Nomenclature

- Given $x_1, x_2, \ldots, x_n$ sample points, with true category labels: $y_1, y_2, \ldots, y_n$

$$\left.\begin{array}{l} y_i = 1 \\ y_i = -1 \end{array}\right\} \begin{array}{l} \text{if point } x_i \text{ is from class } \omega_1 \\ \text{if point } x_i \text{ is from class } \omega_2 \end{array}$$

- Decision are made according to:

$$if\;\; \mathbf{w^t} x_{\mathbf{i}}' = w^{\mathbf{t}} x_i + b > 0 \quad \text{class } \omega_1 \text{ is chosen}$$

$$if\;\; \mathbf{w^t} x_{\mathbf{i}}' = w^{\mathbf{t}} x_i + b < 0 \quad \text{class } \omega_2 \text{ is chosen}$$

- Now these decisions are wrong when $\mathbf{w^t x_i}$ is negative and belongs to class ω_1.

  Let $z_i = \alpha_i x_i$ Then $z_i > 0$ when correctly labelled, negative otherwise.

Support Vector Machines

- Support vector machines differ from standard linear machines in three ways.
- Discriminant function flexibility
  - Linear
    - But with nonlinear preprocessing possible
    - efficient evaluation via kernel trick
- Error function
  - Max margin, constrained by misclassification errors
- Optimization
  - Choice of error function allows global solution
  - Nature of solution focuses on points on points on margin (the support vectors)

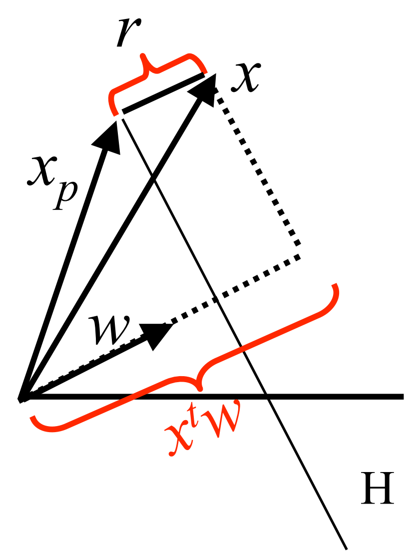
$$\mathbf{x} = \mathbf{x_p} + \frac{r\mathbf{w}}{\|\mathbf{w}\|}$$

$$\text{since } g(\mathbf{x_p}) = 0 \text{ and } \mathbf{w^t w} = \|w\|^2$$

$$g(\mathbf{x}) = \mathbf{w^t x} + w_0 \Rightarrow \mathbf{w^t}\left(\mathbf{x_p} + \frac{r\mathbf{w}}{\|\mathbf{w}\|}\right) + w_0$$

$$= g(\mathbf{x_p}) + \mathbf{w^t w}\frac{r}{\|\mathbf{w}\|}$$

$$\Rightarrow \mathrm{r} = \frac{g(x)}{\|w\|}$$



$$\textit{in particular } \mathrm{d}([0,0],\mathrm{H}) = \frac{\mathrm{w}_0}{\|\mathrm{w}\|}$$

– In conclusion, a linear discriminant function divides the feature space by a hyperplane decision surface

– The orientation of the surface is determined by the normal vector w and the location of the surface is determined by the bias

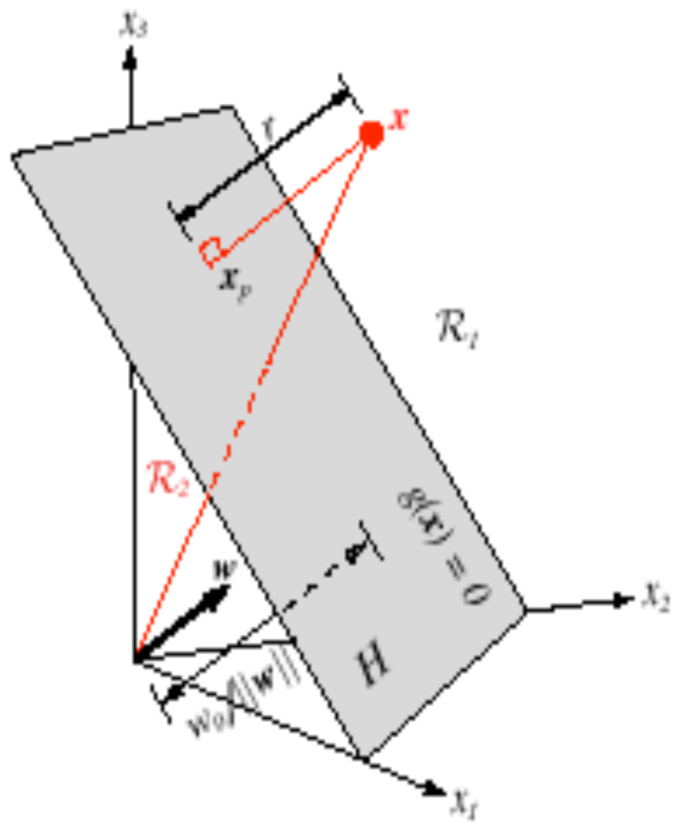


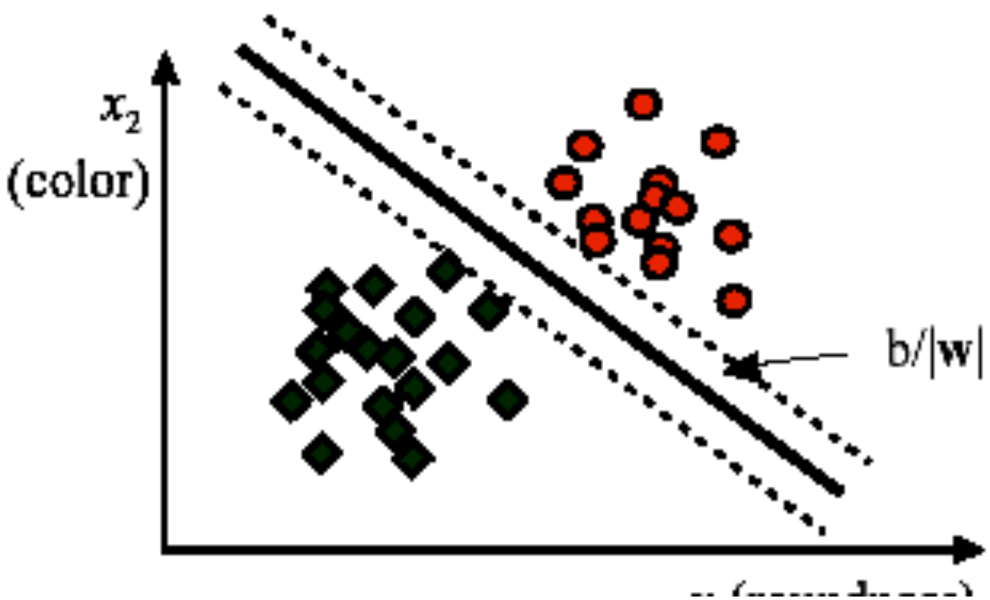
FIGURE 5.2. The linear decision boundary H, where $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$, separates the feature space into two half-spaces $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Support vector machines

