Learning with Kernels

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

- Elements of Statistical Learning Theory
- Kernels and feature spaces
- Support vector algorithms and other kernel methods
- Applications
Roadmap of Today

• Informal introduction to ideas of machine learning
• Learning theory: Uniform convergence
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

B. Schölkopf, Canberra, February 2006
Similarity of Inputs

• symmetric function
  \[ k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \]
  \[ (x, x') \mapsto k(x, x') \]

• 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 \( \Phi : \mathcal{X} \to \mathcal{H} \) 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. [56]

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

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

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
- the decision function is a hyperplane. Will it generalize well?
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.

[72, 73]
Example: Regression Estimation

- **Data:** input-output pairs \((x_i, y_i) \in \mathbb{R} \times \mathbb{R}\)
- **Regularity:** \((x_1, y_1), \ldots (x_m, y_m)\) 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”
Example: Pattern Recognition
Pattern Recognition

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

$$(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\},$$

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. $\rightarrow$ 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

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

- Regression estimation. RM: minimize

\[ R[f] = \int (f(x) - y)^2 \, dP(x, y) \]

— leads to the regression \( y(x) = \int y \, dP(y|x) \).

ERM gives least mean squares: minimize

\[ \sum_i (f(x_i) - y_i)^2 \]

- Density estimation. RM: minimize

\[ R[f] = \int (- \log p(x)) \, dP(x) \]

ERM gives maximum likelihood estimation: maximize

\[ \sum_i \log p(x_i) = \log(\prod_i p(x_i)) \]
Convergence of Means to Expectations

Law of large numbers:

\[ R_{\text{emp}}[f] \to R[f] \]

as \( m \to \infty \).

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

Risk

Function class

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

$R[f]$

$f$

$f_{\text{opt}}$

$f^m$

$R$

$R_{\text{emp}}$

B. Schölkopf, Canberra, February 2006
The Importance of the Set of Functions

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

Training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}$

Test patterns $\bar{x}_1, \ldots, \bar{x}_m \in \mathcal{X}$,
such that $\{\bar{x}_1, \ldots, \bar{x}_m\} \cap \{x_1, \ldots, x_m\} = \{\}$. For any $f$ there exists $f^*$ s.t.:  

1. $f^*(x_i) = f(x_i)$ for all $i$
2. $f^*(\bar{x}_j) \neq f(\bar{x}_j)$ 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’ [32, 82].

→ 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 $\mathbb{E}[\xi]$ (equals $R[f]$)
Chernoff’s Bound

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

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

Useful corollary: Given a \( 2m \)-sample of Bernoulli trials, we have

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

B. Schölkopf, Canberra, February 2006
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

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

- 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
Two Observations

• denote the minimizer of $R$ by $f^{\text{opt}}$, and the minimizer of $R_{\text{emp}}$ by $f^m$. Then we have in particular

$$R[f^m] - R[f^{\text{opt}}] \geq 0$$

and

$$R_{\text{emp}}[f^{\text{opt}}] - R_{\text{emp}}[f^m] \geq 0.$$ 

• For consistency, would like the LHS of both to converge to 0 in probability.

• If the sum of the two converges to 0, we are done.
The sum of these two inequalities satisfies

\[ 0 \leq R[f^m] - R[f^{opt}] + R_{emp}[f^{opt}] - R_{emp}[f^m] \]
\[ = R[f^m] - R_{emp}[f^m] + R_{emp}[f^{opt}] - R[f^{opt}] \]
\[ \leq \sup_{f \in \mathcal{F}} (R[f] - R_{emp}[f]) + (R_{emp}[f^{opt}] - R[f^{opt}]). \]

- second half of RHS: \( f^{opt} \) is fixed (independent of training sample), hence by Chernoff: for all \( \epsilon > 0 \),

\[ \lim_{m \to \infty} P\{|R_{emp}[f^{opt}] - R[f^{opt}]| > \epsilon\} = 0 \]

(“convergence in probability”)
If the first half of RHS also converges to zero (in probability), i.e.,

\[ \lim_{m \to \infty} \mathbb{P}\{ \sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon \} = 0, \]

for all \( \epsilon > 0 \), then

\[ R[f^m] - R[f^{\text{opt}}] \to 0 \]
\[ R_{\text{emp}}[f^{\text{opt}}] - R_{\text{emp}}[f^m] \to 0 \]

in probability — in this case, empirical risk minimization can be seen to be \textit{consistent}.
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.

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

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 (≡ sets of functions) which ensure uniform convergence of risk?

B. Schölkopf, Canberra, February 2006
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

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

where

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

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

$$P(C^1_\epsilon \cup C^2_\epsilon) = P(C^1_\epsilon) + P(C^2_\epsilon) - P(C^1_\epsilon \cap C^2_\epsilon) \leq P(C^1_\epsilon) + P(C^2_\epsilon).$$

Hence by Chernoff’s bound

$$P\left\{ \sup_{f \in \mathcal{F}} (R[f] - R_{\text{emp}}[f]) > \epsilon \right\} \leq P(C^1_\epsilon) + P(C^2_\epsilon) \leq 2 \cdot 2 \exp(-2m\epsilon^2).$$
The Union Bound

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

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

and

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

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

B. Schölkopf, Canberra, February 2006
Symmetrization

Lemma 1 (Vapnik & Chervonenkis (e.g., [69, 20]))

For $m^2 \geq 2$ we have

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

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, $R_{emp}$ measures the loss on the first half of the sample, and $R'_{emp}$ on the second half.
Shattering Coefficient

• Hence, we only need to consider the maximum size of $\mathcal{F}$ on $2m$ points. Call it $\mathcal{N}(\mathcal{F}, 2m)$.

