

Data Mining

Lluís A. Belanche
belanche@lsi.upc.edu

Soft Computing Research Group

Dept. de Llenguatges i Sistemes Informàtics (Software department)

Universitat Politècnica de Catalunya

2011-2012

Data Mining (Part II)

Contents

1. Association rules
2. Bayesian decision theory (I)
3. Bayesian decision theory (II)
4. Naïve Bayes
5. Nearest neighbours
6. Multilayer perceptrons
7. Support Vector Machines
8. Ensemble learning

Support Vector Machines

Preliminaries

- Criterion for building a two-class classifier:

Maximize the width of the margin between the classes

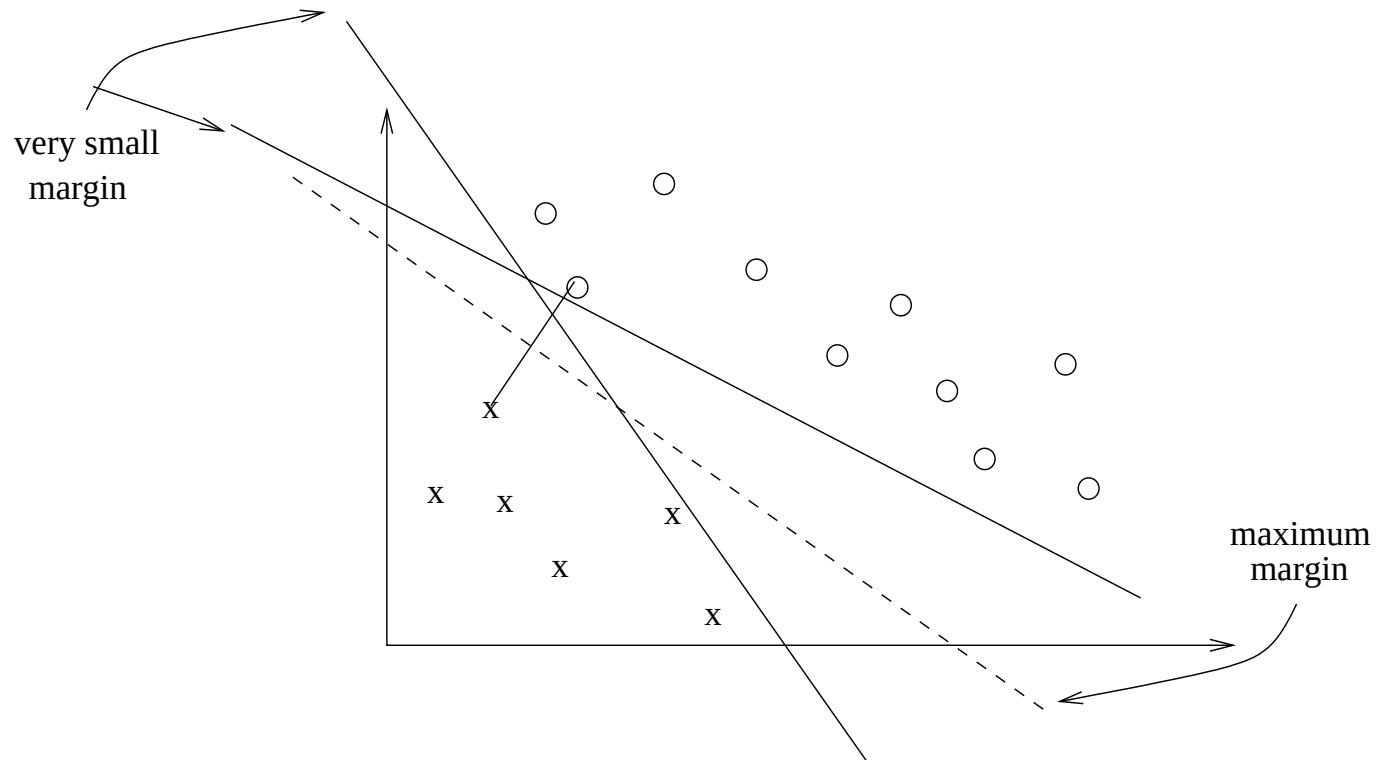
- margin = empty area around the decision boundary, defined by the distance to the nearest training examples

These examples are called the support vectors

- Goal: find the linear decision boundary that maximizes this margin

Support Vector Machines

Preliminaries



Which solution is more likely to lead to better generalization?

