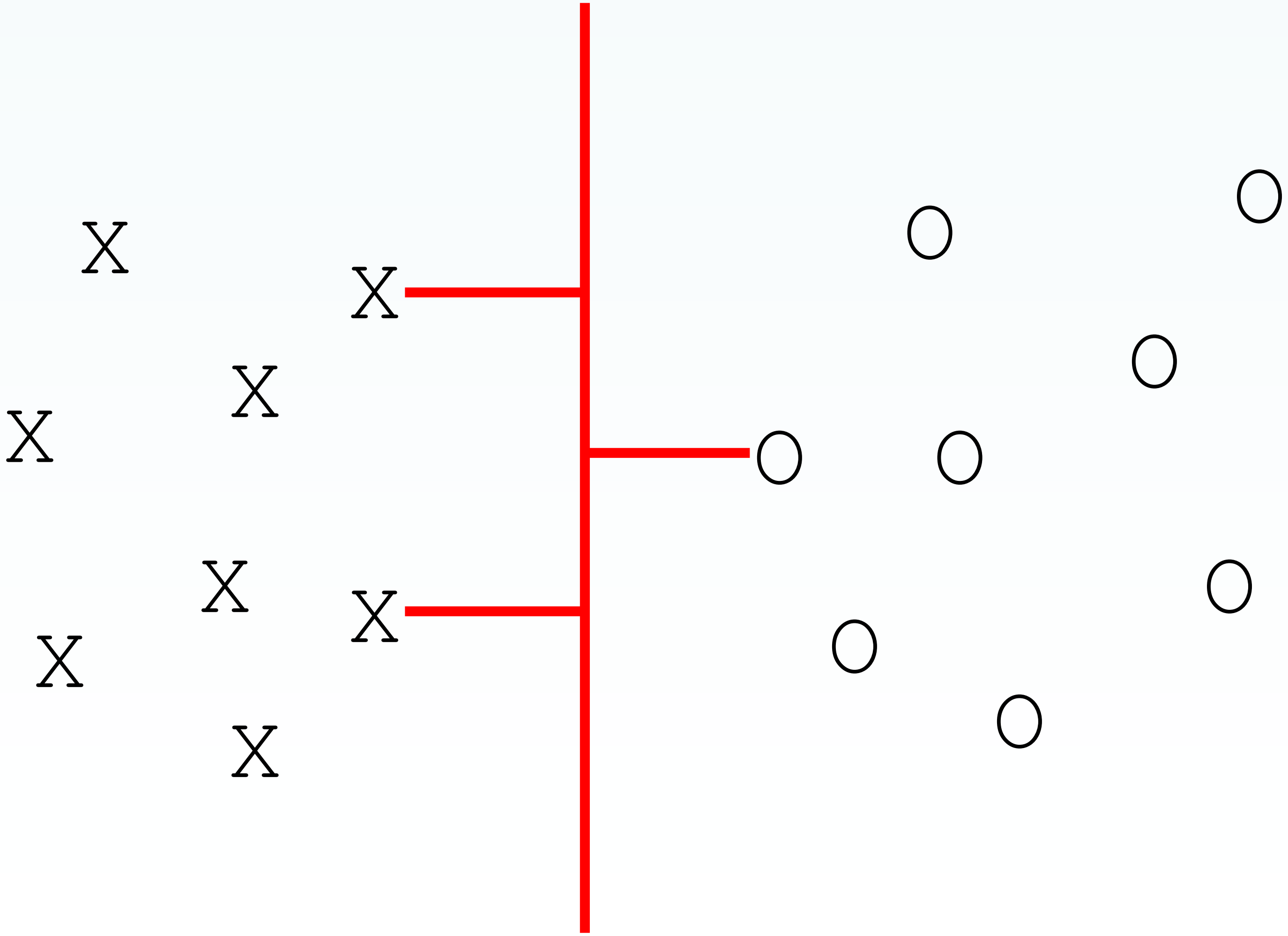
Noise in the Observations



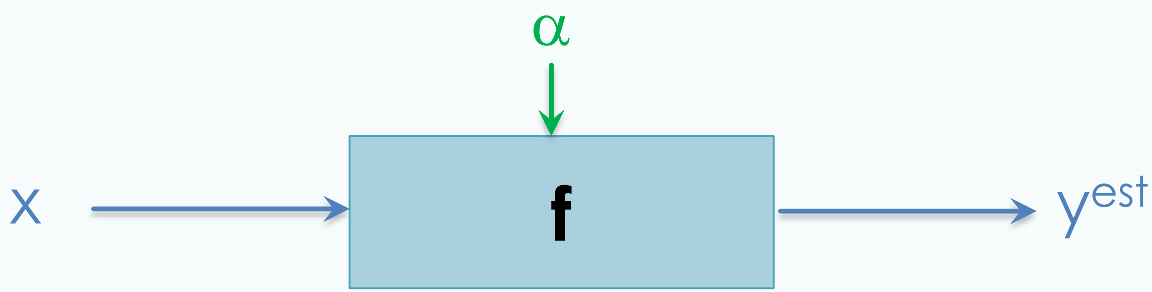
Noise in the Observations



Maximizing the Margin

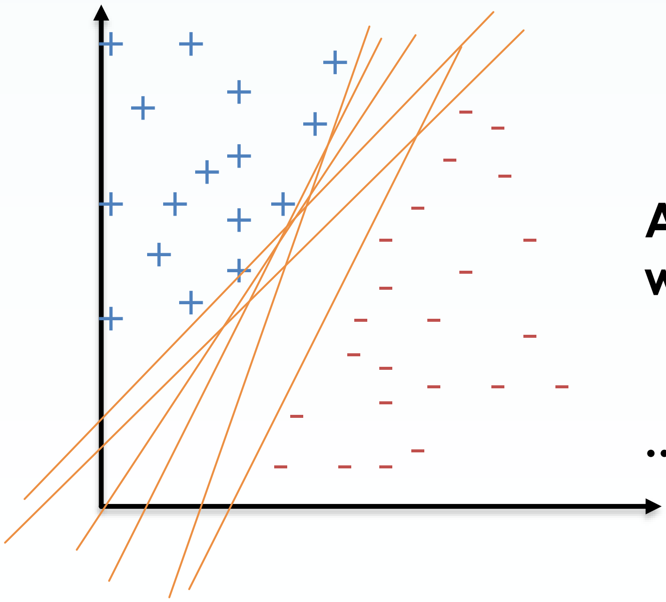


Linear Classifiers



How would you classify this data?

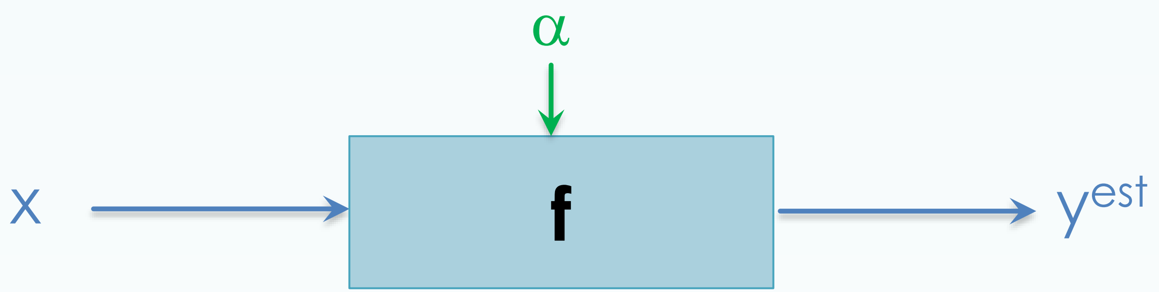
$f(x, w, b) = sgn(w. x - b)$



Any of these would be fine..

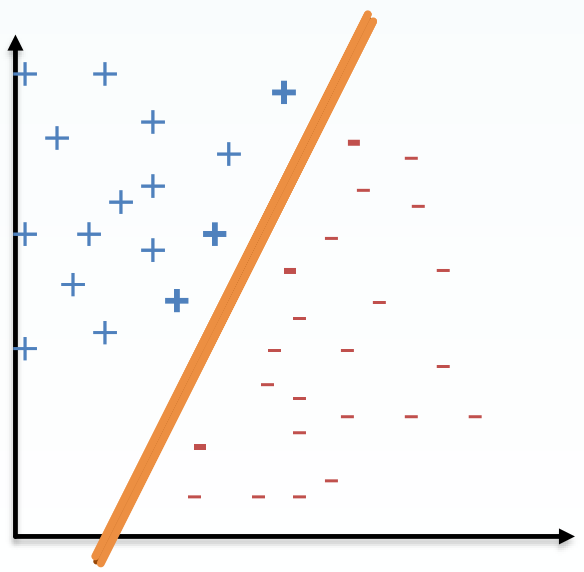
..but which is best?

