Machine Learning

Session 7 Support Vector Machines

Optimal separating Hyperplanes

SVM: Primal and dual formulations

Example

The non separable case

Non Linear svm

Example

From

S. Theordiridis, K.Koutroumbas. Pattern Recognition Elsevier 2009, 3.7.{1,2}

Slides of Ricardo Gutierrez-Osuna

Bishop, 7, 7.1

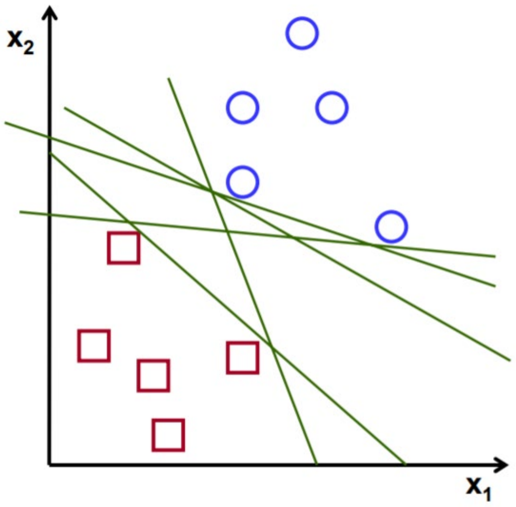
'Optimal' Separating Hyperplane

- Problem Statement

  - Consider the problem of finding an **optimal** separating hyperplane for a **linearly separable** dataset.

$$\{(\mathbf{x}^{(n)}, y^{(n)})\}, n = 1, \dots, N \qquad \mathbf{x}^{(n)} \in \mathbb{R}^D, y^{(n)} \in \{+1, -1\}$$

  - Which hyperplane would you choose?



  - The hyperplane is:

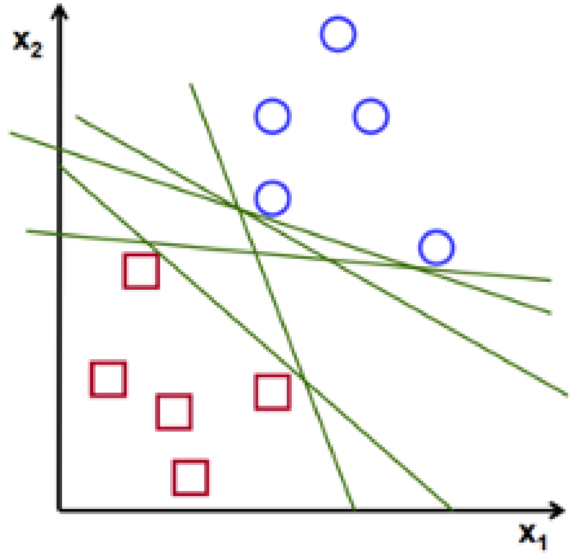
$$\mathbf{w}^\top \mathbf{x} + b = 0$$

  - The discriminant function is

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

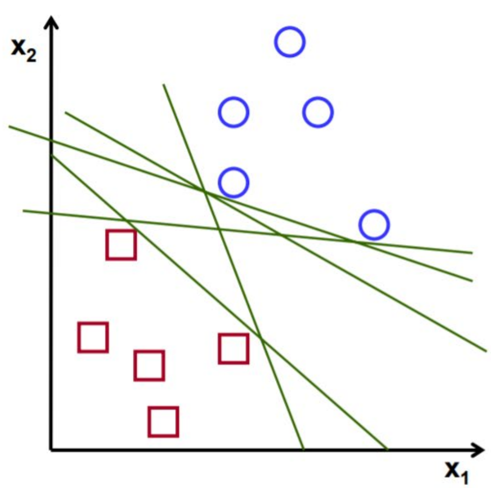
Optimal Separating Hyperplane

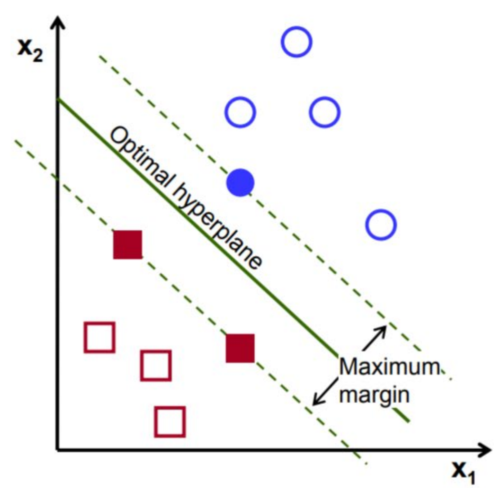
- Problem Statement
  - Which hyperplane would you choose?
    - A hyperplane that passes too close to the training examples will be sensitive to noise and less likely to generalize well for unseen data
    - Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will generalize better



Optimal Separating Hyperplane

- Problem Statement
  - Which hyperplane would you choose?
    - A hyperplane that passes too close to the training examples will be sensitive to noise and less likely to generalize well for unseen data
    - Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will generalize better
  - The optimal separating hyperplane will be the one with the largest **margin**, which is defined as two times the minimum distance of an example to the decision surface.





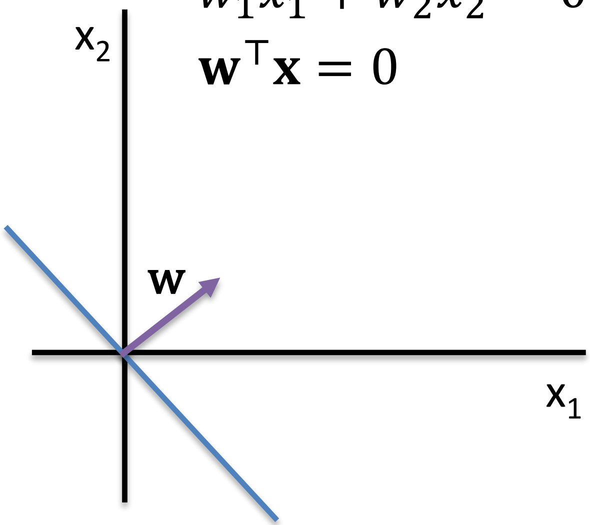
Optimal Separating Hyperplane

- **Geometry:** let's express the margin as a function of the weight vector and bias of the separating hyperplane

Remember:

$w_1x_1 + w_2x_2 = 0$

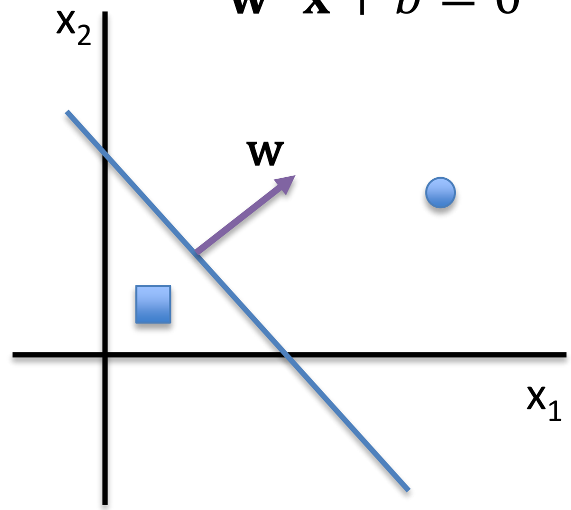
$\mathbf{w}^\top\mathbf{x} = 0$



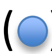
$\mathbf{x}$ are the points orthogonal to $\mathbf{w}$

