Introduction to Kernel Methods

Bernhard Schölkopf
Max-Planck-Institut für biologische Kybernetik
72076 Tübingen, Germany
bernhard.schoelkopf@tuebingen.mpg.de
Roadmap

1. Kernels
2. Support Vector classification
3. Further kernel algorithms
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
  \[ k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \]
  \[ (x, x') \mapsto k(x, x') \]

- 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.

\[
\mathbf{c}_+ := \frac{1}{m_1} \sum_{y_i=1} \Phi(x_i), \quad \mathbf{c}_- := \frac{1}{m_2} \sum_{y_i=-1} \Phi(x_i)
\]

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. [44]

\[
f(x) = \text{sgn} \left( \frac{1}{m_1} \sum_{i:y_i=+1} \langle \Phi(x), \Phi(x_i) \rangle - \frac{1}{m_2} \sum_{i:y_i=-1} \langle \Phi(x), \Phi(x_i) \rangle + b \right)
\]

\[
= \text{sgn} \left( \frac{1}{m_1} \sum_{i:y_i=+1} k(x, x_i) - \frac{1}{m_2} \sum_{i:y_i=-1} k(x, x_i) + b \right)
\]

where

\[
b = \frac{1}{2} \left( \frac{1}{m_2^2} \sum_{(i,j):y_i=y_j=-1} k(x_i, x_j) - \frac{1}{m_1^2} \sum_{(i,j):y_i=y_j=+1} k(x_i, x_j) \right).
\]

- provides a geometric interpretation of Parzen windows
- the decision function is a hyperplane
An Example of a Kernel Algorithm, ctd.

- Demo
- Exercise: derive the Parzen windows classifier by computing the distance criterion directly
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, $N = d = 2$

$$
\langle \Phi(x), \Phi(x') \rangle = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(x'_1^2, \sqrt{2} x'_1 x'_2, x'_2^2)^\top \\
= \langle x, x' \rangle^2 \\
= : k(x, x')
$$

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

B. Schölkopf, Berder, September 2004
The Kernel Trick, II

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

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

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

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 $\lambda_i \geq 0$ [34].
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:
  
  \[ k(x, x') = (\langle x, x' \rangle + c)^d \]
  \[ \text{Sigmoid} \quad k(x, x') = \tanh(\kappa \langle x, x' \rangle + \Theta) \]
  \[ \text{Gaussian} \quad k(x, x') = \exp(-\|x - x'\|^2/(2 \sigma^2)) \]

- Kernels are studied also in the Gaussian Process prediction community (covariance functions) [61, 58, 63, 33]
Positive Definite Kernels

It can be shown that (modulo some details) 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.
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 \( x, x' \in \mathcal{X} \)

B. Schölkopf, Berder, September 2004
The Feature Space for PD Kernels

- define a feature map

\[ \Phi : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{X}} \]

\[ x \mapsto k(., x). \]

E.g., for the Gaussian kernel:

Next steps:
- turn \( \Phi(\mathcal{X}) \) into a linear space
- endow it with a dot product satisfying

\[ \langle k(., x_i), k(., x_j) \rangle = k(x_i, x_j) \]
- complete the space to get a reproducing kernel Hilbert space
Turn it Into a Linear Space

Form linear combinations

\[
\begin{align*}
  f(.) &= \sum_{i=1}^{m} \alpha_i k(., x_i), \\
  g(.) &= \sum_{j=1}^{m'} \beta_j k(., x'_j)
\end{align*}
\]

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

B. Schölkopf, Berder, September 2004
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).
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

B. Schölkopf, Berder, September 2004
Endow it With a Dot Product, II

- It can be shown that $\langle ., . \rangle$ is a p.d. kernel on the set of functions
  \[ \{ f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) | \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \} : \]
  \[
  \sum_{ij} \gamma_i \gamma_j \langle f_i, f_j \rangle = \left\langle \sum_i \gamma_i f_i, \sum_j \gamma_j f_j \right\rangle =: \langle f, f \rangle \\
  = \left\langle \sum_i \alpha_i k(\cdot, x_i), \sum_i \alpha_i k(\cdot, x_i) \right\rangle = \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \geq 0
  \]

