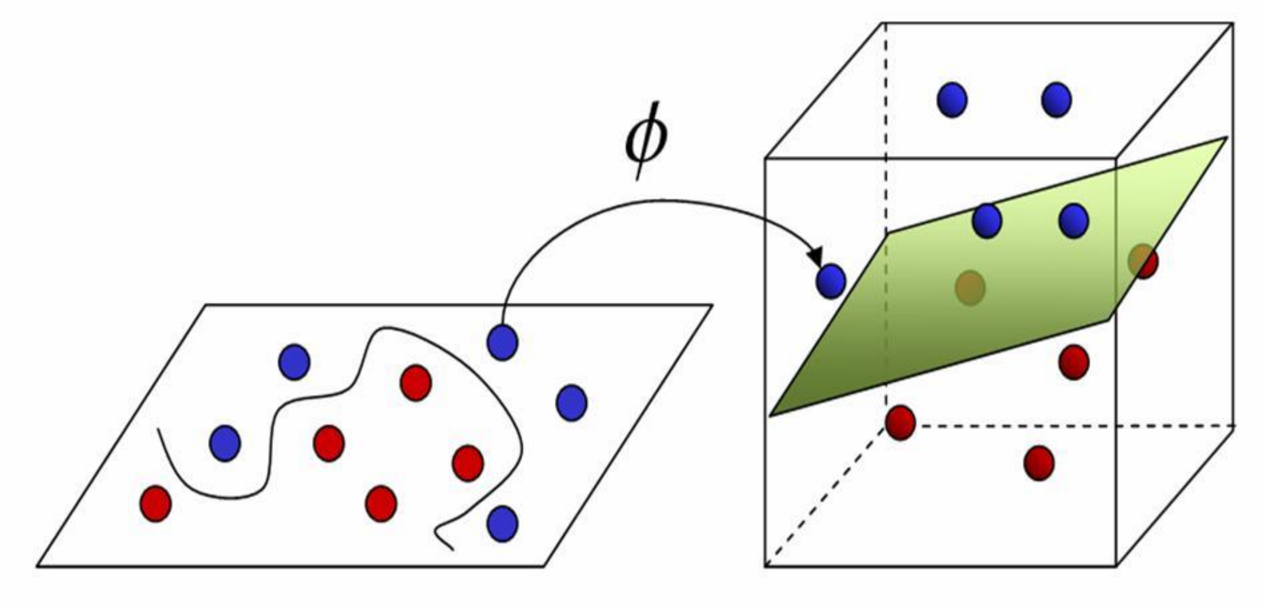


Kernel Methods

Konstantin Tretyakov (kt@ut.ee)

MTAT.03.227 Machine Learning

So far…

- Supervised machine learning
  - Linear models
  - Non-linear models
- Unsupervised machine learning
- Generic scaffolding

So far…

- Supervised machine learning
  - Linear models
    - Least squares regression, SVR
    - Fisher's discriminant, Perceptron, Logistic model, SVM
  - Non-linear models
    - Neural networks, Decision trees, Association rules

- Unsupervised machine learning
  - Clustering/EM, PCA
- Generic scaffolding
  - Probabilistic modeling, ML/MAP estimation
  - Performance evaluation, Statistical learning theory
  - Linear algebra, Optimization methods

Coming up next…

- Supervised machine learning
  - Linear models
    - Least squares regression, SVR
    - Fisher's discriminant, Perceptron, Logistic model, SVM
  - Non-linear models
    - Neural networks, Decision trees, Association rules
    - **Kernel-XXX**
- Unsupervised machine learning
  - Clustering/EM, PCA, **Kernel-XXX**
- Generic scaffolding
  - Probabilistic modeling, ML/MAP estimation
  - Performance evaluation, Statistical learning theory
  - Linear algebra, Optimization methods
  - **Kernels**

- Logistic regression, Perceptron, Max. margin, Fisher's discriminant, Linear regression, Ridge Regression, LASSO, …:

$$f(\boldsymbol{x}) =$$

- Logistic regression, Perceptron, Max. margin, Fisher's discriminant, Linear regression, Ridge Regression, LASSO, …:

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

- Logistic regression, Perceptron, Max. margin, Fisher's discriminant, Linear regression, Ridge Regression, LASSO, …:

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

- PCA, LDA, ICA, …:

$$f(\boldsymbol{x}) =$$

- Logistic regression, Perceptron, Max. margin, Fisher's discriminant, Linear regression, Ridge Regression, LASSO, …:

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

- PCA, LDA, ICA, …:

$$f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$$

- Logistic regression, Perceptron, Max. margin, Fisher's discriminant, Linear regression, Ridge Regression, LASSO, …:

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

- PCA, LDA, ICA, …:

$$f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$$

- K-means:

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

- CCA, GLM, …

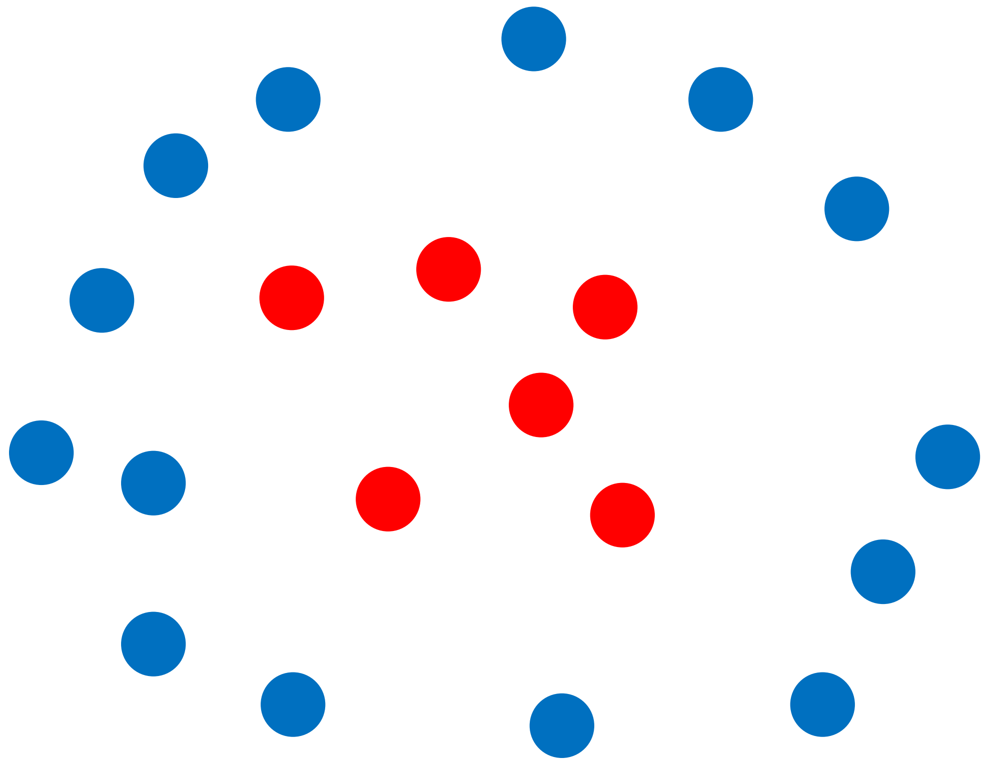
Too much linear

- Logistic regression, Perceptron, Max. margin, Fish dge Reg

- PC
- K-m
- CCA, GLM, …

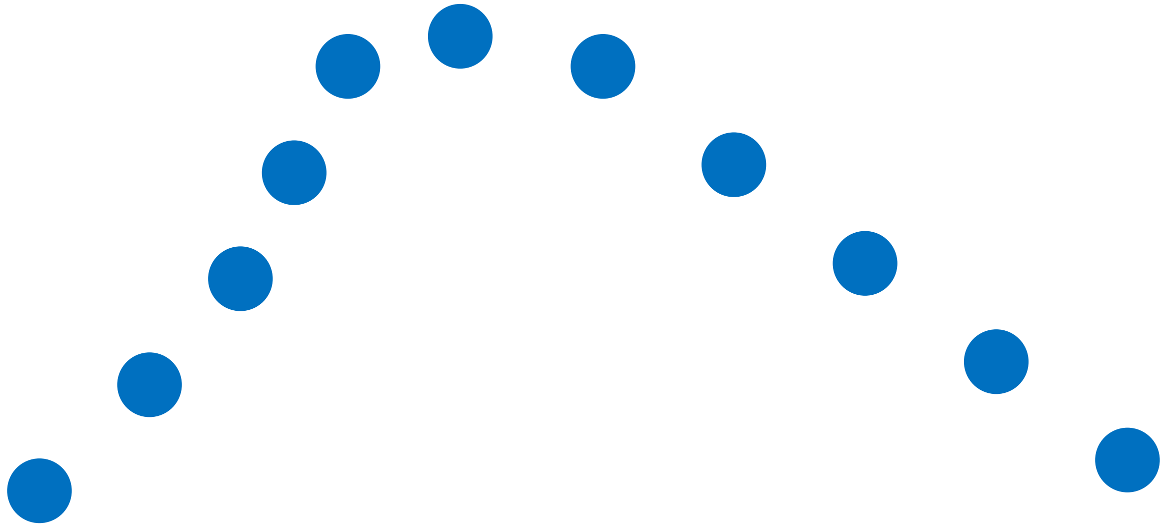
Linear is not enough

- Limited generalization ability



Linear is not enough

- Limited generalization ability



Linear is not enough

- Limited applicability

Text?

Ordinal/Nominal data?

Graphs/Trees/Networks?

Shapes?

Graph nodes?

