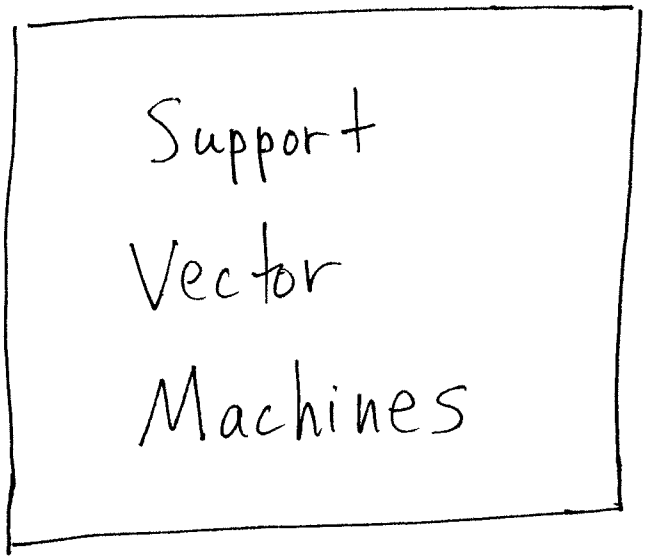

Support
Vector
Machines

# Support Vector Machines

A support vector machine is an extension of an optimal soft margin classifier that allows for nonlinear decision boundaries.

The SVM relies on three key concepts

- the dual QP for the soft margin problem.
- mapping patterns into a high-dimensional feature space
- kernels and the "kernel trick"

# The Soft Margin Dual

Recall the primal QP for the optimal soft-margin hyperplane:

$$\min_{(w,b,\xi)} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i (w^T x_i + b) \geq 1 - \xi_i \,, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0 \,, \quad i = 1, \ldots, n$$

The variables $\xi_i$ are "slack" variables and are used to extend the max-margin principle to nonseparable data.

The Lagrangian is

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C \cdot \sum_{i=1}^{n} \xi_i$$

$$+ \sum_{i=1}^{n} \alpha_i \left(1 - \xi_i - y_i (w^T x_i + b)\right) - \sum_{i=1}^{n} \beta_i \xi_i$$

The saddle point condition implies

$$\frac{\partial L}{\partial w}(w^*, b^*, \xi^*, \alpha^*, \beta^*) = w^* - \sum_{i=1}^{n} \alpha_i^* y_i x_i = 0$$

$$\frac{\partial L}{\partial b}(w^*, b^*, \xi^*, \alpha^*, \beta^*) = - \sum \alpha_i^* y_i = 0$$

$$\frac{\partial L}{\partial \xi_i}(w^*, b^*, \xi^*, \alpha^*, \beta^*) = C - (\alpha_i^* + \beta_i^*) = 0$$

Substituting

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

into the Lagrangian gives the following dual QP

$$\max_{(\alpha,\beta)} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i + \beta_i = C \qquad i = 1, \dots, n$$

$$\alpha_i \geq 0, \ \beta_i \geq 0, \qquad i = 1, \dots, n$$

We can eliminate $\beta$ and finally obtain

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \sum \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \qquad i = 1, \dots, n.$$

The dual has a number of desirable properties:

1. We may obtain $w^*, b^*$ from $\alpha^*$.

   (a) From the KKT conditions we have

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

   $\Rightarrow$ the optimal normal vector is a linear combo. of data points

   (b) Recovering $b^*$ is a little less obvious. We'll return to this later.