- furthermore, it is strictly positive definite:
  \[ f(x)^2 = \langle f, k(\cdot, x) \rangle^2 \leq \langle f, f \rangle \langle k(\cdot, x), k(\cdot, x) \rangle = \langle f, f \rangle k(x, x) \]
  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 \( \langle \psi_i, \psi_j \rangle_{L_2(\mathcal{X})} = \delta_{ij} \)),

this can be achieved by choosing \( \langle ., . \rangle \) such that

\[ \langle \psi_i, \psi_j \rangle = \delta_{ij} / \lambda_i. \]
To see this, compute
\[
\langle k(x, .), k(x', .) \rangle = \left\langle \sum_i \lambda_i \psi_i(x) \psi_i, \sum_j \lambda_j \psi_j(x') \psi_j \right\rangle \\
= \sum_{i,j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \langle \psi_i, \psi_j \rangle \\
= \sum_{i,j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \delta_{i,j} / \lambda_i \\
= \sum_i \lambda_i \psi_i(x) \psi_i(x') \\
= k(x, x').
\]
Deriving the Kernel from the RKHS

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

$$p_x: \mathcal{H} \rightarrow \mathbb{R}$$

$$f \mapsto p_x(f) = f(x)$$

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

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{X}}$$

$$x \mapsto k(., x).$$

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

Consider

$$\Phi_m : \mathcal{X} \rightarrow \mathbb{R}^m$$

$$x \mapsto k(., x)|_{(x_1, \ldots, x_m)} = (k(x_1, x), \ldots, k(x_m, x))^\top$$
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 $G_{ij} := \langle \Phi_m(x_i), \Phi_m(x_j) \rangle$ satisfies $G = K^2$
  where $K_{ij} = k(x_i, x_j)$

• modify $\Phi_m$ to

\[
\Phi_m^w : X \rightarrow \mathbb{R}^m \\
x \mapsto K^{-\frac{1}{2}}(k(x_1, x), \ldots, k(x_m, x))^\top
\]

• this “whitened” map (“kernel PCA map”) satisfies

\[
\langle \Phi_m^w(x_i), \Phi_m^w(x_j) \rangle = k(x_i, x_j)
\]

for all $i, j = 1, \ldots, m$. 
Some Properties of Kernels [44]

If \( k_1, k_2, \ldots \) are pd kernels, then so are

- \( \alpha k_1 \), 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 [23].
Properties of Kernel Matrices, I [41]

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 $K = S D S^\top$, with an orthogonal matrix $S$ and a diagonal matrix $D$ with nonnegative entries. Then
\[ K_{ij} = (S D S^\top)_{ij} = \langle S_i, D S_j \rangle = \langle \sqrt{D} S_i, \sqrt{D} S_j \rangle, \]
where the $S_i$ are the rows of $S$.

We have thus constructed a map $\Phi$ into an $m$-dimensional feature space $\mathcal{H}$ such that
\[ K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle. \]

B. Schölkopf, Berder, September 2004
Properties, II: Functional Calculus [47]

- $K$ symmetric $m \times m$ matrix with eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$
- $f$ a continuous function on $[\lambda_{\min}, \lambda_{\max}]$
- Then there is a symmetric matrix $f(K)$ with eigenvalues in $f([\lambda_{\min}, \lambda_{\max}])$.
- 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 $\mathbb{R}$

$$
(\alpha f + g)(K) = \alpha f(K) + g(K)
$$

$$
(fg)(K) = f(K)g(K) = g(K)f(K)
$$

$$
\|f\|_{\infty, \sigma(K)} = \|f(K)\|
$$

$$
\sigma(f(K)) = f(\sigma(K))
$$

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

input space

feature space

\[ \Phi \]

