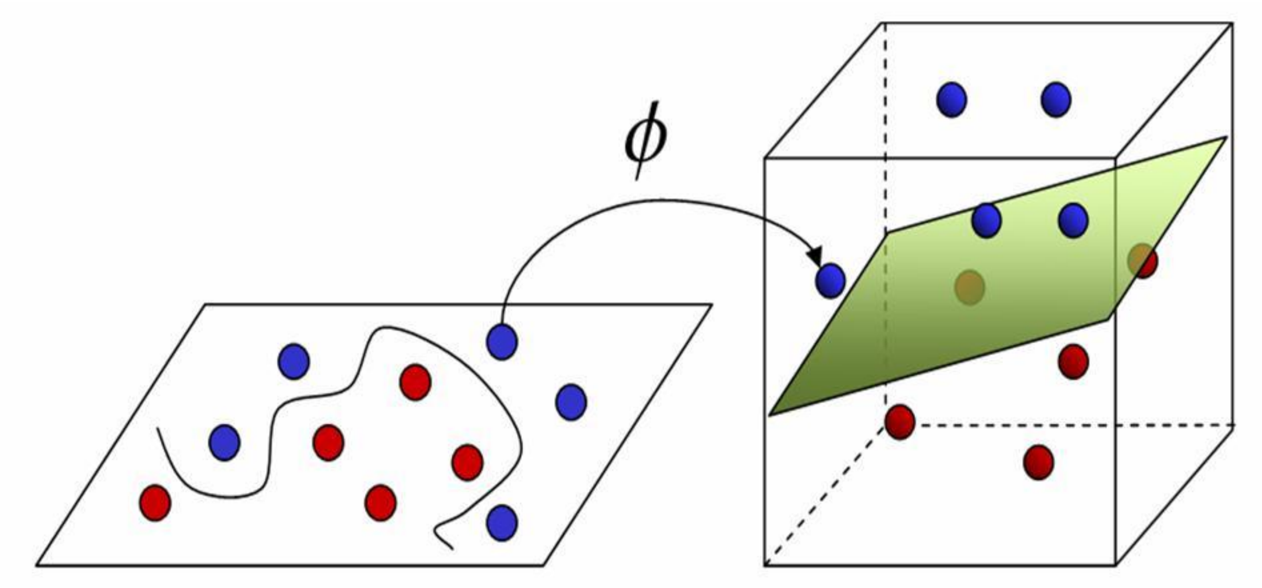


Machine Learning: The Instance-Based Perspective

Konstantin Tretyakov
http://kt.era.ee

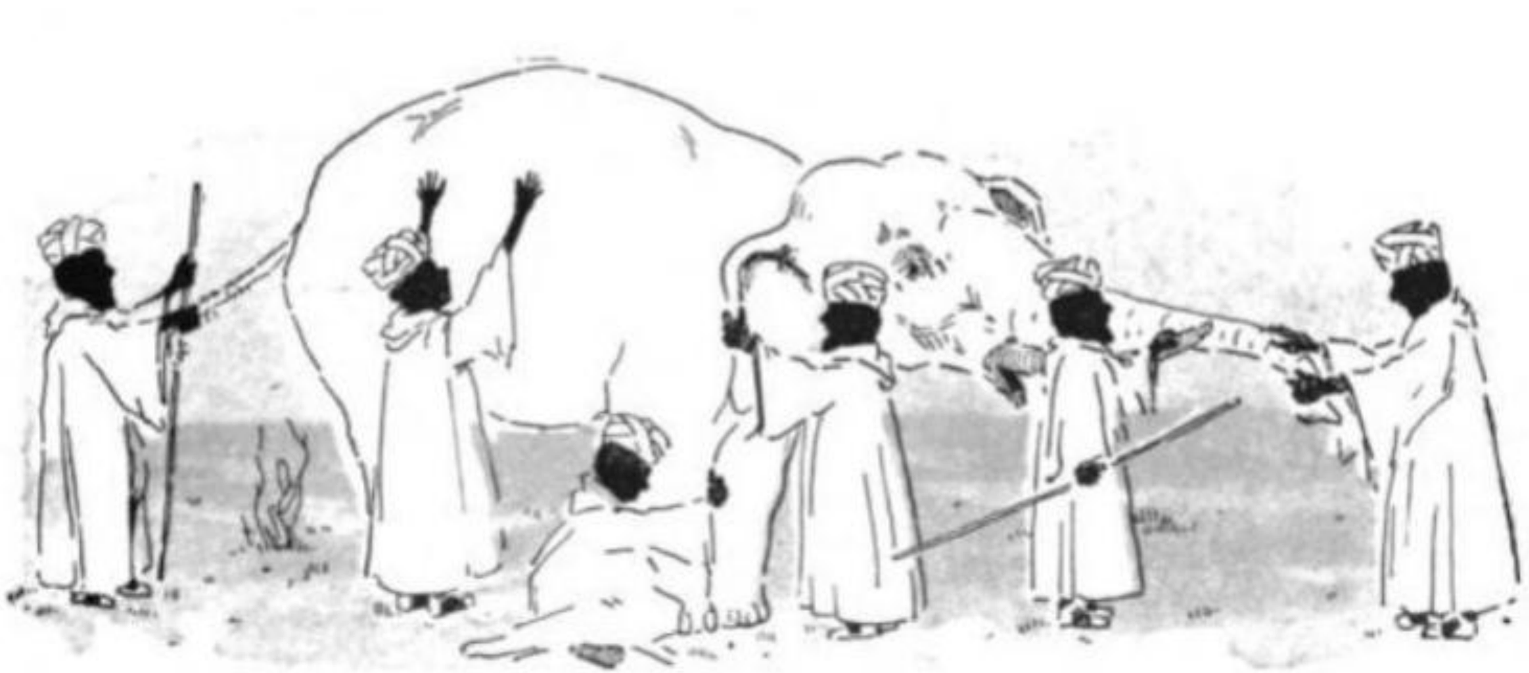
AACIMP 2012
August 2012

STACC Software Technology and Applications Competence Center

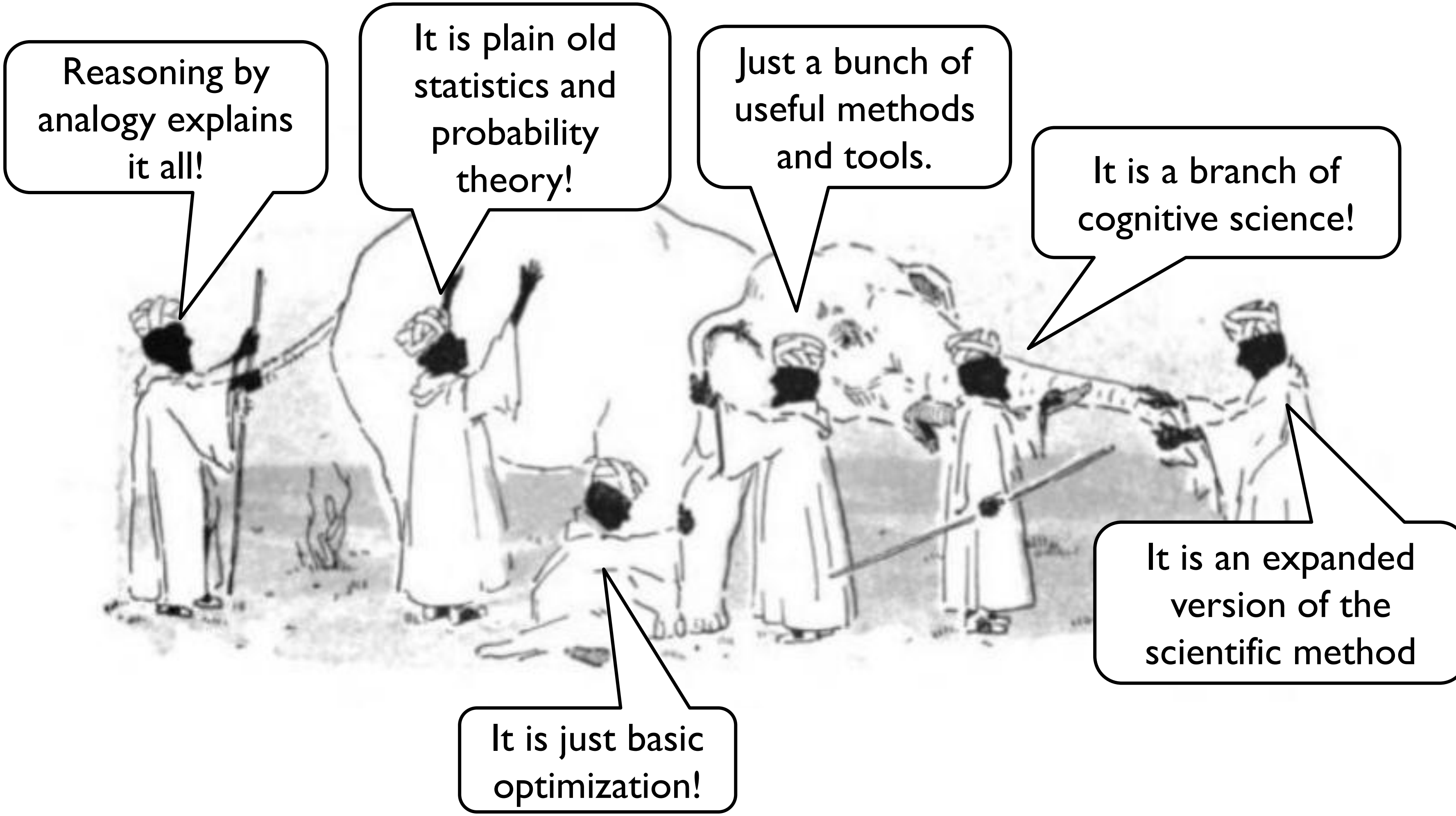
BIIT

TARTU ÜLIKOOL · UNIVERSITAS TARTUENSIS · 1632

So far…

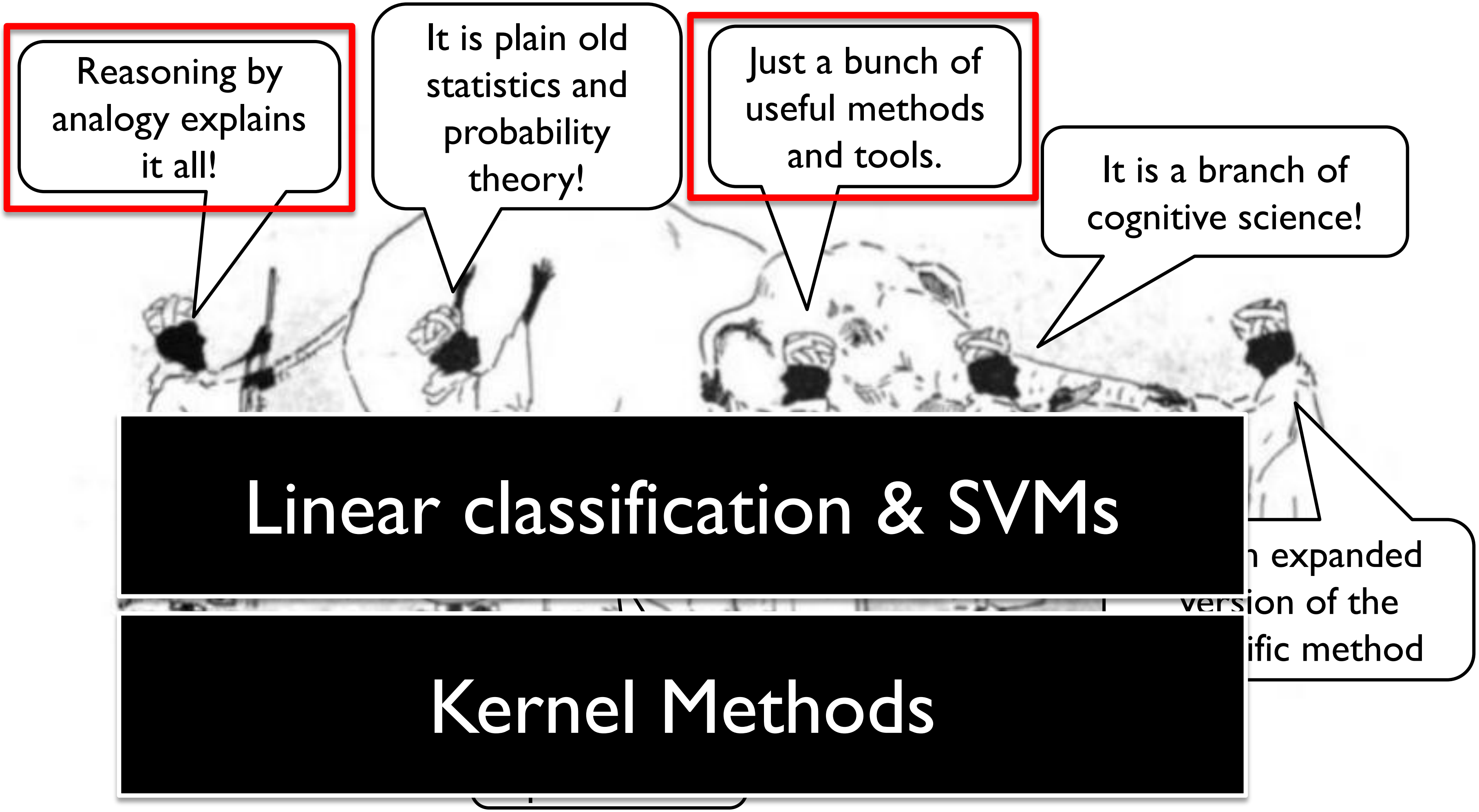


So far…



Reasoning by analogy explains it all!
It is plain old statistics and probability theory!
Just a bunch of useful methods and tools.
It is a branch of cognitive science!
It is an expanded version of the scientific method
It is just basic optimization!

