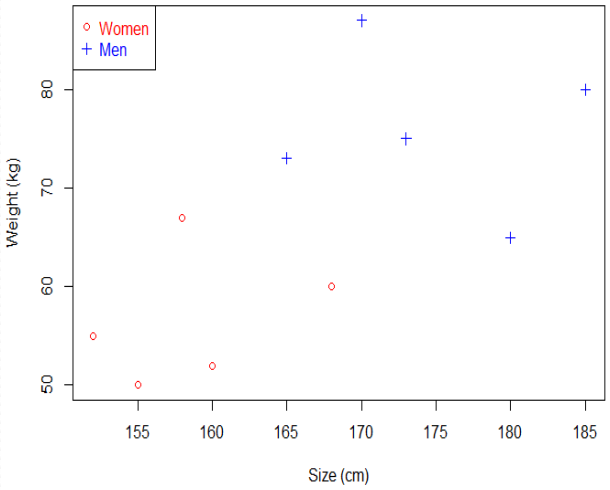
Master-MIRI
Topics on Optimization and Machine Learning (TOML)

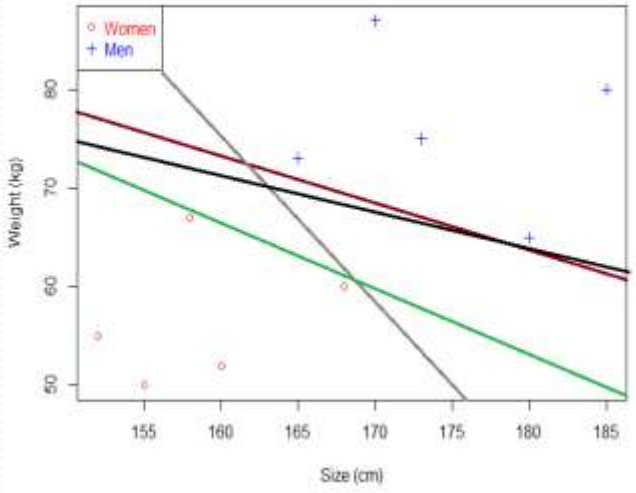
José M. Barceló Ordinas
Departament d'Arquitectura de Computadors (UPC)

Classification: Support-Vector Machines

- **Problem:** we know a dataset (weight, size, gender). We receive a new vector (weight, size), and we want to predict if the person is a man or a woman https://www.svm-tutorial.com ?
- **Solution:** classify the data, e.g. draw an hyperplane that separates the data.
- **However,** there are many hyperplanes that separate the data

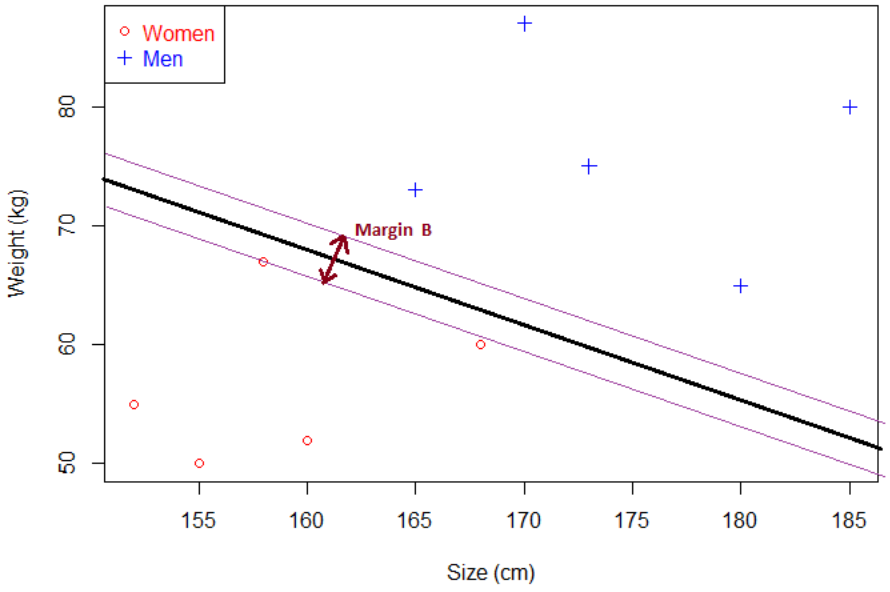
Size (cm)	Weight (Kg)	Women/Men
160	53	W
165	72	M
185	78	M
167	60	W
…	…	…

