

# Statistical Learning Theory and Kernel Methods

---

*Bernhard Schölkopf*

*Max Planck Institute for Biological Cybernetics*

# Statistical Learning Theory

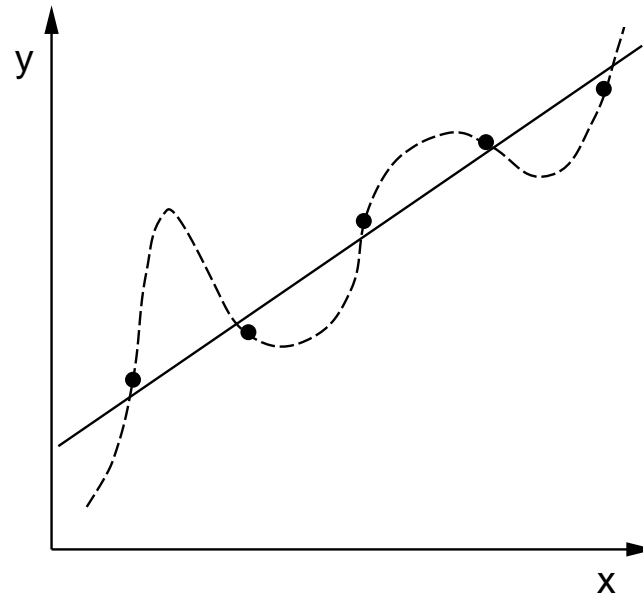
---

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*.

[43, 44]

## Example: Regression Estimation

---



- *Data*: input-output pairs  $(x_i, y_i) \in \mathbb{R} \times \mathbb{R}$
- *Regularity*:  $(x_1, y_1), \dots, (x_m, y_m)$  drawn from  $P(x, y)$
- *Learning*: choose a function  $f : \mathbb{R} \rightarrow \mathbb{R}$  such that the error, averaged over  $P$ , is minimized.
- *Problem*:  $P$  is unknown, so the average cannot be computed — need an “*induction* principle”

# Pattern Recognition

---

Learn  $f : \mathcal{X} \rightarrow \{\pm 1\}$  from examples

$(x_1, y_1), \dots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}$ , generated i.i.d. from  $P(x, y)$ ,

such that the expected misclassification error on a test set, also drawn from  $P(x, y)$ ,

$$R[f] = \int \frac{1}{2} |f(x) - y| dP(x, y),$$

is minimal (*Risk Minimization (RM)*).

**Problem:**  $P$  is unknown.  $\longrightarrow$  need an *induction principle*.

*Empirical risk minimization (ERM)*: replace the average over  $P(x, y)$  by an average over the training sample, i.e. **minimize the training error**

$$R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} |f(x_i) - y_i|$$

# Convergence of Means to Expectations

---

Law of large numbers:

$$R_{\text{emp}}[f] \rightarrow R[f]$$

as  $m \rightarrow \infty$ .

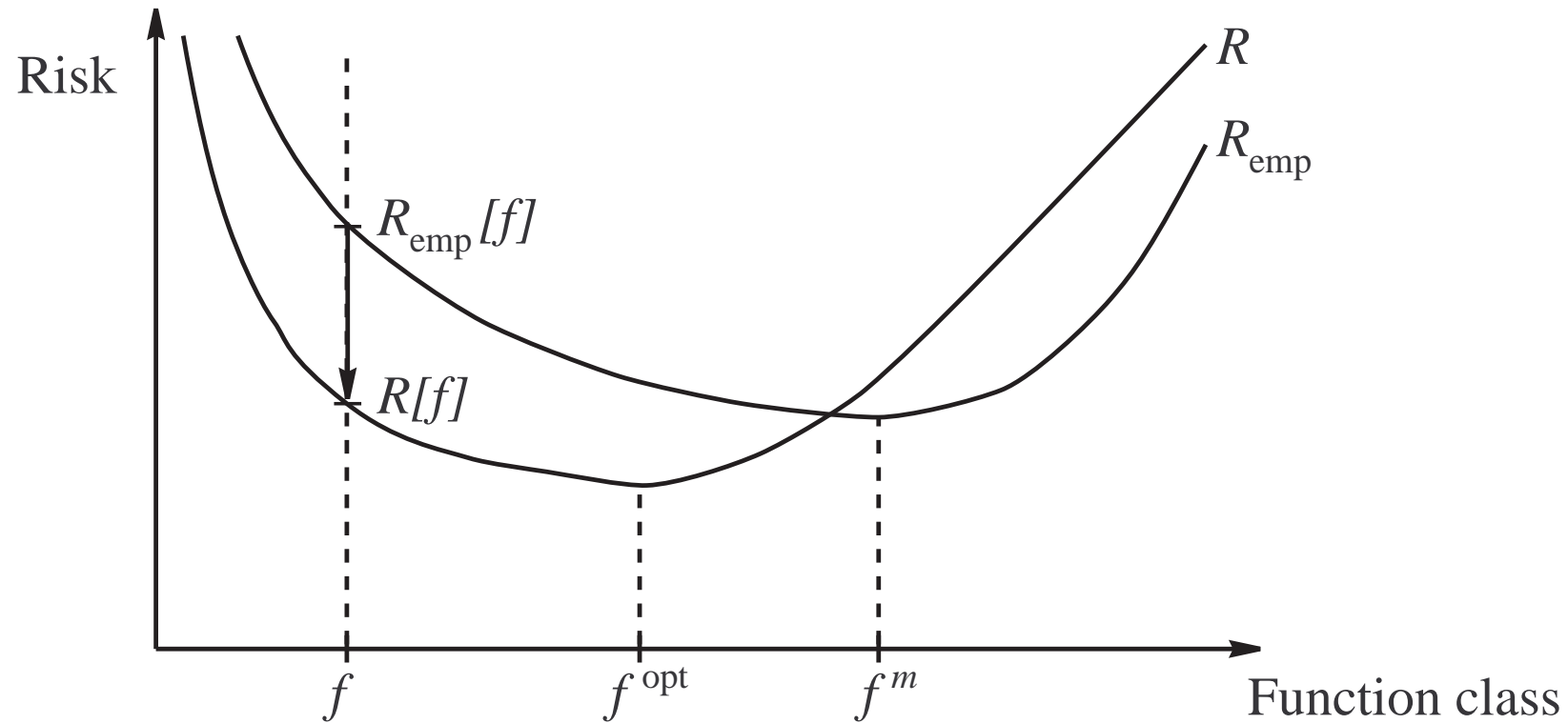
Does this imply that empirical risk minimization will give us the optimal result in the limit of infinite sample size (“*consistency*” of empirical risk minimization)?

**No.**

Need a *uniform* version of the law of large numbers. Uniform over *all functions that the learning machine can implement*.

# Consistency and Uniform Convergence

---



## The Importance of the Set of Functions

---

What about allowing *all* functions from  $\mathcal{X}$  to  $\{\pm 1\}$ ?

Training set  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in \mathcal{X} \times \{\pm 1\}$

Test patterns  $\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_{\bar{m}} \in \mathcal{X}$ ,

such that  $\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_{\bar{m}}\} \cap \{\mathbf{x}_1, \dots, \mathbf{x}_m\} = \{\}$ .

For any  $f$  there exists  $f^*$  s.t.:

1.  $f^*(\mathbf{x}_i) = f(\mathbf{x}_i)$  for all  $i$
2.  $f^*(\bar{\mathbf{x}}_j) \neq f(\bar{\mathbf{x}}_j)$  for all  $j$ .

Based on the training set alone, there is *no* means of choosing which one is better. On the test set, however, they give *opposite* results. There is 'no free lunch' [22, 51].

→ a restriction must be placed on the *functions* that we allow

# Restricting the Class of Functions

---

Two views:

1. Statistical Learning (VC) Theory: take into account the *capacity* of the class of functions that the learning machine can implement
2. The Bayesian Way: place *Prior distributions*  $P(f)$  over the class of functions



## Detailed Analysis

---

- loss  $\xi_i := \frac{1}{2}|f(x_i) - y_i|$  in  $\{0, 1\}$
- the  $\xi_i$  are independent Bernoulli trials
- empirical mean  $\frac{1}{m} \sum_{i=1}^m \xi_i$  (by def: equals  $R_{\text{emp}}[f]$ )
- expected value  $\mathbf{E}[\xi]$  (equals  $R[f]$ )

## Chernoff's Bound

---

$$\mathbb{P} \left\{ \left| \frac{1}{m} \sum_{i=1}^m \xi_i - \mathbf{E}[\xi] \right| \geq \epsilon \right\} \leq 2 \exp(-2m\epsilon^2)$$

- here,  $\mathbb{P}$  refers to the probability of getting a sample  $\xi_1, \dots, \xi_m$  with the property  $\left| \frac{1}{m} \sum_{i=1}^m \xi_i - \mathbf{E}[\xi] \right| \geq \epsilon$  (is a product measure)

Useful corollary: Given a  $2m$ -sample of Bernoulli trials, we have

$$\mathbb{P} \left\{ \left| \frac{1}{m} \sum_{i=1}^m \xi_i - \frac{1}{m} \sum_{i=m+1}^{2m} \xi_i \right| \geq \epsilon \right\} \leq 4 \exp \left( -\frac{m\epsilon^2}{2} \right).$$

## Chernoff's Bound, II

---

Translate this back into machine learning terminology: the probability of obtaining an  $m$ -sample where the training error and test error differ by more than  $\epsilon > 0$  is bounded by

$$\mathbb{P} \{ |R_{\text{emp}}[f] - R[f]| \geq \epsilon \} \leq 2 \exp(-2m\epsilon^2).$$

- refers to one fixed  $f$
- not allowed to look at the data before choosing  $f$ , hence not suitable as a bound on the test error of a learning algorithm using empirical risk minimization

# Uniform Convergence (Vapnik & Chervonenkis)

---

*Necessary and sufficient* conditions for nontrivial **consistency of empirical risk minimization (ERM)**:

One-sided convergence, uniformly over all functions that can be implemented by the learning machine.

$$\lim_{m \rightarrow \infty} P\left\{ \sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon \right\} = 0$$

for all  $\epsilon > 0$ .

