

Kernel Methods

Konstantin Tretyakov (kt@ut.ee)

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, ...:

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

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

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

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

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

- ▶ PCA, LDA, ICA,:

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

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

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

- ▶ PCA, LDA, ICA,:

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

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

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

- ▶ PCA, LDA, ICA, ...:

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

- ▶ K-means:

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

- ▶ CCA, GLM, ...

Too much linear

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

- ▶ PCA

- ▶ K-m

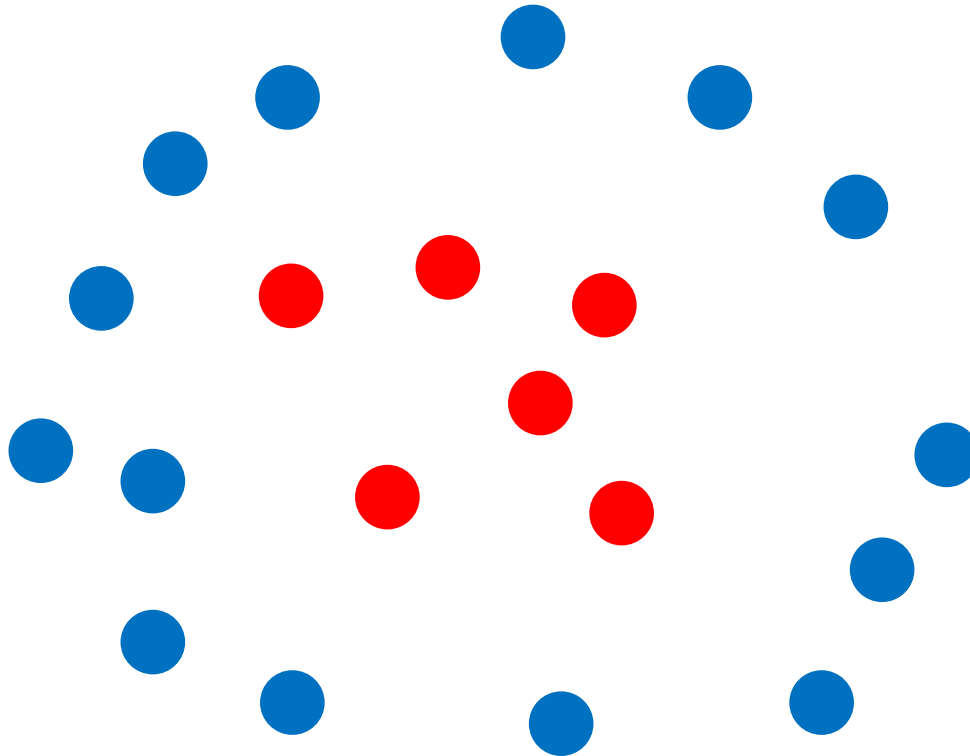


- ▶ CCA, GLM, ...



