Kernel Methods

Bernhard Schölkopf

Max-Planck-Institut für biologische Kybernetik

Roadmap

- Similarity, kernels, feature spaces
- Positive definite kernels and their RKHS
- Kernel means, representer theorem
- Support Vector Machines

Learning and Similarity: some Informal Thoughts

- input/output sets $\mathcal{X}, \mathcal{Y}$
- training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathcal{Y}$
- "generalization": given a previously unseen $x \in \mathcal{X}$, find a suitable $y \in \mathcal{Y}$
- (x, y) should be "similar" to $(x_1, y_1), \ldots, (x_m, y_m)$
- how to measure similarity?
  - for outputs: *loss function* (e.g., for $\mathcal{Y} = \{\pm 1\}$, zero-one loss)
  - for inputs: *kernel*

Similarity of Inputs

- symmetric function

$$\begin{aligned} k : \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \\ (x, x') &\mapsto k(x, x') \end{aligned}$$

- for example, if $\mathcal{X} = \mathbb{R}^N$: canonical dot product

$$k(x, x') = \sum_{i=1}^{N} [x]_i [x']_i$$

- if $\mathcal{X}$ is not a dot product space: assume that k has a representation as a dot product in a linear space $\mathcal{H}$, i.e., there exists a map $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ such that

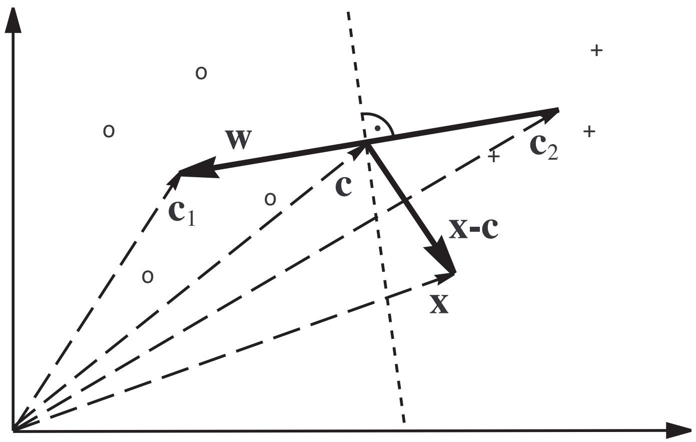
$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle .$$

- in that case, we can think of the patterns as $\Phi(x), \Phi(x')$, and carry out geometric algorithms in the dot product space ("feature space") $\mathcal{H}$.

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. [32]

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

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

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.

- Demo
- Exercise: derive the Parzen windows classifier by computing the distance criterion directly

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*.

[39, 40]

Kernels and Feature Spaces

Preprocess the data with

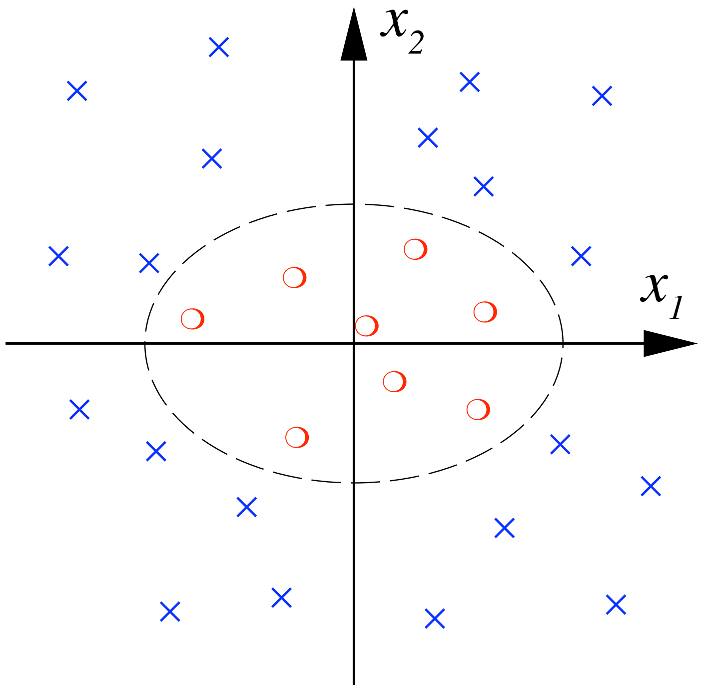
$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \Phi(x), \end{aligned}$$

