Kernels

Bernhard Schölkopf

Max Planck Institute for Intelligent Systems

Statistical Learning Theory

1. started by Vapnik and Chervonenkis in the Sixties
2. model: we observe data generated by an unknown stochastic regularity
3. *learning* = extraction of the regularity from the data
4. the analysis of the learning problem leads to notions of *capacity* of the function classes that a learning machine can implement.
5. *support vector machines* use a particular type of function class: classifiers with large "margins" in a feature space induced by a *kernel*.

[51, 52]

Finding a Good Function Class

- recall: separating hyperplanes in $\mathbb{R}^2$ have a VC dimension of 3.
- more generally: separating hyperplanes in $\mathbb{R}^N$ have a VC dimension of $N+1$.
- hence: separating hyperplanes in high-dimensional feature spaces have extremely large VC dimension, and may not generalize well
- however, *margin* hyperplanes can still have a small VC dimension

Kernels and Feature Spaces

Preprocess the data with

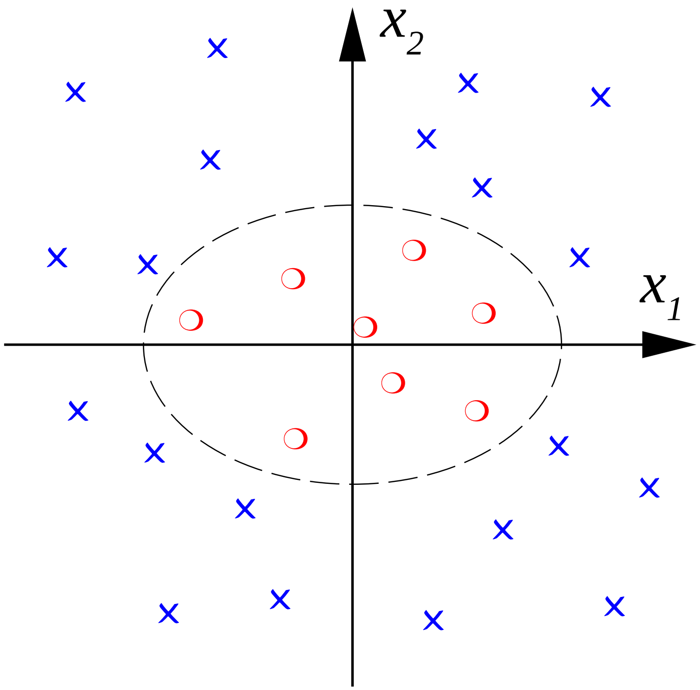
$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \Phi(x), \end{aligned}$$

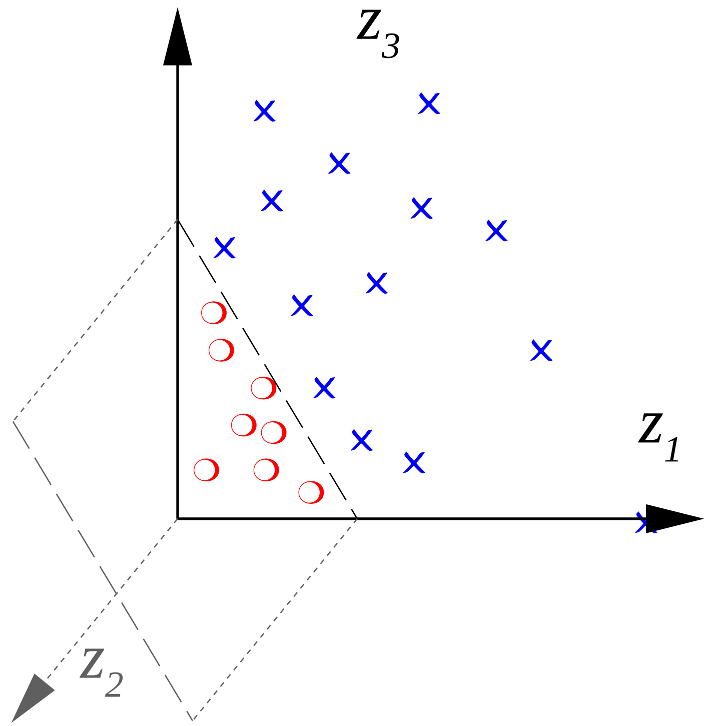
where $\mathcal{H}$ is a dot product space, and learn the mapping from $\Phi(x)$ to y [6].

- usually, $\dim(\mathcal{X}) \ll \dim(\mathcal{H})$
- "Curse of Dimensionality"?
- crucial issue: *capacity*, not *dimensionality*

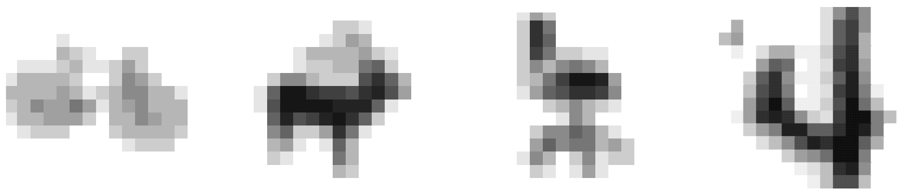
Example: All Degree 2 Monomials

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$





General Product Feature Space



How about patterns $x \in \mathbb{R}^N$ and product features of order d?

Here, $\dim(\mathcal{H})$ grows like N^d.

E.g. $N = 16 \times 16$, and $d = 5 \longrightarrow$ dimension 10^{10}

The Kernel Trick, $N = d = 2$

$$
\begin{aligned}
\left\langle \Phi(x), \Phi(x') \right\rangle &= (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)(x_1'^2, \sqrt{2}\, x_1' x_2', x_2'^2)^\top \\
&= \left\langle x, x' \right\rangle^2 \\
&= : k(x, x')
\end{aligned}
$$

$\longrightarrow$ the dot product in $\mathcal{H}$ can be computed in $\mathbb{R}^2$

The Kernel Trick, II

More generally: $x, x' \in \mathbb{R}^N$, $d \in \mathbb{N}$:

$$\langle x, x' \rangle^d = \left(\sum_{j=1}^{N} x_j \cdot x'_j \right)^d$$

$$= \sum_{j_1,\ldots,j_d=1}^{N} x_{j_1} \cdot \cdots \cdot x_{j_d} \cdot x'_{j_1} \cdot \cdots \cdot x'_{j_d} = \langle \Phi(x), \Phi(x') \rangle ,$$

where Φ maps into the space spanned by all ordered products of d input directions

Mercer's Theorem

If k is a continuous kernel of a positive definite integral operator on $L_2(\mathcal{X})$ (where $\mathcal{X}$ is some compact space),

$$\int_{\mathcal{X}} k(x, x') f(x) f(x')\, dx\, dx' \geq 0,$$

it can be expanded as

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x')$$

using eigenfunctions ψ_i and eigenvalues $\lambda_i \geq 0$ [30].

The Mercer Feature Map

In that case

$$\Phi(x) := \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}$$

satisfies $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$.

Proof:

$$\langle \Phi(x), \Phi(x') \rangle = \left\langle \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}, \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x') \\ \sqrt{\lambda_2}\psi_2(x') \\ \vdots \end{pmatrix} \right\rangle$$

$$= \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x') = k(x, x')$$

Positive Definite Kernels

Let $\mathcal{X}$ be a nonempty set, and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The following two are equivalent:

- k is *positive definite*, i.e., k is symmetric, and for
  - any set of training points $x_1, \ldots, x_m \in \mathcal{X}$ and
  - any $a_1, \ldots, a_m \in \mathbb{R}$

$$\sum_{i,j} a_i a_j K_{ij} \geq 0, \quad \text{where } K_{ij} := k(x_i, x_j)$$

- there exists a map Φ into a dot product space $\mathcal{H}$ such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

$\mathcal{H}$ is a so-called *reproducing kernel Hilbert space.*

If for pairwise distinct points, $\sum = 0$ only if all $a_i = 0$, call k strictly p.d.

The Kernel Trick

- any algorithm that only depends on dot products can be "kernelized"
- this way, we can apply linear methods to vectorial as well as *non-vectorial data*
- think of the kernel as a nonlinear *similarity measure*
- examples of common kernels:

$$\text{Polynomial} \quad k(x,x') = (\langle x, x' \rangle + c)^d$$
$$\text{Gaussian} \quad k(x,x') = \exp(-\|x - x'\|^2/(2\,\sigma^2))$$

- Kernels are also known as covariance functions [58, 56, 59, 29]

Properties of PD Kernels, 1

Assumption: Φ maps $\mathcal{X}$ into a dot product space $\mathcal{H}$; $x, x' \in \mathcal{X}$

Kernels from Feature Maps.
$k(x, x') := \langle \Phi(x), \Phi(x') \rangle$ is a pd kernel on $\mathcal{X} \times \mathcal{X}$.

Kernels from Feature Maps, II
$K(A, B) := \sum_{x \in A, x' \in B} k(x, x')$,
where A, B are finite subsets of $\mathcal{X}$, is also a pd kernel

(Hint: use the feature map $\tilde{\Phi}(A) := \sum_{x \in A} \Phi(x)$)

Properties of PD Kernels, 2 [39, 43]

Assumption: $k, k_1, k_2, \ldots$ are pd; $x, x' \in \mathcal{X}$

$k(x,x) \geq 0$ for all x *(Positivity on the Diagonal)*

$k(x,x')^2 \leq k(x,x)k(x',x')$ *(Cauchy-Schwarz Inequality)*

(Hint: compute the determinant of the Gram matrix)

$k(x,x) = 0$ for all $x \Longrightarrow k(x,x') = 0$ for all x, x' *(Vanishing Diagonals)*

The following kernels are pd:

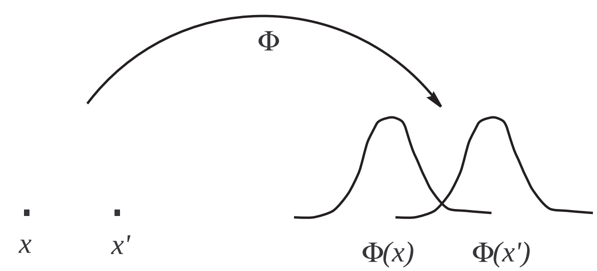
- αk, provided $\alpha \geq 0$
- $k_1 + k_2$
- $k(x,x') := \lim_{n\to\infty} k_n(x,x')$, provided it exists
- $k_1 \cdot k_2$
- tensor products, direct sums, convolutions [23]

The Feature Space for PD Kernels [4, 1, 36]

- define a feature map

$$\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$$
$$x \mapsto k(., x).$$

E.g., for the Gaussian kernel:



Next steps:

- turn $\Phi(\mathcal{X})$ into a linear space
- endow it with a dot product satisfying $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$, i.e., $\langle k(., x), k(., x') \rangle = k(x, x')$
- complete the space to get a *reproducing kernel Hilbert space*

Turn it Into a Linear Space

Form linear combinations

$$f(.) = \sum_{i=1}^{m} \alpha_i k(., x_i),$$

$$g(.) = \sum_{j=1}^{m'} \beta_j k(., x'_j)$$

$(m, m' \in \mathbb{N},\ \alpha_i, \beta_j \in \mathbb{R},\ x_i, x'_j \in \mathcal{X})$.

Endow it With a Dot Product

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

$$= \sum_{i=1}^{m} \alpha_i g(x_i) = \sum_{j=1}^{m'} \beta_j f(x'_j)$$

- This is well-defined, symmetric, and bilinear (more later).
- So far, it also works for non-pd kernels

The Reproducing Kernel Property

Two special cases:

- Assume
$$f(.) = k(., x).$$
  In this case, we have
$$\langle k(., x), g\rangle = g(x).$$
- If moreover
$$g(.) = k(., x'),$$
  we have
$$\langle k(., x), k(., x')\rangle = k(x, x').$$

k is called a *reproducing kernel*

(up to here, have not used positive definiteness)

Endow it With a Dot Product, II

- It can be shown that $\langle .,. \rangle$ is a p.d. kernel on the set of functions $\{f(.) = \sum_{i=1}^m \alpha_i k(.,x_i) | \alpha_i \in \mathbb{R}, x_i \in \mathcal{X}\}$:

$$\sum_{ij} \gamma_i \gamma_j \langle f_i, f_j \rangle = \left\langle \sum_i \gamma_i f_i, \sum_j \gamma_j f_j \right\rangle =: \langle f, f \rangle$$

$$= \left\langle \sum_i \alpha_i k(.,x_i), \sum_i \alpha_i k(.,x_i) \right\rangle = \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

- furthermore, it is *strictly* positive definite:

$$f(x)^2 = \langle f, k(.,x) \rangle^2 \leq \langle f, f \rangle \langle k(.,x), k(.,x) \rangle$$

  hence $\langle f, f \rangle = 0$ implies $f = 0$.

- Complete the space in the corresponding norm to get a Hilbert space $\mathcal{H}_k$.

The Empirical Kernel Map

Recall the feature map

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathbb{R}^{\mathcal{X}} \\ x &\mapsto k(.,x). \end{aligned}$$

- each point is represented by its similarity to *all* other points
- how about representing it by its similarity to a *sample* of points?