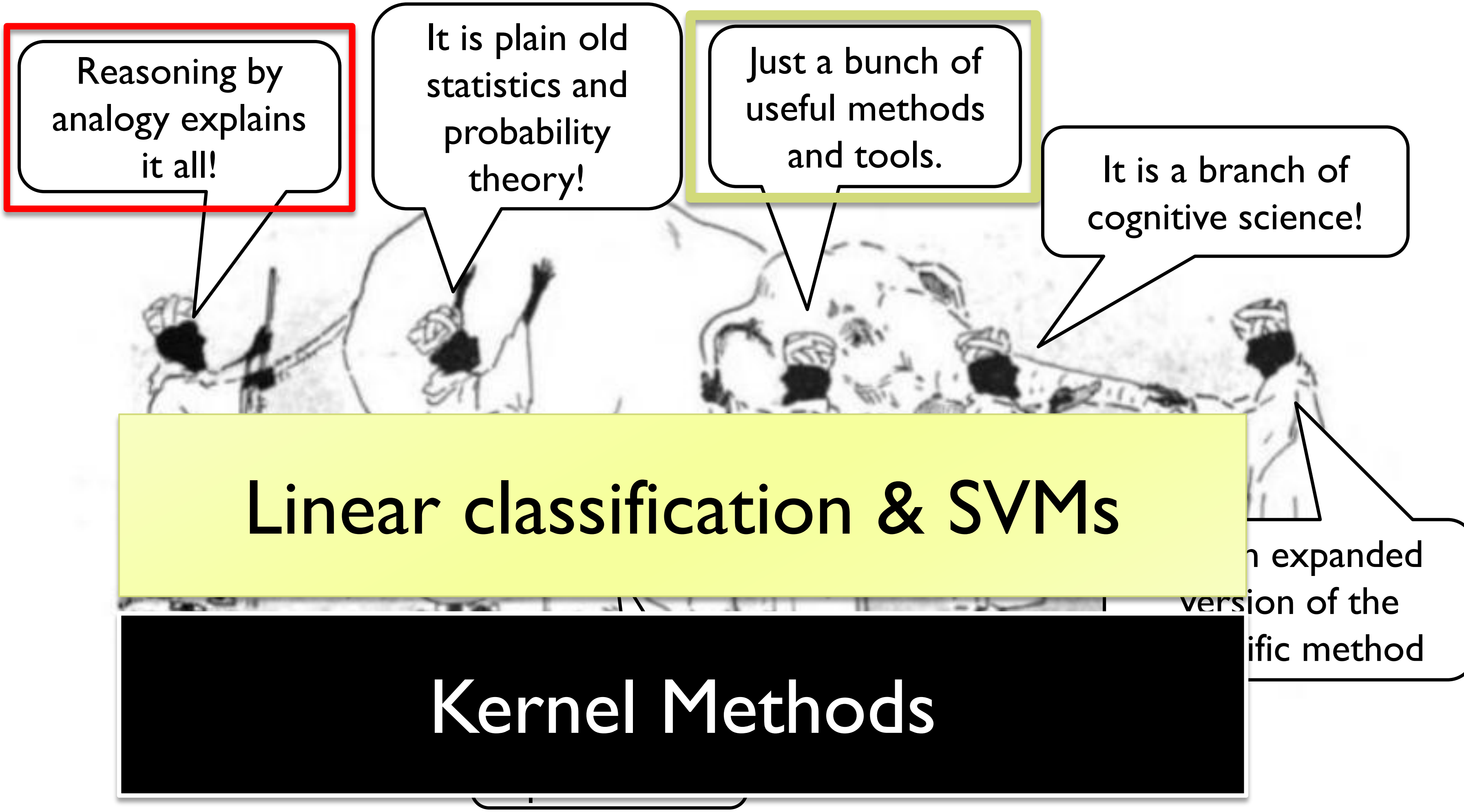
Today



Linear classification & SVMs

Kernel Methods

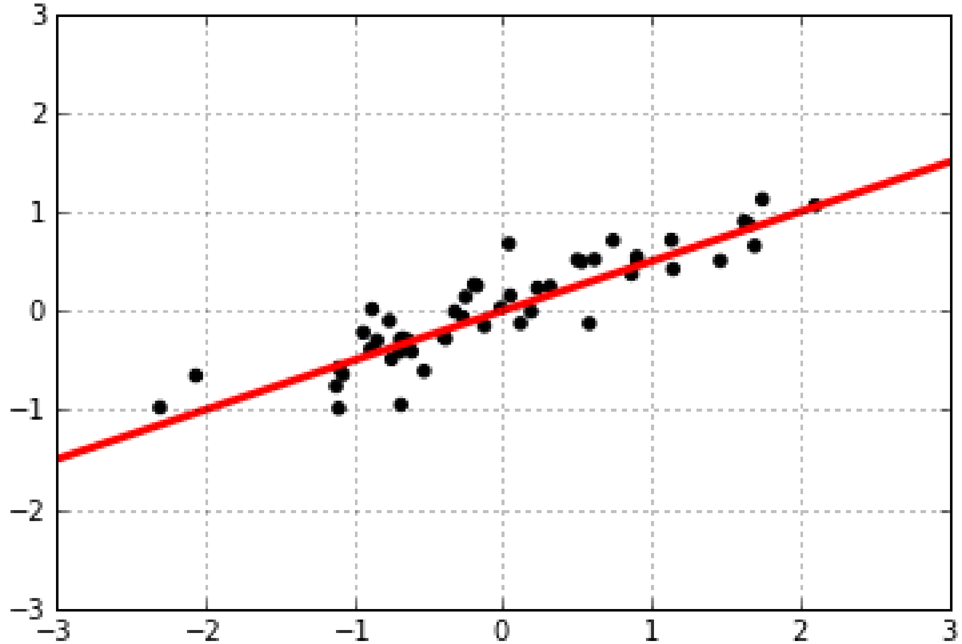
Next



Linear classification & SVMs

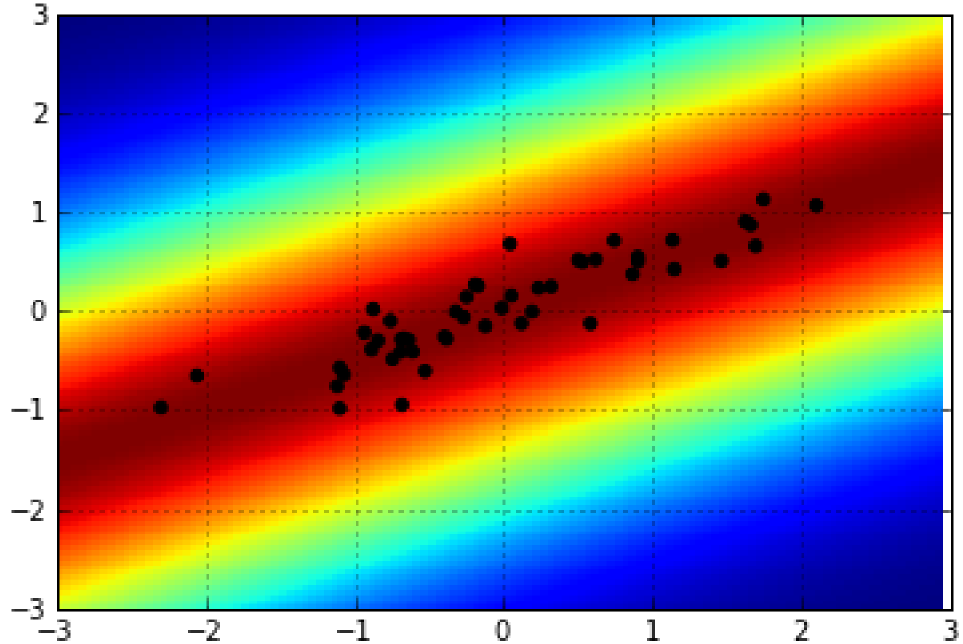
Kernel Methods

Recall linear regression



$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x}$$

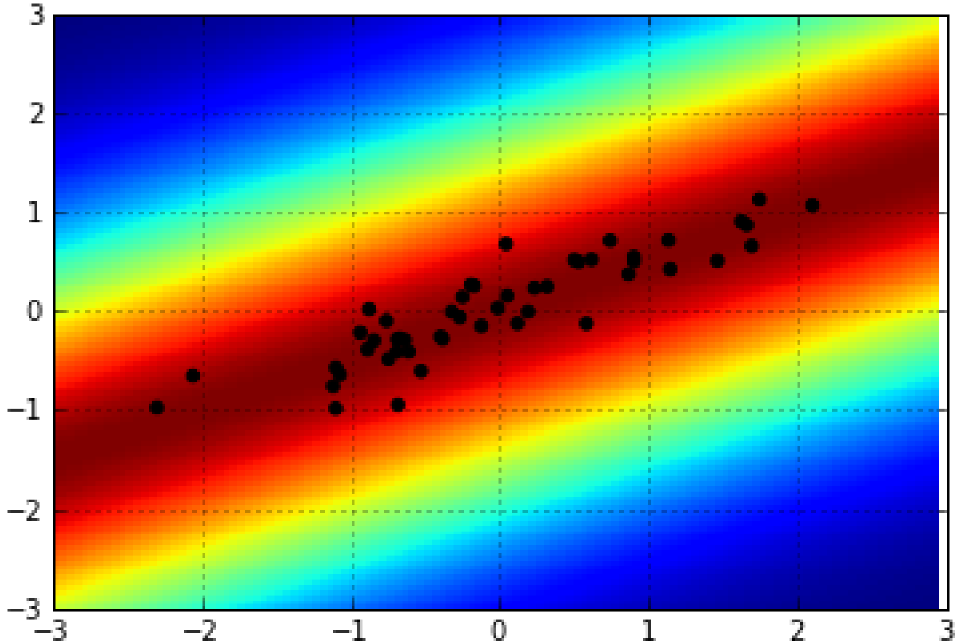
Recall linear regression



$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x}$$

$$E(y|\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Recall linear regression

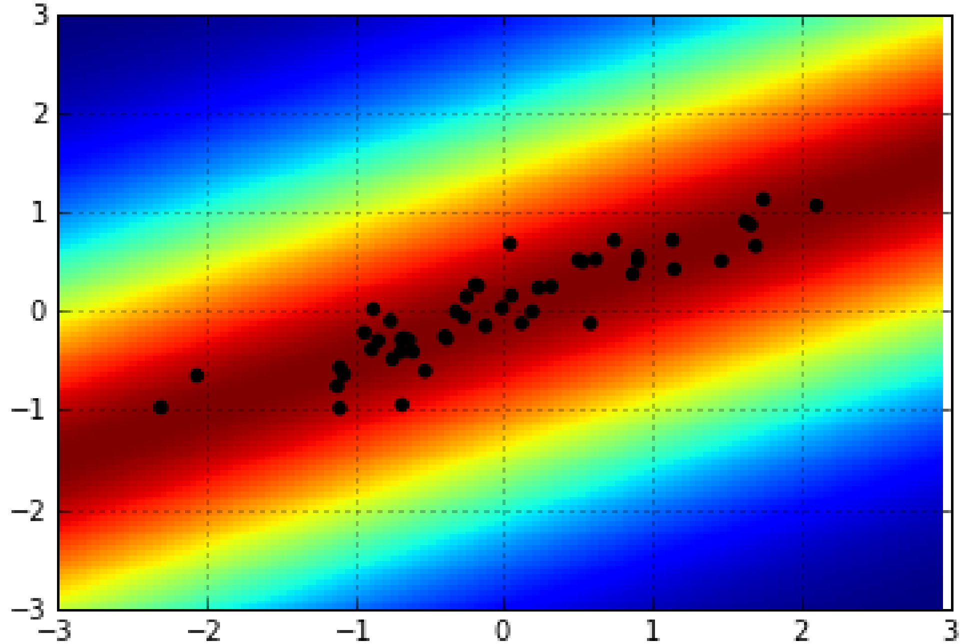


$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x}$$

$$E(y|\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

$$Y \sim \boldsymbol{w}^T \boldsymbol{x} + N(0, \sigma^2)$$

Recall linear regression



$$\hat{y} = \boldsymbol{w}^T\boldsymbol{x}$$

$$E(y|x) = \boldsymbol{w}^T\boldsymbol{x}$$

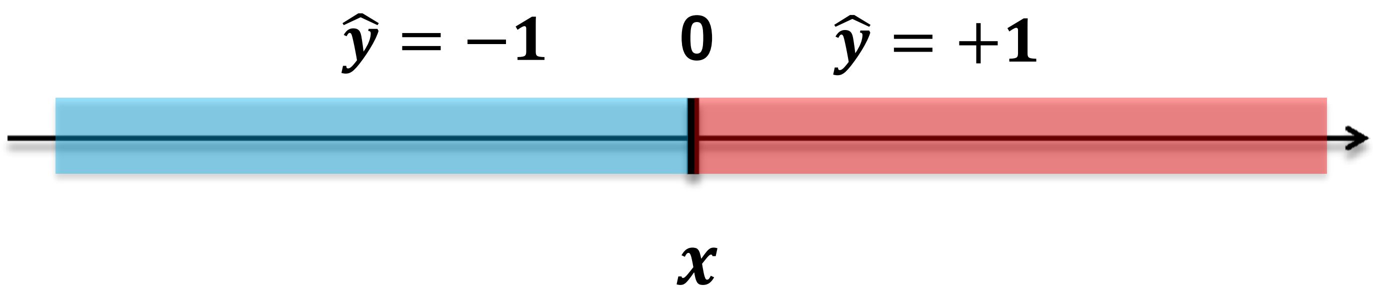
$$Y \sim N(\boldsymbol{w}^T\boldsymbol{x}, \sigma^2)$$

Linear classification

$$\hat{y} = \operatorname{sign}(\boldsymbol{w}^T \boldsymbol{x})$$

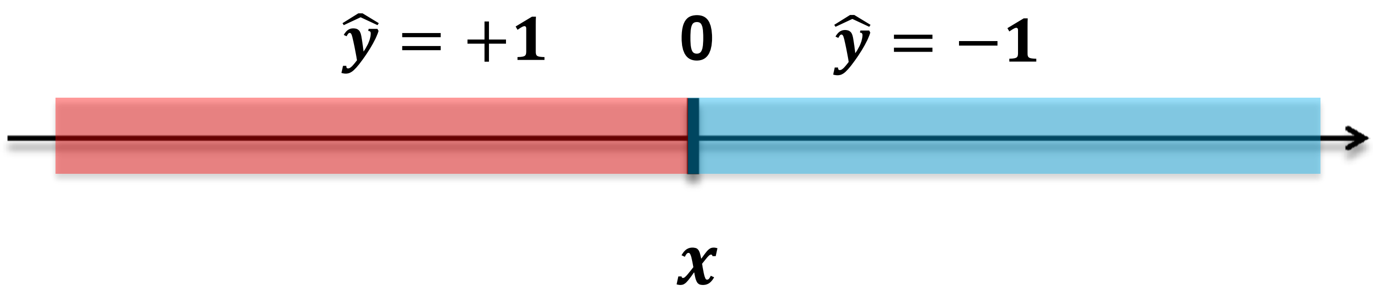
Linear classification

$$\hat{y} = \text{sign}(5 \cdot x)$$



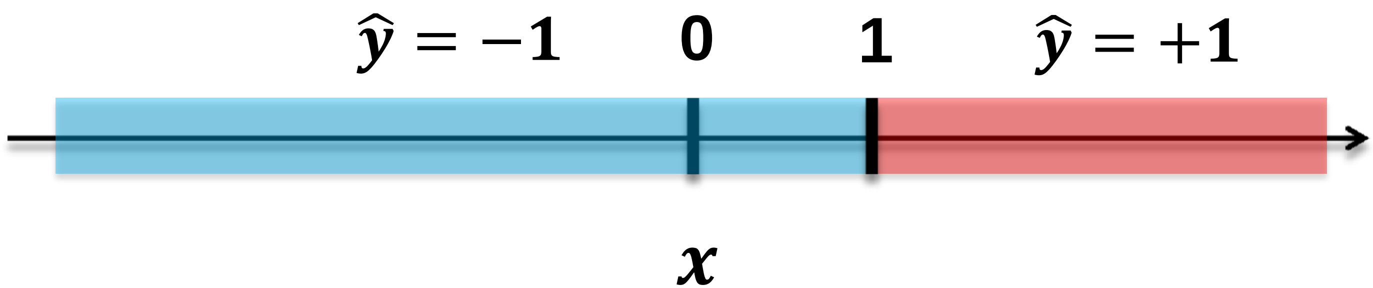
Linear classification

$$\hat{y} = \operatorname{sign}(-5 \cdot x)$$



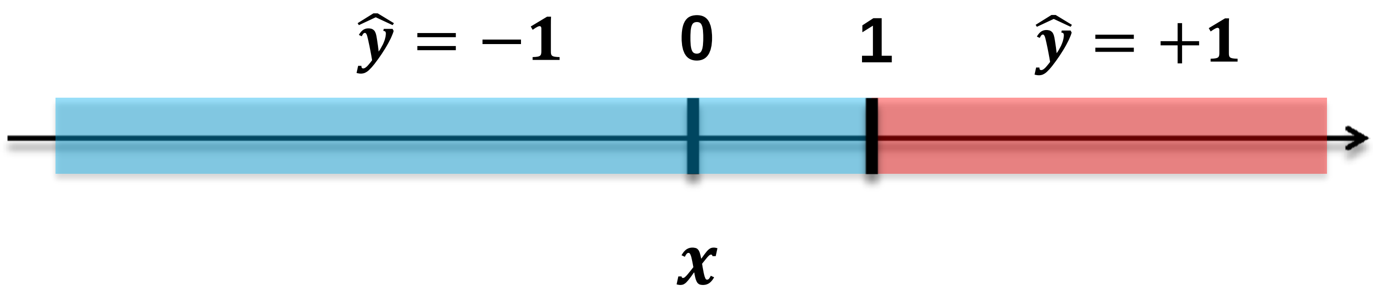
Linear classification

$$\hat{y} = \text{sign}(5 \cdot x - 5)$$



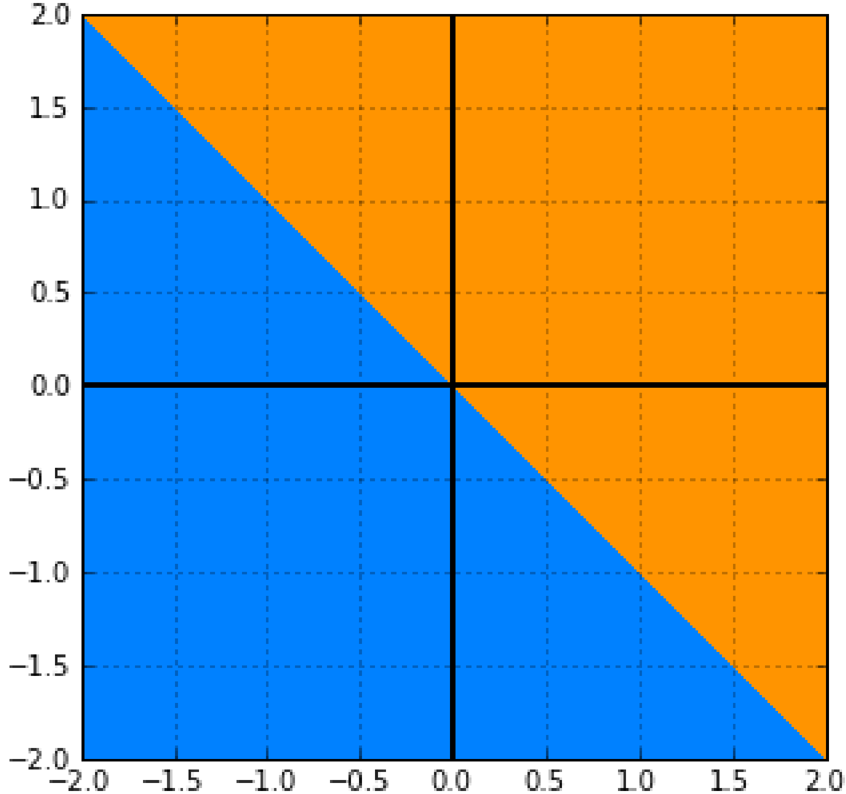
Linear classification

$$\hat{y} = \text{sign}(5 \cdot x - 5)$$
$$\hat{y} = \text{sign}(5 \cdot (x - 1))$$



