Introduction to Support Vector Machines

Starting from slides drawn by Ming-Hsuan Yang and Antoine Cornuéjols

SVM Bibliography

B. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifier", 1992

C. Cortes, V. Vapnik, "Support vector networks". Journal of Machine Learning, 20, 1995.

V. Vapnik. "The nature of statistical learning theory". Springer Verlag, 1995.

C. Burges, "A tutorial on support vector machines for pattern recognition". Data Mining and Knowledge Descovery, 2(2):955-974, 1998.

N. Cristianini, J. Shawe-Taylor, "Support Vector Machines and other kernel-based learning methods". Cambridge University Press, 2000. Andrew Ng, "Support Vector Machines", Stanford University, CS229 Lecture Notes, Part V.

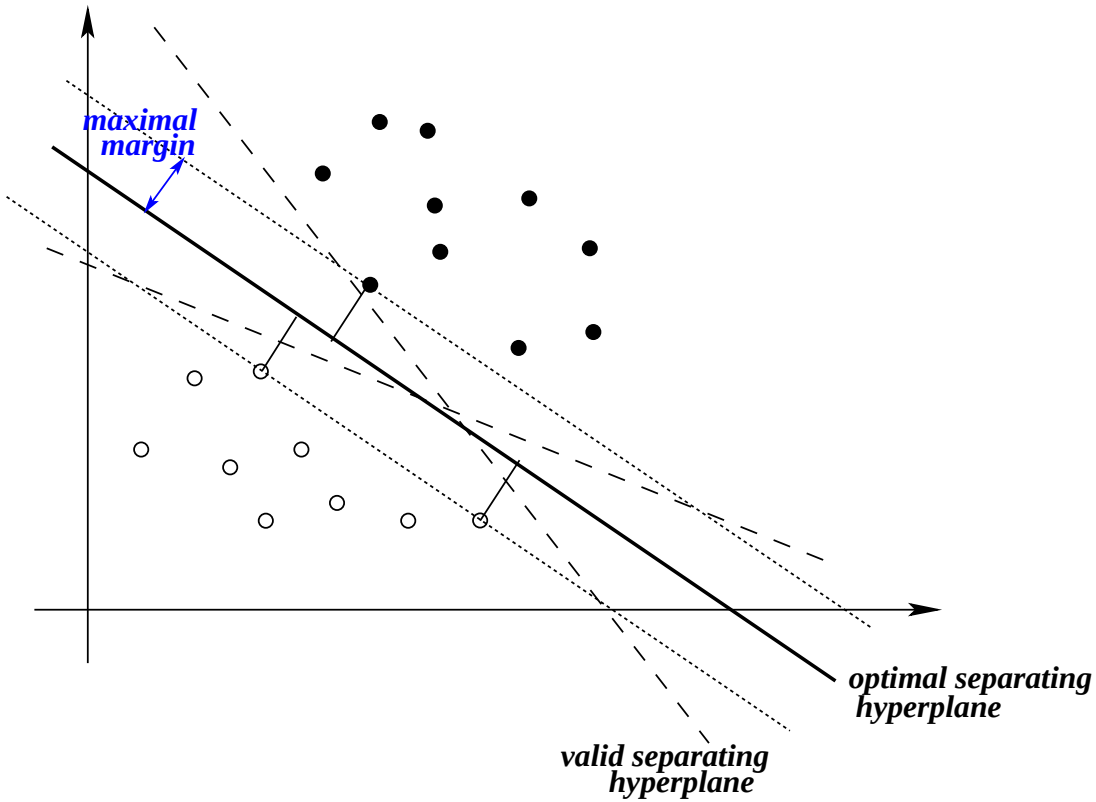
SVM — The Main Idea

Given a set of data points which belong to either of two classes, find an **optimal separating hyperplane**

- maximizing the distance (from closest points) of either class to the separating hyperplane, and
- minimizing the risk of misclassifying the training samples and the *unseen* test samples.

Approach: Formulate a constraint-based optimisation problem, then solve it using **quadratic programming (QP)**.

Optimal Separation Hyperplane



Plan

1. **Linear SVMs**

   The primal form and the dual form of linear SVMs

   Linear SVMs with soft margin

2. **Non-Linear SVMs**

   Kernel functions for SVMs

   An example of non-linear SVM

1. Linear SVMs: Formalisation

Let S be a set of points $x_i \in R^d$ with $i = 1, \ldots, m$. Each point x_i belongs to either of two classes, with label $y_i \in \{-1, +1\}$.