B. Schölkopf, Berder, September 2004
Separating Hyperplane

\[ \langle \mathbf{w}, \mathbf{x} \rangle + b > 0 \]

\[ \langle \mathbf{w}, \mathbf{x} \rangle + b < 0 \]

\[ \{ \mathbf{x} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \} \]
Optimal Separating Hyperplane

\{x \mid \langle w, x \rangle + b = 0 \}

B. Schölkopf, Berder, September 2004
Eliminating the Scaling Freedom

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. $X^* = \{ \mathbf{x}_1, \ldots, \mathbf{x}_r \}$ if $\min_{\mathbf{x}_i \in X} | \langle \mathbf{w}, \mathbf{x}_i \rangle + b | = 1$. 
Canonical Optimal Hyperplane

\{x \mid \langle w, x \rangle + b = -1 \}

\{x \mid \langle w, x \rangle + b = +1 \}

Note:
\[ \langle w, x_1 \rangle + b = +1 \]
\[ \langle w, x_2 \rangle + b = -1 \]

\[ \Rightarrow \quad \langle w, (x_1 - x_2) \rangle = 2 \]
\[ \Rightarrow \quad \langle \frac{w}{||w||}, (x_1 - x_2) \rangle = \frac{2}{||w||} \]
Pattern Noise as Maximum Margin Regularization
Maximum Margin vs. MDL — 2D Case

Can perturb $\gamma$ by $\Delta \gamma$ with $|\Delta \gamma| < \arcsin \frac{\rho}{R}$ and still correctly separate the data.
Hence only need to store $\gamma$ with accuracy $\Delta \gamma$ [44, 57].
Formulation as an Optimization Problem

Hyperplane with maximum margin: minimize

\[ \| \mathbf{w} \|^2 \]

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

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

(i.e. the training data are separated correctly).
Lagrange Function (e.g., [5])

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

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

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

- if a constraint is violated, then $y_i \cdot (\langle w, x_i \rangle + b) - 1 < 0 \rightarrow$
  - $\alpha_i$ will grow to increase $L$ — how far?
  - $w$, $b$ 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 $y_i \cdot (\langle w, x_i \rangle + b) - 1 > 0$, 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
\[
\frac{\partial}{\partial b} L(w, b, \alpha) = 0, \quad \frac{\partial}{\partial w} L(w, b, \alpha) = 0,
\]
i.e.
\[
\sum_{i=1}^{m} \alpha_i y_i = 0
\]
and
\[
w = \sum_{i=1}^{m} \alpha_i y_i x_i.
\]
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 \( i = 1, \ldots, m \) either

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

or

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

The solution is determined by the examples on the margin.

Thus

\[ f(\mathbf{x}) = \text{sgn} \left( \langle \mathbf{x}, \mathbf{w} \rangle + b \right) \]

\[ = \text{sgn} \left( \sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right). \]
A Mechanical Interpretation

Assume that each SV $x_i$ 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:

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

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

via

$$\sum_{i} x_i \times y_i \alpha_i \cdot w / \|w\| = w \times w / \|w\| = 0.$$
Dual Problem

Dual: maximize

\[ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \]

subject to

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

Both the final decision function and the function to be maximized are expressed in dot products \( \langle x_i, x_j \rangle \rightarrow \) can use a kernel to compute

\[ \langle x_i, x_j \rangle = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j). \]
The SVM Architecture

\[ f(\mathbf{x}) = \text{sgn}(\sum_{i} \lambda_i k(\mathbf{x}, \mathbf{x}_i) + b) \]

- \( f(\mathbf{x}) = \text{sgn}(\sum + b) \)
- classification
- weights
- comparison: \( k(\mathbf{x}, \mathbf{x}_i), \text{e.g.} \) \( k(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i)^d \)
- \( \lambda_1, \lambda_2, \lambda_3, \lambda_4 \)
- support vectors \( \mathbf{x}_1 \ldots \mathbf{x}_4 \)
- input vector \( \mathbf{x} \)

B. Schölkopf, Berder, September 2004
Toy Example with Gaussian Kernel

\[ k(x, x') = \exp \left( -\| x - x' \|^2 \right) \]
Nonseparable Problems

If \( y_i \cdot (\langle w, x_i \rangle + b) \geq 1 \) cannot be satisfied, then \( \alpha_i \to \infty \).