- note that this takes into account the whole set of functions that can be implemented by the learning machine
- this is hard to check for a learning machine

Are there properties of **learning machines** ( $\equiv$  sets of functions) which ensure uniform convergence of risk?

## How to Prove a VC Bound

---

Take a closer look at  $P\{\sup_{f \in \mathcal{F}}(R[f] - R_{\text{emp}}[f]) > \epsilon\}$ .

Plan:

- if the function class  $\mathcal{F}$  contains only one function, then Chernoff's bound suffices:

$$P\{\sup_{f \in \mathcal{F}}(R[f] - R_{\text{emp}}[f]) > \epsilon\} \leq 2 \exp(-2m\epsilon^2).$$

- if there are finitely many functions, we use the 'union bound'
- even if there are infinitely many, then *on any finite sample* there are effectively only finitely many (use *symmetrization* and *capacity concepts*)

## The Case of Two Functions

---

Suppose  $\mathcal{F} = \{f_1, f_2\}$ . Rewrite

$$\mathbb{P}\left\{\sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon\right\} = \mathbb{P}(C_\epsilon^1 \cup C_\epsilon^2),$$

where

$$C_\epsilon^i := \{(x_1, y_1), \dots, (x_m, y_m) \mid (R[f_i] - R_{\text{emp}}[f_i]) > \epsilon\}$$

denotes the event that the risks of  $f_i$  differ by more than  $\epsilon$ .

The RHS equals

$$\begin{aligned} \mathbb{P}(C_\epsilon^1 \cup C_\epsilon^2) &= \mathbb{P}(C_\epsilon^1) + \mathbb{P}(C_\epsilon^2) - \mathbb{P}(C_\epsilon^1 \cap C_\epsilon^2) \\ &\leq \mathbb{P}(C_\epsilon^1) + \mathbb{P}(C_\epsilon^2). \end{aligned}$$

Hence by Chernoff's bound

$$\begin{aligned} \mathbb{P}\left\{\sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon\right\} &\leq \mathbb{P}(C_\epsilon^1) + \mathbb{P}(C_\epsilon^2) \\ &\leq 2 \cdot 2 \exp(-2m\epsilon^2). \end{aligned}$$

## The Union Bound

---

Similarly, if  $\mathcal{F} = \{f_1, \dots, f_n\}$ , we have

$$\mathbb{P}\left\{\sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon\right\} = \mathbb{P}(C_\epsilon^1 \cup \dots \cup C_\epsilon^n),$$

and

$$\mathbb{P}(C_\epsilon^1 \cup \dots \cup C_\epsilon^n) \leq \sum_{i=1}^n \mathbb{P}(C_\epsilon^i).$$

Use Chernoff for each summand, to get an extra factor  $n$  in the bound.

Note: this becomes an equality if and only if all the events  $C_\epsilon^i$  involved are *disjoint*.

## Infinite Function Classes

---

- Note: empirical risk only refers to  $m$  points. On these points, the functions of  $\mathcal{F}$  can take at most  $2^m$  values
- for  $R_{\text{emp}}$ , the function class thus “looks” finite
- how about  $R$ ?
- need to use a trick



## Symmetrization

---

**Lemma 1 (Vapnik & Chervonenkis (e.g., [42, 12]))**

*For  $m\epsilon^2 \geq 2$  we have*

$$P\left\{\sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon\right\} \leq 2P\left\{\sup_{f \in \mathcal{F}} (R_{\text{emp}}[f] - R'_{\text{emp}}[f]) > \epsilon/2\right\}$$

*Here, the first  $P$  refers to the distribution of iid samples of size  $m$ , while the second one refers to iid samples of size  $2m$ . In the latter case,  $R_{\text{emp}}$  measures the loss on the first half of the sample, and  $R'_{\text{emp}}$  on the second half.*

# Shattering Coefficient

---

- Hence, we only need to consider the maximum size of  $\mathcal{F}$  on  $2m$  points. Call it  $\mathcal{N}(\mathcal{F}, 2m)$ .
- $\mathcal{N}(\mathcal{F}, 2m) = \max.$  number of different outputs  $(y_1, \dots, y_{2m})$  that the function class can generate on  $2m$  points — in other words, the max. number of different ways the function class can separate  $2m$  points into two classes.
- $\mathcal{N}(\mathcal{F}, 2m) \leq 2^{2m}$
- if  $\mathcal{N}(\mathcal{F}, 2m) = 2^{2m}$ , then the function class is said to *shatter*  $2m$  points.

## Putting Everything Together

---

We now use (1) symmetrization, (2) the shattering coefficient, and (3) the union bound, to get

$$\begin{aligned} & \mathbb{P}\{\sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon\} \\ & \leq 2\mathbb{P}\{\sup_{f \in \mathcal{F}} (R_{\text{emp}}[f] - R'_{\text{emp}}[f]) > \epsilon/2\} \\ & = 2\mathbb{P}\{(R_{\text{emp}}[f_1] - R'_{\text{emp}}[f_1]) > \epsilon/2 \vee \dots \vee (R_{\text{emp}}[f_{\mathcal{N}(\mathcal{F}, 2m)}] - R'_{\text{emp}}[f_{\mathcal{N}(\mathcal{F}, 2m)}]) > \epsilon/2\} \\ & \leq \sum_{n=1}^{\mathcal{N}(\mathcal{F}, 2m)} 2\mathbb{P}\{(R_{\text{emp}}[f_n] - R'_{\text{emp}}[f_n]) > \epsilon/2\}. \end{aligned}$$

ctd.

---

Use Chernoff's bound for each term:\*

$$\mathbb{P} \left\{ \frac{1}{m} \sum_{i=1}^m \xi_i - \frac{1}{m} \sum_{i=m+1}^{2m} \xi_i \geq \epsilon \right\} \leq 2 \exp \left( -\frac{m\epsilon^2}{2} \right).$$

This yields

$$\mathbb{P} \left\{ \sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon \right\} \leq 4\mathcal{N}(\mathcal{F}, 2m) \exp \left( -\frac{m\epsilon^2}{8} \right).$$

- provided that  $\mathcal{N}(\mathcal{F}, 2m)$  does not grow exponentially in  $m$ , this is nontrivial
- such bounds are called *VC type inequalities*
- two types of randomness: (1) the  $\mathbb{P}$  refers to the drawing of the training examples, and (2)  $R[f]$  is an expectation over the drawing of test examples.

\* A rigorous treatment would need to use a second randomization over permutations of the  $2m$ -sample, see [34].

## Confidence Intervals

---

Rewrite the bound: specify the probability with which we want  $R$  to be close to  $R_{\text{emp}}$ , and solve for  $\epsilon$ :

With a probability of at least  $1 - \delta$ ,

$$R[f] \leq R_{\text{emp}}[f] + \sqrt{\frac{8}{m} \left( \ln(\mathcal{N}(\mathcal{F}, 2m)) + \ln \frac{4}{\delta} \right)}.$$

This bound holds independent of  $f$ ; in particular, it holds for the function  $f^m$  minimizing the empirical risk.

## Discussion

---

- tighter bounds are available (better constants etc.)
- cannot minimize the bound over  $f$
- other capacity concepts can be used

## VC Entropy

---

On an example  $(\mathbf{x}, \mathbf{y})$ ,  $f$  causes a loss

$$\xi(x, y, f(x)) = \frac{1}{2}|f(x) - y| \in \{0, 1\}.$$

For a larger sample  $(x_1, y_1), \dots, (x_m, y_m)$ , the different functions  $f \in \mathcal{F}$  lead to a *set* of loss vectors

$$\boldsymbol{\xi}_f = (\xi(x_1, y_1, f(x_1)), \dots, \xi(x_m, y_m, f(x_m))),$$

whose cardinality we denote by

$$\mathcal{N}(\mathcal{F}, (x_1, y_1) \dots, (x_m, y_m)).$$

The *VC entropy* is defined as

$$H_{\mathcal{F}}(m) = \mathbf{E} [\ln \mathcal{N}(\mathcal{F}, (x_1, y_1) \dots, (x_m, y_m))],$$

where the expectation is taken over the random generation of the  $m$ -sample  $(x_1, y_1) \dots, (x_m, y_m)$  from  $P$ .

$H_{\mathcal{F}}(m)/m \rightarrow 0 \iff$  uniform convergence of risks (hence consistency)

## Further PR Capacity Concepts

---

- exchange 'E' and 'ln': *annealed entropy*.

$H_{\mathcal{F}}^{\text{ann}}(m)/m \rightarrow 0 \iff$  exponentially fast uniform convergence

- take 'max' instead of 'E': *growth function*.

Note that  $G_{\mathcal{F}}(m) = \ln \mathcal{N}(\mathcal{F}, m)$ .

$G_{\mathcal{F}}(m)/m \rightarrow 0 \iff$  exponential convergence for all underlying distributions P.

$G_{\mathcal{F}}(m) = m \cdot \ln(2)$  for all  $m \iff$  for any  $m$ , all loss vectors can be generated, i.e., the  $m$  points can be chosen such that by using functions of the learning machine, they can be separated in all  $2^m$  possible ways (*shattered*).



## Structure of the Growth Function

---

**Either**  $G_{\mathcal{F}}(m) = m \cdot \ln(2)$  for all  $m \in \mathbb{N}$

**Or** there exists some *maximal*  $m$  for which the above is possible. Call this number the *VC-dimension*, and denote it by  $h$ . For  $m > h$ ,

$$G_{\mathcal{F}}(m) \leq h \left( \ln \frac{m}{h} + 1 \right).$$

Nothing “in between” linear growth and logarithmic growth is possible.