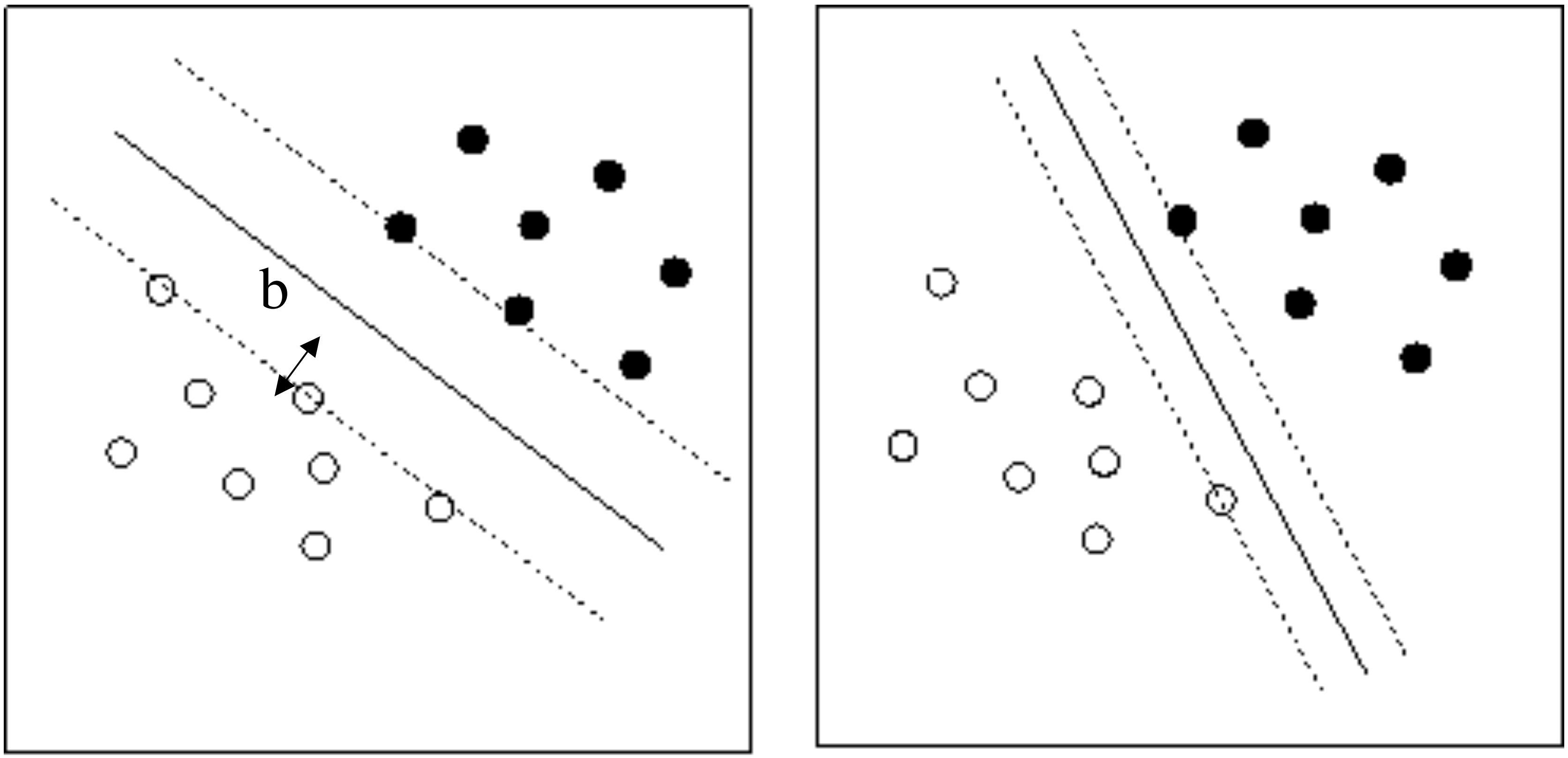
We assign a value $y \in \{+1,-1\}$ to each point in the training set and seek a $\mathbf{w}$ for which $y_i(\mathbf{w}^T\mathbf{x}+w_0)>0$ for all i.

We want to have a margin, so : $y_i(\mathbf{w}^T\mathbf{x}+w_0)\geq b$.
If we scale $|\mathbf{w}|, w_0$ and b, nothing changes, so we set $b=1$.

We get two hyperplanes :
$H_1 : \mathbf{w}^T\mathbf{x}+w_0=+1$
$H_2 : \mathbf{w}^T\mathbf{x}+w_0=-1.$
The size of the margin is $1/|\mathbf{w}|$
The points that lie on the hyperplanes are called *support vectors*.



Margins in data space



(a) Larger margin (b) Smaller margin

Larger margins promote uniqueness for underconstrained problems

- **Therefore, the problem of maximizing the margin is equivalent to**

$$\begin{array}{ll} \text{minimize} & J(w) = \frac{1}{2}\|w\|^2 \\ \text{subject to} & y_i\left(w^T x_i + b\right) \geq 1 \ \forall i \end{array}$$

  - Notice that J(w) is a quadratic function, which means that there exists a single global minimum and no local minima
- **To solve this problem, we will use classical Lagrangian optimization techniques**
  - We first present the Kuhn-Tucker Theorem, which provides an essential result for the interpretation of Support Vector Machines

(Kuhn-Tucker Theorem)

- **Given an optimization problem with convex domain $\Omega \subseteq R^N$**

$$\begin{aligned} &\text{minimize} && f(z) && z \in \Omega \\ &\text{subject to} && g_i(z) \le 0 && i = 1,\dots,k \\ & && h_i(z) = 0 && i = 1,\dots,m \end{aligned}$$

- with $f \in C^1$ convex and g_i, h_i affine, necessary and sufficient conditions for a normal point z^* to be an optimum are the existence of α^*, β^* such that

$$\begin{aligned} &\frac{\partial L(z^*, \alpha^*, \beta^*)}{\partial z} = 0 \\ &\frac{\partial L(z^*, \alpha^*, \beta^*)}{\partial \beta} = 0 \\ &\alpha_i^* g_i(z^*) = 0 && i = 1,\dots,k \\ &g_i(z^*) \le 0 && i = 1,\dots,k \\ &\alpha_i^* \ge 0 && i = 1,\dots,k \end{aligned}$$

where $L(z, \alpha, \beta) = f(z) + \sum_{i=1}^{k} \alpha_i g_i(z) + \sum_{i=1}^{m} \beta_i h_i(z)$

- $L(z,\alpha,\beta)$ is known as a *generalized Lagrangian function*
- The third condition is know as the Karush-Kuhn-Tucker (KKT) complementary condition. It implies that for active constraints $\alpha_i \geq 0$; and for inactive constraints $\alpha_i = 0$
  - As we will see in a minute, the KKT condition allows us to identify the training examples that define the largest margin hyperplane. These examples will be known as **Support Vectors**.

Constrained Optimization Problems

Minimize enforcing Equality Constraints

Find: $\vec{x}^* = \vec{x}_{\min}$ such that $h(\vec{x}^*)=0$

Lagrange Multiplier

$$\min \ f(x_1, x_2) \qquad s.t. \ \ h(x_1, x_2) = 0$$

$$L(x_1, x_2, \upsilon) = f(x_1, x_2) + \upsilon\, h(x_1, x_2) \longleftarrow \begin{pmatrix} L : Lagrange \ \ func \\ \upsilon \ : Lagrange \ \ multiplier \end{pmatrix}$$

$$\frac{\partial L(x_1^*, x_2^*)}{\partial x_1} = \frac{\partial f(x_1^*, x_2^*)}{\partial x_1} + \upsilon \frac{\partial f(x_1^*, x_2^*)}{\partial x_1} = 0$$

$$\frac{\partial L(x_1^*, x_2^*)}{\partial x_2} = \frac{\partial f(x_1^*, x_2^*)}{\partial x_2} + \upsilon \frac{\partial f(x_1^*, x_2^*)}{\partial x_2} = 0$$

$$\nabla L(x^*) = \nabla f(x^*) + \nu \nabla h(x^*) = \underline{0}$$

$$\nabla f(x^*) = -\nu \nabla h(x^*) \longrightarrow \text{geometrical meaning}$$

At the candidate minimum point, gradients of the cost and constraint func are along the same line.