Linear Classifiers



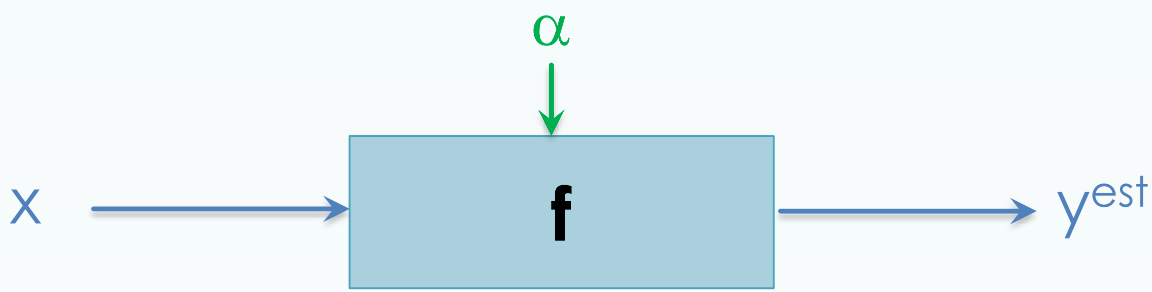
$$f(x, w, b) = \mathrm{sgn}(w. x - b)$$

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



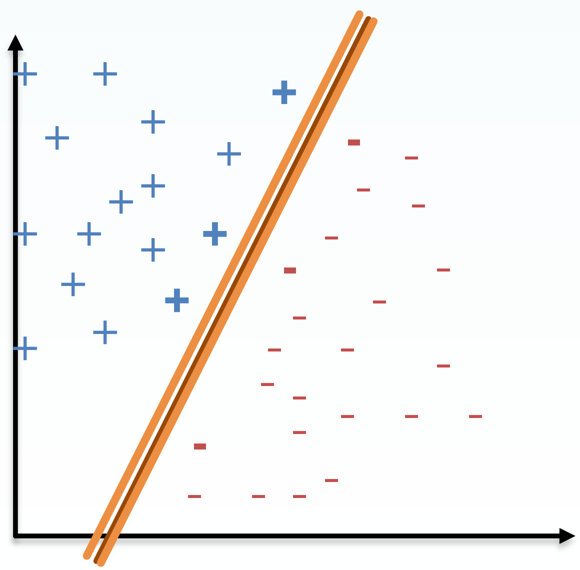
A "good" separator

Linear Classifiers

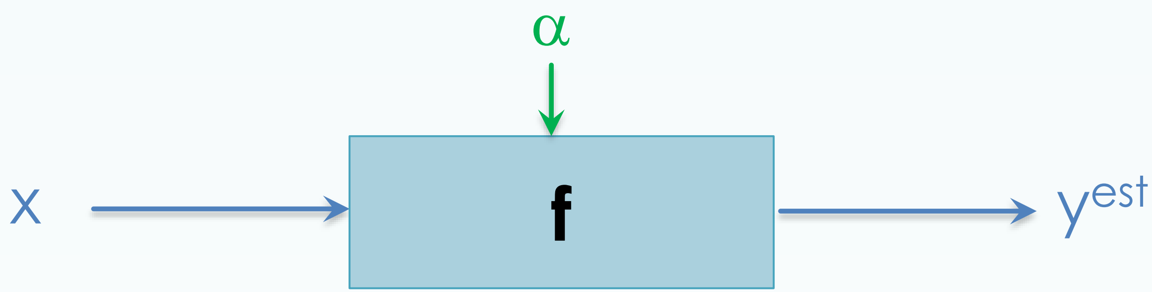


$$f(x, w, b) = \mathrm{sgn}(w.\ x - b)$$

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

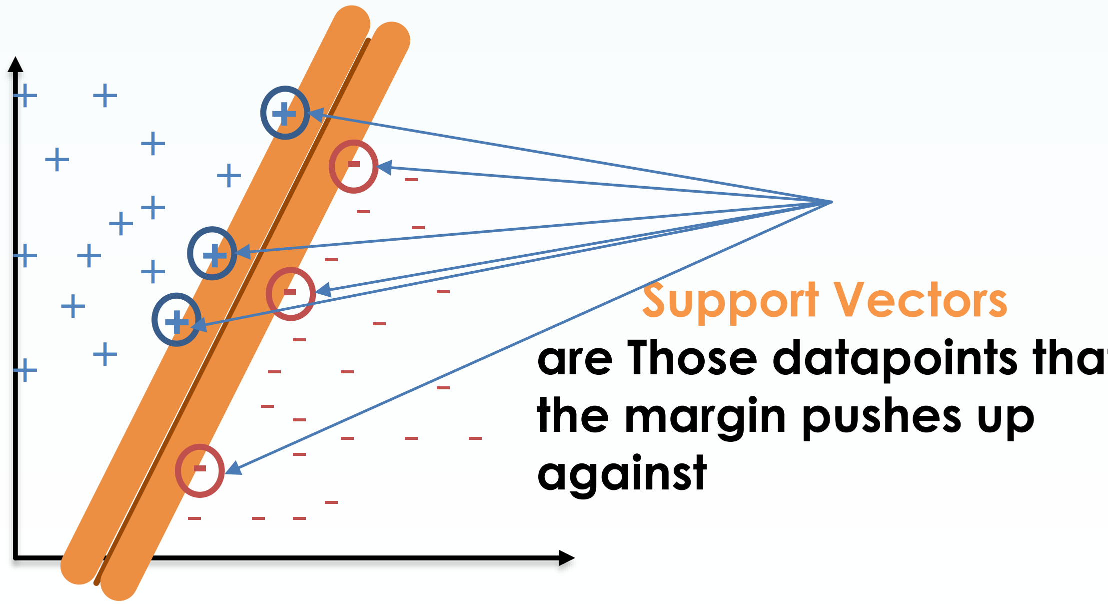


Linear Classifiers



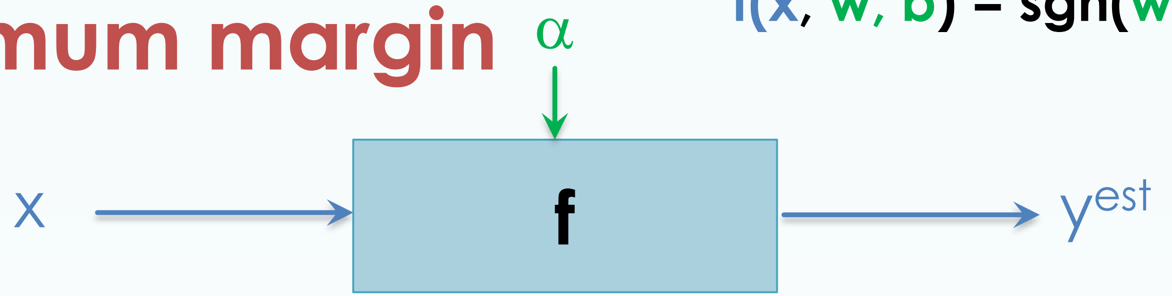
$$f(x, w, b) = \mathrm{sgn}(w \cdot x - b)$$