Support Vector Machines

Preliminaries

Working Hypothesis (intuition):

The larger the margin, the better the generalization

Goal: find the separating hyperplane with the largest margin

Support Vector Machines

Preliminaries

For a two-class classifier, the *VC dimension* is the maximal number k of points that can be separated in all possible 2^k ways (*shattered*) by using functions representable by the classifier.

- Note it is *sufficient* that one set of k points exists that can be shattered for the VC dimension to be at least k
- If the VC dimension of a class is k , this means there is at least one set of k points that can be shattered by one member of the class. It does not mean that every set of k points can be shattered.

Support Vector Machines

Preliminaries

- Why is the VC dimension relevant? Because it provides bounds on the expected risk as a function of the empirical risk and the number of available examples. Indeed, with probability $1 - \eta$:

$$E_{true}(f) \leq E_{emp}(f) + \sqrt{\frac{k(\ln(2m/k) + 1) - \ln(\eta/4)}{m}}$$

where k is the VC dimension of f , and $m > k$ is the training set size.

Support Vector Machines

Preliminaries

- The VC dimension cannot be accurately estimated for non-linear models such as neural networks
- The VC dimension may be infinite (e.g., as in nearest neighbors), requiring infinite amount of data
- The upper bound may sometimes be trivial (e.g., larger than one)
- The set of linear separations in d dimensions has VC dimension $d + 1$

Support Vector Machines

More than an intuition

Theorem. Consider canonical hyperplanes $g(x) = \text{sgn}(\langle w, x \rangle + b)$ w.r.t. a set of training patterns X (dimension d).

The subclass of linear classifiers with margin m has VC dimension k bounded by

$$k \leq \min \left(\left\lceil \frac{R^2}{m^2} \right\rceil, R \right) + 1$$

where R is the radius of the smallest sphere containing X .

Support Vector Machines

Formalisation

$\mathbf{x}_1, \dots, \mathbf{x}_m$ set of training patterns (dimension d) in two classes ω_1, ω_2
 $y_i = \pm 1$, with $y_i = +1$ if $\mathbf{x}_i \in \omega_1$ and $y_i = -1$ if $\mathbf{x}_i \in \omega_2$

We set up

1. An affine function in $\mathbf{x}$ -space: $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$
2. A linear discriminant in $\mathbf{x}$ -space: $\text{sgn}(g(\mathbf{x}))$

We would like, for all $1 \leq i \leq m$:

$\mathbf{w}^T \mathbf{x}_i + b > 0$, for $\mathbf{x}_i \in \omega_1$ ($y_i = +1$)

$\mathbf{w}^T \mathbf{x}_i + b < 0$, for $\mathbf{x}_i \in \omega_2$ ($y_i = -1$) or, equivalently, $y_i(\mathbf{w}^T \mathbf{x}_i + b) > 0$

Support Vector Machines

Formalisation

- Given the plane $\pi : g(\mathbf{x}) = 0$ ($\mathbf{w}^T \mathbf{x} + b = 0$), we have:

$$\text{distance } d(\mathbf{x}, \pi) = \frac{|g(\mathbf{x})|}{\|\mathbf{w}\|} = m(\mathbf{x}) \quad \underline{\text{margin of } \mathbf{x}}$$

- The *support vectors* (SV) are those $\{\mathbf{x}_i / m(\mathbf{x}_i) \leq m(\mathbf{x}_k), \forall 1 \leq k \leq m\}$.
- The optimal separating hyperplane (OSH) is the one that maximizes the margin for linearly separable data:

$$\max_{\mathbf{w}, b} \min_{1 \leq i \leq m} \frac{y_i (\mathbf{w}^T \mathbf{x}_i + b)}{\|\mathbf{w}\|} \quad \text{subject to } y_i (\mathbf{w}^T \mathbf{x}_i + b) > 0, \quad 1 \leq i \leq m$$