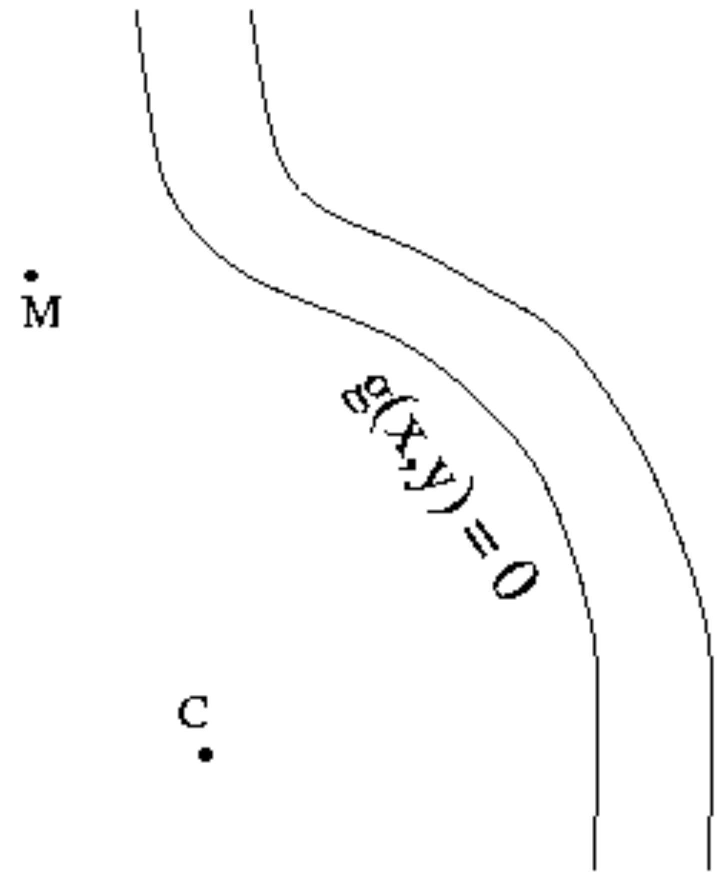
(In other words, ∇f is a linear combination of ∇h

$$L(x,\nu) = f(x) + \nu^T h(x)$$

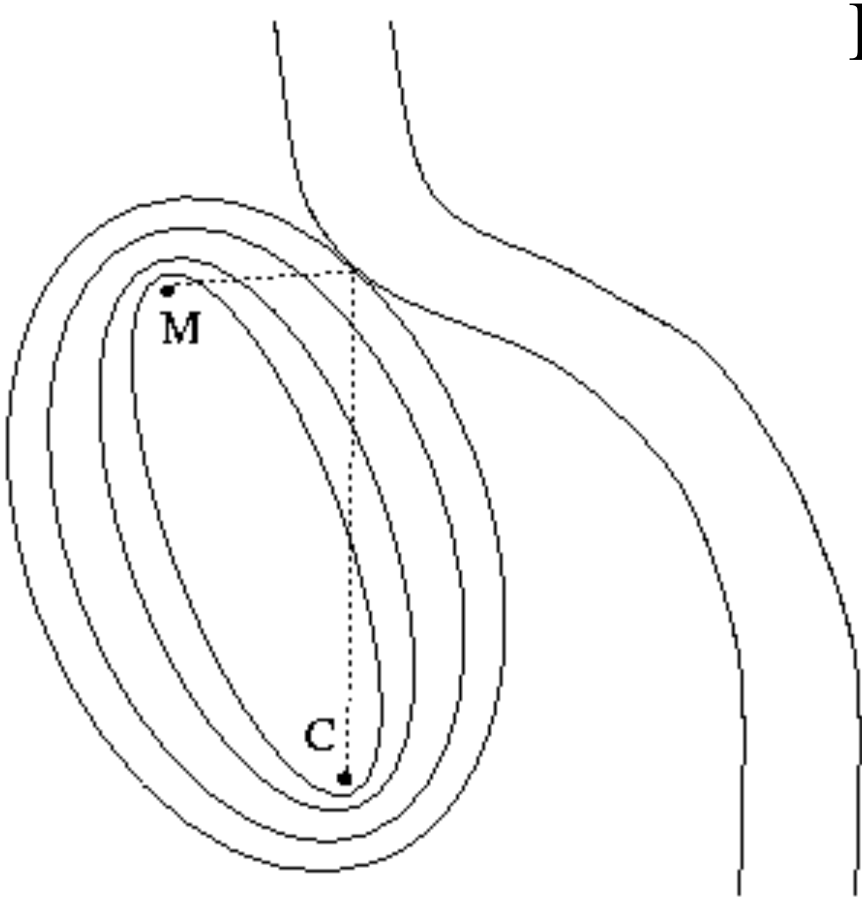
Therefore constrained optimization is converted to unconstrained optimization.

$$\nabla L(x^*,\nu^*) = 0$$

- The "Milkmaid problem"
- It's milking time at the farm, and the milkmaid has been sent to the field to get the day's milk. She is in quite a hurry, because she has a date, so she wants to finish her job as quickly as possible. However, before she gathers the milk, she has to rinse out her bucket in the nearby river.
- Just when she reaches point M, our heroine spots the cow, at point C. She is in a hurry, so she wants to take the shortest possible path from where she is to the river and then to the cow. If the near bank of the river is a curve satisfying the function g(x,y) = 0, what is the shortest path for the milkmaid to take? (Assume that the field is flat and uniform and that all points on the river bank are equally good.)



- Problem:
- Minimize f(P) = d(M,P) + d(P,C),
  - such that g(P) = 0.



F(P,α) = f(P) - α g(P).

$$\nabla F = 0$$

$$\frac{\partial f}{\partial P} + \alpha \frac{\partial g}{\partial P} = 0$$

$$\frac{\partial F}{\partial \alpha} = 0 \quad \rightarrow g(P) = 0$$

Constrained Optimization

Instead of solving

$$\left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \frac{\partial f(\mathbf{w})}{\partial w_2}\right) = (0, 0)$$

deal with Lagrangian

$$L(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \alpha \cdot g(\mathbf{w}) + \beta \cdot h(\mathbf{w})$$

and solve the dual problem by reasoning about the dual variables α, β.

Primal problem:

$$\underline{\text{minimize}} \quad f(\mathbf{w})$$

$$\underline{\text{subject to}} \quad g(\mathbf{w}) \leq 0, \quad h(\mathbf{w}) = 0$$

Dual problem:

$\theta(\alpha, \beta)$ is minimal value of

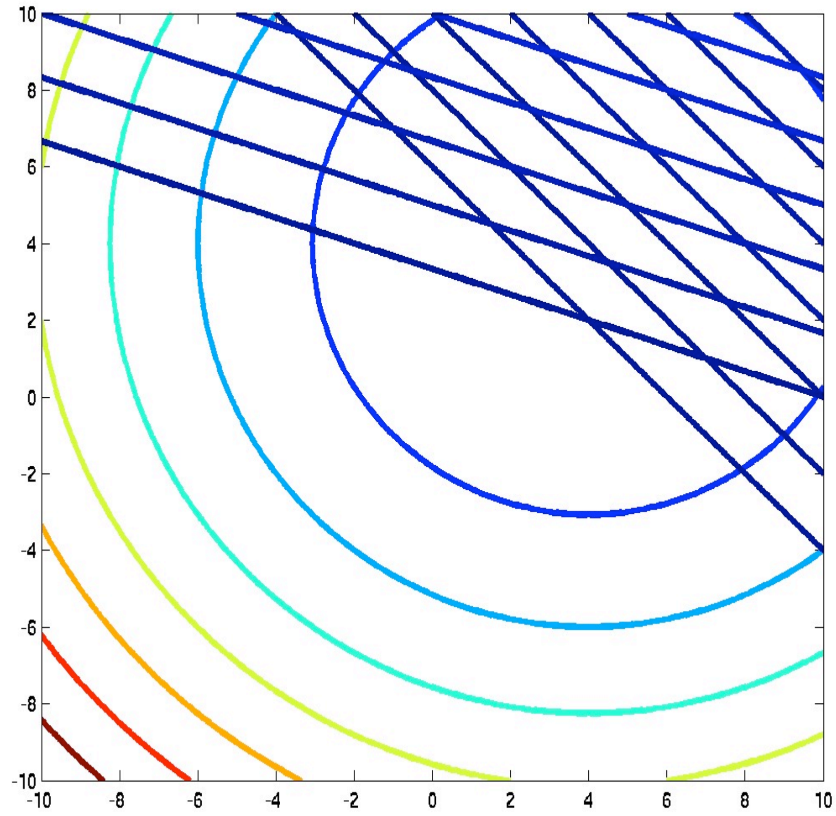
$$L(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \alpha \cdot g(\mathbf{w}) + \beta \cdot h(\mathbf{w})$$

w.r.t. $\mathbf{w}$

$$\underline{\text{maximize}} \quad \theta(\alpha, \beta)$$

$$\underline{\text{subject to}} \quad \alpha \geq 0$$

Kuhn-Tucker Example



Consider the problem

$$\min\left\{ f(\vec{x}) = (x_1 - 4)^2 + (x_2 - 4)^2 \right\},$$

such that

$$g_1(\vec{x}) = x_1 + x_2 \le 6 \quad \text{and}$$

$$g_2(\vec{x}) = x_1 + 3x_2 \le 4$$

We form a new function for minimization:

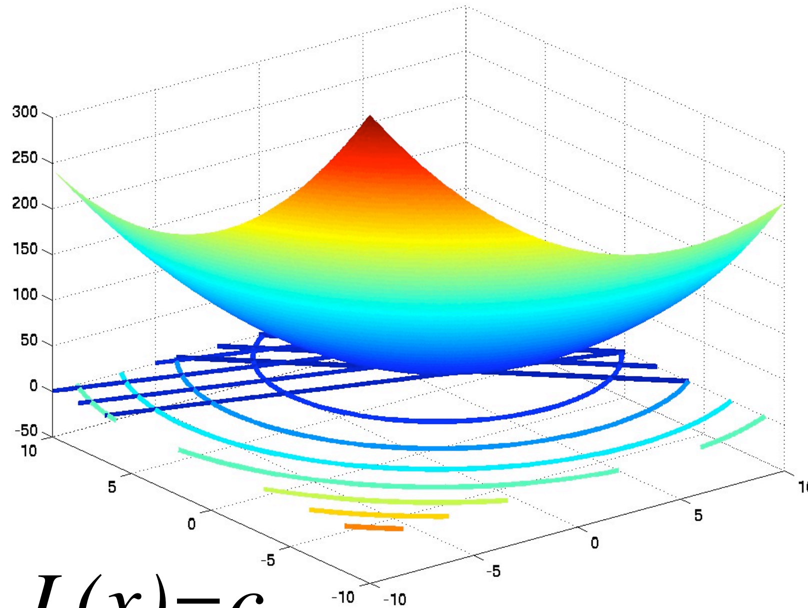
$$L(\vec{x}) = f(\vec{x}) + \nu_1 g_1(\vec{x}) + \nu_2 g_2(\vec{x})$$

$$L(\vec{x}) = (x_1 - 4)^2 + (x_2 - 4)^2 + \nu_1(x_1 + x_2 - 6) + \nu_2(x_1 + 3x_2 - 4)$$

The Kuhn - Tucker conditions are:

$$\nabla L(\vec{x}) = 0, \quad \nu_i \ge 0, \quad \nu_i g_i(\vec{x}) = 0$$

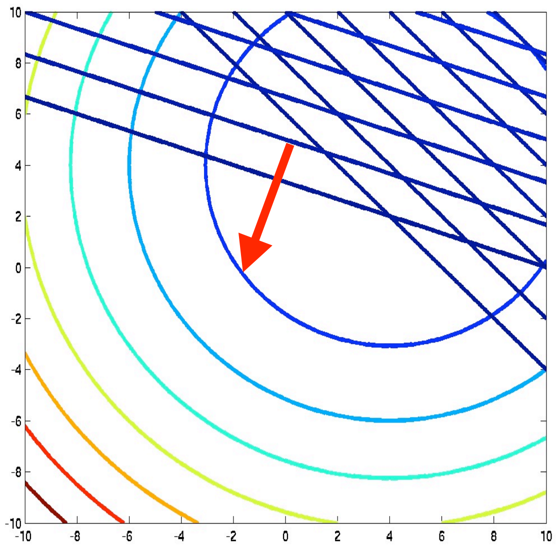
What do the multipliers do?



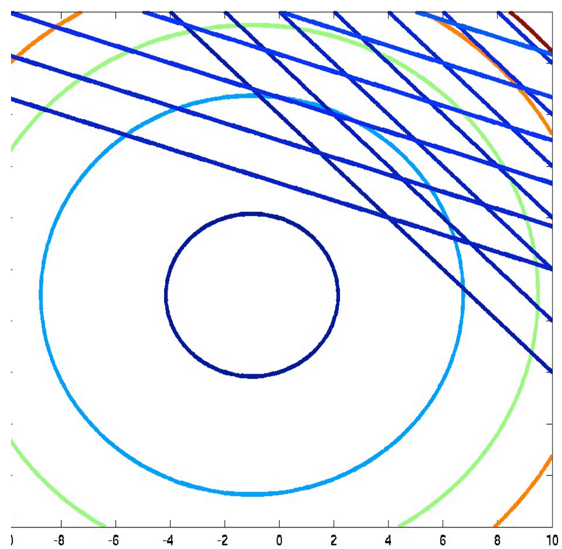
Adding constraint shifts *L(x)*
in direction of constraint normal

Circles: $L(x)=c$

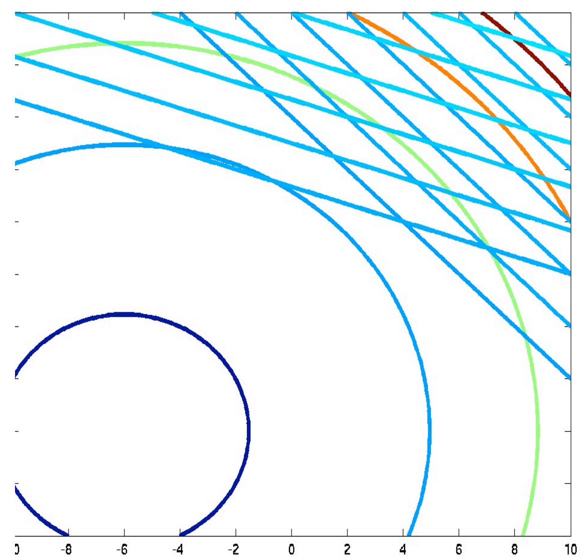
$\nu_1=0,\ \nu_2=0$



$\nu_1=5,\ \nu_2=0$



$\nu_1=10,\ \nu_2=0$



Kuhn-Tucker conditions:

$$\nabla L(\vec{x}) = \begin{bmatrix} \dfrac{\partial L(\vec{x})}{\partial x_1} = 2(x_1 - 4) + \nu_1 + \nu_2 = 0 \\ \dfrac{\partial L(\vec{x})}{\partial x_2} = 2(x_2 - 4) + \nu_1 + 3\nu_2 = 0 \\ \dfrac{\partial L(\vec{x})}{\partial \nu_1} = (x_1 + x_2 - 6) \le 0 \\ \dfrac{\partial L(\vec{x})}{\partial \nu_2} = (x_1 + 3x_2 - 4) \le 0 \end{bmatrix}$$

$$\nu_1 \ge 0$$

$$\nu_2 \ge 0$$

$$\nu_1(x_1 + x_2 - 6) = 0$$

$$\nu_2(x_1 + 3x_2 - 4) = 0$$

Solve for x in terms of ν_1, ν_2
Then substitute and solve for ν_1, ν_2

$$x_1 = -(\nu_1 + \nu_2)/2 + 4$$

$$x_2 = -(\nu_1 + 3\nu_2)/2 + 4$$

Plugging in:

$$\nu_1(-(\nu_1 + \nu_2)/2 + 4 + \\ -(\nu_1 + 3\nu_2)/2 + 4 - 6) = 0$$

$$\Rightarrow \nu_1 = 0 \quad \text{or} \quad \nu_1 = 2 - 2\nu_2$$

$$\nu_2(-(\nu_1 + \nu_2)/2 + 4 + \\ 3(-(\nu_1 + 3\nu_2)/2 + 4) - 4) = 0$$

$$\nu_2 = 0, \quad \nu_1 = (12 - 5\nu_2)/2$$

$$if \quad \nu_1 = 0$$

$$\nu_2 = 12/5$$

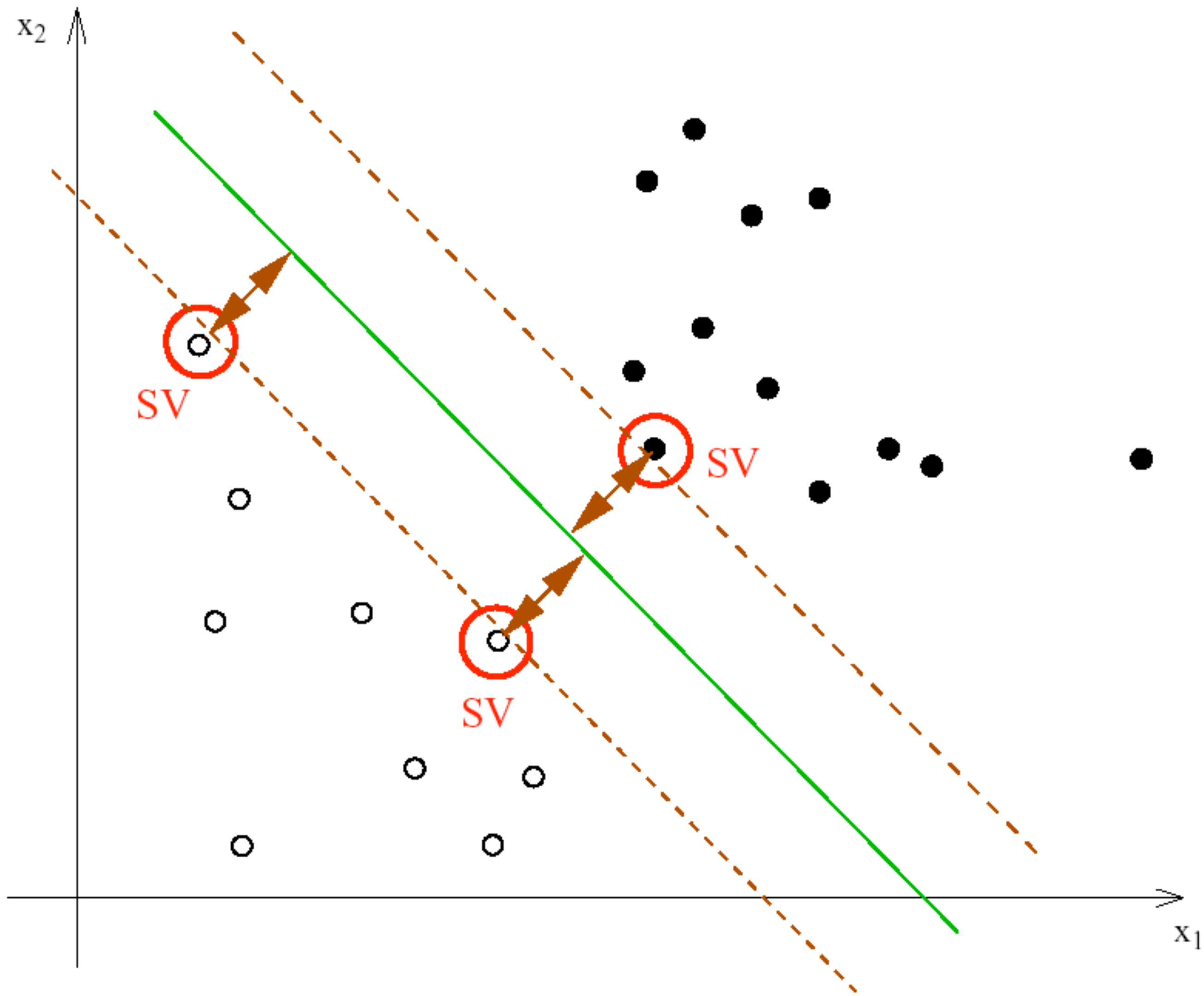
$$if \quad \nu_1 = 2 - 2\nu_2$$

$$\nu_2 = 8$$

$$but \quad if \quad \nu_2 = 0$$

$$\Rightarrow \nu_1 = 2$$

Support Vectors



Now solve SVM problem

Maximizing the margin means minimizing $|\mathbf{w}|$.

But, subject to the inequality constraints:

$C1: \quad y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \quad i = 1, \ldots, n.$

This is constrained optimization and Khun - Tucker gives

$$L_P(\mathbf{w}, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i\left(y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) - 1\right).$$

Taking the derivatives with respect to $w_0, w_1, \ldots, w_p$ and set to zero:

Now solve SVM problem

$$\frac{\partial L_P}{\partial w_j} = \frac{\partial}{\partial w_j}\left[\tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{t=1}^{n}\alpha_i\left(y_t\left(\mathbf{w}^T\mathbf{x}_t + w_0\right) - 1\right)\right] = 0 \quad \text{gives}$$

$$\sum_{t=1}^{n}\alpha_i y_t = 0, \longleftarrow \left(\frac{\partial L}{\partial w_0}\right)$$

$$\left.\begin{aligned} w_1 - \sum_{t=1}^{n}\alpha_i y_t x_{1,t} &= 0 \\ w_2 - \sum_{t=1}^{n}\alpha_i y_t x_{2,t} &= 0 \\ &\vdots \\ w_p - \sum_{t=1}^{n}\alpha_i y_t x_{p,t} &= 0 \end{aligned}\right\} \mathbf{w} = \sum_{t=1}^{n}\alpha_i y_t \mathbf{x}_t$$

Kernel trick

Substitute this in the Lagrangian, to get the *dual form* :

$$L_D = \sum_{t=1}^{n}\alpha_i - \tfrac{1}{2}\sum_{t=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_t y_j \mathbf{x}_t^T\mathbf{x}_j$$

← All we need is inner products!

this quadratic function of $\boldsymbol{\alpha}$ has to be maximized subject to: $\alpha_i \geq 0 \quad \sum_{t=1}^{n}\alpha_i y_t = 0.$

Actual optimization is done by standard general purpose quadratic programmning package.

A point is not allowed to lie within the margin :

$$y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) - 1 \geq 0 \quad i = 1,\ldots,n.$$

In the optimal situation we have :

$$\alpha_i\left(y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) - 1\right) = 0 \quad i = 1,\ldots,n..$$

The Lagrange multipliers α_i are non - negative, so :

if

$$y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) - 1 = 0 \quad \text{(point on the margin)}$$

then $\alpha_i \geq 0$, (*active constraint*) otherwise

$\alpha_i = 0$ (*inactive constraint*).

Points with $\alpha_i \geq 0$ are called *support vectors*

Classification with support vector machines

Once the α_i's have been determined the value of $\mathbf{w}$ can be determined

$$\mathbf{w} = \sum_{t=1}^{n} \alpha_i y_t \mathbf{x}_t = \sum_{t \in SV} \alpha_i y_t \mathbf{x}_t$$

and the value of w_0 can be determined from

$\alpha_i y_t \left(\mathbf{w}^T \mathbf{x}_t + w_0\right) - 1 = 0$ for any t as support vector or as

the average :

$$n_{sv} w_0 + \mathbf{w}^T \sum_{t \in SV} \mathbf{x}_t = \sum_{t \in SV} y_t$$

A new pattern is classified according to the sign of

$\mathbf{w}^T \mathbf{x} + w_0$.

Substituting $\mathbf{w}$ and w_0 gives : assign $\mathbf{x}$ to class ω_1 if

$$\sum_{t \in SV} \alpha_i y_t \mathbf{x}_t^T \mathbf{x} - \frac{1}{n_{sv}} \sum_{t \in SV} \sum_{j \in SV} \alpha_i y_t \mathbf{x}_t^T \mathbf{x}_j + \frac{1}{n_{sv}} \sum_{t \in SV} y_t > 0$$

note : only first term depends on new data pattern $\mathbf{x}$!