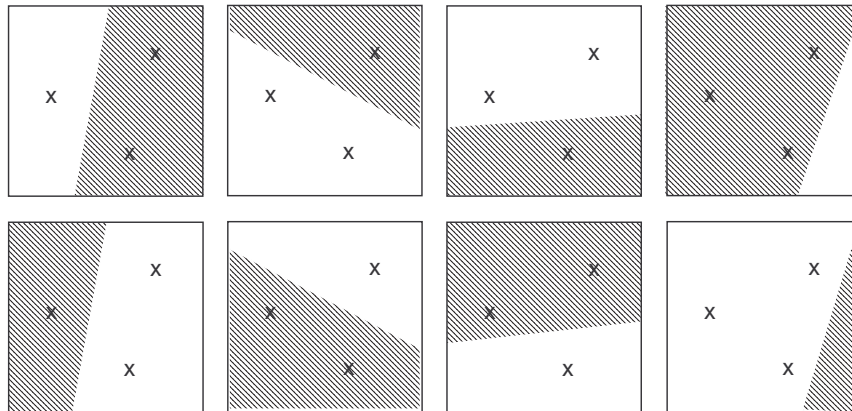
## VC-Dimension: Example

---

Half-spaces in  $\mathbb{R}^2$ :

$$f(x, y) = \text{sgn}(a + bx + cy), \quad \text{with parameters } a, b, c \in \mathbb{R}$$

- Clearly, we can shatter three non-collinear points.
- But we can never shatter four points.
- Hence the VC dimension is  $h = 3$  (in this case, equal to the number of parameters)



## A Typical Bound for Pattern Recognition

---

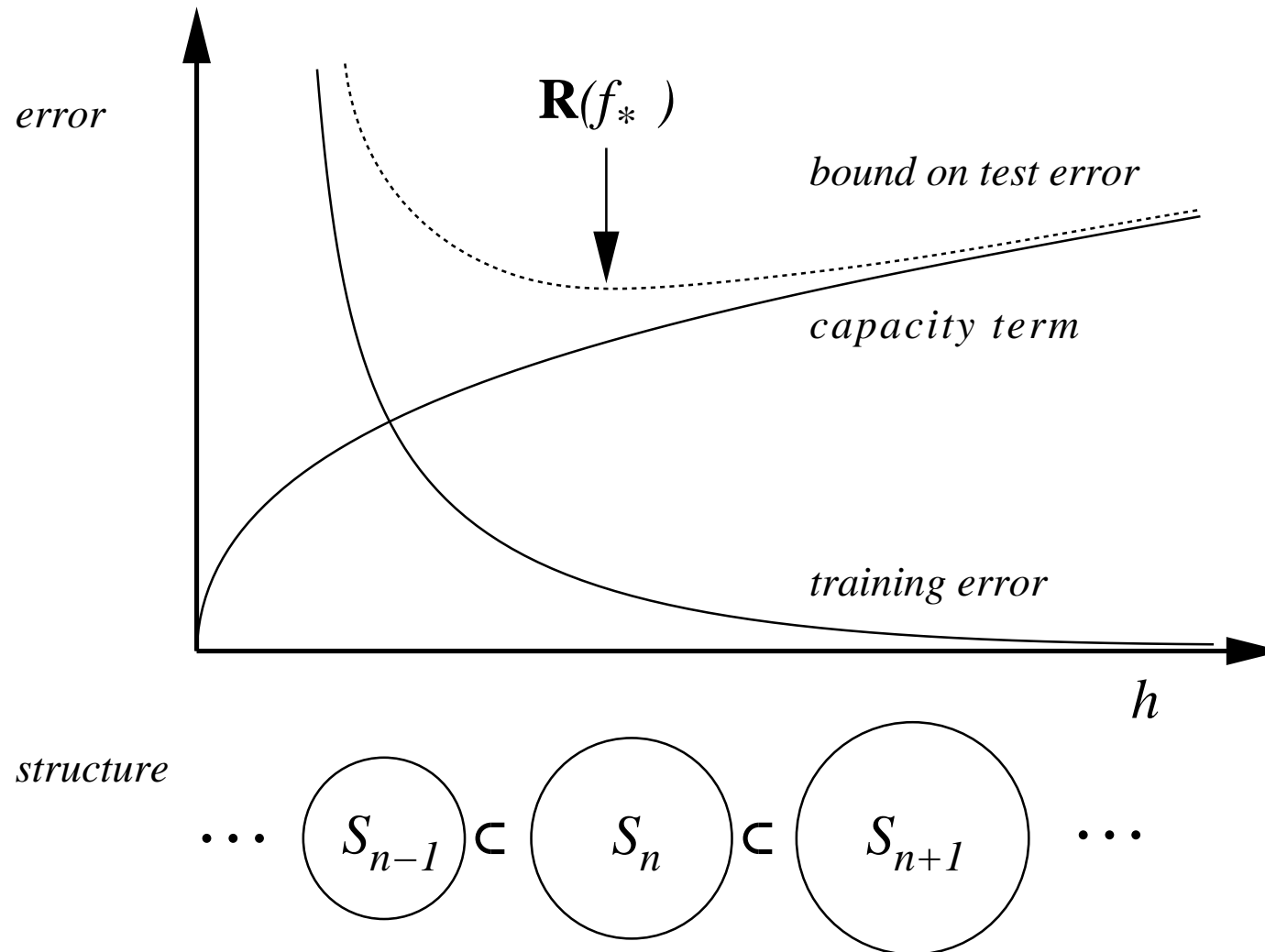
For any  $f \in \mathcal{F}$  and  $m > h$ , with a probability of at least  $1 - \delta$ ,

$$R[f] \leq R_{\text{emp}}[f] + \sqrt{\frac{h \left( \log \frac{2m}{h} + 1 \right) - \log(\delta/4)}{m}}$$

holds.

- does this mean, that we can learn *anything*?
- The study of the consistency of ERM has thus led to concepts and results which lets us formulate another induction principle (structural risk minimization)

# SRM



## 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

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

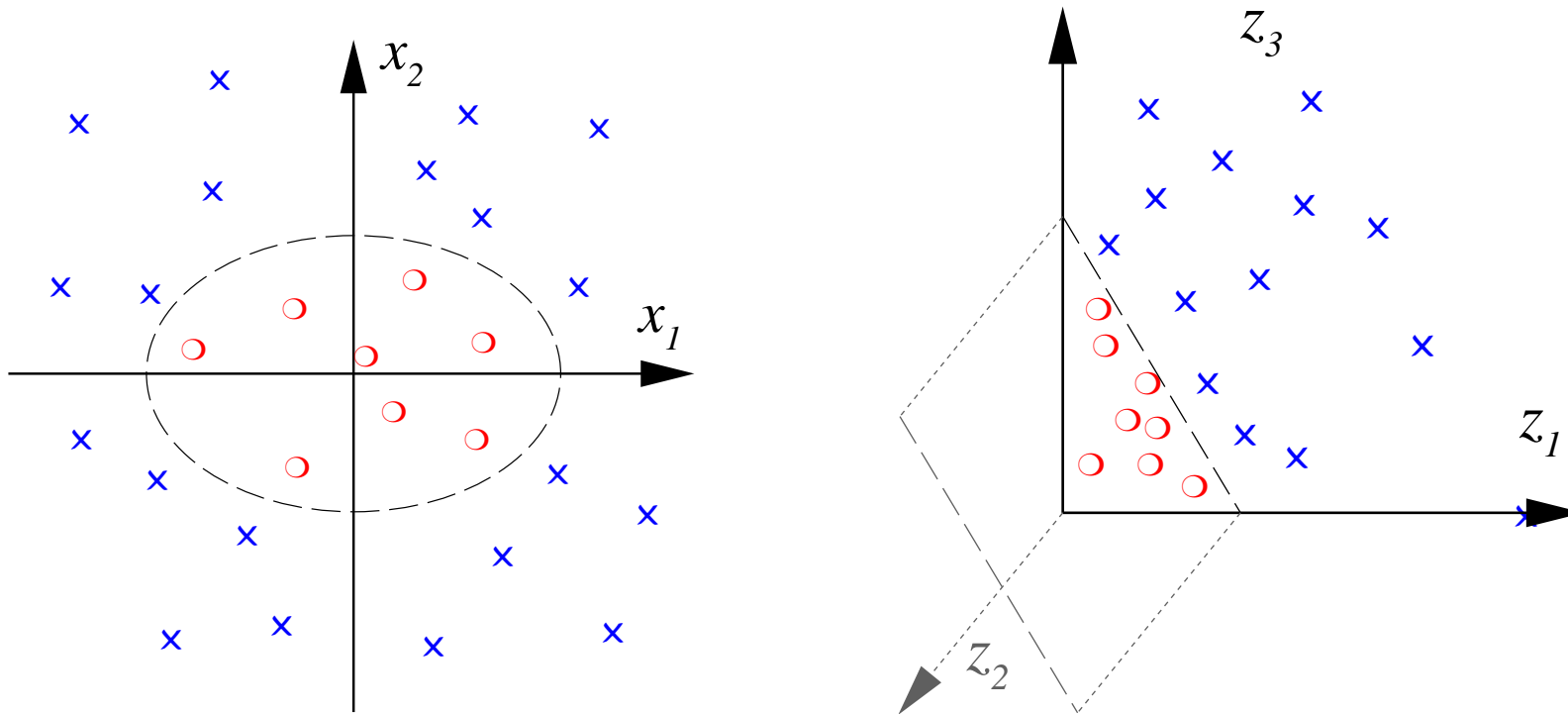
where  $\mathcal{H}$  is a dot product space, and learn the mapping from  $\Phi(x)$  to  $y$  [6].