$w_1x_1 + w_2x_2 + b = 0$

$\mathbf{w}^\top\mathbf{x} + b = 0$



The discriminant is $g(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$

g()>0 class +1

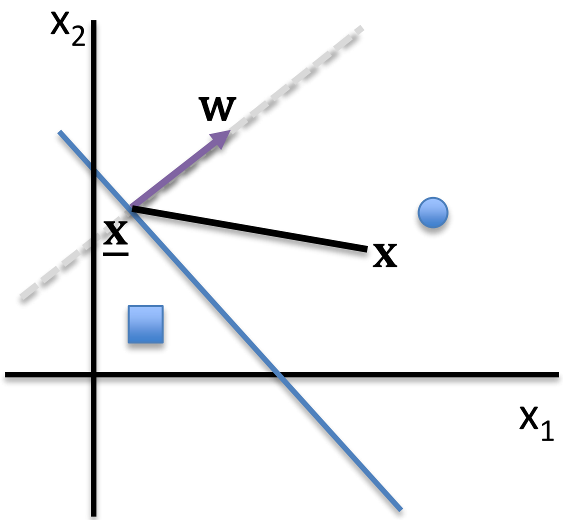
g()<0 class -1

With this language $\mathbf{x}$ could be of D dimensions

Optimal Separating Hyperplane

- **Geometry:** let's express the margin as a function of the weight vector and bias of the separating hyperplane.
- Step 1: The distance between a point $\mathbf{x}$ and a hyperplane $(\mathbf{w}, b)$ is

$$\frac{|\mathbf{w}^\top \mathbf{x} + b|}{||\mathbf{w}||}$$



Project the difference between $\mathbf{x}$ and a point $\underline{\mathbf{x}}$ in the hyperplane

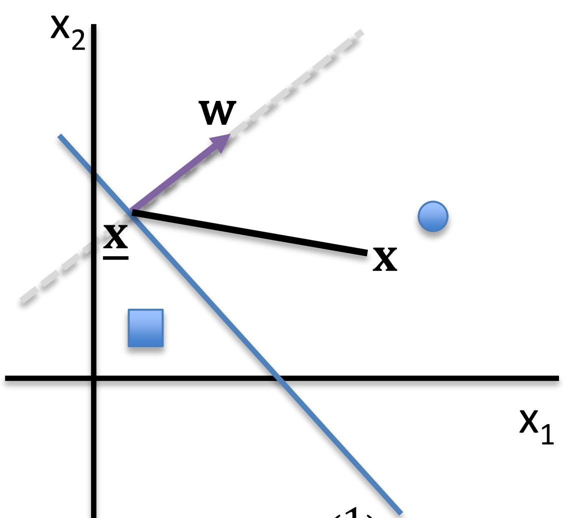
$$\frac{\mathbf{w}^\top}{||\mathbf{w}||}(\mathbf{x} - \underline{\mathbf{x}}) = \frac{\mathbf{w}^\top \mathbf{x}}{||\mathbf{w}||} - \frac{\mathbf{w}^\top \underline{\mathbf{x}}}{||\mathbf{w}||} = \frac{\mathbf{w}^\top \mathbf{x}}{||\mathbf{w}||} + \frac{b}{||\mathbf{w}||} = \frac{\mathbf{w}^\top \mathbf{x} + b}{||\mathbf{w}||}$$

$\underline{\mathbf{x}}$ in the hyperplane $\mathbf{w}^\top \underline{\mathbf{x}} + b = 0$, then: $\mathbf{w}^\top \underline{\mathbf{x}} = -b$

Optimal Separating Hyperplane

- **Geometry:** let's express the margin as a function of the weight vector and bias of the separating hyperplane.
- Step 1: The distance between a point $\mathbf{x}$ and a hyperplane $(\mathbf{w}, b)$ is

$$\frac{|\mathbf{w}^\top \mathbf{x} + b|}{||\mathbf{w}||}$$



Project the difference between $\mathbf{x}$ and a point $\underline{\mathbf{x}}$ in the hyperplane

$$\frac{\mathbf{w}^\top}{\|\mathbf{w}\|}(\mathbf{x} - \underline{\mathbf{x}}) = \frac{\mathbf{w}^\top \mathbf{x}}{\|\mathbf{w}\|} - \frac{\mathbf{w}^\top \underline{\mathbf{x}}}{\|\mathbf{w}\|} = \frac{\mathbf{w}^\top \mathbf{x}}{\|\mathbf{w}\|} + \frac{b}{\|\mathbf{w}\|} = \frac{\mathbf{w}^\top \mathbf{x} + b}{\|\mathbf{w}\|}$$

$\underline{\mathbf{x}}$ in the hyperplane $\mathbf{w}^\top \underline{\mathbf{x}} + b = 0$, then: $\mathbf{w}^\top \underline{\mathbf{x}} = -b$

For example $\mathbf{x}^{(1)} = \binom{1}{1}$ and the hyperplane $x_1 + x_2 - 1 = 0$;

$$\mathrm{dist}\left(\mathbf{x}^{(1)}, x_1 + x_2 - 1 = 0\right) = \left|\frac{(1,1)\binom{1}{1} - 1}{\sqrt{1^2 + 1^2}}\right| = \left|\frac{1}{\sqrt{2}}\right| = \left|\frac{\sqrt{2}}{2}\right|$$

And $\mathbf{x}^{(2)} = \binom{0}{0}$ and the same hyperplane, $\mathrm{dist}\left(\mathbf{x}^{(2)}, x_1 + x_2 - 1 = 0\right) = \left|\frac{(1,1)\binom{0}{0} - 1}{\sqrt{1^2+1^2}}\right| = \left|-\frac{\sqrt{2}}{2}\right|$

(do the figure!)

Optimal Separating Hyperplane

- **Geometry:** let's express the margin as a function of the weight vector and bias of the separating hyperplane.
- Step 1: The distance point $\mathbf{x}$ and a hyperplane $(\mathbf{w}, b)$ is $\frac{|\mathbf{w}^\top \mathbf{x}+b|}{||\mathbf{w}||}$
- Step 2: the distance point-hyperplane is unique, but we can use several equations for the hyperplane. We impose that for the closest points to the hyperplane, we get $\mathbf{w}^\top \mathbf{x} + b = 1$. This is the **Canonical Hyperplane**

Optimal Separating Hyperplane

- **Geometry:** let's express the margin as a function of the weight vector and bias of the separating hyperplane.
- Step 1: The distance point $\mathbf{x}$ and a hyperplane $(\mathbf{w}, b)$ is $\frac{|\mathbf{w}^\top \mathbf{x} + b|}{||\mathbf{w}||}$
- Step 2: the distance point-hyperplane is unique, but we can use several equations for the hyperplane. We impose that for the closest points to the hyperplane, we get $\mathbf{w}^\top \mathbf{x} + b = 1$. This is the **Canonical Hyperplane**

For example, if the closest point to $\mathrm{x}_1 + x_2 - 1 = 0$ is $\mathbf{x}^{(n)} = \binom{2}{2}$, what would be the canonical hyperplane?