Support Vector Machines

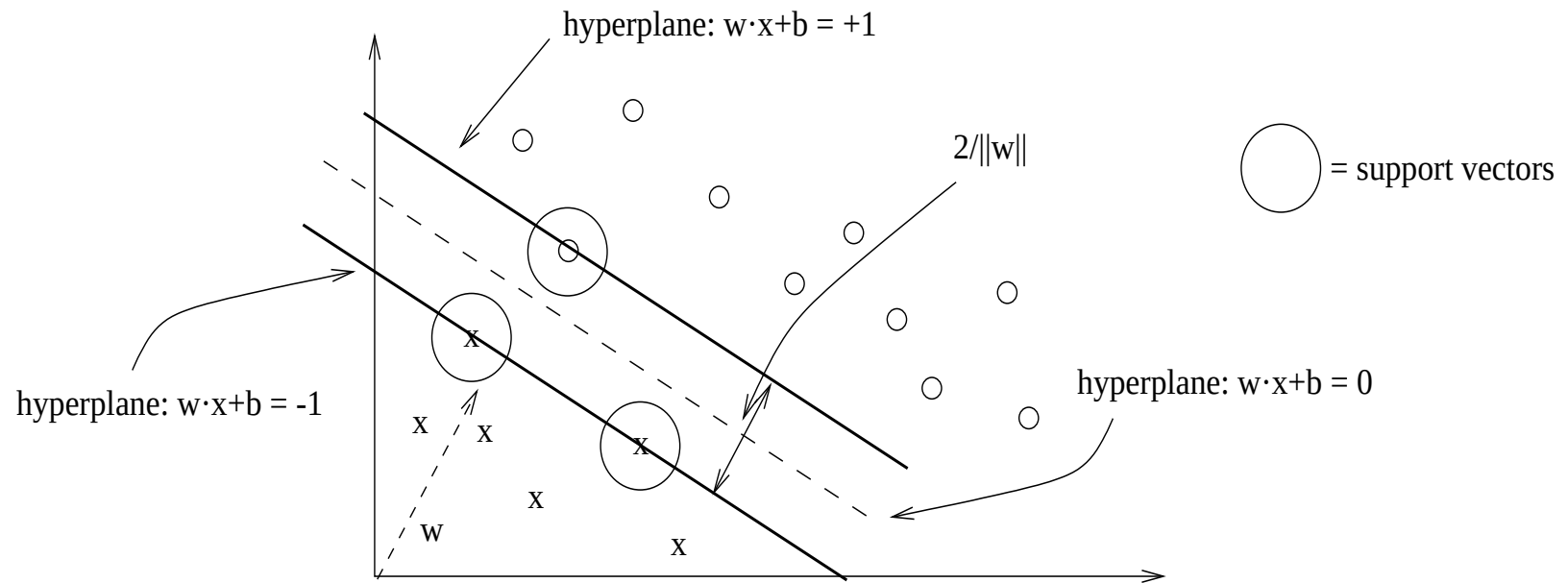
Formalisation

- Rescaling w, b such that $|w^T x + b| = 1$ for the points closest to the hyperplane (the SVs), we obtain $|w^T x + b| \geq 1$.
- The *support vectors* are now those $\{x_i / |w^T x_i + b| = 1\}$.
- The margin of the OSH can be written as twice the distance of any support vector x_{SV} to the OSH:

$$\text{margin} = 2 d(x_{SV}, \pi) = \frac{2}{\|w\|}, \text{ since } |g(x_{SV})| = 1$$

Support Vector Machines

Geometrical view



Support Vector Machines

Formalisation

If we work with these *canonical hyperplanes*, the new OSH is:

$$\max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} / \min_{1 \leq i \leq m} y_i (\mathbf{w}^T \mathbf{x}_i + b) = 1 \right\}$$

subject to $y_i (\mathbf{w}^T \mathbf{x}_i + b) > 0, \quad 1 \leq i \leq m.$

which amounts to:

$$\max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} / y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \right\}, \quad 1 \leq i \leq m$$

Support Vector Machines

The Support Vectors

- It will be shown that the solution for w can be expressed as an expansion

$$w = \sum_{i=1}^m v_i x_i$$

in terms of that subset of training data corresponding to the support vectors (vectors lying exactly at a margin distance).