The set S is linear separable if there are $w \in R^d$ and $w_0 \in R$ such that

$$y_i(w \cdot x_i + w_0) \geq 1 \qquad i = 1, \ldots, m$$

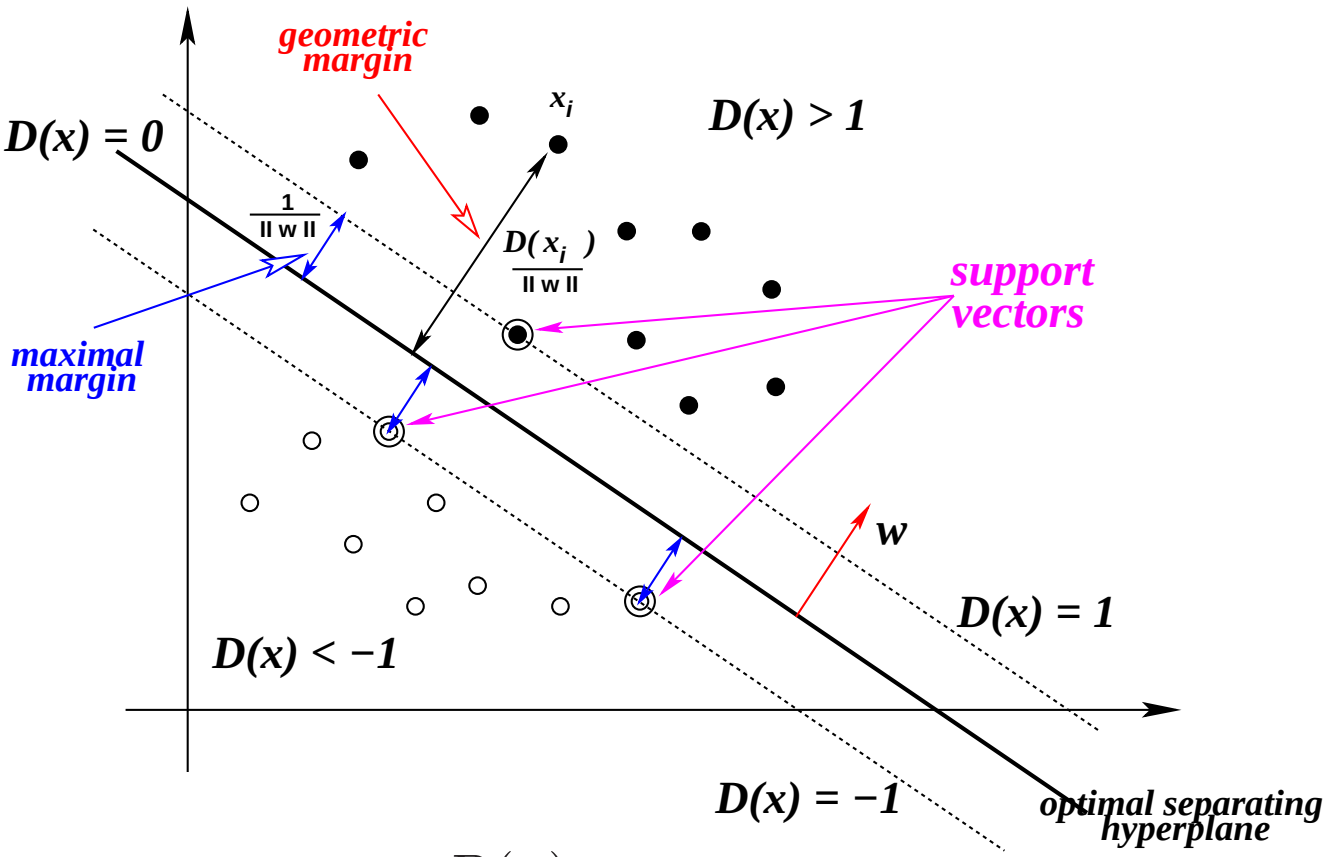
The pair (w, w_0) defines the hyperplane of equation $w \cdot x + w_0 = 0$, named the separating hyperplane.

The signed distance d_i of a point x_i to the separating hyperplane (w, w_0) is given by $d_i = \frac{w \cdot x_i + w_0}{||w||}$.

It follows that $y_i d_i \geq \frac{1}{||w||}$, therefore $\frac{1}{||w||}$ is the lower bound on the distance between points x_i and the separating hyperplane (w, w_0).

Optimal Separating Hyperplane

Given a linearly separable set S, the optimal separating hyperplane is the separating hyperplane for which the distance to the closest (either positive or negative) points in S is maximum, therefore it maximizes $\frac{1}{||w||}$.


geometric margin
x_i
D(x) > 1
D(x) = 0
$\frac{1}{\|w\|}$
$\frac{D(x_i)}{\|w\|}$
support vectors
maximal margin
w
D(x) = 1
D(x) < −1
D(x) = −1
optimal separating hyperplane

$$D(x) = w \cdot x + w_0$$

Linear SVMs: The Primal Form

$$\begin{array}{ll} \textit{minimize} & \frac{1}{2}||w||^2 \\ \textit{subject to} & y_i(w \cdot x_i + w_0) \geq 1 \text{ for } i = 1, \ldots, m \end{array}$$

This is a constrained quadratic problem (QP) with $d+1$ parameters ($w \in R^d$ and $w_0 \in R$). It can be solved by quadratic optimisation methods if d is not very big (10^3).

For large values of d (10^5): due to the Kuhn-Tucker theorem, since the above objective function and the associated constraints are convex, we can use the method of Lagrange multipliers ($\alpha_i \geq 0, i = 1, \ldots, m$) to put the above problem under an equivalent "dual" form.

Note: In the dual form, the variables (α_i) will be subject to much simpler constraints than the variables (w, w_0) in the primal form.

Linear SVMs: Getting the Dual Form

The Lagrangean function associated to the primal form of the given QP is

$$L_P(w, w_0, \alpha) = \frac{1}{2}||w^2|| - \sum_{i=1}^{m} \alpha_i(y_i(w \cdot x_i + w_0) - 1)$$

with $\alpha_i \geq 0, i = 1, \ldots, m$**. Finding the minimum of** L_P **implies**

$$\frac{\partial L_P}{\partial w_0} = -\sum_{i=1}^{m} y_i \alpha_i = 0$$

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^{m} y_i \alpha_i x_i = 0 \Rightarrow w = \sum_{i=1}^{m} y_i \alpha_i x_i$$

$$\textbf{where } \frac{\partial L_P}{\partial w} = (\frac{\partial L_P}{\partial w_1}, \ldots, \frac{\partial L_P}{\partial w_d})$$

By substituting these constraints into L_P **we get its dual form**

$$L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \; x_i \cdot x_j$$

Linear SVMs: The Dual Form

$$\begin{aligned} \textbf{\textit{maximize}} \quad & \textstyle\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \; x_i \cdot x_j \\ \textbf{\textit{subject to}} \quad & \textstyle\sum_{i=1}^{m} y_i \alpha_i = 0 \\ & \alpha_i \geq 0, i = 1, \ldots, m \end{aligned}$$