• $\mathcal{N}(\mathcal{F}, 2m) = \text{max. number of different outputs } (y_1, \ldots, y_{2m}) \text{ that the function class can generate on } 2m \text{ points} — \text{ in other words, the max. number of different ways the function class can separate } 2m \text{ points into two classes.}$

• $\mathcal{N}(\mathcal{F}, 2m) \leq 2^{2m}$

• 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

\[ P\{ \sup_{f \in F} (R[f] - R_{\text{emp}}[f]) > \epsilon \} \]
\[ \leq 2P\{ \sup_{f \in F} (R_{\text{emp}}[f] - R'_{\text{emp}}[f]) > \epsilon / 2 \} \]
\[ = 2P\{ (R_{\text{emp}}[f_1] - R'_{\text{emp}}[f_1]) > \epsilon / 2 \lor \ldots \lor (R_{\text{emp}}[f_{\mathcal{N}(\mathcal{F},2m)}] - R'_{\text{emp}}[f_{\mathcal{N}(\mathcal{F},2m)}]) > \epsilon / 2 \} \]
\[ \leq \sum_{n=1} \, 2P\{ (R_{\text{emp}}[f_n] - R'_{\text{emp}}[f_n]) > \epsilon / 2 \}. \]
Use Chernoff’s bound for each term:

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

This yields

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

• 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 \( P \) 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 [56].
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.

B. Schölkopf, Canberra, February 2006
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 \((x, y)\), \(f\) causes a loss

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

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

\[
\xi_f = (\xi(x_1, y_1, f(x_1)), \ldots, \xi(x_m, y_m, f(x_m))),
\]

whose cardinality we denote by

\[
\mathcal{N}(\mathcal{F}, (x_1, y_1) \ldots, (x_m, y_m)).
\]

The VC entropy is defined as

\[
H_{\mathcal{F}}(m) = \mathbb{E} \left[ \ln \mathcal{N}(\mathcal{F}, (x_1, y_1) \ldots, (x_m, y_m)) \right],
\]

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

\[
H_{\mathcal{F}}(m)/m \to 0 \iff \text{uniform convergence of risks (hence consistency)}
\]
Further PR Capacity Concepts

• exchange ’E’ and ’ln’: *annealed entropy*.

\[ H^{\text{ann}}_{\mathcal{F}}(m)/m \to 0 \iff \text{exponentially fast uniform convergence} \]

• take ’max’ instead of ’E’: *growth function*.

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

\[ G_{\mathcal{F}}(m)/m \to 0 \iff \text{exponential convergence for all underlying distributions } \mathcal{P}. \]

\[ G_{\mathcal{F}}(m) = m \cdot \ln(2) \text{ for all } m \iff \text{for any } m, \text{ all loss vectors can be generated, i.e., the } m \text{ points can be chosen such that by using functions of the learning machine, they can be separated in all } 2^m \text{ possible ways (shattered).} \]
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 \textit{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)
For any $f \in F$ and $m > h$, with a probability of at least $1 - \delta$,

$$R[f] \leq R_{\text{emp}}[f] + \phi \left( \frac{h}{m}, \frac{\log(\delta)}{m} \right)$$

holds, where the confidence term $\phi$ is defined as

$$\phi \left( \frac{h}{m}, \frac{\log(\delta)}{m} \right) = \sqrt{\frac{h \left( \log \frac{2m}{h} + 1 \right) - \log(\delta/4)}{m}}.$$

- 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 a better induction principle: we can use this bound to get a low risk!
- in practice: use as a guideline for designing algorithms

B. Schölkopf, Canberra, February 2006
Examples of Induction Principles

- **Empirical risk minimization:** minimize

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

- **Minimum description length:** minimize some measure of the description length of the sequence \((x_1, y_1), \ldots, (x_m, y_m)\) by a function \(f\).

- **Structural risk minimization (SRM) (Vapnik, 1979):** minimize the RHS of

\[
R[f] \leq R_{\text{emp}}[f] + \phi \left( \frac{h}{m} \right).
\]

To this end, introduce a structure on \(\mathcal{F}\).

Learning machine \(\equiv\) a set of functions and an induction principle
SRM: The Picture

\[ \mathcal{R}(f_{\star}) \]

bound on test error

capacity term

training error

\[ \ldots S_{n-1} \subset S_n \subset S_{n+1} \ldots \]

B. Schölkopf, Canberra, February 2006
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
Separating Hyperplane

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

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

\[ \{ x \mid \langle w, x \rangle + b = 0 \} \]
Canonical Hyperplanes

Note: if $c \neq 0$, then

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

Hence $(cw, cb)$ describes the same hyperplane as $(w, b)$.

**Definition:** The hyperplane is in *canonical* form w.r.t. $X^* = \{x_1, \ldots, x_r\}$ if $\min_{x_i \in X} |\langle w, x_i \rangle + b| = 1$.