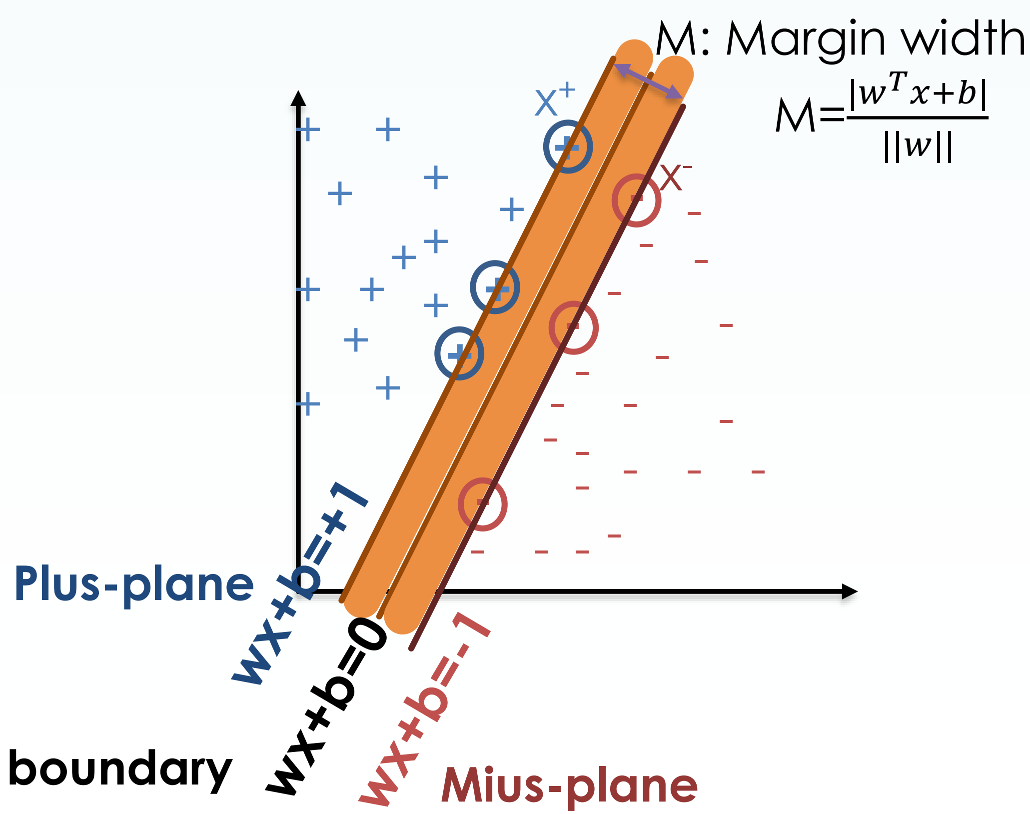
The **maximum margin linear classifier** is the linear classifier with the, um, maximum margin.
This is the simplest kind of SVM (Called an LSVM)



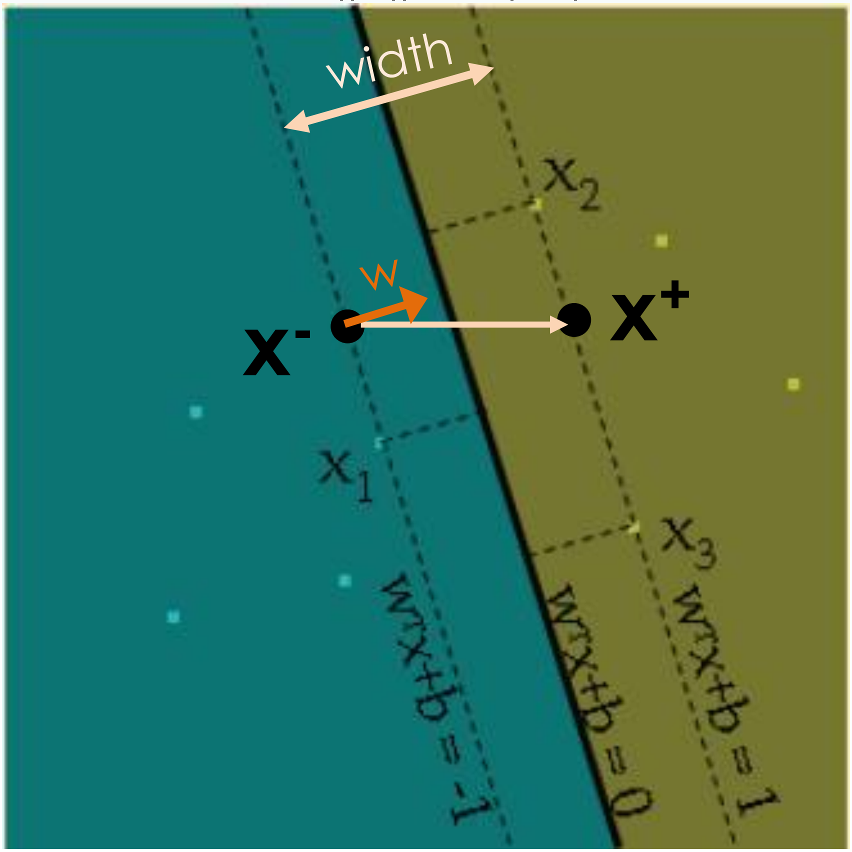
Linear Classifiers maximum margin

$$f(x, w, b) = sgn(w. x - b)$$





$$M = \frac{|w^T x + b|}{||w||} = \frac{2}{||w||}$$



$$h(x) = sign(w^T x + b)$$

x_1, x_2, and x_3 are support vectors

$$width = (x^+ - x^-).\frac{w}{||w||}$$

$$=(w.x^+ - w.x^-).\frac{1}{||w||} = \frac{2}{||w||}$$

Note:

$$w.x^+ + b = +1$$

$$w.x^- + b = -1$$

$$\frac{|w.x + b|}{||w||} = \frac{1}{||w||}$$

$$M = \frac{2}{||w||}$$

- **Compute margin**

$w^T x_1 + b = +c$ and $w^T x_2 + b = -c$

$$\boxed{\times \frac{1}{c}}$$

- Maximize margin

$$\underset{w,b}{Max} \frac{2}{||w||} \qquad \underset{w,b}{Min} ||w||^2 \qquad ||w||^2 = w^T w$$

s.t.

$$(w^T x_1 + b) \geq 1 \quad \forall_{y=1}$$
$$(w^T x_1 + b) \leq -1 \quad \forall_{y=-1}$$

$$\Longleftrightarrow \quad y_i(w^T x_1 + b) \geq 1$$

- Solve efficiently by quadratic programming (QP)
  - Well-studied solution algorithms
- Hyperplane defined by support vectors

- Solving SVM by Quadratic Programming (QP) a.k.a., primal problem

α_i: Lagrange Multiplier

$f(x) - \sum_{i=1}^{n} \alpha_i g(x)$: Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w.x_i + b) - 1]$$

s. t. $\alpha_i \geq 0$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0$$

→parameters are expressed as a linear combination of training points

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$KKT\ cond: \alpha_i [y_i(\mathbf{w}\mathbf{x}_i + b) - 1] = 0$$

→only SVs will have non-zero α_i

The Dual Problem

- If we substitute $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$ to Lagrangian, we have

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b)\right)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i - b\sum_{i=1}^{n} \alpha_i y_i$$

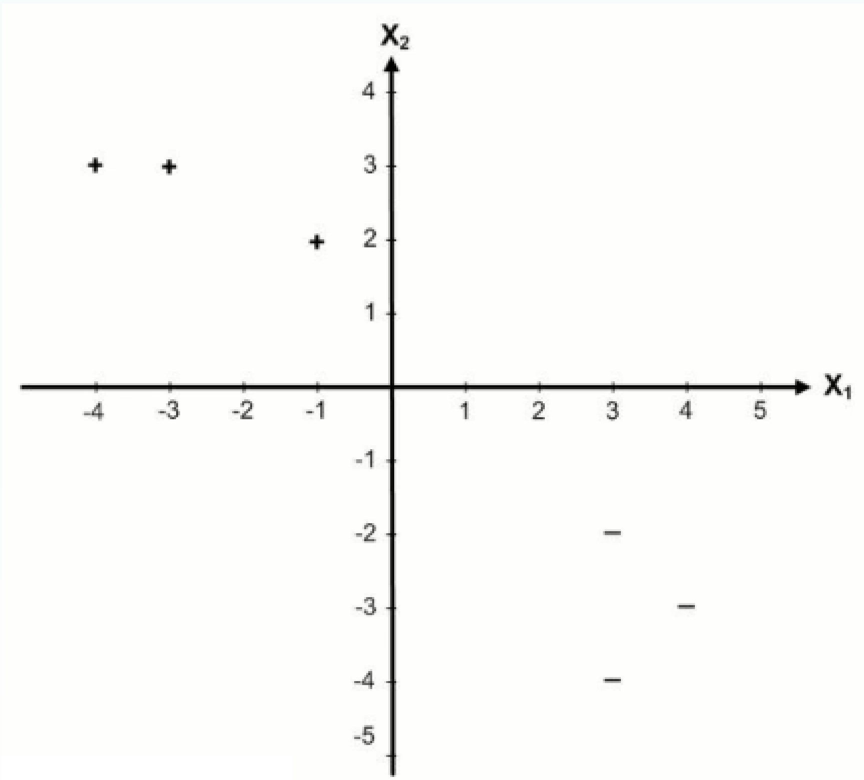
$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