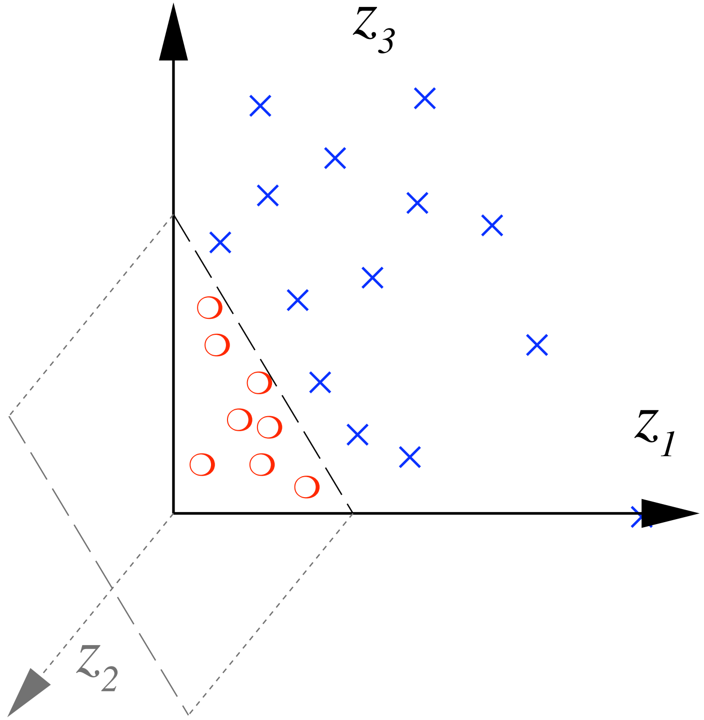
where $\mathcal{H}$ is a dot product space, and learn the mapping from $\Phi(x)$ to y [6].

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

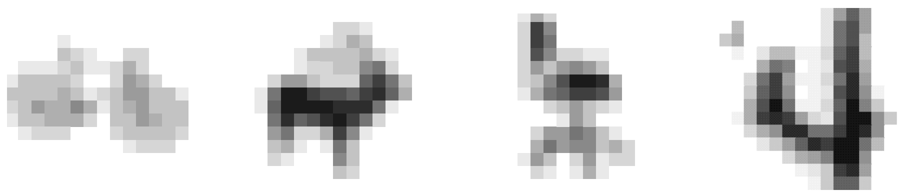
Example: All Degree 2 Monomials

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





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, $N = d = 2$

$$\begin{aligned}\left\langle \Phi(x), \Phi(x') \right\rangle &= (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)(x'^2_1, \sqrt{2}\, x'_1 x'_2, x'^2_2)^\top \\ &= \left\langle x, x' \right\rangle^2 \\ &=: k(x, x')\end{aligned}$$

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

The Kernel Trick, II

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

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

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 $\lambda_i \geq 0$ [26].

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:

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

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

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:

$$\begin{aligned} \text{Polynomial} \quad k(x, x') &= (\langle x, x' \rangle + c)^d \\ \text{Sigmoid} \quad k(x, x') &= \tanh(\kappa \langle x, x' \rangle + \Theta) \\ \text{Gaussian} \quad k(x, x') &= \exp(-\|x - x'\|^2/(2\,\sigma^2)) \end{aligned}$$

- Kernels are also known as covariance functions [44, 41, 45, 25]

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 $x_1, \ldots, x_m \in \mathcal{X}$ and
- any $a_1, \ldots, a_m \in \mathbb{R}$

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, $\sum_{i,j} a_i a_j K_{ij} = 0 \Longrightarrow a = 0$, call it strictly positive definite.