The link between the primal and the dual form:

The optimal solution $(\overline{w}, \overline{w}_0)$ **of the primal QP problem is given by**

$$\overline{w} = \sum_{i=1}^{m} \overline{\alpha}_i y_i x_i$$

$$\overline{\alpha}_i (y_i (\overline{w} \cdot x_i + \overline{w}_0) - 1) = 0 \text{ for any } i = 1, \ldots, m$$

where $\overline{\alpha}_i$ **are the optimal solutions of the above (dual form) optimisation problem.**

Support Vectors

The only $\overline{\alpha}_i$ (solutions of the dual form of our QP problem) that can be nonzero are those for which the constraints $y_i(w \cdot x_i + w_0) \geq 1$ for $i = 1, \ldots, m$ in the primal form of the QP are satisfied with the equality sign.

Because most $\overline{\alpha}_i$ are null, the vector $\overline{w}$ is a linear combination of a relative small percentage of the points x_i.

These points are called support vectors because they are the closest points to the optimal separating hyperplane (OSH) and the only points of S needed to determine the OSH.

The problem of classifying a new data point x is now simply solved by looking at $sign(\overline{w} \cdot x + \overline{w}_0)$.

Linear SVMs with Soft Margin

If the set S is not linearly separable — or one simply ignores whether or not S is linearly separable —, the previous analysis can be generalised by introducing m non-negative ("slack") variables ξ_i, for $i = 1, \ldots, m$ such that

$$y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \text{ for } i = 1, \ldots, m$$

Purpose: to allow for a small number of missclassified points, for better generalisation or computational efficiency.

Generalised OSH

The generalised OSH is then viewed as the solution to the problem:

$$\begin{aligned} &\textit{minimize} && \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ &\textit{subject to} && y_i(w\cdot x_i + w_0) \geq 1 - \xi_i \text{ for } i = 1,\ldots,m \\ & && \xi_i \geq 0 \text{ for } i = 1,\ldots,m \end{aligned}$$

The associated dual form:

$$\begin{aligned} &\textit{maximize} && \sum_{i=1}^{m}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\; x_i\cdot x_j \\ &\textit{subject to} && \sum_{i=1}^{m} y_i\alpha_i = 0 \\ & && 0 \leq \alpha_i \leq C, i = 1,\ldots,m \end{aligned}$$

As before:

$$\begin{aligned} &\overline{w} = \sum_{i=1}^{m}\overline{\alpha}_i y_i x_i \\ &\overline{\alpha}_i(y_i(\overline{w}\cdot x_i + \overline{w}_0) - 1 + \overline{\xi}_i) = 0 \\ &(C - \overline{\alpha}_i)\,\overline{\xi}_i = 0 \end{aligned}$$

The role of C:

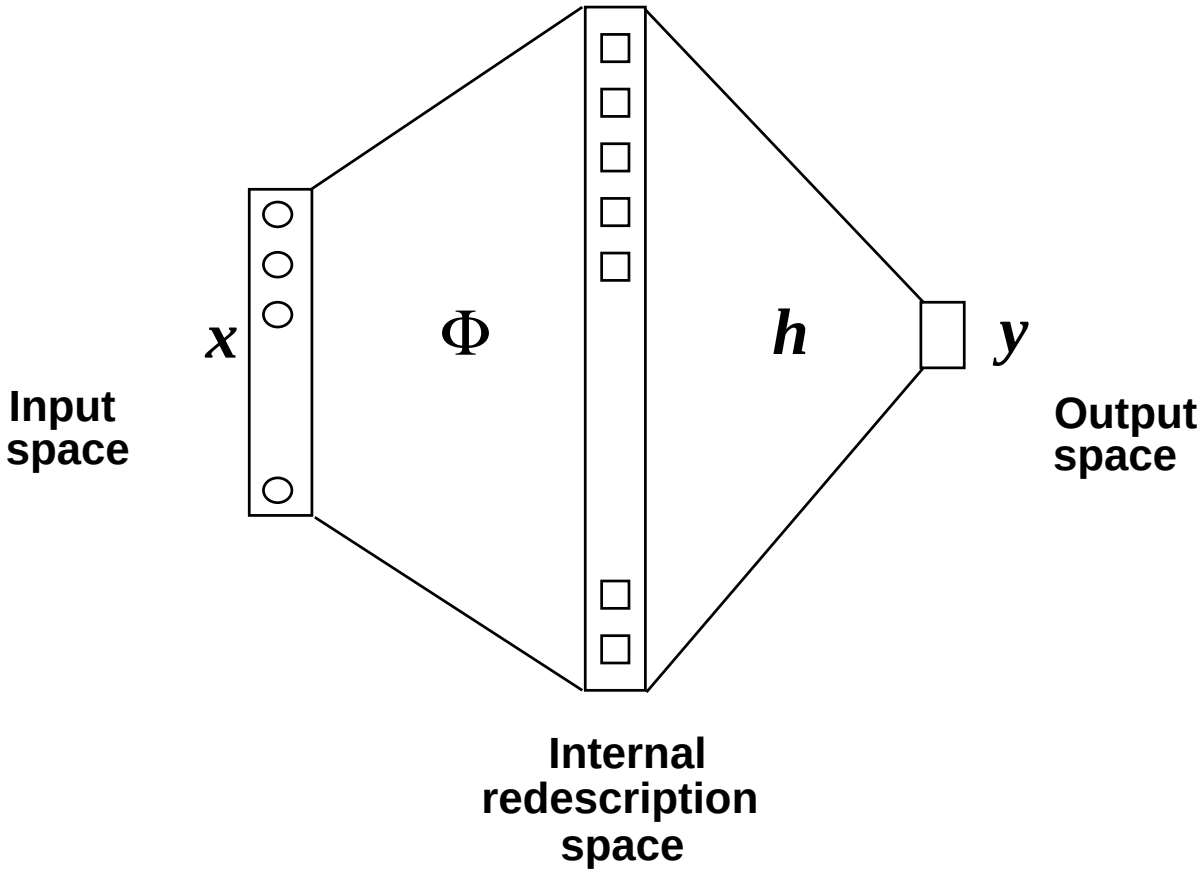
it acts as a regularizing parameter:

- **large $C \Rightarrow$ minimize the number of misclassified points**
- **small $C \Rightarrow$ maximize the minimum distance $\frac{1}{||w||}$**

2. Nonlinear Support Vector Machines

- **Note that the only way the data points appear in (the dual form of) the training problem is in the form of dot products $x_i \cdot x_j$.**
- **In a higher dimensional space, it is very likely that a linear separator can be constructed.**
- **We map the data points from the input space R^d into some space of higher dimension R^n $(n > d)$ using a function $\Phi : R^d \to R^n$**
- **Then the training algorithm would depend only on dot products of the form $\Phi(x_i) \cdot \Phi(x_j)$.**
- **Constructing (via Φ) a separating hyperplane with maximum margin in the higher-dimensional space yields a nonlinear decision boundary in the input space.**

General Schema for Nonlinear SVMs



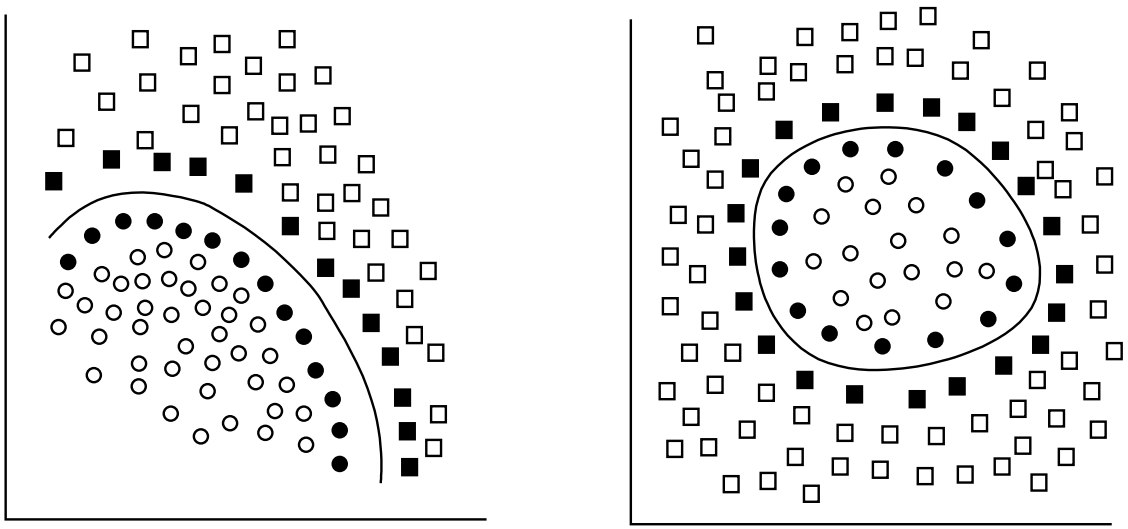
Introducing Kernel Functions

- **But the dot product is computationally expensive...**
- **If there were a "kernel function"** K **such that** $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$**, we would only use** K **in the training algorithm.**
- **All the previous derivations in the model of linear SVM hold (substituting the dot product with the kernel function), since we are still doing a linear separation, but in a different space.**
- **Important remark: By the use of the kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the higher space.**

Some Classes of Kernel Functions for SVMs

- **Polynomial:** $K(x, x') = (x \cdot x' + c)^q$
- **RBF (radial basis function):** $K(x, x') = e^{-\frac{||x-x'||^2}{2\sigma^2}}$
- **Sigmoide:** $K(x, x') = tanh(\alpha x \cdot x' - b)$

An Illustration



(a) ***(b)***

Decision surface

(a) **by a polynomial classifier, and**

(b) **by a RBF.**

Support vectors are indicated in dark fill.

Important Remark

The kernel functions require calculations in $x(\in R^d)$, therefore they are not difficult to compute.

It remains to determine which kernel function K can be associated with a given (redescription space) function Φ.

In practice, one proceeds vice versa:
we test kernel functions about which we know that they correspond to the dot product in a certain space (which will work as redescription space, never made explicit).

Therefore, the user operates by "trial and error"...

Advantage: the only parameters when training an SVM are the kernel function K, and the "tradeoff" parameter C.

Mercer's Theorem (1909):

A Characterisation of Kernel Functions for SVMs

Theorem: **Let** $K : R^d \times R^d \to R$ **be a symmetrical function.**
K **represents a dot product,**
i.e. there is a function $\Phi : R^d \to R^n$ **such that**