Consider

$$\begin{aligned} \Phi_m : \mathcal{X} &\to \mathbb{R}^m \\ x &\mapsto k(.,x)|_{(x_1,\dots,x_m)} = (k(x_1,x),\dots,k(x_m,x))^\top \end{aligned}$$

ctd.

- $\Phi_m(x_1), \dots, \Phi_m(x_m)$ contain *all* necessary information about $\Phi(x_1), \dots, \Phi(x_m)$
- the Gram matrix $G_{ij} := \left\langle \Phi_m(x_i), \Phi_m(x_j) \right\rangle$ satisfies $G = K^2$ where $K_{ij} = k(x_i, x_j)$
- modify Φ_m to

$$\begin{aligned} \Phi_m^w : \mathcal{X} &\to \mathbb{R}^m \\ x &\mapsto K^{-\frac{1}{2}}(k(x_1, x), \dots, k(x_m, x))^\top \end{aligned}$$

- this "whitened" map ("kernel PCA map") satifies

$$\left\langle \Phi_m^w(x_i), \Phi_m^w(x_j) \right\rangle = k(x_i, x_j)$$

for all $i, j = 1, \dots, m$.

Properties of Kernel Matrices, I [37]

Suppose we are given distinct training patterns $x_1, \ldots, x_m$, and a positive definite $m \times m$ matrix K.

K can be diagonalized as $K = SDS^\top$, with an orthogonal matrix S and a diagonal matrix D with nonnegative entries. Then

$$K_{ij} = (SDS^\top)_{ij} = \langle S_i, DS_j \rangle = \left\langle \sqrt{D}S_i, \sqrt{D}S_j \right\rangle,$$

where the S_i are the rows of S.

We have thus constructed a map Φ into an m-dimensional feature space $\mathcal{H}$ such that

$$K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle.$$

Properties, II: Functional Calculus [42]

- K symmetric $m \times m$ matrix with spectrum $\sigma(K)$
- f a continuous function on $\sigma(K)$
- Then there is a symmetric matrix $f(K)$ with eigenvalues in $f(\sigma(K))$.
- compute $f(K)$ via Taylor series, or eigenvalue decomposition of K: If $K = S^\top DS$ (D diagonal and S unitary), then $f(K) = S^\top f(D)S$, where $f(D)$ is defined elementwise on the diagonal
- can treat functions of symmetric matrices like functions on $\mathbb{R}$

$$\begin{aligned}
(\alpha f + g)(K) &= \alpha f(K) + g(K) \\
(fg)(K) &= f(K)g(K) = g(K)f(K) \\
\|f\|_{\infty,\sigma(K)} &= \|f(K)\| \\
\sigma(f(K)) &= f(\sigma(K))
\end{aligned}$$

(the C^*-algebra generated by K is isomorphic to the set of continuous functions on $\sigma(K)$)

Computing Distances in Feature Spaces

Clearly, if k is positive definite, then there exists a map Φ such that

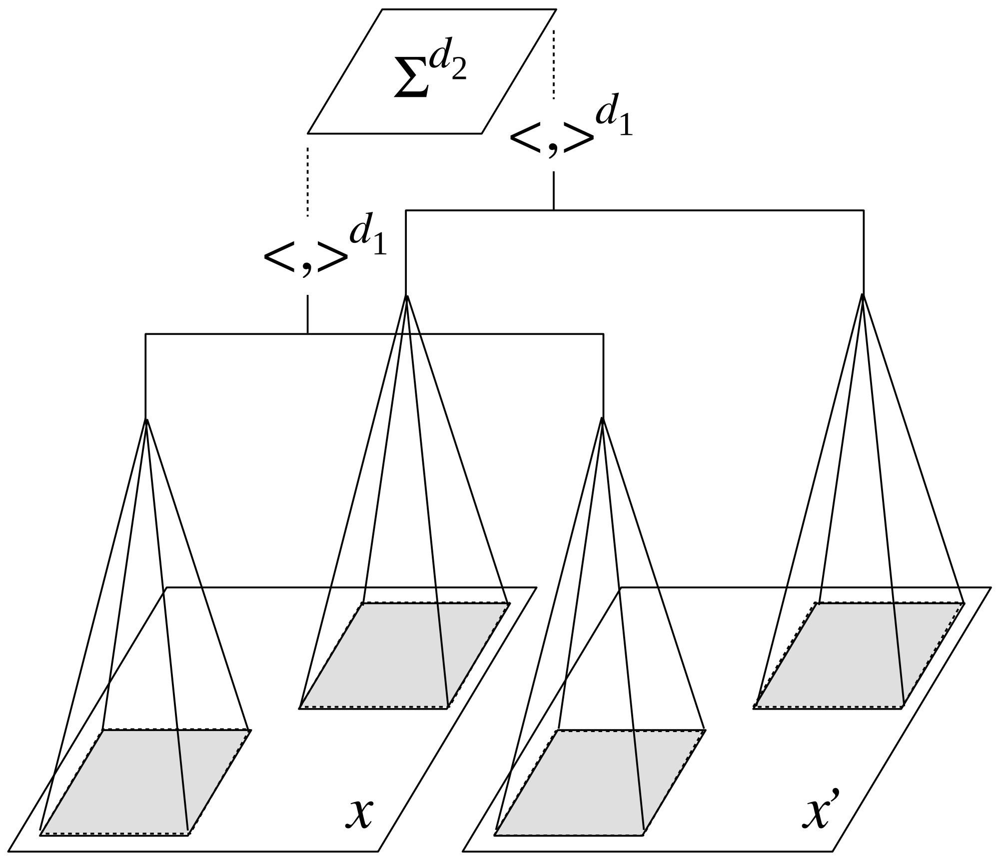
$$\|\Phi(x) - \Phi(x')\|^2 = k(x,x) + k(x',x') - 2k(x,x')$$

(it is the usual feature map).

This embedding is referred to as a *Hilbert space representation* as a distance. It turns out that this works for a larger class of kernels, called *conditionally positive definite*.

In fact, all algorithms that are translationally invariant (i.e. independent of the choice of the origin) in $\mathcal{H}$ work with cpd kernels [39].

Kernels Local in the Image

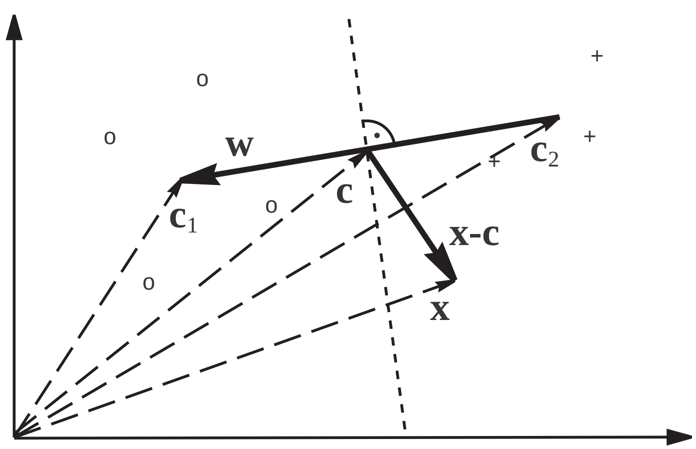


Local products of degree d_1, global products of degree d_2, overall degree $d_1 \cdot d_2$.
[38]

An Example of a Kernel Algorithm

Idea: classify points $\mathbf{x} := \Phi(x)$ in feature space according to which of the two class means is closer.

$$\mathbf{c}_+ := \frac{1}{m_+} \sum_{y_i=1} \Phi(x_i), \quad \mathbf{c}_- := \frac{1}{m_-} \sum_{y_i=-1} \Phi(x_i)$$



Compute the sign of the dot product between $\mathbf{w} := \mathbf{c}_+ - \mathbf{c}_-$ and $\mathbf{x} - \mathbf{c}$.

An Example of a Kernel Algorithm, ctd. [39]

$$f(x) = \operatorname{sgn}\left(\frac{1}{m_+}\sum_{\{i:y_i=+1\}}\langle\Phi(x),\Phi(x_i)\rangle - \frac{1}{m_-}\sum_{\{i:y_i=-1\}}\langle\Phi(x),\Phi(x_i)\rangle + b\right)$$

$$= \operatorname{sgn}\left(\frac{1}{m_+}\sum_{\{i:y_i=+1\}} k(x,x_i) - \frac{1}{m_-}\sum_{\{i:y_i=-1\}} k(x,x_i) + b\right)$$

where

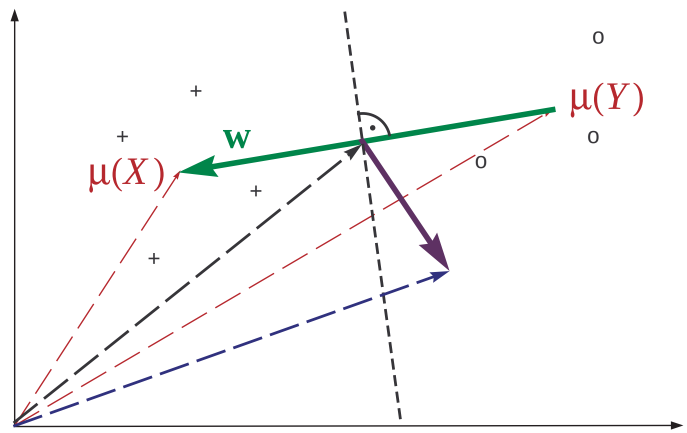
$$b = \frac{1}{2}\left(\frac{1}{m_-^2}\sum_{\{(i,j):y_i=y_j=-1\}} k(x_i,x_j) - \frac{1}{m_+^2}\sum_{\{(i,j):y_i=y_j=+1\}} k(x_i,x_j)\right).$$

- provides a geometric interpretation of Parzen windows

An Example of a Kernel Algorithm, ctd.

- Exercise: derive the Parzen windows classifier by computing the distance criterion directly
- SVMs (ppt)

An example of a kernel algorithm, revisited



$\mathcal{X}$ compact subset of a separable metric space, $m, n \in \mathbb{N}$.

Positive class $X := \{x_1, \ldots, x_m\} \subset \mathcal{X}$
Negative class $Y := \{y_1, \ldots, y_n\} \subset \mathcal{X}$

RKHS means $\mu(X) = \frac{1}{m}\sum_{i=1}^{m} k(x_i, \cdot)$, $\mu(Y) = \frac{1}{n}\sum_{i=1}^{n} k(y_i, \cdot)$.

Get a problem if $\mu(X) = \mu(Y)$!

When do the means coincide?

$k(x,x') = \langle x, x' \rangle$:	the means coincide
$k(x,x') = (\langle x, x' \rangle + 1)^d$:	all empirical moments up to order d coincide
k strictly pd:	$X = Y$.

The mean "remembers" each point that contributed to it.

Proposition 1 *Assume* X, Y *are defined as above,* k *is strictly pd, and for all* i, j, $x_i \neq x_j$, *and* $y_i \neq y_j$.
If for some $\alpha_i, \beta_j \in \mathbb{R} - \{0\}$, *we have*

$$\sum_{i=1}^{m} \alpha_i k(x_i, .) = \sum_{j=1}^{n} \beta_j k(y_j, .), \tag{1}$$

then $X = Y$.

Proof (by contradiction)

W.l.o.g., assume that $x_1 \notin Y$. Subtract $\sum_{j=1}^{n} \beta_j k(y_j, .)$ from (1), and make it a sum over pairwise distinct points, to get

$$0 = \sum_i \gamma_i k(z_i, .),$$

where $z_1 = x_1, \gamma_1 = \alpha_1 \neq 0$, and
$z_2, \dots \in X \cup Y - \{x_1\},\ \gamma_2, \dots \in \mathbb{R}$.
Take the RKHS dot product with $\sum_j \gamma_j k(z_j, .)$ to get

$$0 = \sum_{ij} \gamma_i \gamma_j k(z_i, z_j),$$

with $\gamma \neq 0$, hence k cannot be strictly pd.

