Kernels and Margins

Maria Florina Balcan

10/09/2013

Kernel Methods

Amazingly popular method in ML in recent years.

Lots of Books, Workshops.

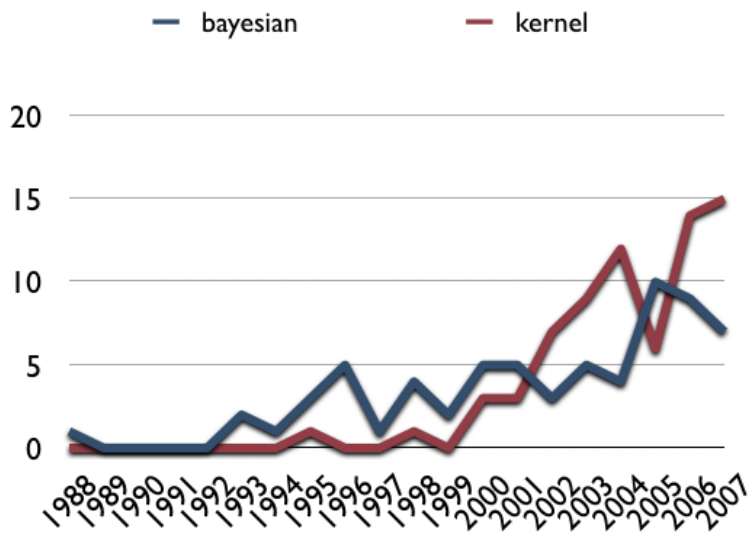
Significant percentage of ICML, NIPS, COLT.

Topic Distribution

Top topics by number of submissions:

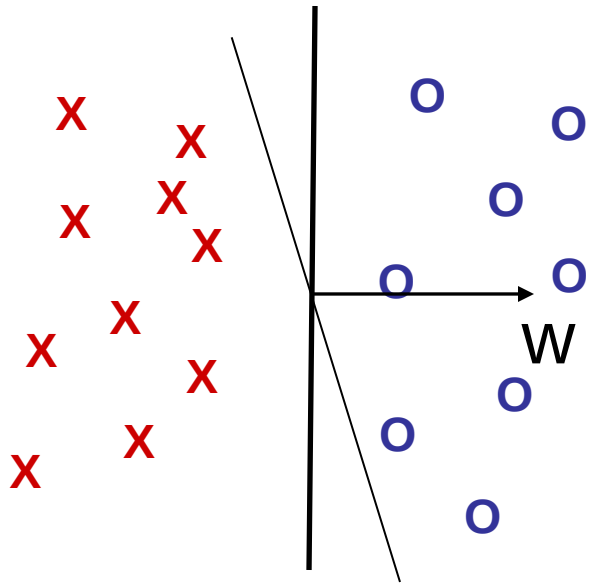
- Kernel methods and support vector machines 94
- Unsupervised learning, clustering 91
- Probabilistic approaches, graphical models 87
- Dimensionality reduction, manifolds and embedding 85
- Statistical models 82
- Reinforcement learning 64
- Semi-supervised learning 62
- Learning from structured data 54
- Bayesian methods 45
- Ensemble methods 37
- Applications and case studies 34
- Learning in vision 30

ICML 2007,
Business meeting



Linear Separators

- Instance space: $X=R^n$
- Hypothesis class of linear decision surfaces in R^n
  - $h(x)=w \cdot x$, if $h(x)> 0$, then label x as +, otherwise label it as -



Lin. Separators: Perceptron algorithm

- Start with all-zeroes weight vector w.
- Given example x, predict positive $\Leftrightarrow$ w $\cdot$ x $\geq$ 0.
- On a mistake, update as follows:
  - Mistake on positive, then update w $\leftarrow$ w + x
  - Mistake on negative, then update w $\leftarrow$ w - x