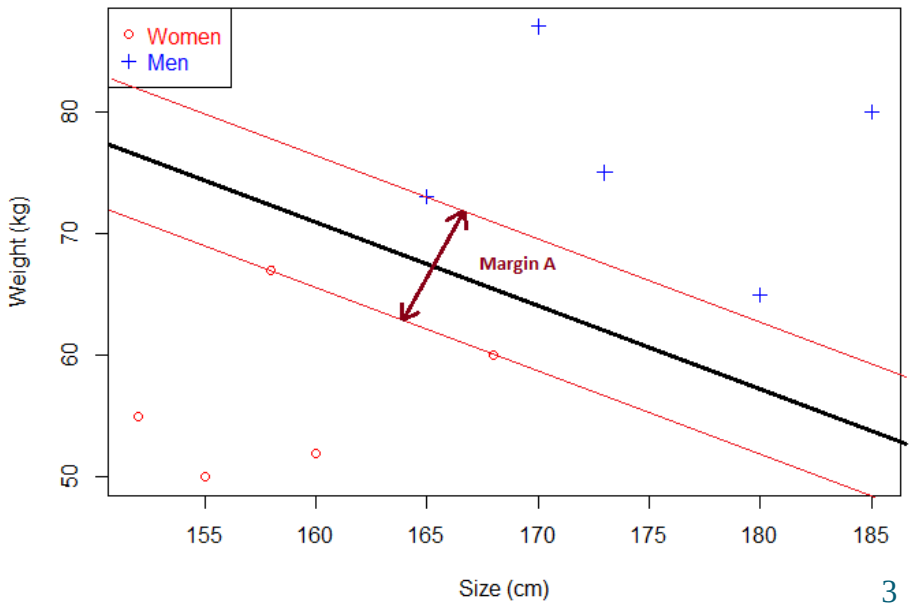




- **Classification: Support-Vector Machines. What it does …**
  - **Then,** select an hyperplane as far as possible from data points from each category.
  - **Again,** there are many hyperplanes.
  - **Margin:** no man's land (there are no data inside the margin)
  - **Solution:** select the hyperplane with biggest margin between data

the objective of the SVM is to find the optimal separating hyperplane which maximizes the margin of the training data → uses an optimization problem (KKT conditions)





SVM – non-overlapping classes. Separating hyperplanes …

- Remember that each data has M features, $\mathbf{x}_i=(x_{i1},\ldots,x_{iM})$, with i=1,…, N data points, and true values $t_i=\{-1,1\}$
- An hyperplane H passing through point $\mathbf{x}_i$ satisfies the following equation:

$H=\{x|\ w^Tx=b\} = \{x|\ w^T(x-x_0)=0\} = \{x|\ w^Tx+w_0=0\} \rightarrow w_0+ w_1x_{i1} + \ldots + w_Mx_{iM} =0$