■

The mean map

$$\mu\colon X = (x_1, \ldots, x_m) \mapsto \frac{1}{m}\sum_{i=1}^{m} k(x_i, \cdot)$$

satisfies

$$\langle \mu(X), f \rangle = \left\langle \frac{1}{m}\sum_{i=1}^{m} k(x_i, \cdot), f \right\rangle = \frac{1}{m}\sum_{i=1}^{m} f(x_i)$$

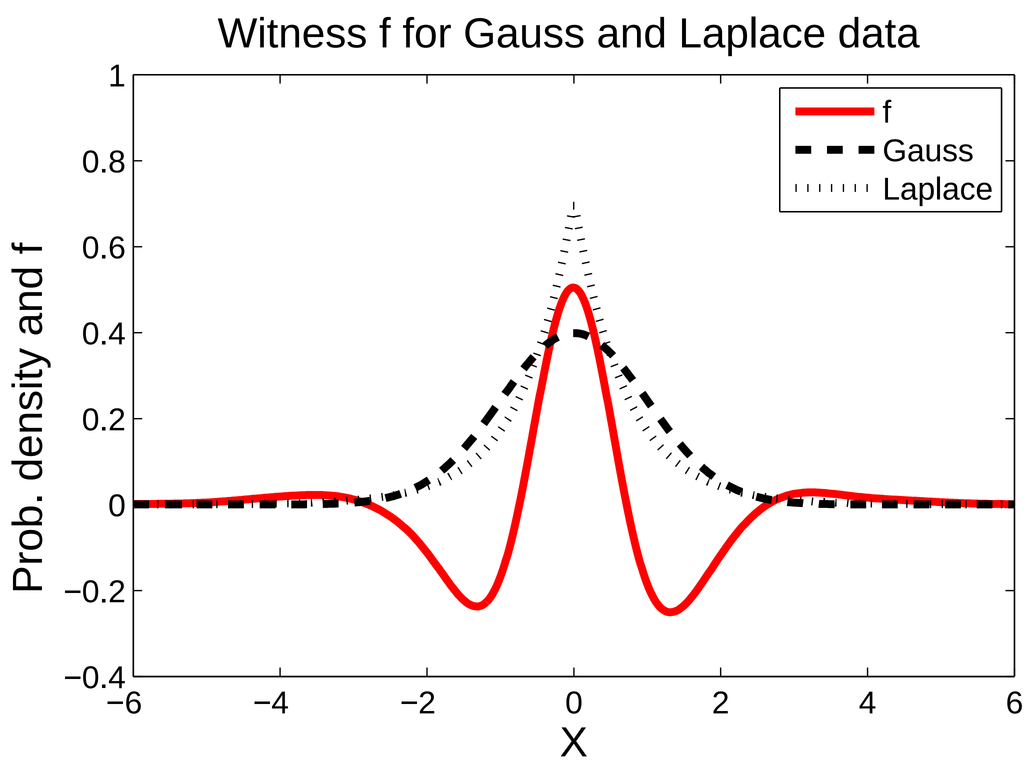
and

$$\|\mu(X)-\mu(Y)\| = \sup_{\|f\|\le 1} |\langle \mu(X) - \mu(Y), f \rangle| = \sup_{\|f\|\le 1} \left| \frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i) \right|.$$

Note: Large distance = can find a function distinguishing the samples

Witness function

$f = \frac{\mu(X)-\mu(Y)}{\|\mu(X)-\mu(Y)\|}$, thus $f(x) \propto \langle \mu(X) - \mu(Y), k(x, .) \rangle)$:



This function is in the RKHS of a Gaussian kernel, but not in the RKHS of the linear kernel.

The mean map for measures

p, q Borel probability measures,
$\mathbf{E}_{x,x' \sim p}[k(x,x')]$, $\mathbf{E}_{x,x' \sim q}[k(x,x')] < \infty$ ($\|k(x,.)\| \leq M < \infty$ is sufficient)

Define

$$\mu\colon p \mapsto \mathbf{E}_{x \sim p}[k(x,\cdot)].$$

Note

$$\langle \mu(p), f \rangle = \mathbf{E}_{x \sim p}[f(x)]$$

and

$$\|\mu(p) - \mu(q)\| = \sup_{\|f\| \leq 1} \left| \mathbf{E}_{x \sim p}[f(x)] - \mathbf{E}_{x \sim q}[f(x)] \right|.$$

Recall that in the finite sample case, for strictly p.d. kernels, μ was injective — how about now?
[47, 17]

Theorem 2 *[15, 13]*

$$p = q \iff \sup_{f \in C(\mathcal{X})} \left| \mathbf{E}_{x \sim p}(f(x)) - \mathbf{E}_{x \sim q}(f(x)) \right| = 0,$$

where $C(\mathcal{X})$ is the space of continuous bounded functions on $\mathcal{X}$.

Combine this with

$$\|\mu(p) - \mu(q)\| = \sup_{\|f\| \leq 1} \left| \mathbf{E}_{x \sim p}[f(x)] - \mathbf{E}_{x \sim q}[f(x)] \right| .$$

Replace $C(\mathcal{X})$ by the unit ball in an RKHS that is dense in $C(\mathcal{X})$ — universal kernel [49], e.g., Gaussian.

Theorem 3 *[19] If k is universal, then*

$$p = q \iff \|\mu(p) - \mu(q)\| = 0.$$

- μ is invertible on its image
  $\mathcal{M} = \{\mu(p) \mid p \text{ is a probability distribution}\}$
  (the "marginal polytope", [57])
- generalization of the *moment generating function* of a RV x with distribution p:

$$M_p(.) = \mathbf{E}_{x \sim p}\left[e^{\langle x, \cdot \rangle}\right].$$

This provides us with a convenient metric on probability distributions, which can be used to check whether two distributions are different — provided that μ is invertible.

Fourier Criterion

Assume we have densities, the kernel is shift invariant ($k(x, y) = k(x - y)$), and all Fourier transforms below exist.
Note that μ is invertible iff

$$\int k(x-y)p(y)\,dy = \int k(x-y)q(y)\,dy \Longrightarrow p = q,$$

i.e.,

$$\hat{k}(\hat{p} - \hat{q}) = 0 \Longrightarrow p = q$$

(Sriperumbudur et al., 2008)

E.g., μ is invertible if $\hat{k}$ has full support. Restricting the class of distributions, weaker conditions suffice (e.g., if $\hat{k}$ has non-empty interior, μ is invertible for all distributions with compact support).

Fourier Optics

Application: p source of incoherent light, I indicator of a finite aperture. In Fraunhofer diffraction, the intensity image is $\propto p * \hat{I}^2$. Set $k = \hat{I}^2$, then this equals $\mu(p)$.
This $\hat{k}$ does not have full support, thus the imaging process is not invertible for the class of *all* light sources (Abbe), but it is if we restrict the class (e.g., to compact support).

Harmeling et al., CVPR 2013

Application 1: Two-sample problem [19]

X, Y i.i.d. m-samples from p, q, respectively.

$$\begin{aligned}\|\mu(p) - \mu(q)\|^2 =& \mathbf{E}_{x,x'\sim p}\left[k(x, x')\right] - 2\mathbf{E}_{x\sim p,y\sim q}\left[k(x, y)\right] + \mathbf{E}_{y,y'\sim q}\left[k(y, y')\right] \\ =& \mathbf{E}_{x,x'\sim p,y,y'\sim q}\left[h((x, y), (x', y'))\right]\end{aligned}$$

with

$$h((x, y), (x', y')) := k(x, x') - k(x, y') - k(y, x') + k(y, y').$$

Define

$$\begin{aligned}D(p, q)^2 &:= \mathbf{E}_{x,x'\sim p,y,y'\sim q} h((x, y), (x', y')) \\ \hat{D}(X, Y)^2 &:= \frac{1}{m(m-1)} \sum_{i\neq j} h((x_i, y_i), (x_j, y_j)).\end{aligned}$$

$\hat{D}(X, Y)^2$ is an unbiased estimator of $D(p, q)^2$.
It's easy to compute, and works on structured data.

Theorem 4 *Assume k is bounded.*
$\hat{D}(X,Y)^2$ converges to $D(p,q)^2$ in probability with rate $\mathcal{O}(m^{-\frac{1}{2}})$.

This *could* be used as a basis for a test, but uniform convergence bounds are often loose..

Theorem 5 *We assume $\mathbf{E}\left(h^2\right) < \infty$. When $p \neq q$, then $\sqrt{m}(\hat{D}(X,Y)^2 - D(p,q)^2)$ converges in distribution to a zero mean Gaussian with variance*

$$\sigma_u^2 = 4\left(\mathbf{E}_z\left[(\mathbf{E}_{z'}h(z,z'))^2\right] - \left[\mathbf{E}_{z,z'}(h(z,z'))\right]^2\right).$$

When $p = q$, then $m(\hat{D}(X,Y)^2 - D(p,q)^2) = m\hat{D}(X,Y)^2$ converges in distribution to

$$\sum_{l=1}^{\infty} \lambda_l \left[q_l^2 - 2\right], \tag{2}$$

where $q_l \sim \mathcal{N}(0,2)$ i.i.d., λ_i are the solutions to the eigenvalue equation

$$\int_{\mathcal{X}} \tilde{k}(x,x')\psi_i(x)dp(x) = \lambda_i\psi_i(x'),$$

and $\tilde{k}(x_i,x_j) := k(x_i,x_j) - \mathbf{E}_x k(x_i,x) - \mathbf{E}_x k(x,x_j) + \mathbf{E}_{x,x'}k(x,x')$ is the centred RKHS kernel.

Application 2: Dependence Measures

Assume that (x, y) are drawn from p_{xy}, with marginals p_x, p_y.

Want to know whether p_{xy} factorizes.
[2, 16]: kernel generalized variance

[20, 21]: kernel constrained covariance, HSIC

Main idea [25, 35]:
x and y independent $\Longleftrightarrow$ $\forall$ bounded continuous functions f, g, we have $\mathrm{Cov}(f(x), g(y)) = 0$.

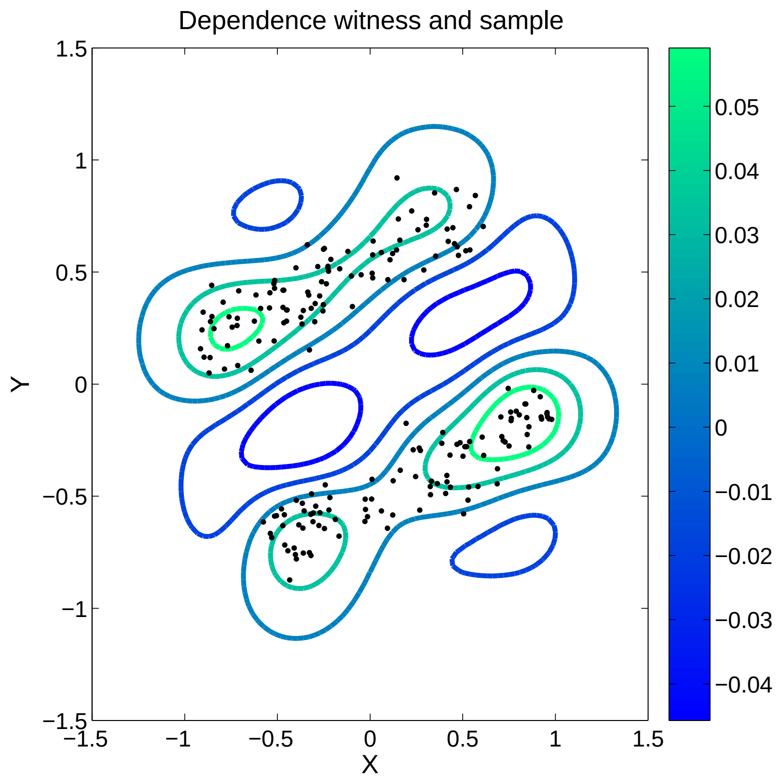
k kernel on $\mathcal{X} \times \mathcal{Y}$.

$$\mu(p_{xy}) := \mathbf{E}_{(x,y)\sim p_{xy}}\left[k((x,y),\cdot)\right]$$
$$\mu(p_x \times p_y) := \mathbf{E}_{x\sim p_x, y\sim p_y}\left[k((x,y),\cdot)\right].$$

Use $\Delta := \left\|\mu(p_{xy}) - \mu(p_x \times p_y)\right\|$ as a measure of dependence.

For $k((x,y),(x',y')) = k_x(x,x')k_y(y,y')$:
Δ^2 equals the Hilbert-Schmidt norm of the covariance operator between the two RKHSs (HSIC), with empirical estimate $m^{-2}\operatorname{tr} HK_xHK_y$, where $H = I - \mathbf{1}/m$ [20, 48].

Witness function of the equivalent optimisation problem:



Application: learning causal structures *(Sun et al., ICML 2007; Fukumizu et al., NIPS 2007)*)

Application 3: Covariate Shift Correction and Local Learning

training set $X = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ drawn from p,
test set $X' = \{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$ from $p' \neq p$.

Assume $p_{y|x} = p'_{y|x}$.

[44]: reweight training set

Minimize

$$\left\| \sum_{i=1}^{m} \beta_i k(x_i, \cdot) - \mu(X') \right\|^2 + \lambda \|\beta\|_2^2 \quad \text{subject to} \quad \beta_i \geq 0, \ \sum_i \beta_i = 1.$$

Equivalent QP:

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2}\beta^\top (K + \lambda \mathbf{1}) \beta - \beta^\top l$$

$$\text{subject to } \beta_i \geq 0 \text{ and } \sum_i \beta_i = 1,$$

where $K_{ij} := k(x_i, x_j)$, $l_i = \langle k(x_i, \cdot), \mu(X') \rangle$.

Experiments show that in underspecified situations (e.g., large kernel widths), this helps [24].

$X' = \{x'\}$ leads to a local sample weighting scheme.