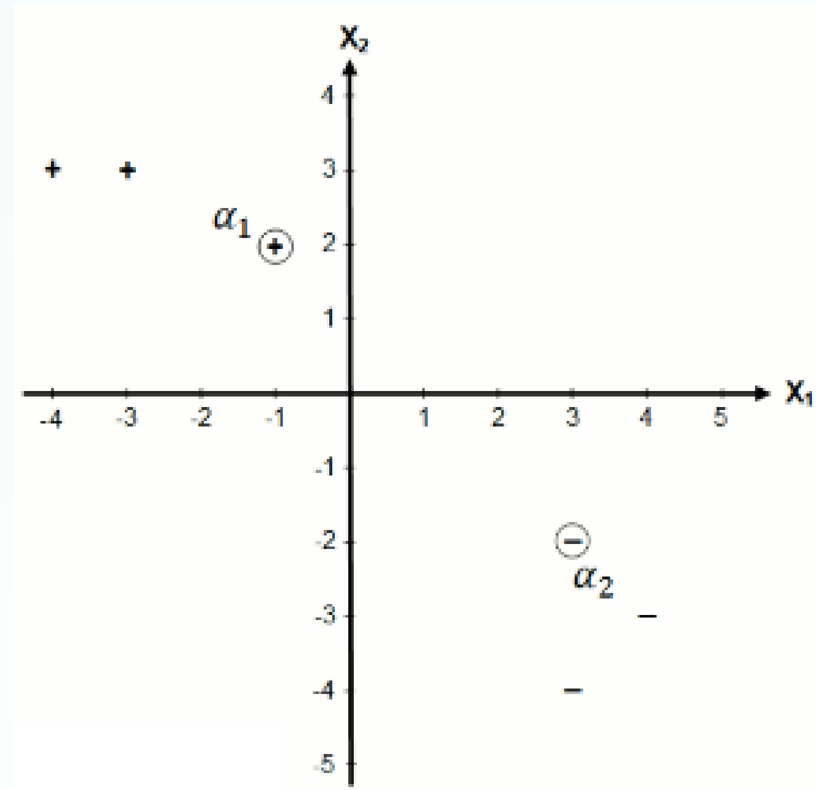
- Note that $\sum_{i=1}^{n} \alpha_i y_i = 0$
- This is a function of α_i only
- The problem space is convex, then it does not get stuck in local minimum/maximam

SVM remarks

- $\boldsymbol{w}^* = \sum_i \alpha_i y_i x_i$
- $w^* x^+ + b = 1 \quad \rightarrow \quad \boldsymbol{b}^* = 1 - \sum_i \alpha_i y_i x_i x^+$
- Classifying new sample $\boldsymbol{x_{test}}$ using $\boldsymbol{w}^*$ and $\boldsymbol{b}^*$
- $\sum_i \alpha_i y_i x_i + b^* \geq 0$ then Class is **+**

SVM example





$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_i \alpha_i \text{ , } (\boldsymbol{x_i} \text{ , } \boldsymbol{x_j} \text{ are SV } \& \ \alpha_i, a_j > 0)$$

$$L(\boldsymbol{\alpha}) = -\tfrac{1}{2}(\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}).(\alpha_1 \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix}) + \alpha_1 + \alpha_2 = -\tfrac{1}{2}(\alpha_1^2$$
$$+ 6\alpha_1\alpha_2 + 9\alpha_2^2 + 4\alpha_1^2 + 8\alpha_1\alpha_2 + 4\alpha_2^2) + \alpha_1 + \alpha_2 = -\tfrac{1}{2}(5\alpha_1^2 + 14\alpha_1\alpha_2$$
$$- 2\alpha_1 - 2\alpha_2)$$

SVM example

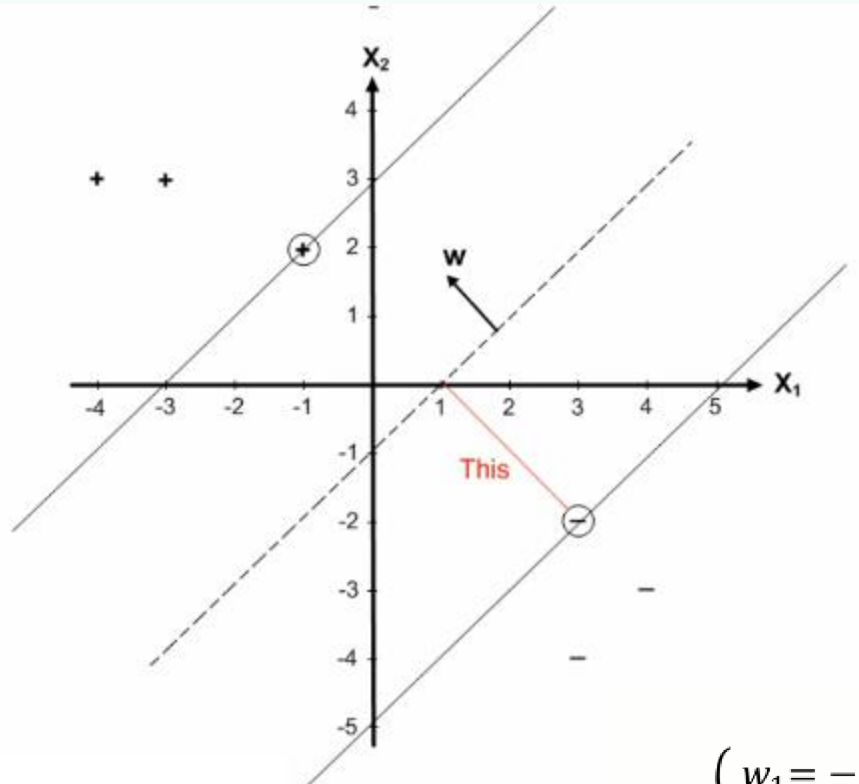
$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_i \alpha_i \text{ , } (\boldsymbol{x_i}, \boldsymbol{x_j} \text{ are SV \& } \alpha_i, a_j > 0)$$

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}(\alpha_1 \begin{bmatrix}-1\\2\end{bmatrix} - \alpha_2 \begin{bmatrix}3\\-2\end{bmatrix}).(\alpha_1 \begin{bmatrix}-1\\2\end{bmatrix} - \alpha_2 \begin{bmatrix}3\\-2\end{bmatrix}) + \alpha_1 + \alpha_2 = -\frac{1}{2}(\alpha_1^2 + 6\alpha_1\alpha_2 + 9\alpha_2^2 + 4\alpha_1^2 + 8\alpha_1\alpha_2 + 4\alpha_2^2) + \alpha_1 + \alpha_2 = -\frac{1}{2}(5\alpha_1^2 + 14\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2)$$

$$\Rightarrow \begin{cases} \dfrac{\partial L(\boldsymbol{\alpha})}{\partial \alpha_1} = 10\alpha_1 + 14\alpha_2 - 2 = 0 \\ \dfrac{\partial L(\boldsymbol{\alpha})}{\partial \alpha_2} = 14\alpha_1 + 26\alpha_2 - 2 = 0 \\ \displaystyle\sum_i \alpha_i y_i = 0 \end{cases} \Rightarrow \begin{cases} 6\alpha_1 + 10\alpha_2 = 1 \\ \alpha_1 = \alpha_2 \end{cases} \Rightarrow \boxed{\alpha_1 = \alpha_2 = \frac{1}{16}}$$

$$\begin{cases} \boldsymbol{w} = \displaystyle\sum_i \alpha_i y_i \boldsymbol{x_i} \rightarrow \boldsymbol{w} = \frac{1}{16}\begin{bmatrix}-1\\2\end{bmatrix} - \frac{1}{16}\begin{bmatrix}3\\-2\end{bmatrix} \rightarrow \boxed{\boldsymbol{w} = \begin{bmatrix}-1/4\\1/4\end{bmatrix}} \\ y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1 = 0 \rightarrow \begin{bmatrix}-\frac{1}{4} & \frac{1}{4}\end{bmatrix}\begin{bmatrix}-1\\2\end{bmatrix} + b - 1 = 0 \rightarrow \boxed{b = 1/4} \end{cases}$$