$$K(x, x') = \Phi(x) \cdot \Phi(x')$$

if and only if

$$\int K(x, x') f(x) f(x') dx\, dx' \geq 0$$

for any function f **such that** $\int f^2(x) dx$ **is finite.**

Remark: **The theorem doesn't say how to construct** Φ**.**

Some simple rules for building (Mercer) kernels

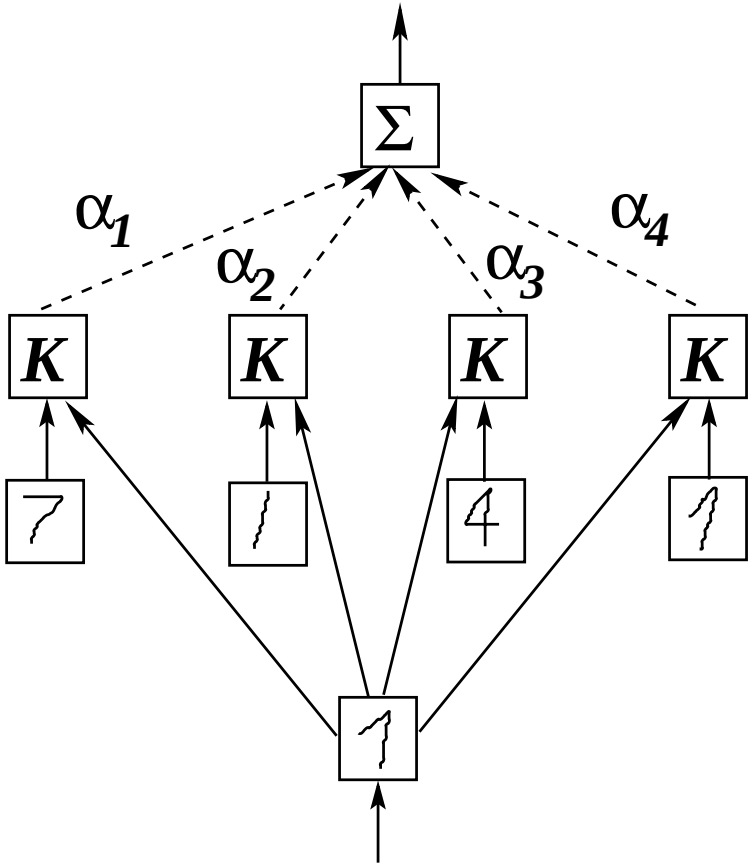
If K_1 and K_2 are kernels over $X \times X$, with $X \subseteq R^n$, then

- $K(x, y) = K_1(x, y) + K_2(x, y)$
- $K(x, y) = aK_1(x, y)$, **with** $a \in R^+$
- $K(x, y) = K_1(x, y)K_2(x, y)$

are also kernels.

Illustrating the General Architecture of SVMs

for the problem of hand-written character recognition



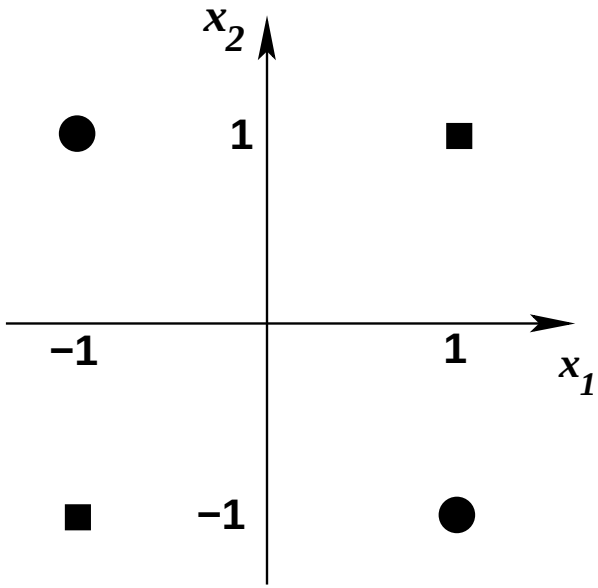
Output: $\boldsymbol{sign}(\sum_i \alpha_i y_i K(x_i, x) + w_0)$

Comparison: $K(x_i, x)$

Support vectors: $x_1, x_2, x_3, \ldots$

Input: x

An Exercise: `xor`



Note: use $K(x, x') = (x \cdot x' + 1)^2$**.**
It can be easily shown that $\Phi(x) = (x_1^2,\ x_2^2,\ \sqrt{2}x_1x_2,\ \sqrt{2}x_1,\ \sqrt{2}x_2,\ 1) \in R^6$ **for** $x = (x_1, x_2) \in R^2$**.**

i	x_i	y_i	$\Phi(i)$
1	$(1,1)$	-1	$(1,1,\sqrt{2},\sqrt{2},\sqrt{2},1)$
2	$(1,-1)$	1	$(1,1,-\sqrt{2},\sqrt{2},-\sqrt{2},1)$
3	$(-1,1)$	1	$(1,1,-\sqrt{2},-\sqrt{2},\sqrt{2},1)$
4	$(-1,-1)$	-1	$(1,1,\sqrt{2},-\sqrt{2},-\sqrt{2},1)$

$$
\begin{aligned}
L_D(\alpha) &= \textstyle\sum_{i=1}^4 \alpha_i - \frac{1}{2}\sum_{i=1}^4 \sum_{j=1}^4 \alpha_i\alpha_j y_i y_j \; \Phi(x_i)\cdot\Phi(x_j) \\
&= \alpha_1+\alpha_2+\alpha_3+\alpha_4- \\
&\quad -\tfrac{1}{2}(\; 9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4+ \\
&\qquad 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4+ \\
&\qquad 9\alpha_3^2 - 2\alpha_3\alpha_4+ \\
&\qquad 9\alpha_4^2)
\end{aligned}
$$

subject to:

$-\alpha_1+\alpha_2+\alpha_3-\alpha_4=0$

$$\frac{\partial L_D(\alpha)}{\partial \alpha_1} = 0 \Leftrightarrow 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1$$
$$\frac{\partial L_D(\alpha)}{\partial \alpha_2} = 0 \Leftrightarrow \alpha_1 - 9\alpha_2 - \alpha_3 + \alpha_4 = -1$$
$$\frac{\partial L_D(\alpha)}{\partial \alpha_3} = 0 \Leftrightarrow \alpha_1 - \alpha_2 - 9\alpha_3 + \alpha_4 = -1$$
$$\frac{\partial L_D(\alpha)}{\partial \alpha_4} = 0 \Leftrightarrow \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$$

$$\bar{\alpha}_1 = \bar{\alpha}_2 = \bar{\alpha}_3 = \bar{\alpha}_4 = \frac{1}{8}$$

$$\bar{w} = \frac{1}{8}(-\Phi(x_1) + \Phi(x_2) + \Phi(x_3) - \Phi(x_4)) = \frac{1}{8}(0,\ 0, -4\sqrt{2},\ 0,\ 0,\ 0)$$
$$\bar{w} \cdot \Phi(x_i) + \bar{w}_0 = y_i \Rightarrow \bar{w}_0 = 0$$

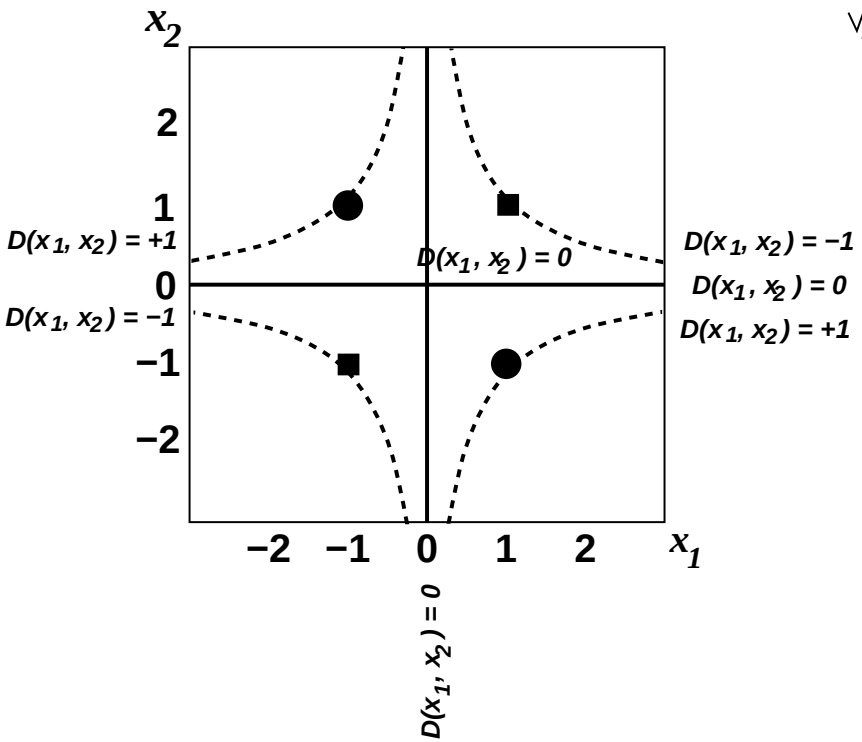
The optimal separation hyperplane: $\bar{w} \cdot \Phi(x) + \bar{w}_0 = 0 \Leftrightarrow -x_1 x_2 = 0$

Test: ***sign*** $(-x_1 x_2)$

The xor Exercise: Result

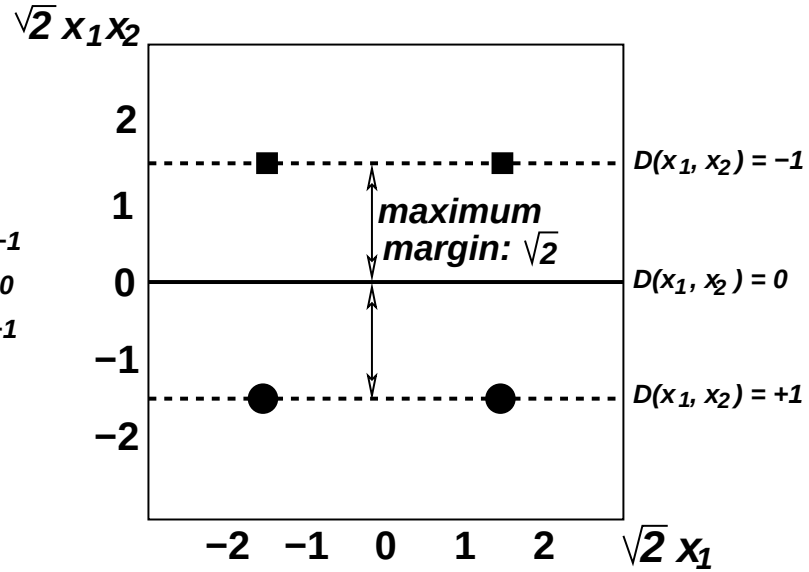
Input space

$$D(x_1, x_2) = -x_1 x_2$$



Feature space

$$D(x_1, x_2) = -\sqrt{2}\, x_1 x_2$$



Concluding Remarks: SVM — Pros and Cons

Pros:

- Find the optimal separation hyperplane.
- Can deal with very high dimentional data.
- Some kernels have infinite Vapnik-Chervonenkis dimension (see Computational learning theory, ch. 7 in Tom Mitchell's book), which means that they can learn very elaborate concepts.
- Usually work very well.

Cons:

- Require both positive and negative examples.
- Need to select a good kernel function.
- Require lots of memory and CPU time.
- There are some numerical stability problems in solving the constrained QP.

Multi-class Classification with SVM

SVMs can only do binary classification.

For M classes, one can use the one-against-the-rest approach: construct a hyperplane between class k and the $M-1$ other classes. $\Rightarrow$ M SVMs.

To predict the output of a new instance, just predict with each of these M SVMs, and then find out which one puts the prediction furthest into the positive region of the instance space.

SVM Implementations

- SVM^{light}
- LIBSVM
- mySVM
- Matlab
- Huller
- ...