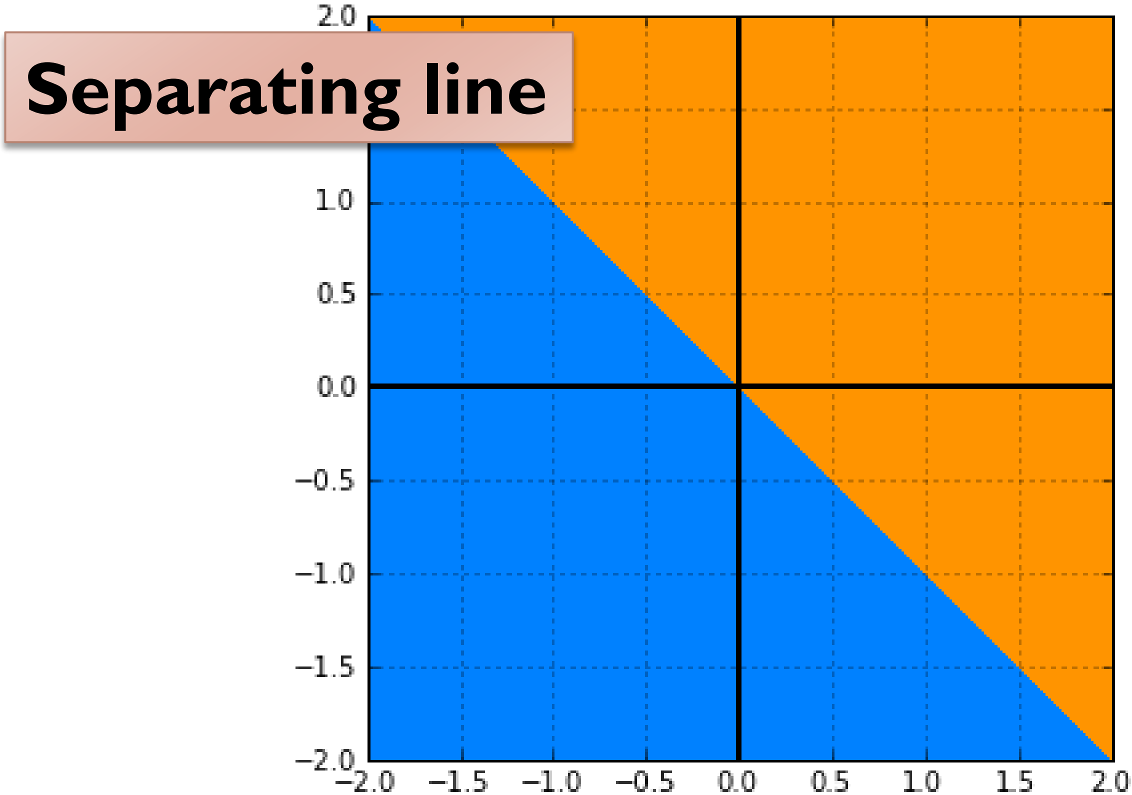
Linear classification

$$\hat{y} = \mathrm{sign}(0.5x_1 + 0.5x_2)$$



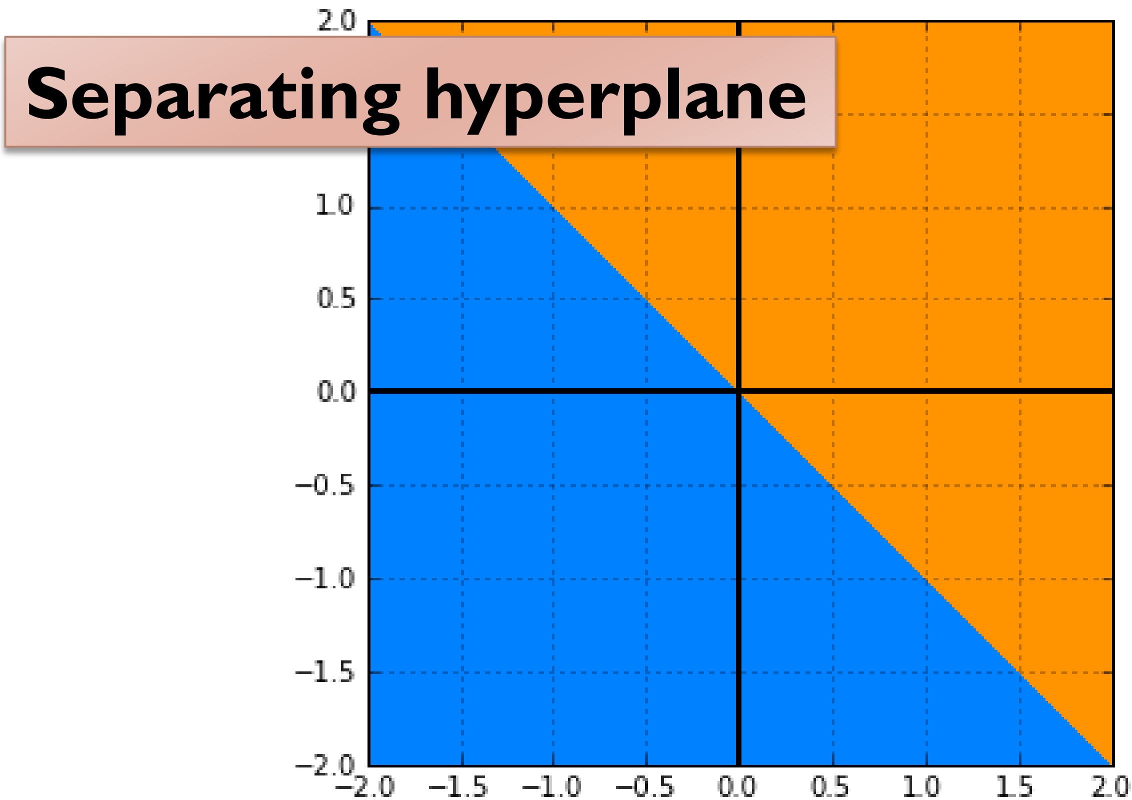
Linear classification

$$\hat{y} = \text{sign}(0.5x_1 + 0.5x_2)$$



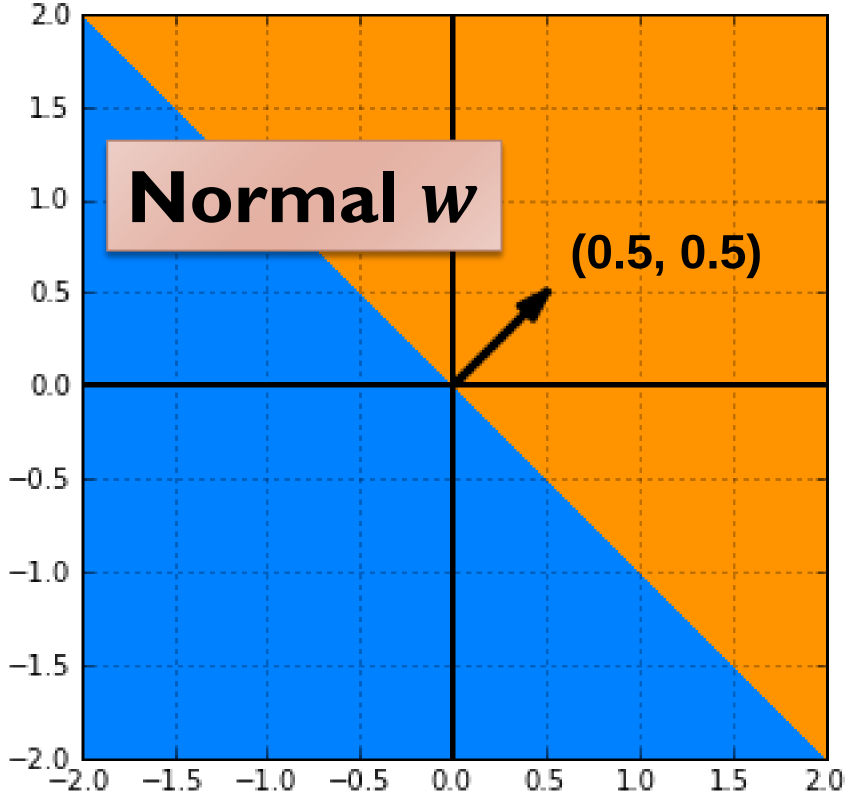
Linear classification

$$\hat{y} = \mathrm{sign}(0.5x_1 + 0.5x_2)$$



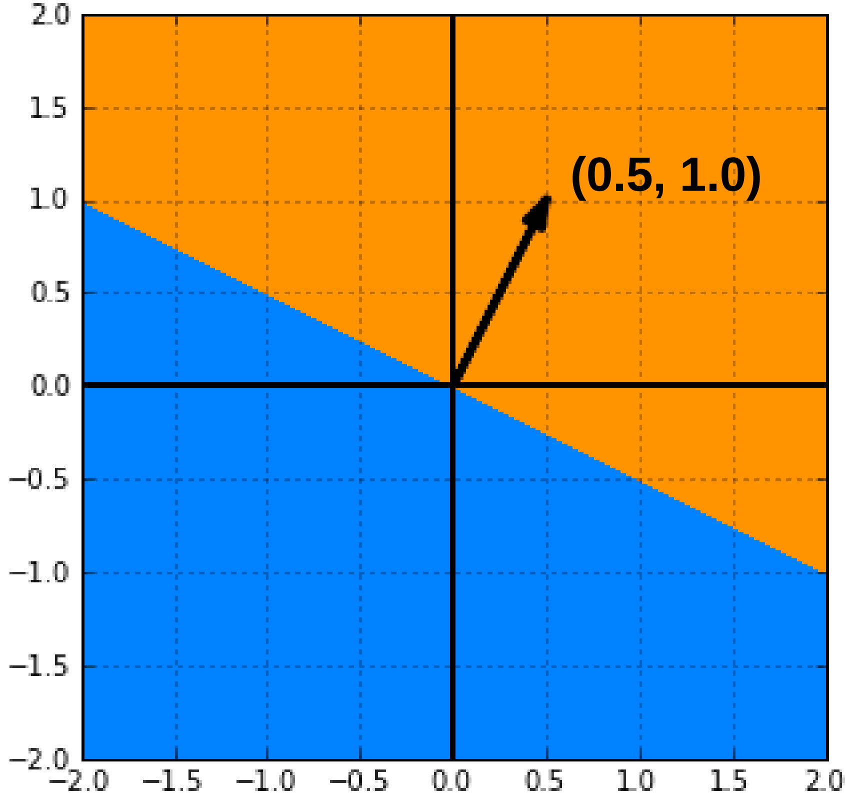
Linear classification

$$\hat{y} = \text{sign}(0.5x_1 + 0.5x_2)$$



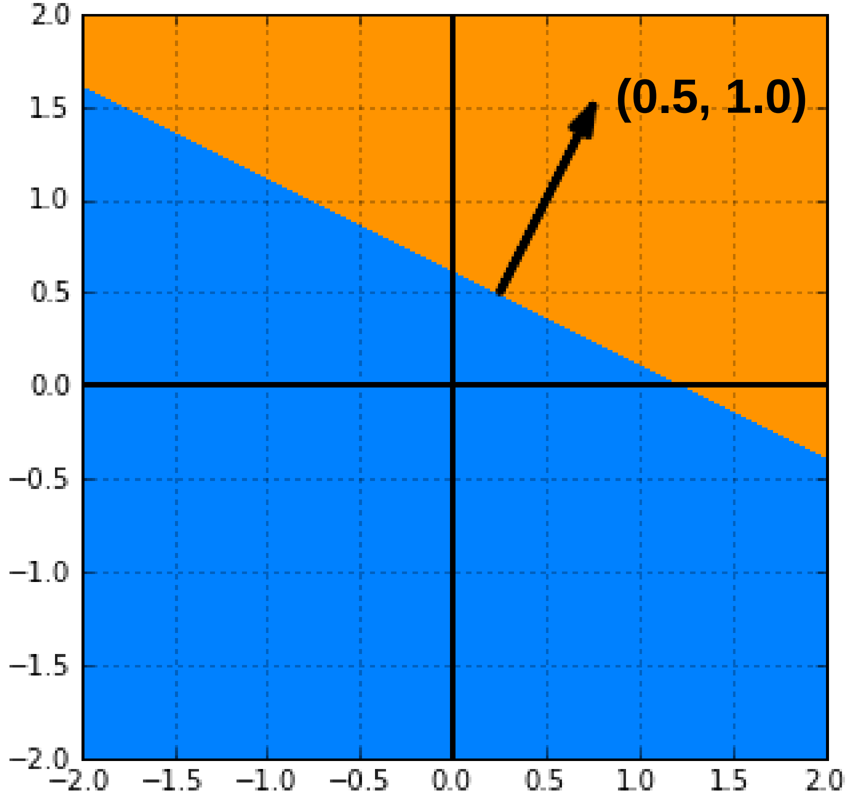
Linear classification

$$\hat{y} = \text{sign}(0.5x_1 + 1.0x_2)$$



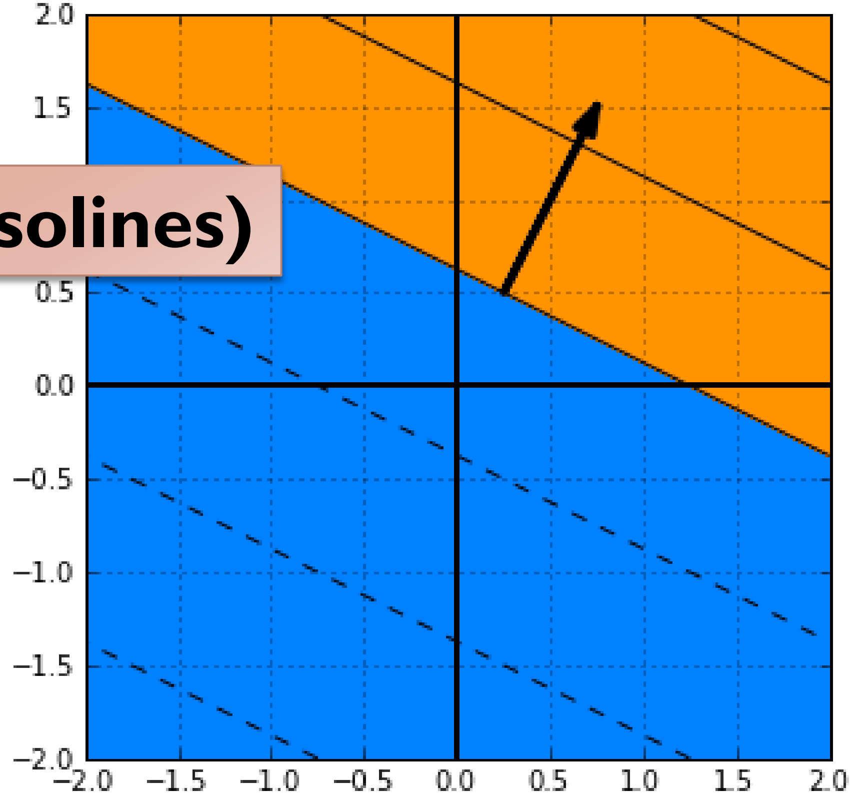
Linear classification

$$\hat{y} = \mathrm{sign}(0.5x_1 + 1.0x_2 - 0.625)$$



Linear classification

$$\hat{y} = \text{sign}(0.5x_1 + 1.0x_2 - 0.625)$$



$f = 2$

Level lines (isolines)