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

## Example: All Degree 2 Monomials

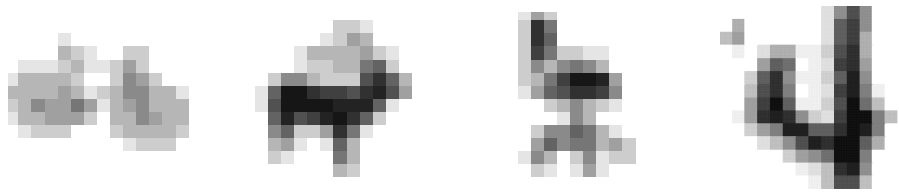
---

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



# 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, $N = d = 2$

---

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

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

## The Kernel Trick, II

---

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

$$\begin{aligned}\langle x, x' \rangle^d &= \left( \sum_{j=1}^N x_j \cdot x'_j \right)^d \\ &= \sum_{j_1, \dots, j_d=1}^N x_{j_1} \cdots x_{j_d} \cdot x'_{j_1} \cdots x'_{j_d} = \langle \Phi(x), \Phi(x') \rangle,\end{aligned}$$

where  $\Phi$  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  $\psi_i$  and eigenvalues  $\lambda_i \geq 0$  [28].

# 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:

$$\begin{aligned} \langle \Phi(x), \Phi(x') \rangle &= \left\langle \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}, \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x') \\ \sqrt{\lambda_2}\psi_2(x') \\ \vdots \end{pmatrix} \right\rangle \\ &= \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x') = k(x, x') \end{aligned}$$

## The Kernel Trick — Summary

---

- *any* algorithm that only depends on dot products can benefit from the kernel trick
- 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:

$$\text{Polynomial } k(x, x') = (\langle x, x' \rangle + c)^d$$

$$\text{Sigmoid } k(x, x') = \tanh(\kappa \langle x, x' \rangle + \Theta)$$

$$\text{Gaussian } k(x, x') = \exp(-\|x - x'\|^2 / (2\sigma^2))$$

- Kernels are also known as covariance functions [49, 47, 50, 27]

## Positive Definite Kernels

---

It can be shown that the admissible class of kernels coincides with the one of **positive definite (pd) kernels**: kernels which are symmetric (i.e.,  $k(x, x') = k(x', x)$ ), and for

- any set of training points  $x_1, \dots, x_m \in \mathcal{X}$  and
- any  $a_1, \dots, a_m \in \mathbb{R}$

satisfy

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

$K$  is called the *Gram matrix* or *kernel matrix*.

If for pairwise distinct points,  $\sum_{i,j} a_i a_j K_{ij} = 0 \implies a = 0$ , call it **strictly** positive definite.

## Elementary Properties of PD Kernels

---

*Kernels from Feature Maps.*

If  $\Phi$  maps  $\mathcal{X}$  into a dot product space  $\mathcal{H}$ , then  $\langle \Phi(x), \Phi(x') \rangle$  is a pd kernel on  $\mathcal{X} \times \mathcal{X}$ .

*Positivity on the Diagonal.*

$k(x, x) \geq 0$  for all  $x \in \mathcal{X}$

*Cauchy-Schwarz Inequality.*

$k(x, x')^2 \leq k(x, x)k(x', x')$  (Hint: compute the determinant of the Gram matrix)

*Vanishing Diagonals.*

$k(x, x) = 0$  for all  $x \in \mathcal{X} \implies k(x, x') = 0$  for all  $x, x' \in \mathcal{X}$

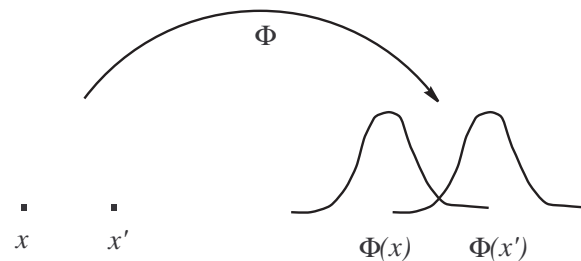
# The Feature Space for PD Kernels

[4, 1, 33]

- define a feature map

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

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.,  $\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$
- complete the space to get a *reproducing kernel Hilbert space*



# Turn it Into a Linear Space

---

Form linear combinations

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

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

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

## Endow it With a Dot Product

---

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

- 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

$$f(\cdot) = k(\cdot, x).$$

In this case, we have

$$\langle k(\cdot, x), g \rangle = g(x).$$

- If moreover

$$g(\cdot) = k(\cdot, x'),$$

we have

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

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

$$\begin{aligned} \sum_{ij} \gamma_i \gamma_j \langle f_i, f_j \rangle &= \left\langle \sum_i \gamma_i f_i, \sum_j \gamma_j f_j \right\rangle =: \langle f, f \rangle \\ &= \left\langle \sum_i \alpha_i k(\cdot, x_i), \sum_i \alpha_i k(\cdot, x_i) \right\rangle = \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \end{aligned}$$

- furthermore, it is *strictly* positive definite:

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

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

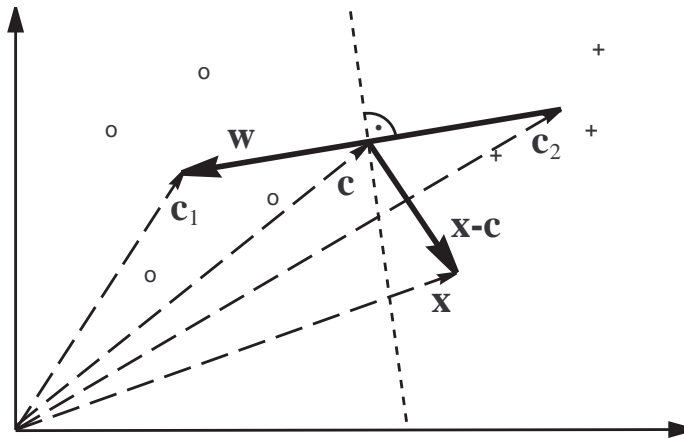
- Complete the space in the corresponding norm to get a Hilbert space  $\mathcal{H}_k$ .

## 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. [34]

---

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

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

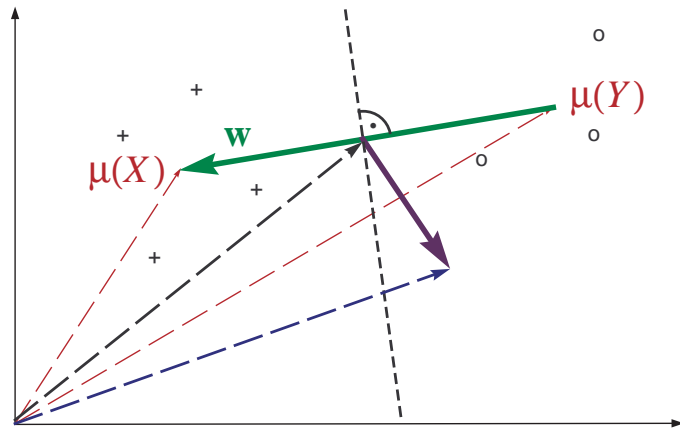
## An Example of a Kernel Algorithm, ctd.

---

- Demo
- 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  $X := \{x_1, \dots, x_m\} \subset \mathcal{X}$

Negative class  $Y := \{y_1, \dots, y_n\} \subset \mathcal{X}$

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 2** *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*

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

*then  $X = Y$ .*

## Proof (by contradiction)

---

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

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

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  $\sum_j \gamma_j k(z_j, \cdot)$  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

---

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

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

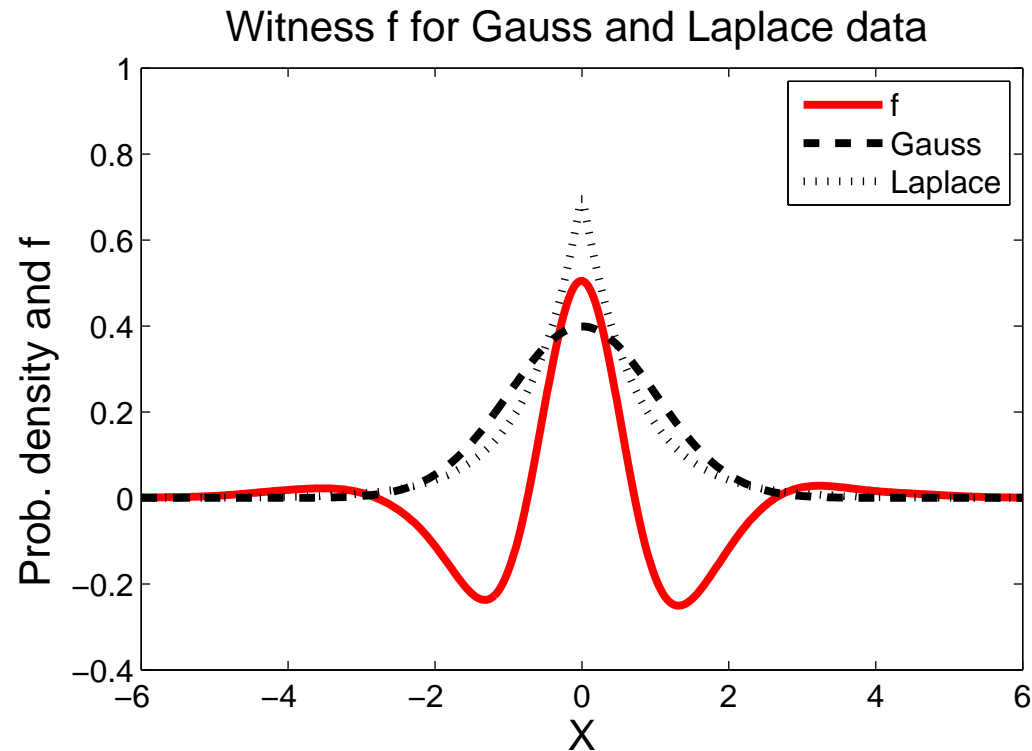
$$\|\mu(X) - \mu(Y)\| = \sup_{\|f\| \leq 1} |\langle \mu(X) - \mu(Y), f \rangle| = \sup_{\|f\| \leq 1} \left| \frac{1}{m} \sum_{i=1}^m f(x_i) - \frac{1}{n} \sum_{i=1}^n f(y_i) \right|.$$

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

## Witness function

---

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



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,

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

Define

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

Note

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

and

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

Recall that in the finite sample case, for strictly p.d. kernels,  $\mu$  was injective — how about now?