Support Vector Machines

The Support Vectors

- The discriminant function will be written

$$f_{\text{SVM}}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b) = \text{sgn} \left(\sum_{i=1}^m v_i \mathbf{x}^T \mathbf{x}_i + b \right)$$

- The decision rule is the usual for linear discriminants:

$$\begin{cases} f_{\text{SVM}}(\mathbf{x}) > 0 & \implies \mathbf{x} \in \omega_1 \\ f_{\text{SVM}}(\mathbf{x}) < 0 & \implies \mathbf{x} \in \omega_2 \end{cases}$$

- The support vectors are those training samples that define the optimal separating hyperplane and are the most difficult patterns to classify. In a sense, they are the **most informative** patterns

Support Vector Machines

Formulation

$$\begin{array}{ll}\text{minimize} & \frac{1}{2} \|w\|^2 \\ \text{subject to} & y_i (w^T x_i + b) \geq 1, \quad 1 \leq i \leq m\end{array}$$

This is solved (numerically) by QP techniques:

- Quadratic (therefore convex) function subject to linear constraints
- Unique solution (or set of equivalent ones)
- Therefore, NO LOCAL MINIMA

Support Vector Machines

Margin violations

- In practice, we allow small margin violations ε_i , for each x_i :

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}\|w\|^2 + C \sum_{i=1}^m \varepsilon_i \\ \text{subject to} & y_i (w^T x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \quad 1 \leq i \leq m\end{array}$$

- The ε_i are *slack* variables, leading to a *soft margin* ($\varepsilon_i > 0$ implying the margin of x_i is less than 1).
- Classic QP problem with linear constraints.

Support Vector Machines

Lagrangian form (primal)

- We first construct the Lagrangian:

$$\mathcal{L} = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^m \alpha_i \left\{ y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \varepsilon_i \right\} + C \sum_{i=1}^m \varepsilon_i - \sum_{i=1}^m \mu_i \varepsilon_i$$

- Interpretation:
 - First term: “maximize the margin”
 - Second term: “classify the points correctly” (up to small violations)
 - Rest: “minimize the violations”

Support Vector Machines

Lagrangian form

$$\text{minimize} \quad \mathcal{L} \quad \text{subject to} \quad 0 \leq \alpha_i \leq C$$

- The $\alpha_i, \mu_i \geq 0$ are the *Lagrange multipliers* (auxiliary variables to cope with the constraints).
- The solution of this QP problem is the saddle point of $\mathcal{L}$.

Support Vector Machines

Lagrangian form

- The minimum is taken with respect to $\mathbf{w}, b$ and the maximum with respect to the α_i .
- The gradient of $\mathcal{L}$ with respect to $\mathbf{w}, b$ must vanish:

$$\frac{\delta \mathcal{L}}{\delta b} = \sum_{i=1}^m \alpha_i y_i = 0, \quad \frac{\delta \mathcal{L}}{\delta \mathbf{w}} = \mathbf{w} - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = 0, \quad \frac{\delta \mathcal{L}}{\delta \varepsilon_i} = C - \alpha_i - \mu_i = 0$$

$$\alpha_i \left\{ y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \varepsilon_i \right\} = 0 \text{ (KKT complementarity conditions)}$$

as stated previously, $v_i \equiv \alpha_i y_i > 0$ only for support vectors.

Support Vector Machines

Dual formulation

- The Lagrangian $\mathcal{L}$ is convex: its minimization is equivalent to the maximization of its dual problem $\mathcal{L}_D$:

$$\mathcal{L}_D = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\Rightarrow \text{maximize } \mathcal{L}_D \text{ subject to } 0 \leq \alpha_i \leq C, \sum_{i=1}^m \alpha_i y_i = 0$$

Support Vector Machines

Dual formulation

- Note how neither μ_i, ε_i appear in the dual form.
- Also, w has been incorporated in it.
- Maximization is only with respect to the α_i .

$$\text{maximize } \mathcal{L}_D = \langle \alpha, 1 \rangle - \frac{1}{2} \alpha^t Q \alpha, \quad Q = (q_{ij}), q_{ij} = y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to } \langle \alpha, y \rangle = 0, \quad 0 \leq \alpha \leq C1$$

Support Vector Machines

Dual formulation

A closer look at the KKT complementarity conditions:

- $\alpha_i = 0$ implies $y_i (\langle \mathbf{x}_i, \mathbf{x} \rangle + b) > 1$ and $\varepsilon_i = 0$ ($\mathbf{x}_i$ is not a SV)
- $\alpha_i \in (0, C)$ implies $y_i (\langle \mathbf{x}_i, \mathbf{x} \rangle + b) = 1$ and $\varepsilon_i = 0$ ($\mathbf{x}_i$ is a non-bound SV)
- $\alpha_i = C$ implies $y_i (\langle \mathbf{x}_i, \mathbf{x} \rangle + b) < 1$ and $\varepsilon_i > 0$ ($\mathbf{x}_i$ is a bound SV)

Support Vector Machines

Result

- The discriminant function is:

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

with $\alpha_i > 0$ *only* for the support vectors (for the rest, $\alpha_i = 0$).