The Representer Theorem

Theorem 6 *Given: a p.d. kernel k on $\mathcal{X} \times \mathcal{X}$, a training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonic increasing real-valued function Ω on $[0, \infty[$, and an arbitrary cost function $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$*

Any $f \in \mathcal{H}_k$ minimizing the regularized risk functional

$$c\left((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))\right) + \Omega\left(\|f\|\right) \tag{3}$$

admits a representation of the form

$$f(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .).$$

Remarks

- significance: many learning algorithms have solutions that can be expressed as expansions in terms of the training examples
- original form, with mean squared loss

  $$c((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2,$$

  and $\Omega(\|f\|) = \lambda\|f\|^2$ ($\lambda > 0$): [27]
- generalization to non-quadratic cost functions: [10]
- present form: [39]
- recent generalizations: [31, 12]

Proof

Decompose $f \in \mathcal{H}$ into a part in the span of the $k(x_i, .)$ and an orthogonal one:

$$f = \sum_i \alpha_i k(x_i, .) + f_\perp,$$

where for all j

$$\langle f_\perp, k(x_j, .)\rangle = 0.$$

Application of f to an arbitrary training point x_j yields

$$\begin{aligned} f(x_j) &= \langle f, k(x_j, .)\rangle \\ &= \left\langle \sum_i \alpha_i k(x_i, .) + f_\perp, k(x_j, .) \right\rangle \\ &= \sum_i \alpha_i \langle k(x_i, .), k(x_j, .)\rangle, \end{aligned}$$

independent of $f_\perp$.

Proof: second part of (3)

Since $f_\perp$ is orthogonal to $\sum_i \alpha_i k(x_i, .)$, and Ω is strictly monotonic, we get

$$\begin{aligned}\Omega(\|f\|) &= \Omega\left(\|\sum_i \alpha_i k(x_i, .) + f_\perp\|\right) \\ &= \Omega\left(\sqrt{\|\sum_i \alpha_i k(x_i, .)\|^2 + \|f_\perp\|^2}\right) \\ &\geq \Omega\left(\|\sum_i \alpha_i k(x_i, .)\|\right), \end{aligned} \tag{4}$$

with equality occuring if and only if $f_\perp = 0$.
Hence, any minimizer must have $f_\perp = 0$. Consequently, any solution takes the form

$$f = \sum_i \alpha_i k(x_i, .).$$

Application: Support Vector Classification

Here, $y_i \in \{\pm 1\}$. Use

$$c\left(\left(x_i, y_i, f\left(x_i\right)\right)_i\right) = \frac{1}{\lambda}\sum_i \max\left(0, 1 - y_i f\left(x_i\right)\right),$$

and the regularizer $\Omega\left(\|f\|\right) = \|f\|^2$.
$\lambda \to 0$ leads to the hard margin SVM

Further Applications

Bayesian MAP Estimates. Identify (3) with the negative log posterior (cf. Kimeldorf & Wahba, 1970, Poggio & Girosi, 1990), i.e.

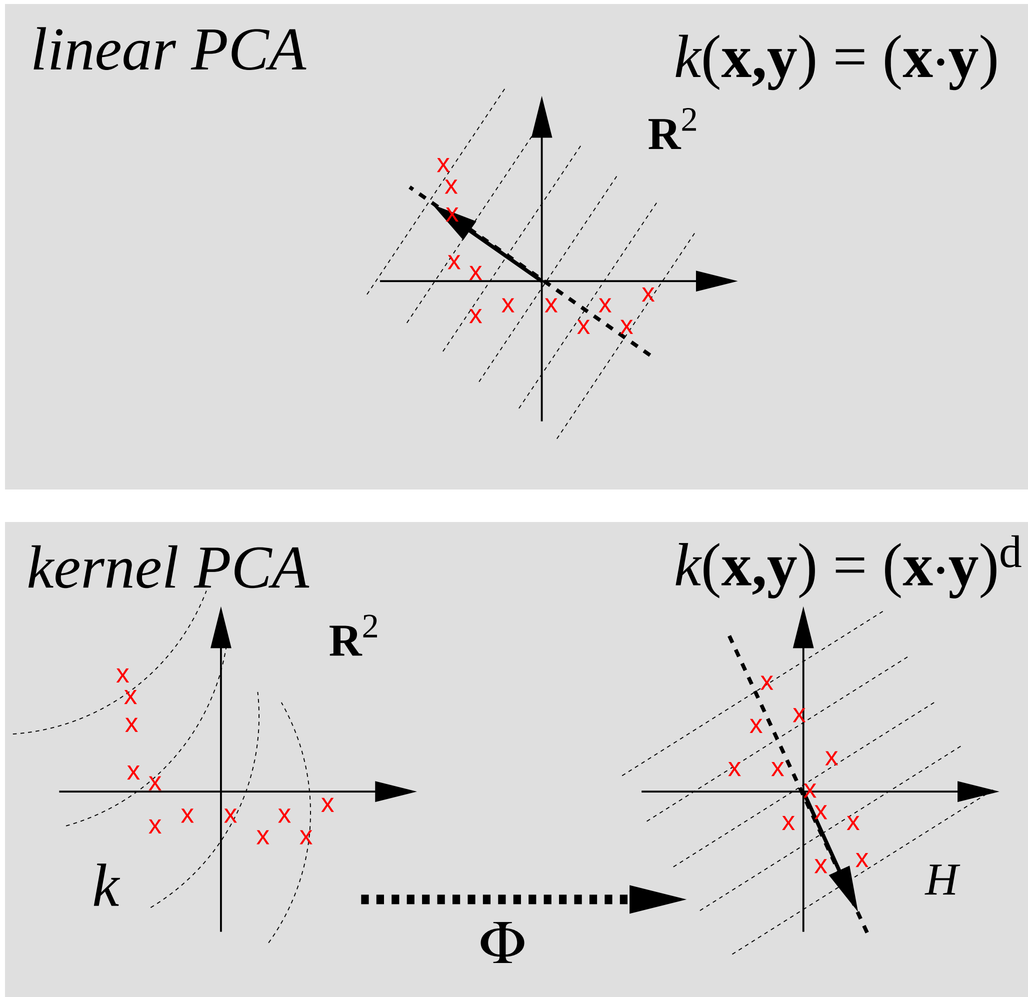
- $\exp(-c((x_i, y_i, f(x_i))_i))$ — likelihood of the data
- $\exp(-\Omega(\|f\|))$ — prior over the set of functions; e.g., $\Omega(\|f\|) = \lambda\|f\|^2$ — Gaussian process prior [59] with covariance function k
- minimizer of (3) = MAP estimate

Kernel PCA (see below) can be shown to correspond to the case of

$$c((x_i, y_i, f(x_i))_{i=1,\dots,m}) = \begin{cases} 0 & \text{if } \frac{1}{m}\sum_i \left(f(x_i) - \frac{1}{m}\sum_j f(x_j)\right)^2 = 1 \\ \infty & \text{otherwise} \end{cases}$$

with Ω an arbitrary strictly monotonically increasing function.

Kernel PCA



Kernel PCA, II

$$x_1, \ldots, x_m \in \mathcal{X}, \qquad \Phi : \mathcal{X} \to \mathcal{H}, \qquad C = \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j)\Phi(x_j)^\top$$

Eigenvalue problem

$$\lambda \mathbf{V} = C\mathbf{V} = \frac{1}{m} \sum_{j=1}^{m} \left\langle \Phi(x_j), \mathbf{V} \right\rangle \Phi(x_j).$$

For $\lambda \neq 0$, $\mathbf{V} \in \mathrm{span}\{\Phi(x_1), \ldots, \Phi(x_m)\}$, thus

$$\mathbf{V} = \sum_{i=1}^{m} \alpha_i \Phi(x_i),$$

and the eigenvalue problem can be written as

$$\lambda \left\langle \Phi(x_n), \mathbf{V} \right\rangle = \left\langle \Phi(x_n), C\mathbf{V} \right\rangle \text{ for all } n = 1, \ldots, m$$

Kernel PCA in Dual Variables

In term of the $m \times m$ Gram matrix

$$K_{ij} := \left\langle \Phi(x_i), \Phi(x_j) \right\rangle = k(x_i, x_j),$$

this leads to

$$m\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$.

Solve

$$m\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha}$$

$\longrightarrow (\lambda_n, \boldsymbol{\alpha}^n)$

$$\langle \mathbf{V}^n, \mathbf{V}^n \rangle = 1 \Longleftrightarrow \lambda_n \langle \boldsymbol{\alpha}^n, \boldsymbol{\alpha}^n \rangle = 1$$

thus divide $\boldsymbol{\alpha}^n$ by $\sqrt{\lambda_n}$

Feature extraction

Compute projections on the Eigenvectors

$$\mathbf{V}^n = \sum_{i=1}^{m} \alpha_i^n \Phi(x_i)$$

in $\mathcal{H}$:

for a test point x with image $\Phi(x)$ in $\mathcal{H}$ we get the features

$$\begin{aligned} \langle \mathbf{V}^n, \Phi(x) \rangle &= \sum_{i=1}^{m} \alpha_i^n \langle \Phi(x_i), \Phi(x) \rangle \\ &= \sum_{i=1}^{m} \alpha_i^n k(x_i, x) \end{aligned}$$

The Kernel PCA Map

Recall

$$\begin{aligned}\Phi_m^w : \mathcal{X} &\to \mathbb{R}^m \\ x &\mapsto K^{-\frac{1}{2}}(k(x_1, x), \ldots, k(x_m, x))^\top\end{aligned}$$

If $K = UDU^\top$ is K's diagonalization, then $K^{-1/2} = UD^{-1/2}U^\top$. Thus we have

$$\Phi_m^w(x) = UD^{-1/2}U^\top(k(x_1, x), \ldots, k(x_m, x))^\top.$$

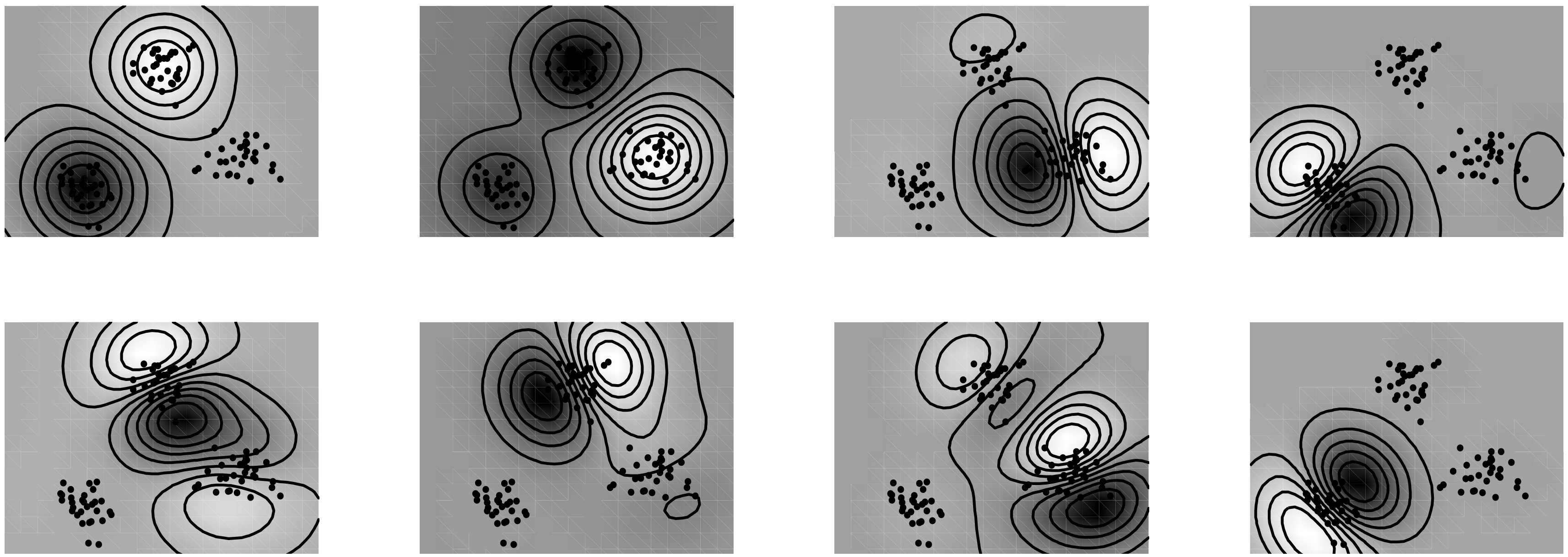
We can drop the leading U (since it leaves the dot product invariant) to get a map

$$\Phi_{KPCA}^w(x) = D^{-1/2}U^\top(k(x_1, x), \ldots, k(x_m, x))^\top.$$

The rows of $U^\top$ are the eigenvectors $\boldsymbol{\alpha}^n$ of K, and the entries of the diagonal matrix $D^{-1/2}$ equal $\lambda_i^{-1/2}$.

Toy Example with Gaussian Kernel

$$k(x, x') = \exp\left(-\|x - x'\|^2\right)$$



KPCA includes various spectral dimensionality reduction algorithms as special cases with data-dependent kernels [22].

Spectral clustering

K similarity matrix; $D_{ii} = \sum_j K_{ij}$

Normalized cuts (Shi & Malik, 2000):

- map inputs to corresponding entries of the second smallest eigenvector of the normalized Laplacian

$$L = I - D^{-1/2} K D^{-1/2}.$$

- Partition them based on these values.