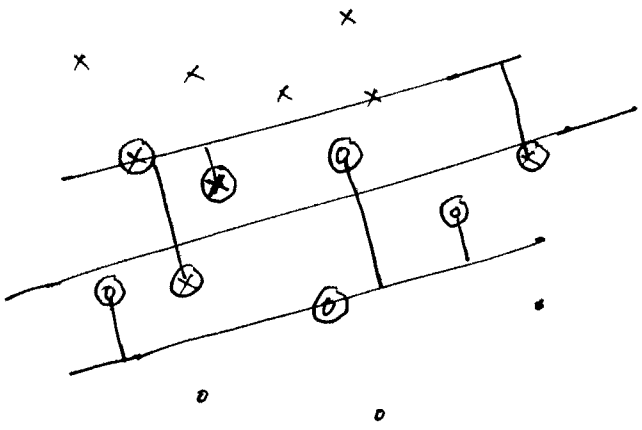
2. From the KKT conditions we have

$$\alpha_i^* \cdot \left(1 - \xi_i^* - y_i (w^{*T} x_i + b^*)\right) = 0 \quad \forall i.$$

Recall that $x_i$ for which

$$y_i (w^{*T} x_i + b^*) = 1 - \xi_i^*$$

are called <u>support vectors</u>. These are the points that are on or inside the margin of separation



By the KKT condition, either $x_i$ is a support vector, or $\alpha_i^* = 0$

extremely important fact!

Conclusion: We may write

$$w^* = \sum_{\substack{\text{support} \\ \text{vectors}}} \alpha_i^* y_i x_i$$

It has been widely demonstrated empirically that only a small fraction of the training patterns are support vectors (those that are closest to the decision boundary).

Therefore, the soft-margin criterion produces a hyperplane with a sparse representation. This is advantageous for efficient storage and evaluation.

3. If $\alpha_i^* < C$, then $\xi_i^* = 0$.

To see this recall the Lagrange multipliers $\beta_i$ corresponding to the constraints $\xi_i \geq 0$. By the KKT conditions, we have that

$$\beta_i^* \cdot \xi_i^* = 0.$$

Since $\alpha_i^* + \beta_i^* = C$, the claim follows.

Exercise | Suggest a procedure for determining $b^*$ using 2. and 3. above.

1 (b) (revisited)

If $0 < \alpha_i^* < C$, then

$$y_i (w^{*T} x_i + b^*) = 1$$

$$\Rightarrow b^* = y_i - w^{*T} x_i$$

flawed logic. Overfitting occurs when $C$ large

Comments

(i) Such an $i$ should always exist unless $C$ is extremely small, in which case you are probably overfitting.

(ii) Since numerical errors may affect the estimation of $w^*$, it is advisable to average the above procedure over all $i$ such that $0 < \alpha_i^* < C$.

4. The dual QP and the final classifier only involve the data through <u>inner products</u>

- $$\max_{\alpha} \quad -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \sum \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \quad \forall i$$

- $$w^{*T} x + b^* = \sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x \rangle + b^*$$

This fact will be critical later...

# Non linear Classification

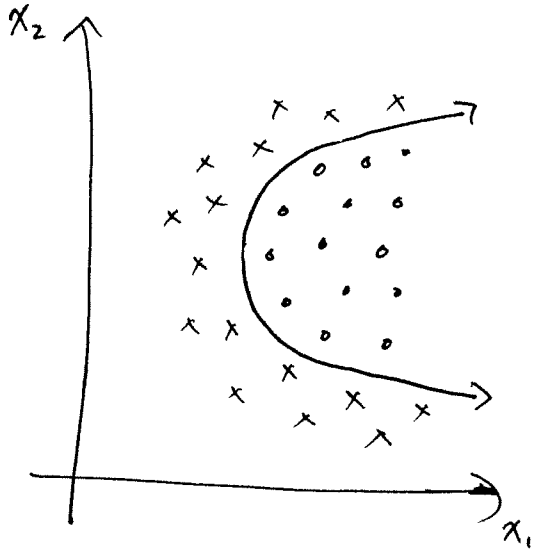
Non linear classification using the max-margin principle is achieved through two ideas:

A. Map patterns into high-dimensional feature space

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \longmapsto \underline{\Phi}(x) = \begin{bmatrix} \varphi_1(x) \\ \vdots \\ \varphi_m(x) \end{bmatrix}$$

where $m \gg d$ and $\varphi_j : \mathbb{R}^d \to \mathbb{R}$ are <u>non linear</u>