- By setting $C = \infty$ we are in the *hard* margin case
 $\implies C$ represents a tradeoff between margin width and training error.
- $b = - \left(\sum_{i=1}^m \sum_{j=1}^m \alpha_i y_i \alpha_j y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right) / \left(\sum_{i=1}^m \alpha_i \right)$

Support Vector Machines

Generalization error

- Leave-one-out method

$m_s \leq m$ = number of support vectors

$$E_m[\text{error}] \leq \frac{E_m[m_s]}{m}$$

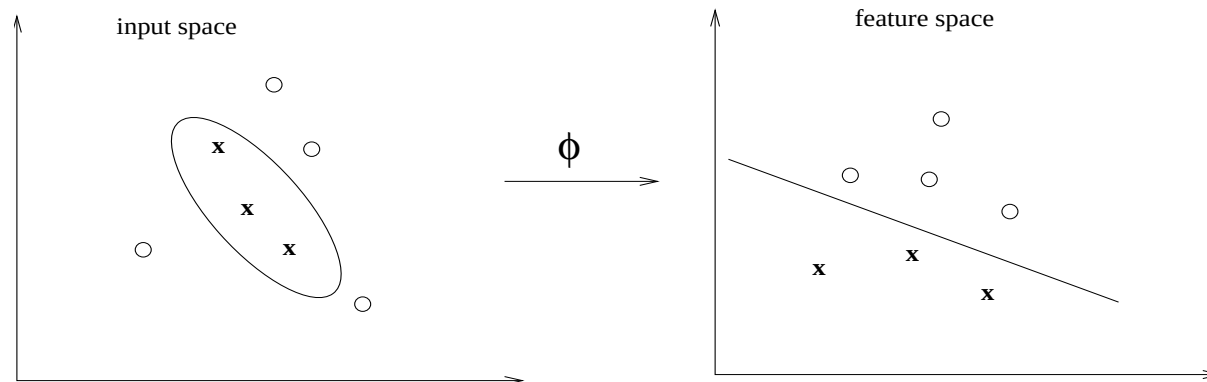
(the expectation is taken over all training sets of size m)

- The complexity of an SVM classifier depends on m_s , not on the dimension of the feature space.

Support Vector Machines

The Kernel Trick

Basic idea: map the input data into some other dot product space (called the feature space) via a non-linear mapping $\phi : \mathbb{R}^n \longrightarrow F$.



Support Vector Machines

The Kernel Trick

The kernel trick Let F be such that a dot product is defined $\longrightarrow$ the “trick” consists of performing the mapping and the dot product simultaneously by using a kernel function K of the form:

$$K(u, v) = \langle \phi(u), \phi(v) \rangle$$

By the use of K , it is possible to compute the hyperplane without explicitly carrying out the map into F .

Support Vector Machines

The Kernel Trick

- Suppose we take $K(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle^d$ (a simple choice).
- What is the underlying mapping ϕ here?
 $\implies$ Answer: this choice of kernel corresponds to a map leading into the space spanned by all products of exactly d dimensions of $\mathbb{R}^n$.
- Let us take, for instance, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, and take $d = 2$:

$$\begin{aligned} K(\mathbf{u}, \mathbf{v}) &= \langle \mathbf{u}, \mathbf{v} \rangle^2 = \left[\left\langle \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \right\rangle \right]^2 \\ &= (u_1 v_1 + u_2 v_2)^2 = (u_1 v_1)^2 + 2u_1 v_1 u_2 v_2 + (u_2 v_2)^2 \\ &= u_1^2 v_1^2 + (\sqrt{2} u_1 u_2)(\sqrt{2} v_1 v_2) + u_2^2 v_2^2 \\ &= \left\langle \begin{pmatrix} u_1^2 \\ \sqrt{2} u_1 u_2 \\ u_2^2 \end{pmatrix}, \begin{pmatrix} v_1^2 \\ \sqrt{2} v_1 v_2 \\ v_2^2 \end{pmatrix} \right\rangle = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle \end{aligned}$$

- Therefore, $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^t$; this K is a kind of polynomial kernel.