Solutions

Solutions

- Feature space

- Kernels

Solutions

- Feature space
  - Nonlinear feature spaces

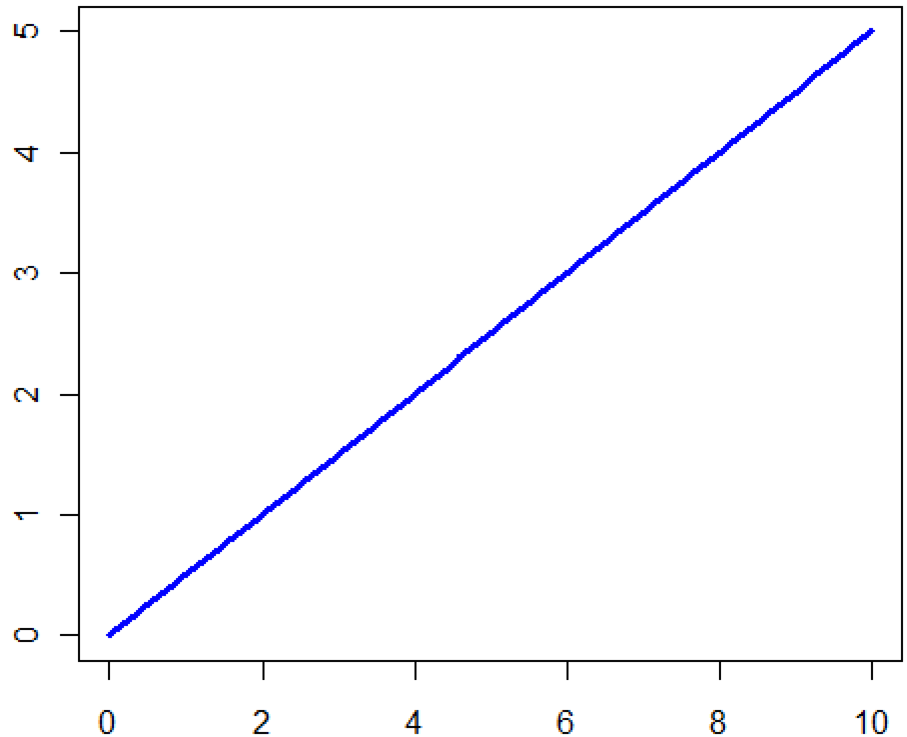
Important idea #1

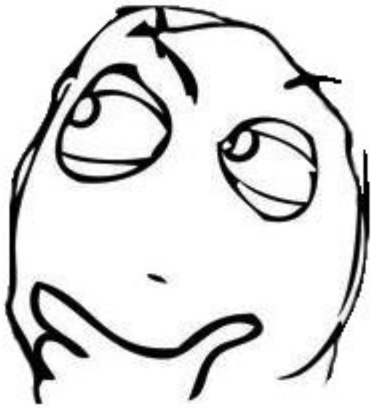
- Kernels
  - The Kernel Trick
  - Dual representation

