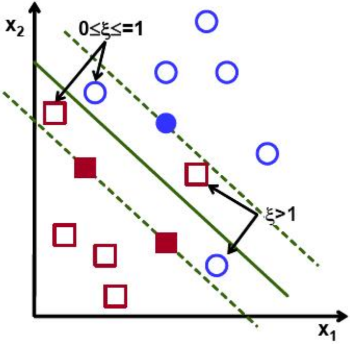
Support Vector Machines

Non linearly separable problems (I)



- We introduce the slack variables $\xi_i \geq 0$
- Now the constraints are

$$t_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i$$

- $\xi_i = 0$ for points out of the margin that are correctly classified:

$$t_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$$

- $0 \leq \xi_i \leq 1$ for points inside the margin that are correctly classified
- $\xi_i > 1$ for points that are not correctly classified

- **New goal:** to maximize the margin while penalizing wrongly classified patterns

Non linearly separable problems (II)

Optimization problem

- Minimize with respect to $\mathbf{w}$, b and ξ:

$$J(\mathbf{w}, \xi) = C \sum_{i=1}^{n} \xi_i + \frac{1}{2}||\mathbf{w}||^2$$

- Subject to the constraints:

$$t_i(\mathbf{w}^t \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- $\sum_{i=1}^{n} \xi_i$ is an upper bound to the total number of errors
- The C parameter controls the relative weight given to the training classification error and to the complexity (margin)
  - Higher C favours models with smaller error
  - Lower C favours simpler models

Non linearly separable problems (III)

- As before, we introduce Lagrange multipliers α_i and μ_i

$$L(\mathbf{w}, b, \alpha, \mu) =$$
$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[t_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\mu_i\xi_i$$

- The KKT conditions are now:

$$\alpha_i \geq 0$$
$$t_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i \geq 0$$
$$\alpha_i[t_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i] = 0$$
$$\mu_i \geq 0$$
$$\xi_i \geq 0$$
$$\mu_i\xi_i = 0$$

Non linearly separable problems (IV)

- Setting the gradient of L wrt $\mathbf{w}$ equal to 0 we get:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i t_i \mathbf{x}_i$$

- Setting the derivative of L wrt b equal to 0 we get:

$$0 = \sum_{i=1}^{n} \alpha_i t_i$$

- Setting the derivative of L wrt ξ_i equal to 0 we get:

$$\alpha_i = C - \mu_i$$

- Substituting this expressions in L we get the **dual problem**

The dual problem

Dual problem

- Maximize with respect to α_i:

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \mathbf{x}_j$$

- Subject to the constraints:

$$0 \leq \alpha_i \leq C$$

$$\sum_{i=1}^{n} \alpha_i t_i = 0$$

- The problem is esentially the same as in the linearly separable case, but with different constraints

Support vectors (I)

As before, we have:

- $\alpha_i = 0$ for points out of the margin that are correctly classified
  - These points do not contribute to the definition of the separating hyperplane
- The rest of the points are **support vectors**
  - They satisfy:

$$t_i(\mathbf{w}^t \mathbf{x}_i + b) = 1 - \xi_i$$

$$\alpha_i > 0$$

Support vectors (II)

- Support vectors satisfy $t_i(\mathbf{w}^t\mathbf{x}_i + b) = 1 - \xi_i$, with $\alpha_i > 0$
- Two possibilities:
  - $\alpha_i < C$, $\mu_i > 0$ and $\xi_i = 0$; these points are **on** the margin
  - $\alpha_i = C$, $\mu_i = 0$ and $\xi_i > 0$; these points are **inside** the margin (correctly classified if $\xi_i \leq 1$, wrongly classified if $\xi_i > 1$)
- The separating hyperplane is given by:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i t_i \mathbf{x}_i$$