Note: w is a weighted sum of the incorrectly classified examples

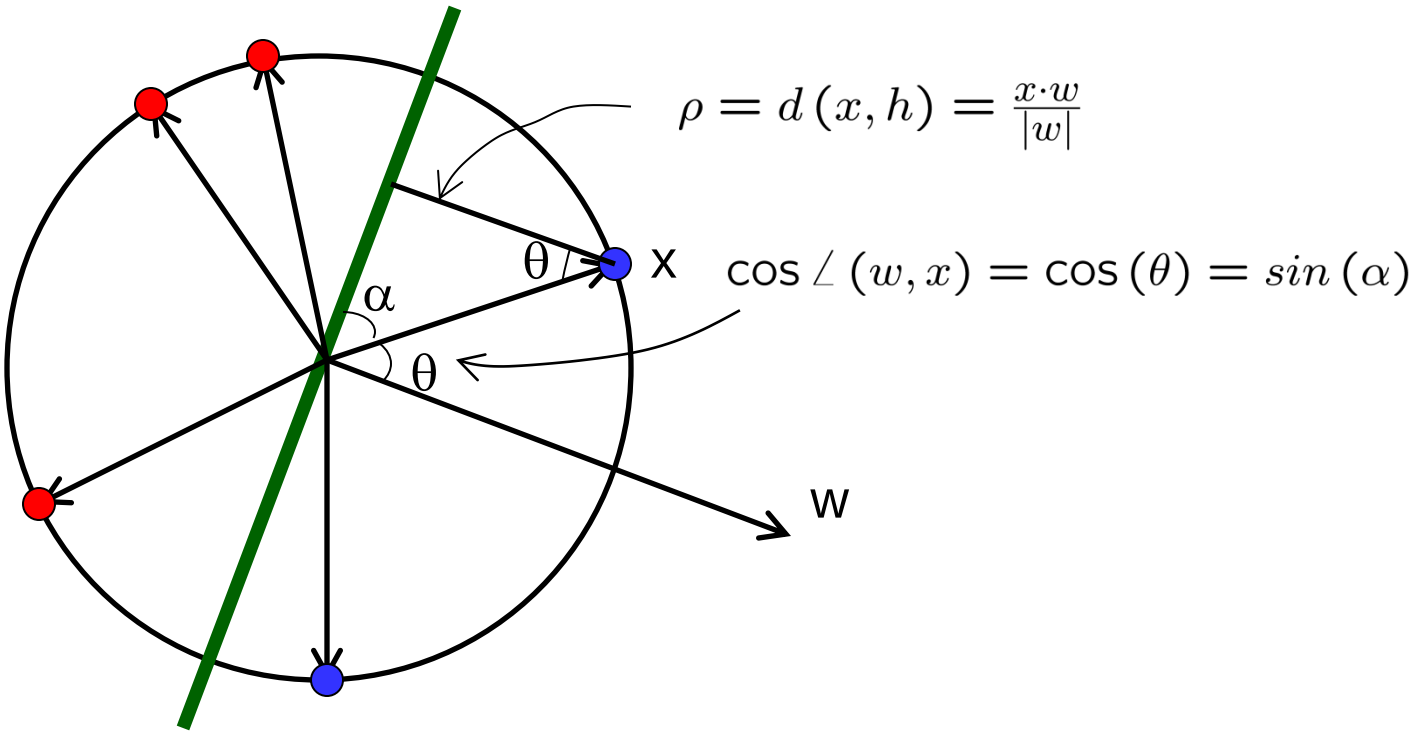
$$w = a_{i_1} x_{i_1} + \cdots + a_{i_k} x_{i_k}$$

$$w \cdot x = a_{i_1} x_{i_1} \cdot x + \cdots + a_{i_k} x_{i_k} \cdot x$$

Guarantee: mistake bound is $1/\gamma^2$

Geometric Margin

- If S is a set of **labeled** examples, then a vector w has margin γ w.r.t. S if $\min_{(x,\ell)\in S} \left[\ell(x)\frac{w\cdot x}{|w||x|}\right] \geq \gamma$



What if Not Linearly Separable

Problem: data not linearly separable in the most natural feature representation.

Example:  vs 

No good linear separator in pixel representation.

Solutions:

- Classic: "Learn a more complex class of functions".

- Modern: "Use a Kernel" **(prominent method in 2000-2010)**

Overview of Kernel Methods

What is a Kernel?