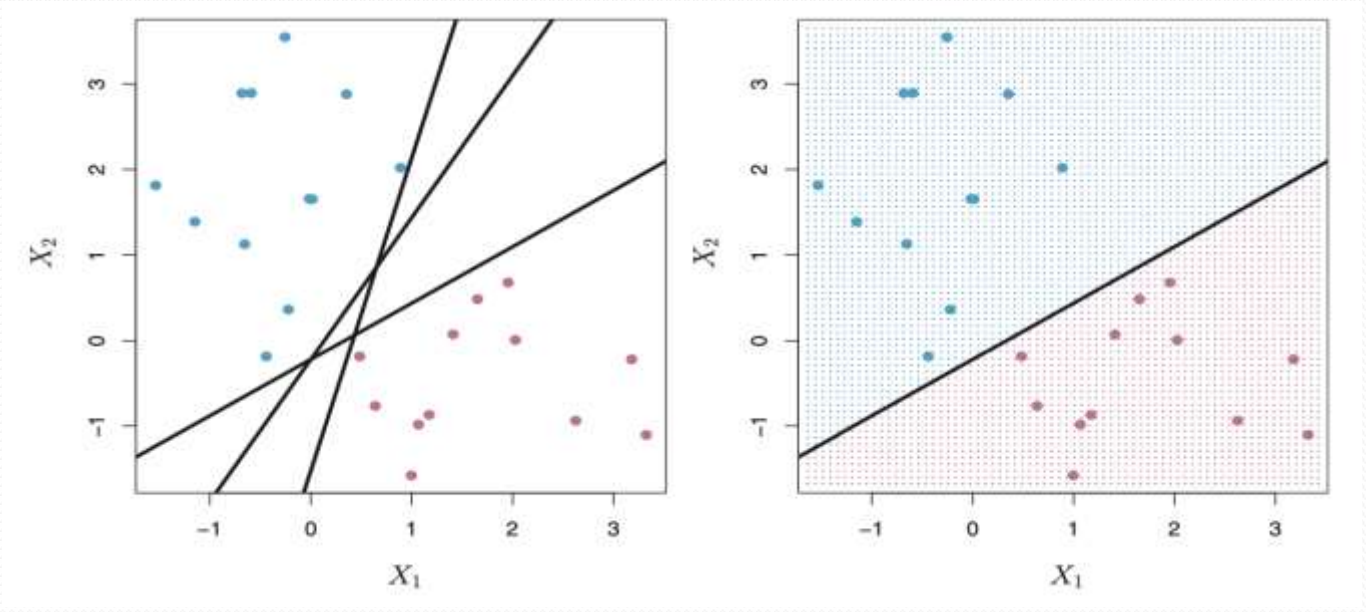
- A **separating hyperplane** has the following property:

$w_0+ w_1x_{i1} + \ldots + w_Mx_{iM} < 0$ if $t_i= -1$ (belongs to class C_1)

$w_0+ w_1x_{i1} + \ldots + w_Mx_{iM} > 0$ if $t_i= 1$ (belongs to class C_2)

And, a separating hyperplane has the following property:

$t_i\ (w_0+ w_1x_{i1} + \ldots + w_Mx_{iM}) > 0$



SVM – non-overlapping classes. Summarizing the goal …

- We take N samples, so there are N target values: $\mathbf{t}^T = (t_1, t_2, \ldots, t_N)$ for each $\mathbf{x_n} = (x_{n1}, \ldots, x_{nM})$, with M features. Each target value $t_n = \{-1, 1\}$. Thus, we classify each $y(\mathbf{x_n}, \mathbf{w})$ as being -1 or 1,
- Let us assume a mapping $\phi(\mathbf{x})$ for the type of curve that separates the target classes, e.g. linear, polynomial, etc. For simplicity, we first assume that the mapping is linear: $\phi(\mathbf{x}) = \mathbf{x}$ (we will later extend to other mappings, including non-linear). Then:

$$y(\mathbf{x},\mathbf{w}) = w_0 + \mathbf{w}^T \phi(\mathbf{x}) = w_0 + \mathbf{w}^T \mathbf{x} = w_0 + \sum_{j=1\ldots M} w_j x_j = w_0 + w_1 x_1 + \ldots + w_M x_M,$$

- We are interested in finding $(w_0, \mathbf{w})$, such that satisfies that $y(\mathbf{x_n}, \mathbf{w}) > 0$ for points belonging to $t_n = 1$ and satisfies that $y(\mathbf{x_n}, \mathbf{w}) < 0$ for points belonging to $t_n = -1$,
- In other words: $y(\mathbf{x_n}, \mathbf{w})\, t_n > 0$ <u>for all points n=1,…,N</u>.