Important idea #2

Important idea #3

$$f(x) = wx$$

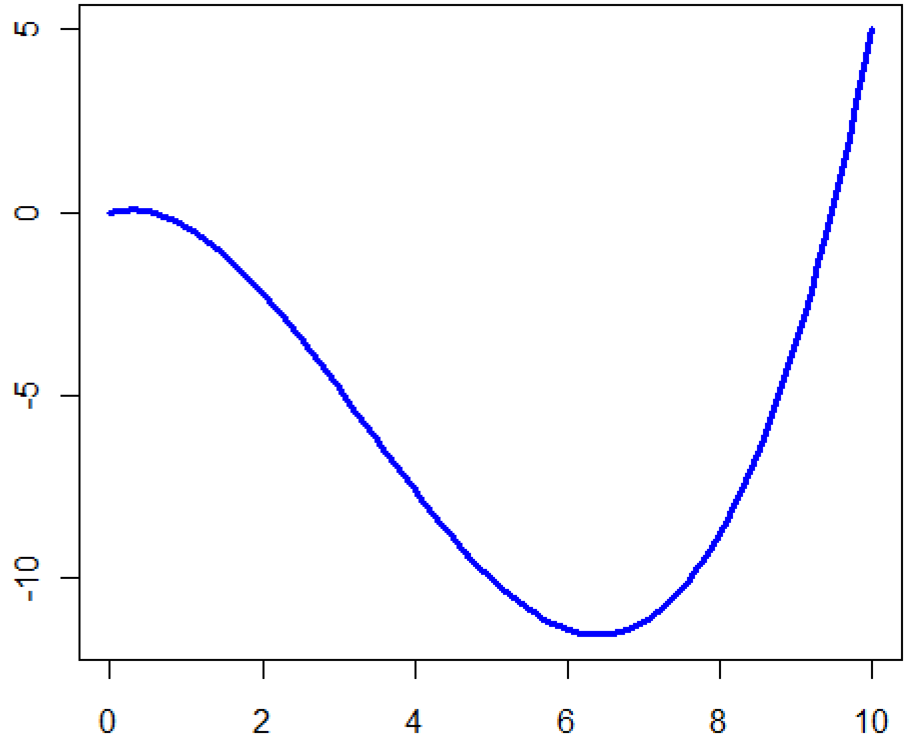


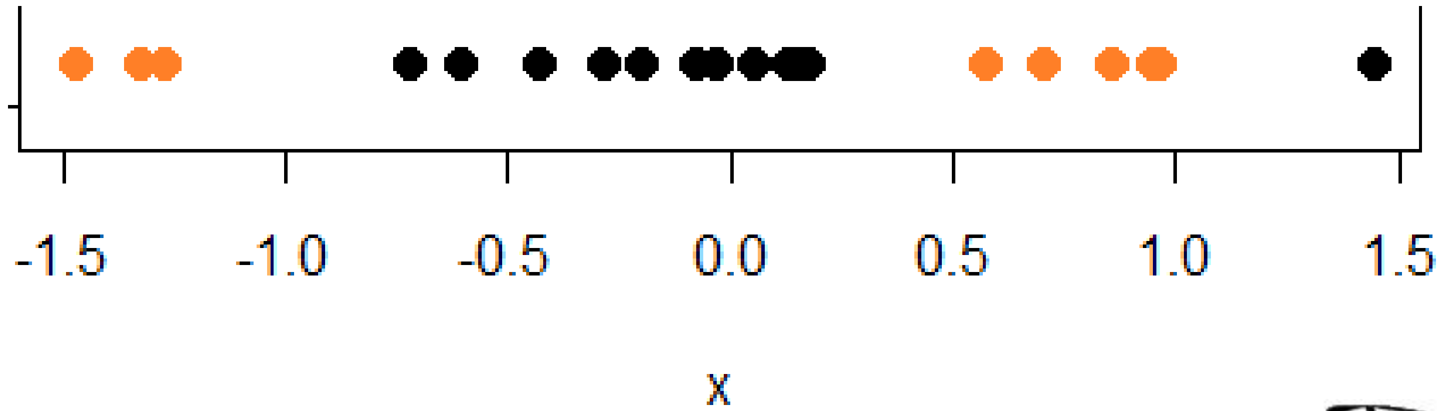


$$x \rightarrow x' \coloneqq \phi(x) \coloneqq (x, x^2, x^3)$$

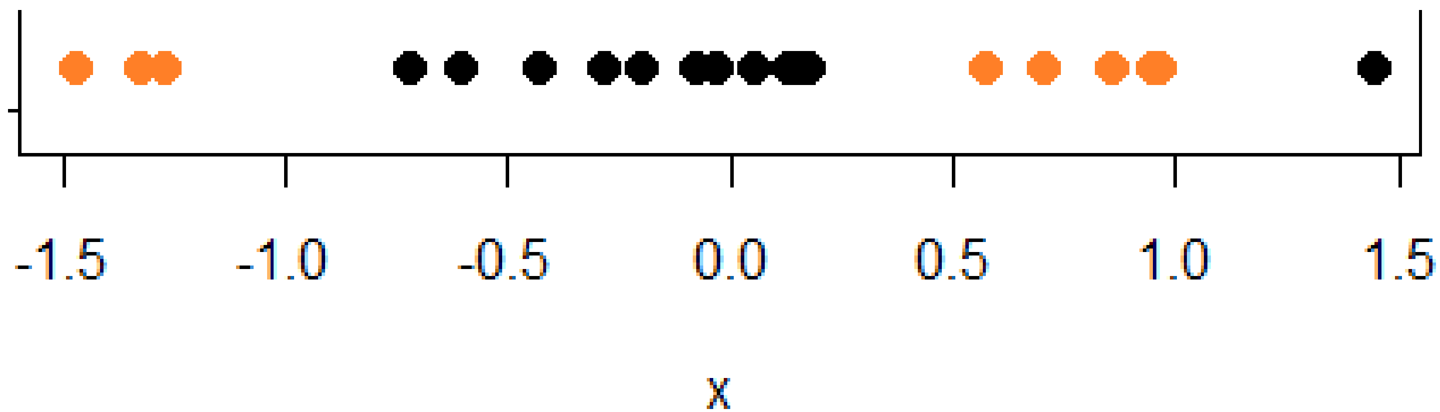