- $(1,1)\binom{2}{2} - 1 = 3$ and we want 1 as the result. We take:

$\left(\frac{1}{3}, \frac{1}{3}\right)\binom{2}{2} - \frac{1}{3} = \frac{3}{3}$ The canonical hyperplane will be $\boxed{\frac{1}{3}x_1 + \frac{1}{3}x_2 - \frac{1}{3} = 0}$

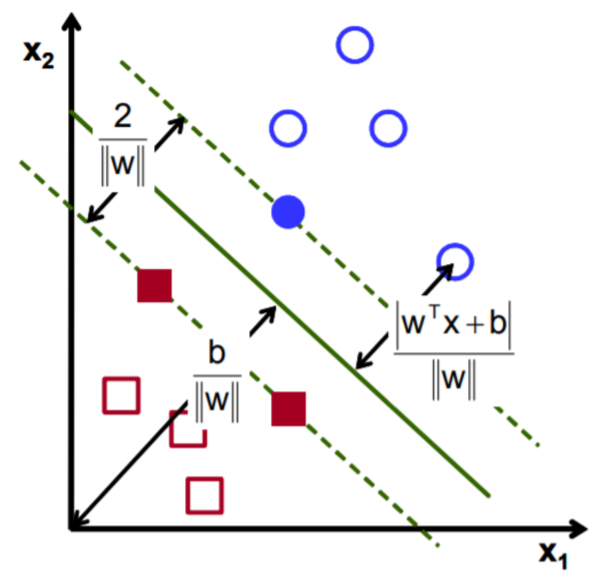
- With these two steps, the two closest points to this canonical hyperplane will have distance $\frac{1}{||\mathbf{w}||}$ and the margin will be: $\boxed{m = \frac{2}{||\mathbf{w}||}}$

Optimal Separating Hyperplane

- Geometry
  - *We choose the solution for which the discriminant function becomes one for the training examples closest to the boundary*

$$|\mathbf{w}^\top \mathbf{x} + b| = 1$$

This is known as the **canonical hyperplane**



  - Therefore, the distance from the closest example to the boundary is $\frac{|\mathbf{w}^\top \mathbf{x}+b|}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}$,
  - And the **margin becomes**

$$m = \frac{2}{||\mathbf{w}||}$$

We estimate $\mathbf{w}$ in such a way that the margin becomes largest.
Equivalently: minimize $||\mathbf{w}||$ such that it classifies well all the points

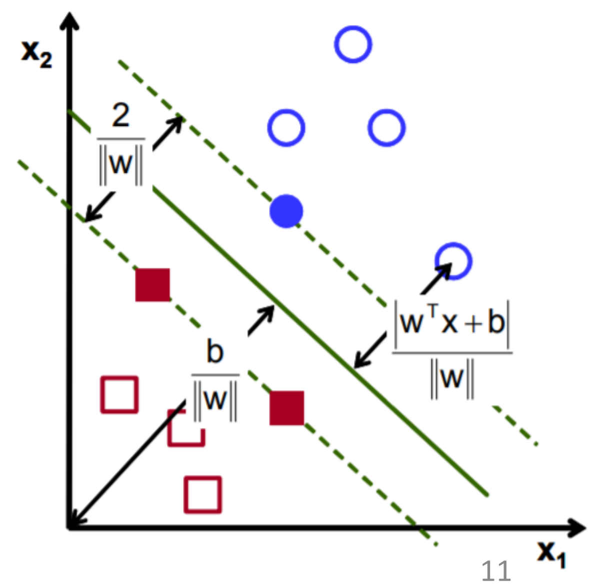
Optimal Separating Hyperplane

- **Geometry** We want a $\mathbf{w}$ minimum such that classifies well all the points.
- Step 3: Convert each point in a restriction:
  - If $y^{(n)} = +1$, we want $\mathbf{w}^\top \mathbf{x}^{(n)} + b \geq +1$, so we impose: $y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1$
  - If $y^{(n)} = -1$, we want $\mathbf{w}^\top \mathbf{x}^{(n)} + b \leq -1$, so we impose: $y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1$

We want the minimum of $\|\mathbf{w}\| = \sqrt{x_1^2 + \cdots + x_D^2}$ subject to a constraint (one constraint for each point)

$$y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1, \quad n = 1 \dots N$$

Once found $\mathbf{w}$, the **canonical** hyperplane is calculated and this discriminant function is our classifier



The Optimization Problem

- The problem of maximizing the margin $m = \frac{2}{||\mathbf{w}||}$ is equivalent to

$$\text{minimize} \quad J(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to} \quad y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1, \qquad \forall n = 1, \dots, N$$

  - $J(\mathbf{w})$ 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 $\Omega \subset \mathbb{R}^D$

$$
\begin{array}{lll}
\text{minimize} & f(\mathbf{z}) & \mathbf{z} \in \Omega \\
\text{subject to} & g_n(\mathbf{z}) \leq 0 & n \in 1, \dots, N \\
 & h_m(\mathbf{z}) = 0 & m \in 1, .., M
\end{array}
$$

  – With $f \in C^1$ convex and g_n, h_m affine, necessary and sufficient conditions for a normal point $\mathbf{z}^*$ to be an optimum are the existence of $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ such that

$$
\begin{array}{ll}
\dfrac{\partial \mathcal{L}(\mathbf{z}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{z}} = 0 & \\
\dfrac{\partial \mathcal{L}(\mathbf{z}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0 & \\
\alpha_n^* g_n(\mathbf{z}^*) = 0 & n \in 1, \dots, N \\
g_n(\mathbf{z}^*) \leq 0 & n \in 1, \dots, N \\
\alpha_n^* \geq 0 & n \in 1, \dots, N
\end{array}
$$

where $\mathcal{L}(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{z}) + \sum_{n=1}^{N} \alpha_n g_n(\mathbf{z}) + \sum_{m=1}^{M} \beta_m h_m(\mathbf{z})$ is known as *generalized Lagrangian function*

Kuhn-Tucker Theorem

- The third condition: $\alpha_n^* g_n(\mathbf{z}^*) = 0, \quad n \in 1, \dots, N$

  is known as the Karush-Kuhn-Tucker (KKT) complementary condition. It implies that

  for active constraints $\rightarrow \alpha_n^* \geq 0$

  for inactive constraints $\rightarrow \alpha_n^* = 0$

Kuhn-Tucker Theorem

- The third condition: $\alpha_n^* g_n(\mathbf{z}^*) = 0, \quad n \in 1, \ldots, N$

  is known as the Karush-Kuhn-Tucker (KKT) complementary condition. It implies that

  for active constraints $\rightarrow \alpha_n^* \geq 0$
  for inactive constraints $\rightarrow \alpha_n^* = 0$

- The KKT condition will allows us to identify the training examples that define the largest margin hyperplane
  - For these examples, $\alpha_n^* \geq 0$ and they are known as **Support Vectors**
  - For the rest of examples, $\alpha_n^* = 0$

The Lagrangian dual problem (1/4)

- Constrained minimization of $J(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$ is solved by introducing the Lagrangian