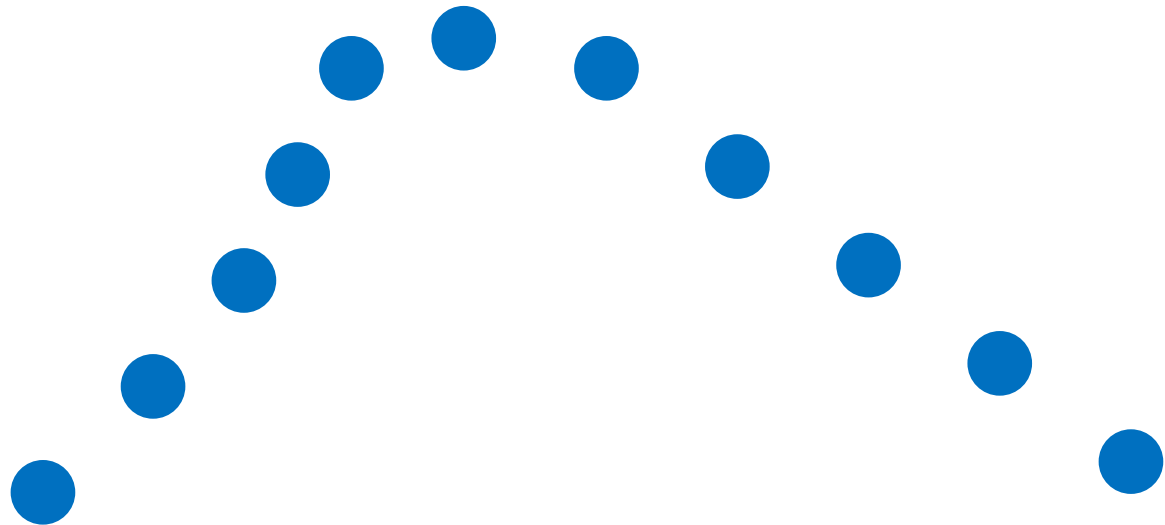
Linear is not enough

- ▶ 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

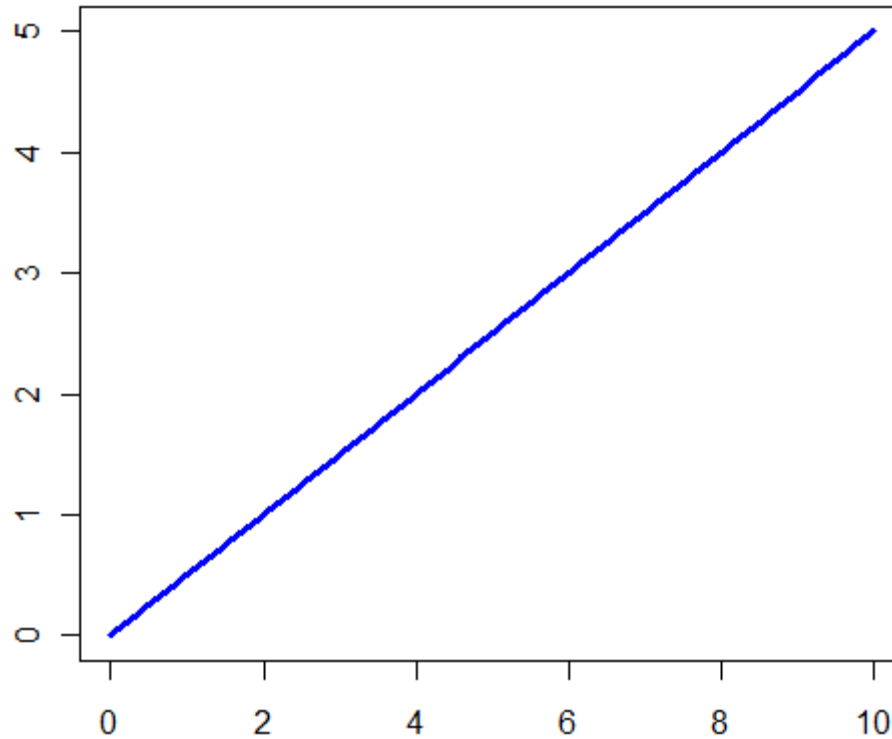


Important idea #2