[39, 17]

---

**Theorem 3** [15, 13]

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

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

Combine this with

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

Replace  $C(\mathcal{X})$  by the unit ball in an RKHS that is dense in  $C(\mathcal{X})$  — **universal** kernel [41], e.g., Gaussian.

**Theorem 4** [19] *If  $k$  is universal, then*

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

- 
- $\mu$  is invertible on its image

$\mathcal{M} = \{\mu(p) \mid p \text{ is a probability distribution}\}$   
(the “marginal polytope”, [48])

- generalization of the *moment generating function* of a RV  $x$  with distribution  $p$ :

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

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

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

i.e.,

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

(Sriperumbudur et al., 2008)

E.g.,  $\mu$  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,  $\mu$  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).

## Uniform convergence bounds

---

Let  $X$  be an i.i.d.  $m$ -sample from  $p$ . The discrepancy

$$\|\mu(p) - \mu(X)\| = \sup_{\|f\| \leq 1} \left| \mathbf{E}_{x \sim p}[f(x)] - \frac{1}{m} \sum_{i=1}^m f(x_i) \right|$$

can be bounded using uniform convergence methods [40].

## Application 1: Two-sample problem [19]

---

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

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

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 5** 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 6** We assume  $\mathbf{E}(h^2) < \infty$ . 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

$$\sum_{l=1}^{\infty} \lambda_l [q_l^2 - 2], \quad (2)$$

where  $q_l \sim \mathcal{N}(0, 2)$  i.i.d.,  $\lambda_i$  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 [23, 32]:

$x$  and  $y$  independent  $\iff \forall$  bounded continuous functions  $f, g$ ,  
we have  $\text{Cov}(f(x), g(y)) = 0$ .

---

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

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

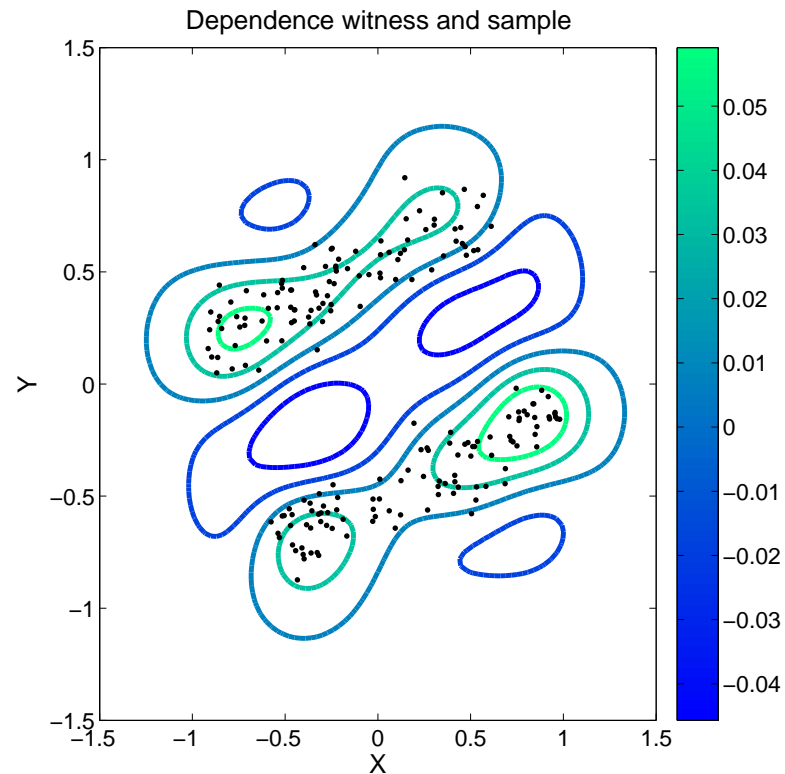
Use  $\Delta := \|\mu(p_{xy}) - \mu(p_x \times p_y)\|$  as a measure of dependence.

For  $k((x,y), (x',y')) = k_x(x,x')k_y(y,y')$ :

$\Delta^2$  equals the Hilbert-Schmidt norm of the covariance operator between the two RKHSs (HSIC), with empirical estimate  $m^{-2} \text{tr} HK_x HK_y$ , where  $H = I - \mathbf{1}/m$  [20, 40].

---

Witness function of the equivalent optimisation problem:



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



# The Representer Theorem

---

**Theorem 7** *Given: a p.d. kernel  $k$  on  $\mathcal{X} \times \mathcal{X}$ , a training set  $(x_1, y_1), \dots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$ , a strictly monotonic increasing real-valued function  $\Omega$  on  $[0, \infty[$ , and an arbitrary cost function  $c : (\mathcal{X} \times \mathbb{R}^2)^m \rightarrow \mathbb{R} \cup \{\infty\}$*

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

$$c((x_1, y_1, f(x_1)), \dots, (x_m, y_m, f(x_m))) + \Omega(\|f\|) \quad (3)$$

*admits a representation of the form*

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

## 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

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

and  $\Omega(\|f\|) = \lambda \|f\|^2$  ( $\lambda > 0$ ): [25]

- generalization to non-quadratic cost functions: [10]
- present form: [34]

## Proof

---

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

$$f = \sum_i \alpha_i k(x_i, \cdot) + f_{\perp},$$

where for all  $j$

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

Application of  $f$  to an arbitrary training point  $x_j$  yields

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

independent of  $f_{\perp}$ .

## Proof: second part of (3)

---

Since  $f_{\perp}$  is orthogonal to  $\sum_i \alpha_i k(x_i, \cdot)$ , and  $\Omega$  is strictly monotonic, we get

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

with equality occurring if and only if  $f_{\perp} = 0$ .

Hence, any minimizer must have  $f_{\perp} = 0$ . Consequently, any solution takes the form

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

## Application: Support Vector Classification

---

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

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

and the regularizer  $\Omega(\|f\|) = \|f\|^2$ .

$\lambda \rightarrow 0$  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 [50] with covariance function  $k$
- minimizer of (3) = MAP estimate

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

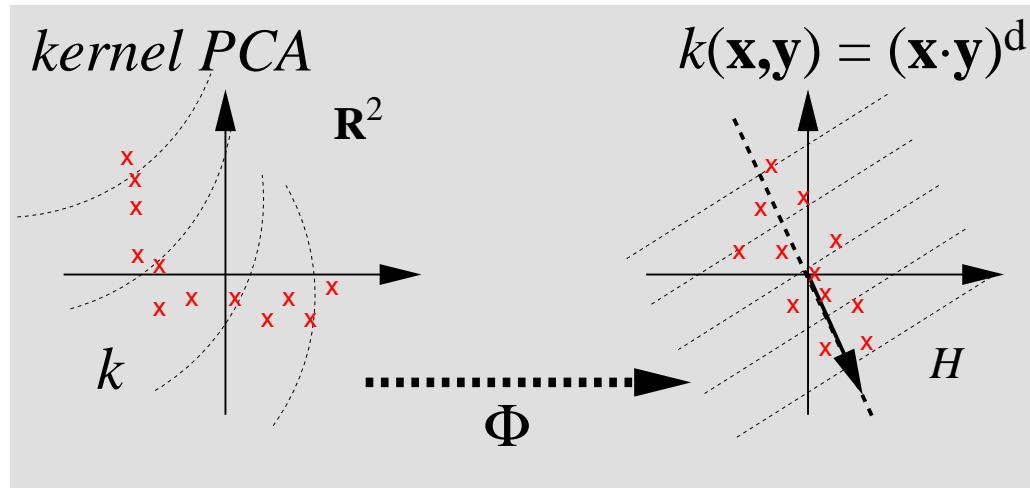
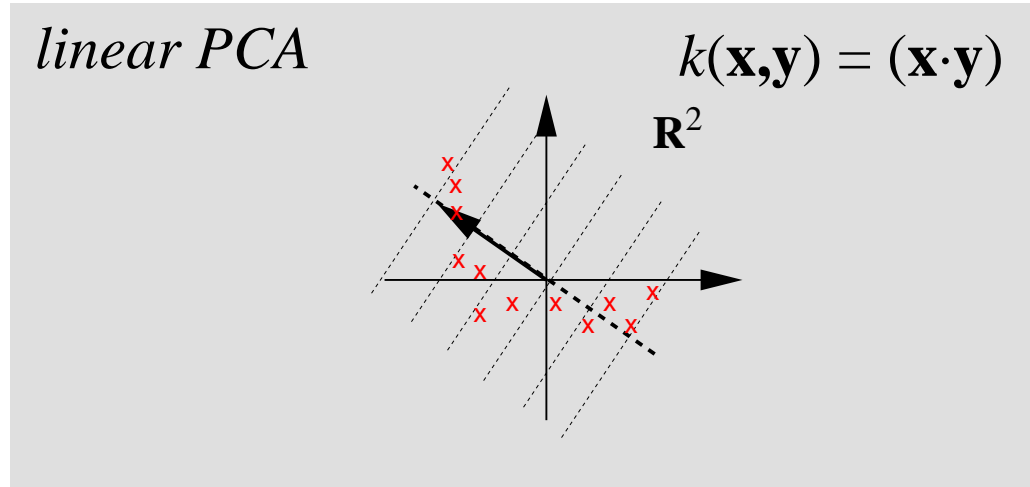
$$c((x_i, y_i, f(x_i))_{i=1,\dots,m}) = \begin{cases} 0 & \text{if } \frac{1}{m} \sum_i \left( f(x_i) - \frac{1}{m} \sum_j f(x_j) \right)^2 = 1 \\ \infty & \text{otherwise} \end{cases}$$

with  $g$  an arbitrary strictly monotonically increasing function.