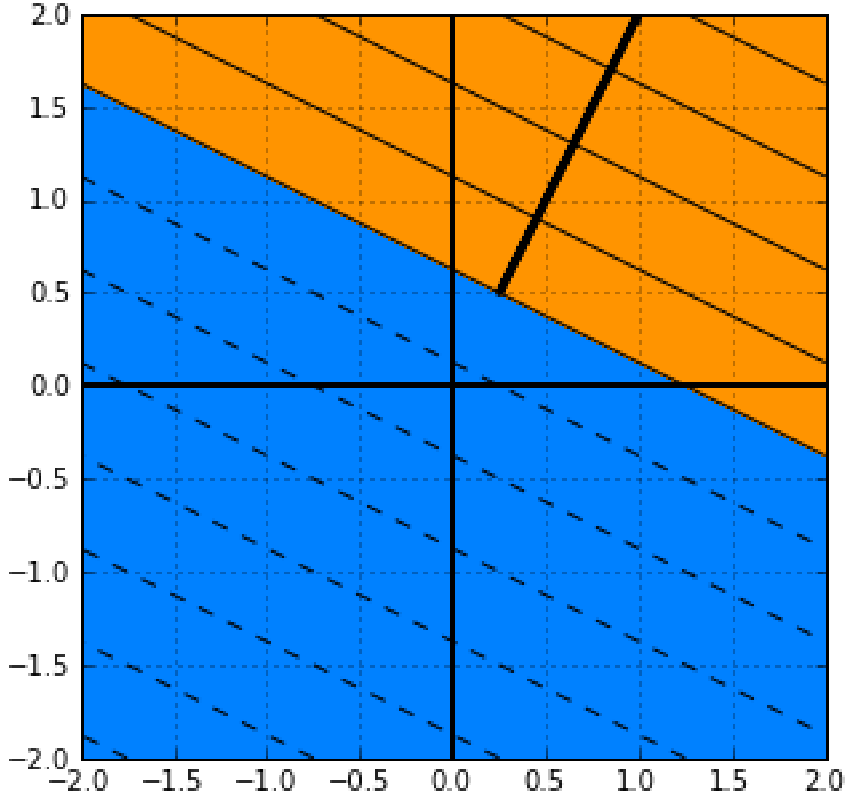
$f = 1$

$f = 0$

$f = -1$

Linear classification

$$\hat{y} = \text{sign}(1.0x_1 + 2.0x_2 - 1.250)$$

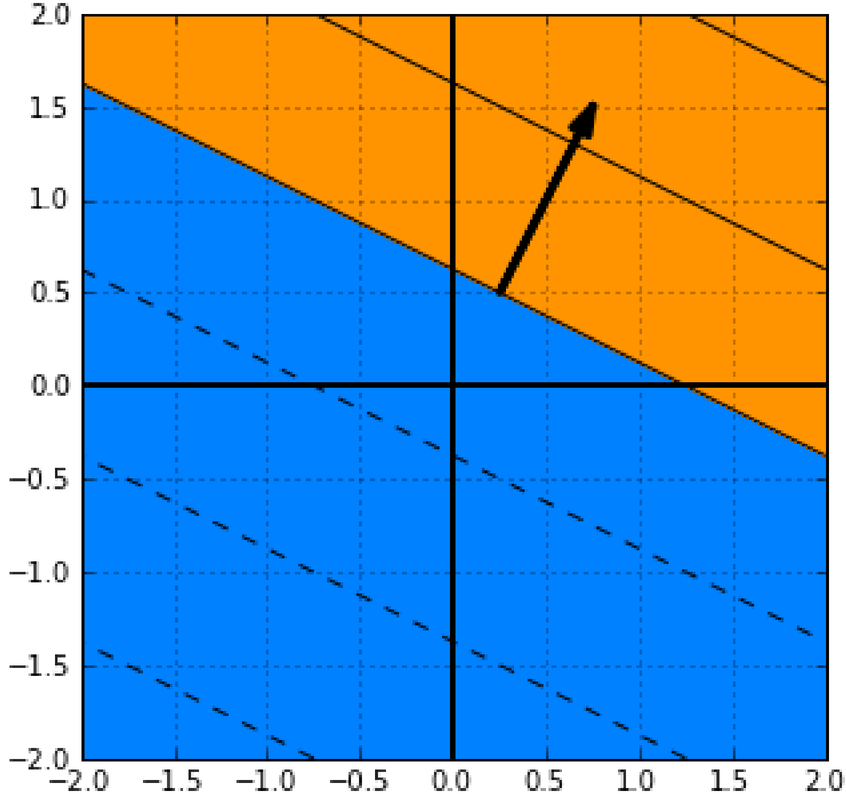


$f = 1$
$f = 0$

$f = -1$

Linear classification

$$\hat{y} = \mathrm{sign}(0.5x_1 + 1.0x_2 - 0.625)$$



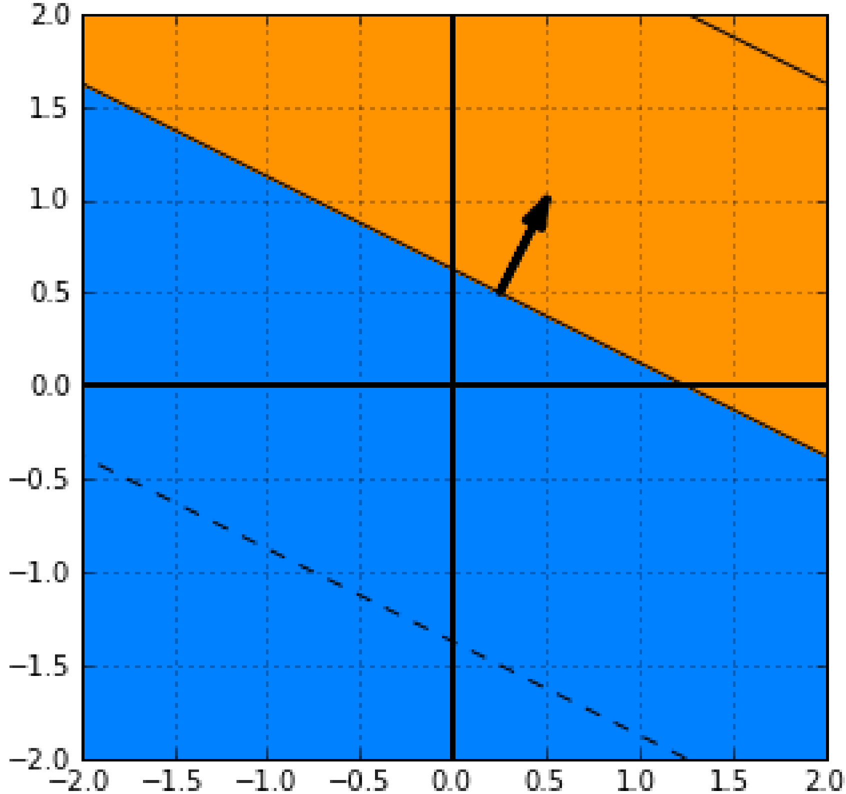
$f = 1$

$f = 0$

$f = -1$

Linear classification

$$\hat{y} = \text{sign}(0.2x_1 + 0.5x_2 - 0.312)$$

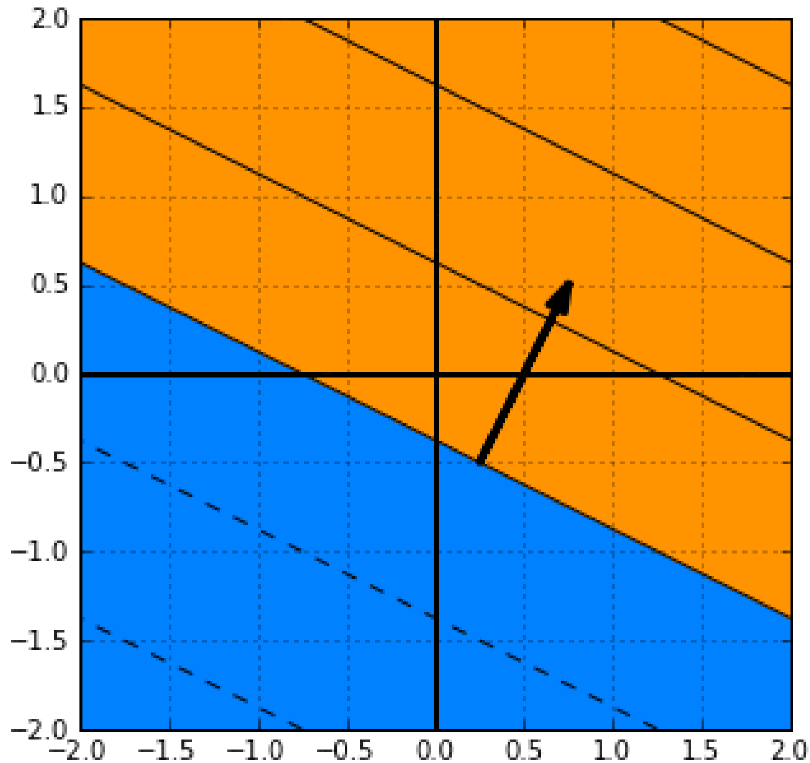


$f = 1$

$f = 0$

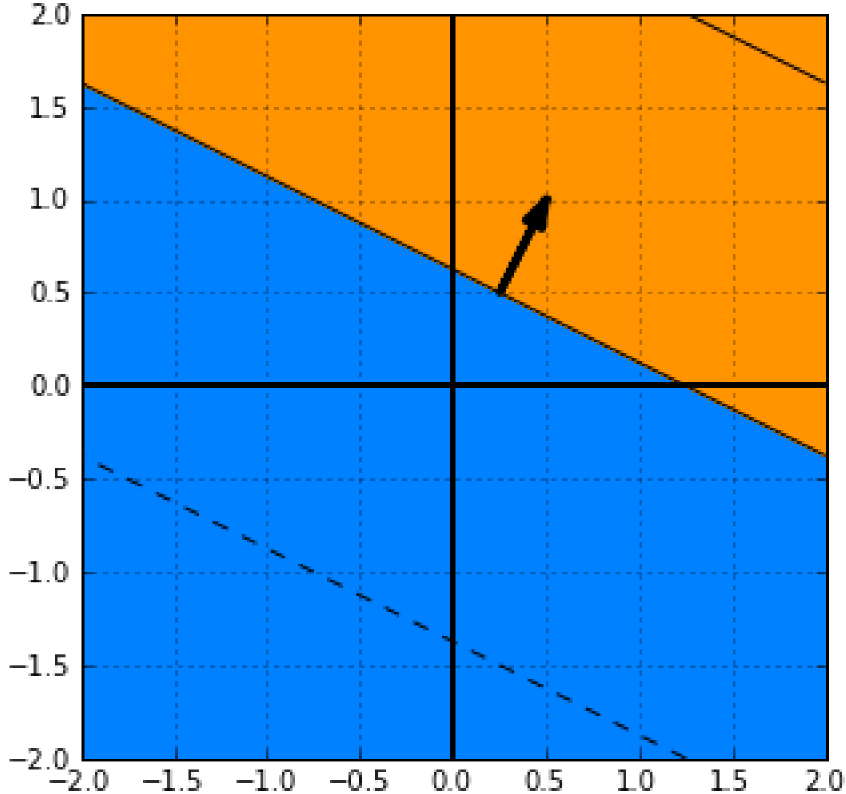
$f = -1$

Level Lines



Linear classification

$$\hat{y} = \mathrm{sign}(0.2x_1 + 0.5x_2 - 0.312)$$



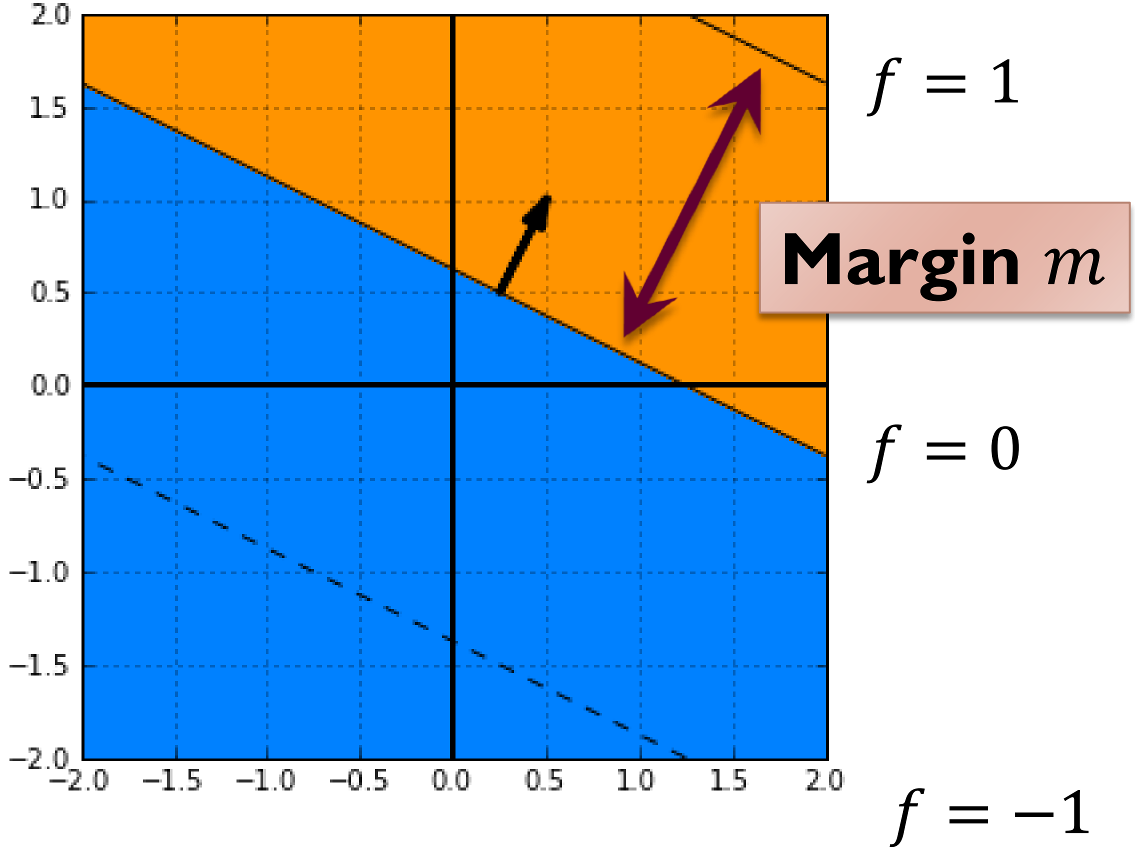
$f = 1$

$f = 0$

$f = -1$

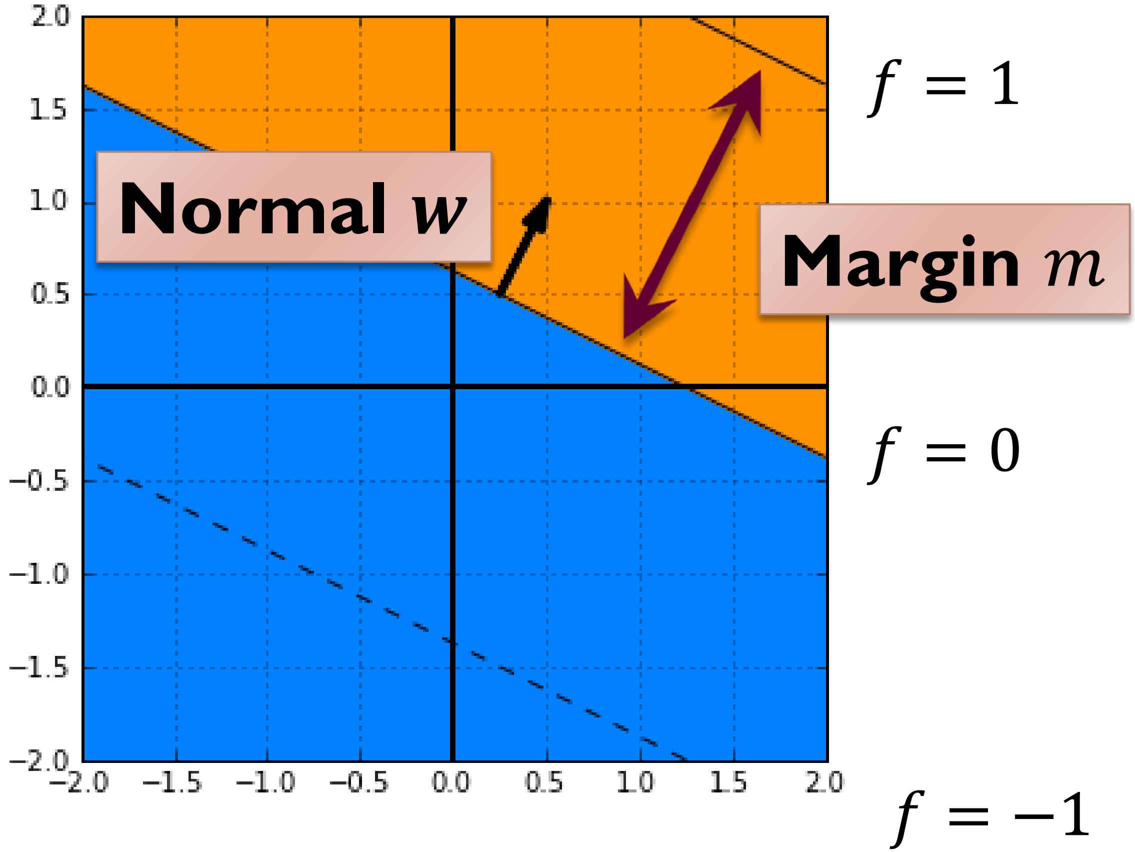
Linear classification

$$\hat{y} = \mathrm{sign}(0.2x_1 + 0.5x_2 - 0.312)$$



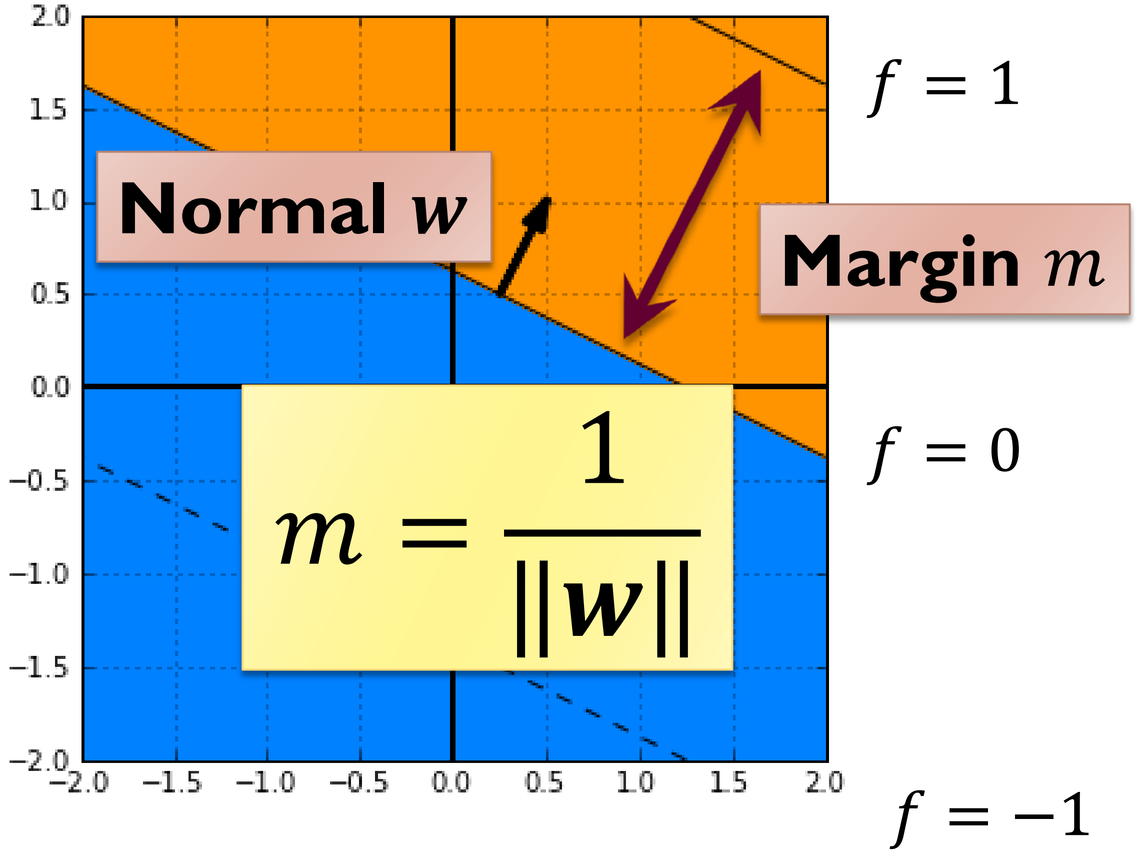
Linear classification

$$\hat{y} = \mathrm{sign}(0.2x_1 + 0.5x_2 - 0.312)$$

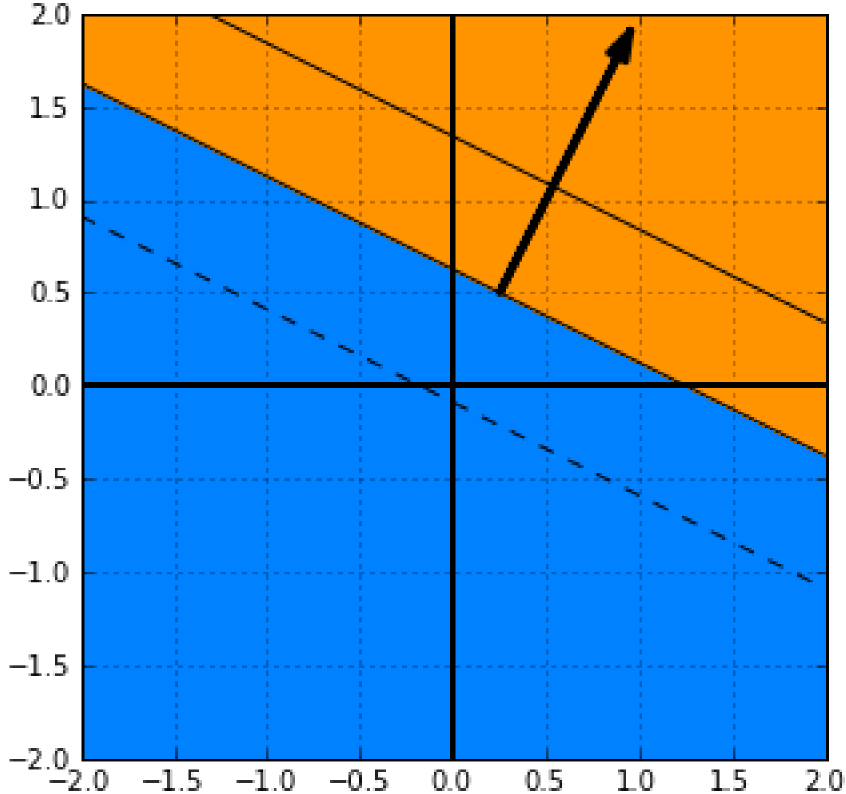


Linear classification

$$\hat{y} = \text{sign}(0.2x_1 + 0.5x_2 - 0.312)$$



Linear classification

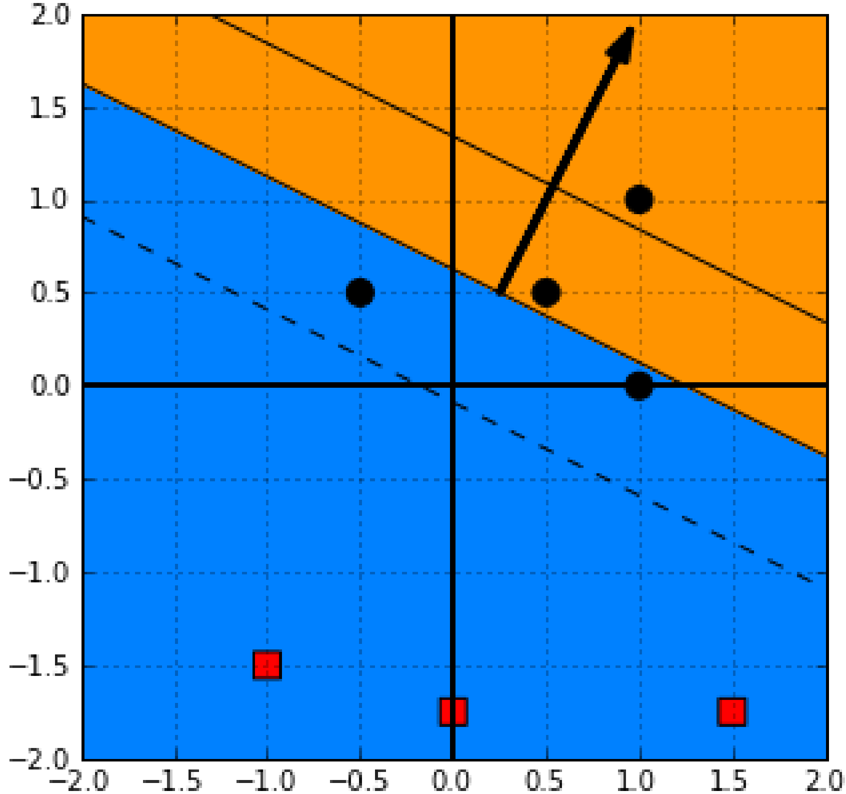


$f = 1$

$f = 0$

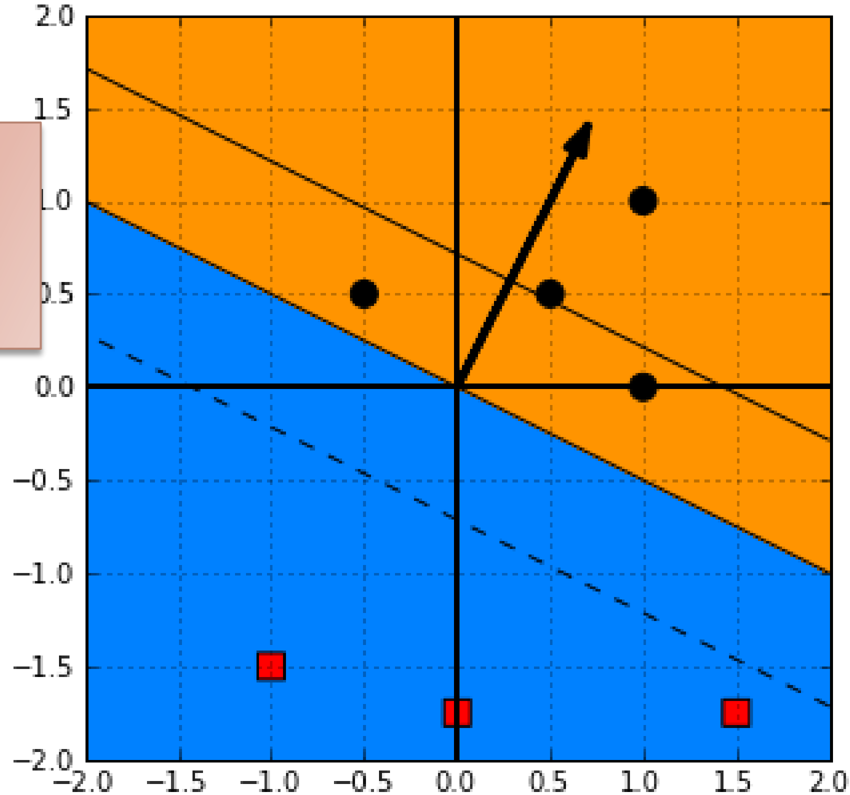
$f = -1$

Linear classification



Linear classification

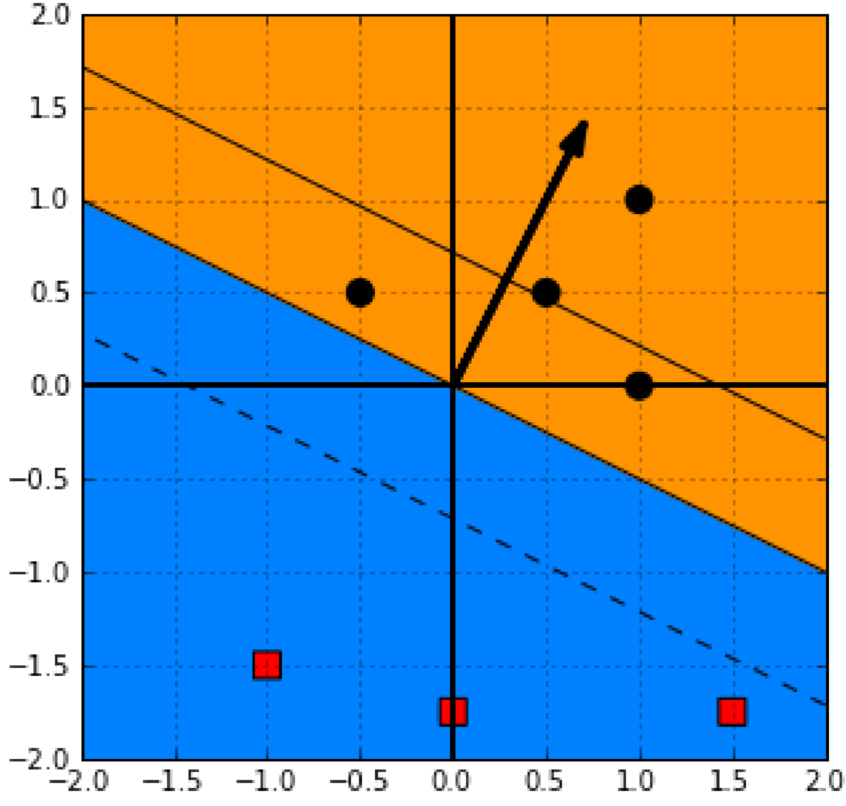
Linearly separable



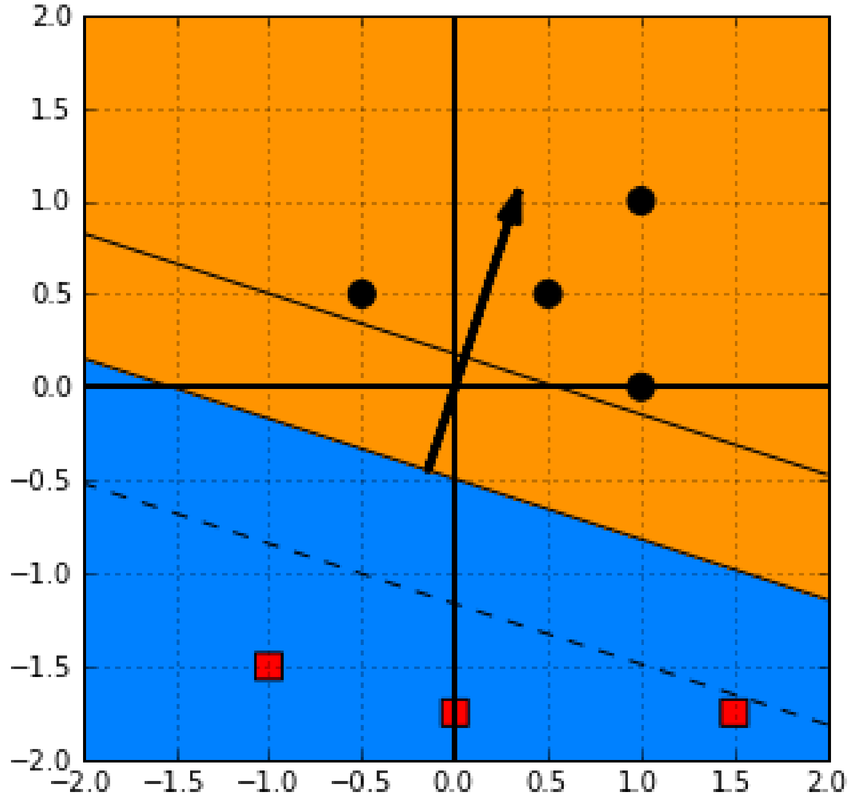