Modify the constraint to

\[
y_i \cdot (\langle w, x_i \rangle + b) \geq 1 - \xi_i
\]

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 [13]*: for $C > 0$, minimize

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

subject to $y_i \cdot (\langle w, x_i \rangle + b) \geq 1 - \xi_i$, $\xi_i \geq 0$ (margin $2/\|w\|$)

*\nu-SVM [46]*: for $0 \leq \nu < 1$, minimize

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

subject to $y_i \cdot (\langle w, x_i \rangle + b) \geq \rho - \xi_i$, $\xi_i \geq 0$ (margin $2\rho/\|w\|$)

B. Schölkopf, Berder, September 2004
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: $\ldots = \nu = \ldots$

B. Schölkopf, Berder, September 2004
Duals, Using Kernels

**C-SVM dual:** maximize

\[
W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)
\]

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

**\(\nu\)-SVM dual:** maximize

\[
W(\alpha) = -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)
\]

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: \textit{decision function}:

\[
f(x) = \text{sgn} \left( \sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b \right)
\]

B. Schölkopf, Berder, September 2004
The Representer Theorem

Theorem 1 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 $\Omega$ 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

$$c((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) + \Omega(\|f\|) \quad (1)$$

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 \( \Omega(\|f\|) = \lambda \|f\|^2 \) (\( \lambda > 0 \)): [29]
- generalization to non-quadratic cost functions: [14]
- present form: [44]
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

$$f(x_j) = \langle f, k(x_j, .) \rangle$$

$$= \left\langle \sum_i \alpha_i k(x_i, .) + f_\perp, k(x_j, .) \right\rangle$$

$$= \sum_i \alpha_i \langle k(x_i, .), k(x_j, .) \rangle,$$

independent of $f_\perp$.
Proof: second part of (1)

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

$$\Omega(\|f\|) = \Omega \left( \| \sum_i \alpha_i k(x_i, .) + f_\perp \| \right)$$

$$= \Omega \left( \sqrt{\| \sum_i \alpha_i k(x_i, .) \|^2 + \| f_\perp \|^2} \right)$$

$$\geq \Omega \left( \| \sum_i \alpha_i k(x_i, .) \| \right),$$

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

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

$$f = \sum_i \alpha_i k(x_i, .).$$
Application: Support Vector Classification

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

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

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

$\lambda \rightarrow 0$ leads to the hard margin SVM.
Further Applications

Bayesian MAP Estimates. Identify (1) with the negative log posterior (cf. Kimeldorf & Wahba, 1970, Poggio & Giroisi, 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 [63] with covariance function \( k \)
- minimizer of (1) = MAP estimate

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

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

with \( g \) an arbitrary strictly monotonically increasing function.
SVM Training

- naive approach: the complexity of maximizing

\[ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \]

scales with the third power of the training set size \( m \)

- only SVs are relevant \( \rightarrow \) only compute \( (k(x_i, x_j))_{ij} \) for SVs. Extract them iteratively by cycling through the training set in chunks [53].

- in fact, one can use chunks which do not even contain all SVs [35]. 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 [37].