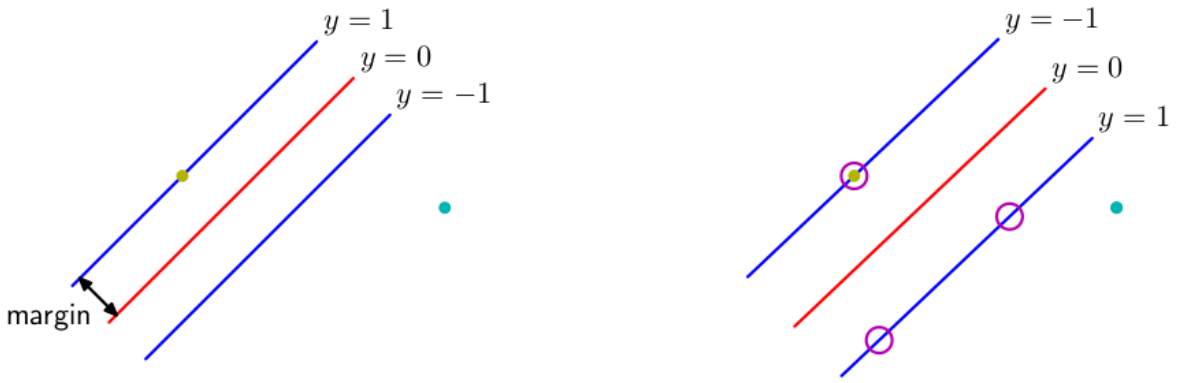
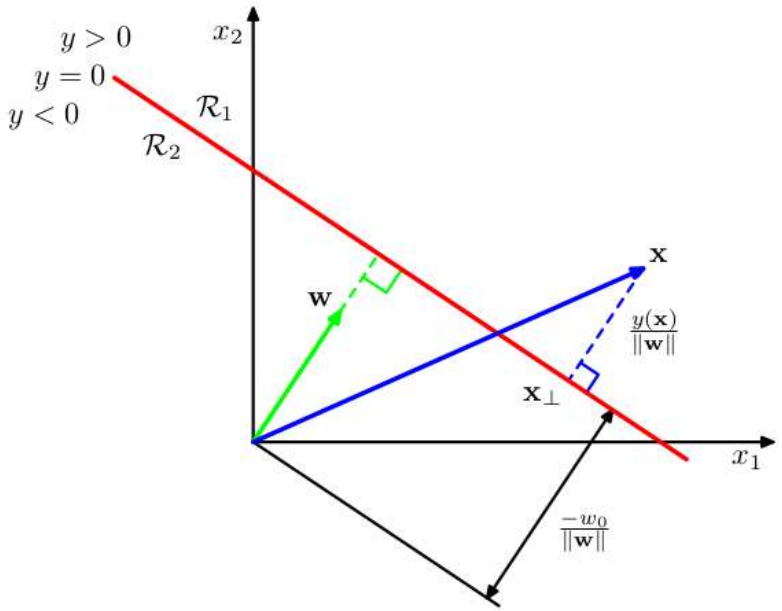


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

- **SVM – non-overlapping classes. The optimization problem …**
  - Since $y(\mathbf{x},\mathbf{w})= w_0 + \mathbf{w}^T \mathbf{x}$, with vector **w** normal to the hyperplane y(**x**,**w**) and **u**=**w**/||**w**|| the unitary normal vector. Then, the vector $\mathbf{x}_\perp$ **x**= **x** - $\mathbf{x}_\perp$= d**u**= d **w**/||**w**|| i.e., is proportional to the unitary normal vector **u**, and **x**= $\mathbf{x}_\perp$ + d **w**/||**w**|| or $\mathbf{x}_\perp$= **x** - d **w**/||**w**||,
  - a point $\mathbf{x}_\perp$ in the hyperplane satisfies 0= $y(\mathbf{x}_\perp,\mathbf{w})= w_0 + \mathbf{w}^T \mathbf{x}_\perp = w_0 + \mathbf{w}^T(\mathbf{x} - d\ \mathbf{w}/||\mathbf{w}||) = w_0 + \mathbf{w}^T\mathbf{x} - d\ \mathbf{w}^T\mathbf{w}/||\mathbf{w}|| = w_0 + \mathbf{w}^T\mathbf{x} - d||\mathbf{w}||^2/||\mathbf{w}|| = w_0 + \mathbf{w}^T\mathbf{x} - d||\mathbf{w}||$, and $d= (w_0 + \mathbf{w}^T\mathbf{x})/||\mathbf{w}|| = y(\mathbf{x},\mathbf{w})/||\mathbf{w}||$,
  - then, the perpendicular distance from a data point to the surface defined by the hyperplane $y(\mathbf{x},\mathbf{w}) = w_0 + \mathbf{w}^T\mathbf{x}$ **is** $d = y(\mathbf{x},\mathbf{w})/||\mathbf{w}||$,
  - since we want that all the data points are well classified: $y(\mathbf{x}_n,\mathbf{w})t_n > 0$ for all points n=1,…,N, then we want that the distance of a point $\mathbf{x}_n$ to the hyperplane is given by: $t_n y(\mathbf{x}_n,\mathbf{w})/||\mathbf{w}||$.