Perceptron

- Start with arbitrary $(\boldsymbol{w}, w_0)$
- Find a misclassified example $(\boldsymbol{x}_i, y_i)$
  - i.e. $\mathrm{sign}(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \neq y_i$
- Update:
  - $\boldsymbol{w} \coloneqq \boldsymbol{w} + y_i\boldsymbol{x}_i$
  - $w_0 \coloneqq w_0 + y_i$

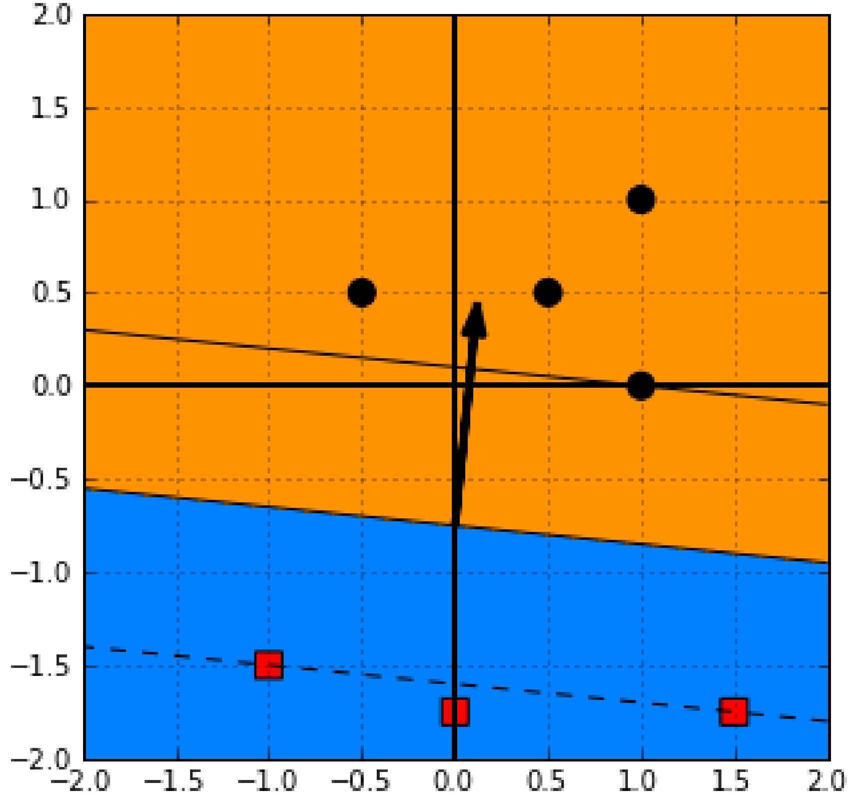
Linear classification



Linear classification



Linear classification

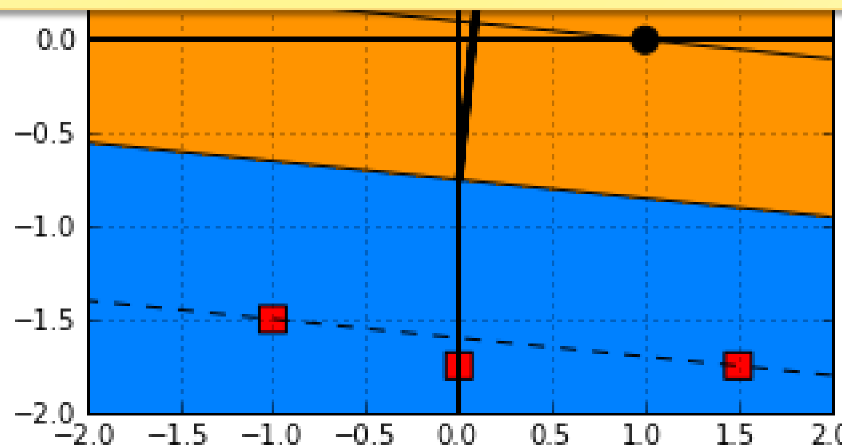


Maximal Margin Classifier

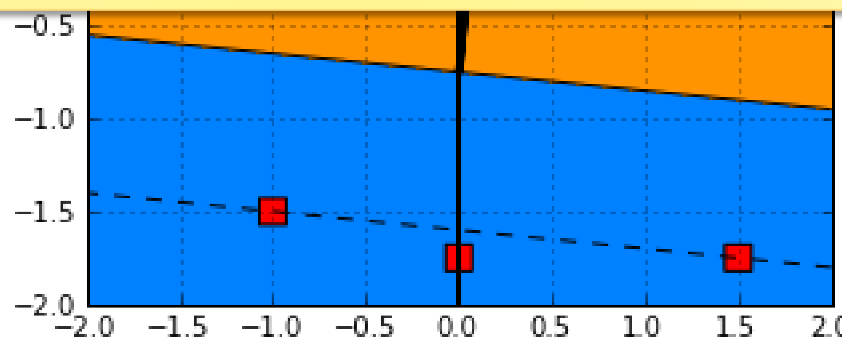
Maximal Margin Classifier

$$\mathbf{argmax}_w\ m$$

s.t. no points violate the margin



Maximal Margin Classifier



Maximal Margin Classifier

$$\mathrm{argmax}_w \frac{1}{||w||}$$

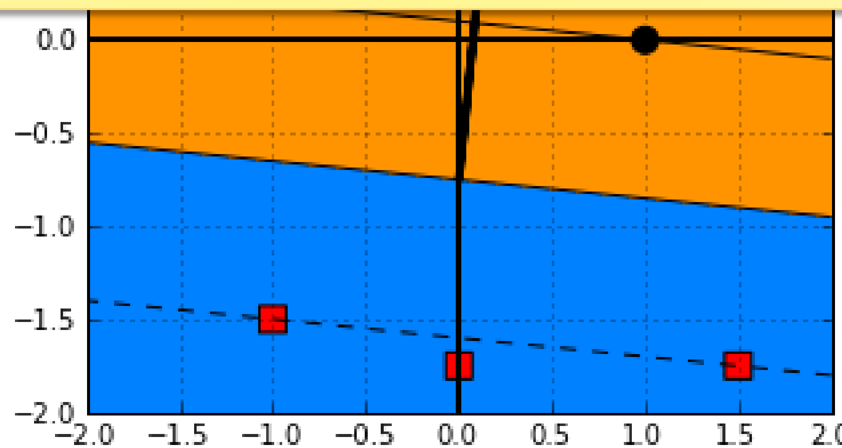
s.t. no points violate the margin

Maximal Margin Classifier

Maximal Margin Classifier

$$\operatorname{argmin}_w ||w||$$

s.t. no points violate the margin

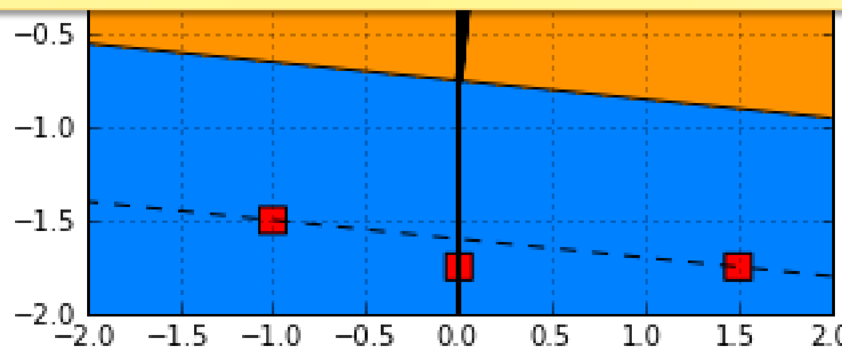


Maximal Margin Classifier

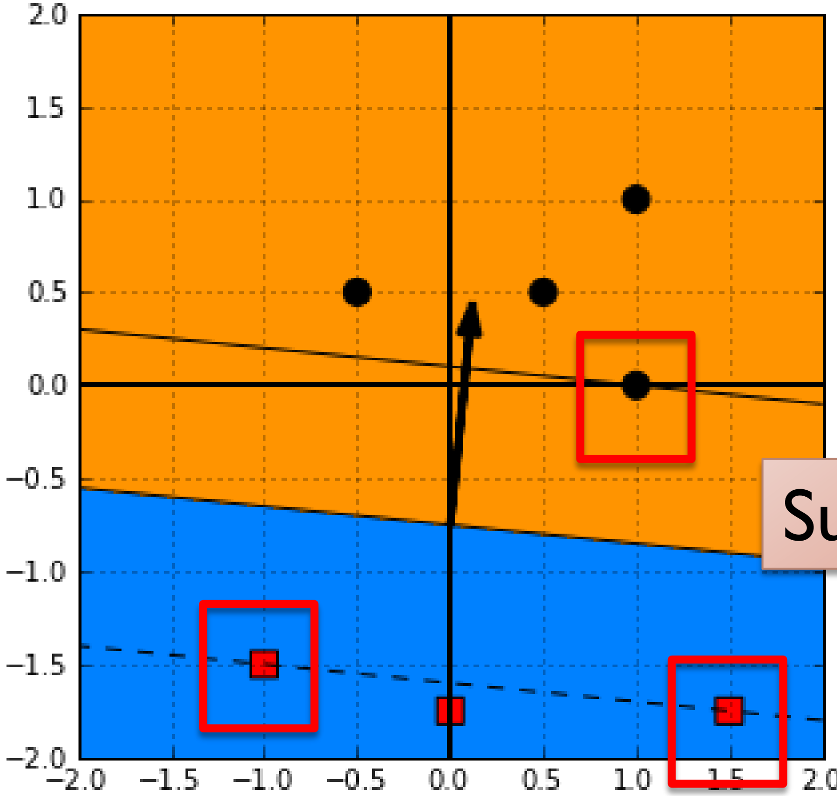
Maximal Margin Classifier

$$\mathrm{argmin}_w \frac{1}{2} \|w\|^2$$

s.t. no points violate the margin

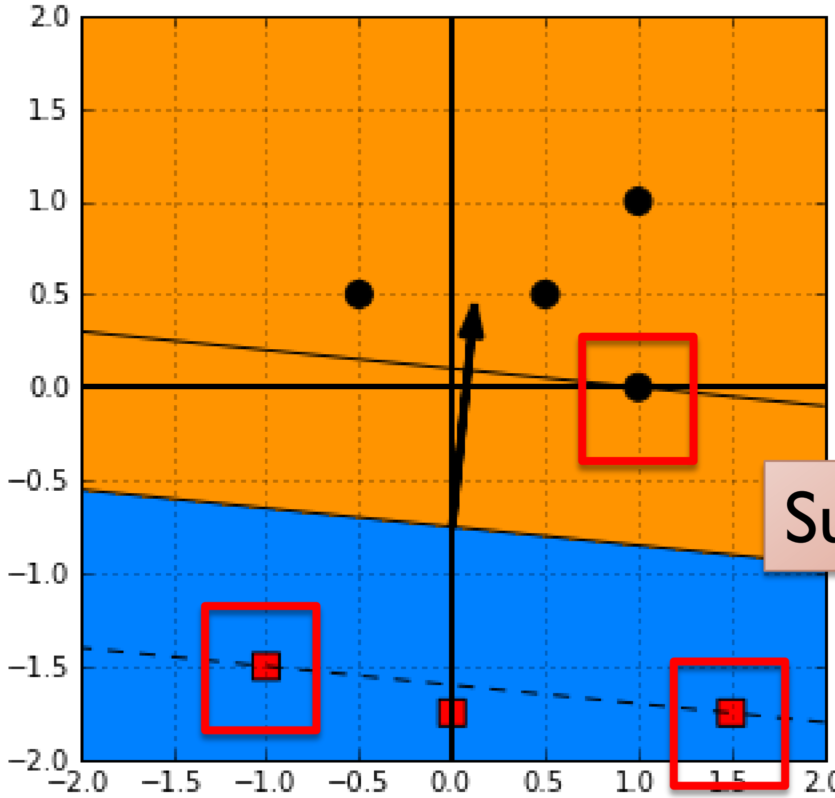


Support Vectors



Support vectors

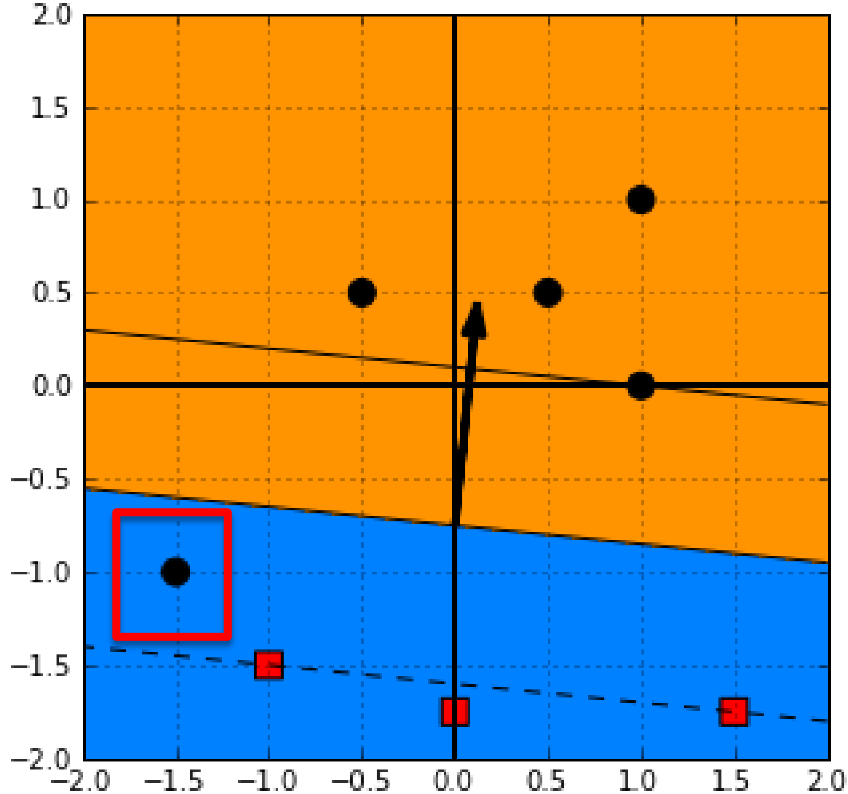
Support Vectors



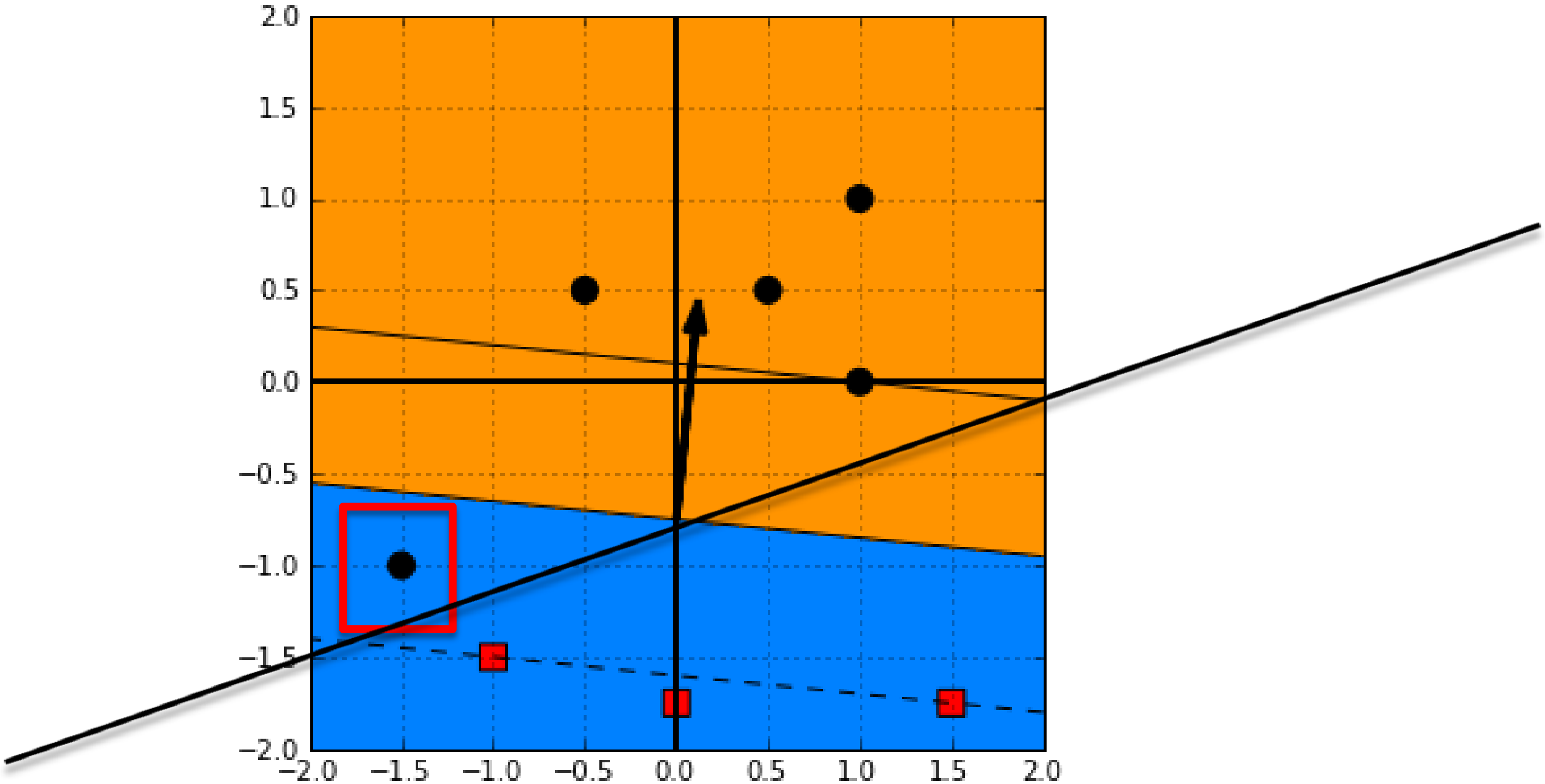
Support vectors

Sparsity!