Support Vector Machines

Choices for the Kernel

Polynomial kernels (relation to GLDs)

$$K(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^d, \quad d \in \mathbb{N}$$

Gaussian kernels (relation to RBFNNs)

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{1}{2} \frac{\|\mathbf{u} - \mathbf{v}\|^2}{\sigma^2}\right) \quad \sigma^2 \in \mathbb{R}$$

Sigmoidal kernels (relation to MLPs)

$$K(\mathbf{u}, \mathbf{v}) = g(\alpha \langle \mathbf{u}, \mathbf{v} \rangle + \beta)$$

with g a sigmoidal (e.g., logistic, tanh, ...) and some choices for α, β .

Support Vector Machines

Characterization of Kernels

Every choice of K leading to a *positive semi-definite* matrix $\mathbf{K} = (k_{ij})$, where $k_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, leads to a map

$$\phi : X \longrightarrow F$$

such that $K(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle_F$ holds.

Support Vector Machines

SVM training

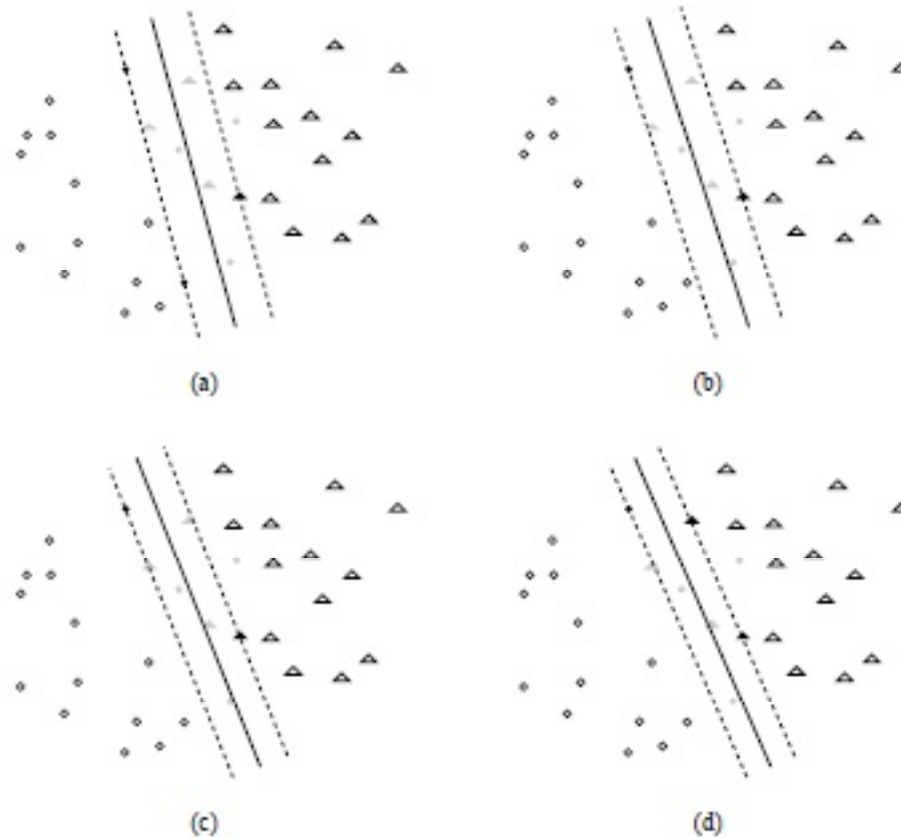
- Back to the hyperplane classifier, we substitute the $\mathbf{x}_i$ by its mapped vector $\phi(\mathbf{x}_i)$, then build the optimal hyperplane (maximum margin that classifies correctly) in F .
- The discriminant function is now:

$$f_{\text{SVM}}(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right)$$

- The QP problem is formulated exactly as before with $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$

Support Vector Machines

The role of the C parameter



OSH for four increasing values of C . Both the margin and the training error are non-increasing functions of C