Note that for canonical hyperplanes, the distance of the closest point to the hyperplane ("margin") is $1/\|w\|:

$$\min_{x_i \in X} \left| \frac{\langle w, x_i \rangle}{\|w\|} + \frac{b}{\|w\|} \right| = \frac{1}{\|w\|}.$$ 

B. Schölkopf, Canberra, February 2006
Theorem 2 (Vapnik [69]) Consider hyperplanes $\langle \mathbf{w}, \mathbf{x} \rangle = 0$ where $\mathbf{w}$ is normalized such that they are in canonical form w.r.t. a set of points $X^* = \{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$, i.e.,

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

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

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

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

B. Schölkopf, Canberra, February 2006
Proof Strategy (Gurvits, 1997)

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

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \text{ for all } i = 1, \ldots, r.$$  \hspace{1cm} (1)

Two steps:

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

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

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

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

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

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

Combine both:

$$\frac{r}{\Lambda} \leq \left\| \sum_{i=1}^{r} y_i x_i \right\|.$$  

(2)

B. Schölkopf, Canberra, February 2006
Consider independent random labels $y_i \in \{\pm 1\}$, uniformly distributed (Rademacher variables).

\[
E \left[ \left\| \sum_{i=1}^{r} y_i x_i \right\|^2 \right] = \sum_{i=1}^{r} E \left[ \left\langle y_i x_i, \sum_{j=1}^{r} y_j x_j \right\rangle \right] \\
= \sum_{i=1}^{r} E \left[ \left\langle y_i x_i, \left( \left( \sum_{j \neq i} y_j x_j \right) + y_i x_i \right) \right\rangle \right] \\
= \sum_{i=1}^{r} \left( \left( \sum_{j \neq i} E \left[ \left\langle y_i x_i, y_j x_j \right\rangle \right] \right) + E \left[ \left\langle y_i x_i, y_i x_i \right\rangle \right] \right) \\
= \sum_{i=1}^{r} E \left[ \|y_i x_i\|^2 \right] = \sum_{i=1}^{r} \|x_i\|^2
\]
Part II, ctd.

Since \( \|x_i\| \leq R \), we get

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

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

Hence

\[
\left\| \sum_{i=1}^{r} y_i x_i \right\|^2 \leq rR^2.
\]

B. Schölkopf, Canberra, February 2006
Part I and II Combined

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

Part II: \( \left\| \sum_{i=1}^{r} y_i x_i \right\|^2 \leq r R^2 \)

Hence

\[
\frac{r^2}{\Lambda^2} \leq r R^2,
\]

i.e.,

\[
r \leq R^2 \Lambda^2.
\]

B. Schölkopf, Canberra, February 2006
Pattern Noise as Maximum Margin Regularization
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 $\Delta \gamma$ [56, 75].
Kernels and Feature Spaces

Preprocess the data with

\[
\Phi : \mathcal{X} \rightarrow \mathcal{H}
\]

\[
x \mapsto \Phi(x),
\]

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

• usually, \( \text{dim}(\mathcal{X}) \ll \text{dim}(\mathcal{H}) \)
• “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 $d = 5 \rightarrow$ 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 $\mathbb{R}^2$
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.

B. Schölkopf, Canberra, February 2006
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 \( \lambda_i \geq 0 \) [42].

B. Schölkopf, Canberra, February 2006
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:

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

B. Schölkopf, Canberra, February 2006
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:
  - Polynomial \( k(x, x') = (\langle x, x' \rangle + c)^d \)
  - Sigmoid \( k(x, x') = \tanh(\kappa \langle x, x' \rangle + \Theta) \)
  - Gaussian \( k(x, x') = \exp(-\|x - x'\|^2/(2 \sigma^2)) \)
- Kernels are studied also in the Gaussian Process prediction community (covariance functions) [79, 76, 81, 41]
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

$$\sum_{i,j} a_i a_j K_{ij} \geq 0,$$

where $K_{ij} := k(x_i, x_j)$.

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

B. Schölkopf, Canberra, February 2006
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

• 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

\[ \langle \Phi(x), \Phi(x') \rangle = k(x, x'), \text{ i.e., } \langle k(., x), k(., x') \rangle = k(x, x') \]

• complete the space to get a reproducing kernel Hilbert space
Turn it Into a Linear Space

Form linear combinations

\[ f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i), \]
\[ g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j) \]

\((m, m' \in \mathbb{N}, \alpha_i, \beta_j \in \mathbb{R}, x_i, x'_j \in \mathcal{X}).\)
Endow it With a Dot Product

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

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

• This is well-defined, symmetric, and bilinear (more later).

B. Schölkopf, Canberra, February 2006
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
Endow it With a Dot Product, II

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

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

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

- 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 \( \langle \psi_i, \psi_j \rangle_{L_2(X)} = \delta_{ij} \)), this can be achieved by choosing \( \langle ., . \rangle \) such that
\[ \langle \psi_i, \psi_j \rangle = \delta_{ij}/\lambda_i. \]
To see this, compute

\[
\langle k(x, .), k(x', .) \rangle = \left\langle \sum_i \lambda_i \psi_i(x) \psi_i, \sum_j \lambda_j \psi_j(x') \psi_j \right\rangle
\]

\[
= \sum_{i,j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \langle \psi_i, \psi_j \rangle
\]

\[
= \sum_{i,j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \delta_{ij} / \lambda_i
\]

\[
= \sum_i \lambda_i \psi_i(x) \psi_i(x')
\]

\[
= k(x, x').
\]
Deriving the Kernel from the RKHS

An RKHS is a Hilbert space $\mathcal{H}$ of functions $f$ where all point evaluation functionals
$$p_x : \mathcal{H} \to \mathbb{R}$$
$$f \mapsto p_x(f) = f(x)$$
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

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

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

• 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 $i, j = 1, \ldots, m$.  

B. Schölkopf, Canberra, February 2006
Some Properties of Kernels \[56\]

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 \( X \)
  (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 \[30\].
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

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

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

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


Properties, II: Functional Calculus [60]

- $K$ 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 $\mathbb{R}$
  
  \[
  \begin{align*}
  (\alpha f + g)(K) &= \alpha f(K) + g(K) \\
  (fg)(K) &= f(K)g(K) = g(K)f(K) \\
  \|f\|_{\infty,\sigma(K)} &= \|f(K)\| \\
  \sigma(f(K)) &= f(\sigma(K))
  \end{align*}
  \]

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

Clearly, if \( k \) is positive definite, then there exists a map \( \Phi \) such that
\[
\|\Phi(x) - \Phi(x')\|^2 = k(x, x) + k(x', x') - 2k(x, x')
\]
(it is the usual feature map).