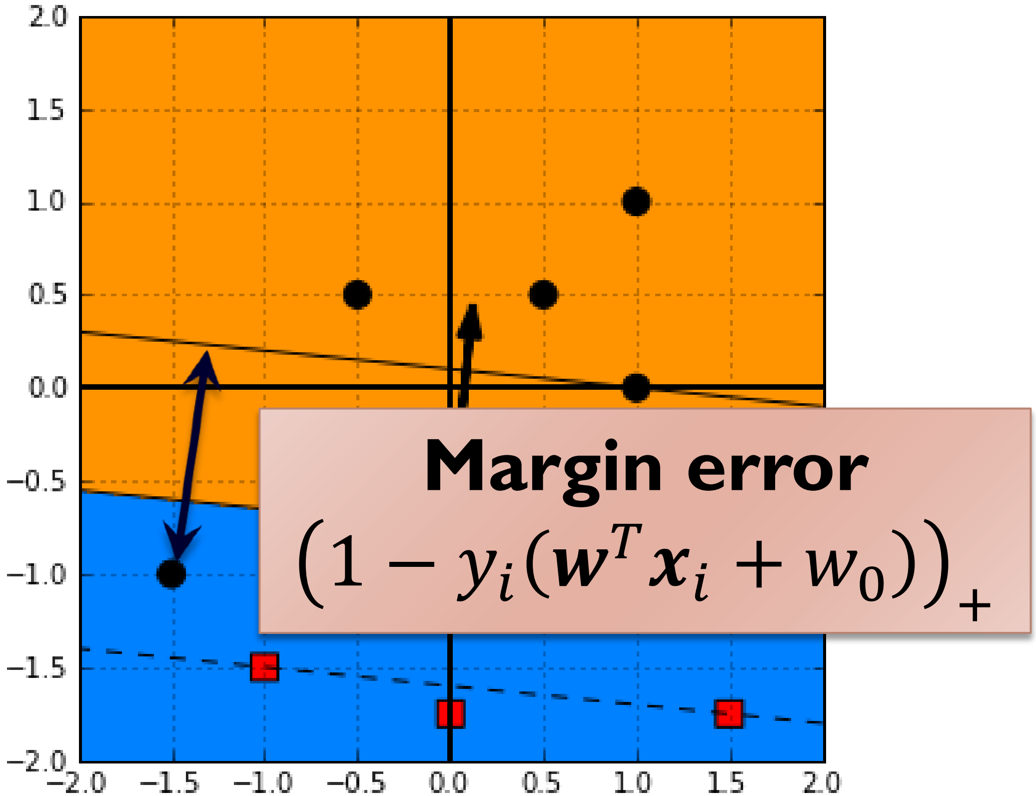
What about outliers?



What about outliers?



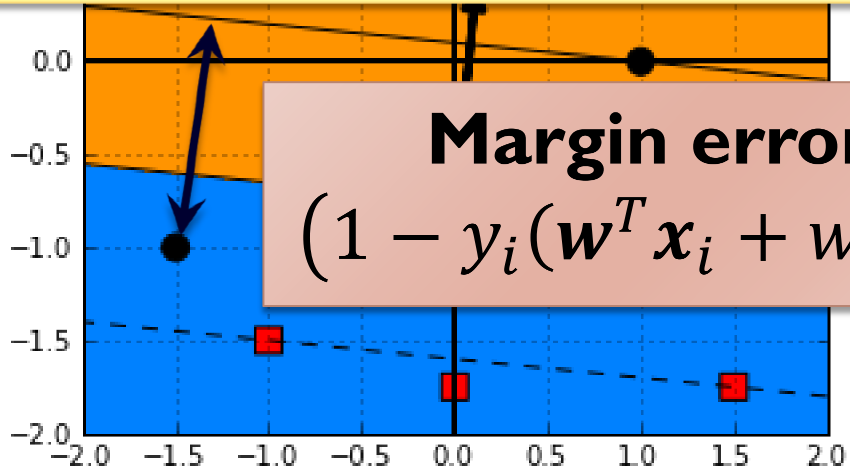
Soft-Margin Error



Soft-Margin Classifier

Soft Margin Classifier

$$\operatorname{argmin}_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \left(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)\right)_+$$



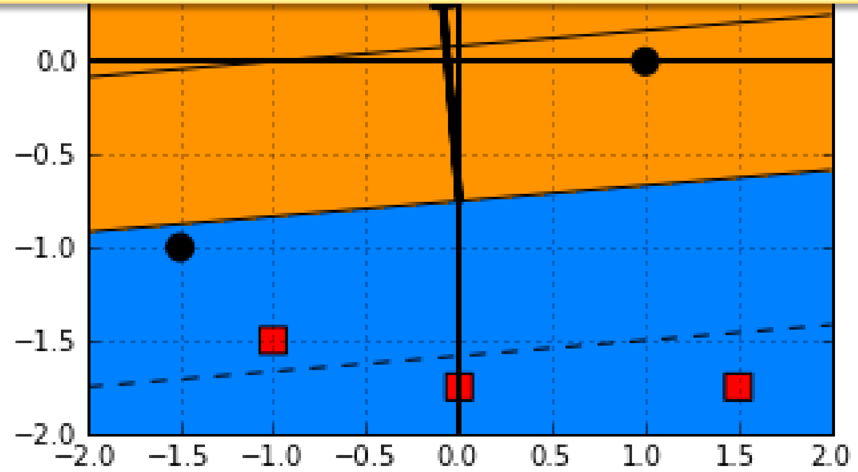
Margin error

$$\left(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)\right)_+$$

Soft-Margin Classifier

Soft Margin Classifier

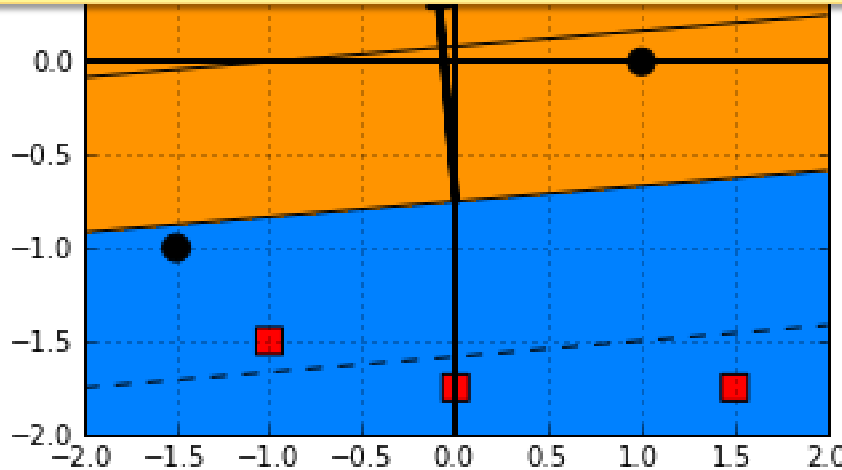
$$\operatorname{argmin}_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \left(1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0)\right)_+$$



Soft-Margin Classifier

Soft Margin Classifier

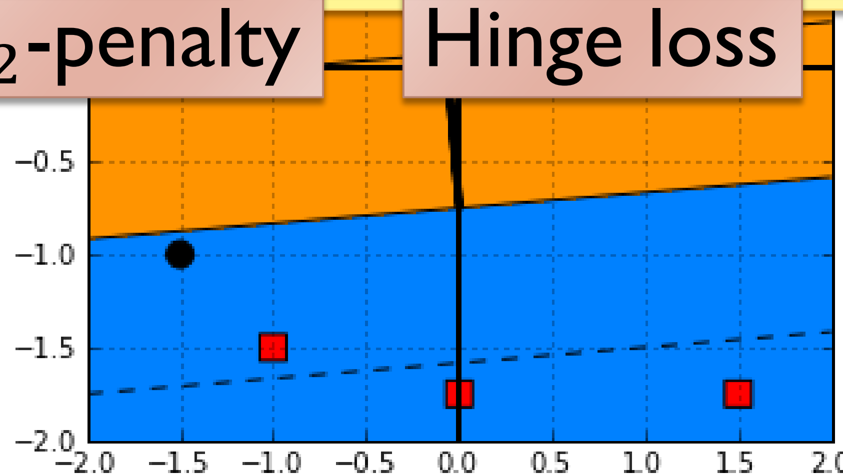
$$\text{argmin}_{\boldsymbol{w}} \boxed{\frac{1}{2}\|\boldsymbol{w}\|^2} + \boxed{C\sum_i \left(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)\right)_+}$$



Soft-Margin Classifier

Soft Margin Classifier

$$\operatorname{argmin}_{\boldsymbol{w}} \underbrace{\boxed{\frac{1}{2}\|\boldsymbol{w}\|^2}}_{\ell_2\text{-penalty}} + \underbrace{\boxed{C\sum_i \left(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)\right)_+}}_{\text{Hinge loss}}$$



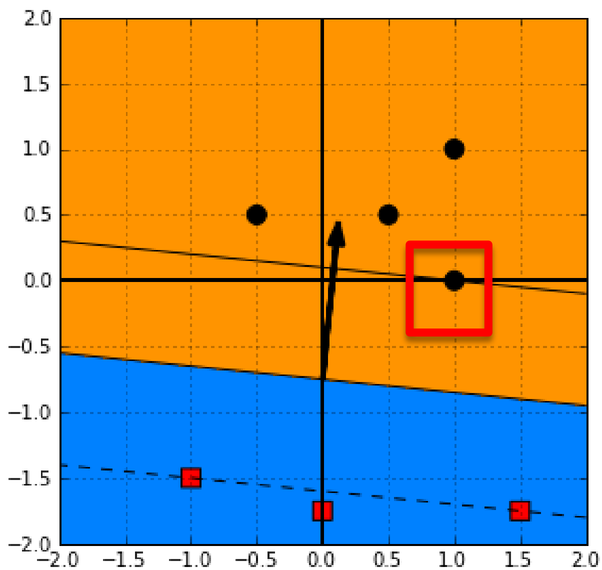
Support Vector Classifier

```
from sklearn.svm import SVC

m = SVC(C=1, kernel='linear')
m.fit(X, y)
m.predict(X_new)
```

Quiz

- The smaller is the classifier's _________, the larger is its margin.
- If the point lies exactly on the margin (on the correct side), it is called ___________.



Quiz

- If you remove all non-support vectors from the dataset and re-train the model, the resulting hyperplane will be ______________.

Quiz

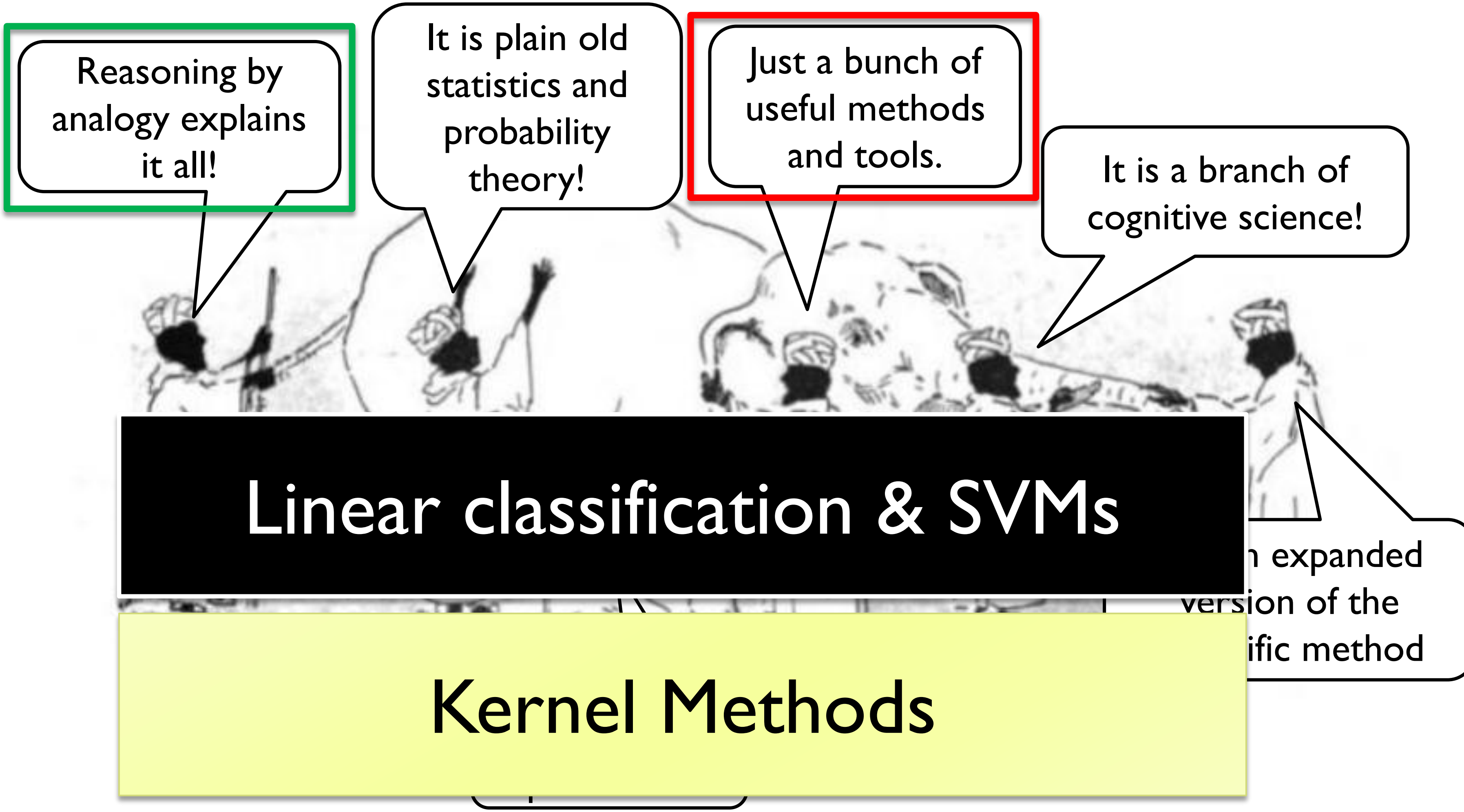
- The margin error of a point exactly on the margin is ____________.

- The margin error of a point exactly on the separating hyperplane is ________________.

Margin error

$$\left(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)\right)_+$$

Next



Linear classification & SVMs

Kernel Methods

The Kernel-based Approach

Nonlinear feature mapping

The Kernel trick

Dual representation

- Linear regression (OLS, Ridge):

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

- Linear classification (SVM):

$$f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

- Linear regression (OLS, Ridge, LASSO, SVR,…):

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

- Linear classification (SVM, Perceptron, Fisher's discriminant, Logistic regression, NB, …):

$$f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

- Linear regression (OLS, Ridge, LASSO, SVR,…):

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

- Linear classification (SVM, Perceptron, Fisher's discriminant, Logistic regression, NB, …):

$$f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

- PCA, LDA, ICA, CCA…:

$$\boldsymbol{x}_T = \boldsymbol{A}\boldsymbol{x}$$

- K-means:

$$\boldsymbol{c}_i = \frac{1}{m}\boldsymbol{X}_i\boldsymbol{1}$$

- Linear regression (OLS, Ridge, LASSO, SVR,…):