Important idea #3

$$f(x) = wx$$

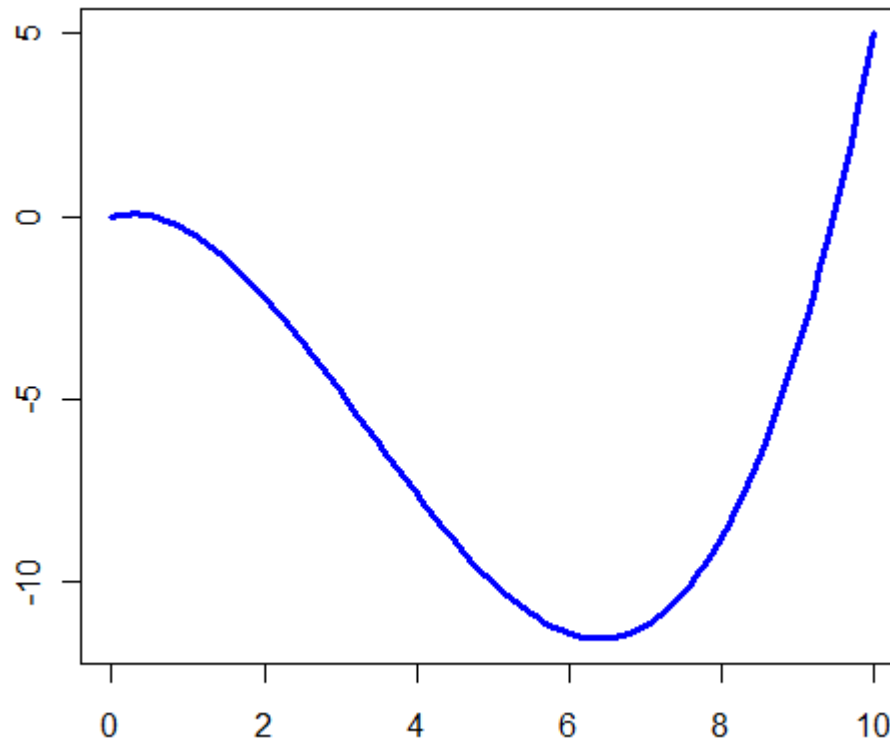


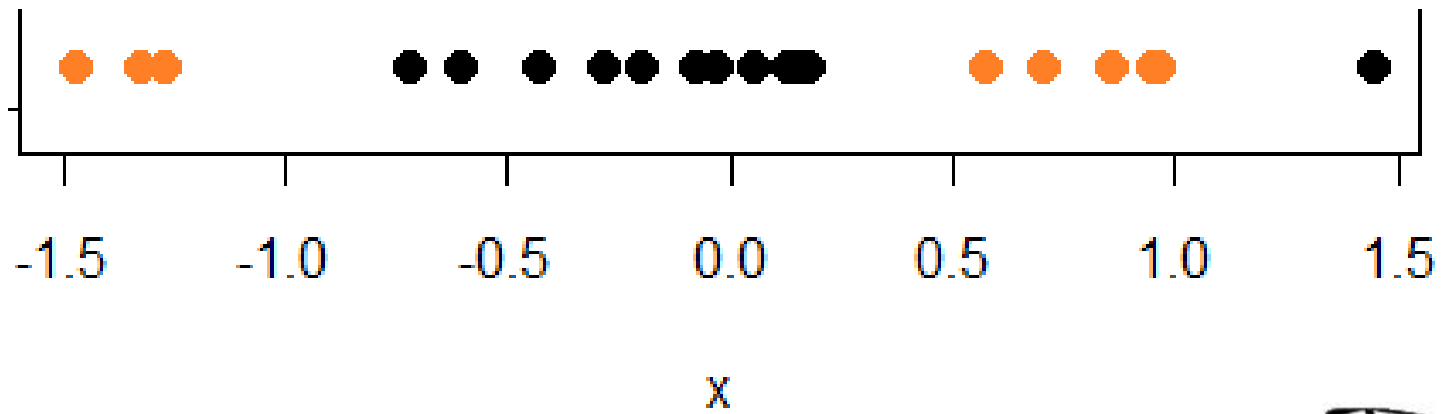
$$x \rightarrow x' := \phi(x) := (x, x^2, x^3)$$