- A kernel K is a legal def of dot-product: i.e. there exists an implicit mapping Φ such that K( , )=Φ ()$\cdot\Phi$ ().

Why Kernels matter?

- Many algorithms interact with data only via dot-products.

  So, if replace $x \cdot y$ with K(x,y), they act implicitly as if data was in the higher-dimensional ϕ-space.

- If data is linearly separable by margin in ϕ-space, then good sample complexity.

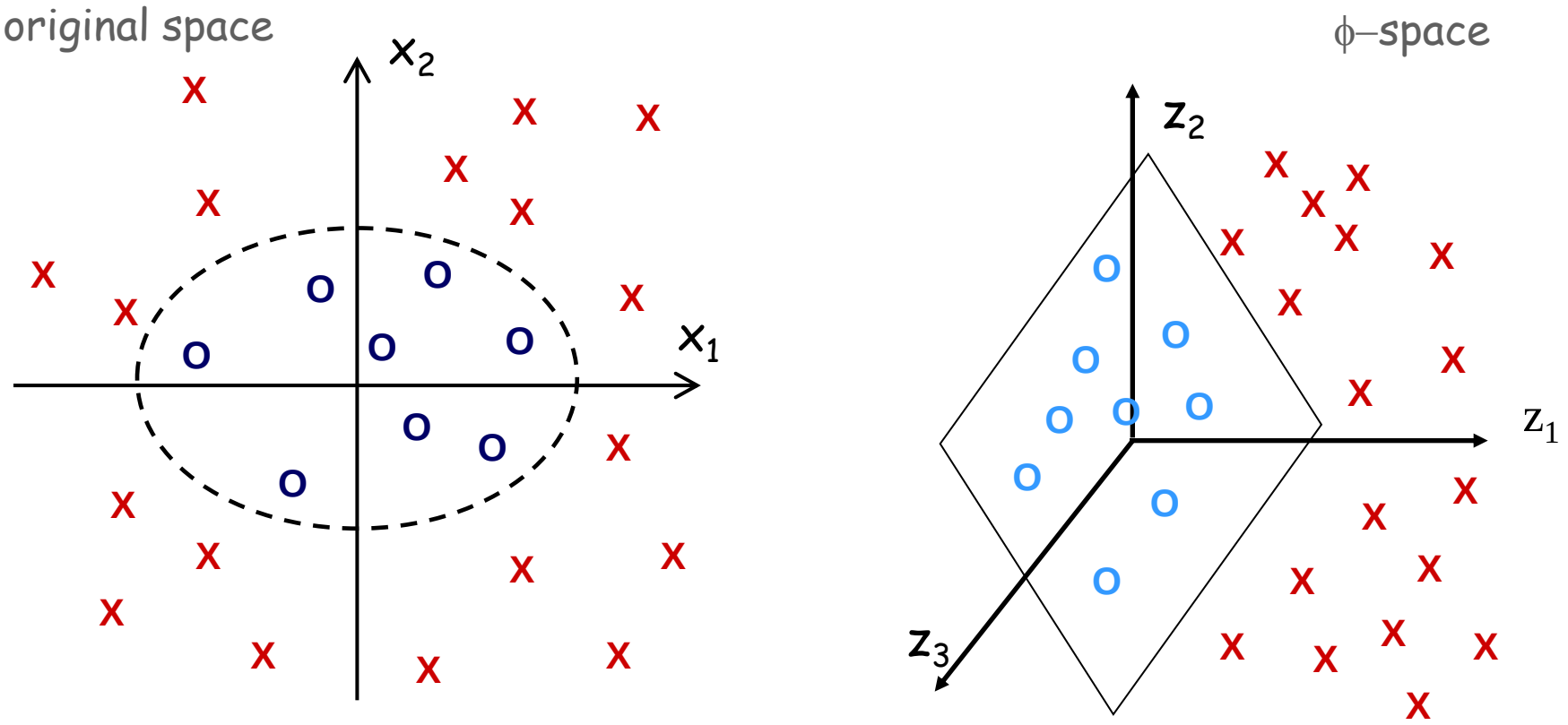
Kernels

- $K(\cdot,\cdot)$ - kernel if it can be viewed as a **legal** definition of inner product:
  - $\exists\ \phi: X \rightarrow R^N$ such that $K(x,y) = \phi(x) \cdot \phi(y)$
    - range of ϕ - "ϕ-space"
    - N can be very large
  - But think of ϕ as implicit, not explicit!