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

In fact, all algorithms that are translationally invariant (i.e. independent of the choice of the origin) in \( \mathcal{H} \) work with cpd kernels [56].
Support Vector Classifiers

input space

feature space

\[ \Phi \]
Separating Hyperplane

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

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

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

\[ \{ x \mid \langle w, x \rangle + b = 0 \} \]
Note: if $c \neq 0$, then
\[
\{ x | \langle w, x \rangle + b = 0 \} = \{ x | \langle cw, x \rangle + cb = 0 \}.
\]
Hence $(cw, cb)$ describes the same hyperplane as $(w, b)$.

**Definition:** The hyperplane is in *canonical* form w.r.t. $X^* = \{ x_1, \ldots, x_r \}$ if $\min_{x_i \in X} | \langle w, x_i \rangle + b | = 1$. 
Canonical Optimal Hyperplane

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

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

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

\begin{align*}
\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||}
\end{align*}

B. Schölkopf, Canberra, February 2006
Formulation as an Optimization Problem

Hyperplane with maximum margin: minimize

\[ \|w\|^2 \]

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

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

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

B. Schölkopf, Canberra, February 2006
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

\[ w = \sum_{i=1}^{m} \alpha_i y_i x_i \]

where for all \( i = 1, \ldots, m \) either

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

or

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

The solution is determined by the examples on the margin.

Thus

\[ f(x) = \text{sgn} (\langle x, w \rangle + b) \]
\[ = \text{sgn} \left( \sum_{i=1}^{m} \alpha_i y_i \langle x, x_i \rangle + b \right). \]
Why it is Good to Have Few SVs

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

**Theorem [70]:** Denote $\#SV(m)$ the number of SVs obtained by training on $m$ examples randomly drawn from $P(x, y)$, and $E$ the expectation. Then

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

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

Assume that each SV $\mathbf{x}_i$ exerts a perpendicular force of size $\alpha_i$ and sign $y_i$ on a solid plane sheet lying along the hyperplane. Then the solution is mechanically stable:

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

implies that the forces sum to zero.

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

implies that the torques sum to zero, via

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

B. Schölkopf, Canberra, February 2006
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 \( \langle x_i, x_j \rangle \) can use a kernel to compute

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

B. Schölkopf, Canberra, February 2006
The SVM Architecture

\[ f(x) = \text{sgn} \left( \sum \lambda_i k(x, x_i) + b \right) \]

- **Input vector** \( x \)
- **Support vectors** \( x_1 \ldots x_4 \)
- **Comparison**: \( k(x, x_i) \), e.g.,
  - \( k(x, x_i) = (x \cdot x_i)^d \)
  - \( k(x, x_i) = \exp(-||x-x_i||^2 / c) \)
  - \( k(x, x_i) = \tanh(\kappa(x \cdot x_i) + \theta) \)
- **Weights** \( \lambda_1 \ldots \lambda_4 \)
- **Classification**

B. Schölkopf, Canberra, February 2006
Toy Example with Gaussian Kernel

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

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

Modify the constraint to

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

with

$$\xi_i \geq 0$$

(“soft margin”) and add

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

in the objective function.
Soft Margin SVMs

**C-SVM** [15]: 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** [58]: 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 $2\rho/\|w\|$)
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, Canberra, February 2006
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: decision function:

\[
f(x) = \text{sgn} \left( \sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b \right)
\]
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 $w_0, b_0, \xi_0$ minimizes the primal problem of $C$-SVC, for $C = 1$, subject to

$$y_i \cdot (\langle x_i, w \rangle + b) \geq \rho - \xi_i.$$ 

To recover the constraint

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

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

B. Schölkopf, Canberra, February 2006
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 [69].

• in fact, one can use chunks which do not even contain all SVs [43]. 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 [46].

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

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

B. Schölkopf, Canberra, February 2006
### 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>[11]</td>
</tr>
<tr>
<td>Tangent distance</td>
<td>1.1%</td>
<td>[62]</td>
</tr>
<tr>
<td>LeNet4</td>
<td>1.1%</td>
<td>[39]</td>
</tr>
<tr>
<td>Boosted LeNet4</td>
<td>0.7%</td>
<td>[39]</td>
</tr>
<tr>
<td>Translation invariant SVM</td>
<td>0.56%</td>
<td>[19]</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., [35, 33, 31, 12, 67, 9, 84, 26, 24, 14, 22, 45, 77, 83]
Speeding up the decision rule

Approximate
\[ w = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i) \]
by
\[ w' = \sum_{i=1}^{N_z} \gamma_i \Phi(z_i), \]
with \( N_z \ll m \): Minimize
\[ \rho = \| w - w' \|^2 \]
Note that \( \rho \) can be expressed in terms of \( k \) by using
\[ \langle \Phi(x), \Phi(x') \rangle = k(x, x') \]
Construct the new expansion sequentially.
“reduced set methods”, [e.g. 10, 11, 44, 53]

B. Schölkopf, Canberra, February 2006
Face Detection

• scan test images in several resolutions

• critical issue: runtime speed. Compute sequential approximation via reduced set expansion.