- http://www.kernel-machines.org/software.html
MNIST Benchmark

handwritten character benchmark (60000 training & 10000 test examples, 28 × 28)
**MNIST Error Rates**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>test error</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear classifier</td>
<td>8.4%</td>
<td>[7]</td>
</tr>
<tr>
<td>3-nearest-neighbour</td>
<td>2.4%</td>
<td>[7]</td>
</tr>
<tr>
<td>SVM</td>
<td>1.4%</td>
<td>[9]</td>
</tr>
<tr>
<td>Tangent distance</td>
<td>1.1%</td>
<td>[49]</td>
</tr>
<tr>
<td>LeNet4</td>
<td>1.1%</td>
<td>[31]</td>
</tr>
<tr>
<td>Boosted LeNet4</td>
<td>0.7%</td>
<td>[31]</td>
</tr>
<tr>
<td>Translation invariant SVM</td>
<td>0.56%</td>
<td>[17]</td>
</tr>
</tbody>
</table>

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

Other successful applications: e.g., [27, 25, 24, 10, 51, 8, 65, 20, 19, 12, 18, 36, 59, 64]
Further Kernel Algorithms — Design Principles

1. “Kernel module”
   - similarity measure $k(x, x')$, where $x, x' \in \mathcal{X}$
   - data representation
     (in associated feature space where $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$) — thus can construct geometric algorithms
   - function class (“representer theorem,” $f(x) = \sum_i \alpha_i k(x, x_i)$)

2. “Learning module”
   - classification
   - quantile estimation / novelty detection
   - feature extraction
   - ...

B. Schölkopf, Berder, September 2004
Unsupervised SVM Learning

\( x_1, \ldots, x_m \in \mathcal{X} \) i.i.d. sample from \( P \)

- extreme view: unsupervised learning = density estimation
- easier problem: for \( \alpha \in (0, 1] \), compute a region \( R \) such that
  \[
  P(R) \approx \alpha,
  \]
  i.e., estimate *quantiles* of a distribution, not its density.
- becomes well-posed using a regularizer: find “smoothest” region that contains a certain fraction of the probability mass
- given only the training data, we will get a trade-off: try to enclose many training points (more than \( \alpha \)) in a smooth region
Multi-Dimensional Quantiles

• $\mathcal{C}$ a class of measurable subsets of $\mathcal{X}$
• $\lambda$ a real-valued function on $\mathcal{C}$
• quantile function with respect to $(P, \lambda, \mathcal{C})$:
  \[ U(\alpha) = \inf \{ \lambda(C') | P(C') \geq \alpha, C' \in \mathcal{C} \} \quad 0 < \alpha \leq 1. \]

• present case [43]: $\lambda(C') \propto \frac{1}{\text{margin}^2}$, where
  \[ \mathcal{C} := \{ \text{half-spaces in } \mathcal{H}, \text{ not containing the origin} \} \]
Separating Unlabelled Data from the Origin

One can show: if $\Phi(x_1), \ldots, \Phi(x_m)$ are separable from the origin in $\mathcal{H}$, then the solution of

$$\min_{\mathbf{w} \in \mathcal{H}} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad \langle \mathbf{w}, \Phi(x_i) \rangle \geq 1$$

is the normal vector of the hyperplane separating the data from the origin with maximum margin.
$\nu$-Soft Margin Separation

For $\nu \in (0, 1]$, compute

$$\min_{w \in H, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}} \quad \frac{1}{2} ||w||^2 + \frac{1}{m} \sum_i \xi_i - \nu \rho$$

subject to

$$\langle w, \Phi(x_i) \rangle \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad \text{for all } i.$$

Result:

- the decision function $f(x) = \text{sgn}(\langle w, \Phi(x) \rangle - \rho)$ will be positive for “most” examples $x_i$ contained in the training set
- $||w||$ will be small, hence the separation from the origin large

Related approaches: enclose data in a sphere [42, 50]
Deriving the Dual Problem

Using multipliers $\alpha_i, \beta_i \geq 0$, we introduce a Lagrangian

$$L = \frac{\|w\|^2}{2} + \frac{1}{\nu m} \sum_i \xi_i - \rho - \sum_i \alpha_i (\langle w, \Phi(x_i) \rangle - \rho + \xi_i) - \sum_i \beta_i \xi_i,$$

and set the derivatives w.r.t. the primal variables $w, \xi, \rho$ equal to zero, yielding

$$w = \sum_i \alpha_i \Phi(x_i),$$  \hspace{1cm} (3)

$$\alpha_i = \frac{1}{\nu m} - \beta_i \leq \frac{1}{\nu m},$$  \hspace{1cm} (4)

$$\sum_i \alpha_i = 1.$$  \hspace{1cm} (5)