Example

E.g., for n=2, d=2, the kernel $K(x,y) = (x \cdot y)^d$ corresponds to

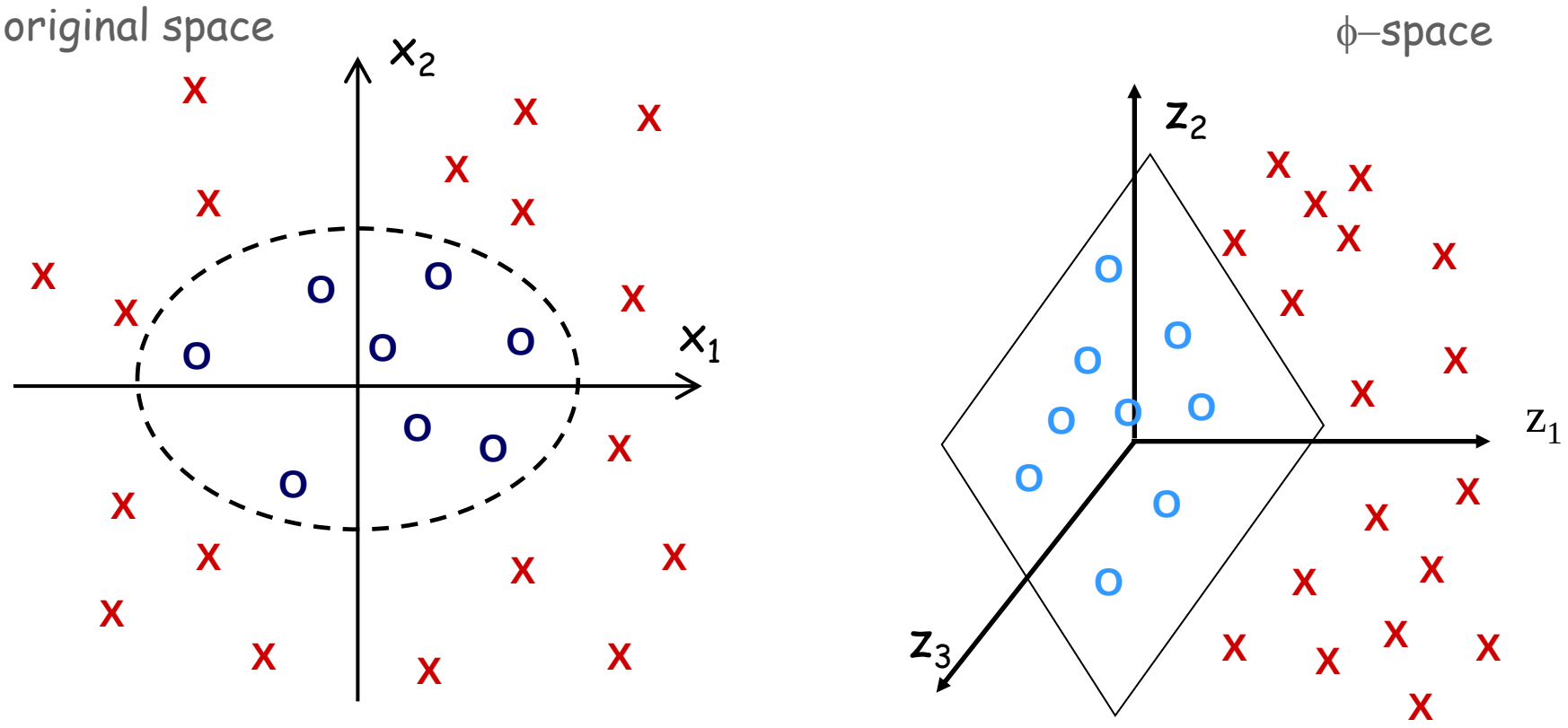
$$(x_1, x_2) \mapsto \phi(x) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$