# 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,...)





## Kernel PCA, II

---

$$x_1, \dots, x_m \in \mathcal{X}, \quad \Phi : \mathcal{X} \rightarrow \mathcal{H}, \quad \mathbf{C} = \frac{1}{m} \sum_{j=1}^m \Phi(x_j) \Phi(x_j)^\top$$

Eigenvalue problem

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

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

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

and the eigenvalue problem can be written as

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

## Kernel PCA in Dual Variables

---

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

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

this leads to

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

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

Solve

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

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

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

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

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

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

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

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

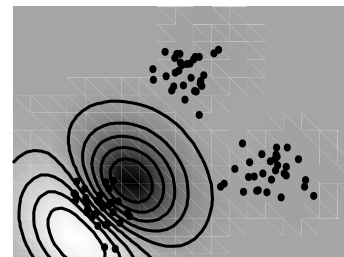
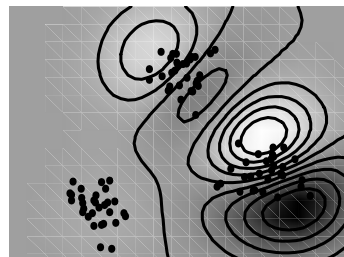
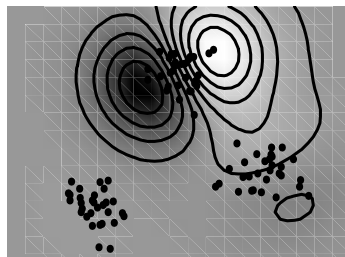
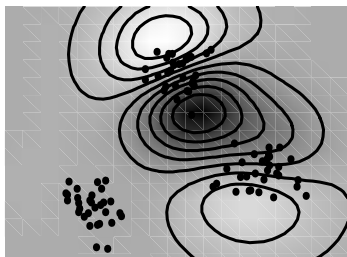
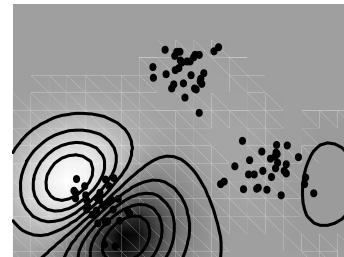
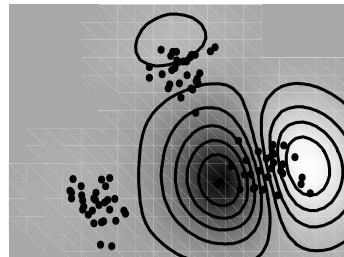
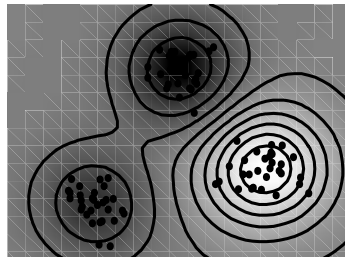
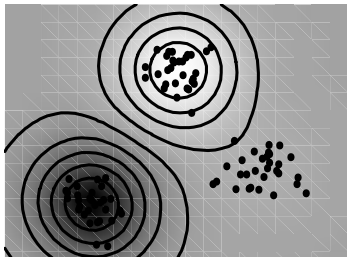
$$\Phi_{KPCA}^w(x) = D^{-1/2}U^\top(k(x_1, x), \dots, k(x_m, x))^\top.$$

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

# Toy Example with Gaussian Kernel

---

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



# Super-Resolution

(Kim, Franz, & Schölkopf, 2004)



a. original image of resolution  $528 \times 396$



b. low resolution image ( $264 \times 198$ ) stretched to the original scale



c. bicubic interpolation



d. supervised example-based learning based on nearest neighbor classifier



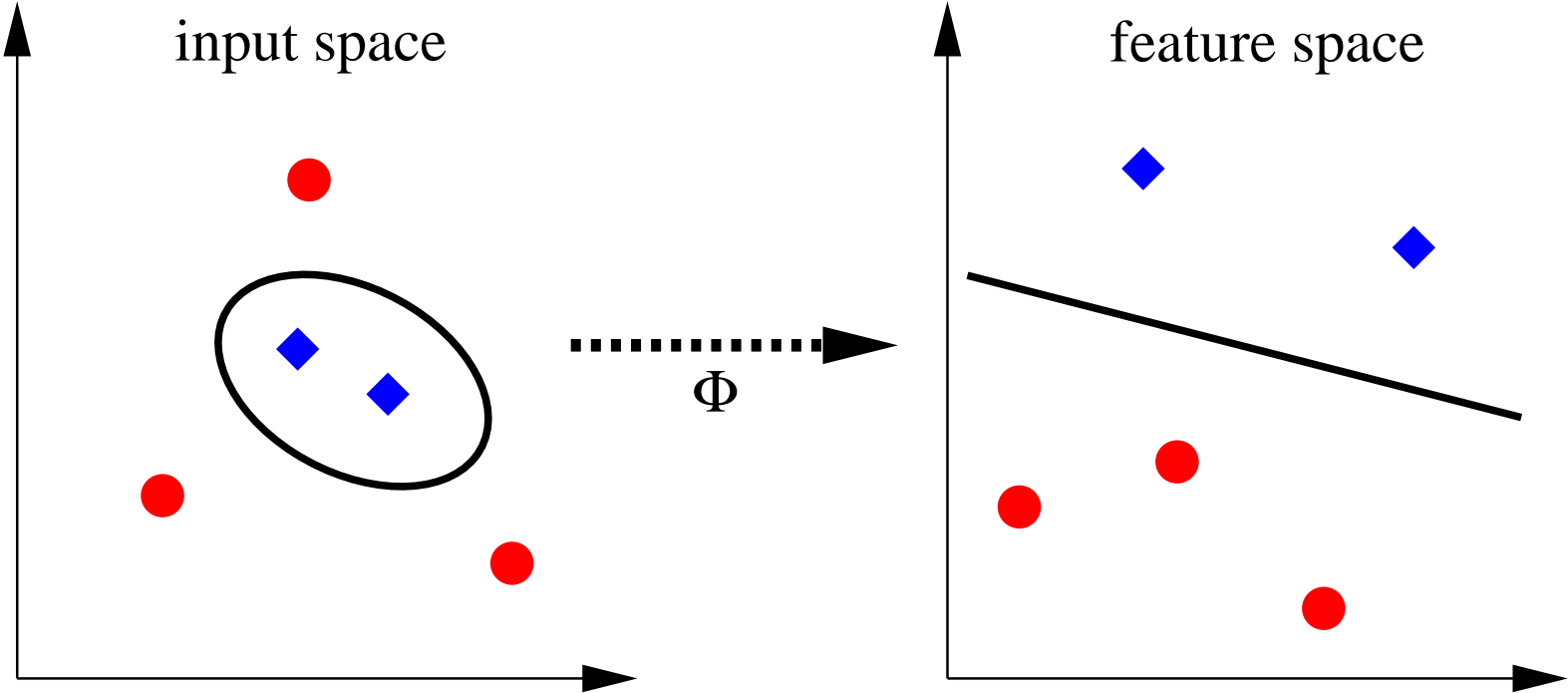
f. unsupervised KPCA reconstruction



g. enlarged portions of a-d, and f (from left to right)

Comparison between different super-resolution methods.