Verification



$\boldsymbol{w}.\boldsymbol{x} + b = 0 \;\rightarrow w_1x_1 + w_2x_2 + b = 0$

Standard equation of a line: $\boxed{x_2 = -\frac{w_1}{w_2}x_1 - \frac{b}{w_2}}$

The separator line equation: $x_2 = x_1 - 1$

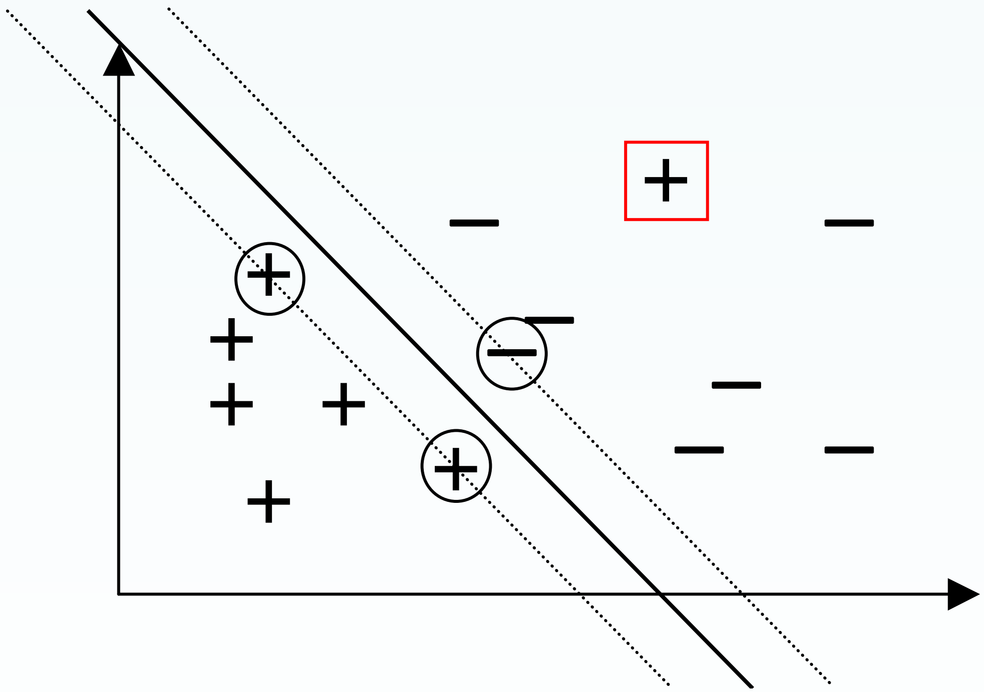
$$\begin{cases} x_2 = -\frac{w_1}{w_2}x_1 - \frac{b}{w_2} \\ x_2 = x_1 - 1 \end{cases} \Rightarrow \frac{w_1}{w_2} = -1\,,\; \frac{b}{w_2} = 1$$

This$= 2\sqrt{2} = \frac{1}{||\boldsymbol{w}||} \Rightarrow ||\boldsymbol{w}|| = \frac{\sqrt{2}}{4} = \sqrt{w_1^2 + w_2^2}$

$$\begin{cases} w_1 = -k \\ w_2 = k \\ b = k \end{cases} (k > 0) \Rightarrow \begin{cases} \sqrt{w_1^2 + w_2^2} = \frac{\sqrt{2}}{4} \\ \sqrt{w_1^2 + w_2^2} = \sqrt{2}\,k \end{cases} \Rightarrow \begin{cases} w_1 = -\frac{1}{4} \\ w_2 = \frac{1}{4} \\ b = \frac{1}{4} \end{cases}$$

$$\begin{cases} \alpha_+ = \alpha_- \\ \begin{bmatrix} -1/4 \\ 1/4 \end{bmatrix} = \alpha_+ \begin{bmatrix} -1 \\ 2 \end{bmatrix} - \alpha_- \begin{bmatrix} 3 \\ -2 \end{bmatrix} \end{cases} \Rightarrow \boxed{\alpha_+ = \alpha_- = \frac{1}{16}}$$

What if the data is not linearly separable?



- Hyperplane with the largest margin
- A hyperplane that correctly separates many instances as possible

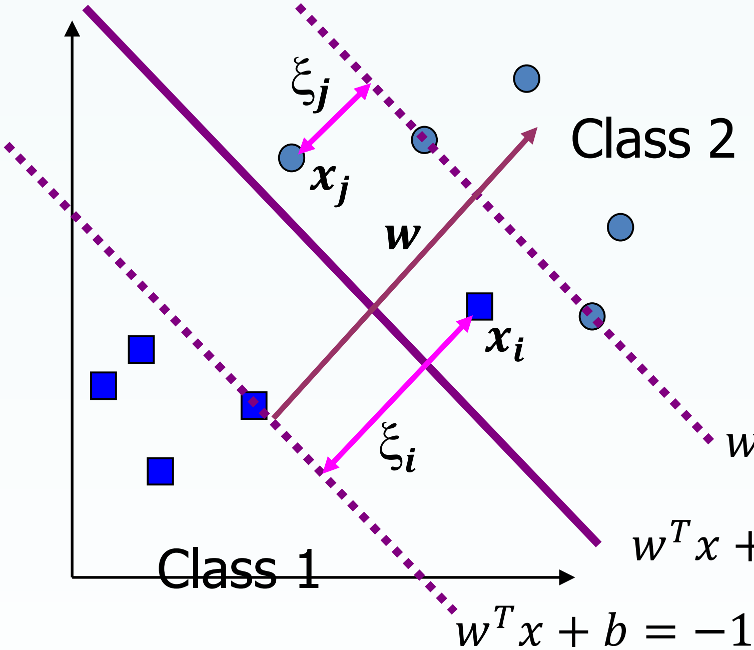
Soft Margin Hyperplane with slack variables

$$w^T x + b \geq +1$$

$$w^T x + b \geq 1 - \xi_i \quad \boldsymbol{\xi_i \geq 0}$$

$$w^T x + b \leq -1 + \xi_i \quad \boldsymbol{\xi_i \geq 0}$$

ξ_i are "slack variables" in optimization



Minimize $\frac{1}{2}||w||^2 + C \sum_i \xi_i^2$ $\qquad \boldsymbol{\xi_i \geq 0, C > 0}$

Subject to $\boldsymbol{y_i}(w^T x + b) \geq \boldsymbol{1} - \xi_i \quad \boldsymbol{\xi_i \geq 0}$

C : tradeoff parameter between error and margin (undrfitting vs. overfitting)

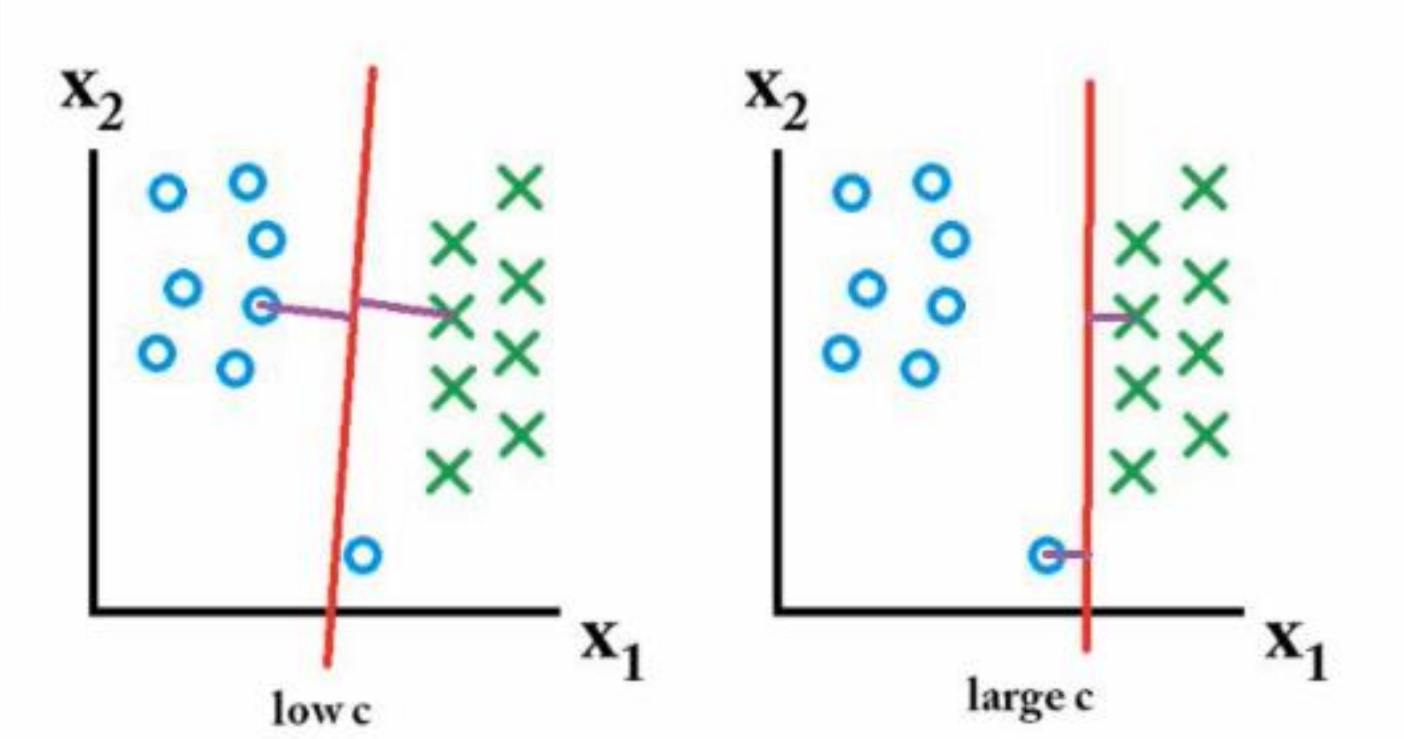
Maximize: $\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