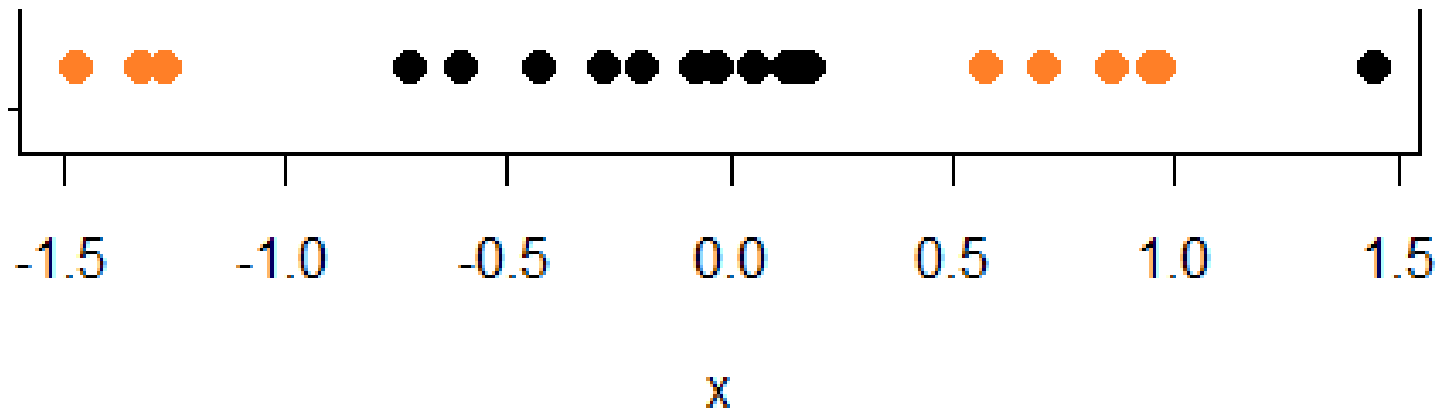


Nonlinear feature space

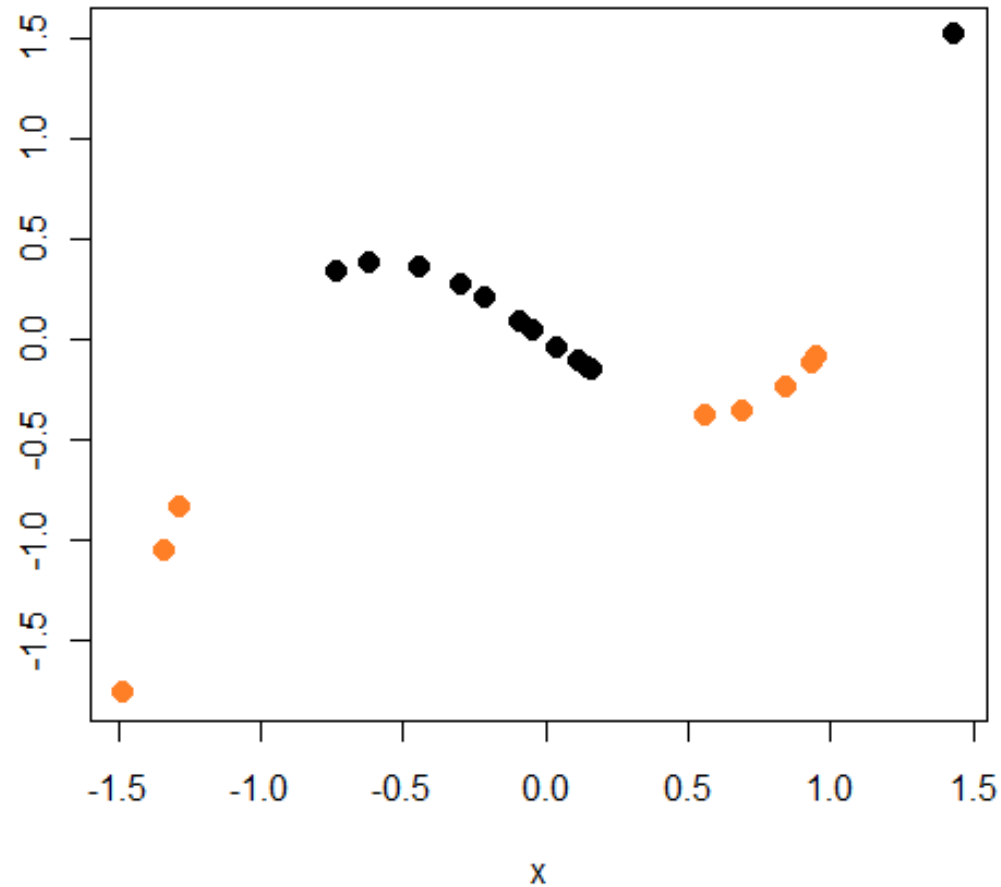
$$x \rightarrow x' := \phi(x) := (x, x^2, x^3)$$
$$f(x') = w_1x + w_2x^2 + w_3x^3$$