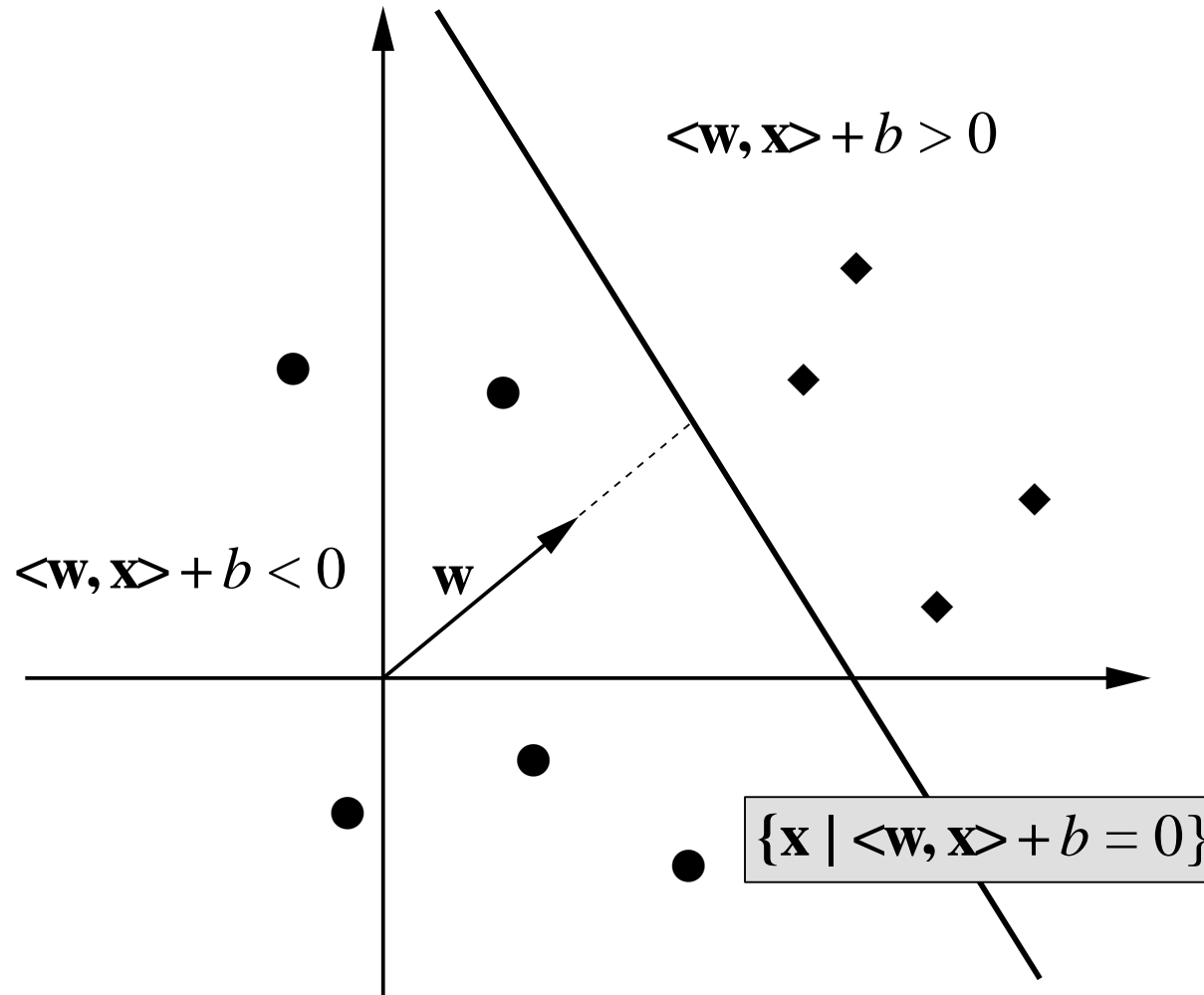
# Support Vector Classifiers



[6]

# Separating Hyperplane

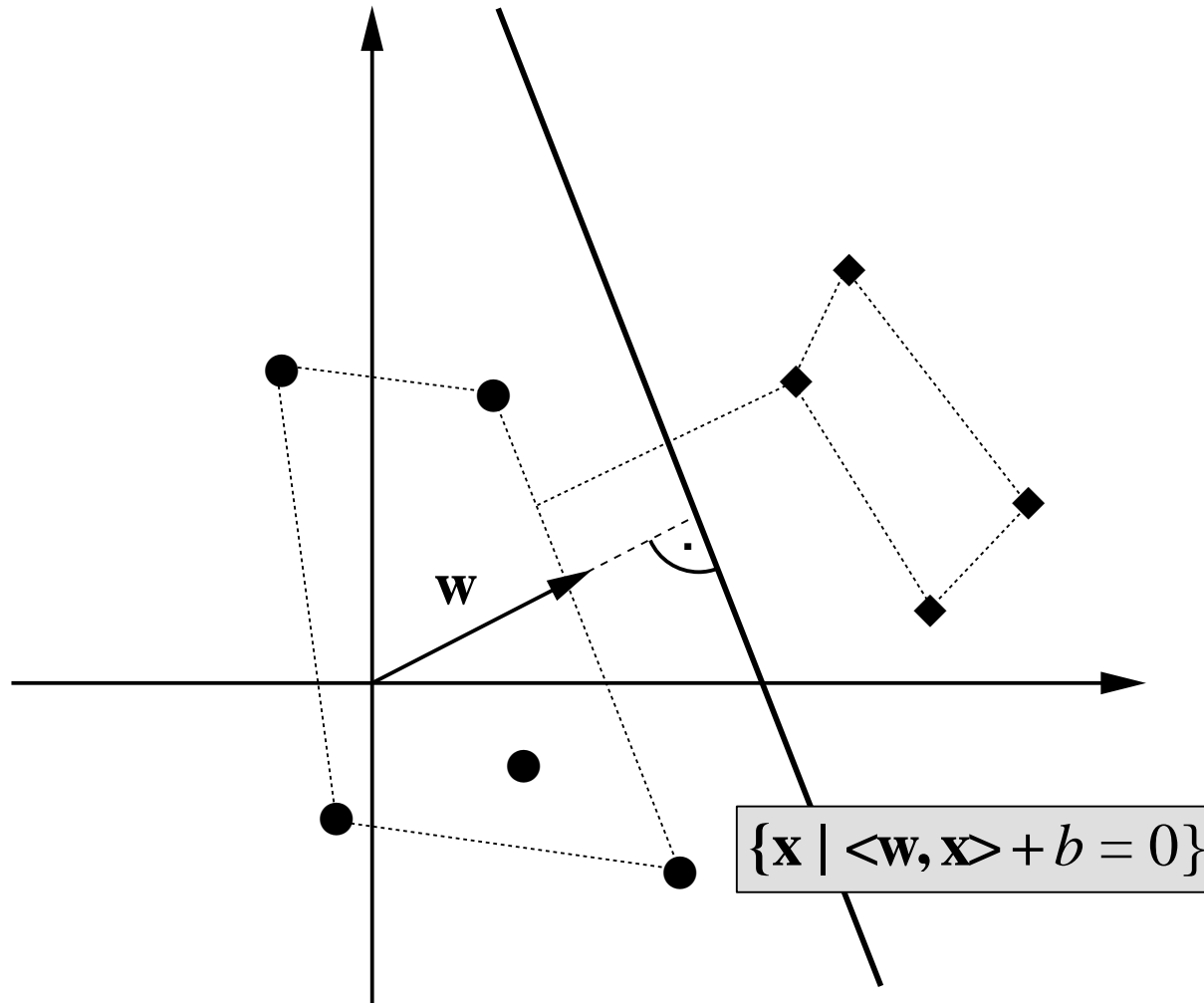
---





# Optimal Separating Hyperplane

[46]



## Eliminating the Scaling Freedom

---

[43]

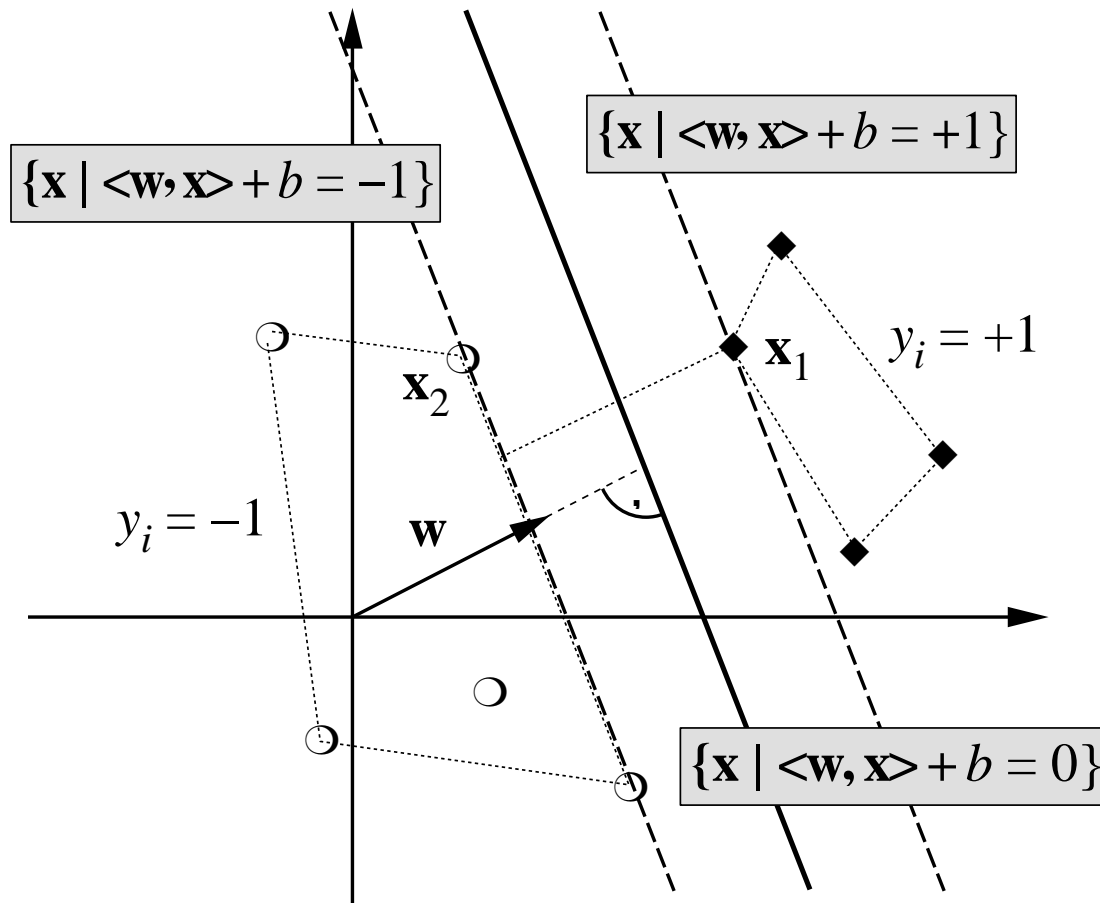
Note: if  $c \neq 0$ , then

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

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

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

# Canonical Optimal Hyperplane



Note:

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

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

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

# 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

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

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

- if a constraint is violated, then  $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 < 0 \longrightarrow$ 
  - $\alpha_i$  will grow to increase  $L$  — how far?
  - $\mathbf{w}$ ,  $b$  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  $\alpha_i$  (*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

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

or

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

The solution is determined by the examples on the margin.