• need to evaluate on average 2 – 3 kernels per image location [49]

after 0, 1 (13.3% patches remaining), 10 (2.6%), 20 (0.01%) and 30 (0.002%) kernels

templates:

B. Schölkopf, Canberra, February 2006
Invariant Hyperplanes

Consider decision functions \( f(x) = \text{sgn}(g(x)) \), where

\[
g(x) := \sum_{i=1}^{m} v_i \langle Bx, Bx_i \rangle + b.
\]

To get local invariance under transformations of the Lie group \( \{L_t\} \), minimize the regularizer

\[
\frac{1}{m} \sum_{j=1}^{m} \left( \frac{\partial}{\partial t} \bigg|_{t=0} g(L_t x_j) \right)^2.
\]

This corresponds to an SV optimization after preprocessing with

\[
B = C^{-\frac{1}{2}},
\]

where

\[
C = \frac{1}{m} \sum_{j=1}^{m} \left( \frac{\partial}{\partial t} \bigg|_{t=0} \mathcal{L}_t x_j \right) \left( \frac{\partial}{\partial t} \bigg|_{t=0} \mathcal{L}_t x_j \right)^\top.
\]
The Tangent Covariance Matrix

\[ C = \text{covariance matrix of } \pm \frac{\partial}{\partial t} \big|_{t=0} \mathcal{L}_t x \]

Preprocessing of \( x \):

\[ B x = C^{-\frac{1}{2}} x = S D^{-\frac{1}{2}} S^\top x \]

1. project \( x \) onto the Eigenvectors of \( C \)

2. divide by the square roots of the Eigenvalues, i.e.: the directions of main variance of \( \pm \frac{\partial}{\partial t} \big|_{t=0} \mathcal{L}_t x \) are scaled back

- in practice, use \( C_\lambda := (1 - \lambda)C + \lambda I \)

- for the nonlinear case, use the kernel PCA map [13]
USPS Digit Recognition Application [13]

Results for 4 invariance transformations (translations) and different trade-offs between margin maximization and invariance enforcement (left: standard SVM).

B. Schölkopf, Canberra, February 2006
SV Regression: $\varepsilon$-Insensitive Loss (Vapnik, 1995)

Goal: generalize SV pattern recognition to regression, preserving the following properties:

- formulate the algorithm for the linear case, and then use kernel trick
- sparse representation of the solution in terms of SVs

$\varepsilon$-Insensitive Loss:

$$|y - f(x)|_\varepsilon := \max\{0, |y - f(x)| - \varepsilon\}$$

Estimate a linear regression $f(x) = \langle w, x \rangle + b$ by minimizing

$$\frac{1}{2} \|w\|^2 + \frac{C}{m} \sum_{i=1}^{m} |y_i - f(x_i)|_\varepsilon.$$
$\varepsilon$-SV Regression Estimation

B. Schölkopf, Canberra, February 2006
Formulation as an Optimization Problem

Estimate a linear regression

\[ f(x) = \langle w, x \rangle + b \]

with precision \( \varepsilon \) by minimizing

\[
\begin{align*}
\text{minimize} & \quad \tau(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \\
\text{subject to} & \quad (\langle w, x_i \rangle + b) - y_i \leq \varepsilon + \xi_i \\
& \quad y_i - (\langle w, x_i \rangle + b) \leq \varepsilon + \xi_i^* \\
& \quad \xi_i, \xi_i^* \geq 0
\end{align*}
\]

for all \( i = 1, \ldots, m \).
Dual Problem, In Terms of Kernels

For $C > 0, \varepsilon \geq 0$ chosen a priori,

maximize $W(\alpha, \alpha^*) = -\varepsilon \sum_{i=1}^{m} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) y_i$

$- \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(x_i, x_j)$

subject to $0 \leq \alpha_i, \alpha_i^* \leq C, \ i = 1, \ldots, m,$ and $\sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0.$

The regression estimate takes the form

$f(x) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) k(x_i, x) + b,$
\( \nu \)-SV Regression

Again, use \( \nu \) to eliminate another parameter: Estimate \( \varepsilon \) from the data s.t. the \( \nu \)-property holds.

Primal problem: for \( 0 \leq \nu \leq 1 \), minimize

\[
\tau(w, \varepsilon) = \frac{1}{2} \|w\|^2 + C \left( \nu \varepsilon + \frac{1}{m} \sum_{i=1}^{m} |y_i - f(x_i)|_{\varepsilon} \right)
\]

B. Schölkopf, Canberra, February 2006
A Graphical Proof of the $\nu$-Property

Cost function: $\frac{1}{2C}\|w\|^2 + \nu\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(\xi_i + \xi^*_i)$

B. Schölkopf, Canberra, February 2006
The $\nu$-Property

**Proposition 3** Assume $\varepsilon > 0$. The following statements hold:

(i) $\nu$ is an upper bound on the fraction of errors.

(ii) $\nu$ is a lower bound on the fraction of SVs.

(iii) Suppose the data were generated iid from a ’well-behaved’* distribution $P(x, y)$. With probability 1, asymptotically, $\nu$ equals both the fraction of SVs and the fraction of errors.

* Essentially, $P(x, y) = P(x)P(y|x)$ with $P(y|x)$ continuous (some details omitted).
Identical machine parameters ($\nu = 0.2$), but different amounts of noise in the data.
\(\varepsilon\)-SV-Regression, Run on the Same Data

Identical machine parameters \((\varepsilon = 0.2)\), but different amounts of noise in the data.