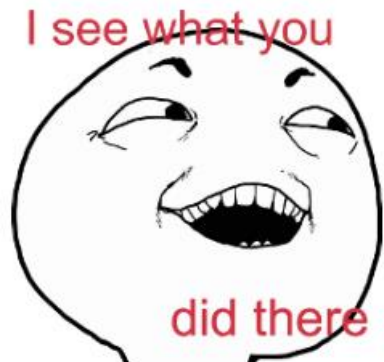
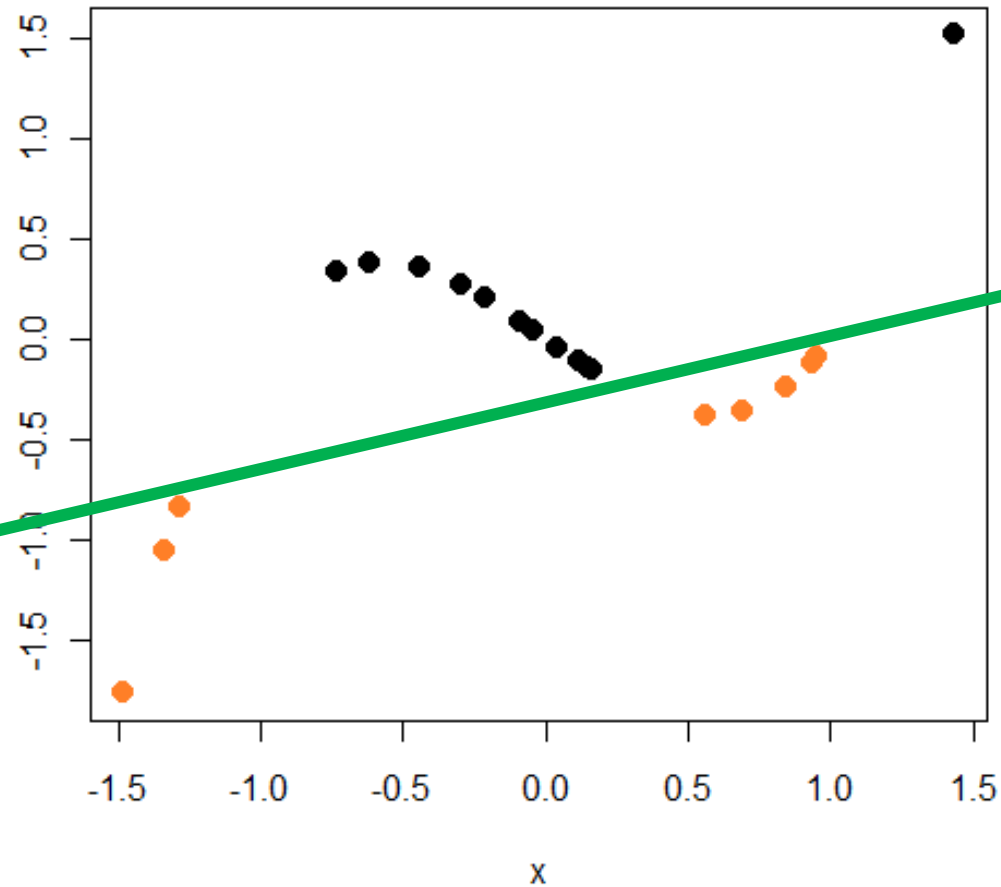




$$x \rightarrow \phi(x) = (x, x^3 - x)$$



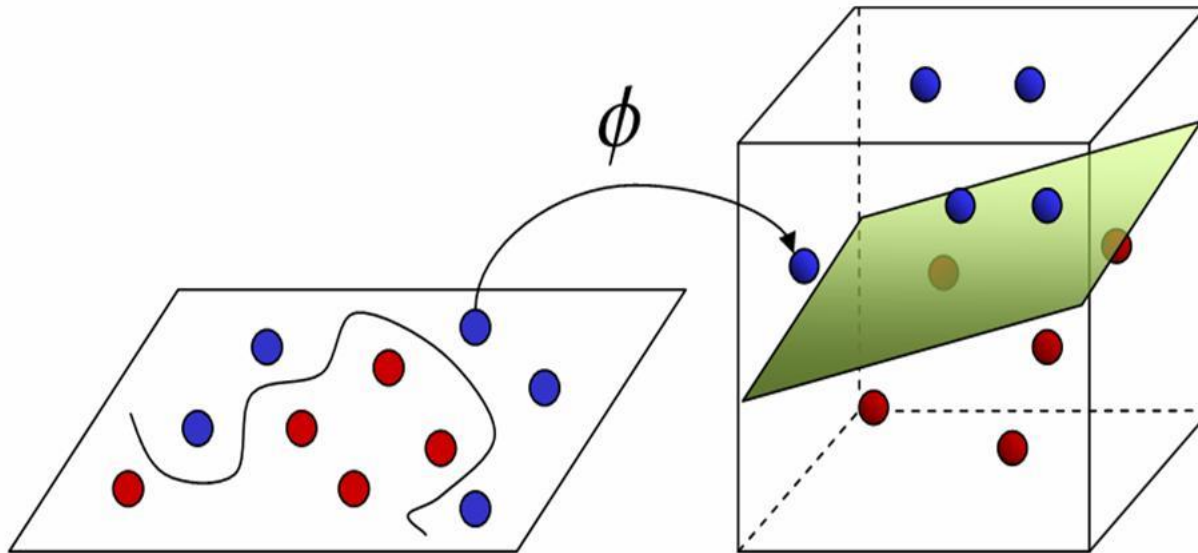
$$x \rightarrow \phi(x) = (x, x^3 - x)$$



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

Nonlinear feature space

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



+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?