Nonlinear feature space

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

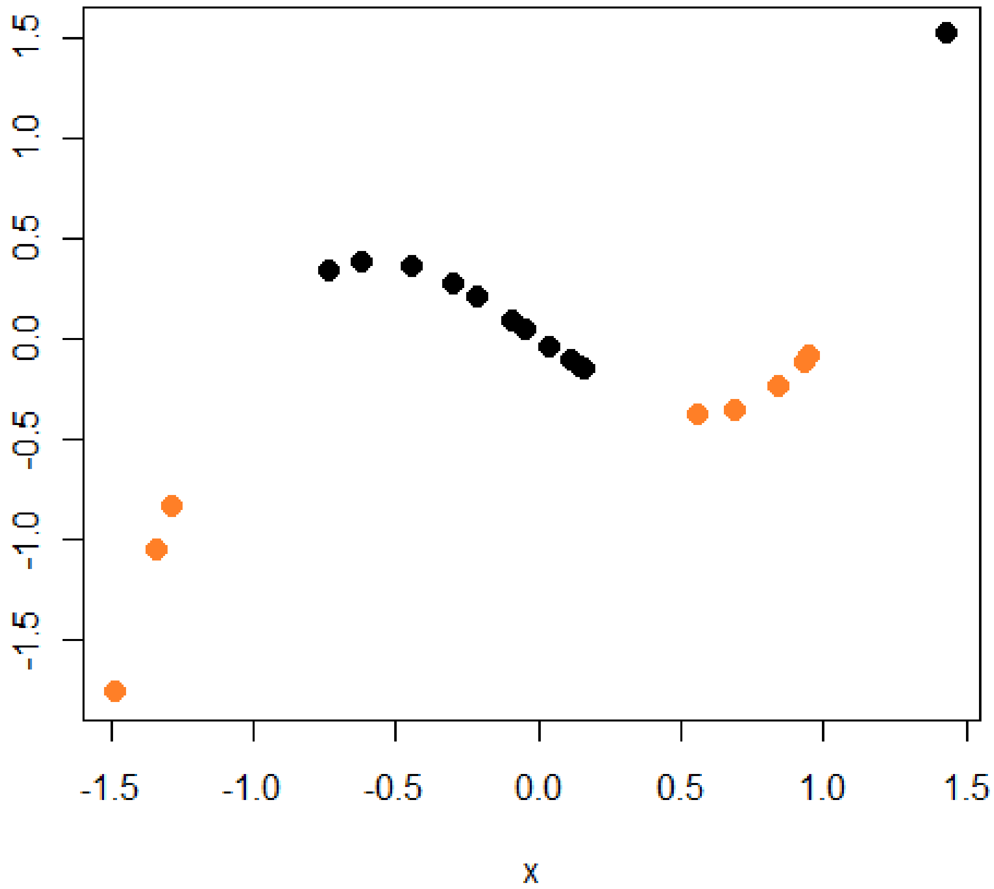




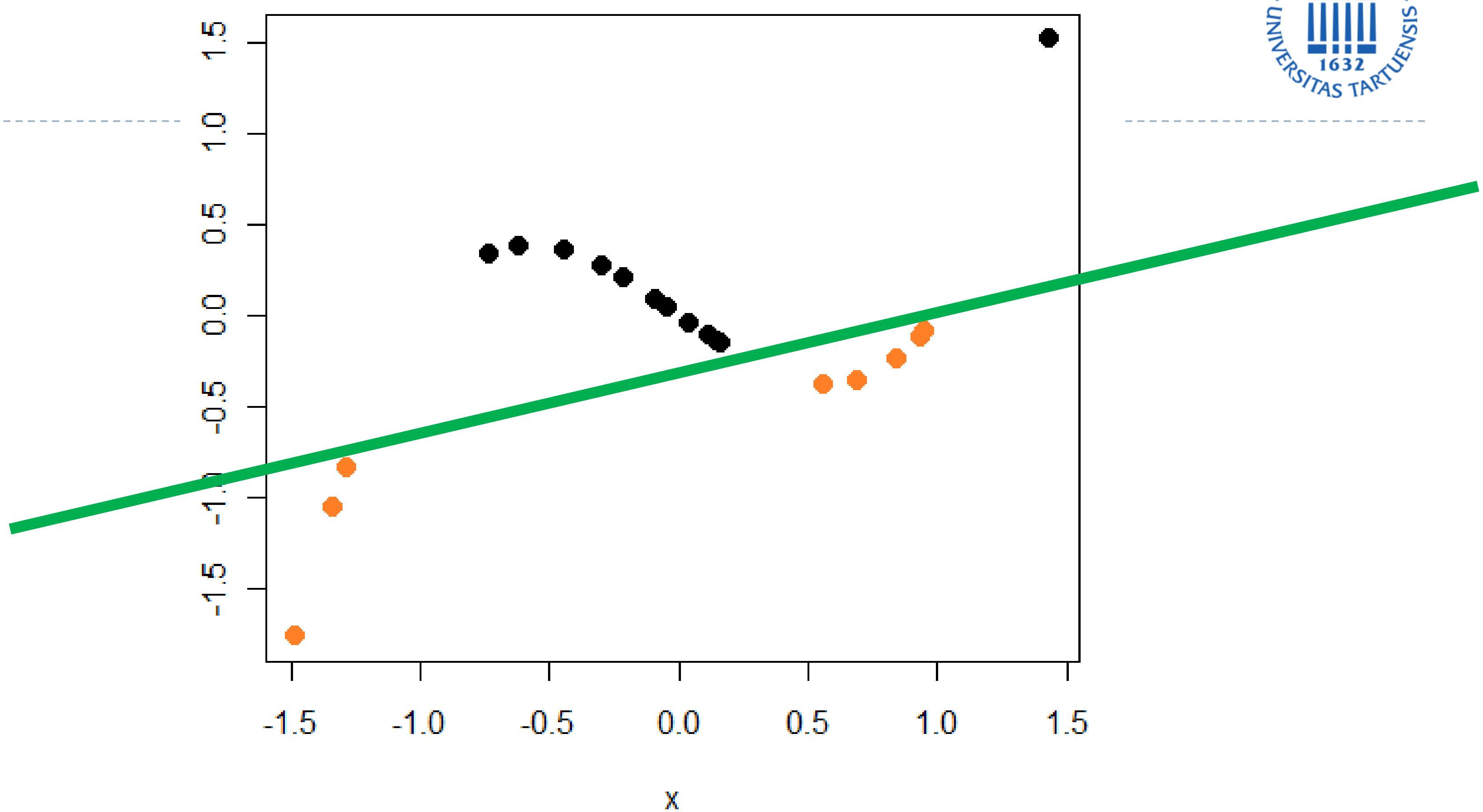
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
x