- Line her's disc

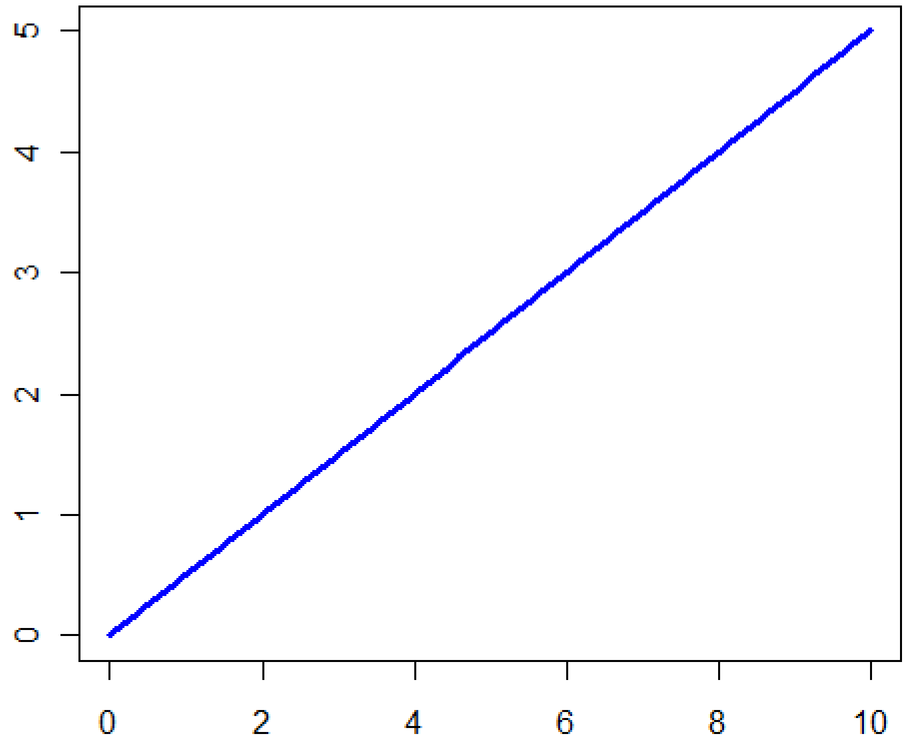
- PC

- K-m

$$\boldsymbol{c}_i = \frac{\quad}{m}\boldsymbol{x}_i\mathbf{1}$$

Recall polynomial regression

$$f(x) = wx$$

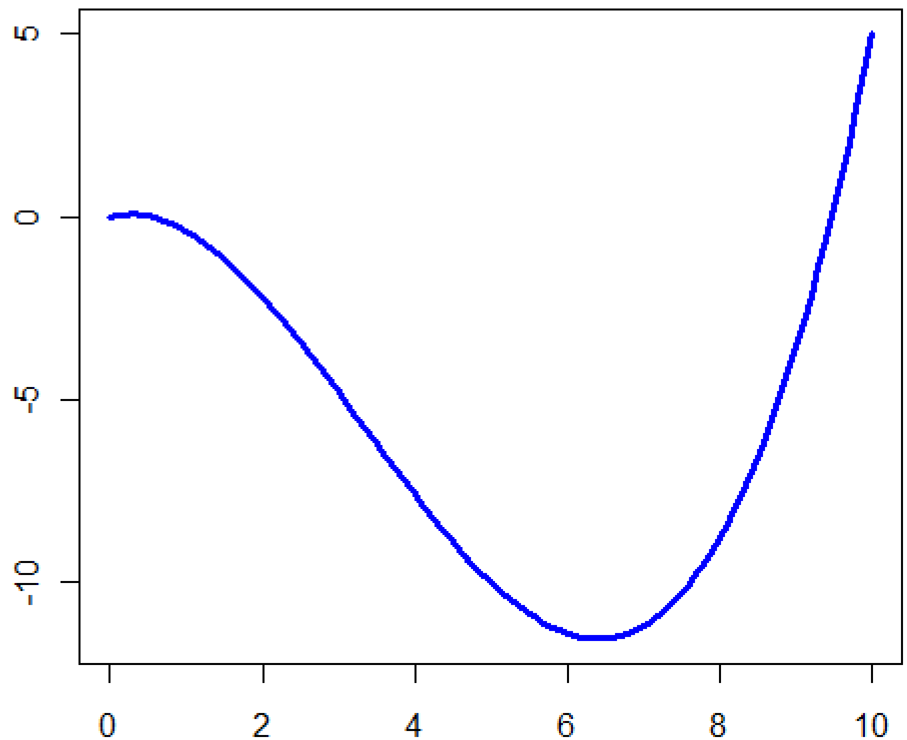


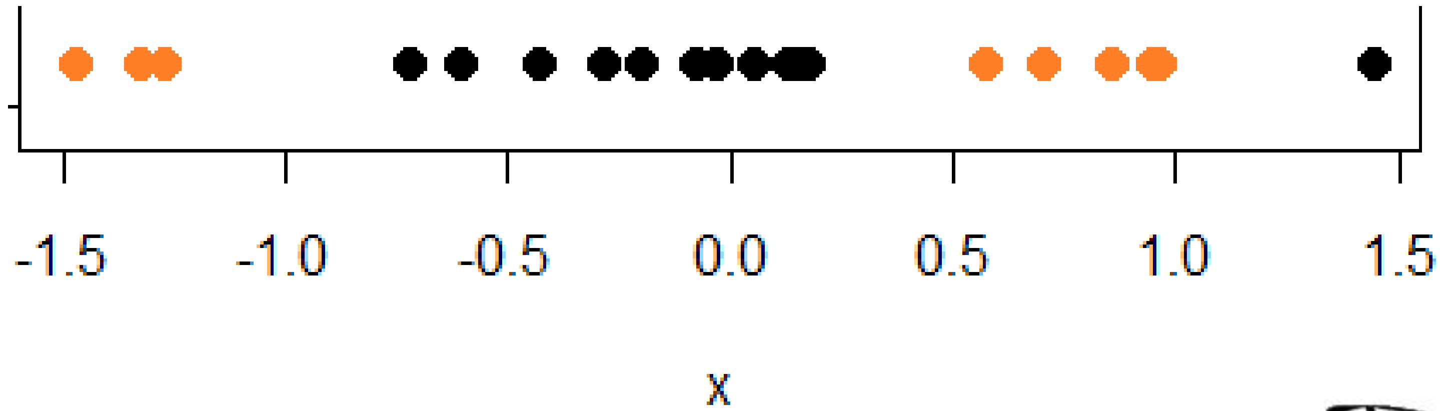
Recall polynomial regression

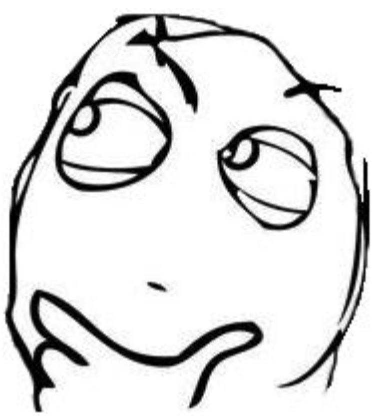
$$x \to x' := \phi(x) := (x, x^2, x^3)$$

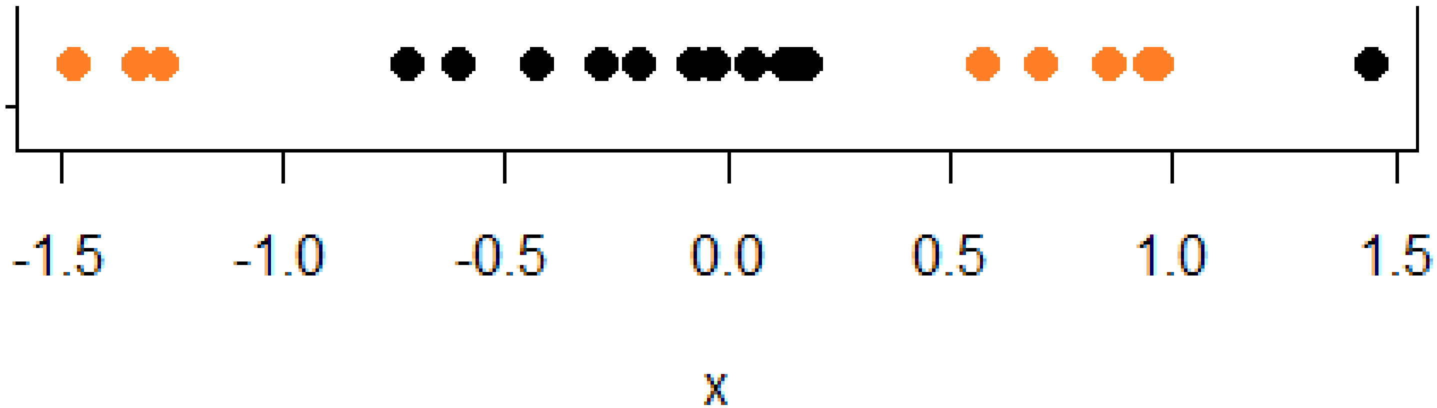
Nonlinear feature space

$$x \to x' := \phi(x) := (x, x^2, x^3)$$
$$f(x') = w_1 x + w_2 x^2 + w_3 x^3$$

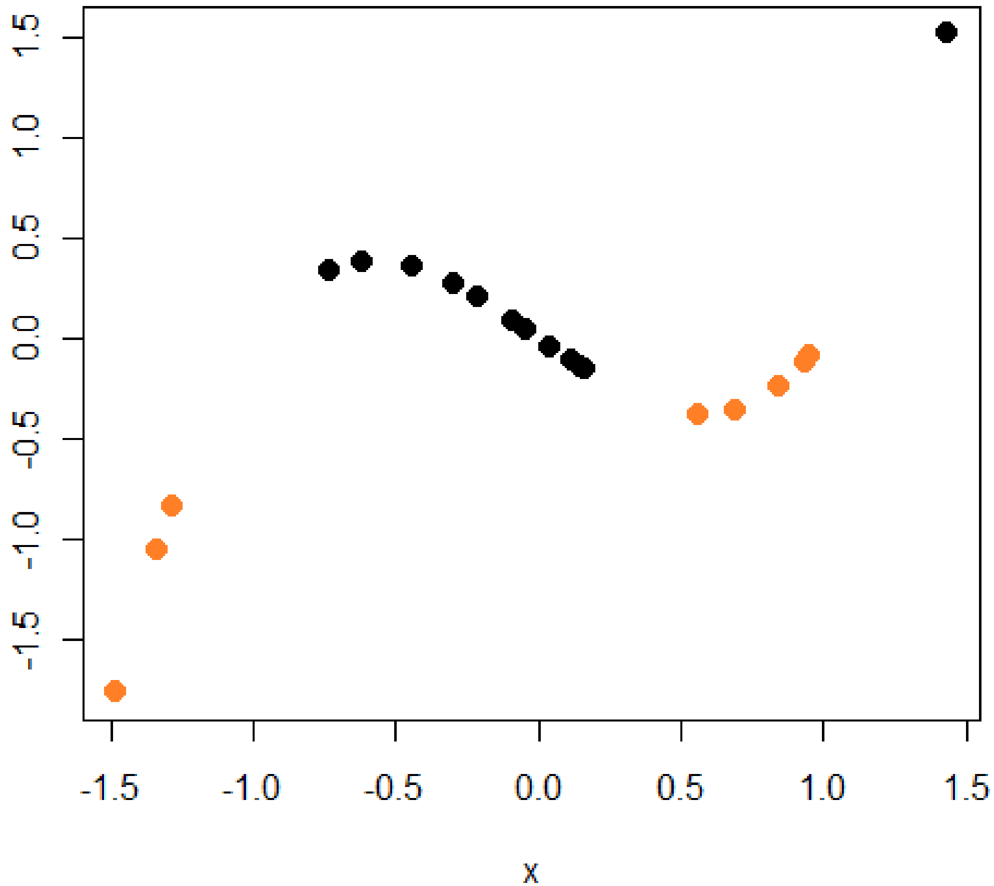



-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
x

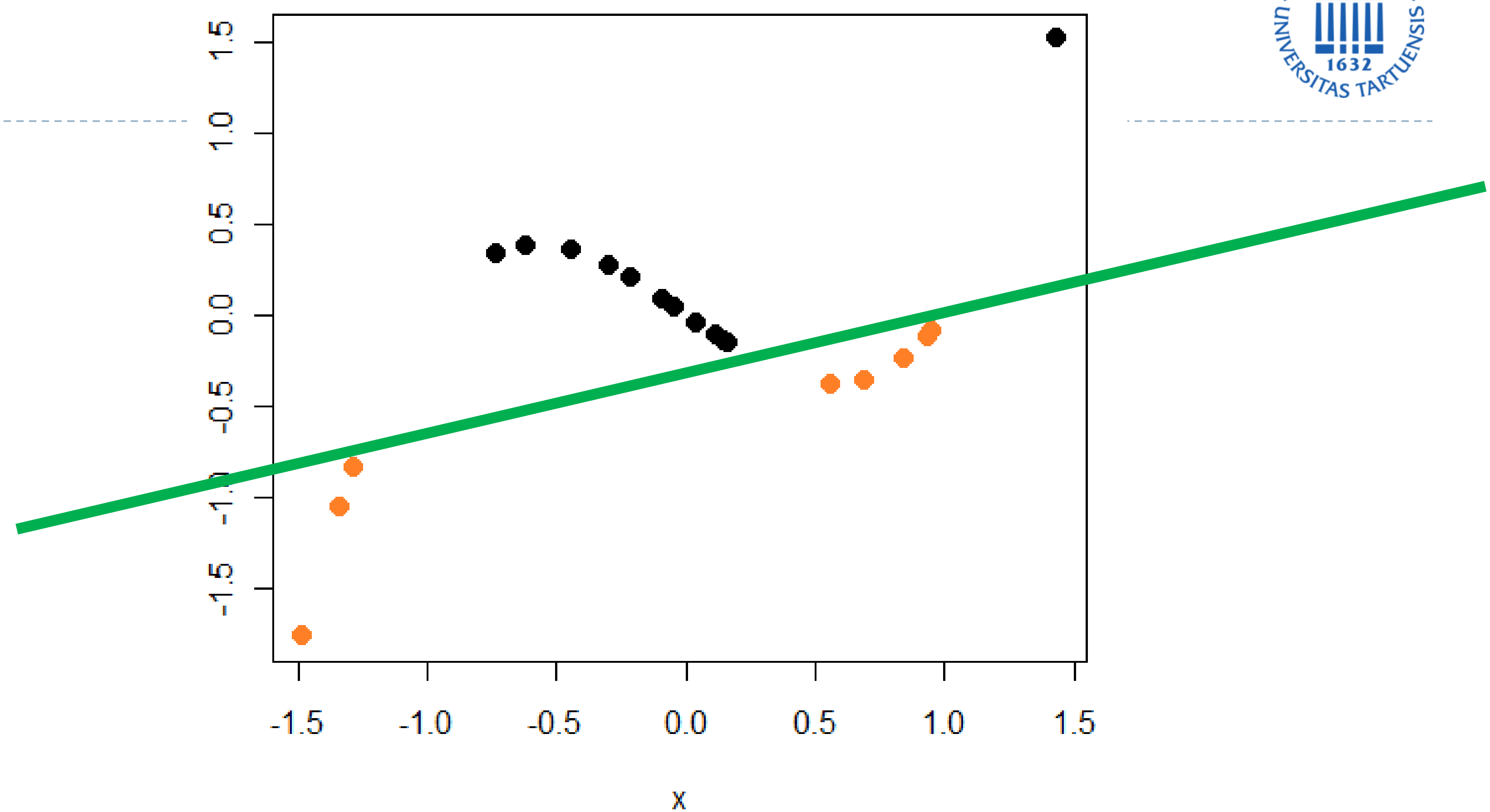


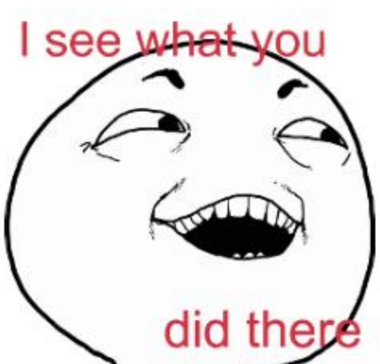


$$x \rightarrow \phi(x) = (x,\ x^3 - x)$$



$$x \to \phi(x) = (x,\ x^3 - x)$$

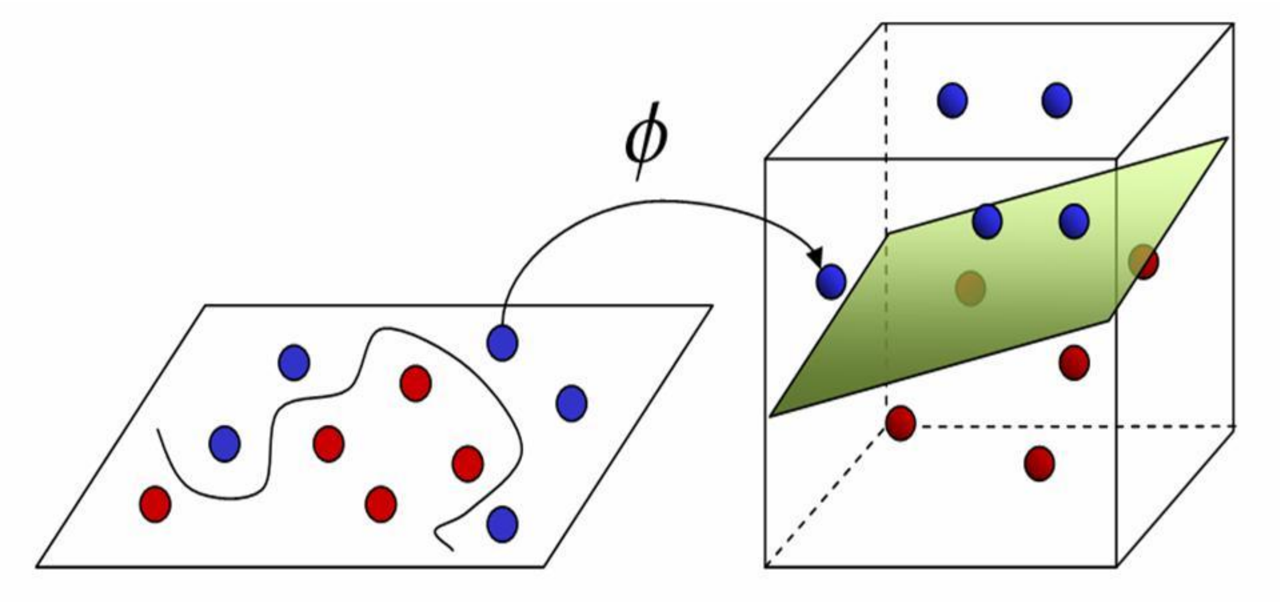




$$x \to \phi(x) = (x,\ x^3 - x)$$

Nonlinear feature space

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x})$$



Support for arbitrary data types

$$\phi(\text{text}) = \text{word counts}$$
$$\phi(\text{graph}) = \text{node degrees}$$
$$\phi(\text{tree}) = \text{path lengths}$$
$$\ldots$$

What if the dimensionality is high?

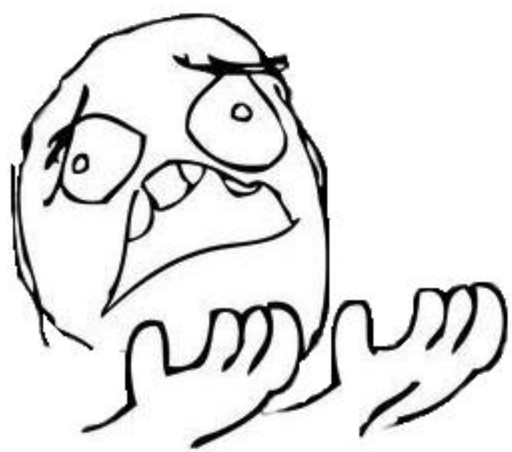
$$(x_1, x_2, \ldots, x_m) \rightarrow (x_1 x_1, x_1 x_2, \ldots, x_m x_m)$$

What if the dimensionality is high?

$$(x_1, x_2, \dots, x_m) \rightarrow (x_1 x_1, x_1 x_2, \dots, x_m x_m)$$

$O(m^2)$ elements

For all k-wise products: $O(m^k)$



The Kernel-based Approach

Nonlinear feature mapping

The Kernel trick

Dual representation

The Kernel Trick

- Let $\phi(\boldsymbol{x}) = (x_1 x_1, x_1 x_2, \ldots, x_m x_m)$
- Consider

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \sum_{ij} \phi(\boldsymbol{x})_{ij} \phi(\boldsymbol{y})_{ij}$$

The Kernel Trick

- Let $\phi(\boldsymbol{x}) = (x_1x_1, x_1x_2, \ldots, x_mx_m)$
- Consider

$$\langle\phi(\boldsymbol{x}), \phi(\boldsymbol{y})\rangle = \sum_{ij} \phi(\boldsymbol{x})_{ij}\phi(\boldsymbol{y})_{ij}$$

$$= \sum_{ij} x_ix_jy_iy_j$$

The Kernel Trick

- Let $\phi(\boldsymbol{x}) = (x_1 x_1, x_1 x_2, \ldots, x_m x_m)$
- Consider

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \sum_{ij} \phi(\boldsymbol{x})_{ij} \phi(\boldsymbol{y})_{ij}$$

$$= \sum_{ij} x_i x_j y_i y_j = \sum_{ij} x_i y_i x_j y_j$$

The Kernel Trick

- Let $\phi(\boldsymbol{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$
- Consider

$$\langle\phi(\boldsymbol{x}), \phi(\boldsymbol{y})\rangle = \sum_{ij} \phi(\boldsymbol{x})_{ij}\phi(\boldsymbol{y})_{ij}$$

$$= \sum_{ij} x_ix_jy_iy_j = \sum_{ij} x_iy_ix_jy_j$$

$$= \sum_i x_iy_i \sum_j x_jy_j$$

The Kernel Trick

- Let $\phi(\boldsymbol{x}) = (x_1 x_1, x_1 x_2, \ldots, x_m x_m)$
- Consider

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \sum_{ij} \phi(\boldsymbol{x})_{ij} \phi(\boldsymbol{y})_{ij}$$

$$= \sum_{ij} x_i x_j y_i y_j = \sum_{ij} x_i y_i x_j y_j$$

$$= \sum_i x_i y_i \sum_j x_j y_j = \left(\sum_i x_i y_i \right)^2$$

The Kernel Trick

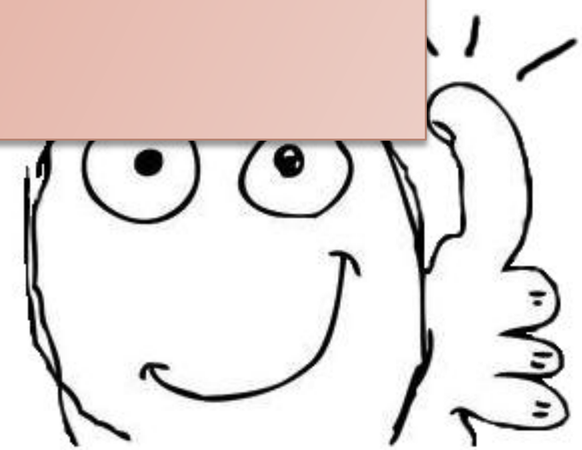
- Let $\phi(\boldsymbol{x}) = (x_1 x_1, x_1 x_2, \dots, x_m x_m)$
- $$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

