Support Vector Machines

Introduction (I)

- Consider the classification problem:
  $\{(\mathbf{x_1}, t_1), (\mathbf{x_2}, t_2), ..., (\mathbf{x_n}, t_n)\}$
  - n is the number of training patterns
  - $\mathbf{x_i}$ is the attribute vector for pattern i
  - t_i is the class label for pattern i, $t_i \in \{-1, 1\}$
- A classifier is a function $f(\mathbf{x}, \Theta)$ that assigns each $\mathbf{x_i}$ an estimation of its class $y_i = f(\mathbf{x_i}, \Theta)$
- We usually train the classifier parameters Θ in order to minimize a risk function defined over the training data:

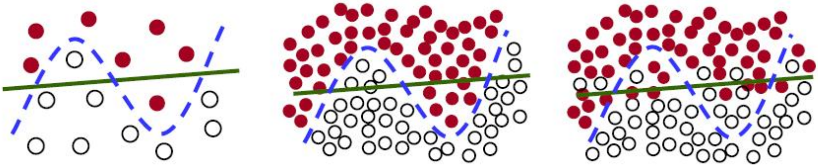
$$R_{train}[f] = \frac{1}{n} \sum_{i=1}^{n} C(y_i, t_i)$$

- Where $C(y, t)$ is a cost function, usually the mean squared error:

$$C(y, t) = (y - t)^2$$

Introduction (II)

- When the number of training patterns is small, we may obtain a classifier that **overfits** the training data and has a poor generalization capability
- How can we prevent overfitting?
  - A common approach involves controlling the **model complexity**: a simpler model is preferred over a more complex one as far as they both provide a similar classification accuracy



The VC dimension (I)

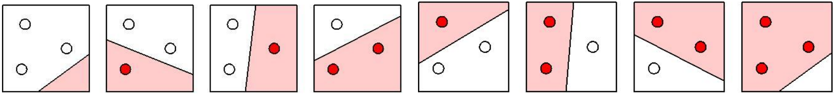
- The Vapnik-Chervonenkis (VC) dimension measures the complexity of a given family of functions $f(\mathbf{x}; \boldsymbol{\Theta})$
  - f represents the family
  - $\boldsymbol{\Theta}$ is the set of parameters
- The VC dimension of a family $f(\mathbf{x}; \boldsymbol{\Theta})$ is defined as the maximum number of patterns that can be explained by this family
- More complex families are able to fit more complex data sets, but they present a lower generalization capability

The VC dimension (II)

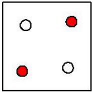
- **Shattering**:
  - Consider a dataset with n patterns $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ belonging to 2 different classes
  - There exist 2^n different ways to assign the class labels
  - For example, if $n = 3$ there are 8 different such class assignments: $\{(-1,-1,-1), (-1,-1,1), ..., (1,1,1)\}$
  - The family of functions $f(\mathbf{x}; \mathbf{\Theta})$ **shatters** the dataset if for any possible class assignment α there exists a set of parameters $\mathbf{\Theta}_\alpha$ such that $f(\mathbf{x}; \mathbf{\Theta}_\alpha)$ solves it
- The **VC dimension** of the family $f(\mathbf{x}; \mathbf{\Theta})$ is defined as the size of the larger set which can be shattered by $f(\mathbf{x}; \mathbf{\Theta})$
  - If the VC dimension of $f(\mathbf{x}; \mathbf{\Theta})$ is h, then there exists at least one set with h points which can be shattered by $f(\mathbf{x}; \mathbf{\Theta})$

The VC dimension (III)

- Example
  - Consider the family $f(\mathbf{x}; \boldsymbol{\Theta})$ of hyperplanes in $\mathbb{R}^2$
  - $f(\mathbf{x}; \boldsymbol{\Theta}) = w_0 + w_1 x_1 + w_2 x_2$
  - $\boldsymbol{\Theta} = (w_0, w_1, w_2)$
- It is possible to find a set of $n = 3$ points that is shattered using hyperplanes (all different class assignments are solved)



- But this is not possible for $n = 4$

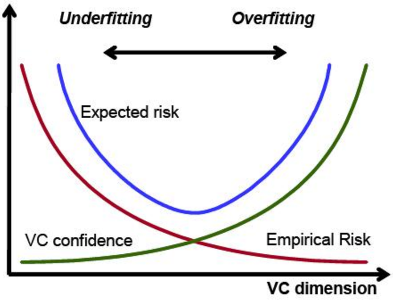


- So the VC dimension of the family of hyperplanes in $\mathbb{R}^2$ is 3

Structural Risk Minimization (I)

- Vapnik & Chervonenkis
- To obtain an optimal classifier we should balance the empirical risk measured on the training data and the VC dimension of the model
- With probability $1 - \eta$, the expected risk is upper bounded by:
$$E[R[f]] \leq R_{train}[f] + \sqrt{\frac{h(\log \frac{2n}{h} + 1) - \log \frac{\eta}{4}}{n}}$$
where
  - h is the VC dimension of f
  - n is the number of training patterns
  - $n > h$
- The second term is called **VC confidence**
- When n/h increases, VC decreases and the empirical risk becomes a better approximation of the expected risk