B. Build a <u>linear</u> classifier in this feature space, which induces a non linear classifier in the original space.




$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$$

To compute the optimal normal vector $w \in \mathbb{R}^m$, we solve

$$\max_{\alpha} \; -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle + \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C, \quad i = 1, \dots, n$$

and set

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i \Phi(x_i)$$

In fact, we don't need to compute $w^*$; the final classifier is

$$f(x) = \text{sign}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i \langle \Phi(x), \Phi(x_i) \rangle + b^* \right\}$$

Again, $b^*$ can be found from

$$b^* = y_i - \sum_{j=1}^{n} \alpha_j^* y_j \langle \Phi(x_i), \Phi(x_j) \rangle$$

where $i$ is such that $0 < \alpha_i^* < C$.

Again, the transformed patterns are only involved through inner products.

Key property: The size of the dual (i.e., the number of variables) depends only on $n$, the sample size, and not on $m$.

There are two potential drawbacks of this approach to nonlinear classification:

(i) overfitting may occur

(ii) the computational burden of computing $\langle \Phi(x), \Phi(x') \rangle$ many times. ($O(n^2)$ times to be precise)

These two concerns, especially the second, can be addressed with the help of kernels

## Kernels

Given a feature map $\Phi$, we can define the function

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

As we have seen, the soft-margin dual and the final nonlinear classifier can be expressed in terms of $k$.

Wouldn't it be great if $k(x, x')$ had a nice simple form?

Example | Consider the function

$$k(x, x') = (x \cdot x')^p$$

$$\mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$$

If $d=2$, $p=2$, we have

$$k(x, x') = \left( [x_1 \;\; x_2] \cdot \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \right)^2$$

$$= \left( \sum_{i=1}^{2} x_i x_i' \right)^2$$

$$= \left( \sum_{i=1}^{2} x_i x_i' \right) \left( \sum_{j=1}^{2} x_j x_j' \right)$$

$$= \sum_{i=1}^{2} \sum_{j=1}^{2} (x_i x_j) \cdot (x_i' x_j')$$

$$= x_1^2 \cdot (x_1')^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 \cdot (x_2')^2$$

What feature mapping does this correspond to?

$$\Rightarrow \quad \Phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{bmatrix}$$

Now suppose $d$ is arbitrary, and $p = 2$.

$$k(x, x') = \langle x, x' \rangle^2$$

$$= \left( \sum_{i=1}^{d} x_i x_i' \right)^2$$

$$= \left( \sum_{i=1}^{d} x_i x_i' \right) \left( \sum_{j=1}^{d} x_j x_j' \right)$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{d} (x_i x_j)(x_i' x_j')$$

What is the dimension of the corresponding feature space?

$$\Phi(x) = \Big[ \underbrace{x_1^2, x_2^2, \dots, x_d^2}_{d}, \underbrace{\sqrt{2}\, x_1 x_2, \sqrt{2}\, x_1 x_3, \dots \sqrt{2}\, x_{d-1} x_d}_{\frac{d(d-1)}{2}} \Big]^T$$

**Exercise** Describe the feature space when $d$ and $p$ are arbitrary.

Solution

$$k(x, x') = \left( \sum_{i=1}^{d} x_i x_i' \right)^p$$

Consider $d = 2, \; p = 3$

$$\Rightarrow \quad k(x, x') = (x_1 x_1' + x_2 x_2')^3$$

$$= x_1^3 \cdot (x_1')^3 + x_2^3 (x_2')^3$$

$$+ 3(x_1^2 x_2) \cdot [(x_1')^2 x_2'] + 3(x_1 x_2^2) \cdot [x_1' (x_2')^2]$$

$$= \sum_{j=0}^{3} \binom{3}{j} (x_1 x_1')^j (x_2 x_2')^{3-j}$$

$$= \sum_{j=0}^{3} \binom{3}{j} (x_1^j x_2^{3-j}) [(x_1')^j (x_2')^{3-j}]$$

↑ binomial coefficients

$$\Rightarrow \quad \Phi(x) = \left[ x_1^3, \; \sqrt{3} x_1^2 x_2, \; \sqrt{3} x_1 x_2^2, \; x_2^3 \right]$$