Patterns with $\alpha_i > 0$ are Support Vectors.
Dual Problem

\[
\min_{\alpha \in \mathbb{R}^m} \quad \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)
\]
subject to \(0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad \sum_i \alpha_i = 1.\)

The decision function is

\[
f(x) = \text{sgn} \left( \sum_i \alpha_i k(x_i, x) - \rho \right).
\]

— a thresholded sparsified Parzen windows estimator

B. Schölkopf, Berder, September 2004
Support Vectors and Outliers

\[ SV := \{i \mid \alpha_i > 0\}; \quad OL := \{i \mid \xi_i > 0\} \]

The KKT-Conditions imply:

- \( \xi_i > 0 \implies \alpha_i = 1/(\nu m) \), hence \( OL \subset SV \)
- \( SV \setminus OL \subset \{i \mid \sum_j \alpha_j k(x_j, x_i) - \rho = 0\} \)
The Meaning of $\nu$

Proposition.

(i)

$$\frac{|OL|}{m} \leq \nu \leq \frac{|SV|}{m}$$

(ii) Suppose $P$ does not contain discrete components, and the kernel is analytic and non-constant. With probability 1, asymptotically,

$$\frac{|OL|}{m} = \nu = \frac{|SV|}{m}.$$
Toy Examples using $k(x, y) = \exp\left(-\frac{||x-y||^2}{c}\right)$

<table>
<thead>
<tr>
<th>$\nu$, width $c$</th>
<th>0.5, 0.5</th>
<th>0.5, 0.5</th>
<th>0.1, 0.5</th>
<th>0.5, 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVs/OLs</td>
<td>0.54, 0.43</td>
<td>0.59, 0.47</td>
<td>0.24, 0.03</td>
<td>0.65, 0.38</td>
</tr>
</tbody>
</table>
USPS Handwritten Digit Outlier Detection

Typical examples (random selection):

```
6 9 2 8 1 8 6 5 3 2 3 8 7 0 3 0 8 2 7
```

Experiment: perform outlier detection on the 2007-element USPS test set (using $\nu = 5\%$)

Next slides: the outliers, ranked by their “badness”
$-513$
−507
-282
-216
$-162 \ 0$
$-143$
$-143 \ 6$
$-93 \ 5$
An Application to Implicit Surface Modelling

using a modified one-class SVM (Schölkopf et al., next NIPS):

B. Schölkopf, Berder, September 2004
Kernel PCA

\[ k(x,y) = (x \cdot y) \]

\[ k(x,y) = (x \cdot y)^d \]

B. Schölkopf, Berder, September 2004
Kernel PCA, II

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

Eigenvalue problem

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

For \( \lambda \neq 0 \) and \( \mathbf{v} \in \text{span}\{\Phi(x_1), \ldots, \Phi(x_m)\} \), thus

\[ \mathbf{v} = \sum_{i=1}^{m} \alpha_i \Phi(x_i), \]

and the eigenvalue problem can be written as

\[ \lambda \langle \Phi(x_n), \mathbf{v} \rangle = \langle \Phi(x_n), C \mathbf{v} \rangle \quad \text{for all} \quad n = 1, \ldots, m \]
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

$$m\lambda K \alpha = K^2 \alpha$$

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

Solve

$$m\lambda \alpha = K \alpha$$

$$\longrightarrow (\lambda_n, \alpha^n)$$

$$\langle V^n, V^n \rangle = 1 \iff \lambda_n \langle \alpha^n, \alpha^n \rangle = 1$$

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

B. Schölkopf, Berder, September 2004
Feature extraction

Compute projections on the Eigenvectors

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

in \( \mathcal{H} \):