Example

$$\phi : R^2 \to R^3 \qquad (x_1, x_2) \mapsto (z_1, z_2, z_3) := \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$

$$\begin{aligned}\phi(x)\cdot\phi(x') &= \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)\left(x_1'^2, x_2'^2, \sqrt{2}x_1'x_2'\right)^T \\ &= (x_1x_1' + x_2x_2')^2 = (x \cdot x')^2 =: K(x, x')\end{aligned}$$



Example

Note: feature space need not be unique

$$\phi : R^2 \to R^3 \qquad (x_1, x_2) \mapsto \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$

$$\begin{aligned}\phi(x)\cdot\phi(x') &= \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)\left(x_1'^2, x_2'^2, \sqrt{2}x_1'x_2'\right)^T \\ &= (x\cdot x')^2 =: K(x, x')\end{aligned}$$

$$\phi_1 : R^2 \to R^4 \qquad (x_1, x_2) \mapsto \left(x_1^2, x_2^2, x_1x_2, x_2x_1\right)$$

$$\begin{aligned}\phi_1(x)\cdot\phi_2(x') &= \left(x_1^2, x_2^2, x_1x_2, x_2x_1\right)\left(x_1'^2, x_2'^2, x_1'x_2', x_2'x_1'\right)^T \\ &= (x\cdot x')^2 =: K(x, x')\end{aligned}$$

Kernels

More examples:

- Linear: $K(x,y)=x \cdot y$
- Polynomial: $K(x,y) =(x \cdot y)^d$ or $K(x,y) =(1+x \cdot y)^d$
- Gaussian: $K(x, y) = \exp\left[-\frac{|x-y|^2}{2\sigma^2}\right]$

Theorem

K is a kernel iff

- K is symmetric
- for any set of training points $x_1, x_2, \ldots, x_m$ and for any $a_1, a_2, \ldots, a_m \in R$ we have:

$$\sum_{i,j} a_i a_j K(x_i, x_j) \geq 0$$

Kernelizing a learning algorithm

- If all computations involving instances are in terms of inner products then:
  - Conceptually, work in a very high diml space and the alg's performance depends only on linear separability in that extended space.
  - Computationally, only need to modify the alg. by replacing each $x \cdot y$ with a $K(x,y)$.
- Examples of kernalizable algos: Perceptron, SVM.

Lin. Separators: Perceptron algorithm

- Start with all-zeroes weight vector w.
- Given example x, predict positive $\Leftrightarrow$ w $\cdot$ x $\geq$ 0.
- On a mistake, update as follows:
  - Mistake on positive, then update w $\leftarrow$ w + x
  - Mistake on negative, then update w $\leftarrow$ w - x

Easy to kernelize since w is a weighted sum of examples:

$$w = a_{i_1} x_{i_1} + \cdots + a_{i_k} x_{i_k}$$

Replace $w \cdot x = a_{i_1} x_{i_1} \cdot x + \cdots + a_{i_k} x_{i_k} \cdot x$ with

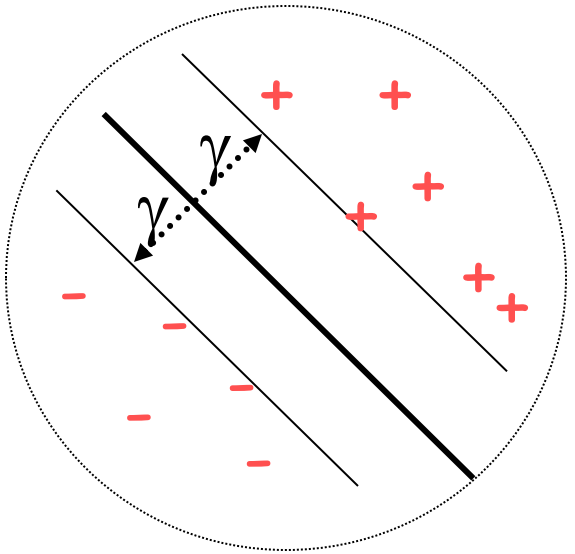
$$a_{i_1} K(x_{i_1}, x) + \cdots + a_{i_k} K(x_{i_k}, x)$$

Note: need to store all the mistakes so far.