Support Vector Machines

SVMs for regression

Here we choose the loss: $|y - f(\mathbf{x})|_\epsilon = \max(|y - f(\mathbf{x})| - \epsilon, 0)$

$$\text{minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m (\varepsilon_i + \varepsilon_i^*)$$

subject to $(\mathbf{w}^T \mathbf{x}_i + b) - y_i \leq \epsilon + \varepsilon_i, \quad y_i - (\mathbf{w}^T \mathbf{x}_i + b) \leq \epsilon + \varepsilon_i^*, \quad \varepsilon_i, \varepsilon_i^* \geq 0, \quad 1 \leq i \leq m$

Support Vector Machines

SVM Example: XOR

$$\left\{ \begin{array}{ll} \mathbf{x}_1 = (1, 1)^t & \in \omega_1 \quad (z_1 = +1) \\ \mathbf{x}_2 = (1, -1)^t & \in \omega_2 \quad (z_2 = -1) \\ \mathbf{x}_3 = (-1, -1)^t & \in \omega_1 \quad (z_3 = +1) \\ \mathbf{x}_4 = (-1, 1)^t & \in \omega_2 \quad (z_4 = -1) \end{array} \right.$$

- $K(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^2$ where $d = 2, m = 4$
- This leads to $d' = 6$ and ϕ homogeneous second order polynomials

Support Vector Machines

SVM Example: XOR

$$\left. \begin{array}{l} \phi_0(\mathbf{x}) = 1 \\ \phi_1(\mathbf{x}) = \sqrt{2}x_1 \\ \phi_2(\mathbf{x}) = \sqrt{2}x_2 \\ \phi_3(\mathbf{x}) = \sqrt{2}x_1x_2 \\ \phi_4(\mathbf{x}) = x_1^2 \\ \phi_5(\mathbf{x}) = x_2^2 \end{array} \right\} \mathbf{y}_i = \phi(\mathbf{x}_i) = (\phi_0(\mathbf{x}_i), \dots, \phi_5(\mathbf{x}_i))$$

- By using the kernel, the new set of features $\phi_k(\mathbf{x}_i)$ need not be calculated.

Support Vector Machines

SVM Example: XOR

$$\text{maximize} \quad \sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i=1}^4 \sum_{j=1}^4 \alpha_i \alpha_j z_i z_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad \alpha_1 + \alpha_3 - (\alpha_2 + \alpha_4) = 0, \quad \alpha_i \geq 0, \quad 1 \leq i \leq 4$$

Solution:

- Since the problem is symmetric: $\alpha_1 = \alpha_3, \alpha_2 = \alpha_4$.
- We end up in $\alpha_i = 1/8$.
- This means all four training points are support vectors.

Support Vector Machines

SVM Example: XOR

For illustrative purposes, we compute the weight vector w :

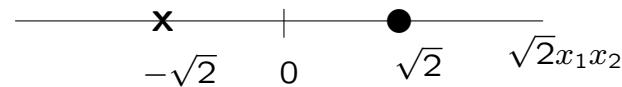
$$\begin{cases} \phi(\mathbf{x}_1) = (1, +\sqrt{2}, +\sqrt{2}, +\sqrt{2}, 1, 1)^t \\ \phi(\mathbf{x}_2) = (1, -\sqrt{2}, -\sqrt{2}, +\sqrt{2}, 1, 1)^t \\ \phi(\mathbf{x}_3) = (1, +\sqrt{2}, -\sqrt{2}, -\sqrt{2}, 1, 1)^t \\ \phi(\mathbf{x}_4) = (1, -\sqrt{2}, +\sqrt{2}, -\sqrt{2}, 1, 1)^t \end{cases}$$

$$\text{Therefore } w = \frac{1}{8} \{ \phi(\mathbf{x}_1) + \phi(\mathbf{x}_2) - \phi(\mathbf{x}_3) - \phi(\mathbf{x}_4) \} = (0, 0, 0, \frac{\sqrt{2}}{2}, 0, 0)^t$$

Support Vector Machines

SVM Example: XOR

- Looking at the mapped vectors, it is clear that their fourth component ($\sqrt{2}x_1x_2$) can itself discriminate between the two classes:



$$\left. \begin{array}{ll} \sqrt{2}x_1x_2 > 0 & \longrightarrow \mathbf{x} \in \omega_1 \\ \sqrt{2}x_1x_2 < 0 & \longrightarrow \mathbf{x} \in \omega_2 \end{array} \right\}$$

The discriminant function is then $g(\mathbf{x}) = x_1x_2$ and (from the picture) the margin is $2\sqrt{2}$.