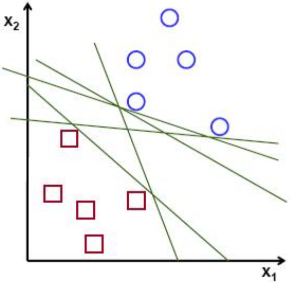
Structural Risk Minimization (II)

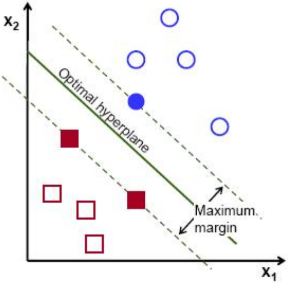


- We should select the model with the lowest upper bound to the expected risk
- In practical terms, computing the VC dimension is not feasible in most situations
- Linear models are an exception

Optimal separating hyperplane (I)

- Consider the problem $\{(\mathbf{x_1}, t_1), (\mathbf{x_2}, t_2), ..., (\mathbf{x_n}, t_n)\}$
  - n patterns, 2 classes, $t_i \in \{-1, 1\}$, linearly separable
- Which is the **optimal separating hyperplane**?
- It seems reasonable to maximize the **margin** (minimum distance from any point to the decision boundary)
  - The higher the margin is, the more tolerant our model is to statistical fluctuations (higher generalization capability)





Optimal separating hyperplane (II)

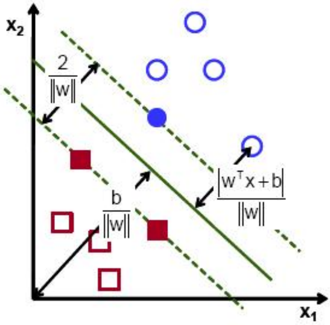
- This intuition is supported by the results of SRM
- The VC dimension of a separating hyperplane with margin m is bounded by the following upper bound:

$$h \leq \text{mín}(\lceil \frac{R^2}{m^2} \rceil, d) + 1$$

  - d is the dimension
  - R is the radius of the smallest hypersphere that contains all data points
- When we maximize the margin we are minimizing the VC dimension, and so increasing the generalization capability of the model
- If the margin is large enough the VC dimension, and so the model complexity, can be small even when the dimension d is very large

Optimal separating hyperplane (III)

- We want to find the separating hyperplane $\mathbf{w}^t\mathbf{x} + b = 0$ that maximizes the margin



- The distance from point $\mathbf{x}_i$ to the hyperplane is given by:

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

- **Canonical hyperplane**: $|\mathbf{w}^t\mathbf{x} + b| = 1$ for the closest points
- Using this canonical representation, the margin is

$$m = \frac{1}{||\mathbf{w}||}$$

Optimal separating hyperplane (IV)

The problem of maximizing the margin is equivalent to the following

Optimization problem

- Minimize (with respect to $\mathbf{w}$ and b):

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

- Subject to the constraints $t_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 \;\; \forall i$

- To solve this problem we introduce a Lagrange multiplier $\alpha_i \geq 0$ for each of the constraints and obtain the Lagrangian function:

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

Optimal separating hyperplane (V)

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

The solution to the original optimization problem can be obtained by optimizing the Lagrangian function $L(\mathbf{w}, b, \alpha)$ with respect to $\mathbf{w}$, b and α_i subject to the

Karush-Kuhn-Tucker (KKT) conditions

$$\alpha_i \geq 0$$

$$t_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 \geq 0$$

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

- $\alpha_i = 0$ implies $t_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 > 0$ (**inactive** constraint)
- $\alpha_i > 0$ implies $t_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 = 0$ (**active** constraint)

The dual problem (I)

- Setting the gradient of $L(\mathbf{w}, b, \alpha)$ with respect to $\mathbf{w}$ and b equal to 0 we get

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

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

- And substituting these expressions back into $L(\mathbf{w}, b, \alpha)$ we obtain the **dual problem**

The dual problem (II)

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:

$$\alpha_i \geq 0$$

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

Support vectors (I)

- Recall the KKT conditions:

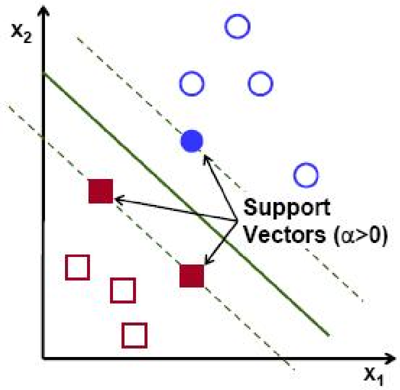
$$\alpha_i \geq 0$$

$$t_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 \geq 0$$

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

- For any $\mathbf{x}_i$, one and only one of the following two conditions holds:
  - $\alpha_i = 0$; these points do not contribute to the definition of the separating hyperplane
  - $t_i(\mathbf{w}^t \mathbf{x}_i + b) = 1$; these points define the separating hyperplane, they are called **support vectors**

Support vectors (II)



Support vectors (III)

- Only support vectors are needed to define the optimal separating hyperplane
- The vector $\mathbf{w}$ is obtained as

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

- The parameter b can then be obtained from any support vector using

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

- Note that only support vectors are necessary to perform classification

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