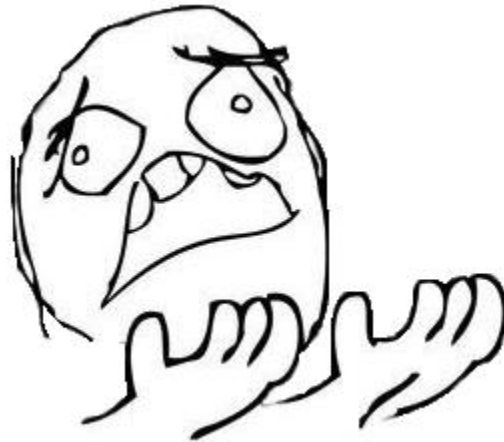
$$(x_1, x_2, \dots, x_m) \rightarrow (x_1 x_1, x_1 x_2, \dots, x_m x_m)$$

What if the dimensionality is high?

$$(x_1, x_2, \dots, x_m) \rightarrow (x_1x_1, x_1x_2, \dots, x_mx_m)$$

$O(m^2)$ elements

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



The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

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

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= \sum_{ij} \phi(\mathbf{x})_{ij} \phi(\mathbf{y})_{ij} \\ &= \sum_{ij} x_i x_j y_i y_j\end{aligned}$$

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= \sum_{ij} \phi(\mathbf{x})_{ij} \phi(\mathbf{y})_{ij} \\ &= \sum_{ij} x_i x_j y_i y_j = \sum_{ij} x_i y_i x_j y_j\end{aligned}$$

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= \sum_{ij} \phi(\mathbf{x})_{ij} \phi(\mathbf{y})_{ij} \\ &= \sum_{ij} x_i x_j y_i y_j = \sum_{ij} x_i y_i x_j y_j \\ &= \sum_i x_i y_i \sum_j x_j y_j\end{aligned}$$

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

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

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

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

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= \sum_{ij} \phi(\mathbf{x})_{ij} \phi(\mathbf{y})_{ij} \\ &= \left(\sum_i x_i y_i \right)^2 = \langle \mathbf{x}, \mathbf{y} \rangle^2\end{aligned}$$

The Kernel Trick

▶ Let $\phi(\mathbf{x}) = (x_1x_1, x_1x_2, \dots, x_mx_m)$

▶ Consider

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

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



The Kernel Trick

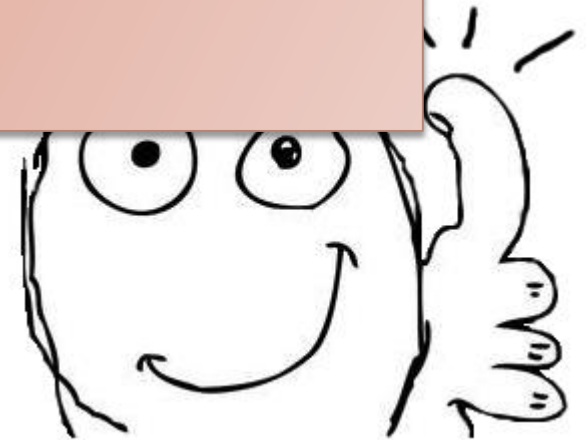
▶ Let

▶ Con

Polynomial kernel

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

($\underbrace{\quad}_i$)



The Kernel Trick

What about:

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

The Kernel Trick

What about:

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle + 0.5 \langle \mathbf{x}, \mathbf{y} \rangle^2$$
$$= \sum_i x_i y_i + 0.5 \sum_{ij} \phi_{ij}(\mathbf{x}) \phi_{ij}(\mathbf{y})$$

The Kernel Trick

What about:

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

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

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

The Kernel Trick



What about:

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

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

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

The Kernel Trick

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:

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

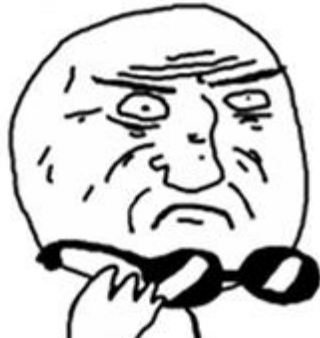
The Kernel Trick

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!

MOTHER OF GOD...