Thus

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

## Why it is Good to Have Few SVs

---

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

**Theorem [45]:** Denote  $\#SV(m)$  the number of SVs obtained by training on  $m$  examples randomly drawn from  $P(\mathbf{x}, y)$ , and  $\mathbf{E}$  the expectation. Then

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

Here,  $\text{Prob}(\text{test error})$  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

---

[8]

Assume that each SV  $\mathbf{x}_i$  exerts a perpendicular force of size  $\alpha_i$  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

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

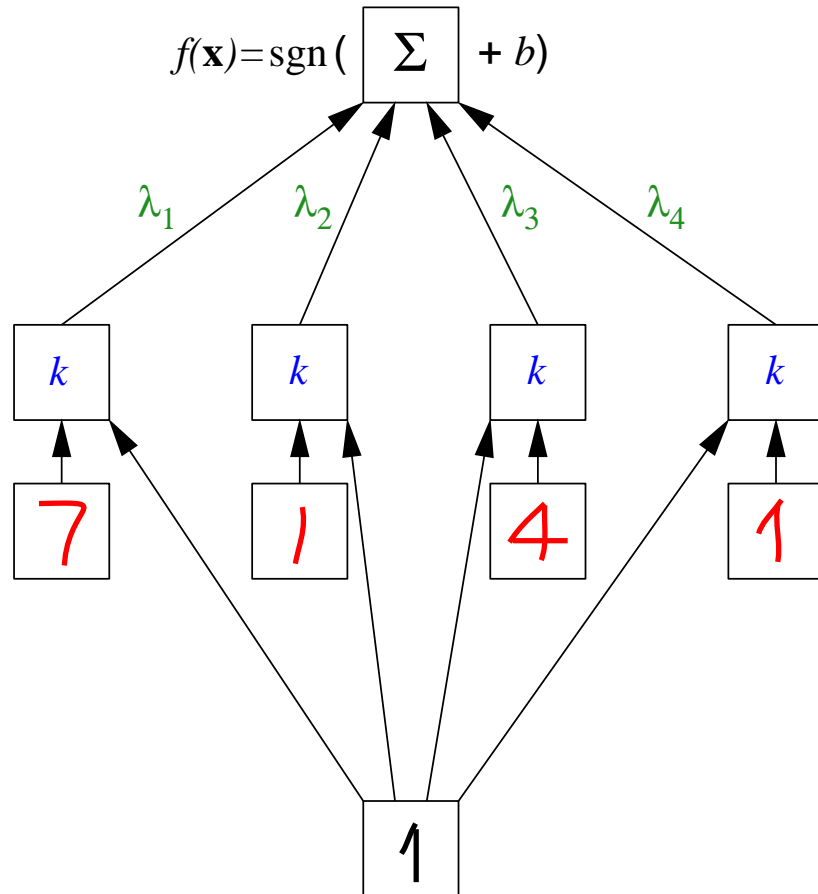
subject to

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

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



classification

weights

comparison:  $k(\mathbf{x}, \mathbf{x}_i)$ , e.g.  $k(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i)^d$

support vectors  
 $\mathbf{x}_1 \dots \mathbf{x}_4$

input vector  $\mathbf{x}$

$$f(\mathbf{x}) = \text{sgn}(\Sigma \lambda_i k(\mathbf{x}, \mathbf{x}_i) + b)$$

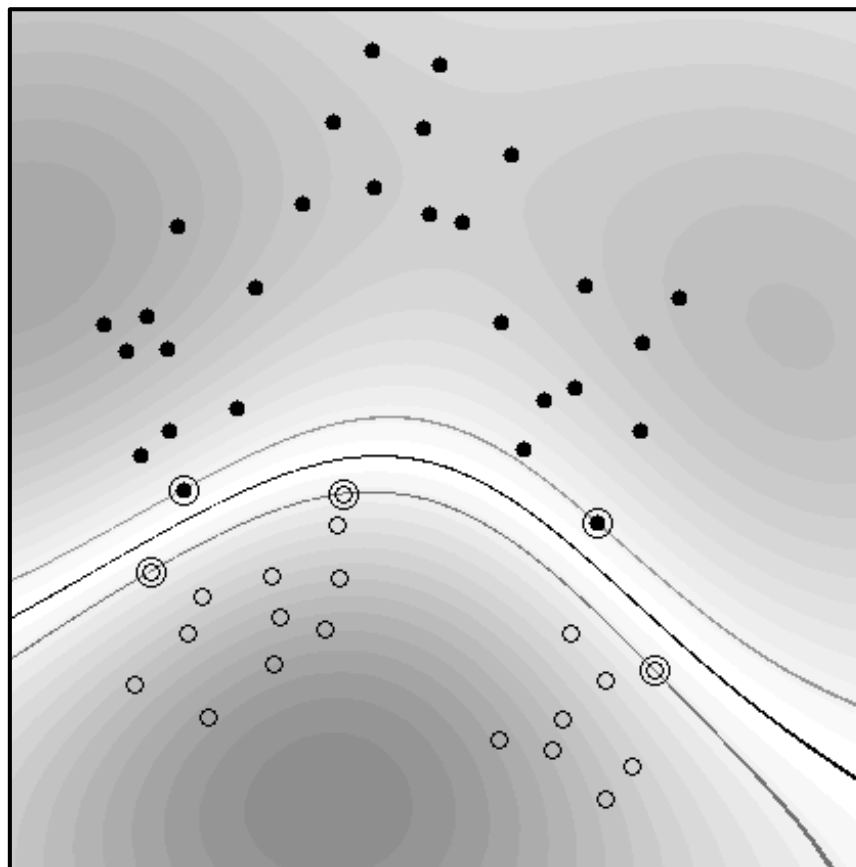
$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2 / c)$$

$$k(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa(\mathbf{x} \cdot \mathbf{x}_i) + \theta)$$

## 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  $\alpha_i \rightarrow \infty$ .

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

$\nu$ -SVM [36]: 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  $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i$ ,  $\xi_i \geq 0$  (margin  $\rho/\|\mathbf{w}\|$ )

## The $\nu$ -Property

---

SVs:  $\alpha_i > 0$

“margin errors:”  $\xi_i > 0$

KKT-Conditions  $\implies$

- 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

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

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

$\nu$ -SVM dual: maximize

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

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

In both cases: *decision function*:

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



## Connection between $\nu$ -SVC and $C$ -SVC

---

**Proposition.** If  $\nu$ -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  $\rho$ , 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  $\rho^2$ , with the  $C$ -SV target with  $C = 1/\rho$ .

## SVM Training

---

- naive approach: the complexity of maximizing

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

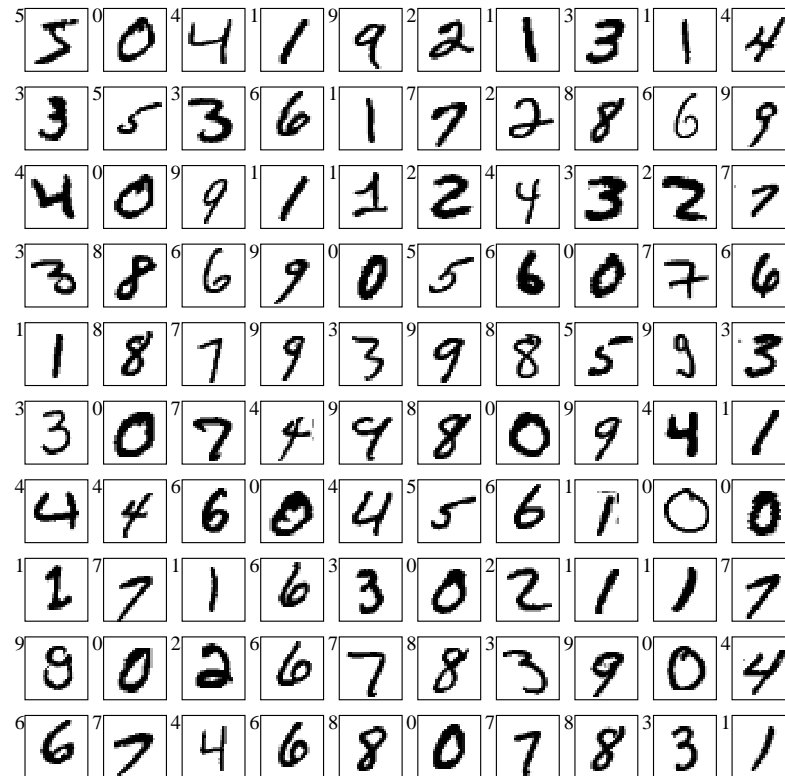
scales with the third power of the training set size  $m$

- 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 [42].
- in fact, one can use chunks which do not even contain all SVs [29]. 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 [30].
- <http://www.kernel-machines.org/software.html>

# MNIST Benchmark

---

handwritten character benchmark (60000 training & 10000 test examples,  $28 \times 28$ )



## 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%	[37]
LeNet4	1.1%	[26]
Boosted LeNet4	0.7%	[26]
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] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of mathematics. Springer, New York, 1996.
- [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, Cambridge, MA, USA, 09 2008. MIT Press.
- [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] I. A. Ibragimov and R. Z. Has'minskii. *Statistical Estimation — Asymptotic Theory*. Springer-Verlag, New York, 1981.
- [23] J. Jacod and P. Protter. *Probability Essentials*. Springer, New York, 2000.
- [24] 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.

- [25] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.
- [26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.
- [27] 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.
- [28] 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.
- [29] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab., 1996.
- [30] 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.
- [31] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.
- [32] A. Rényi. On measures of dependence. *Acta Math. Acad. Sci. Hungar.*, 10:441–451, 1959.
- [33] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.
- [34] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [35] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [36] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.
- [37] 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.
- [38] A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

- [39] 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, 10 2007. Springer.
- [40] 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.
- [41] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, 2:67–93, 2001.
- [42] V. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).
- [43] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, NY, 1995.
- [44] V. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.
- [45] 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).
- [46] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.
- [47] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.
- [48] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Department of Statistics, September 2003.
- [49] H. L. Weinert. *Reproducing Kernel Hilbert Spaces*. Hutchinson Ross, Stroudsburg, PA, 1982.
- [50] 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.
- [51] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.



# 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.

---


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

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