B. Schölkopf, Canberra, February 2006
**Toy Examples: Estimating a Noisy Sinc Function**


\[ \nu = 0.2 \]

<table>
<thead>
<tr>
<th>( m )</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon )</td>
<td>0.27</td>
<td>0.22</td>
<td>0.23</td>
<td>0.25</td>
<td>0.26</td>
<td>0.26</td>
<td>0.26</td>
<td>0.26</td>
</tr>
<tr>
<td>fraction of errors</td>
<td>0.00</td>
<td>0.10</td>
<td>0.14</td>
<td>0.18</td>
<td>0.19</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>fraction of SVs</td>
<td>0.40</td>
<td>0.28</td>
<td>0.24</td>
<td>0.23</td>
<td>0.21</td>
<td>0.21</td>
<td>0.20</td>
<td>0.20</td>
</tr>
</tbody>
</table>

- automatically computed \( \varepsilon \) largely independent of \( m \)
- asymptotics consistent with theorem
Boston Housing Benchmark

- 506 examples, 13-dimensional.

Results (MSE):
- Bagging regression trees: 11.7 [8]
- $\varepsilon$-SV regression: 7.6 [64]

- 100 runs, with 25 randomly selected test points.
- training set is split into actual training set and validation set (80 points) for selecting $\varepsilon$, $C$, and kernel parameters
**Comparison: $\nu$ vs. $\varepsilon$**

<table>
<thead>
<tr>
<th>$\nu$-SVR</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>automatic $\varepsilon$</td>
<td>2.6</td>
<td>1.7</td>
<td>1.2</td>
<td>0.8</td>
<td>0.6</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MSE</td>
<td>9.4</td>
<td>8.7</td>
<td>9.3</td>
<td>9.5</td>
<td>10.0</td>
<td>10.6</td>
<td>11.3</td>
<td>11.3</td>
<td>11.3</td>
<td>11.3</td>
</tr>
<tr>
<td>Errors</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>SVs</td>
<td>0.3</td>
<td>0.4</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\varepsilon$-SVR</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>11.3</td>
<td>9.5</td>
<td>8.8</td>
<td>9.7</td>
<td>11.2</td>
<td>13.1</td>
<td>15.6</td>
<td>18.2</td>
<td>22.1</td>
<td>27.0</td>
<td>34.3</td>
</tr>
<tr>
<td>Errors</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SVs</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- RBF kernel, $C$ and $\sigma$ chosen as in [59]
Parametric Error Models

Use a tube of varying radius $\zeta(x) \geq 0$:

minimize

$$
\tau(w, \xi^(*), \varepsilon) = \|w\|^2/2 + C \cdot \left( \nu m \varepsilon + \sum_{i=1}^{m} (\xi_i + \xi_i^*) \right)
$$

subject to

$$(\langle w, x_i \rangle + b) - y_i \leq \varepsilon \zeta(x_i) + \xi_i$$

$$y_i - (\langle w, x_i \rangle + b) \leq \varepsilon \zeta(x_i) + \xi_i^*$$

$$\xi_i^(*) \geq 0, \quad \varepsilon \geq 0.$$ 

This leads to the “usual” dual, with the modified last constraint

$$\sum_{i=1}^{m} (\alpha_i + \alpha_i^*) \zeta(x_i) \leq C m \nu.$$ 

B. Schölkopf, Canberra, February 2006
Assumption: we have prior knowledge indicating that the noise is modulated by $\zeta(x) = \sin^2((2\pi/3)x)$. 

B. Schölkopf, Canberra, February 2006
constant-radius tube  
parametric model using $\zeta(x)$
Robustness of SV Regression

Proposition. Using SVR with $|.|_\varepsilon$, local movements of target values of points outside the tube do not change the estimated regression.

Proof.

1. Shift $y_i$ locally $\rightarrow (x_i, y_i)$ still outside the tube $\rightarrow$ original dual solution $\alpha^{(*)}$ still feasible ($\alpha_i^{(*)} = C$, since all points outside the tube are at the upper bound).

2. The primal solution, with $\xi_i$ transformed according to the movement, is also feasible.

3. The KKT conditions are still satisfied, as still $\alpha_i^{(*)} = C$. Thus [5, e.g.], $\alpha^{(*)}$ is still the optimal solution.
The Representer Theorem

**Theorem 4** 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

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

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 \( \Omega(\|f\|) = \lambda \|f\|^2 (\lambda > 0) \): [37]

• generalization to non-quadratic cost functions: [16]

• present form: [56]
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 $f_\perp$. 

B. Schölkopf, Canberra, February 2006
**Proof: second part of (3)**

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

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

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

\[
\geq \Omega \left( \| \sum_i \alpha_i k(x_i, .) \| \right), \tag{4}
\]

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

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

\[
f = \sum_i \alpha_i k(x_i, .).
\]
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 $\Omega(\|f\|) = \|f\|^2$.

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

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

- \( \exp(-c((x_i, y_i, f(x_i)))_i) \) — likelihood of the data
- \( \exp(-\Omega(\|f\|)) \) — prior over the set of functions; e.g., \( \Omega(\|f\|) = \lambda \|f\|^2 \) — Gaussian process prior [81] with covariance function \( k \)
- minimizer of (3) = MAP estimate

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

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

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

$$\|w\| = \|Pf\|,$$

where

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

and

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

Example: for the Gaussian kernel, $P$ is a linear combination of differential operators.
\[ \|w\|^2 = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \]
\[ = \sum_{i,j} \alpha_i \alpha_j \langle k(x_i, .), \delta_{x_j}(.) \rangle \]
\[ = \sum_{i,j} \alpha_i \alpha_j \langle k(x_i, .), (P^* Pk)(x_j, .) \rangle \]
\[ = \sum_{i,j} \alpha_i \alpha_j \langle (P^* Pk)(x_i, .), (Pk)(x_j, .) \rangle \}
\[ = \langle \left( P \sum_{i} \alpha_i k(x_i, .) \right), \left( P \sum_{j} \alpha_j k(x_j, .) \right) \rangle_D \]
\[ = \| P f \|^2, \]
using \( f(x) = \sum_i \alpha_i k(x_i, x) \).
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, Canberra, February 2006
SV Morphing