For arbitrary $d, p$, use <u>multinomial coefficients</u>:

$$k(x, x') = \left( \sum_{i=1}^{d} x_i x_i' \right)^p$$

$$= \sum_{\substack{(j_1, \dots, j_d) \\ \sum j_d = p}} \binom{p}{j_1\ j_2 \cdots j_d} (x_1 x_1')^{j_1} \cdots (x_d x_d')^{j_d}$$

$$\Rightarrow \quad \Phi(x) \to \left[ \cdots , \sqrt{\binom{p}{j_1 \cdots j_d}}\, x_1^{j_1} \cdots x_d^{j_d}, \cdots \right]^T$$

$$\binom{d+p-1}{p}$$

In general, the feature space consists of all <u>monomials</u> of degree $p$, with the weight determined by the structure of the exponents.

$k(x, x) = \langle x \cdot x' \rangle^p$ is called a <u>polynomial kernel</u>

**Definition** A **kernel** is any mapping

$$k: \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}, \quad (x, x') \longmapsto k(x, x').$$

If there exists a feature space $\mathcal{H}$ and a feature map $\Phi: \mathbb{R}^d \to \mathcal{H}$ such that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$ then $k$ is an **inner product kernel**.

We have seen one example: polynomial kernels. Are there others? What are sufficient conditions of $k$ for it to be an inner product kernel?

It turns out there are two different characterizations:

- Mercer kernels
- Positive semi-definite kernels.

Note: these two approaches may associate a kernel to two distinct feature maps/spaces.

# Mercer kernels

Consider the space $L^2(\mathcal{X})$, the set of square-integrable functions on $\mathcal{X} \subseteq \mathbb{R}^d$, a compact (closed and bounded) subset of $\mathbb{R}^d$.

A kernel $k$ defines an integral operator

$T_k : L^2(\mathcal{X}) \longrightarrow L^2(\mathcal{X})$ by

$$(T_k \psi)(x) = \int_{\mathcal{X}} k(x, x') \psi(x') \, dx'.$$

An eigenfunction of $T_k$ satisfies

$$(T_k \psi)(x) = \lambda \psi(x)$$

for some $\lambda \in \mathbb{R}$.

**Theorem** (Mercer) If $k$ is continuous, symmetric, and satisfies

$$\int_{\underline{X} \times \underline{X}} k(x, x') \psi(x) \psi(x') \, dx \, dx' \geq 0$$

for all $\psi \in L^2(\underline{X})$, then

(a) there exist eigenfunctions $\psi_1, \psi_2, \ldots$ and eigenvalues $\lambda_1 \geq 0$, $\lambda_2 \geq 0, \ldots$ such that

$$k(x, x') = \sum_i \lambda_i \psi_i(x) \psi_i(x')$$

(b) the convergence in (a) is uniform

(c) the eigenfunctions are orthonormal in $L^2(\underline{X})$

Conclusion:

We may view $L^2(X)$ as the feature space and

$$\Phi(x) = [\dots, \underbrace{\sqrt{\lambda_j}\,\psi_j(x)}_{=:\phi_j(x)}, \dots]$$

Note: The feature space may be infinite.

Kernels satisfying the hypothesis of Mercer's Theorem are called Mercel kernels.

# PSD kernels

We say a symmetric kernel is positive semi-definite if, for all $n$ and for all $x_1, \ldots, x_n$, the matrix

$$[k(x_i, x_j)]_{i,j}$$

is positive semi-definite.

**Theorem.** Let $k$ be a symmetric kernel. Then $k$ is an inner product kernel iff $k$ is a PSD kernel.

**Proof:** One direction is easy. Which is it?

The easy direction is to show IP $\Rightarrow$ PSD.

Assume $\exists\ \mathcal{H}$, $\Phi : \mathbb{R}^d \to \mathcal{H}$ and an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$$

for all $x, x'$.