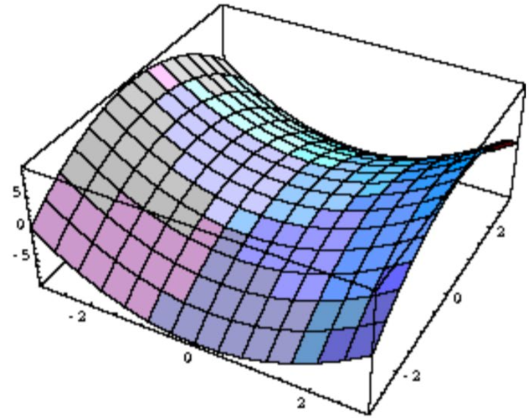
$$L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} \alpha_n (y^{(n)} (\mathbf{w}^\top \mathbf{x}^{(n)} + b) - 1)$$

Which yields an unconstrained optimization problem that is solved by:

- Minimizing L_p with respect to the primal variables $\mathbf{w}, b$ AND
- Maximizing L_p with respect to the dual variables α_n

Thus, the optimum is defined by a saddle point (see below for illustration)

This is known as the **Lagrangian primal problem**



A saddle point

The Lagrangian dual problem (2/4)

- To simplify the primal problem, we eliminate the primal variables $\mathbf{w}, b$ using the first Kuhn-Tucker condition $\frac{\partial \mathcal{L}}{\partial \mathbf{z}} = 0$ on:

$$\boxed{L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} \alpha_n \left(y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) - 1\right)}$$

- Expansion of L_p yields

$$L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{w}^\top \mathbf{x}^{(n)} - b \sum_{n=1}^{N} \alpha_n y^{(n)} + \sum_{n=1}^{N} \alpha_n$$

- Differentiating $L_p(\mathbf{w}, b, \boldsymbol{\alpha})$ with respect to $\mathbf{w}, b$, and setting to zero yields

$$\frac{\partial L_p(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{x}^{(n)}$$

$$\frac{\partial L_p(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \rightarrow \sum_{n=1}^{N} \alpha_n y^{(n)} = 0$$

The Lagrangian dual problem (3/4)

$$L_p(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^\top\mathbf{w} - \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{w}^\top \mathbf{x}^{(n)} - b \sum_{n=1}^{N} \alpha_n y^{(n)} + \sum_{n=1}^{N} \alpha_n$$

- Using the optimality condition $\frac{\partial J}{\partial \mathbf{w}} = 0$, the **first** term in L_p can be expressed as

$$\boxed{\mathbf{w} = \sum_{m=1}^{N} \alpha_m y^{(m)} \mathbf{x}^{(m)}}$$

$$\mathbf{w}^\mathrm{T}\mathbf{w} = \mathbf{w}^\mathrm{T} \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{x}^{(n)} = \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{w}^\top \mathbf{x}^{(n)}$$

$$= \sum_{n=1}^{N} \alpha_n y^{(n)} \left(\sum_{m=1}^{N} \alpha_m y^{(m)} \mathbf{x}^{(m)} \right)^\top \mathbf{x}^{(n)} = \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)\top} \mathbf{x}^{(n)}$$

- The second term in L_p can be expressed in the same way
- The third term in L_p is zero by virtue of the optimality condition $\frac{\partial J}{\partial b} = 0$

The Lagrangian dual problem (4/4)

- Merging these expressions together we obtain

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)}$$

Subject to the simpler constraints $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$

[the 1st term is the previous 3rd term and the 2nd one comes from $({}^1/_2 - 1)\mathbf{w}^\top \mathbf{w}$]

- This is known as the **Lagrangian dual problem**

The Lagrangian dual problem (4/4)

- Merging these expressions together we obtain

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)}$$

Subject to the simpler constraints $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$

[the 1st term is the previous 3rd term and the 2nd one comes from $({}^1/_2 - 1)\mathbf{w}^\top\mathbf{w}$]

- This is known as the **Lagrangian dual problem**
- Remarks:
  - We have transformed the problem of finding a saddle point for $L_p(\mathbf{w}, b, \boldsymbol{\alpha})$ into the easier one of maximizing $L_D(\boldsymbol{\alpha})$. Notice that $L_D(\boldsymbol{\alpha})$ depends on the Lagrange multipliers $\boldsymbol{\alpha}$, but it **does not** depend on $(\mathbf{w}, b)$

The Lagrangian dual problem (4/4)

- Merging these expressions together we obtain

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)}$$

Subject to the simpler constraints $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$

[the 1st term is the previous 3rd term and the 2nd one comes from $({}^1/_2 - 1)\mathbf{w}^\top\mathbf{w}$]

- This is known as the **Lagrangian dual problem**
- Remarks:
  - We have transformed the problem of finding a saddle point for $L_p(\mathbf{w}, b, \boldsymbol{\alpha})$ into the easier one of maximizing $L_D(\boldsymbol{\alpha})$. Notice that $L_D(\boldsymbol{\alpha})$ depends on the Lagrange multipliers $\boldsymbol{\alpha}$, but it **does not** depend on $(\mathbf{w}, b)$
  - The **primal problem scales with dimensionality D** ($\mathbf{w}$ has one coefficient for each dimension), whereas **the dual problem scales with N**, the amount of training data (there is one Lagrange multiplier α_n per example)

The Lagrangian dual problem (4/4)

- Merging these expressions together we obtain

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)}$$

Subject to the simpler constraints $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$

[the 1st term is the previous 3rd term and the 2nd one comes from $({}^1/_2 - 1)\mathbf{w}^\top\mathbf{w}$]

- This is known as the **Lagrangian dual problem**
- Remarks:
  - We have transformed the problem of finding a saddle point for $L_p(\mathbf{w}, b, \boldsymbol{\alpha})$ into the easier one of maximizing $L_D(\boldsymbol{\alpha})$. Notice that $L_D(\boldsymbol{\alpha})$ depends on the Lagrange multipliers $\boldsymbol{\alpha}$, but it **does not** depend on $(\mathbf{w}, b)$
  - The **primal problem scales with dimensionality D** ($\mathbf{w}$ has one coefficient for each dimension), whereas **the dual problem scales with N**, the amount of training data (there is one Lagrange multiplier α_n per example)
  - Moreover, the training data appears only as dot products $\mathbf{x}_n^\top \mathbf{x}_m$

Support Vectors

- The KTT complementary condition states that, for every point in the training set, the following equality must hold

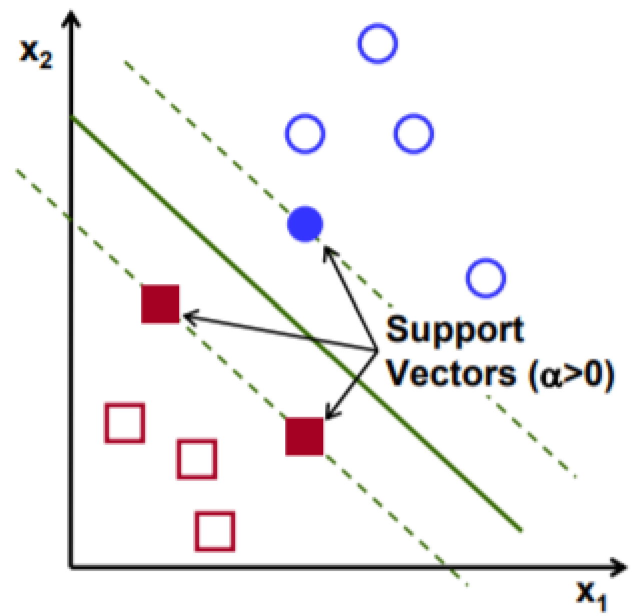
$$\alpha_n\left(y^{(n)}\left(\mathbf{w}^\top\mathbf{x}^{(n)}+b\right)-1\right)=0 \qquad \forall n=1,\dots,N$$