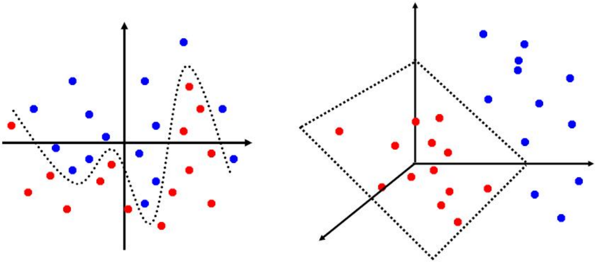
- With b obtained from any support vector with $\alpha_i < C$

$$t_i(\mathbf{w}^t\mathbf{x}_i + b) = 1$$

- We only need the support vectors to perform classification

Non-linear problems (I)

- **Cover's theorem:** A classification problem which is projected onto a high dimensional space is more likely to be linearly separable
- Using this idea, the SVMs perform two steps:
  1. They make a non linear projection of the data onto a high dimensional space
  2. They find the best separating hyperplane in that space



Non-linear problems (II)

Projecting onto a high dimensional space presents two main problems:

1. "Curse of dimensionality"
   - Much more patterns are needed to train the models
   - The models are more prone to overfitting
   - SVMs overcome this problem by maximizing the margin; note that the model complexity depends only on the margin, not on the dimension
2. Much higher computational cost
   - SVMs overcome this problem by making the projection only implicitly (thanks to the **kernel** trick)

Kernel methods (I)

- **Kernel:** function $k(\mathbf{x}_i, \mathbf{x}_j)$ that can be expressed as the dot product
$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i)^t \mathbf{\Phi}(\mathbf{x}_j)$$
for some transformation $\mathbf{\Phi}(\mathbf{x})$
- Example: the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t \mathbf{x}_j)^2$, with $\mathbf{x}_i \in \mathbb{R}^2$, can be expressed as
$$k(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2)^t$$
  - The associated transformation is
$$\mathbf{\Phi}(\mathbf{x}) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^t$$

Kernel methods (II)

The SVM general strategy:

1. $\mathbf{x}_i \in \mathbb{R}^d$, con $i = 1, 2, ..., n$
2. Find a non linear transformation $\mathbf{z} = \mathbf{\Phi}(\mathbf{x})$, with $\mathbf{z} \in \mathbb{R}^T$ and $T > d$, such that $\mathbf{\Phi}(\mathbf{x})^t \mathbf{\Phi}(\mathbf{y}) = k(\mathbf{x}, \mathbf{y})$ for a given kernel k
3. In this T-dimensional space the two classes are more likely to be linearly separated
4. Find the optimal separating hyperplane in this transformed space
$$\mathbf{w}^t \mathbf{\Phi}(\mathbf{x}) + b = 0$$

Kernel methods (III)

- As before, $\mathbf{w}$ is given by the support vectors ($\alpha_i \neq 0$)

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i t_i \mathbf{\Phi}(\mathbf{x}_i)$$

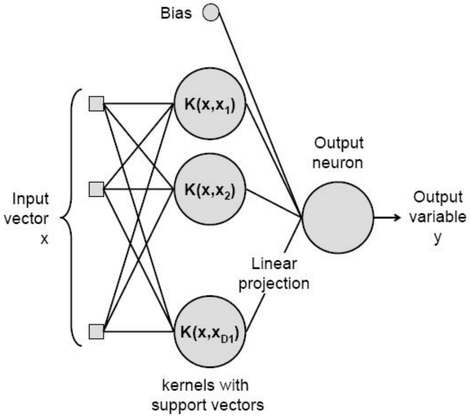
- The b coefficient is obtained from a support vector with $\alpha_i < C$

$$t_i(\mathbf{w}^t \mathbf{\Phi}(\mathbf{x}_i) + b) = 1$$

- Finally, to classify a new pattern $\mathbf{x}$ we must evaluate

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i t_i \mathbf{\Phi}(\mathbf{x}_i) \mathbf{\Phi}(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i t_i k(\mathbf{x}_i, \mathbf{x}) + b$$