...powerpoint
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 [54]: $\lambda(C) \propto \frac{1}{\text{margin}^2}$, 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

\[
\min_{\mathbf{w} \in \mathcal{H}, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{1}{m} \sum_i \xi_i - \nu \rho
\]

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

Result:

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

Related approaches: enclose data in a sphere [52, 65]
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

$$w = \sum_i \alpha_i \Phi(x_i), \quad (5)$$

$$\alpha_i = \frac{1}{\nu m} - \beta_i \leq \frac{1}{\nu m}, \quad (6)$$

$$\sum_i \alpha_i = 1. \quad (7)$$

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

\[
\min_{\alpha \in \mathbb{R}^m} \quad \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j)
\]
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, Canberra, February 2006
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>

B. Schölkopf, Canberra, February 2006
Error Bound for Single-Class Classification

For $x \in \mathcal{X}, \theta \in \mathbb{R}$, let $d(x, f, \theta) := \max\{0, \theta - f(x)\}$. Similarly for $X := (x_1, \ldots, x_m)$, $\mathcal{D}(X, f, \theta) := \sum_{x \in X} d(x, f, \theta)$.

**Theorem 5** Denote

- $X \in \mathcal{X}^m$ a sample generated from an unknown distribution $P$, without discrete components
- $f_w$ the solution of the optimization problem,
- $R_{w, \rho} := \{x|f_w(x) \geq \rho\}$ the induced decision region.

With probability $1 - \delta$, for any $\gamma > 0$,

$$P \left\{ x'|x' \not\in R_{w, \rho - \gamma} \right\} \leq \frac{2}{m} \left( k + \log m^2 / (2\delta) \right),$$

where

$$k = \frac{c_1 \log(c_2 \hat{\gamma}^2 m)}{\hat{\gamma}^2} + \frac{2\mathcal{D}}{\hat{\gamma}} \log \left( e \left( \frac{(2m - 1)\hat{\gamma}}{2\mathcal{D}} + 1 \right) \right) + 2,$$

$c_1 = 16c^2$, $c_2 = \ln(2)/(4c^2)$, $c = 103$, $\hat{\gamma} = \gamma/\|w\|$, $\mathcal{D} = \mathcal{D}(X, f_w, 0, \rho) = \mathcal{D}(X, f_w, \rho, 0)$. 
Discussion

• algorithm tries to enclose training sample in $R_w, \rho$
• theorem bounds the probability that test points will be in the larger region $R_w, \rho - \gamma$
• a small $\gamma$ leads to a small region but a large complexity term
• a small $\|w\|$ leads to a small complexity term (recall $\hat{\gamma} = \gamma/\|w\|$)
Typical examples (random selection):

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

Next slides: the outliers, ranked by their “badness”
Kernel PCA

**linear PCA**

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

**kernel PCA**

\[ k(x,y) = (x \cdot y)^d \]
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 \( \lambda \neq 0 \), \( \mathbf{V} \in \text{span}\{\Phi(x_1), \ldots, \Phi(x_m)\} \), thus

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

and the eigenvalue problem can be written as

\[
\lambda \langle \Phi(x_n), \mathbf{V} \rangle = \langle \Phi(x_n), C \mathbf{V} \rangle \text{ for all } n = 1, \ldots, m
\]
**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, Canberra, February 2006
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, Canberra, February 2006
The Kernel PCA Map

Recall

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

$$x \mapsto K^{-\frac{1}{2}}(k(x_1, x), \ldots, k(x_m, x))^\top$$

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

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

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

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

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

B. Schölkopf, Canberra, February 2006
Toy Example with Gaussian Kernel

\[ k(x, x') = \exp \left( -\|x - x'\|^2 \right) \]
Kernel PCA Denoising

Idea: in feature space, discard higher-order principal components, and compute approximate pre-images [53].

Original data, first 8 feature extractors (*left*), pre-images computed by retaining 1...8 components in feature space (*right*).
## Comparison of Different Algorithms

<table>
<thead>
<tr>
<th>kernel PCA (4 PCs)</th>
<th>nonlinear autoencoder</th>
<th>Principal Curves</th>
<th>linear PCA (1 PC)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="kernel PCA" /></td>
<td><img src="image" alt="nonlinear autoencoder" /></td>
<td><img src="image" alt="Principal Curves" /></td>
<td><img src="image" alt="linear PCA" /></td>
</tr>
</tbody>
</table>

[53, 29, 21]

B. Schölkopf, Canberra, February 2006
### Denoising of USPS Digits

<table>
<thead>
<tr>
<th>$n = 1$</th>
<th>Gaussian noise</th>
<th>‘speckle’ noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>orig.</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
<tr>
<td>noisy</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
</tbody>
</table>

**PCA Reconstruction**

<table>
<thead>
<tr>
<th>$n = 1$</th>
<th>Gaussian noise</th>
<th>‘speckle’ noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
<tr>
<td>$C$</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
<tr>
<td>$A$</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
<tr>
<td>$K$</td>
<td>0123456789</td>
<td>0123456789</td>
</tr>
</tbody>
</table>

**Kernel PCA Reconstruction**

Other applications: face modeling [48], image superresolution (see below).
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.
Example of natural image super-resolution: a. original image of resolution $528 \times 396$, b. low resolution image ($264 \times 198$) stretched to the original scale, c. reconstruction of the high-frequency components.
Super-Resolution

(Kim, Franz, & Schölkopf, 2004)

Comparison between different super-resolution methods.