- Therefore, for each example, either $\alpha_n = 0$ or $y^{(n)}\left(\boldsymbol{w}^\top\mathbf{x}^{(n)} + b - 1\right) = 0$ must hold

- Those points for which $\alpha_n > 0$ must then lie on one of the two hyperplanes that define the largest margin (only at these hyperplanes the term $y^{(n)}\left(\mathbf{w}^\top\mathbf{x}^{(n)} + b - 1\right)$ becomes zero)

  These points are known as **Support Vectors**

- All the other must have $\alpha_n = 0$



Support Vectors

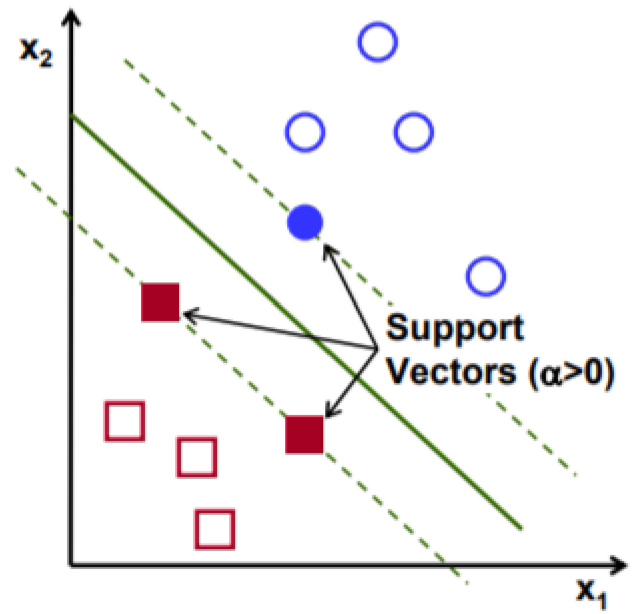
- The KTT complementary condition states that, for every point in the training set, the following equality must hold

$$\alpha_n\left(y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) - 1\right) = 0 \qquad \forall n = 1, \dots, N$$

- Therefore, for each example, either $\alpha_n = 0$ or $y^{(n)}\left(\boldsymbol{w}^\top \mathbf{x}^{(n)} + b - 1\right) = 0$ must hold

- Note that only the support vectors contribute to defining the optimal hyperplane

$$\frac{\partial J(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{x}^{(n)}$$



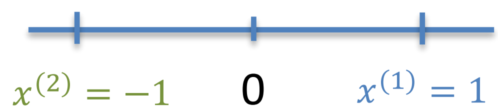
- The bias b is found from the KKT complementary condition on the support vectors
- Therefore, the complete dataset could be replaced by only the support vectors

Example

- **Example**: $\{(x^{(1)} = 1, y^{(1)} = +1), (x^{(2)} = -1, y^{(2)} = -1)\}$
  We want the the maxim margin hyperplane that separate both classes.
- Plot the situation

Example

- **Example**: $\{(x^{(1)} = 1, y^{(1)} = +1), (x^{(2)} = -1, y^{(2)} = -1)\}$
  We want the the maxim margin hyperplane that separate both classes.
- Plot the situation



- Model of the classifier for this case: $g(x) = wx + b$
- w will be the solution to: $\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 \rightarrow \min_{w} \frac{w^2}{2}$

  subject to: $y^{(n)}(\mathbf{w}^\top \mathbf{x}^{(n)} + b) \geq 1$ for $\mathrm{n} = 1$ and $\mathrm{n} = 2$

  $$+1(w \cdot 1 + b) \geq 1 \rightarrow w + b - 1 \geq 0$$
  $$-1(w \cdot (-1) + b) \geq 1 \rightarrow w - b - 1 \geq 0$$

- Primal

(note we put minus – before all constraints)

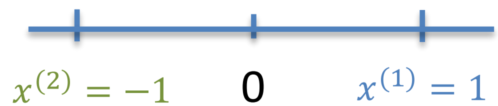
$$L_p(w, b, \alpha_1, \alpha_2) = \frac{w^2}{2} - \alpha_1(w + b - 1) - \alpha_2(w - b - 1)$$

$\frac{\partial L_p}{\partial w} = w - \alpha_1 - \alpha_2 = 0 \implies w = \alpha_1 + \alpha_2$ (in general : $\mathbf{w} = \sum_{m=1}^{N} \alpha_m y^{(m)} \mathbf{x}^{(m)}$)

$\frac{\partial L_p}{\partial b} = -\alpha_1 + \alpha_2 = 0 \implies \alpha_1 = \alpha_2$

Example

- **Example**: $\{(x^{(1)} = 1, y^{(1)} = +1), (x^{(2)} = -1, y^{(2)} = -1)\}$
  We want the the maxim margin hyperplane that separate both classes
- Plot the situation



We have: $w = \alpha_1 + \alpha_2$ and $\alpha_1 = \alpha_2$ and $w = 2\alpha_1$

To obtain the dual $L_D(\alpha_1, \alpha_2)$ we need to substitute the previous result in

$$L_p(w, b, \alpha_1, \alpha_2) = \frac{w^2}{2} - \alpha_1(w + b - 1) - \alpha_2(w - b - 1)$$

$$L_D(\alpha_1, \alpha_2) = \frac{2\alpha_1}{2} - \alpha_1(2\alpha_1 + b - 1) - \alpha_1(2\alpha_1 - b - 1) = 2\alpha_1^2 - 4\alpha_1 + 2\alpha_1 = -2\alpha_1^2 + 2\alpha_1$$

$$\frac{\partial L_D(\alpha_1, \alpha_2)}{\partial \alpha_1} = -4\alpha_1 + 2 = 0 \rightarrow \alpha_1 = \frac{1}{2}$$

Then $w = \alpha_1 + \alpha_2 = \frac{1}{2} + \frac{1}{2} = 1 \implies$ **both points are Support Vectors**

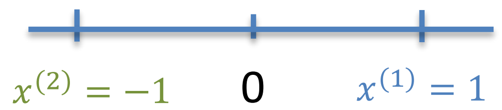
- The classifier will be $g(x) = 1x + b$

$$\left.\begin{array}{l} \text{over the support } x^{(1)} : g\left(x^{(1)}\right) = g(1) = 1 + b = 1 \\ \text{over the support } x^{(2)} : g\left(x^{(2)}\right) = g(-1) = -1 + b = -1 \end{array}\right\} \quad \text{Then b=0}$$

The classifier will be $g(x) = x$ **and the Margin** $\frac{2}{||1||} = 2$

Example

- **Example**: $\{(x^{(1)} = 1, y^{(1)} = +1), (x^{(2)} = -1, y^{(2)} = -1)\}$
  We want the the maxim margin hyperplane that separate both classes
- Plot the situation