S.t. $\sum_{i=1}^{N} \alpha_i y_i = 0,$

$0 \leq \alpha_i \leq C, \quad \forall i$

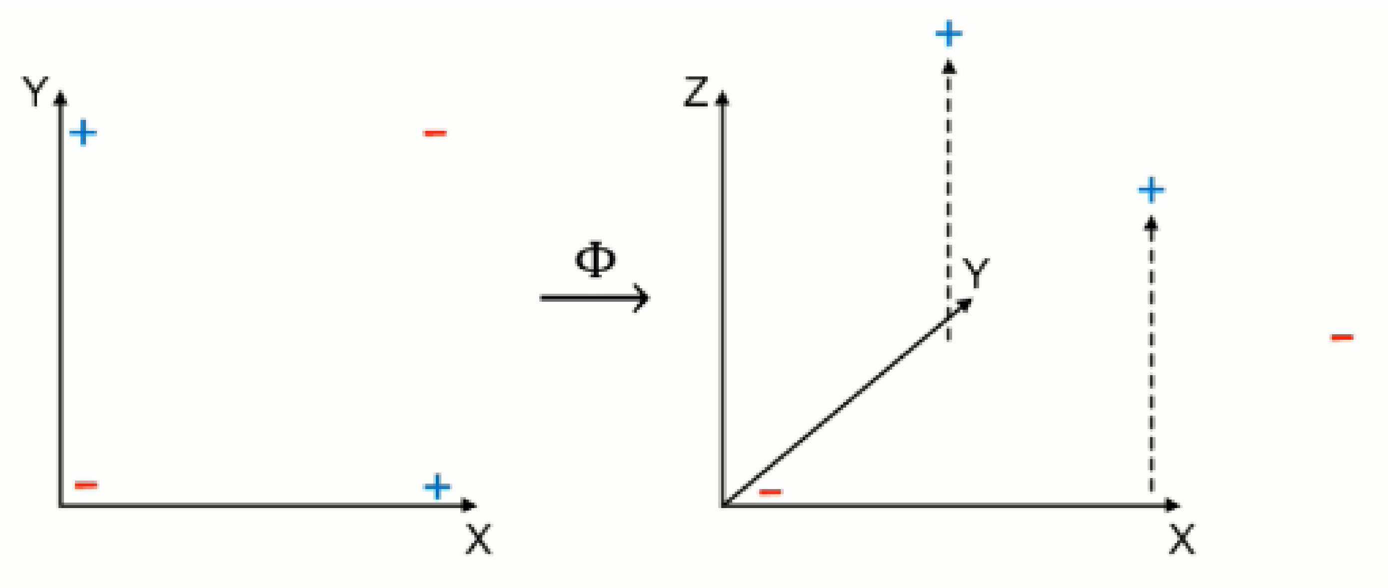
Soft margin SVM

- The effect of the C parameter on the margin



Kernel trick intuition

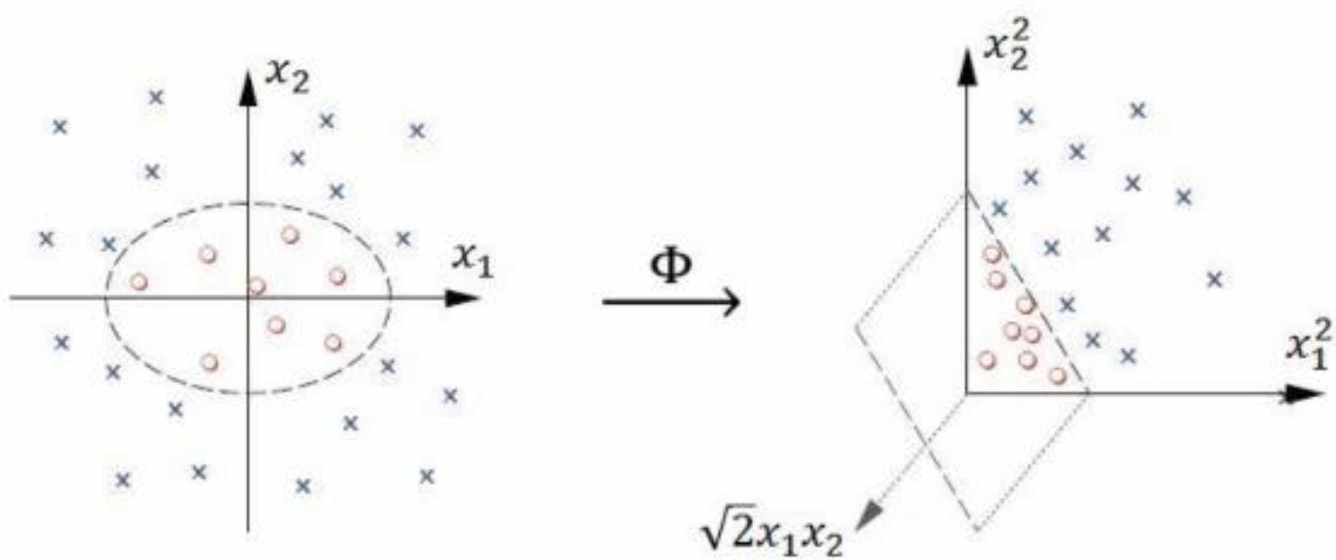
- SVM solution for linearly inseparable problems, such as XOR
  - Kernel trick: using a linear classifier to solve a non-linear problem



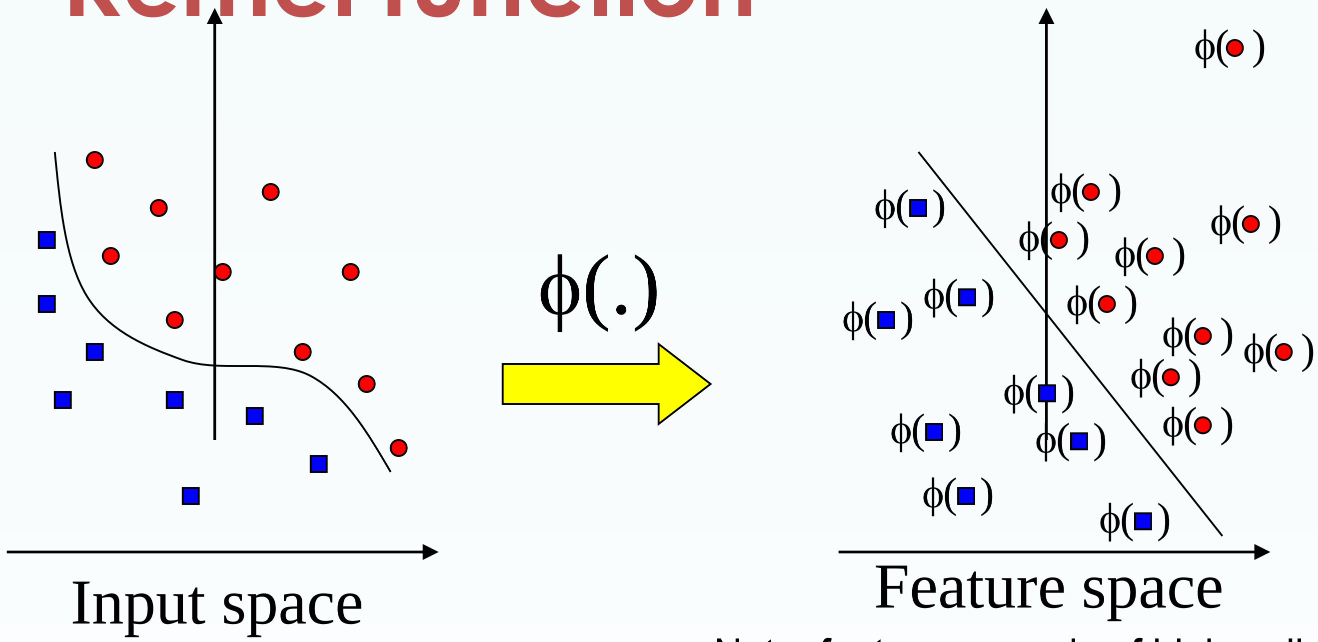
Kernel trick (cont.)