B. Schölkopf, Canberra, February 2006
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 $w : \mathcal{H} \to \mathcal{H}'$

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

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

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

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

B. Schölkopf, Canberra, February 2006
Shown are all digits where at least one of the two algorithms makes a mistake (73 mistakes for $k$-NN, 23 for KDE).

(from [80])

B. Schölkopf, Canberra, February 2006
Vector Quantization

- given a set of \( m \) data vectors \( X = x_1, \ldots, x_m \)
- wish to represent them by a reduced number of \( M \) ‘codebook’ vectors \( V = v_1, \ldots, v_M \)
- Codebook \( V \) is chosen such that some overall measure of distortion is (approximately) minimized when each \( x \) is represented by its ‘nearest’ \( v \):

\[
E_{VQ} = \sum_{n=1}^{m} D [v(x_n), x_n]
\]

where \( v(x_n) = \arg\min_{v \in V} D [v, x_n] \)

- A common distortion is squared Euclidean distance: \( D [v, x_n] = \|v - x_n\|^2 \)
Kernel VQ

- Conventionally: specify codebook size $M$ and minimize $EV_Q$ over $V$
  - e.g., Linde-Buzo-Gray (LBG) algorithm
- kernel approach [66]:
  - specify a maximum distortion guarantee:
    \[ D[v(x_n), x_n] \leq R \] (\(^*)\)
  - constrain the codebook to be a subset of the data set:
    \[ \{v_1, \ldots, v_M\} \subseteq \{x_1, \ldots, x_m\} \]
  - try to find $v_1, \ldots, v_M$ with minimal $M$ such that (\(^*)\) holds
    
    (Tipping & Schölkopf, 2001 [66])
• define a kernel:

\[ k(x_i, x_n) = \begin{cases} 
1 & \text{if } D[x_i, x_n] \leq R \\
0 & \text{otherwise} 
\end{cases} \]

• seek a sparse vector \( \mathbf{w} = (w_1, \ldots, w_m) \) such that for all \( x_n \)

\[
\sum_{i=1}^{m} w_i k(x_i, x_n) > 0
\]

• Every \( x_n \) lies within ‘distance’ \( R \) of at least one \( x_i \) for which \( w_m > 0 \)

• recall the empirical kernel map

\[ \Phi_m(x) = (k(x_1, x), \ldots, k(x_m, x)) \]
• seek solutions with few positive $w_m$ by solving the optimization problem:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_q$$

subject to $\mathbf{w}^\top \Phi_m(x_n) \geq 1$ for all $x_n \in X$

• Ideally, we would choose $q = 0$, since $\|\mathbf{w}\|_0$ counts the non-zero coefficients

• But $q = 1$ leads to a tractable *linear programming* problem

• Penalizers of the form $\|\mathbf{w}\|_1$ generally lead to sparse solutions
Practicalities

• Actual penalty used:

\[ \sum_{m=1}^{m} \frac{|w_m|}{c_m} \]

\[ c_m = \sum_n k(x_m, x_n) \] the number of examples in the support of \( k(x_m, x) \)

– this improves sparsity without affecting the constraints

• perform a final ‘pruning’ step since symmetries in many tasks still give a number of superfluous vectors

– a consequence of using the \( q = 1 \) rather than \( q = 0 \) penalty

– typically, this step removes a further 1%–5% of vectors

B. Schölkopf, Canberra, February 2006
$R=0.1 \quad M=42$
\( R = 0.4 \quad M = 4 \)
Application to Block Coding of Images

- Popular use of conventional VQ

- Example 384 × 256 image:

- Split into 8 × 8 blocks

- $X$ comprises $m = 1536$ examples of 192-dimensional vectors (64 × 3 colours)
Original Image (288KB)

LP–VQ reconstruction with $R=200$, 144KB (50%)

LP–VQ reconstruction with $R=500$, 33KB (12%)

LBG reconstruction, 33KB (12%)

B. Schölkopf, Canberra, February 2006
## Image Statistics

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Ratio</th>
<th>$R$</th>
<th>$M$</th>
<th>$E_{max}$</th>
<th>$E_{rms}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>288KB</td>
<td>100%</td>
<td>0</td>
<td>1536</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LP-VQ Reconstruction</td>
<td>144KB</td>
<td>50%</td>
<td>200</td>
<td>757</td>
<td>199.9</td>
<td>88.7</td>
</tr>
<tr>
<td>LP-VQ Reconstruction</td>
<td>33KB</td>
<td>12%</td>
<td>500</td>
<td>170</td>
<td>499.5</td>
<td>283.8</td>
</tr>
<tr>
<td>LBG Reconstruction</td>
<td>33KB</td>
<td>12%</td>
<td>-</td>
<td>170</td>
<td>816.4</td>
<td>229.8</td>
</tr>
</tbody>
</table>

B. Schölkopf, Canberra, February 2006
Discussion

• Complementary approach to standard VQ
• Useful where:
  – a ‘genuine’ $R$ exists
  – ‘outliers’ must be accurately coded
  – prototypes must be representative of data
  – as an initialiser for standard VQ
• Need not be a vector space as long as $D[v, x_n]$ defined
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 [13]
  - kernels for discrete objects [30, 78, 40, 18, 74]
  - kernels based on generative models [34, 61, 68]
  - local kernels [e.g., 84]
  - complex kernels from simple ones [30, 2], global kernels from local ones [38]
  - functional calculus for kernel matrices [60]
  - model selection, e.g., via alignment [17]
  - kernels for dimensionality reduction [27]
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, Canberra, February 2006