There is an alternative way to obtain the result, using only the dual and the two results of the primal

From the primal we know:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} x^{(n)} = (+1)\alpha_1(+1) + (-1)\alpha_2(-1) = \alpha_1 + \alpha_2$$

$$\sum_{n=1}^{N} \alpha_n y^{(n)} = 0 \implies \alpha_1 - \alpha_2 = 0$$

The dual (in matrix notation):

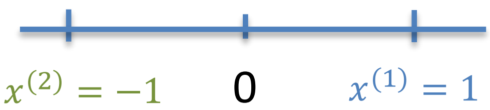
$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)} =$$

$$= (\alpha_1 .. \alpha_N) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - \frac{1}{2} (\alpha_1 .. \alpha_N) \begin{pmatrix} y^{(1)} \mathbf{x}^{(1)^\top} \mathbf{x}^{(1)} y^{(1)} & \dots & y^{(1)} \mathbf{x}^{(1)^\top} \mathbf{x}^{(N)} y^{(N)} \\ \vdots & \dots & \vdots \\ y^{(N)} \mathbf{x}^{(N)^\top} \boldsymbol{x}^{(1)} y^{(1)} & \dots & y^{(N)} \mathbf{x}^{(N)^\top} \mathbf{x}^{(N)} y^{(N)} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} =$$

$$= (\alpha_1 .. \alpha_N) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - \frac{1}{2} (\alpha_1 .. \alpha_N) \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$$

Example

- **Example**: $\{(x^{(1)} = 1, y^{(1)} = +1), (x^{(2)} = -1, y^{(2)} = -1)\}$
  We want the the maxim margin hyperplane that separate both classes
- Plot the situation



There is an alternative way to obtain the result, using only the dual and the two results of the primal

$$L_D(\alpha_1, \alpha_2) = (\alpha_1, \alpha_2)\begin{pmatrix}1\\1\end{pmatrix} - \frac{1}{2} = (\alpha_1, \alpha_2)\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}\begin{pmatrix}\alpha_1\\\alpha_2\end{pmatrix}$$

$$\frac{\partial L_D}{\partial \boldsymbol{\alpha}} = \begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}\begin{pmatrix}\alpha_1\\\alpha_2\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \implies \quad 1 = \alpha_1 + \alpha_2$$

From the primal we know:

- $\mathbf{w} = \alpha_1 + \alpha_2$
- $\alpha_1 - \alpha_2 = 0$

Then: $\alpha_1 = \alpha_2 \quad 1 = 2\alpha_1 \quad$ and $\alpha_1 = \alpha_1 = 1/2$
$w = 1$

The classifier will be $g(x) = x$ and the Margin $\frac{2}{||1||} = 2$

Using

$$\frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}$$

and for $\mathbf{A}$ symmetric matrix

$$\frac{\partial \mathbf{x}^\top \mathbf{A}\,\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}$$

Example (recap)

From the set learning points

- plot the points and write the model $g(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x} + b$
- For each point, create the constraints, formulate the primal and obtain:
  - $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} x^{(n)}$ the vector is a linear combination of only Support Vectors
  - $\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$
- Formulate the Dual
  - $L_D(\boldsymbol{\alpha}) = (\alpha_1 .. \alpha_N)\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix} - \frac{1}{2}(\alpha_1 .. \alpha_N)\begin{pmatrix} y^{(1)}\mathbf{x}^{(1)\top}\mathbf{x}^{(1)}y^{(1)} & \dots & y^{(1)}\mathbf{x}^{(1)\top}\mathbf{x}^{(N)}y^{(N)} \\ \vdots & \dots & \vdots \\ y^{(N)}\mathbf{x}^{(N)\top}\boldsymbol{x}^{(1)}y^{(1)} & \dots & y^{(N)}\mathbf{x}^{(N)\top}\mathbf{x}^{(N)}y^{(N)} \end{pmatrix}\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_N\end{pmatrix}$

Obtain: $\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix} = \begin{pmatrix} y^{(1)}\mathbf{x}^{(1)\top}\mathbf{x}^{(1)}y^{(1)} & \dots & y^{(1)}\mathbf{x}^{(1)\top}\mathbf{x}^{(N)}y^{(N)} \\ \vdots & \dots & \vdots \\ y^{(N)}\mathbf{x}^{(N)\top}\boldsymbol{x}^{(1)}y^{(1)} & \dots & y^{(N)}\mathbf{x}^{(N)\top}\mathbf{x}^{(N)}y^{(N)} \end{pmatrix}\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_N\end{pmatrix}$

Find $\alpha_1 .. \alpha_N$

The $\alpha_i \neq 0$ are the Support Vectors

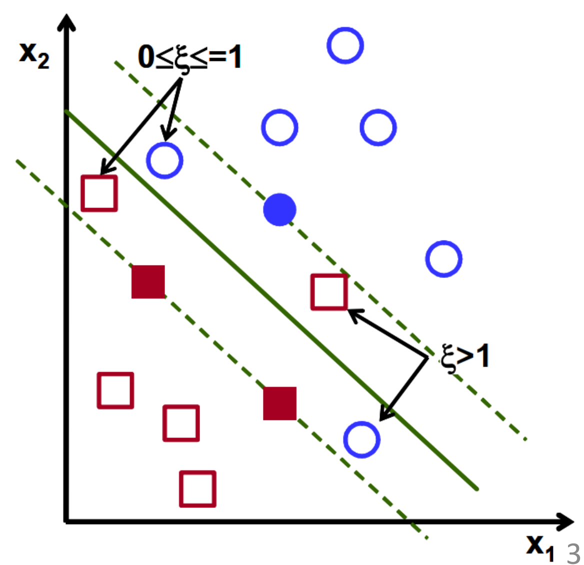
Impose that in the model the supports give $+1$ or -1, and calculate b

The non-separable case

- **So far, we focused on linearly separable problems**
  - SVMs can be modified to handle datasets that are *non*-linearly separable
- **Solution**
  - The solution for the non-separable case is to *introduce slack variables* that relax the constraints of the canonical hyperplane equation

$$y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1 - \xi_n \qquad \forall n = 1, \dots, N$$

- For $0 \leq \xi_n \leq 1$, the data points fall on the **right side** of the hyperplane, but within the region of maximum margin
- For $\xi_n > 1$, the data points fall on the **wrong side** of the hyperplane



The non-separable case

- We minimize the following objective

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^2 + C\sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) \geq 1 - \xi_n, \\ & \xi_n \geq 0, \quad \forall n = 1, \dots, N \end{aligned}$$

Interpretation of C

- Represents a trade-off between misclassification and capacity
- **Large** C favors solutions with **few classification errors**
- **Small** C favors **low-complexity** solutions
- C can be viewed as a ***regularization*** *parameter*
- Typically determined through ***cross-validation***

The non-separable case

Solution

- We can derive the *dual problem* as

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)^\top} \mathbf{x}^{(n)}$$

subject to

$$0 \leq \alpha_n \leq C$$
$$\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$$

- **Remarks**

  - Neither the slack variables nor associated Lagrange multipliers appear in the formulation

  - The problem is the same as the linearly separable case, with the difference in the constraints $0 \leq \alpha_n$ that become $0 \leq \alpha_n \leq C$