Meila & Shi (2001):

- map inputs to entries of leading eigenvectors of

$$D^{-1} K$$

- continue with k-means

Kernel PCA (1998):

- map test point x to RKHS, project on leading eigenvectors of K:

$$\langle V^n, k(x,.)\rangle = \sum_{i=1}^{m} \alpha_i^n \langle k(x_i,.), k(x,.)\rangle = \sum_{i=1}^{m} \alpha_i^n k(x_i, x)$$

Link Kernel PCA — Spectral Clustering

Projection of a *training point* x_t onto the nth eigenvector equals

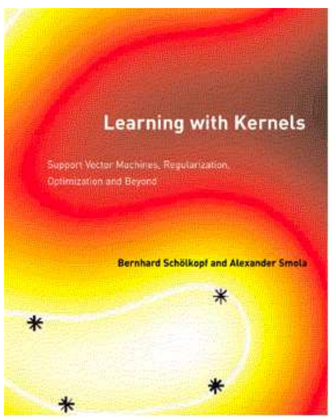
$$(K\alpha^n)_t = \lambda_n \alpha_t^n.$$

where $\langle \alpha^n, \alpha^n \rangle = \lambda_n^{-1}$.
The eigenvector α^n thus contains the projections of the training set.

- for a connected graph, the normalized Laplacian has a single 0 eigenvalue. Its (pseudo-)inverse is the *discrete Green's function* of the diffusion process governed by L. It can be viewed as a kernel matrix, encoding the dot product implying the commute time metric *(Ham, Lee, Mika, Schölkopf, 2004)*
- the kPCA matrix is centered, and thus has a single eigenvalue 0 (for strictly p.d. kernel) that corresponds to the 0 eigenvalue of the normalized Laplacian.
- inversion inverts the order of the remaining eigenvalues.

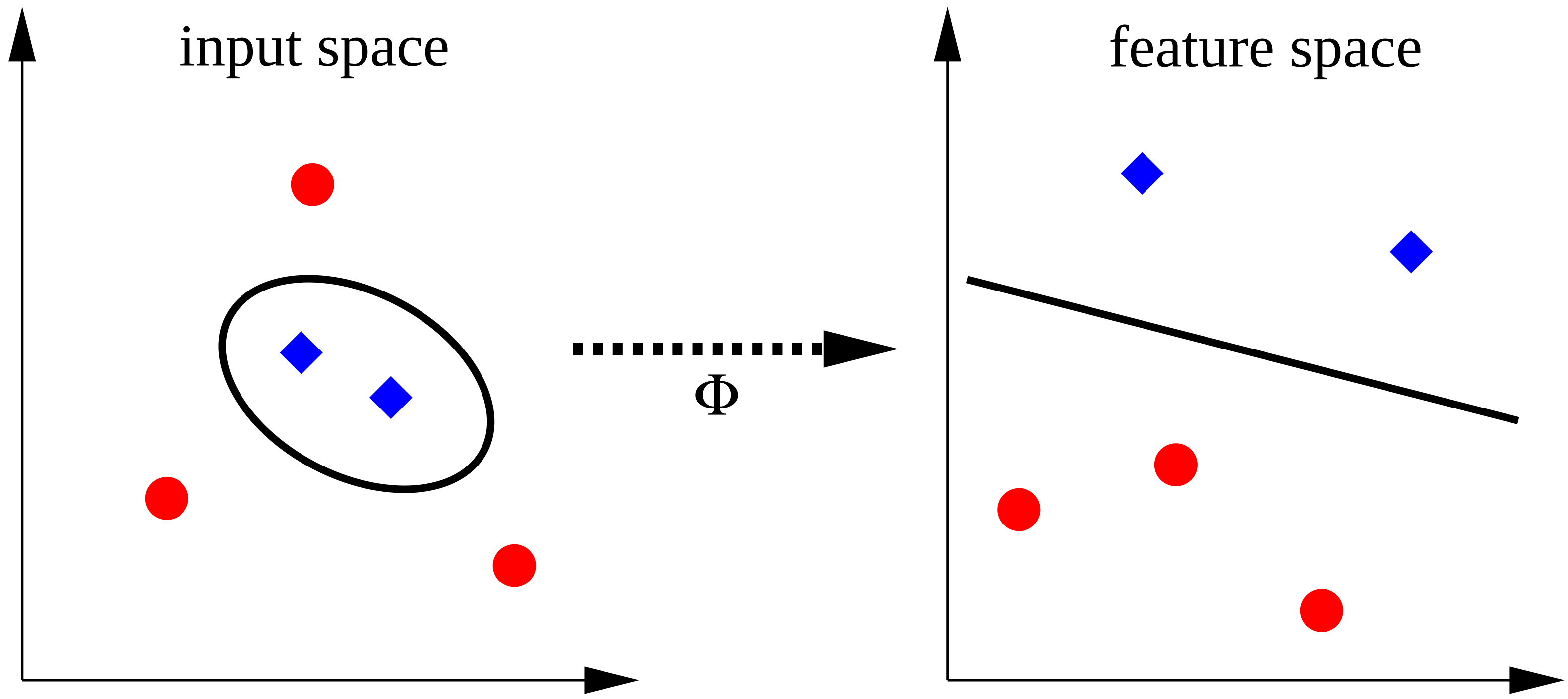
Conclusion

- the kernel corresponds to
  - a similarity measure for the data, or
  - a (linear) representation of the data, or
  - a hypothesis space for learning,
- kernels allow the formulation of a multitude of geometrical algorithms (Parzen windows, 2-sample tests, SVMs, kernel PCA,...)



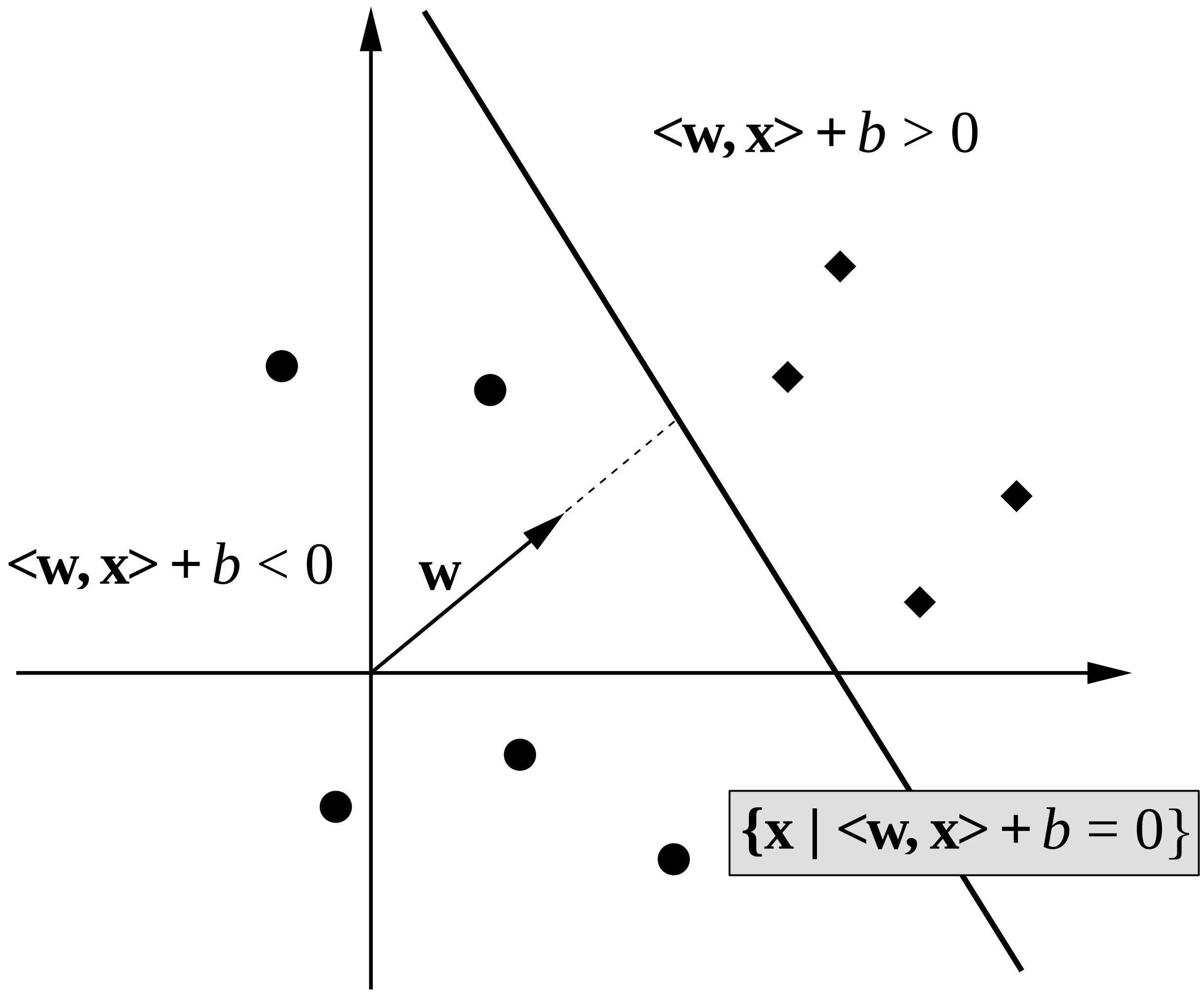
For further information, cf. `http://www.kernel-machines.org`.

Support Vector Classifiers

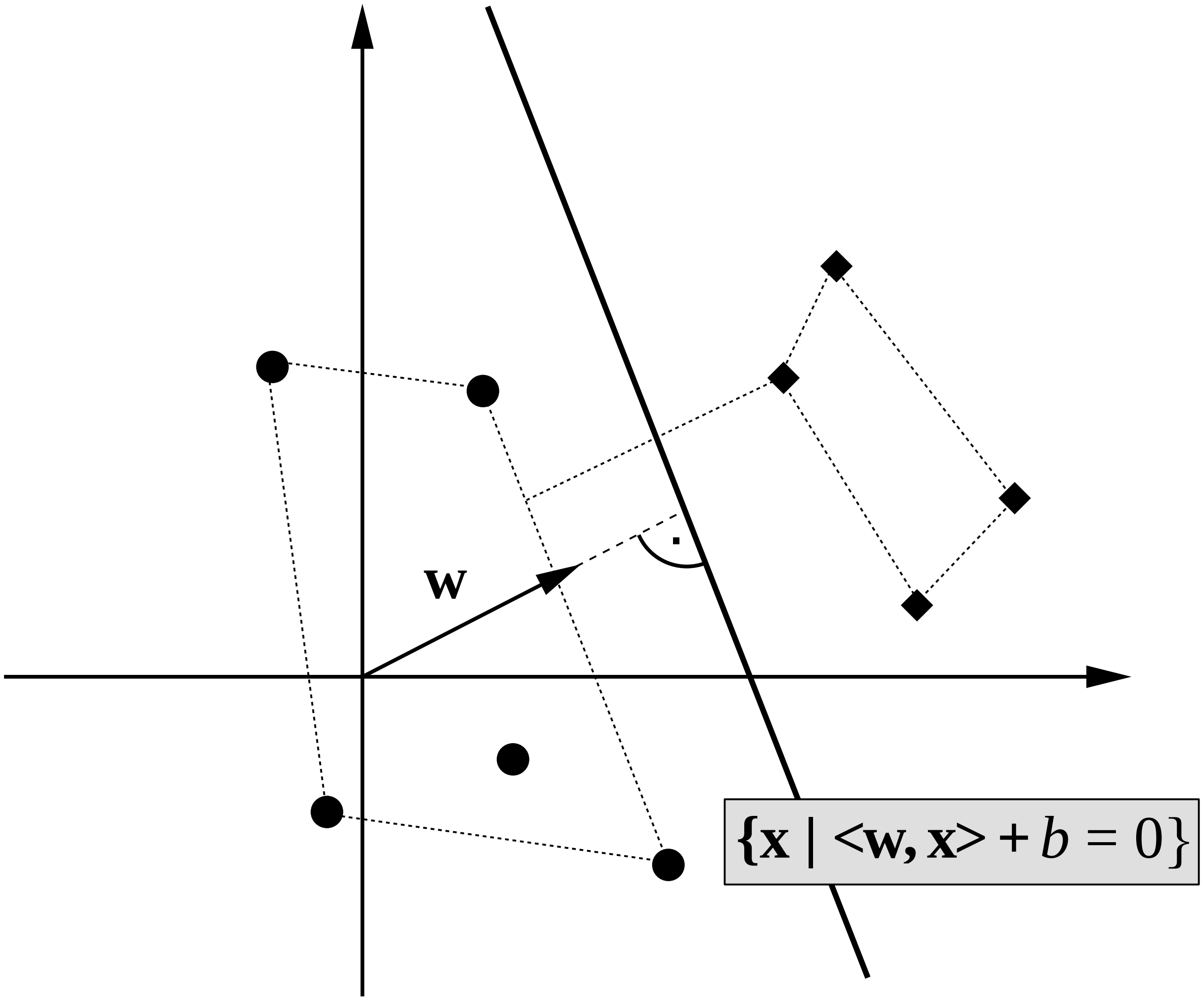


[6]