- Higher dimensional feature space

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$



Kernel function



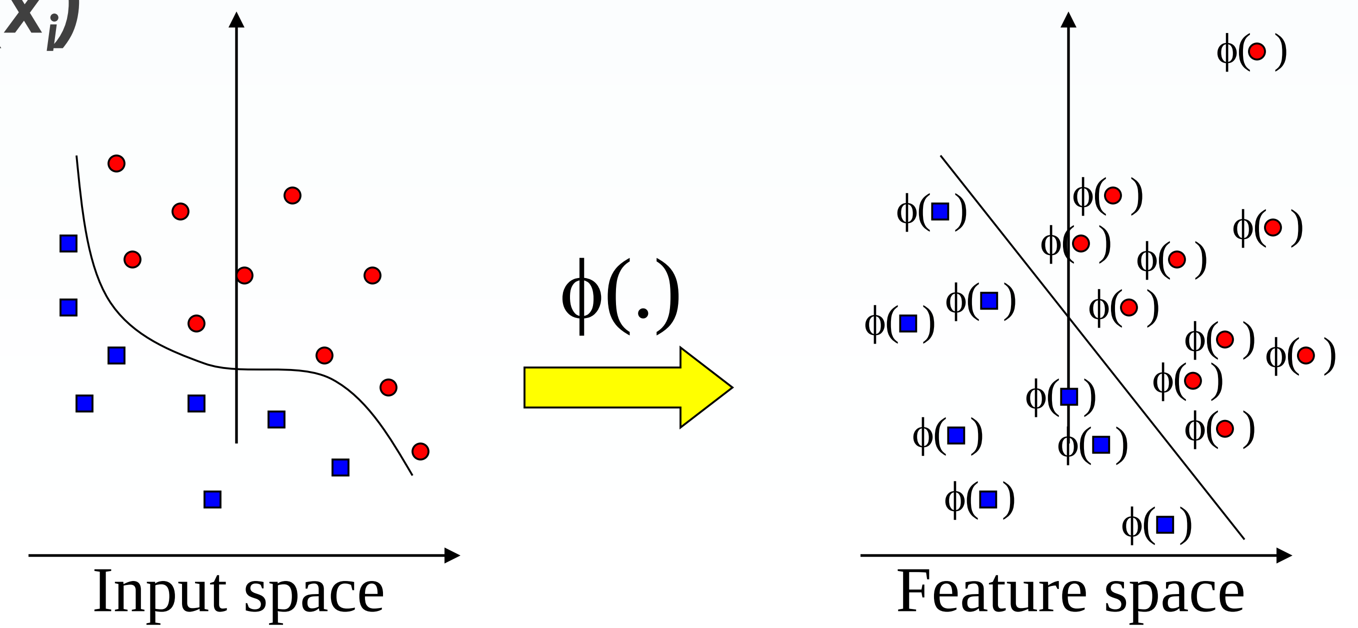
Note: feature space is of higher dimension than the input space in practice

$$\begin{cases} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} + \sum_i \alpha_i \\ h(\boldsymbol{x_{test}}) = sgn(\sum_i \alpha_i y_i \boldsymbol{x_i}.\boldsymbol{x_{test}} + b) \end{cases}$$

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{\Phi}(\boldsymbol{x_i}).\boldsymbol{\Phi}(\boldsymbol{x_j}) + \sum_i \alpha_i$$

Kernel

- Transform $\mathbf{x} \rightarrow \phi(\mathbf{x})$
- The linear algorithm depends only on $\mathbf{x}\mathbf{x}_i$, hence transformed algorithm depends only on $\phi(\mathbf{x})\phi(\mathbf{x}_i)$
- Use kernel function $K(x_i, x_j)$ such that $K(x_i, x_j) = \phi(x)\,\phi(x_j)$

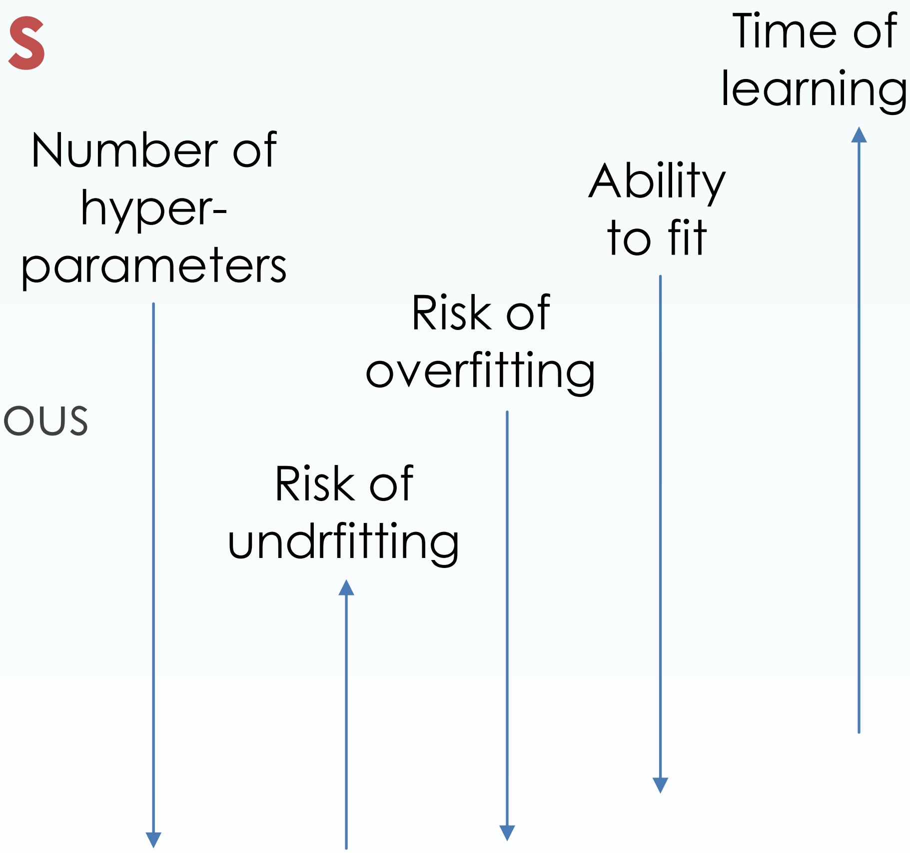


Kernel function properties

Function $K(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel if:

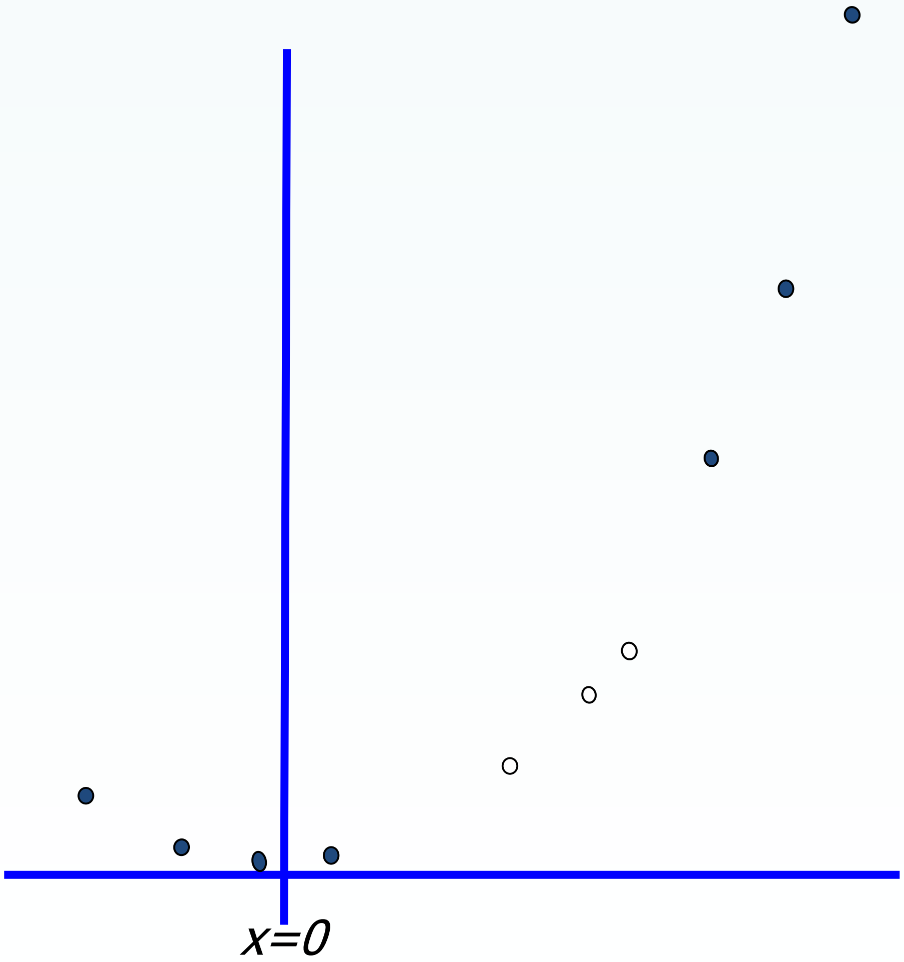
- It computes an inner product in <u>some</u> space $\mathbb{Z}$.
  - We just need to know that space $\mathbb{Z}$ exists!
- It is symmetric / commutative, i.e. $K(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}', \boldsymbol{x})$.
- It should (preferably) be positive semi-definite, i.e. satisfy Mercer's theorem.

Kernel functions



- Linear
  - $K(x, x') = (x.x' + c)$
    - For c=0, it is homogenous
- Polynomial
  - $K(x, x') = (ax.x' + c)^d$
    - $d > 1$
- Gaussian RBF
  - $K(x, x') = exp\left(-\frac{||x-x'||^2}{2\sigma^2}\right) = exp(-\gamma||x - x'||^2)$
    - $\gamma = \frac{1}{2\sigma^2}$
- Sigmoid function (Hyperbolic tangent)
  - $K(x, x') = tanh(kx, x' + c)$

Harder 1-dimensional dataset

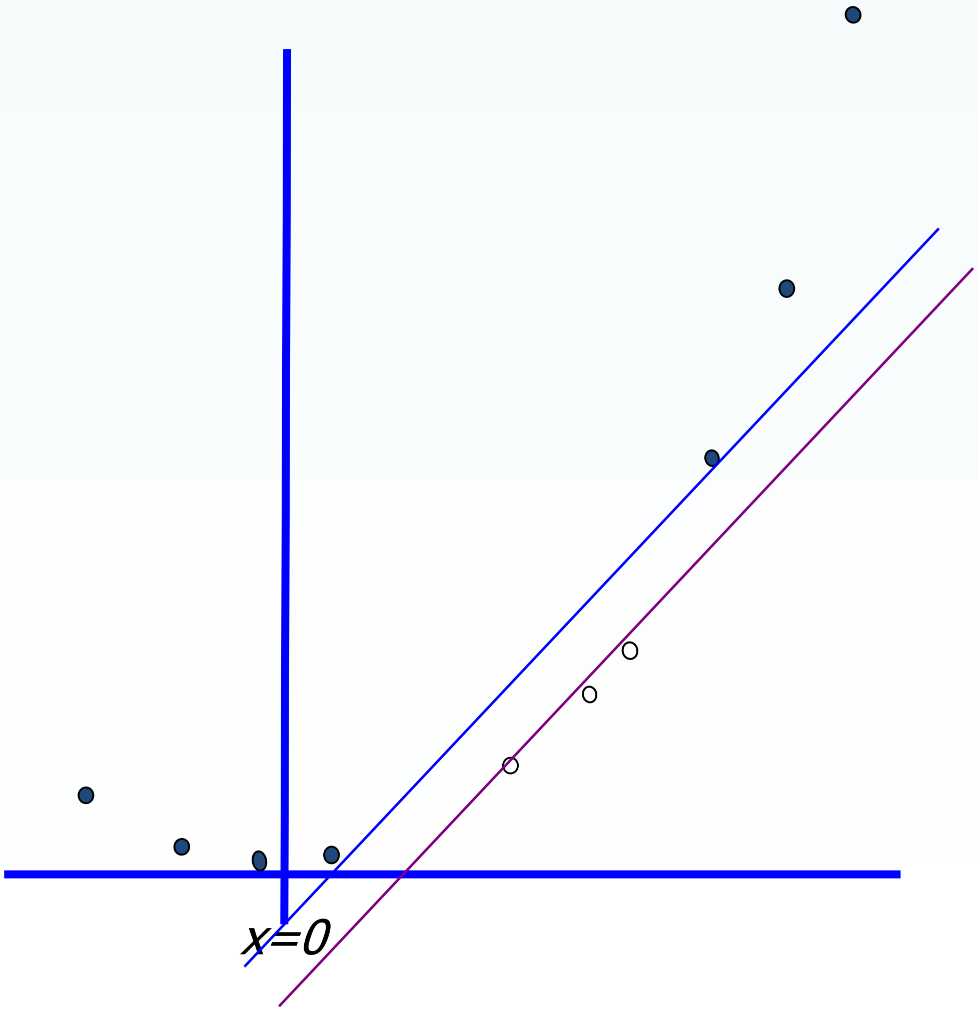


Remember how permitting non-linear basis functions made linear regression so much nicer?

Let's permit them here too

$$\mathbf{z}_k = (x_k, x_k^2)$$

Harder 1-dimensional dataset



Remember how permitting non-linear basis functions made linear regression so much nicer?

Let's permit them here too

$$\mathbf{z}_k = (x_k, x_k^2)$$

Kernel rules

- If K, K' are kernels, then:
  - K+K' is a kernel
  - cK is a kernel, if c>0
  - aK+bK' is a kernel, for a,b >0
  - Etc etc etc……

Multiclass SVMs (one-versus rest)

- SVM is a two-class classifier
- Several suggested methods for combining multiple two-class classifiers
- Most used approach: one versus rest
  - Also recommended by Vapnik
  - using data from class C_k as the positive examples and data from the remaining k-1 classes as negative examples
- Disadvantages
  - Input assigned to multiple classes simultaneously
  - Training sets are imbalanced (90% are one class and 10% are another)– symmetry of original problem is lost

Multiclass SVMs (one-versus one)

- Train k(k-1)/2 different 2-class SVMs on all possible pairs of classes
- Classify test points according to which class has highest number of votes
- Again leads to ambiguities in classification
- For large k requires significantly more training time than one-versus rest
  - Also more computation time for evaluation
    - Can be alleviated by organizing into a directed acyclic graph (DAGSVM)

SVM

- Strengths
  - Training is relatively easy
  - Good generalization in theory and practice
  - Work well with few training instances
  - Find globally best model, No local optimal, unlike in neural networks
  - It scales relatively well to high dimensional data
  - Tradeoff between classifier complexity and error can be controlled explicitly
- Weaknesses
  - Need to choose a “good” kernel function.

Conclusion

- SVMs find optimal linear separator
  - They pick the hyperplane that maximises the margin
  - The optimal hyperplane turns out to be a linear combination of support vectors
- The kernel trick makes SVMs non-linear learning algorithms
  - Transform nonlinear problems to higher dimensional space using kernel functions; then there is more chance that in the transformed space the classes will be linearly separable.

Reading

- C. M. Bishop, **Pattern recognition and machine learning**, Springer, 2006. (ch. 7)
- E. Alpaydin, **Introduction to Machine Learning**, 3rd ed., The MIT Press, 2014. (ch. 13)
- C. Cortes, V. Vapnik. **Support-vector networks**, Machine Learning, (20) 273-297, 1995.
- V. Vapnik. **The nature of statistical learning theory**. 2nd ed., Springer, 1999.