Now let $n$, $x_1, \ldots, x_n \in \mathcal{X}$, $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ be arbitrary. We have

$$\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)$$

$$= \sum \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle$$

$$= \left\langle \sum \alpha_i \Phi(x_i), \sum \alpha_i \Phi(x_i) \right\rangle$$

since $\langle \cdot, \cdot \rangle$ is bilinear

$$\geq 0$$

since $\langle \cdot, \cdot \rangle$ is pos. def.

Now assume $k$ is PSD. The feature map associated with $k$ will be

$$\Phi(x) = \text{the function that maps } z \text{ to } k(z, x)$$

$$=: k(\cdot, x)$$

We will show

(a) the image of $\Phi$ is a vector space

(b) it has a dot product

(c) the dot product satisfies

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

(a) Define the vector space

$$\mathcal{H} = \left\{ \psi(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \quad n \geq 1, \right.$$
$$\left. x_1, \ldots, x_n \in \mathcal{X}, \alpha_1, \ldots, \alpha_n \in \mathbb{R} \right\}$$

(b) If $\psi, \psi' \in \mathcal{H}$, where

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

$$\psi'(\cdot) = \sum_{j=1}^{n'} \alpha'_j k(\cdot, x'_j)$$

then define

$$\langle \psi, \psi' \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \alpha'_j k(x_i, x'_j)$$

we must show $\langle \cdot, \cdot \rangle$ is

(i) well-defined
(ii) symmetric
(iii) bilinear
(iv) positive definite.

} easy

(i) We must show that $\langle \cdot, \cdot \rangle$ does not depend on the choice of expansion coefficients. Note that

$$\langle \psi, \psi' \rangle = \sum_{j=1}^{n} \alpha_j' \psi(x_j')$$

so it is independent of the expansion of $\psi$.

Similarly for $\psi'$!

(iv) $$\langle \psi, \psi \rangle = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

since $k$ is PSD. It remains to show

$\langle \psi, \psi \rangle = 0 \Rightarrow \psi \equiv 0$.

Exercise | Show that for any $\psi$,

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

**Lemma**: If $k$ is PSD, then $|k(x_1, x_2)|^2 \leq k(x_1, x_1) \cdot k(x_2, x_2)$.

**Pf**: Let $K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$ where $K_{ij} = k(x_i, x_j)$, $i, j = 1, 2$.

Since $k$ is a PSD kernel, $K$ is a PSD matrix

$\Rightarrow$ its eigenvalues are $\geq 0$

$\Rightarrow$ its determinant is $\geq 0$

$\Rightarrow K_{11} \cdot K_{22} - K_{21} \cdot K_{21} \geq 0$.

Since $K_{12} = K_{21}$ (by symmetry), the result follows.

**Observation**: The inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is itself a PSD kernel (verify as an exercise).

**Remark** The lemma is analogous to the Cauchy-Schwarz inequality.

It follows (from the lemma and the observation) that

$$|\langle k(\cdot, x), \psi \rangle|^2 \leq k(x,x) \cdot \langle \psi, \psi \rangle$$

Hence, if $\langle \psi, \psi \rangle = 0$, then (by the exercise)

$$|\psi(x)|^2 = |\langle k(\cdot, x), \psi \rangle|^2 = 0 \qquad \forall x$$

$$\Longrightarrow \psi \equiv 0.$$

(c). Applying the exercise with $\psi = k(\cdot, x')$, we obtain

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

Actually, this follows trivially from the definition of $\langle \cdot, \cdot \rangle$.

## Examples of inner-product kernels

1. Homogeneous polynomial kernel

$$k(x, x') = \langle x \cdot x' \rangle^p, \qquad p = 1, 2, \ldots$$

2. Inhomogeneous polynomial kernel

$$k(x, x') = (\langle x \cdot x' \rangle + c)^p, \qquad p = 1, 2, \ldots \quad c > 0$$

$\exists \rightarrow$ monomial of degree $\leq p$

3. Gaussian kernel (with isotropic covariance)

$$k(x, x') = (2\pi\sigma^2)^{-\frac{d}{2}} \exp\left\{-\frac{\|x - x'\|^2}{2\sigma^2}\right\}$$