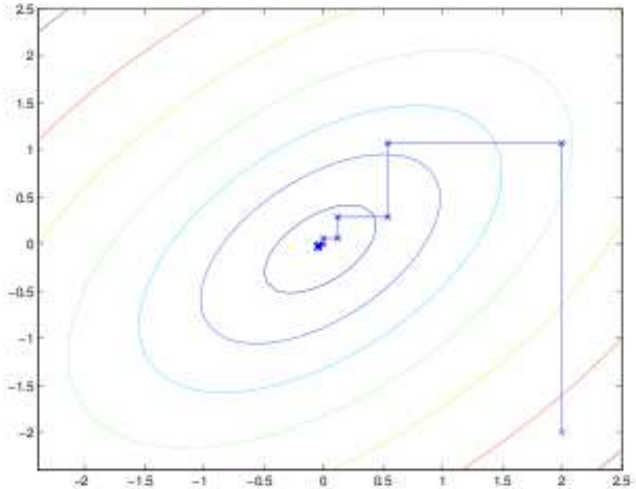
The SMO (Sequential Minimal Optimization) algorithm
John Pratt, 1998

Optimization problem:

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} -\frac{1}{2} y_i y_j \alpha_i \alpha_j x_i \cdot x_j$$

$$\textbf{s. t.} \quad 0 \le \alpha_i \le C, \; i = 1, \ldots, m$$

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



Andrew Ng, Stanford, 2012 fall, ML course, Lecture notes 3.

Algorithm:

Repeat till convergence {

1. **Select some pair α_i and α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).**
2. **Reoptimize $W(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed.**

}

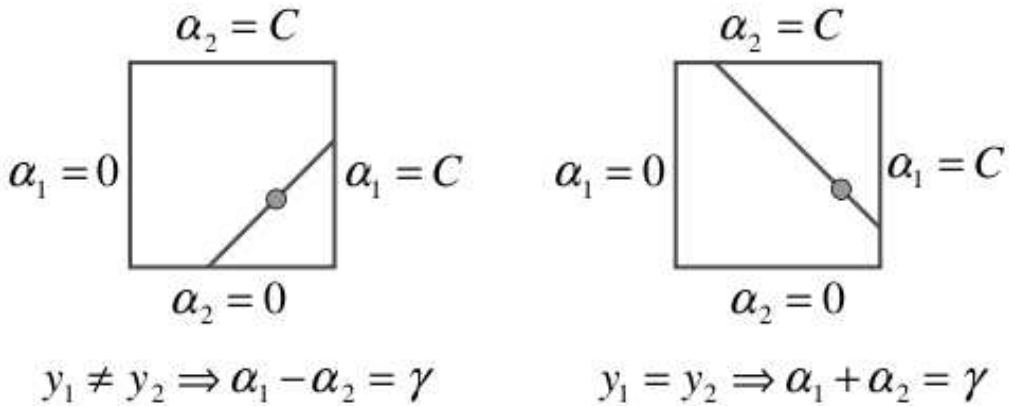
Update equations:

- $$\alpha_j^{\textbf{\textit{new, unclipped}}} = \alpha_j - \frac{y_j(E_i - E_j)}{\eta}$$
$$\alpha_j^{\textbf{\textit{new, clipped}}} = \begin{cases} H & \textbf{if } \alpha_j^{\textbf{\textit{new, unclipped}}} > H \\ \alpha_j^{\textbf{\textit{new, unclipped}}} & \textbf{if } L \leq \alpha_j^{\textbf{\textit{new, unclipped}}} \leq H \\ L & \textbf{if } \alpha_j^{\textbf{\textit{new, unclipped}}} < L \end{cases}$$

  - **where**
$$E_k = w \cdot x_k + w_0 - y_k$$
$$w = \sum_{i=1}^{4} y_i \alpha_i x_i,$$
$$\eta = - \parallel x_i - x_j \parallel^2$$
$$L = \max(0, \alpha_j - \alpha_i) \textbf{ şi } H = \min(C, C + \alpha_j - \alpha_j) \textbf{ if } y_i \neq y_j$$
$$L = \max(0, \alpha_j + \alpha_i - C) \textbf{ şi } H = \min(C, \alpha_j + \alpha_j) \textbf{ if } y_i = y_j$$

- $$\alpha_1^{\textbf{\textit{new}}} = \alpha_1^{\textbf{\textit{old}}} + y_1 y_2 (\alpha_2^{\textbf{\textit{old}}} - \alpha_2^{\textbf{\textit{new}}})$$

Assume that α_i, α_2 are the free dual variables, and let the i and j indices be used to index other variables.



Credit: John Pratt, *Fast training of SVMs using Sequential Minimal Optimization*, 2000

The two Lagrange multipliers must fulfill all the constraints of the full problem.The inequality constraints cause the Lagrange multipliers to lie in the box. The liniar equality constraint causes them to lie on a diagonal line. Thefore, one step of SMO must find an optimum of the objective function on a diagonal line segment. In this figure, $\gamma = \alpha_1^{old} + s\alpha_2^{old}$, is a constant that depends on the previous values of α_1 and α_2, and $s = y_1 y_2$.