The Kernel Trick

Gaussian kernel

$K(x, y)$

$$K(x, y) =$$

$$= \exp(-\gamma \|x - y\|^2)$$

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



The Kernel Trick

$K(x, y)$ **Exponential kernel** \rangle

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





Kernels

$$\begin{aligned}
 k(x, y) &= x^T y + c & k(x, y) &= \sqrt{\|x - y\|^2 + c^2} & k(x, y) &= \frac{\theta}{\|x - y\|} \sin \frac{\|x - y\|}{\theta} \\
 k(x, y) &= (\alpha x^T y + c)^d & k(x, y) &= \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) & k(x, y) &= \frac{1}{\sqrt{\|x - y\|^2 + c^2}} \\
 k(x, y) &= \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right) & k(x, y) &= \frac{2}{\pi} \arccos\left(-\frac{\|x - y\|}{\sigma}\right) - \frac{2}{\pi} \frac{\|x - y\|}{\sigma} \sqrt{1 - \left(\frac{\|x - y\|}{\sigma}\right)^2} \\
 k(x, y) &= \sum_{k=1}^n \exp(-\sigma(x^k - y^k)^2) & k(x, y) &= 1 - \frac{3\|x - y\|}{2\sigma} + \frac{1}{2} \left(\frac{\|x - y\|}{\sigma}\right)^3 & k(x, y) &= -\log(\|x - y\|^d + 1) \\
 k(x, y) &= \tanh(\alpha x^T y + c) & k(x, y) &= -\|x - y\|^d & k(x, y) &= \frac{1}{1 + \frac{\|x - y\|^2}{c}} \\
 k(x, y) &= 1 - \frac{\|x - y\|^2}{\|x - y\|^2 + c} & k(x, y) &= 1 + xy + xy \min(x, y) - \frac{x + y}{2} \min(x, y)^2 + \frac{1}{3} \min(x, y)^3 \\
 k(x, y) &= B_{2p+1}(x - y) & k(x, y) &= \sum_{i=1}^n \min(x_i, y_i) & k(x, y) &= \prod_{i=1}^N h\left(\frac{x_i - c}{a}\right) h\left(\frac{y_i - c}{a}\right) \\
 k(x, y) &= \sum_{i=1}^m \min(|x_i|^\alpha, |y_i|^\beta) & k(x, y) &= \frac{J_{v+1}(\sigma\|x - y\|)}{\|x - y\|^{-n(v+1)}} & k(x, y) &= 1 - \sum_{i=1}^n \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)} \\
 \kappa_l(a, b) &= \sum_{c \in \{0:1\}} P(Y = c | X_l = a) P(Y = c | X_l = b) & k(x, y) &= \frac{1}{1 + \|x - y\|^d}
 \end{aligned}$$



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 $K(\mathbf{x}, \mathbf{y})$ is a *kernel*, if

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

for some *feature map* ϕ .

Kernel matrix

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

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

is called the *kernel matrix*.

Kernel matrix

- ▶ Let X be the data matrix, then

$$K = XX^T$$

is the kernel matrix for the *linear* kernel

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$$

Kernel matrix

- ▶ Let X be the data matrix, then

$$K = XX^T$$

is the kernel matrix for the *linear* kernel

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$$

- ▶ Let ϕ be a feature mapping. Then*

$$K = \phi(X)\phi(X)^T$$

is the kernel matrix for the corresponding kernel $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$.

Kernel theorem

- ▶ Not every function K is a kernel!

Example?

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 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 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
- ★ $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$
- ★ $\kappa(x, z) = x^T Bz$ 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

- ★ $\kappa(x, z) = \kappa_1(x, z) + \kappa_2(x, z)$
- ★ $\kappa(x, z) = \alpha \kappa_1(x, z)$
- ★ $\kappa(x, z) = \kappa_1(x, z) \kappa_2(x, z)$
- ★ $\kappa(x, z) = f(x) f(z)$ 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 space scaling

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 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



Kernel normalization

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

▶ Then

$$\begin{aligned} K'(x, y) &= \langle \phi'(x), \phi'(y) \rangle = \\ &= \left\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(y)}{\|\phi(y)\|} \right\rangle = \frac{\langle \phi(x), \phi(y) \rangle}{\sqrt{\|\phi(x)\|^2 \|\phi(y)\|^2}} = \\ &= \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}} \end{aligned}$$

Kernel matrix normalization

► Then

$$\begin{aligned} K'(x, y) &= \langle \phi'(x), \phi'(y) \rangle = \\ &\left\langle \frac{\phi(x)}{\|\phi(x)\|}, \frac{\phi(y)}{\|\phi(y)\|} \right\rangle = \frac{\langle \phi(x), \phi(y) \rangle}{\sqrt{\|\phi(x)\|^2 \|\phi(y)\|^2}} = \\ &= \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}} \end{aligned}$$