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



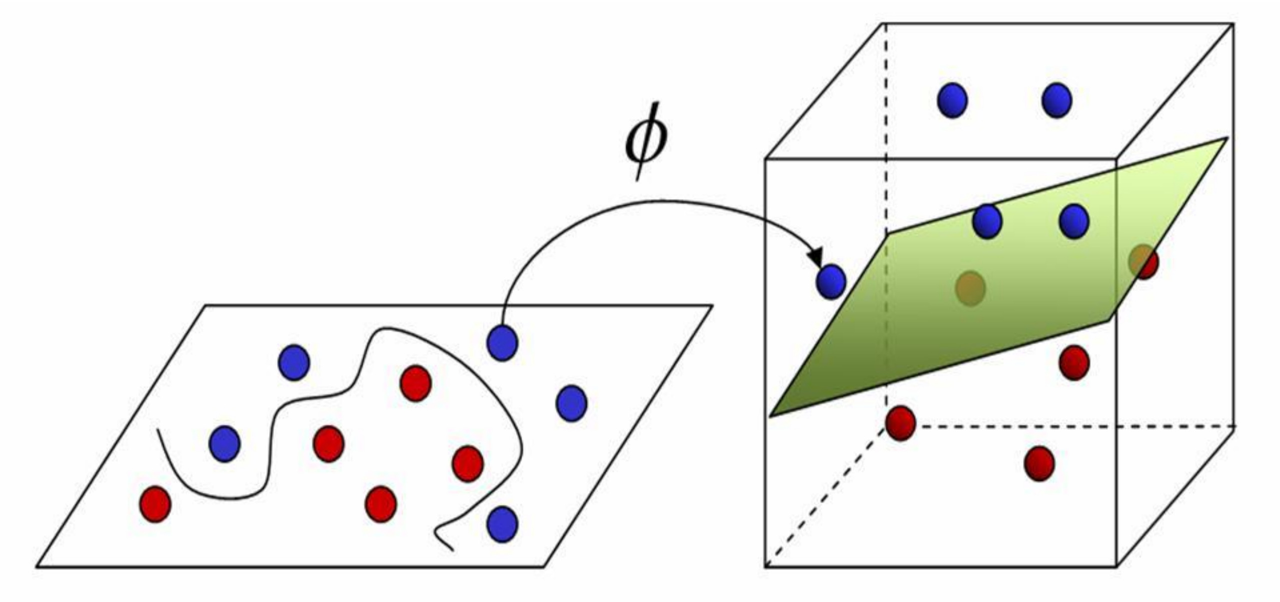
$$x \rightarrow \phi(x) = (x,\ x^3 - x)$$



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

Nonlinear feature space

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



+Support for arbitrary data types

$$\phi(\text{text}) = \text{word counts}$$

$$\phi(\text{graph}) = \text{node degrees}$$

$$\phi(\text{tree}) = \text{path lengths}$$

…

What if the dimensionality is high?

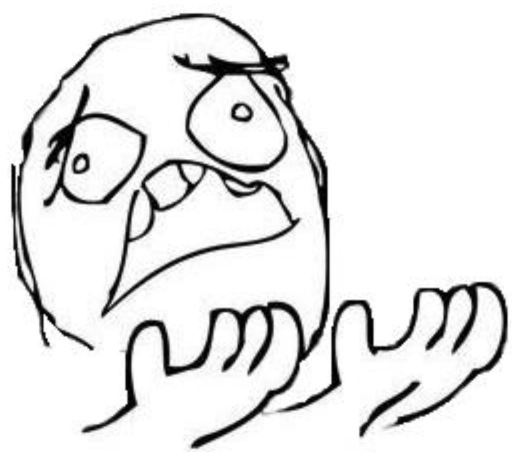
$$(x_1, x_2, \ldots, x_m) \to (x_1 x_1, x_1 x_2, \ldots, x_m x_m)$$

What if the dimensionality is high?