Separating Hyperplane



Optimal Separating Hyperplane



Eliminating the Scaling Freedom

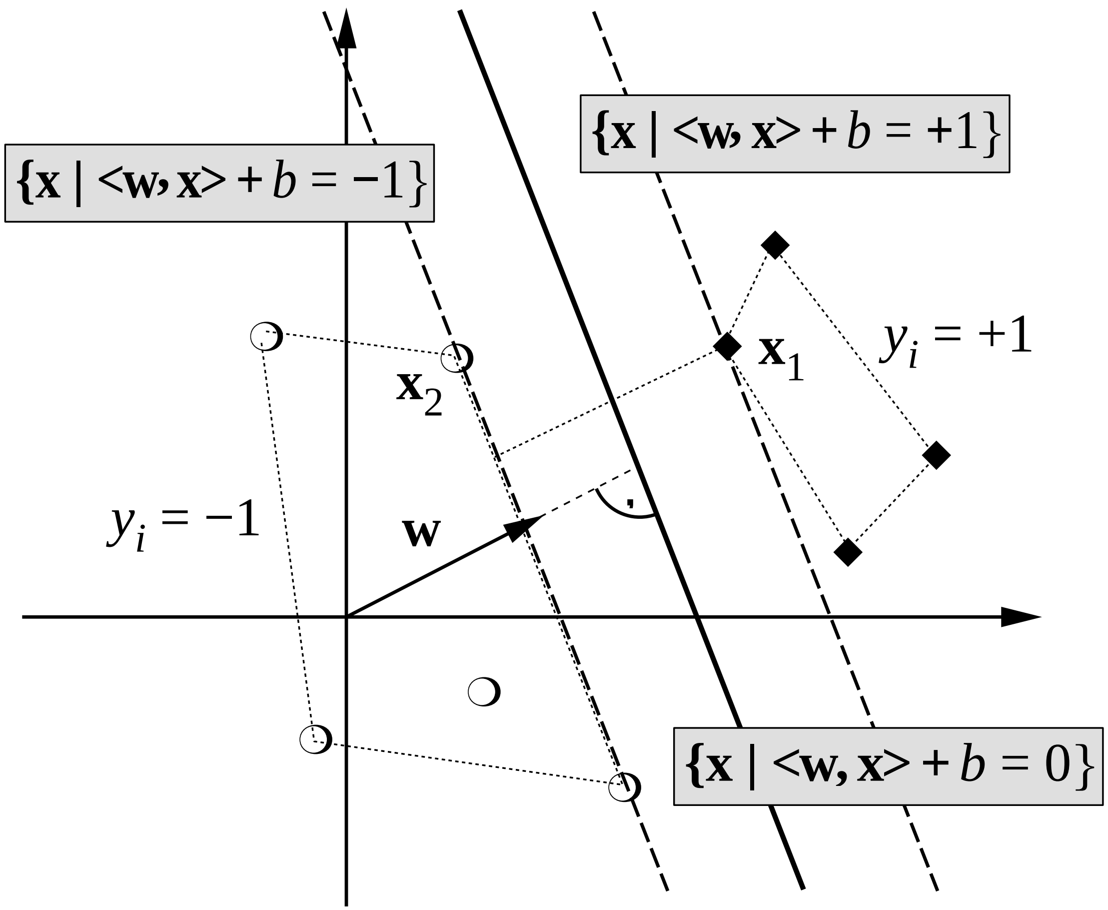
Note: if $c \neq 0$, then

$$\{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\} = \{\mathbf{x} | \langle c\mathbf{w}, \mathbf{x} \rangle + cb = 0\}.$$

Hence $(c\mathbf{w}, cb)$ describes the same hyperplane as $(\mathbf{w}, b)$.

Definition: The hyperplane is in *canonical* form w.r.t. $X^* = \{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ if $\min_{\mathbf{x}_i \in X} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$.

Canonical Optimal Hyperplane



Note:

$$\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = +1$$
$$\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = -1$$

$$\Rightarrow \quad \langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle = 2$$

$$\Rightarrow \quad \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle = \frac{2}{\|\mathbf{w}\|}$$

Note: if $c \neq 0$, then

$$\{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\} = \{\mathbf{x} | \langle c\mathbf{w}, \mathbf{x} \rangle + cb = 0\}.$$

Hence $(c\mathbf{w}, cb)$ describes the same hyperplane as $(\mathbf{w}, b)$.

Definition: The hyperplane is in *canonical* form w.r.t. $X^* = \{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ if $\min_{\mathbf{x}_i \in X} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$.

Note that for canonical hyperplanes, the distance of the closest point to the hyperplane ("margin") is $1/\|\mathbf{w}\|$:
$\min_{\mathbf{x}_i \in X} \left| \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_i \right\rangle + \frac{b}{\|\mathbf{w}\|} \right| = \frac{1}{\|\mathbf{w}\|}$.

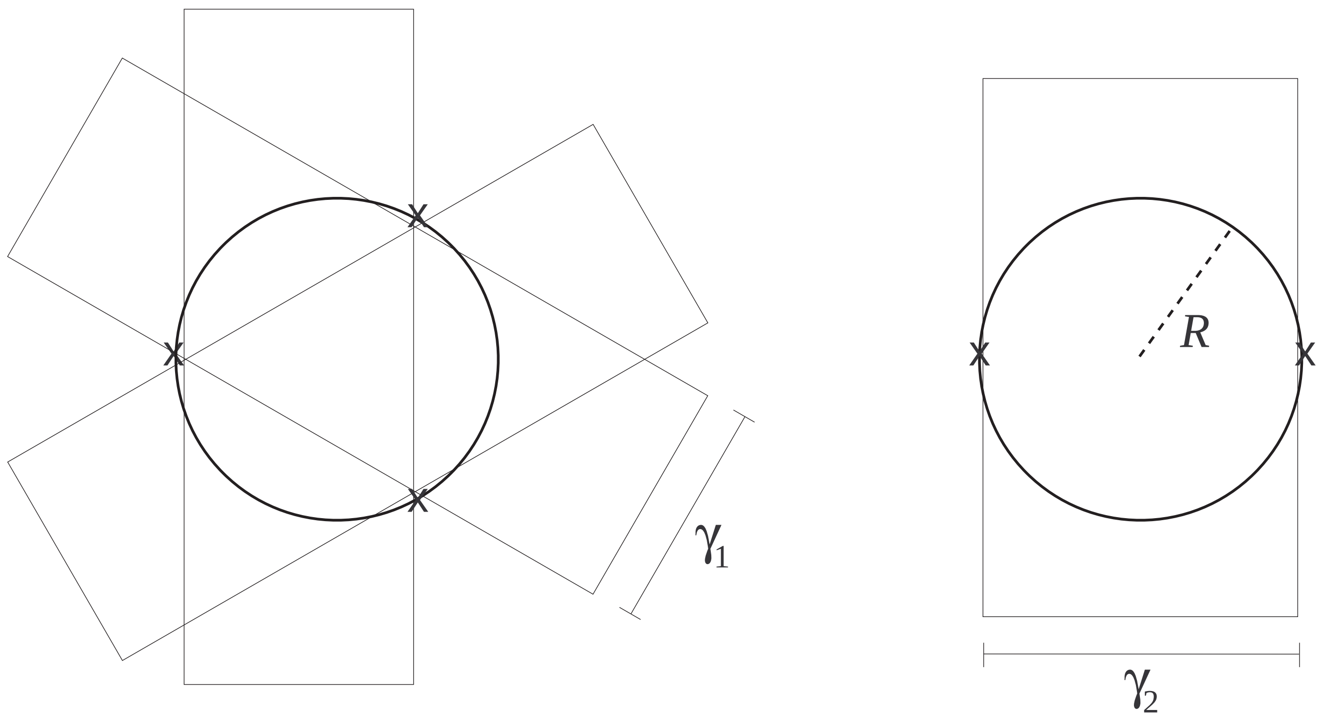
Theorem 7 (Vapnik [50]) *Consider hyperplanes* $\langle \mathbf{w}, \mathbf{x} \rangle = 0$ *where* $\mathbf{w}$ *is normalized such that they are in canonical form w.r.t. a set of points* $X^* = \{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$, *i.e.,*

$$\min_{i=1,\ldots,r} |\langle \mathbf{w}, \mathbf{x}_i \rangle| = 1.$$

The set of decision functions $f_{\mathbf{w}}(\mathbf{x}) = \operatorname{sgn} \langle \mathbf{x}, \mathbf{w} \rangle$ *defined on* X^* *and satisfying the constraint* $\|\mathbf{w}\| \leq \Lambda$ *has a VC dimension satisfying*

$$h \leq R^2 \Lambda^2.$$

Here, R *is the radius of the smallest sphere around the origin containing* X^*.



x
x
x
γ_1
R
x
x
γ_2

Proof Strategy (Gurvits, 1997)

Assume that $\mathbf{x}_1, \ldots, \mathbf{x}_r$ are shattered by canonical hyperplanes with $\|\mathbf{w}\| \leq \Lambda$, i.e., for all $y_1, \ldots, y_r \in \{\pm 1\}$,

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \text{ for all } i = 1, \ldots, r. \tag{5}$$

Two steps:

- prove that the more points we want to shatter (5), the larger $\|\sum_{i=1}^r y_i \mathbf{x}_i\|$ must be
- upper bound the size of $\|\sum_{i=1}^r y_i \mathbf{x}_i\|$ in terms of R

Combining the two tells us how many points we can at most shatter.

Part I

Summing (5) over $i = 1, \ldots, r$ yields

$$\left\langle \mathbf{w}, \left(\sum_{i=1}^{r} y_i \mathbf{x}_i \right) \right\rangle \geq r.$$

By the Cauchy-Schwarz inequality, on the other hand, we have

$$\left\langle \mathbf{w}, \left(\sum_{i=1}^{r} y_i \mathbf{x}_i \right) \right\rangle \leq \|\mathbf{w}\| \left\| \sum_{i=1}^{r} y_i \mathbf{x}_i \right\| \leq \Lambda \left\| \sum_{i=1}^{r} y_i \mathbf{x}_i \right\|.$$

Combine both:

$$\frac{r}{\Lambda} \leq \left\| \sum_{i=1}^{r} y_i \mathbf{x}_i \right\|. \tag{6}$$

Part II

Consider independent random labels $y_i \in \{\pm 1\}$, uniformly distributed (*Rademacher variables*).

$$
\begin{aligned}
\mathbf{E}\left[\left\|\sum_{i=1}^{r} y_i \mathbf{x}_i\right\|^2\right] &= \sum_{i=1}^{r} \mathbf{E}\left[\left\langle y_i \mathbf{x}_i, \sum_{j=1}^{r} y_j \mathbf{x}_j \right\rangle\right] \\
&= \sum_{i=1}^{r} \mathbf{E}\left[\left\langle y_i \mathbf{x}_i, \left(\left(\sum_{j \neq i} y_j \mathbf{x}_j\right) + y_i \mathbf{x}_i\right)\right\rangle\right] \\
&= \sum_{i=1}^{r} \left(\left(\sum_{j \neq i} \mathbf{E}\left[\langle y_i \mathbf{x}_i, y_j \mathbf{x}_j \rangle\right]\right) + \mathbf{E}\left[\langle y_i \mathbf{x}_i, y_i \mathbf{x}_i \rangle\right]\right) \\
&= \sum_{i=1}^{r} \mathbf{E}\left[\|y_i \mathbf{x}_i\|^2\right] = \sum_{i=1}^{r} \|\mathbf{x}_i\|^2
\end{aligned}
$$

Part II, ctd.

Since $\|\mathbf{x}_i\| \leq R$, we get

$$\mathbf{E}\left[\left\|\sum_{i=1}^{r} y_i \mathbf{x}_i\right\|^2\right] \leq rR^2.$$

- This holds for the *expectation* over the random choices of the labels, hence there must be at least one set of labels for which it also holds true. Use this set.