The Kernel Trick

- Let

Polynomial kernel

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + R)^d$$



What about:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + 0.5\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2?$$

The Kernel Trick

What about:

$$K(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{x},\boldsymbol{y}\rangle + 0.5\langle \boldsymbol{x},\boldsymbol{y}\rangle^2$$

$$= \sum_i x_i y_i + 0.5 \sum_{ij} \phi_{ij}(\boldsymbol{x})\phi_{ij}(\boldsymbol{y})$$

The Kernel Trick

What about:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + 0.5\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

$$= \sum_i x_i y_i + 0.5 \sum_{ij} \phi_{ij}(\boldsymbol{x})\phi_{ij}(\boldsymbol{y})$$

$$= \langle (x_1, \dots, x_m, \sqrt{}0.5 x_1 x_1, \dots, \sqrt{}0.5 x_m x_m),$$
$$(y_1, \dots, y_m, \sqrt{}0.5 y_1 y_1, \dots, \sqrt{}0.5 y_m y_m) \rangle$$

The Kernel Trick

What about:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + 0.5 \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

$$= \sum_i x_i y_i + 0.5 \sum_{ij} \phi_{ij}(\boldsymbol{x}) \phi_{ij}(\boldsymbol{y})$$

$$= \langle (x_1, \ldots, x_m, \sqrt{0.5} x_1 x_1, \ldots, \sqrt{0.5} x_m x_m),$$
$$(y_1, \ldots, y_m, \sqrt{0.5} y_1 y_1, \ldots, \sqrt{0.5} y_m y_m) \rangle$$

What about:

$$K(\boldsymbol{x},\boldsymbol{y}) = 1 + \langle \boldsymbol{x},\boldsymbol{y}\rangle + \frac{1}{2}\langle \boldsymbol{x},\boldsymbol{y}\rangle^2 + \frac{1}{6}\langle \boldsymbol{x},\boldsymbol{y}\rangle^3 + \frac{1}{24}\langle \boldsymbol{x},\boldsymbol{y}\rangle^4?$$

What about:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{\infty} \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^i}{i!}$$

The Kernel Trick

What about:

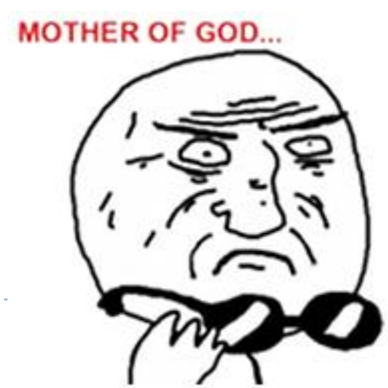
$$K(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=0}^{\infty} \frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle^i}{i!} = \exp\langle \boldsymbol{x},\boldsymbol{y}\rangle$$

The Kernel Trick

What about:

$$K(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=0}^{\infty} \frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle^{i}}{i!} = \exp\langle \boldsymbol{x},\boldsymbol{y}\rangle$$

Infinite-dimensional feature space!



The Kernel Trick

Gaussian (RBF) kernel

$$K(\boldsymbol{x},\boldsymbol{y}) =$$
$$= \exp(-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|^2)$$
$$= \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right)$$

The Kernel Trick

Exponential kernel

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|}{2\sigma^2}\right)$$

Kernels

$$k(x,y) = x^T y + c$$

$$k(x,y) = \sqrt{\|x-y\|^2 + c^2}$$

$$k(x,y) = \frac{\theta}{\|x-y\|} \sin \frac{\|x-y\|}{\theta}$$

$$k(x,y) = (\alpha x^T y + c)^d$$

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

$$k(x,y) = \frac{1}{\sqrt{\|x-y\|^2 + c^2}}$$

$$k(x,y) = \exp\left(-\frac{\|x-y\|}{2\sigma^2}\right)$$

$$k(x,y) = \frac{2}{\pi} \arccos(-\frac{\|x-y\|}{\sigma}) - \frac{2}{\pi} \frac{\|x-y\|}{\sigma} \sqrt{1 - \left(\frac{\|x-y\|}{\sigma}\right)^2}$$

$$k(x,y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$$

$$k(x,y) = 1 - \frac{3}{2}\frac{\|x-y\|}{\sigma} + \frac{1}{2}\left(\frac{\|x-y\|}{\sigma}\right)^3$$

$$k(x,y) = -log(\|x-y\|^d + 1)$$

$$k(x,y) = \tanh(\alpha x^T y + c)$$

$$k(x,y) = -\|x-y\|^d$$

$$k(x,y) = \frac{1}{1 + \frac{\|x-y\|^2}{\sigma}}$$

$$k(x,y) = 1 - \frac{\|x-y\|^2}{\|x-y\|^2 + c}$$

$$k(x,y) = 1 + xy + xy\ min(x,y) - \frac{x+y}{2}\ min(x,y)^2 + \frac{1}{3}\min(x,y)^3$$

$$k(x,y) = B_{2p+1}(x-y)$$

$$k(x,y) = \sum_{i=1}^{n} \min(x_i, y_i)$$

$$k(x,y) = \prod_{i=1}^{N} h(\frac{x_i - c}{a})\ h(\frac{y_i - c}{a})$$

$$k(x,y) = \sum_{i=1}^{m} \min(|x_i|^\alpha, |y_i|^\beta)$$

$$k(x,y) = \frac{J_{v+1}(\sigma\|x-y\|)}{\|x-y\|^{-n(v+1)}}$$

$$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$$

$$\kappa_l(a,b) = \sum_{c \in \{0:1\}} P(Y = c \mid X_l = a)\ P(Y = c \mid X_l = b)$$

$$k(x,y) = \frac{1}{1 + \|x-y\|^d}$$

Structured data kernels

- String kernels
  - P-spectrum kernels
  - All-subsequences kernels
  - Gap-weighted subsequences kernels
  - …
- Graph & tree kernels
  - Co-rooted subtrees
  - All subtrees
  - Random walks
  - …

Kernel

- A function $K(\boldsymbol{x}, \boldsymbol{y})$ is a *kernel*, if

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$$

for some *feature map* ϕ.

Kernel matrix

- For a given kernel function K and a finite dataset $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$ the $n \times n$ *matrix*

$$\boldsymbol{K}_{ij} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

is called the *kernel matrix*.

Kernel matrix

- Let $\boldsymbol{X}$ be the data matrix, then
$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^T$$
is the kernel matrix for the *linear* kernel
$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y}$$

Kernel matrix

- Let $\boldsymbol{X}$ be the data matrix, then
$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^T$$
is the kernel matrix for the *linear* kernel
$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$$

- Let ϕ be a feature mapping. Then*
$$\boldsymbol{K} = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T$$
is the kernel matrix for the corresponding kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$.

Kernel theorem

- Not every function K is a kernel!

  e.g. $K(x, y) = -1$ is not

- Not every $n \times n$ matrix is a Kernel matrix!

Kernel theorem

- Theorem:

K is a kernel function $\Leftrightarrow$ K is *symmetric positive semidefinite*

- A function is *positive semidefinite* iff for any finite dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ the corresponding *kernel matrix* is positive semidefinite.

Kernel closure

- $\kappa(x, z) = \kappa_1(x, z) + \kappa_2(x, z)$
- $\kappa(x, z) = \alpha\kappa_1(x, z)$
- $\kappa(x, z) = \kappa_1(x, z)\kappa_2(x, z)$
- $\kappa(x, z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$
- $\kappa(x, z) = x^T B z$ where B is a psd matrix.

P. Agius – L3, Spring 2008

Kernel closure

Feature space concatenation

- $\kappa(x, z) = \kappa_1(x, z) + \kappa_2(x, z)$
- $\kappa(x, z) = \alpha\kappa_1(x, z)$
- $\kappa(x, z) = \kappa_1(x, z)\kappa_2(x, z)$
- $\kappa(x, z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$
- $\kappa(x, z) = x^T B z$ where B is a psd matrix.

P. Agius – L3, Spring 2008

Kernel closure

Feature space scaling

- $\kappa(x,z) = \kappa_1(x,z) + \kappa_2(x,z)$
- $\kappa(x,z) = \alpha\kappa_1(x,z)$
- $\kappa(x,z) = \kappa_1(x,z)\kappa_2(x,z)$
- $\kappa(x,z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x,z) = \kappa_3(\phi(x),\phi(z))$
- $\kappa(x,z) = x^T B z$ where B is a psd matrix.

P. Agius – L3, Spring 2008

Kernel closure

- $\kappa(x, z) = \kappa_1(x, z) + \kappa$
- $\kappa(x, z) = \alpha\kappa_1(x, z)$
- $\kappa(x, z) = \kappa_1(x, z)\kappa_2(x, z)$
- $\kappa(x, z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$
- $\kappa(x, z) = x^T B z$ where B is a psd matrix.

Feature space tensor product

P. Agius – L3, Spring 2008

Kernel closure

- $\kappa(x,z) = \kappa_1(x,z) + \kappa_2(x,z)$
- $\kappa(x,z) = \alpha\kappa_1(x,z)$
- $\kappa(x,z) = \kappa_1(x,z)\kappa_2(x,z)$
- $\kappa(x,z) = f(x)f(z)$ where f is a real-valued function
- $\kappa(x,z) = \kappa_3(\phi(x),\phi(z))$
- $\kappa(x,z) = x^T B z$ where B is a psd matrix.

Feature map composition

P. Agius – L3, Spring 2008

The Kernel-based Approach

Nonlinear feature mapping

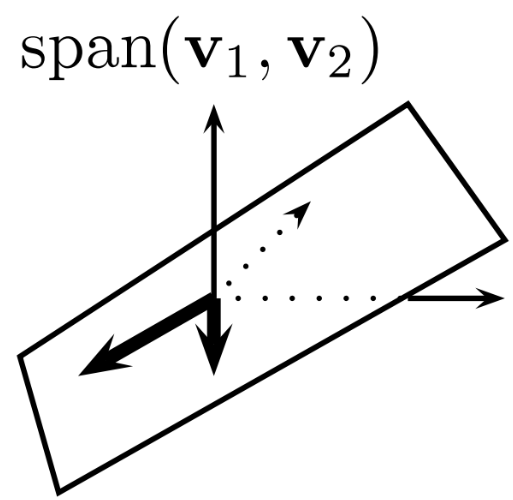
The Kernel trick

Dual representation

Dual representation

Consider the *linear span* of $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, i.e. the set of all vectors $\boldsymbol{w}$ of the form

$$\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{x}_i = \boldsymbol{X}^T \boldsymbol{\alpha}$$



Dual representation

Whenever

$$\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{x}_i = \boldsymbol{X}^T \boldsymbol{\alpha}$$

we shall refer to $\boldsymbol{\alpha}$ as the **dual representation** of $\boldsymbol{w}$.

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} =$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^{T}\boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} =$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^{T}\boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$
$$\langle \boldsymbol{w}, \boldsymbol{u} \rangle =$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T\boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$
$$\langle \boldsymbol{w}, \boldsymbol{u} \rangle = \boldsymbol{w}^T\boldsymbol{u} = \boldsymbol{\alpha}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\beta} = \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\beta}$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$
$$\langle \boldsymbol{w}, \boldsymbol{u} \rangle = \boldsymbol{w}^T \boldsymbol{u} = \boldsymbol{\alpha}^T \mathbf{X}\mathbf{X}^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}$$
$$\|\boldsymbol{w} - \boldsymbol{u}\|^2 =$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$
$$\langle \boldsymbol{w}, \boldsymbol{u} \rangle = \boldsymbol{w}^T \boldsymbol{u} = \boldsymbol{\alpha}^T \mathbf{X}\mathbf{X}^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}$$
$$\|\boldsymbol{w} - \boldsymbol{u}\|^2 = \langle \boldsymbol{w} - \boldsymbol{u}, \boldsymbol{w} - \boldsymbol{u} \rangle = \cdots$$

Dual coordinates

Let

$$\boldsymbol{w} = \mathbf{X}^T \boldsymbol{\alpha}$$
$$\boldsymbol{u} = \mathbf{X}^{\mathrm{T}} \boldsymbol{\beta}$$

Then

$$2\boldsymbol{w} = \mathbf{X}^{\mathrm{T}}(2\boldsymbol{\alpha})$$
$$\boldsymbol{w} + \boldsymbol{u} = \mathbf{X}^{\mathrm{T}}(\boldsymbol{\alpha} + \boldsymbol{\beta})$$
$$\langle \boldsymbol{w}, \boldsymbol{u} \rangle = \boldsymbol{w}^T \boldsymbol{u} = \boldsymbol{\alpha}^T \mathbf{X}\mathbf{X}^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}$$
$$\|\boldsymbol{w} - \boldsymbol{u}\|^2 = \langle \boldsymbol{w} - \boldsymbol{u}, \boldsymbol{w} - \boldsymbol{u} \rangle = \cdots$$

So what?

The Representer Theorem

The SVM weight vector lies in the span of the data points, i.e.

$$\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{x}_i = \boldsymbol{X}^T \boldsymbol{\alpha}$$

for some $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$

The Representer Theorem

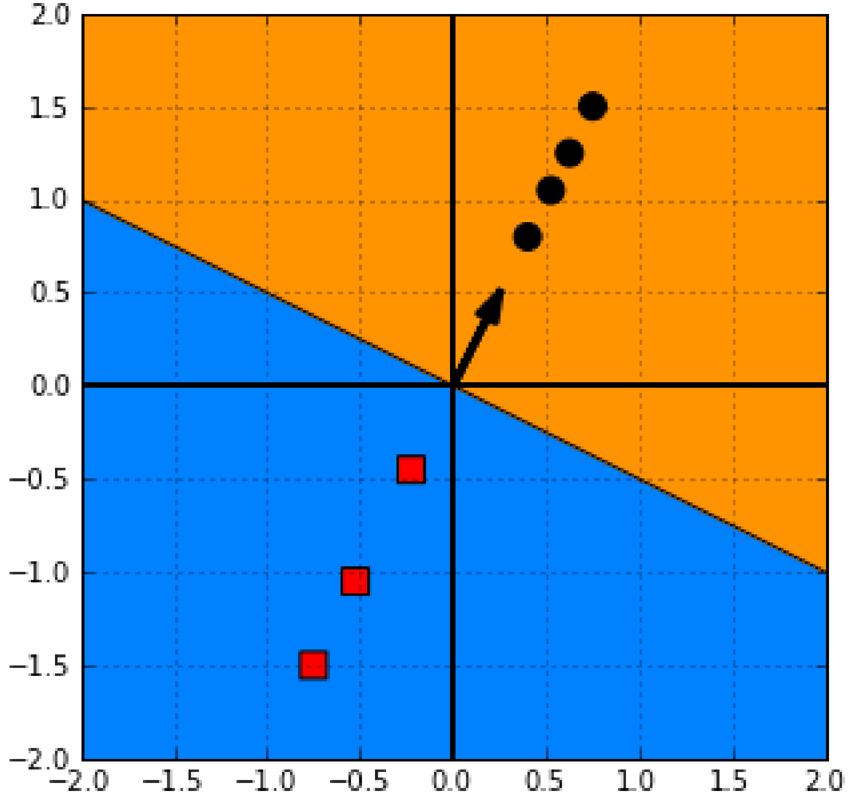
The SVM weight vector lies in the span of the data points, i.e.

$$\boldsymbol{w} = \sum_i \alpha_i \boldsymbol{x}_i = \boldsymbol{X}^T \boldsymbol{\alpha}$$

for some $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$

- Easy to prove
- Actually holds for **pretty much any linear model** with ℓ_2-penalty.

The Representer Theorem



Example

- Recall OLS regression:

$$\boldsymbol{w} = \underline{\hspace{5cm}}$$

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}$$

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}[(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}]$$

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}[(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}]$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}\boldsymbol{\alpha}$$

Kernelization

1. $\boldsymbol{x}_i \to \phi(\boldsymbol{x}_i)$

Nonlinear feature mapping

Kernelization

1. $\boldsymbol{x}_i \to \phi(\boldsymbol{x}_i)$

   Nonlinear feature mapping

2. $$\boldsymbol{w} = \sum_i \alpha_i \phi(\boldsymbol{x}_i)$$

   Dual representation

Kernelization

1. $\boldsymbol{x}_i \rightarrow \phi(\boldsymbol{x}_i)$ Nonlinear feature mapping
2. $\boldsymbol{w} = \sum_i \alpha_i \phi(\boldsymbol{x}_i)$ Dual representation
3.

$$f(\boldsymbol{z}) = \langle \boldsymbol{w}, \phi(\boldsymbol{z}) \rangle + w_0 = \left\langle \sum_i \alpha_i \phi(\boldsymbol{x}_i), \phi(\boldsymbol{z}) \right\rangle + w_0 =$$

$$= \sum_i \alpha_i \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{z}) \rangle + w_0$$

$$= \sum_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{z}) + w_0$$

The Kernel trick

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}[(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}]$$
$$\boldsymbol{w} = \boldsymbol{X}^{T}\boldsymbol{\alpha}$$

Example

- Recall OLS regression:

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$

$$\boldsymbol{w} = \boldsymbol{X}^{T}(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}$$

$$\boldsymbol{w} = \boldsymbol{X}^{T}[(\boldsymbol{X}\boldsymbol{X}^{T})^{+}\boldsymbol{y}]$$

$$\boldsymbol{w} = \boldsymbol{X}^{T}\boldsymbol{\alpha}$$

$$\boldsymbol{\alpha} = \boldsymbol{K}^{+}\boldsymbol{y} = \boldsymbol{K}^{-1}\boldsymbol{y}$$

Kernelization: Summary

- Take a linear learning algorithm

$$\boldsymbol{X}, \boldsymbol{y} \to \boldsymbol{w}, w_0$$

- Rewrite it to use only inner products of data points and return the dual representation of $\boldsymbol{w}$

$$\boldsymbol{K}, \boldsymbol{y} \to \boldsymbol{\alpha}, w_0$$

- Plug in any kernel and play!
- Congratulations, you've got a nonlinear model!

Quiz

The three ingredients of kernel methods:

- ____________
- ____________
- ____________

- Function/matrix K is a kernel function/matrix iff it is ____________
- Dual representation: ___ = ___ __

Quiz

Those algoritms have kernelized versions:

______________________________ . . .

Questions?