Why it is Good to Have Few SVs

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

Theorem [66]: Denote $\#\mathrm{SV}(m)$ the number of SVs obtained by training on m examples randomly drawn from $\mathrm{P}(\mathbf{x}, y)$, and $\mathbf{E}$ the expectation. Then

$$\mathbf{E}\left[\mathrm{Prob(test\ error)}\right] \leq \frac{E\left[\#\mathrm{SV}(m)\right]}{m}$$

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

Nonlinear support vector machines

We seek a discriminant function

$$g(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + w_0$$

with decision rule:

$$\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \quad \mathbf{x} \in \begin{cases} \omega_1 \text{ with corresponding value } y_i = +1 \\ \omega_2 \text{ with corresponding value } y_i = -1 \end{cases}$$

The dual form of the Lagrangian now becomes:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}^T(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}_j)$$

solution (expressed in support vectors):

$$\mathbf{w} = \sum_{i \in SV} \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i)$$

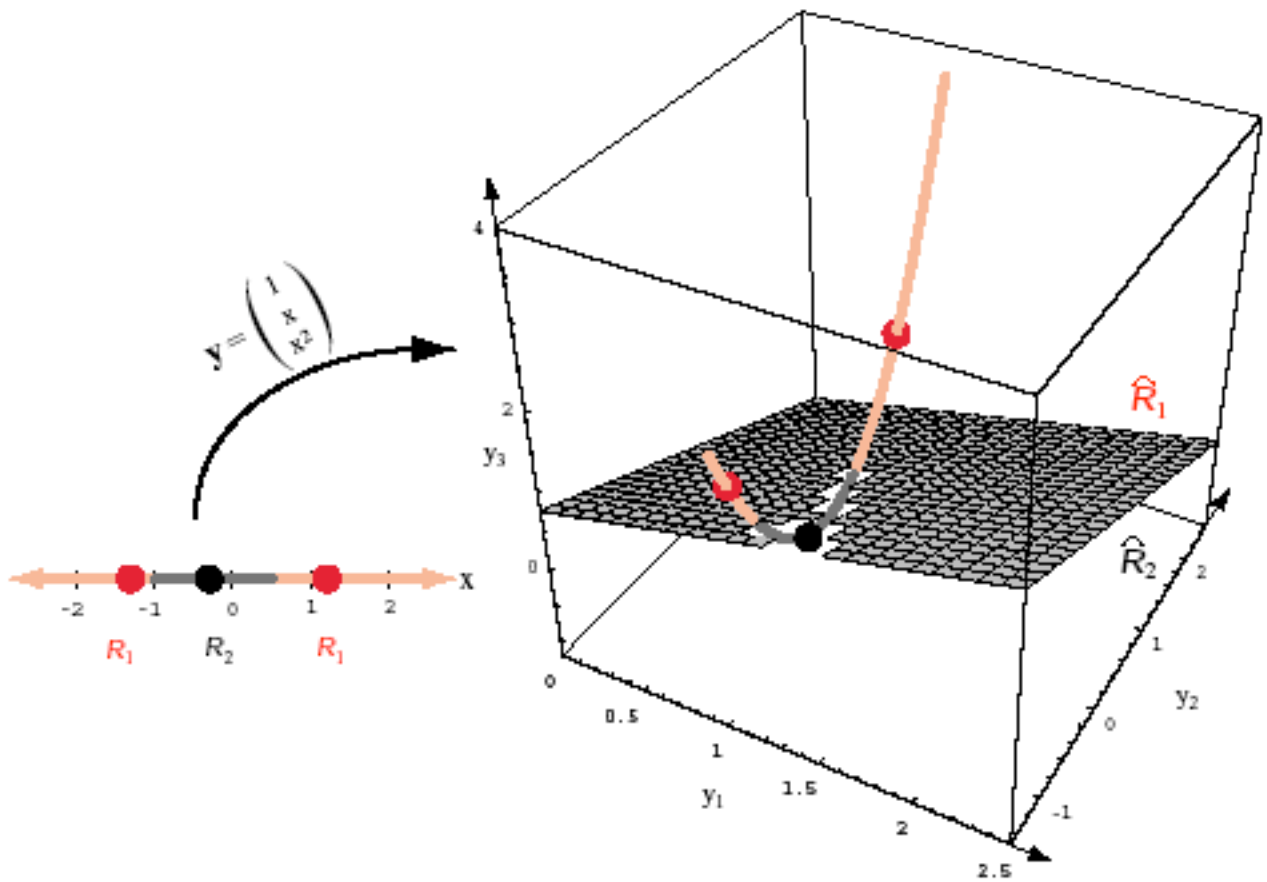


Figure 5.5: The mapping $\mathbf{y} = (1, x, x^2)^t$ takes a line and transforms it to a parabola in three dimensions. A plane splits the resulting $\mathbf{y}$ space into regions corresponding to two categories, and this in turn gives a non-simply connected decision region in the one-dimensional x space.

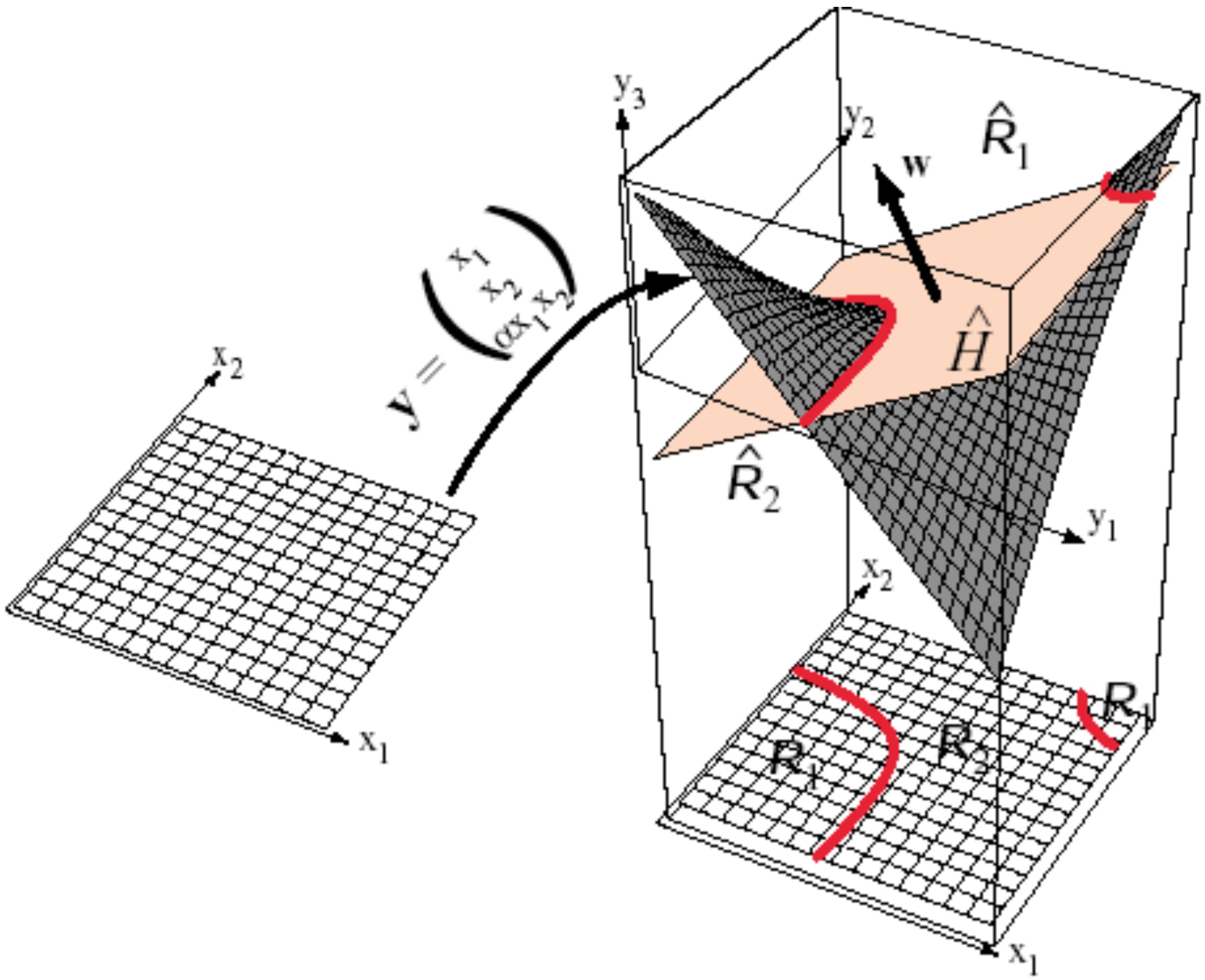


Figure 5.6: The two-dimensional input space $\mathbf{x}$ is mapped through a polynomial function f to $\mathbf{y}$. Here the mapping is $y_1 = x_1$, $y_2 = x_2$ and $y_3 \propto x_1x_2$. A linear discriminant in this transformed space is a hyperplane, which cuts the surface. Points to the positive side of the hyperplane $\hat{H}$ correspond to category ω_1, and those beneath it ω_2. Here, in terms of the $\mathbf{x}$ space, $\mathcal{R}_1$ is a not simply connected.