Support Vector Machines

SVM Example: XOR

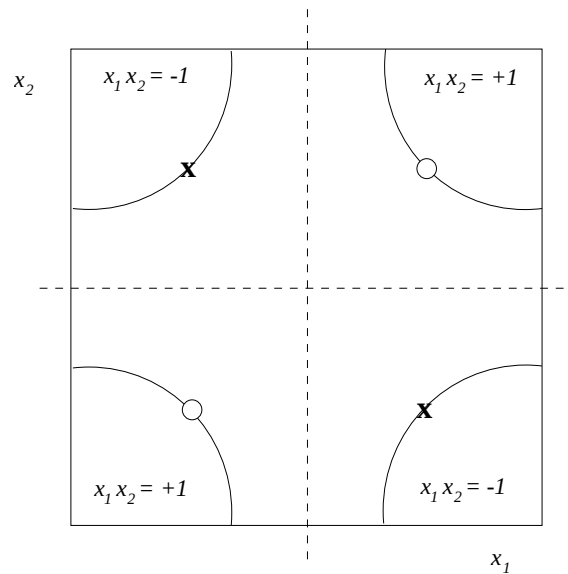
Let us check this $b = -\frac{1}{2}(1 + (-1)) = 0$

$$\pi : \langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \equiv \frac{\sqrt{2}}{2}(\sqrt{2}x_1x_2) = 0 \equiv x_1x_2 = 0$$

and the margin is $2/\|\mathbf{w}\| = 2/\frac{\sqrt{2}}{2} = \frac{4}{\sqrt{2}} = 2\sqrt{2}$.

Support Vector Machines

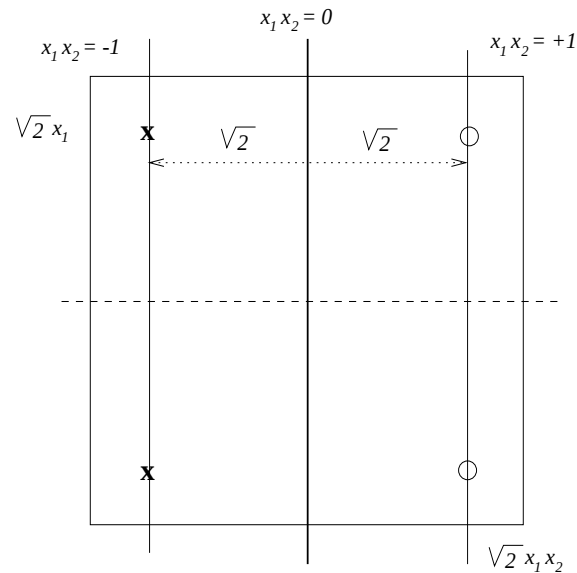
SVM Example: XOR



Original input space x_1, x_2

Support Vector Machines

SVM Example: XOR



Feature space (projection onto $\sqrt{2}x_1x_2, \sqrt{2}x_1$)

Support Vector Machines

Important remarks

- The fact that the optimal hyperplane is determined only by the support vectors is most remarkable, since usually this number will be small.
- This means all the relevant information in the data set is summarized by the support vectors.

(we would have obtained the same result by using *only* the support vectors from the outset)

Support Vector Machines

In conclusion ... PROS

- Training = solving a QP problem
- Generalization can be estimated easily after training
- Decision surfaces can be RBFs, polynomials, multilayer perceptrons, ...
- Few parameters to set: complexity depends on m_s , not on dimension of the mapped space.
- Provide with a compact description of original data
- Unlikely (though not impossible) to overfit: careful control of the bias/variance tradeoff
- Excellent performance on OCR, text categorization, face detection, ...

Support Vector Machines

In conclusion ... CONS

- Choice of best kernel is an open issue
- More efficient algorithms for solving big QP instances are being developed
- Sometimes fraction of SVs very high (need other bounds, out-of-sample error, ...)
- SVMs for regression problems carry an extra parameter (three in all)