Generalize Well if Good Margin

- If data is linearly separable by margin in ϕ-space, then good sample complexity.

  If margin γ in ϕ-space, then need sample size of only $\tilde{O}(1/\gamma^2)$ to get confidence in generalization.



$|\phi(x)| \leq 1$

- Cannot rely on standard VC-bounds since the dimension of the phi-space might be very large.
  - VC-dim for the class of linear sep. in R^m is m+1.

Kernels & Large Margins

- If S is a set of **labeled** examples, then a vector w in the ϕ-space has margin γ if:

$$\min_{(x,\ell)\in S}\left[\ell\frac{w\cdot\phi(x)}{|w||\phi(x)|}\right]\geq\gamma$$

- A vector w in the ϕ-space has margin γ with respect to P if:

$$\Pr_{(x,\ell)\in P}\left[\ell\frac{w\cdot\phi(x)}{|w||\phi(x)|}<\gamma\right]=0$$

- A vector w in the ϕ-space has error α at margin γ if:

$$\Pr_{(x,l)\in P}\left[l\frac{w\cdot\phi(x)}{|w||\phi(x)|}<\gamma\right]\leq\alpha$$

(α,γ)-good kernel

Large Margin Classifiers

- **If** large margin, then the amount of data we need depends only on $1/\gamma$ and is independent on the dim of the space!
  - If large margin γ and if our alg. produces a large margin classifier, then the amount of data we need depends only on $1/\gamma$ [Bartlett & Shawe-Taylor '99]
  - If large margin, then Perceptron also behaves well.
  - Another nice justification based on Random Projection [Arriaga & Vempala '99].

Kernels & Large Margins

- Powerful combination in ML in recent years!
  - A kernel implicitly allows mapping data into a high dimensional space and performing certain operations there without paying a high price computationally.
  - If data indeed has a large margin linear separator in that space, then one can avoid paying a high price in terms of sample size as well.

Kernels Methods

Offer great modularity.

- No need to change the underlying learning algorithm to accommodate a particular choice of kernel function.
- Also, we can substitute a different algorithm while maintaining the same kernel.

Kernels, Closure Properties

- Easily create new kernels using basic ones!

$$K(x,y) = c_1 K_1(x,y) + c_2 K_2(x,y), c_1 \geq 0, c_2 \geq 0$$

$$\phi(x) = (\sqrt{c_1}\phi_1(x), \sqrt{c_2}\phi_2(x))$$

$$\phi(x) \cdot \phi(y) = c_1 \phi_1(x) \cdot \phi_1(y) + c_2 \phi_2(x) \cdot \phi_2(y)$$

Kernels, Closure Properties

- Easily create new kernels using basic ones!

$$K(x,y) = K_1(x,y) \cdot K_2(x,y)$$

$$\phi(x) = (\phi_{1,i}(x)\phi_{2,j}(x))_{i\in\{1,\ldots,n\},j\in\{1,\ldots,m\}}$$

$$\begin{aligned}
\phi(x)\cdot\phi(y) &= \textstyle\sum_{i,j} \phi_{1,i}(x)\phi_{2,j}(x)\phi_{1,i}(y)\phi_{2,j}(y) \\
&= \textstyle\sum_i \left(\phi_{1,i}(x)\phi_{1,i}(y) \sum_j \phi_{2,j}(x)\phi_{2,j}(y)\right) \\
&= \textstyle\sum_i \left(\phi_{1,i}(x)\phi_{1,i}(y) K_2(x,y)\right) \\
&= K_1(x,y)\cdot K_2(x,y)
\end{aligned}$$

What we really care about are good kernels not only legal kernels!

Good Kernels, Margins, and Low Dimensional Mappings

- Designing a kernel function is much like designing a feature space.

- Given a **good** kernel K, we can reinterpret K as defining a new set of $\tilde{O}(1/\gamma^2)$ features.

[Balcan-Blum -Vempala, MLJ'06]