The non-separable case

Solution

- We can derive the *dual problem* as

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \mathbf{x}^{(m)\top} \mathbf{x}^{(n)}$$

subject to

$$0 \le \alpha_n \le C$$

$$\sum_{n=1}^{N} \alpha_n y^{(n)} = 0$$

- **Remarks**
  - The optimum solution for the weights remains the same

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} \mathbf{x}^{(n)}$$

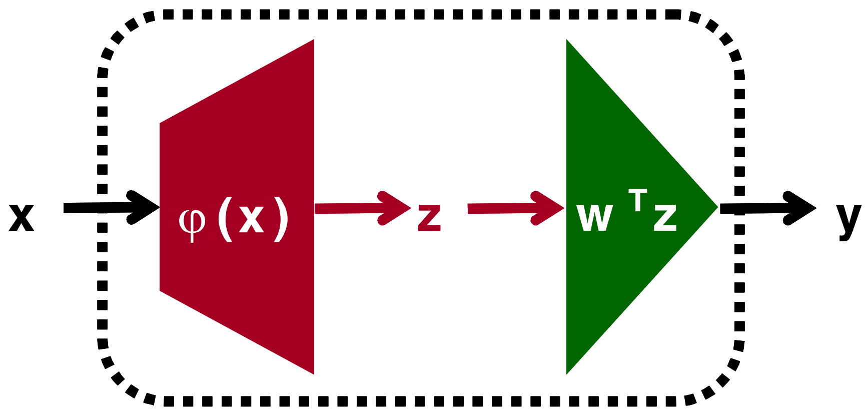
  - The bias can be found with a training point for which $0 < \alpha_n < C$ $(\xi_n = 0)$

$$\alpha_n\left[y^{(n)}\left(\mathbf{w}^\top \mathbf{x}^{(n)} + b\right) - 1 + \xi_n\right] = 0$$

Non-linear SVMs

Cover's theorem on the separability of patterns

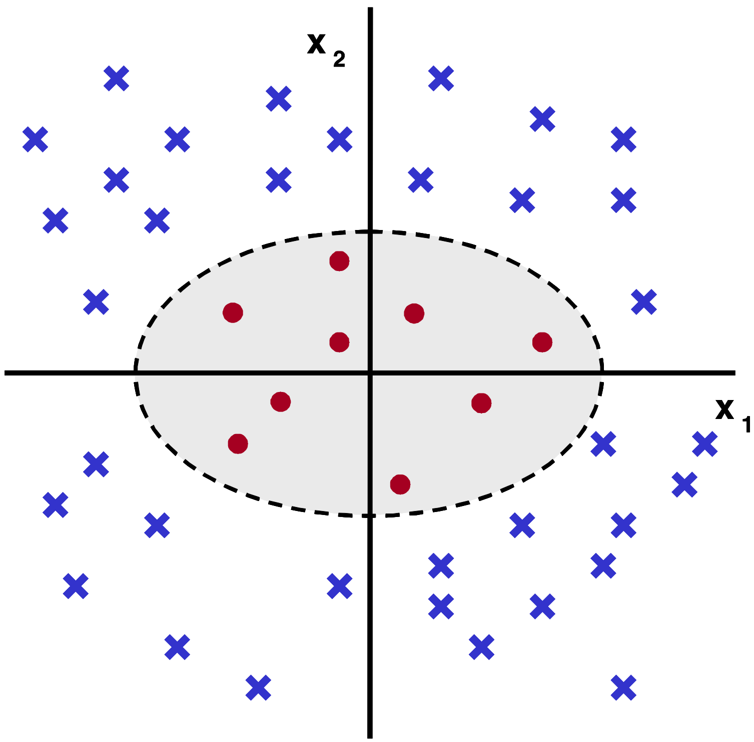
- *"A complex pattern-classification problem cast in a high-dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space"*
- The power of SVMs resides in the fact that they represent a robust and efficient implementation of Cover's theorem
- SVMs operate in two stages
  - Perform a non-linear mapping of the feature vector x onto a high-dimensional space that is hidden from the inputs or the outputs
  - Construct an optimal separating hyperplane in the high-dim space

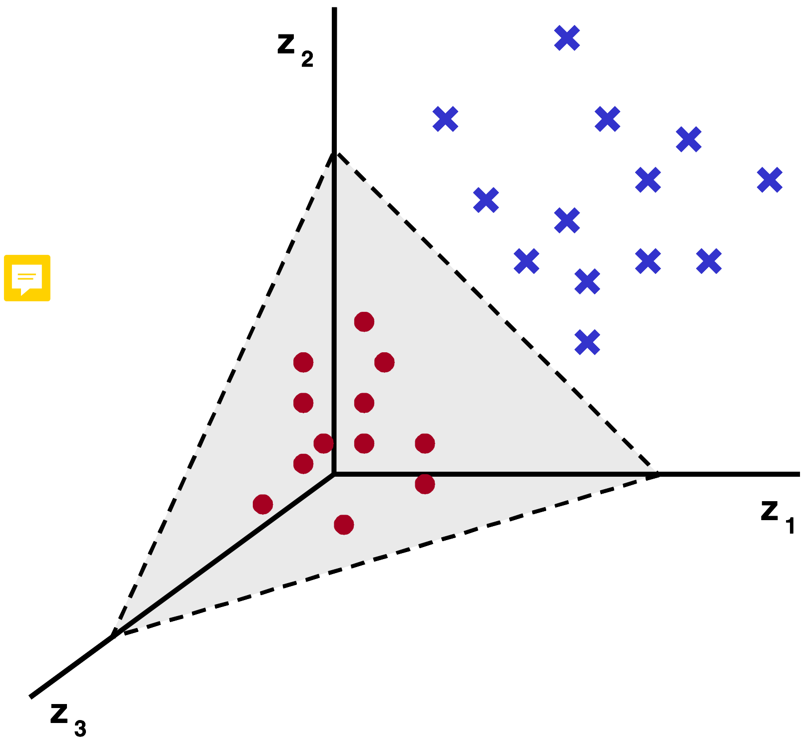


Nonlinear SVMs

$$\varphi : R^2 \rightarrow R^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$





[Schölkopf, 2002 @; http://kernel-machines.org/]

Nonlinear SVMs

- Naïve application of this concept by simply projecting to a high-dimensional non-linear manifold has two major problems:

1. **Statistical**: operation on high-dimensional spaces is ill-conditioned due to the *curse of dimensionality* and the subsequent risk of overfitting

2. **Computational**: working in high-dimensions requires higher computational power, which poses limits on the size of the problems that can be tackled

Nonlinear SVMs

- SVMs bypass these two problems in a robust and efficient way

1. Generalization capabilities in the high-dimensional manifold are ensured by enforcing a **largest margin classifier**

– SVMs optimize the **the margin** (dual is independent of D)

2. High-dimensional projection is **implicit**

– The SVM solution depends only on the dot product $\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}$ between training examples

– Operations in high-dimensional space $\boldsymbol{\phi}(\mathbf{x})$ do not have to be performed ***explicitly*** if we find a function

$$K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^{\top} \boldsymbol{\phi}\left(\mathbf{x}^{(m)}\right)$$