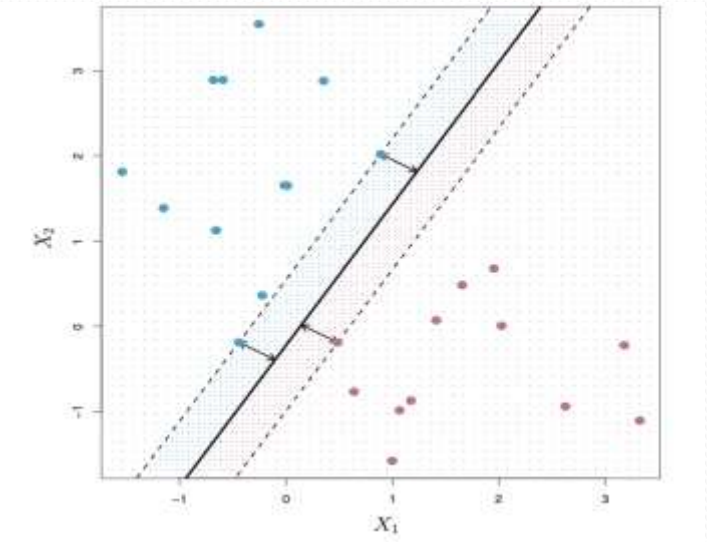
Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



SVM – non-overlapping classes. The optimization problem …

- We want the **maximal margin hyperplane** (optimal separating hyperplane) which is a separating hyperplane that is farthest from the training data,
- for that, we compute the perpendicular distance from each training observation to a given separating hyperplane, that is, the smallest such distance is the minimal distance from the observations to the hyperplane (known as **margin**),
- so, the **maximal margin hyperplane** is the separating hyperplane for which the margin is largest, that is, the hyperplane that has the farthest minimum distance to the training observations,
- the **margin** is the perpendicular distance to the closest point (minimize over all n=1,…,N data points), and moreover we have to find **w** and w_0, so that we maximize the distance to the closest point $d_n = t_n\, y(\mathbf{x_n},\mathbf{w})\ \} / ||\mathbf{w}|| > 0$,

$$\textbf{(P1)} \quad \arg\max_{\{\mathbf{w},w_0\}} \{ \min_{\{n\}} \{d_n\} \} = \arg\max_{\{\mathbf{w},w_0\}} \{ \min_{\{n\}} \{ t_n\, y(\mathbf{x_n},\mathbf{w}) \} / ||\mathbf{w}|| \}$$



SVM – non-overlapping classes. The optimization problem …

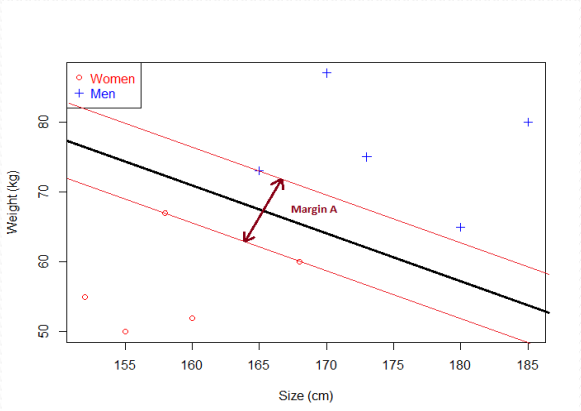
- The **margin** is the perpendicular distance to the closest point (minimize over all n=1,…,N data points), and moreover we have to find **w** and w_0, so that we maximize the distance to the closest point $d_n = t_n\, y(\mathbf{x_n},\mathbf{w}) \} / ||\mathbf{w}|| > 0$,

  (P1) $\arg\max_{\{w,w_0\}}\{ \min_{\{n\}} \{d_n\} \} = \arg\max_{\{w,w_0\}}\{ \min_{\{n\}} \{ t_n\, y(\mathbf{x_n},\mathbf{w}) \} / ||\mathbf{w}|| \}$