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

Kernel matrix centering

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \frac{1}{n} \sum_k \mathbf{x}_k$$

Kernel matrix centering

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \frac{1}{n} \sum_k \mathbf{x}_k$$

CHALLENGE ACCEPTED



Kernel matrix centering

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \frac{1}{n} \sum_k \mathbf{x}_k$$

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

Kernel matrix centering

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \frac{1}{n} \sum_k \mathbf{x}_k$$

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

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

Kernel matrix centering

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \frac{1}{n} \sum_k \mathbf{x}_k$$

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

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

$$\mathbf{X}\mathbf{X}^T$$

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

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

Kernel matrix centering

$$XX^T$$

$$\rightarrow XX^T - \frac{1}{n} \mathbf{1}\mathbf{1}^T XX^T - \frac{1}{n} XX^T \mathbf{1}\mathbf{1}^T + \frac{1}{n^2} \mathbf{1}\mathbf{1}^T XX^T \mathbf{1}\mathbf{1}^T$$

$$K_{\text{cent}}$$

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



The Dual Representation

- ▶ Let A be the input space, and let B be the higher-dimensional feature space.
- ▶ Let $\phi: A \rightarrow B$ be the feature map.
- ▶ Fix a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset A$

- ▶ Let $w = \sum_i \alpha_i \phi(\mathbf{x}_i) \in B$

- ▶ We say that α_i are the *dual coordinates* for w .



Dual coordinates

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

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

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

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

$$2\mathbf{w} =$$

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

Dual coordinates

Let

$$\mathbf{w} = \mathbf{E}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{E}^T \boldsymbol{\beta}$$

Then

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

$$\mathbf{w} + \mathbf{u} =$$

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

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

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

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

$$\langle \mathbf{w}, \mathbf{u} \rangle =$$

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

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

$$\langle \mathbf{w}, \mathbf{u} \rangle = \mathbf{w}^T \mathbf{u} = \boldsymbol{\alpha}^T \mathbf{\Xi} \mathbf{\Xi}^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\beta}$$

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

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

$$\langle \mathbf{w}, \mathbf{u} \rangle = \mathbf{w}^T \mathbf{u} = \boldsymbol{\alpha} \mathbf{\Xi} \mathbf{\Xi}^T \boldsymbol{\beta} = \boldsymbol{\alpha} \mathbf{K} \boldsymbol{\beta}$$

$$\|\mathbf{w} - \mathbf{u}\|^2 =$$

Dual coordinates

Let

$$\mathbf{w} = \mathbf{\Xi}^T \boldsymbol{\alpha}$$

$$\mathbf{u} = \mathbf{\Xi}^T \boldsymbol{\beta}$$

Then

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

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

$$\langle \mathbf{w}, \mathbf{u} \rangle = \mathbf{w}^T \mathbf{u} = \boldsymbol{\alpha} \mathbf{\Xi} \mathbf{\Xi}^T \boldsymbol{\beta} = \boldsymbol{\alpha} \mathbf{K} \boldsymbol{\beta}$$

$$\|\mathbf{w} - \mathbf{u}\|^2 = \mathbf{w}^T \mathbf{w} + \mathbf{u}^T \mathbf{u} - 2\mathbf{w}^T \mathbf{u} = \dots$$

Dual coordinates

Let

$$\begin{aligned} \mathbf{w} &= \mathbf{\Xi}^T \boldsymbol{\alpha} \\ \mathbf{u} &= \mathbf{\Xi}^T \boldsymbol{\beta} \end{aligned}$$

Then

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

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

$$\langle \mathbf{w}, \mathbf{u} \rangle = \mathbf{w}^T \mathbf{u} = \boldsymbol{\alpha}^T \mathbf{\Xi} \mathbf{\Xi}^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\beta}$$

$$\|\mathbf{w} - \mathbf{u}\|^2 = \mathbf{w}^T \mathbf{w} + \mathbf{u}^T \mathbf{u} - 2\mathbf{w}^T \mathbf{u} = \dots$$



Kernelization

- ▶ Recall the Perceptron:

Kernelization

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

Kernelization

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

Kernelization

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

Kernelization

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

Kernelization

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

Kernelization

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



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 algorithms have kernelized versions:

_____ ...