Hence

$$\left\|\sum_{i=1}^{r} y_i \mathbf{x}_i\right\|^2 \leq rR^2.$$

Part I and II Combined

Part I: $\left(\frac{r}{\Lambda}\right)^2 \leq \left\|\sum_{i=1}^{r} y_i \mathbf{x}_i\right\|^2$

Part II: $\left\|\sum_{i=1}^{r} y_i \mathbf{x}_i\right\|^2 \leq rR^2$

Hence

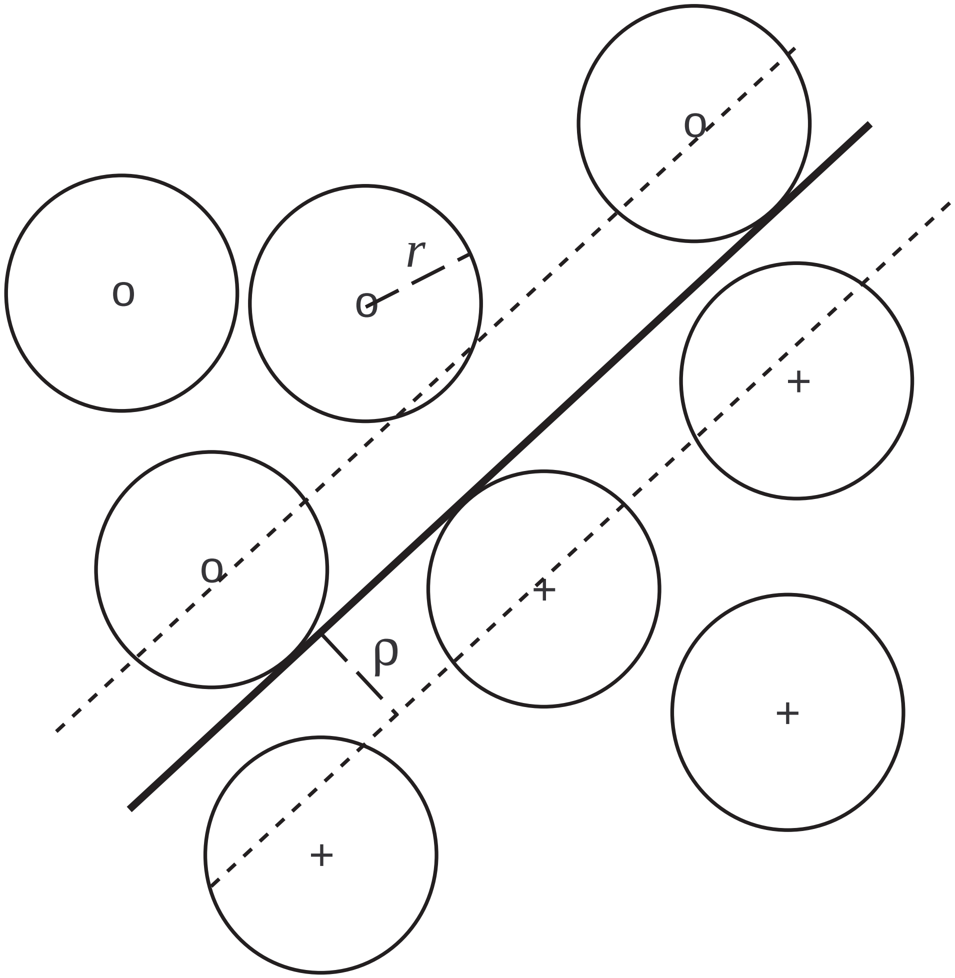
$$\frac{r^2}{\Lambda^2} \leq rR^2,$$

i.e.,

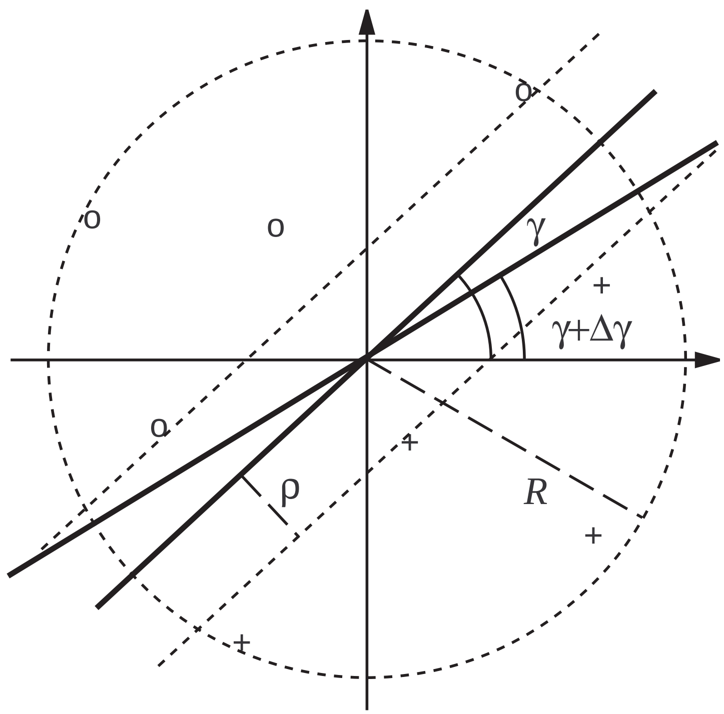
$$r \leq R^2 \Lambda^2,$$

completing the proof.

Pattern Noise as Maximum Margin Regularization



Maximum Margin vs. MDL — 2D Case



Can perturb γ by $\Delta\gamma$ with $|\Delta\gamma| < \arcsin\frac{\rho}{R}$ and still correctly separate the data.
Hence only need to store γ with accuracy $\Delta\gamma$ [39, 55].

Formulation as an Optimization Problem

Hyperplane with maximum margin: minimize

$$\|\mathbf{w}\|^2$$

(recall: margin $\sim 1/\|\mathbf{w}\|$) subject to

$$y_i \cdot [\langle \mathbf{w}, \mathbf{x}_i \rangle + b] \geq 1 \quad \text{for } i = 1 \dots m$$

(i.e. the training data are separated correctly).

Lagrange Function (e.g., [5])

Introduce Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i \cdot [\langle \mathbf{w}, \mathbf{x}_i \rangle + b] - 1\right).$$

L has to minimized w.r.t. the *primal variables* $\mathbf{w}$ and b and maximized with respect to the *dual variables* α_i

- if a constraint is violated, then $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 < 0 \longrightarrow$
  - $\cdot$ α_i will grow to increase L — how far?
  - $\cdot$ $\mathbf{w}$, b want to decrease L; i.e. they have to change such that the constraint is satisfied. If the problem is separable, this ensures that $\alpha_i < \infty$.
- similarly: if $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 > 0$, then $\alpha_i = 0$: otherwise, L could be increased by decreasing α_i *(KKT conditions)*

Derivation of the Dual Problem

At the extremum, we have

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0,$$

i.e.

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

and

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i.$$

Substitute both into L to get the *dual problem*

The Support Vector Expansion

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

where for all $i = 1, \dots, m$ either

$$y_i \cdot [\langle \mathbf{w}, \mathbf{x}_i \rangle + b] > 1 \quad \Longrightarrow \alpha_i = 0 \longrightarrow \mathbf{x}_i \text{ irrelevant}$$

or

$$y_i \cdot [\langle \mathbf{w}, \mathbf{x}_i \rangle + b] = 1 \quad (\textit{on} \text{ the margin}) \longrightarrow \mathbf{x}_i \text{ “Support Vector”}$$

The solution is determined by the examples on the margin.

Thus

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn}\left(\langle \mathbf{x}, \mathbf{w} \rangle + b\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b\right). \end{aligned}$$

Why it is Good to Have Few SVs

Leave out an example that does not become SV $\longrightarrow$ same solution.

Theorem [53]: Denote $\#\mathrm{SV}(m)$ the number of SVs obtained by training on m examples randomly drawn from $\mathrm{P}(\mathbf{x}, y)$, and $\mathbf{E}$ the expectation. Then

$$\mathbf{E}\left[\mathrm{Prob}(\text{test error})\right] \leq \frac{E\left[\#\mathrm{SV}(m)\right]}{m}$$

Here, Prob(test error) refers to the expected value of the risk, where the expectation is taken over training the SVM on samples of size $m-1$.

A Mechanical Interpretation

Assume that each SV $\mathbf{x}_i$ exerts a perpendicular force of size α_i and sign y_i on a solid plane sheet lying along the hyperplane.

Then the solution is mechanically stable:

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \quad \text{implies that the forces sum to zero}$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \quad \text{implies that the torques sum to zero,}$$

via

$$\sum_i \mathbf{x}_i \times y_i \alpha_i \cdot \mathbf{w}/\|\mathbf{w}\| = \mathbf{w} \times \mathbf{w}/\|\mathbf{w}\| = 0.$$

Dual Problem

Dual: maximize

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

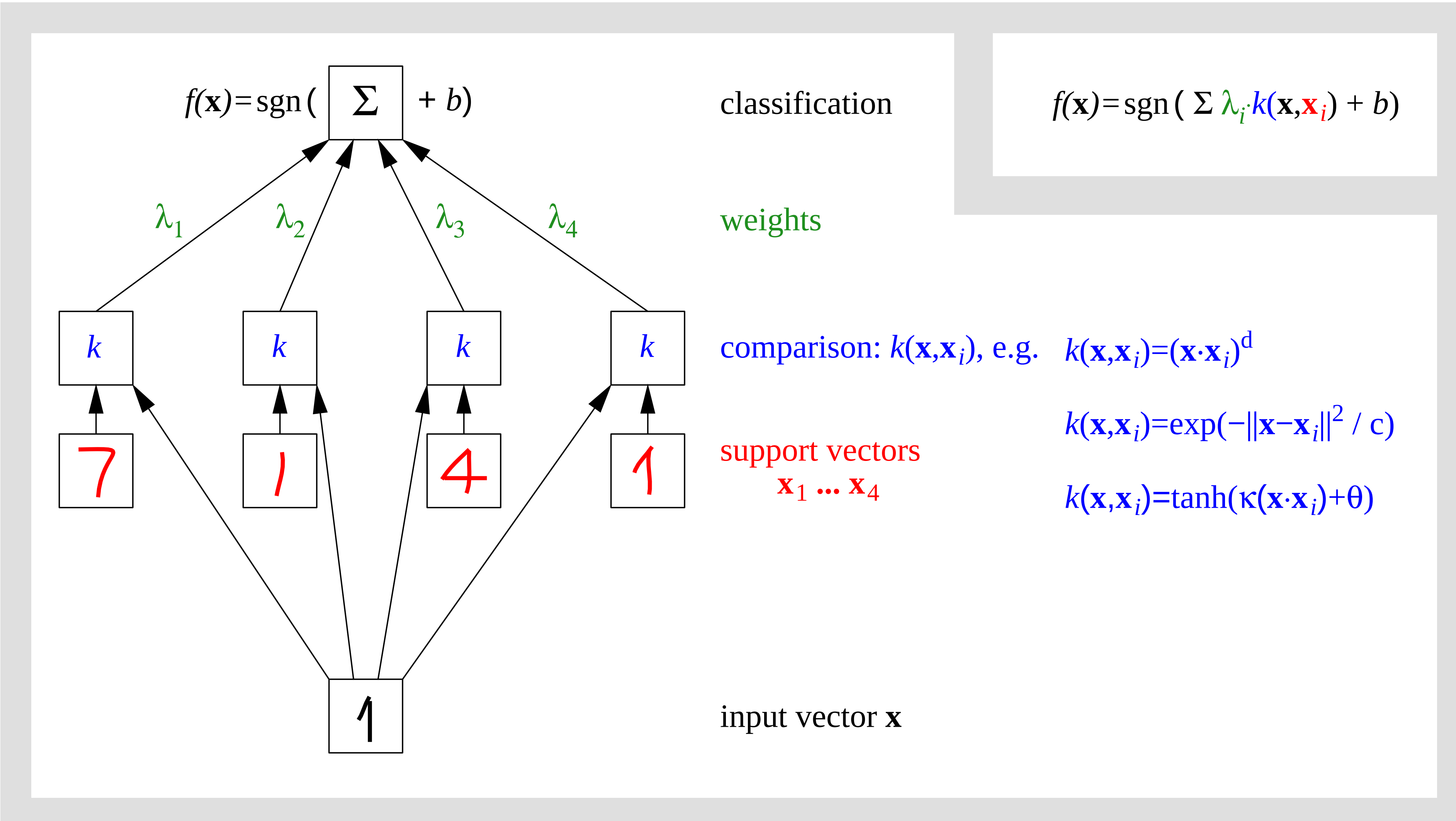
subject to

$$\alpha_i \geq 0, \quad i = 1, \ldots, m, \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

Both the final decision function and the function to be maximized are expressed in dot products $\longrightarrow$ can use a kernel to compute

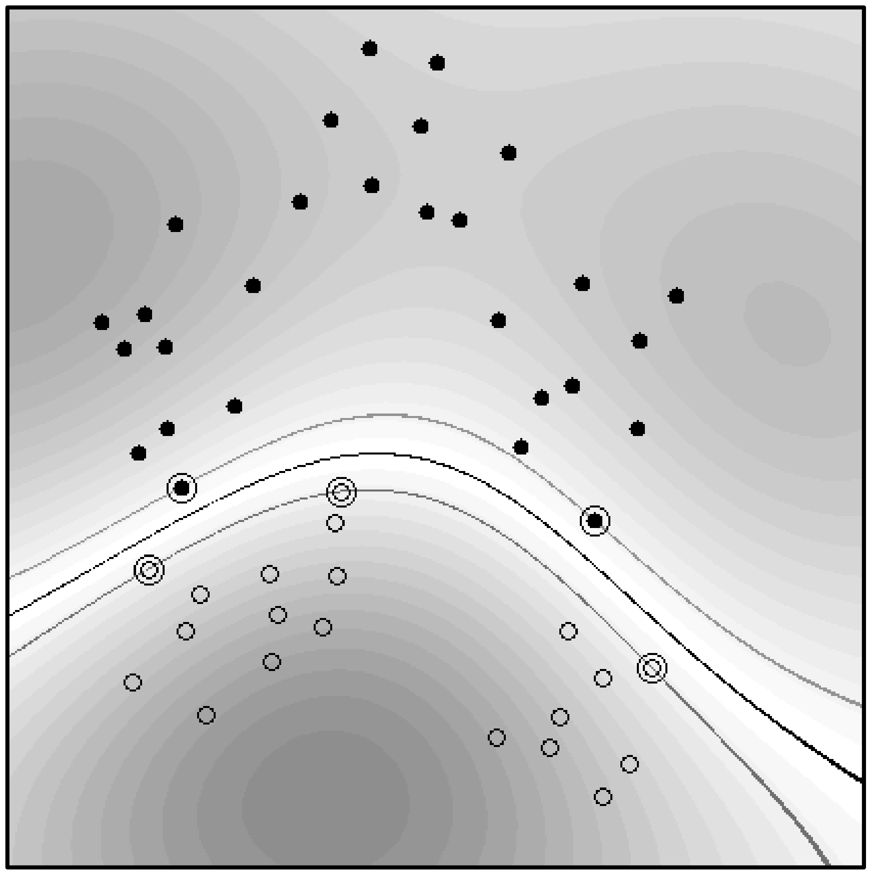
$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j).$$

The SVM Architecture



Toy Example with Gaussian Kernel

$$k(x, x') = \exp\left(-\|x - x'\|^2\right)$$



Nonseparable Problems *[3, 9]*

If $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ cannot be satisfied, then $\alpha_i \to \infty$.