Elementary Properties of PD Kernels

Kernels from Feature Maps.
If Φ 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)

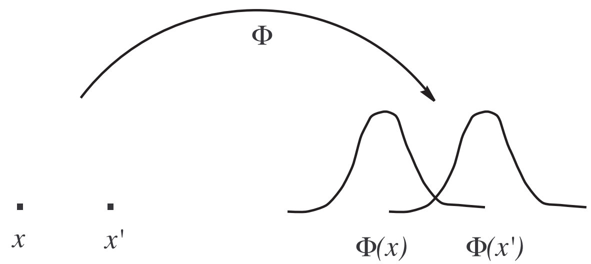
Vanishing Diagonals.
$k(x, x) = 0$ for all $x \in \mathcal{X} \Longrightarrow k(x, x') = 0$ for all $x, x' \in \mathcal{X}$

The Feature Space for PD Kernels [5, 2, 29]

- define a feature map

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

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

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

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

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

Explicit Construction of the RKHS Map for Mercer Kernels

Recall that the dot product has to satisfy

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

For a Mercer kernel

$$k(x,x') = \sum_{j=1}^{N_F} \lambda_j \psi_j(x)\psi_j(x')$$

(with $\lambda_i > 0$ for all i, $N_F \in \mathbb{N} \cup \{\infty\}$, and $\left\langle \psi_i, \psi_j \right\rangle_{L_2(\mathcal{X})} = \delta_{ij}$), this can be achieved by choosing $\langle .,. \rangle$ such that

$$\langle \psi_i, \psi_j \rangle = \delta_{ij}/\lambda_i.$$

ctd.

To see this, compute

$$
\begin{aligned}
\langle k(x,.), k(x',.)\rangle &= \left\langle \sum_i \lambda_i \psi_i(x)\psi_i, \sum_j \lambda_j \psi_j(x')\psi_j \right\rangle \\
&= \sum_{i,j} \lambda_i \lambda_j \psi_i(x)\psi_j(x')\langle \psi_i, \psi_j\rangle \\
&= \sum_{i,j} \lambda_i \lambda_j \psi_i(x)\psi_j(x')\delta_{ij}/\lambda_i \\
&= \sum_i \lambda_i \psi_i(x)\psi_i(x') \\
&= k(x, x').
\end{aligned}
$$

Deriving the Kernel from the RKHS

An RKHS is a Hilbert space $\mathcal{H}$ of functions f where all *point evaluation functionals*

$$\begin{aligned} p_x \colon \mathcal{H} &\longrightarrow \mathbb{R} \\ f &\mapsto p_x(f) = f(x) \end{aligned}$$

exist and are continuous.

Continuity means that whenever f and f' are close in $\mathcal{H}$, then $f(x)$ and $f'(x)$ are close in $\mathbb{R}$. This can be thought of as a topological prerequisite for generalization ability.

By Riesz' representation theorem, there exists an element of $\mathcal{H}$, call it r_x, such that

$$\langle r_x, f \rangle = f(x),$$

in particular,

$$\langle r_x, r_{x'} \rangle = r_{x'}(x).$$

Define $k(x, x') := r_x(x') = r_{x'}(x)$.

(cf. Canu & Mary, 2002)

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

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

ctd.

- $\Phi_m(x_1), \ldots, \Phi_m(x_m)$ contain *all* necessary information about $\Phi(x_1), \ldots, \Phi(x_m)$
- 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

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

- 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 $i, j = 1, \ldots, m$.

Some Properties of Kernels [32, 34]

If $k_1, k_2, \ldots$ are pd kernels, then so are

- αk_1, provided $\alpha \geq 0$
- $k_1 + k_2$
- $k_1 \cdot k_2$
- $k(x, x') := \lim_{n\to\infty} k_n(x, x')$, provided it exists
- $k(A, B) := \sum_{x\in A, x'\in B} k_1(x, x')$, where A, B are finite subsets of $\mathcal{X}$
  (using the feature map $\tilde{\Phi}(A) := \sum_{x\in A} \Phi(x)$)

Further operations to construct kernels from kernels: tensor products, direct sums, convolutions [19].

Properties of Kernel Matrices, I [30]

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

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

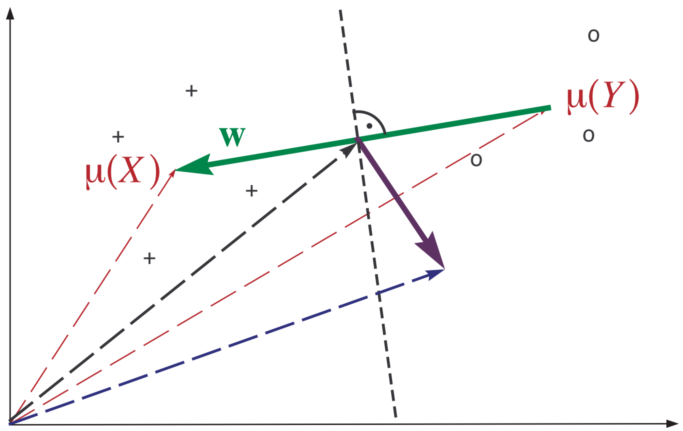
Properties, II: Functional Calculus [33]

- 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 $\sigma(K)$)

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.