– $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right)$ is called a **kernel function**

Implicit mappings: an example

- Consider a problem in **two dimensions**
- Assume we choose kernel function $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2$
- Our goal is to find a non-linear projection $\phi(\cdot)$ such that

$$\left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2 = \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^\top \boldsymbol{\phi}\left(\mathbf{x}^{(m)}\right)$$

Implicit mappings: an example

- Consider a problem in **two dimensions**
- Assume we choose kernel function $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2$
- Our goal is to find a non-linear projection $\phi(\cdot)$ such that

$$\left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2 = \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^\top \boldsymbol{\phi}\left(\mathbf{x}^{(m)}\right)$$

- Performing the expansion of $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right)$

$$K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2 = \left(\left(x_1^{(n)}, x_2^{(n)}\right)^\top \left(x_1^{(m)}, x_2^{(m)}\right)\right)^2 = \left(x_1^{(n)}x_1^{(m)} + x_2^{(n)}x_2^{(m)}\right)^2$$

$$= \left({x_1^{(n)}}^2, \sqrt{2}x_1^{(n)}x_2^{(n)}, {x_2^{(n)}}^2\right)^\top \left({x_1^{(m)}}^2, \sqrt{2}x_1^{(m)}x_2^{(m)}, {x_2^{(m)}}^2\right)$$

3 dimensions!

Implicit mappings: an example

- Consider a problem in **two dimensions**
- Assume we choose kernel function $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2$
- Our goal is to find a non-linear projection $\phi(\cdot)$ such that

$$\left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2 = \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^\top \boldsymbol{\phi}\left(\mathbf{x}^{(m)}\right)$$

- Performing the expansion of $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right)$

$$K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2 = \left(\left(x_1^{(n)}, x_2^{(n)}\right)^\top \left(x_1^{(m)}, x_2^{(m)}\right)\right)^2 = \left(x_1^{(n)}x_1^{(m)} + x_2^{(n)}x_2^{(m)}\right)^2$$

$$= \left(x_1^{(n)2}, \sqrt{2}x_1^{(n)}x_2^{(n)}, x_2^{(n)2}\right)^\top \left(x_1^{(m)2}, \sqrt{2}x_1^{(m)}x_2^{(m)}, x_2^{(m)2}\right)$$

3 dimensions!

So in using the kernel $K\left(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}\right) = \left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2$ we are implicitly operating on a higher-dimensional non-linear manifold defined by

$$\boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right) = \left(x_1^{(n)2}, \sqrt{2}x_1^{(n)}x_2^{(n)}, x_2^{(n)2}\right)^\top$$

The inner product can be computed in 2 dimensions by means of the kernel $\left(\mathbf{x}^{(n)\top}\mathbf{x}^{(m)}\right)^2$ without ever having to project onto 3 dimensions!

Kernel Methods

Let's now see how to put together all these concepts:

- Assume that original feature vector lives in a space $\mathbb{R}^D$
- Our interest is projecting onto a higher dimensional space $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^M (M > D)$, where classes have a better chance of being linearly separable
- The separating hyperplane in M will be defined by

$$\sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) + b = 0$$

- To eliminate the bias term b, as always, we consider a constant feature $\phi_0(\mathbf{x}) = 1$
- The resulting hyperplane becomes $\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) = 0$
- From our previous results, the optimal (maximum margin) hyperplane in the implicit space is given by $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} \boldsymbol{\phi}\left(\mathbf{x}^{(\boldsymbol{n})}\right)$

Kernel Methods

Merging this optimal weight vector with the hyperplane equation:

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) = 0$$

$$\left(\sum_{n=1}^{N} \alpha_n y^{(n)} \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right) \right)^\top \boldsymbol{\phi}(\mathbf{x}) = 0$$

$$\sum_{n=1}^{N} \alpha_n y^{(n)} \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^\top \boldsymbol{\phi}(\mathbf{x}) = 0$$

And, since $\boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right)^\top \boldsymbol{\phi}\left(\mathbf{x}^{(m)}\right) = K(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$ the optimal hyperplane becomes

$$\sum_{n=1}^{N} \alpha_n y^{(n)} K(\mathbf{x}^{(n)}, \mathbf{x}) = 0$$

Therefore, classification of an unknown example $\mathbf{x}$ is performed by computing the weighted sum of the kernel with respect to the support vectors $\mathbf{x}^{(n)}$(remember that only the support vectors have non-zero dual variables α_n)

Kernel Methods

How do we compute dual variables α_n in the implicit space?

- Very simple: we use the same optimization problem as before and replace the dot product $\boldsymbol{\phi}^\top(\boldsymbol{x}^{(n)})\boldsymbol{\phi}(\boldsymbol{x}^{(m)})$ with the kernel $K(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$

- The Lagrangian dual problem for the non-linear SVM is simply

$$L_D(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} K\left(\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(m)}\right)$$

Subject to the constraints

$$\begin{cases} \displaystyle\sum_{n=1}^{N} \alpha_n y^{(n)} = 0 \\ 0 \le \alpha_n \le C \qquad n = 1, \dots, N \end{cases}$$

Kernel Methods

Illustration : the XOR problem

- **Dataset** :

  Class +1 $\mathbf{x}^{(1)} = (+1,+1), \mathbf{x}^{(4)} = (-1,-1)$

  Class -1 $\mathbf{x}^{(2)} = (-1,+1), \mathbf{x}^{(3)} = (+1,-1)$
- Kernel function

  Polynomial of 2nd order: $K(\mathbf{x},\mathbf{x}') = (\mathbf{x}^{\top}\mathbf{x}' + 1)^2$
- **Solution**
  - The implicit mapping can be shown to be five dimensional

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix}1 & \sqrt{2}x_1 & \sqrt{2}x_2 & \sqrt{2}x_1x_2 & x_1^2 & x_2^2\end{bmatrix}^{\top}$$

  - To achieve linear separability, we use $C = \infty$
  - The objective function for the dual problem becomes

$$L_D(\boldsymbol{\alpha}) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\sum_{n=1}^{4}\sum_{m=1}^{4}\alpha_n\alpha_m y^{(n)}y^{(m)}K_{n,m}$$

Subject to the constraints $\sum_{n=1}^{N}\alpha_n y^{(n)} = 0$ and $0 \le \alpha_n \le C, n = 1, \dots, N$

– where the inner product is represented as a 4×4 K matrix

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

– Optimizing with respect to the Lagrange multipliers leads to the following system of equations

$$\begin{gathered} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1 \end{gathered}$$

– whose solution is $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.125$

– Thus, all data points are support vectors in this case

- For this simple problem, it is worthwhile to write the decision surface in terms of the polynomial expansion

$$\mathbf{w} = \sum_{n=1}^{4} \alpha_n y^{(n)} \boldsymbol{\phi}\left(\mathbf{x}^{(n)}\right) = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}^{\mathrm{T}}$$

- Resulting in the intuitive non-linear discriminant function

$$g(\mathbf{x}) = \sum_{i=1}^{6} w_i \phi_i(\mathbf{x}) = x_1 x_2$$

- Which has zero empirical error in the XOR dataset

Decision function defined by the SVM

– Notice that the decision boundaries are non-linear in the original space R^2, but linear in the implicit space R^6