Kernels and Feature Spaces

Preprocess the data with

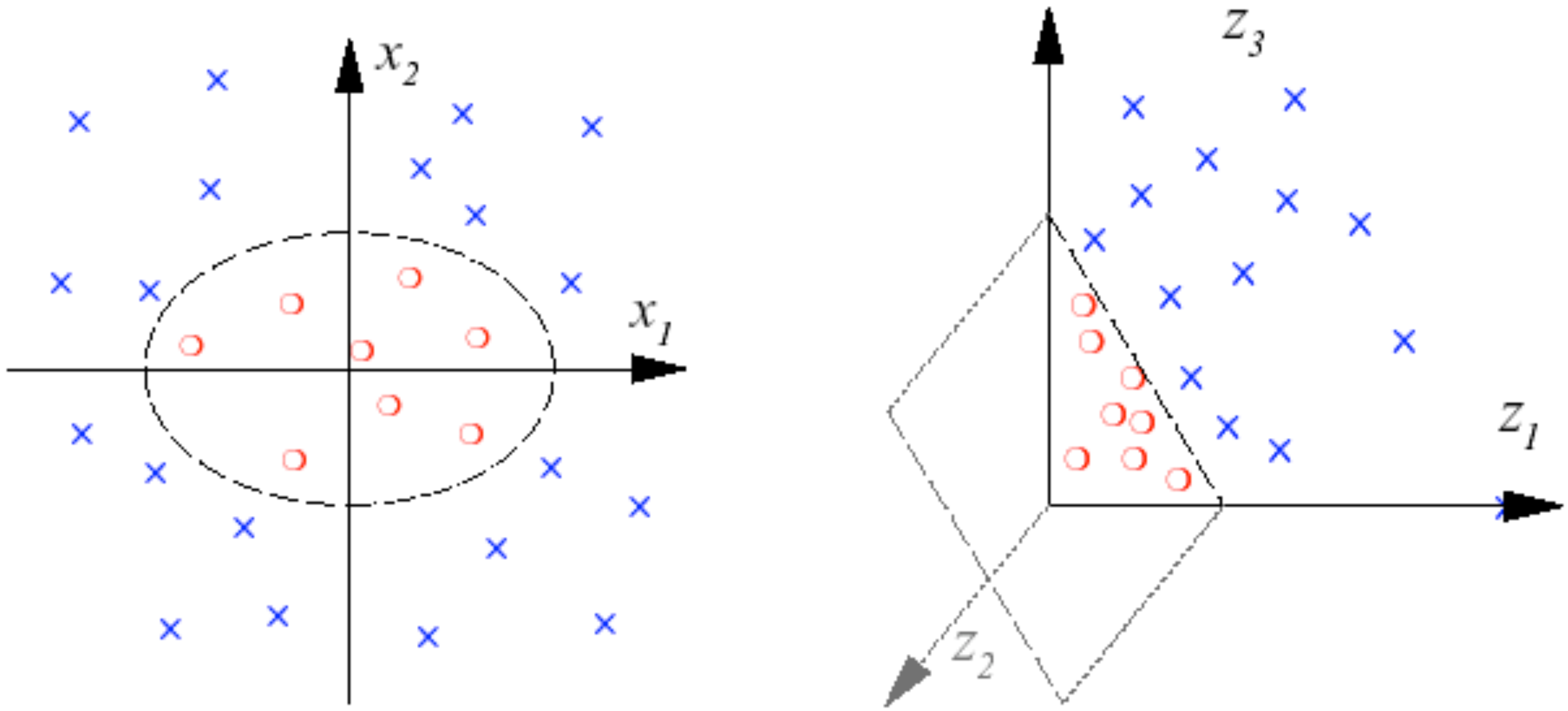
$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \Phi(x), \end{aligned}$$

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

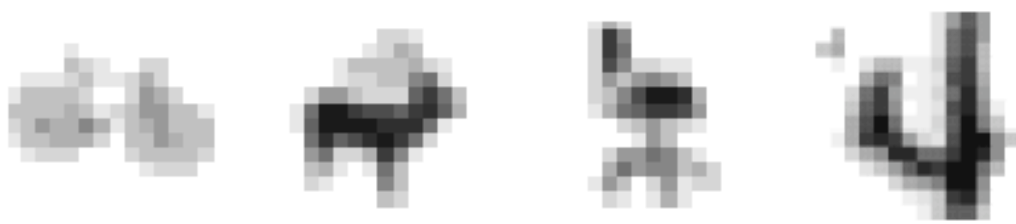
- usually, $\dim(\mathcal{X}) \ll \dim(\mathcal{H})$
- "Curse of Dimensionality"?
- crucial issue: *capacity*, not *dimensionality*

Example: All Degree 2 Monomials

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



General Product Feature Space



How about patterns $x \in \mathbb{R}^N$ and product features of order d?

Here, $\dim(\mathcal{H})$ grows like N^d.

E.g. $N = 16 \times 16$, and $d = 5 \longrightarrow$ dimension 10^{10}

The Kernel Trick, N=2, d=2

$$\Phi(\vec{x}) = \left[x_1^2, \sqrt{2}x_1x_2, x_2^2\right]$$

$$\begin{aligned}
(< x,z >)^2 &= (x_1z_1 + x_2z_2)^2 \\
&= (x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2) \\
&= \left\langle [x_1^2, \sqrt{2}x_1x_2, x_2^2],[z_1^2, \sqrt{2}z_1z_2, z_2^2] \right\rangle \\
&= \left\langle \Phi(\vec{x}), \Phi(\vec{z}) \right\rangle \\
&= K\left(\vec{x}, \vec{z}\right)
\end{aligned}$$

- Thus the dot product in the non-linear feature space can be computed in $\Re^2$ via the kernel function.

The Kernel Trick, II

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

$$\begin{aligned}
\langle x, x' \rangle^d &= \left(\sum_{j=1}^{N} x_j \cdot x'_j \right)^d \\
&= \sum_{j_1,\ldots,j_d=1}^{N} x_{j_1} \cdot \cdots \cdot x_{j_d} \cdot x'_{j_1} \cdot \cdots \cdot x'_{j_d} = \left\langle \Phi(x), \Phi(x') \right\rangle,
\end{aligned}$$

where Φ maps into the space spanned by all ordered products of d input directions