∎

Exercise: generalize to the case of nonsingular kernel (i.e., leading to nonsingular Gram matrices for pairwise distinct points).

Generalization

We will prove a more general statement, without assuming positive definiteness.

Definition 2 *We call a kernel* $k : \mathcal{X}^2 \to \mathbb{R}$ nonsingular *if for any* $n \in \mathbb{N}$ *and pairwise distinct* $x_1, \ldots, x_n \in \mathcal{X}$*, the Gram matrix* $(k(x_i, x_j))_{ij}$ *is nonsingular.*

Note that strictly positive definite kernels are nonsingular: if the matrix K is singular, then there exists a $\beta \neq 0$ such that $K\beta = 0$, hence $\beta^\top K \beta = 0$, hence k is not strictly positive definite.

Proposition 3 *Assume* X, Y *are defined as above,* k *is nonsingular, 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*

$$\sum_{i=1}^{m} \alpha_i k(x_i, .) = \sum_{j=1}^{n} \beta_j k(y_j, .), \tag{2}$$

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 (2), 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 $z_2, \dots \in X \cup Y - \{x_1\}, \gamma_2, \dots \in \mathbb{R}$.
Similar to the pd case, k induces a linear space with a bilinear form satisfying the reproducing kernel property.
Take the bilinear form between $\sum_j \lambda_j k(z_j, .)$ and the above, to get

$$0 = \sum_{ij} \lambda_j \gamma_i k(z_j, z_i) = \lambda^\top K \gamma,$$

where $\lambda \in \mathbb{R}$ is arbitrary. Hence $K\gamma = 0$. However, $\gamma \neq 0$, hence K is singular.

Since the z_i are pairwise distinct, k cannot be nonsingular. ∎

The mean map

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

satisfies

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

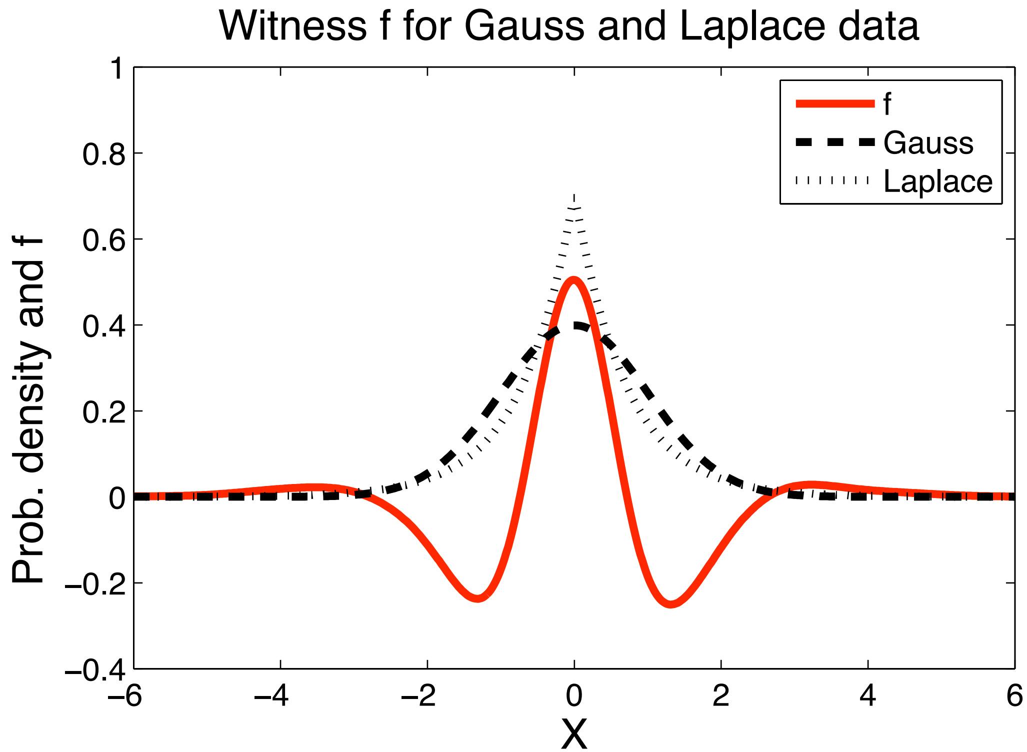
and

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

Note: distance in the RKHS = solution of a high-dimensional optimization problem.