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

$O(m^2)$ elements

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



The Kernel Trick

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

- Consider

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

The Kernel Trick

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

- Consider

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

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

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

- Consider

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

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

$$= \sum_i x_i y_i \sum_j x_j y_j$$

The Kernel Trick

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

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

- Consider

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

$$= \left(\sum_i x_i y_i \right)^2 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

The Kernel Trick

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

- Consider

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

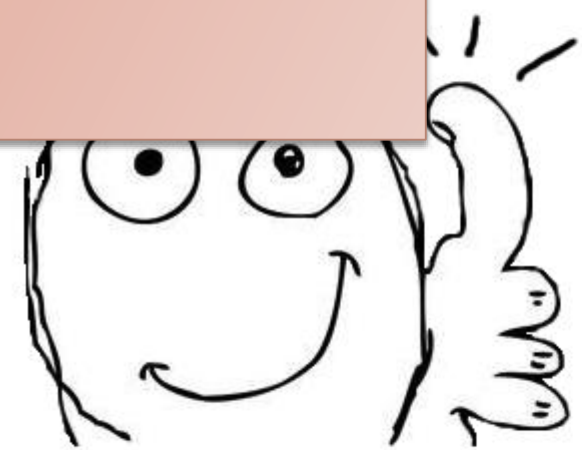
$$= \left(\sum_i x_i y_i \right)^2 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

The Kernel Trick

- Let
- Co

Polynomial kernel

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



The Kernel Trick

What about:

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

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:

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

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

$$= \langle \left(x_1, \dots, x_m, \sqrt{0.5}x_1x_1, \dots, \sqrt{0.5}x_mx_m\right), \left(y_1, \dots, y_m, \sqrt{0.5}y_1y_1, \dots, \sqrt{0.5}y_my_m\right)\rangle$$

What about:

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

The Kernel Trick

What about:

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

The Kernel Trick

What about:

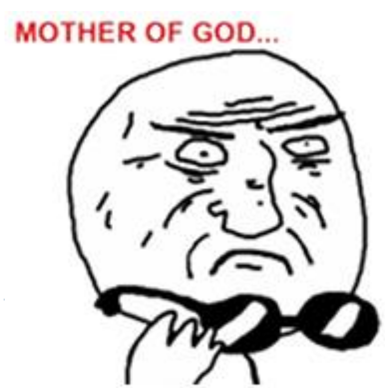
$$K(x,y) = \sum_{i=0}^{\infty} \frac{\langle x, y\rangle^i}{i!} = \exp\langle x, y\rangle$$

The Kernel Trick

What about:

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

Infinite-dimensional feature space!



The Kernel Trick

$K($ … $\rangle$

Gaussian 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

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

http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html

Kernels

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

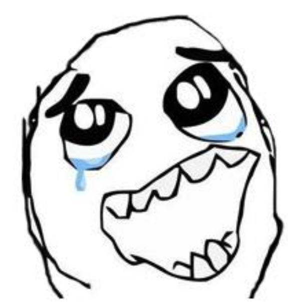
$$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$$

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

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

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 $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ 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!

Example?

Kernel theorem

- Not every function K is a kernel!

  $$\text{e.g. } K(x, y) = -1 \text{ is not}$$

- 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 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ 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
- $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$
- $\kappa(x, z) = x^T Bz$ where B is a psd matrix.

Feature map composition

P. Agius – L3, Spring 2008

Kernel normalization

- Let $\phi'(x) = \frac{\phi(x)}{\|\phi(x)\|}$
- Then

$$K'(x, y) = \langle \phi'(x), \phi'(y) \rangle =$$
$$\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(y)}{\|\phi(y)\|} \rangle = \frac{\langle \phi(x), \phi(y) \rangle}{\sqrt{\|\phi(x)\|^2 \|\phi(y)\|^2}} =$$
$$= \frac{K(x, y)}{\sqrt{K(x, x) K(y, y)}}$$

Kernel matrix normalization

- Then

$$K'(x,y) = \langle \phi'(x), \phi'(y) \rangle =$$

$$\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(y)}{\|\phi(y)\|} \rangle = \frac{\langle \phi(x), \phi(y) \rangle}{\sqrt{\|\phi(x)\|^2 \|\phi(y)\|^2}} =$$

$$= \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}}$$

$$K'_{ij} := \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}}$$

Kernel matrix centering

$$\boldsymbol{x}_i \to \boldsymbol{x}_i - \frac{1}{n}\sum_k \boldsymbol{x}_k$$

Kernel matrix centering

$$\boldsymbol{x}_i \to \boldsymbol{x}_i - \frac{1}{n}\sum_k \boldsymbol{x}_k$$

Kernel matrix centering

$$\boldsymbol{x}_i \to \boldsymbol{x}_i - \frac{1}{n}\sum_k \boldsymbol{x}_k$$

$$\boldsymbol{X} \to \boldsymbol{X} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\boldsymbol{X}$$

Kernel matrix centering

$$\boldsymbol{x}_i \to \boldsymbol{x}_i - \frac{1}{n}\sum_k \boldsymbol{x}_k$$

$$\boldsymbol{X} \to \boldsymbol{X} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\boldsymbol{X}$$

$$\boldsymbol{X}\boldsymbol{X}^T \to \left(\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\right)\left(\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\right)^T$$