→ constant can be dropped

→ also called radial basis function (RBF) kernel

4. Many others . . see Schölkopf and Smola, Learning with Kernels, MIT Press, 2002

# Support Vector Machines

## With Offset

$$f(x) = \text{sign}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i k(x, x_i) + b^* \right\}$$

where $\alpha^*$ is the solution of

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0, \quad i = 1, \ldots, n$$

most $\alpha_i^* = 0$

## Without Offset

$$f(x) = \text{sign}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i k(x, x_i) \right\}$$

where $\alpha^*$ is the solution of

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{s.t.} \quad C \geq \alpha_i \geq 0, \quad i = 1, \ldots, n$$

The offset is usually needed for polynomial kernels to perform well, but not for the Gaussian kernel.

## Connection to Parzen window

Recall the Parzen window classifies:

$$f(x) = \text{sign}\left\{ \frac{n_+}{n}\frac{1}{n_+}\sum_{y_i=1} k(x,x_i) - \frac{n_-}{n}\frac{1}{n_-}\sum_{y_i=-1} k(x,x_i) \right\}$$

$$= \text{sign}\left\{ \frac{1}{n}\sum_{i=1}^{n} y_i k(x,x_i) \right\}$$

where $k(x,x')$ is the Gaussian kernel.

This has the form of a Gaussian-kernel SVM without offset and with

$$\alpha_i^* = \frac{1}{n}.$$

Since the SVM is more flexible in that the $\alpha_i^*$ are determined adaptively, it clearly has the potential to outperform the Parzen window.

picture taken from Scholkopf and Smola.

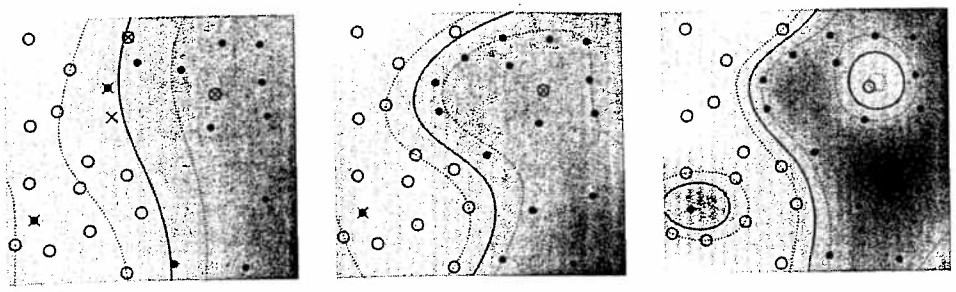


**Figure 7.10** 2D toy example of a binary classification problem solved using a soft margin SVC. In all cases, a Gaussian kernel (7.27) is used. From left to right, we decrease the kernel width. Note that for a large width, the decision boundary is almost linear, and the data set cannot be separated without error (see text). Solid lines represent decision boundaries; dotted lines depict the edge of the margin (where (7.34) becomes an equality with $\xi_i = 0$).

## Summary of SVMs

- Optimal soft margin hyperplane in high-dimensional feature space $\Rightarrow$ nonlinear decision in original space
- Feature mapping is implicit. Only inner products of features are needed, and these are computed using inner-product kernels.
- Solution is sparse (few support vectors)
- Solution obtained by solving a convex quadratic program. Efficient implementations exist.

# Kernel Fisher Discriminant Analysis

Recall Fisher discriminant analysis (FDA):

A hyperplane is defined by a normal vector $w$

maximizing

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

$S_B$ ← between-class scatter

$S_W$ ← within-class scatter

where

$$S_B = (m_- - m_+)(m_- - m_+)^T$$

and

$$S_W = \sum_{q=\pm} \sum_{i \in I_q} (x_i - m_q)(x_i - m_q)^T$$

$I_q$: index set for class $q$:

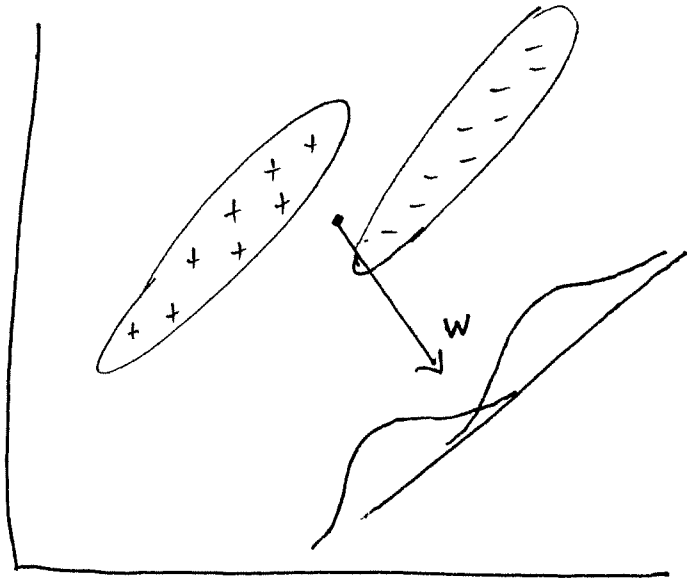
$$I_q = \{i : y_i = q\}$$

$S_W \propto$ sample covariance matrix

The solution is

$$w = S_w^{-1}(m_+ - m_-)$$

(same as LDA)




## FDA in Feature Space

Suppose we want to transform $x$ via

$$\Phi : \mathbb{R}^d \longrightarrow \mathcal{H}$$

and perform FDA in $\mathcal{H}$.

Assume the solution $w \in \mathcal{H}$ has an expansion

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

for some $\alpha_i \in \mathbb{R}$.

Our goal is to express $J(w)$ in terms of

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

**Exercise** Express $J(w) = J(\alpha)$ in terms of $\alpha$ and $k(x, x')$. That is, plug in $w = \sum \alpha_i \Phi(x_i)$ to $J(w)$ and simplify.

Solution) The following notation is useful:

$$n_+ = |I_+|, \quad n_- = |I_-|$$

$$K = [k(x_i, x_j)]_{i,j=1}^{n}$$

$\mu_+$ = average of columns of $K$ corresponding to $y_i = +1$

$\mu_-$ = " " " " " " " $y_i = -1$

$$M = K \cdot K^T - \sum_{q=\pm} n_q \mu_q \mu_q^T$$

Then

$$J(\alpha) = \frac{\left(\sum \alpha_i \Phi(x_i)\right)^T (m_- - m_+)(m_- - m_+)^T \left(\sum \alpha_i \Phi(x_i)\right)}{\left(\sum \alpha_i \Phi(x_i)\right)^T \left[\sum_{q=\pm} \sum_{i \in I_q} (\Phi(x_i) - m_q)(\Phi(x_i) - m_q)^T\right] \left(\sum \alpha_i \Phi(x_i)\right)}$$

Let's look at

$$(m_- - m_+)^T \left(\sum \alpha_i \Phi(x_i)\right)$$

$$= \left(\frac{1}{n_-} \sum_{i \in I_-} \Phi(x_i) - \frac{1}{n_+} \sum_{i \in I_+} \Phi(x_i)\right)^T \left(\sum_{i=1}^{n} \alpha_i \Phi(x_i)\right)$$

$$=$$

$$= \alpha^T (\mu_- - \mu_+)$$

$$\Rightarrow \text{numerator} = \left[\alpha^T (\mu_- - \mu_+)\right]^2.$$

A slightly more involved calculation shows

$$\text{denominator} = \alpha^T M \alpha$$

Conclusion

$$J(\alpha) = \frac{\left[\alpha^T (\mu_- - \mu_+)\right]}{\alpha^T M \alpha}$$

depends only on the kernel $k(x, x')$.

Therefore, we may apply the "kernel trick" to compute $w$ implicitly.

The final classifier is given by

$$f(x) = \text{sign}\left\{ \sum_{i=1}^{n} \alpha_i k(x, x_i) + b \right\}$$

where $b$ is determined separately.

The optimization problem may be converted to a convex QP and solved efficiently.

The final decision rule is nonlinear in the original space.