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?

Theorem 4 *[13, 10]*

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

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

Replace $C(\mathcal{X})$ by the unit ball in an RKHS that is dense in $C(\mathcal{X})$ — universal kernel [38], e.g., Gaussian.

Theorem 5 *[16] 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", [42])
- generalization of the *moment generating function* of a RV x with distribution p:

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

Uniform convergence bounds

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

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

can be bounded using uniform convergence methods [37].

Application 1: Two-sample problem [16]

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

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

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

Theorem 6 *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 7 *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*

$$\sum_{l=1}^{\infty} \lambda_l \left[q_l^2 - 2\right], \tag{3}$$

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.
[3, 14]: kernel generalized variance

[17, 18]: kernel constrained covariance, HSIC

Main idea [22, 28]:
x and y independent $\Longleftrightarrow$ $\forall$ bounded continuous functions f, g,
we have $\mathrm{Cov}(f(x), g(y)) = 0$.

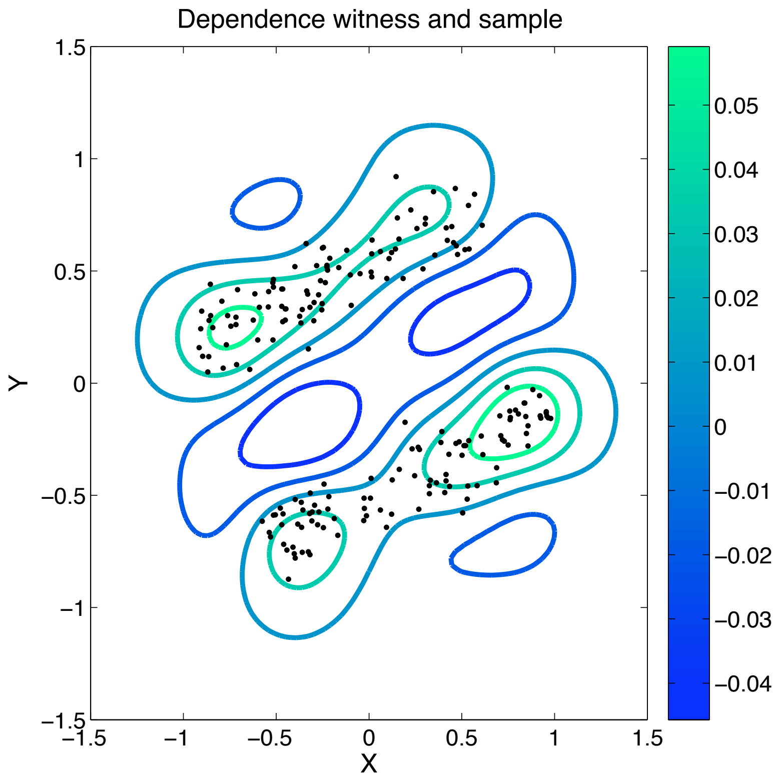
k kernel on $\mathcal{X} \times \mathcal{Y}$.

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

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 $m^{-2}\,\mathrm{tr}\,HK_xHK_y$, where $H = I - \mathbf{1}/m$ [17, 37].

Witness function of the equivalent optimisation problem:



Application: learning causal structures *(Sun, Janzing, Schölkopf, Fukumizu, ICML 2007; Fukumizu, Gretton, Sun, Schölkopf, 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 $X' = \left\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\right\}$ from $p' \neq p$.

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

[35]: reweight training set

Minimize

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

Equivalent QP:

$$\begin{gathered} \underset{\beta}{\text{minimize}} \ \frac{1}{2} \beta^\top \left(K + \lambda \mathbf{1} \right) \beta - \beta^\top l \\ \text{subject to } \beta_i \geq 0 \text{ and } \sum_i \beta_i = 1, \end{gathered}$$

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

Experiments show that in underspecified situations (e.g., large kernel widths), this helps [21].