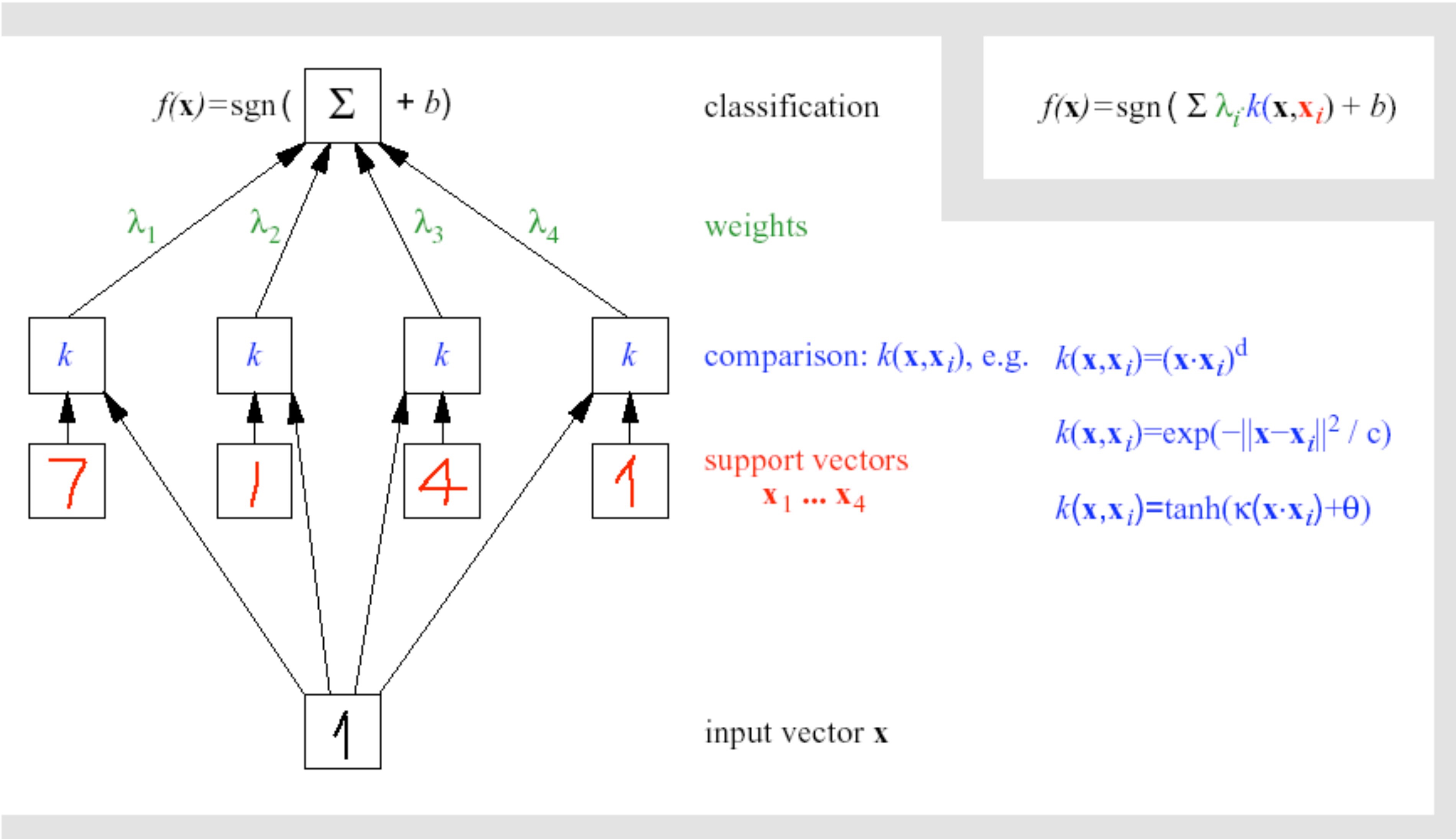
The Kernel Trick — Summary

- *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:

$$\begin{aligned} \text{Polynomial} \quad k(x, x') &= (\langle x, x' \rangle + c)^d \\ \text{Sigmoid} \quad k(x, x') &= \tanh(\kappa \langle x, x' \rangle + \Theta) \\ \text{Gaussian} \quad k(x, x') &= \exp(-\|x - x'\|^2 / (2\,\sigma^2)) \end{aligned}$$

- Kernel are studied also in the Gaussian Process prediction community (covariance functions) [71, 68, 72, 40] —
course

The SVM Architecture



Classification

A new pattern is classified according to the sign of

$\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + w_0.$

Substituting $\mathbf{w}$ gives:

$$g(\mathbf{x}) = \sum_{i \in SV} \alpha_{\mathrm{i}} y_i \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}) + w_0, \text{ in which}$$

$$w_0 = \frac{1}{N_{\tilde{S}V}} \left\{ \sum_{i \in SV} y_i - \sum_{\mathrm{i} \in SV, j \in SV} \alpha_{\mathrm{i}} y_i \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) \right\}.$$

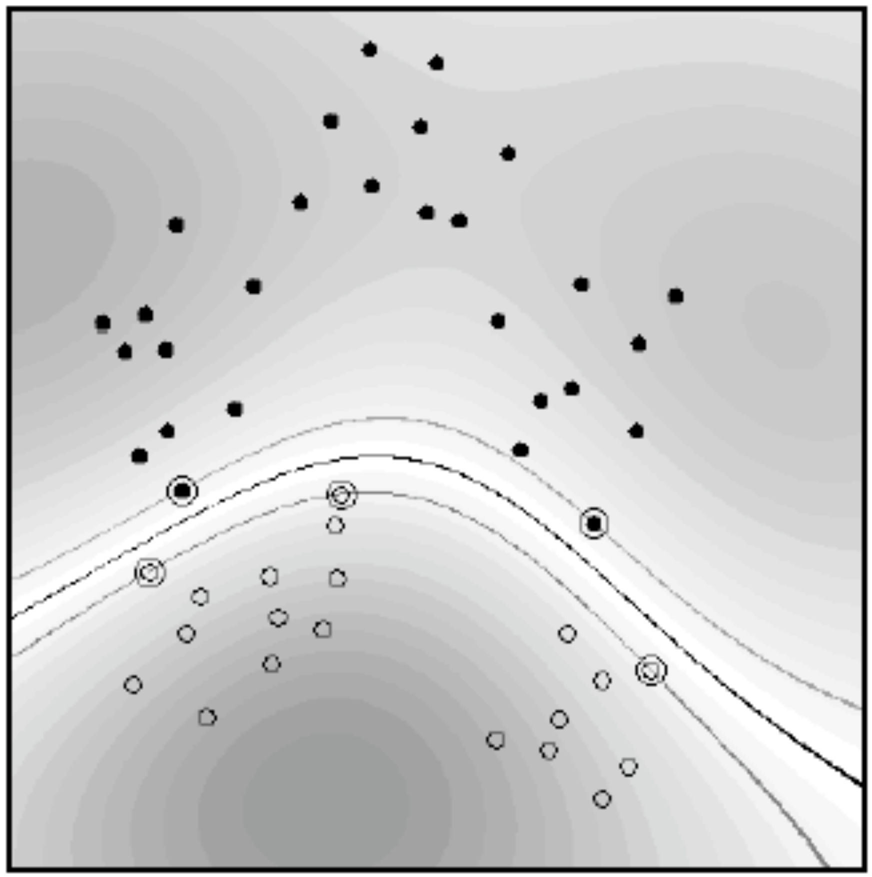
Note that classification depends only on inner products of transformed feature vectors $\boldsymbol{\phi}(\mathbf{x})$.

Some feature spaces come with a kernel $\mathbf{K}$ (or vice versa) such that:

$K(\mathbf{x}, \mathbf{y}) = \boldsymbol{\phi}^T(\mathbf{x})\boldsymbol{\phi}(\mathbf{y}).$

Toy Example with Gaussian Kernel

$$k(x, x') = \exp\left(-\|x - x'\|^2\right)$$



Simple example (XOR problem)

$$\Phi(w) = \tfrac{1}{2} w^T w$$

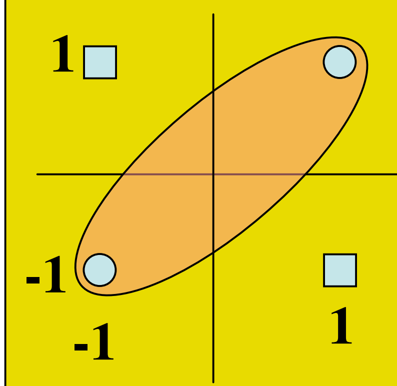
$$L(w,b,\alpha) = \tfrac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i [y_i (w^T x_i + b) - 1]$$

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \varphi(x_i)^T \varphi(x_j)$$

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right)$$

$$\varphi(x) = [1, x_1^2, \sqrt{2} x_1 x_2, x_2^2, \sqrt{2} x_1, \sqrt{2} x_2]^T$$

$$K(x, x_i) = (1 + x^T x_i)^2$$
$$= 1 + x_1^2 x_{i1}^2 + 2 x_1 x_{i1} x_2 x_{i2} + x_2^2 x_{i2}^2 + 2 x_1 x_{i1} + 2 x_2 x_{i2}$$



Input vec.	**y**
[-1,-1]	-1
[-1,+1]	+1
[+1,-1]	+1
[+1,+1]	-1

K evaluated for all pairs of inputs:

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

Simple example(cont.)

Dual formulation

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$$
$$- \tfrac{1}{2}(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1 + 9\alpha_2^2$$
$$+ 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_2^4)$$

$$9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1$$
$$-\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1$$
$$-\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1$$
$$\alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$$

$$\alpha_{o,1} = \alpha_{o,2} = \alpha_{o,3} = \alpha_{o,4} = \tfrac{1}{8}$$

$$Q_o(\alpha) = \tfrac{1}{4}$$

Four Input vectors are All support vectors

$$\frac{1}{2}\|w_o\|^2 = \frac{1}{4}, \|w_o\| = \frac{1}{\sqrt{2}}$$

$$w_o = \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)$$

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.