Kernel matrix centering

$$\boldsymbol{x}_i \to \boldsymbol{x}_i - \frac{1}{n}\sum_k \boldsymbol{x}_k$$

$$\boldsymbol{X} \to \boldsymbol{X} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\boldsymbol{X}$$

$$\boldsymbol{X}\boldsymbol{X}^T \to \left(\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\right)\left(\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\right)^T$$

$$\boldsymbol{X}\boldsymbol{X}^T$$

$$\to \boldsymbol{X}\boldsymbol{X}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\boldsymbol{X}^T - \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T\mathbf{1}\mathbf{1}^T$$

$$+ \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\boldsymbol{X}^T\mathbf{1}\mathbf{1}^T$$

Kernel matrix centering

$$\boldsymbol{X}\boldsymbol{X}^T$$
$$\to \boldsymbol{X}\boldsymbol{X}^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\boldsymbol{X}^T - \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^T\mathbf{1}\mathbf{1}^T$$
$$+ \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\boldsymbol{X}\boldsymbol{X}^T\mathbf{1}\mathbf{1}^T$$

$$\boldsymbol{K}_{\text{cent}}$$
$$:= \boldsymbol{K} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\boldsymbol{K} - \frac{1}{n}\boldsymbol{K}\mathbf{1}\mathbf{1}^T + \frac{1}{n^2}\mathbf{1}\mathbf{1}^T\boldsymbol{K}\mathbf{1}\mathbf{1}^T$$

The Dual Representation

- Let A be the input space, and let B be the higher-dimensional feature space.
- Let $\phi: A \to B$ be the feature map.
- Fix a dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \subset A$
- Let $w = \sum_i \alpha_i \phi(\boldsymbol{x}_i) \in B$
- We say that α_i are the *dual coordinates* for w.

Dual coordinates

$$\boldsymbol{w} = \sum_i \alpha_i \phi(\boldsymbol{x}_i) = \phi(\boldsymbol{X})^T \boldsymbol{\alpha} = \boldsymbol{\Xi}^T \boldsymbol{\alpha}$$

Note that $\boldsymbol{\Xi}\boldsymbol{\Xi}^T = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T = \boldsymbol{K}$

Now we can do all of the useful stuff using dual coordinates only.

Dual coordinates

Let

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

Then

$$2\boldsymbol{w} =$$

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Dual coordinates

Let

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

Then

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

Kernelization

- Recall the Perceptron:

Kernelization

- Recall the Perceptron:

  - Initialize $\boldsymbol{w} := \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
  - Update weights:
    - $\boldsymbol{w} := \boldsymbol{w} + \mu y_i \boldsymbol{x_i}$
    - $b := b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{w} \coloneqq \mathbf{0} \Leftrightarrow \boldsymbol{\alpha} \coloneqq \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
  - Update weights:
    - $\boldsymbol{w} \coloneqq \boldsymbol{w} + \mu y_i \boldsymbol{x_i}$
    - $b \coloneqq b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{w} := \mathbf{0} \Leftrightarrow \boldsymbol{\alpha} := \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
  - Update weights:
    - $\boldsymbol{w} := \boldsymbol{w} + \mu y_i \boldsymbol{x_i} \Leftrightarrow \alpha_i := \alpha_i + \mu y_i$
    - $b := b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{\alpha} := \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
  - Update weights:
    - $\alpha_i := \alpha_i + \mu y_i$
    - $b := b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{\alpha} \coloneqq \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
    - $\boldsymbol{w}^T \boldsymbol{x}_i + b \neq y_i \Leftrightarrow \sum_j \alpha_j \boldsymbol{x}_j^T \boldsymbol{x}_i + b \neq y_i$
  - Update weights:
    - $\alpha_i \coloneqq \alpha_i + \mu y_i$
    - $b \coloneqq b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{\alpha} \coloneqq \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
    - $\boldsymbol{w}^T \boldsymbol{x}_i + b \neq y_i \Leftrightarrow \boldsymbol{K}_i \boldsymbol{\alpha} + b \neq y_i$
  - Update weights:
    - $\alpha_i \coloneqq \alpha_i + \mu y_i$
    - $b \coloneqq b + \mu y_i$

Kernelization

- Recall the Perceptron:
  - Initialize $\boldsymbol{\alpha} \coloneqq \mathbf{0}$
  - Find a misclassified example (x_i, y_i)
    - $\boldsymbol{K}_i\boldsymbol{\alpha} + b \neq y_i$
  - Update weights:
    - $\alpha_i \coloneqq \alpha_i + \mu y_i$
    - $b \coloneqq b + \mu y_i$

Quiz

Today we heard three important ideas

- Important idea #1: ____________
- Important idea #2: ____________
- Important idea #3: ____________

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

Quiz

Those algoritms have kernelized versions:

______________________________ ...