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

Application 4: Measure estimation and dataset squashing [9, 4, 1, 37]

Given a sample X, minimize

$$\|\mu(X) - \mu(p)\|^2$$

over a convex combination of measures p_i,

$$p = \sum_i \alpha_i p_i, \quad \alpha_i \geq 0, \quad \sum_i \alpha_i = 1.$$

This can be written as a convex QP with objective function

$$\|\mu(X) - \mu(p)\|^2 = \alpha^\top Q\alpha + 1_m^\top K 1_m - 2\alpha^\top L 1_m,$$

where

$$\begin{aligned} L_{ij} &:= \mathbf{E}_{x\sim p_i}\left[k(x, x_j)\right] \\ Q_{ij} &:= \mathbf{E}_{x\sim p_i, x'\sim p_j}\left[k(x, x')\right] \\ K_{ij} &= k(x_i, x_j) \\ 1_m &:= (1/m, \ldots, 1/m)^\top \in \mathbb{R}^m. \end{aligned}$$

In practice, use

$$\alpha^\top [Q + \lambda I]\alpha - 2\alpha^\top L 1_m$$

Some cases where Q and L can be computed in closed form [37]:

- Gaussian p_i and k (cf. [4, 43])
- X training set, Dirac measures $p_i = \delta_{x_i}$: dataset squashing, [11]
- X test set, Dirac measures $p_i = \delta_{y_i}$ centered on the training points Y: covariate shift correction [20]

The Representer Theorem

Theorem 8 *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 $f \in \mathcal{H}$ minimizing the regularized risk functional

$$c\left((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))\right) + \Omega\left(\|f\|\right) \tag{4}$$

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 $\Omega(\|f\|) = \lambda\|f\|^2$ ($\lambda > 0$): [24]
- generalization to non-quadratic cost functions: [8]
- present form: [32]

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 (4)

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

$$\begin{aligned} \Omega(\|f\|) &= \Omega\left(\|\sum\nolimits_i \alpha_i k(x_i, .) + f_\perp\|\right) \\ &= \Omega\left(\sqrt{\|\sum\nolimits_i \alpha_i k(x_i, .)\|^2 + \|f_\perp\|^2}\right) \\ &\geq \Omega\left(\|\sum\nolimits_i \alpha_i k(x_i, .)\|\right), \end{aligned} \tag{5}$$

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

$$f = \sum\nolimits_i \alpha_i k(x_i, .).$$

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

Further Applications

Bayesian MAP Estimates. Identify (4) 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 [45] with covariance function k
- minimizer of (4) = 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.

The Pre-Image Problem

- due to the representer theorem, the solution of kernel algorithms usually corresponds to a single vector in $\mathcal{H}$

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

However, there is usually no $x \in \mathcal{X}$ such that

$$\Phi(x) = \mathbf{w},$$

i.e., $\Phi(\mathcal{X})$ is not closed under linear combinations — it is a nonlinear manifold (cf. [7, 31]).

Conclusion so far

- 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,...)

References

[1] Y. Altun and A.J. Smola. Unifying divergence minimization and statistical inference via convex duality. In H.U. Simon and G. Lugosi, editors, *Proc. Annual Conf. Computational Learning Theory*, LNCS, pages 139–153. Springer, 2006.

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

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

[4] N. Balakrishnan and D. Schonfeld. A maximum entropy kernel density estimator with applications to function interpolation and texture segmentation. In *SPIE Proceedings of Electronic Imaging: Science and Technology. Conference on Computational Imaging IV*, San Jose, CA, 2006.

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

[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] C. J. C. Burges. Geometry and invariance in kernel based methods. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 89–116, Cambridge, MA, 1999. MIT Press.

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

[9] M. Dudík, S. Phillips, and R.E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In *Proc. Annual Conf. Computational Learning Theory*. Springer Verlag, 2004.

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

[11] W. DuMouchel, C. Volinsky, C. Cortes, D. Pregibon, and T. Johnson. Squashing flat files flatter. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 1999.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

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

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

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

[21] 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.

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

[23] 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.

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

[25] 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.

[26] 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.

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

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

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

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

[31] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

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

[33] 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.

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

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

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

[37] 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.

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

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

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

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

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

[43] C. Walder, K. Kim, and B. Schölkopf. Sparse multiscale gaussian process regression. Technical Report 162, Max-Planck-Institut für biologische Kybernetik, 2007.

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

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