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

\[ \langle V^n, \Phi(x) \rangle = \sum_{i=1}^{m} \alpha_i^n \langle \Phi(x_i), \Phi(x) \rangle \]

\[ = \sum_{i=1}^{m} \alpha_i^n k(x_i, x) \]

B. Schölkopf, Berder, September 2004
Toy Example with Gaussian Kernel

\[ k(x, x') = \exp \left( -\|x - x'\|^2 \right) \]
Denoising of USPS Digits

<table>
<thead>
<tr>
<th></th>
<th>Gaussian noise</th>
<th>‘speckle’ noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>orig. noisy</td>
<td>01234567890123456789</td>
<td>01234567890123456789</td>
</tr>
<tr>
<td>$n = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P = 4$</td>
<td>01234567890123456789</td>
<td>01234567890123456789</td>
</tr>
<tr>
<td>$C = 16$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A = 64$</td>
<td>01234567890123456789</td>
<td>01234567890123456789</td>
</tr>
<tr>
<td>$K = 256$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K = 4$</td>
<td>01234567890123456789</td>
<td>01234567890123456789</td>
</tr>
<tr>
<td>$P = 16$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C = 64$</td>
<td>01234567890123456789</td>
<td>01234567890123456789</td>
</tr>
<tr>
<td>$A = 256$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

linear PCA reconstruction

kernel PCA reconstruction

Another application: face modeling [39].
Natural Image KPCA Model

Training images of size $396 \times 528$. The $12 \times 12$ training patterns are obtained by sampling 2,500 patches at random from each image.

B. Schölkopf, Berder, September 2004
Super-Resolution

(Kim, Franz, & Schölkopf, 2004)

Comparison between different super-resolution methods.
Kernel Dependency Estimation

Given two sets $\mathcal{X}$ and $\mathcal{Y}$ with kernels $k$ and $k'$, and training data $(x_i, y_i)$.

Estimate a dependency $\mathbf{w} : \mathcal{H} \rightarrow \mathcal{H'}$

$$\mathbf{w}(\cdot) = \sum_{i,j} \alpha_{ij} \Phi'(y_j) \langle \Phi(x_i), \cdot \rangle.$$ 

This can be evaluated in various ways, e.g., given an $x$, we can compute the pre-image

$$y = \arg\min_{y} ||\mathbf{w}(\Phi(x)) - \Phi'(y)||.$$ 

A convenient way of learning the $\alpha_{ij}$ is to work in the kernel PCA basis.
Application to Image Completion

Shown are all digits where at least one of the two algorithms makes a mistake (73 mistakes for $k$-NN, 23 for KDE).

(from [62])

B. Schölkopf, Berder, September 2004
Kernel Machines Research

- optimization and implementation: QP, SDP (Lanckriet et al., 2002), online versions, ...
- theory of empirical inference: sharper capacity measures and bounds (Bartlett, Bousquet, & Mendelson, 2002), generalized evaluation spaces (Mary & Canu, 2002), ...
- kernel design
  - transformation invariances [11]
  - kernels for discrete objects [23, 60, 32, 16, 56]
  - kernels based on generative models [26, 48, 52]
  - local kernels [e.g., 65]
  - complex kernels from simple ones [23, 2], global kernels from local ones [30]
  - functional calculus for kernel matrices [47]
  - model selection, e.g., via alignment [15]
  - kernels for dimensionality reduction [21]
Conclusion

- crucial ingredients of SV algorithms: kernels that can be represented as dot products, and large margin regularizers
- kernels allow the formulation of a multitude of geometrical algorithms (Parzen windows, SVMs, kernel PCA,...)
- the choice of a kernel corresponds to
  - choosing a similarity measure for the data, or
  - choosing a (linear) representation of the data, or
  - choosing a hypothesis space for learning,
and should reflect prior knowledge about the problem at hand.

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


B. Schölkopf, Berder, September 2004