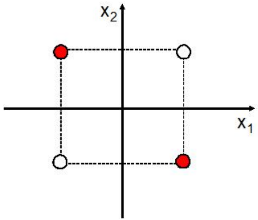
General structure of a SVM



$$y = \sum_{i=1}^{n} \alpha_i t_i k(\mathbf{x}_i, \mathbf{x}) + b$$

A simple example (I)

- XOR in 2D
  - Class 1: $\mathbf{x}_1 = (-1, -1)$, $\mathbf{x}_2 = (1, 1)$, $t = 1$
  - Class 2: $\mathbf{x}_3 = (1, -1)$, $\mathbf{x}_4 = (-1, 1)$, $t = -1$



- We use the kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^t\mathbf{y} + 1)^2$
  - The associated transformation is

$$\mathbf{\Phi}(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2)^t$$

- We take $C = \infty$ to favour small error models

A simple example (II)

- The dual problem is

$$\tilde{L}(\alpha) = \sum_{i=1}^{4} \alpha_i + -\frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j t_i t_j k(\mathbf{x}_i, \mathbf{x}_j)$$

- With the constraints

$$\alpha_i \geq 0$$

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

- The kernel can be expressed as $k(\mathbf{x}_i, \mathbf{x}_j) = K_{ij}$, with

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

- Then

$$\tilde{L}(\alpha) = \sum_{i=1}^{4} \alpha_i - \frac{9}{2} \sum_{i=1}^{4} \alpha_i^2 - \alpha_1\alpha_2 + \alpha_1\alpha_3 + \alpha_1\alpha_4 + \alpha_2\alpha_3 + \alpha_2\alpha_4 - \alpha_3\alpha_4$$

A simple example (III)

- We optimize with respect to the multipliers α_i

$$\frac{\partial \tilde{L}(\alpha)}{\partial \alpha_1} = 0 \Longrightarrow 9\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 1$$

$$\frac{\partial \tilde{L}(\alpha)}{\partial \alpha_2} = 0 \Longrightarrow \alpha_1 + 9\alpha_2 - \alpha_3 - \alpha_4 = 1$$

$$\frac{\partial \tilde{L}(\alpha)}{\partial \alpha_3} = 0 \Longrightarrow -\alpha_1 - \alpha_2 + 9\alpha_3 + \alpha_4 = 1$$

$$\frac{\partial \tilde{L}(\alpha)}{\partial \alpha_4} = 0 \Longrightarrow -\alpha_1 - \alpha_2 + \alpha_3 + 9\alpha_4 = 1$$

- To obtain the solution

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

- Note that all the points are support vectores and they are on the margin

A simple example (IV)

- The classification function is given by

$$f(\mathbf{x}) = \frac{1}{8}\sum_{i=1}^{4} t_i k(\mathbf{x}_i, \mathbf{x}) + b$$

- We obtain b from

$$\mathbf{w} = \frac{1}{8}(\mathbf{\Phi}(\mathbf{x}_1) + \mathbf{\Phi}(\mathbf{x}_2) - \mathbf{\Phi}(\mathbf{x}_3) - \mathbf{\Phi}(\mathbf{x}_4))$$

$$t_i(\mathbf{w}^t \mathbf{\Phi}(\mathbf{x}_i) + b) = 1$$

- Which leads to $b = 0$
- Operating, we finally obtain

$$f(\mathbf{x}) = x_1 x_2$$

which, as we already know, solves the XOR problem

Summary: Advantages of SVMs

- No local minima (quadratic problem)
- The optimal solution can be found in polynomial time
- Small number of free parameters: C, kernel type and kernel parameters. They can be automatically adjusted using **cross-validation**
- Stable result (it does not depend on initial random values)
- Sparse solution (it only takes into account the support vectors)
- Maximizing the margin allows to control the complexity independently on the number of dimensions
- Good generalization capability