Proof

[following N. Cristianini and J. Shawe-Taylor, *An introduction to SVM*, 2000, pag. 138-140]

The objective function:

$$
\begin{aligned}
W(\alpha_1, \alpha_2, \ldots, \alpha_m) &= \textstyle\sum_{i=1}^m \alpha_i - \frac{1}{2} y_i y_j \alpha_i \alpha_j x_i \cdot x_j \\
v_i &\overset{\textbf{\textit{not.}}}{=} \left(\textstyle\sum_{j=3}^m y_j \alpha_j x_j \right) \cdot x_i = f(x_i) - \textstyle\sum_{j=1}^2 y_j \alpha_j x_j \cdot x_i \ \textbf{for } i = 1, 2 \\
\Rightarrow W(\alpha_1, \alpha_2) &= \alpha_1 + \alpha_2 - \tfrac{1}{2}\alpha_1^2 x_1^2 - \tfrac{1}{2}\alpha_2^2 x_2^2 - y_1 y_2 \alpha_1 \alpha_2 x_1 \cdot x_2 - y_1 \alpha_1 v_1 - y_1 \alpha_2 v_2 + \textbf{const}
\end{aligned}
$$

From

$$
\begin{aligned}
\textstyle\sum_{i=1}^m y_i \alpha_i = 0 \quad &\Rightarrow \quad \alpha_1^{\textbf{\textit{old}}} + \alpha_2^{\textbf{\textit{old}}} = \alpha_1^{\textbf{\textit{new}}} + \alpha_2^{\textbf{\textit{new}}} = \gamma \ \textbf{(another constant)} \\
\Rightarrow \alpha_1^{\textbf{\textit{new}}} &= \alpha_1^{\textbf{\textit{old}}} + y_1 y_2 (\alpha_2^{\textbf{\textit{old}}} - \alpha_2^{\textbf{\textit{new}}}) \\
s &\overset{\textbf{\textit{not.}}}{=} y_1 y_2 \\
\Rightarrow W(\alpha_2) &= \gamma - s\alpha_2 + \alpha_2 - \tfrac{1}{2}(\gamma - s\alpha_2)^2 x_1^2 - \tfrac{1}{2}\alpha_2^2 x_2^2 \\
&\quad - y_1 y_2 x_1 \cdot x_2 (\gamma - s\alpha_2)\alpha_2 - y_1 v_1 (\gamma - s\alpha_2) - y_2 v_2 \alpha_2 + \textbf{const}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial W(\alpha_2)}{\partial \alpha_2} &= -s + 1 + \tfrac{1}{2}\, x_1^2\, 2s(\gamma - s\alpha_2) - \tfrac{1}{2}\, x_2^2\, 2\alpha_2 \\
&\quad -y_1 y_2\, (x_1 \cdot x_2)\, \gamma + 2\, y_1 y_2\, (x_1 \cdot x_2)\, s\alpha_2 + y_1 v_1 s - y_2 v_2 \\
&= -s + 1 + x_1^2(s\gamma - \alpha_2) - x_2^2 \alpha_2 - s\gamma x_1^2 - s\gamma\, (x_1 \cdot x_2)\, + y_1 v_1 s - y_2 v_2
\end{aligned}
$$

Finding the stationary point:

$$
\frac{\partial W(\alpha_2)}{\partial \alpha_2} = 0 \Rightarrow \alpha_2^{\textbf{\textit{new, unclipped}}}(x_1^2 + x_2^2 - 2x_1 \cdot x_2) = 1 - s + \gamma s x_1^2 - \gamma s\, (x_1 \cdot x_2) + y_2 v_1 - y_2 v_2
$$

$$
\begin{aligned}
1 - s + \gamma s x_1^2 - \gamma s\, (x_1 \cdot x_2) + y_2 v_1 - y_2 v_2 &= y_2(y_2 - y_1 + \gamma y_1\, (x_1^2 - x_1 \cdot x_2) + v_1 - v_2) \\
v_1 - v_2 &= f(x_1) - y_1 \alpha_1^{\textbf{\textit{old}}} x_1^2 - y_2 \alpha_2^{\textbf{\textit{old}}}\, x_1 \cdot x_2 \\
&\quad -f(x_2) + y_1 \alpha_1^{\textbf{\textit{old}}} x_1 \cdot x_2 + y_2 \alpha_2^{\textbf{\textit{old}}} x_2^2 \\
&= f(x_1) - y_1(\gamma - s\alpha_2^{\textbf{\textit{old}}}) x_1^2 - y_2 \alpha_2^{\textbf{\textit{old}}}\, x_1 \cdot x_2 \\
&\quad -f(x_2) + y_1(\gamma - s\alpha_2^{\textbf{\textit{old}}}) x_1 \cdot x_2 + y_2 \alpha_2^{\textbf{\textit{old}}} x_2^2 \\
1 - s + \gamma s x_1^2 - \gamma s\, (x_1 \cdot x_2) + y_2 v_1 - y_2 v_2 &= y_2(y_2 - y_1 + f(x_1) + y_1 s\alpha_2^{\textbf{\textit{old}}} x_1^2 - y_2 \alpha_2^{\textbf{\textit{old}}} x_1 \cdot x_2 \\
&\quad -f(x_2) - y_1 s\alpha_2^{\textbf{\textit{old}}} x_1 \cdot x_2 + y_2 \alpha_2^{\textbf{\textit{old}}} x_2^2) \\
&= y_2(f(x_1) - y_1 - f(x_2) + y_2) \\
&\quad +\alpha_2^{\textbf{\textit{old}}}(x_1^2 + x_2^2 - 2x_1 \cdot x_2) \\
&= y_2(E_1 - E_2) + \alpha_2^{\textbf{\textit{old}}}(x_1^2 + x_2^2 - 2x_1 \cdot x_2) \\
\Rightarrow \alpha_2^{\textbf{\textit{new, unclipped}}}(x_1^2 + x_2^2 - 2x_1 \cdot x_2) &= y_2(E_1 - E_2) + \alpha_2^{\textbf{\textit{old}}}(x_1^2 + x_2^2 - 2x_1 \cdot x_2) \\
\Rightarrow \alpha_2^{\textbf{\textit{new, unclipped}}} &= \alpha_2^{\textbf{\textit{old}}} - \frac{y_2(E_1 - E_2)}{\eta}
\end{aligned}
$$