Modify the constraint to

$$y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$$

with

$$\xi_i \geq 0$$

("*soft margin*") and add

$$C \cdot \sum_{i=1}^{m} \xi_i$$

in the objective function.

Soft Margin SVMs

C-SVM [9]: for $C > 0$, minimize

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

subject to $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$ (margin $1/\|\mathbf{w}\|$)

ν-SVM [41]: for $0 \leq \nu < 1$, minimize

$$\tau(\mathbf{w}, \boldsymbol{\xi}, \rho) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i}\xi_i$$

subject to $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i, \quad \xi_i \geq 0$ (margin $\rho/\|\mathbf{w}\|$)

The ν-Property

SVs: $\alpha_i > 0$
"margin errors:" $\xi_i > 0$

KKT-Conditions $\Longrightarrow$

- All margin errors are SVs.
- Not all SVs need to be margin errors.
  Those which are *not* lie exactly on the edge of the margin.

Proposition:
1. *fraction of Margin Errors* $\leq \nu \leq$ *fraction of SVs.*
2. *asymptotically:* ... $= \nu =$ ...

Duals, Using Kernels

C-SVM dual: maximize

$$W(\boldsymbol{\alpha}) = \sum\nolimits_i \alpha_i - \frac{1}{2}\sum\nolimits_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0.$

ν-SVM dual: maximize

$$W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum\nolimits_{ij} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $0 \leq \alpha_i \leq \frac{1}{m}, \quad \sum_i \alpha_i y_i = 0, \quad \sum_i \alpha_i \geq \nu$

In both cases: *decision function*:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum\nolimits_{i=1}^{m} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

Connection between ν-SVC and C-SVC

Proposition. If ν-SV classification leads to $\rho > 0$, then C-SV classification, with C set a priori to $1/\rho$, leads to the same decision function.

Proof. Minimize the primal target, then fix ρ, and minimize only over the remaining variables: nothing will change. Hence the obtained solution $\mathbf{w}_0, b_0, \boldsymbol{\xi}_0$ minimizes the primal problem of C-SVC, for $C = 1$, subject to

$$y_i \cdot (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq \rho - \xi_i.$$

To recover the constraint

$$y_i \cdot (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1 - \xi_i,$$

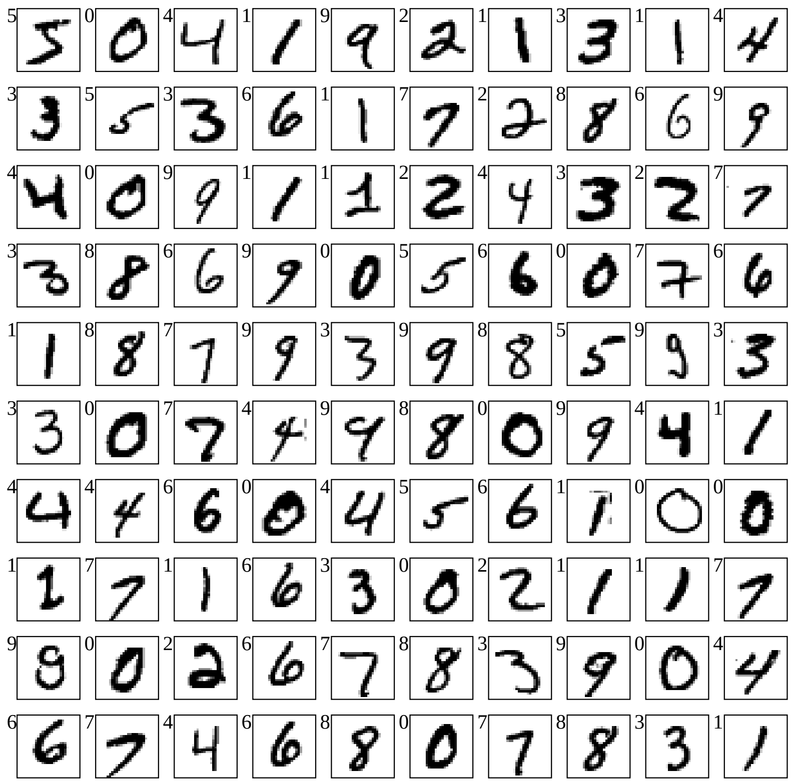
rescale to the set of variables $\mathbf{w}' = \mathbf{w}/\rho, b' = b/\rho, \boldsymbol{\xi}' = \boldsymbol{\xi}/\rho$. This leaves us, up to a constant scaling factor ρ^2, with the C-SV target with $C = 1/\rho$.

SVM Training

- naive approach: the complexity of maximizing
$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$
scales with the third power of the training set size m
- only SVs are relevant $\longrightarrow$ only compute $(k(\mathbf{x}_i, \mathbf{x}_j))_{ij}$ for SVs. Extract them iteratively by cycling through the training set in chunks [50].
- in fact, one can use chunks which do not even contain all SVs [32]. Maximize over these sub-problems, using your favorite optimizer.
- the extreme case: by making the sub-problems very small (just two points), one can solve them analytically [33].
- http://www.kernel-machines.org/software.html

MNIST Benchmark

handwritten character benchmark (60000 training & 10000 test examples, 28×28)



MNIST Error Rates

Classifier	test error	reference
linear classifier	8.4%	[7]
3-nearest-neighbour	2.4%	[7]
SVM	1.4%	[8]
Tangent distance	1.1%	[45]
LeNet4	1.1%	[28]
Boosted LeNet4	0.7%	[28]
Translation invariant SVM	0.56%	[11]

Note: the SVM used a polynomial kernel of degree 9, corresponding to a feature space of dimension $\approx 3.2 \cdot 10^{20}$.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[2] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2002.

[3] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[4] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, New York, 1984.

[5] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[6] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, July 1992. ACM Press.

[7] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of the 12th International Conference on Pattern Recognition and Neural Networks, Jerusalem*, pages 77–87. IEEE Computer Society Press, 1994.

[8] C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector learning machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 375–381, Cambridge, MA, 1997. MIT Press.

[9] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[10] D. Cox and F. O'Sullivan. Asymptotic analysis of penalized likelihood and related estimators. *Annals of Statistics*, 18:1676–1695, 1990.

[11] D. DeCoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 2001. Accepted for publication. Also: Technical Report JPL-MLTR-00-1, Jet Propulsion Laboratory, Pasadena, CA, 2000.

[12] F Dinuzzo and B Schölkopf. The representer theorem for Hilbert spaces: a necessary and sufficient condition. In P Bartlett, FCN Pereira, CJC. Burges, L Bottou, and KQ Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 189–196, 2012.

[13] R. M. Dudley. *Real analysis and probability.* Cambridge University Press, Cambridge, UK, 2002.

[14] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 171–203, Cambridge, MA, 2000. MIT Press.

[15] R. Fortet and E. Mourier. Convergence de la réparation empirique vers la réparation théorique. *Ann. Scient. École Norm. Sup.*, 70:266–285, 1953.

[16] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. Mach. Learn. Res.*, 5:73–99, 2004.

[17] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 489–496, Red Hook, NY, USA, 2008. Curran.

[18] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

[19] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, volume 19. The MIT Press, Cambridge, MA, 2007.

[20] A. Gretton, O. Bousquet, A.J. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. U. Simon, and E. Tomita, editors, *Proceedings Algorithmic Learning Theory*, pages 63–77, Berlin, Germany, 2005. Springer-Verlag.

[21] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *J. Mach. Learn. Res.*, 6:2075–2129, 2005.

[22] J. Ham, D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 369–376, New York, NY, USA, 2004. ACM Press.

[23] D. Haussler. Convolutional kernels on discrete structures. Technical Report UCSC-CRL-99-10, Computer Science Department, University of California at Santa Cruz, 1999.

[24] J. Huang, A.J. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, volume 19. The MIT Press, Cambridge, MA, 2007.

[25] J. Jacod and P. Protter. *Probability Essentials*. Springer, New York, 2000.

[26] G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.

[27] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[29] D. J. C. MacKay. Introduction to gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 133–165. Springer-Verlag, Berlin, 1998.

[30] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

[31] K Muandet, K Fukumizu, F Dinuzzo, and B Schölkopf. Learning from distributions via support measure machines. In P Bartlett, FCN Pereira, CJC. Burges, L Bottou, and KQ Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 10–18, 2012.

[32] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab., 1996.

[33] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

[34] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.

[35] A. Rényi. On measures of dependence. *Acta Math. Acad. Sci. Hungar.*, 10:441–451, 1959.

[36] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.

[37] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, München, 1997. Doktorarbeit, TU Berlin. Download: http://www.kernel-machines.org.

[38] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 640–646, Cambridge, MA, 1998. MIT Press.

[39] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[40] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[41] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[42] B. Schölkopf, J. Weston, E. Eskin, C. Leslie, and W. S. Noble. A kernel approach for learning from almost orthogonal patterns. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *13th European Conference on Machine Learning (ECML 2002) and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2002), Helsinki*, volume 2430/2431 of *Lecture Notes in Computer Science*, pages 511–528, Berlin, 2002. Springer.

[43] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

[44] H. Shimodaira. Improving predictive inference under convariance shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90, 2000.

[45] P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5. Proceedings of the 1992 Conference*, pages 50–58, San Mateo, CA, 1993. Morgan Kaufmann.

[46] A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

[47] A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In M. Hutter, R. A. Servedio, and E. Takimoto, editors, *Algorithmic Learning Theory: 18th International Conference*, pages 13–31, Berlin, 2007. Springer.

[48] A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Proc. Intl. Conf. Algorithmic Learning Theory*, volume 4754 of *LNAI*. Springer, 2007.

[49] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, 2:67–93, 2001.

[50] V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

[51] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, NY, 1995.

[52] V. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.

[53] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979).

[54] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.

[55] U. von Luxburg, O. Bousquet, and B. Schölkopf. A compression approach to support vector model selection. Technical Report 101, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2002. see more detailed JMLR version.

[56] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[57] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Department of Statistics, September 2003.

[58] H. L. Weinert. *Reproducing Kernel Hilbert Spaces*. Hutchinson Ross, Stroudsburg, PA, 1982.

[59] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer, 1998.

Regularization Interpretation of Kernel Machines

The norm in $\mathcal{H}$ can be interpreted as a regularization term (Girosi 1998, Smola et al., 1998, Evgeniou et al., 2000): if P is a regularization operator (mapping into a dot product space $\mathcal{D}$) such that k is Green's function of P^*P, then

$$\|\mathbf{w}\| = \|Pf\|,$$

where

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \Phi(x_i)$$

and

$$f(x) = \sum_i \alpha_i k(x_i, x).$$

Example: for the Gaussian kernel, P is a linear combination of differential operators.

$$
\begin{aligned}
\|\mathbf{w}\|^2 &= \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \\
&= \sum_{i,j} \alpha_i \alpha_j \left\langle k(x_i, .), \delta_{x_j}(.) \right\rangle \\
&= \sum_{i,j} \alpha_i \alpha_j \left\langle k(x_i, .), (P^* P k)(x_j, .) \right\rangle \\
&= \sum_{i,j} \alpha_i \alpha_j \left\langle (Pk)(x_i, .), (Pk)(x_j, .) \right\rangle_{\mathcal{D}} \\
&= \left\langle (P \sum_i \alpha_i k)(x_i, .), (P \sum_j \alpha_j k)(x_j, .) \right\rangle_{\mathcal{D}} \\
&= \|Pf\|^2,
\end{aligned}
$$

using $f(x) = \sum_i \alpha_i k(x_i, x)$.