- this optimization problem is quite complex to be solved, thus we have to find an equivalent problem. If we rescale $\mathbf{w} \rightarrow k\mathbf{w}$ and $w_0 \rightarrow k w_0$, then we can force that $t_n y(\mathbf{x_n},\mathbf{w}) = t_n(w_0 + \mathbf{w^T x}) = 1$ for the point/s that is/are closest/s to the hyperplane,
- in this case, all the points will satisfy $t_n y(\mathbf{x_n},\mathbf{w}) = t_n(w_0 + \mathbf{w^T x}) \geq 1$, except to the closest one/s that satisfies the $\min_{\{n\}}$ that satisfies $t_n y(\mathbf{x_n},\mathbf{w})=1$. Then the problem becomes: $\text{maximize}_{\{w,w0\}} \{1/||\mathbf{w}||\}$
- The points that hold the equality are said to be **active** and the ones that are >1 are said to be **inactive**. Moreover, $\text{maximize}_{\{w,w0\}}\ 1/||\mathbf{w}||$ is the same than minimize $||\mathbf{w}||^2 = \mathbf{w^T w}$,
- now, the optimization problem becomes:

(P1')	minimize	$\frac{1}{2}\, ||\mathbf{w}||^2$
	s.t.	$t_n(w_0 + \mathbf{w^T x}) \geq 1$, for n=1,…, N
	var	w_0, $\mathbf{w}$

- Once we get w_0^* and $\mathbf{w^*}$, for predicting a new point $\mathbf{x_{N+1}}$, we get $y(\mathbf{x_{N+1}},\mathbf{w^*})$. If positive then it belongs to class C_1, if negative if belongs to class C_2.



SVM – non-overlapping classes. Solving the optimization problem …

- In general, remember that the mapping can be any surface defined by a mapping $\phi(\mathbf{x})$ (not necessarily linear), and $y(\mathbf{x},\mathbf{w}) = w_0 + \mathbf{w}^T \phi(\mathbf{x})$,

- Then, the optimization problem becomes:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\,||\mathbf{w}||^2 \\ &\text{s.t.} && t_n(w_0 + \mathbf{w}^T\phi(\mathbf{x})) \geq 1, \quad \text{for } n=1,\ldots, N \\ &\text{var} && w_0, \mathbf{w} \end{aligned}$$

This quadratic optimization problem (with M features) has computational complexity $O(M^3)$.

- In order to solve this problem, let us obtain the dual problem, defining first the **Lagrangian**:

$$L(w_0, \mathbf{w}, \mathbf{a}) = \tfrac{1}{2}\,||\mathbf{w}||^2 - \sum_{n=1,\ldots,N} a_n \left(t_n (w_0 + \mathbf{w}^T\phi(\mathbf{x})) - 1\right),$$

- where $\mathbf{a}^T=(a_1,\ldots, a_N)$ are the **Lagrange multipliers** (be careful with the sign "–" before the a_n),

- Setting the derivatives with respect w_0, $\mathbf{w}$ equal to 0:

$$\begin{aligned} \mathbf{w} &= \sum_{n=1,\ldots,N} a_n t_n \phi(\mathbf{x}) \\ 0 &= \sum_{n=1,\ldots,N} a_n t_n, \end{aligned}$$

SVM – non-overlapping classes. Solving the optimization problem …

- Eliminating **w** and w_0 from $L(w_0, \mathbf{w}, \mathbf{a})$, the **Lagrange function** becomes:

$$g(\mathbf{a}) = \sum_{n=1,\dots,N} a_n - \tfrac{1}{2} \sum_{n=1,\dots,N} \sum_{m=1,\dots,N} a_n a_m t_n t_m \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_m),$$

- Then, the **dual optimization problem** becomes:

$$\begin{aligned} \text{maximize} \quad & g(\mathbf{a}) = \sum_{n=1,\dots,N} a_n - \tfrac{1}{2} \sum_{n=1,\dots,N} \sum_{m=1,\dots,N} a_n a_m t_n t_m \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_m), \\ \text{s.t.} \quad & \sum_{n=1,\dots,N} a_n t_n = 0 \\ & a_n \geq 0, \text{ for } n=1,\dots, N \\ \text{var} \quad & \mathbf{a} \in R^N \end{aligned}$$

- We know that the function $k(\mathbf{x},\mathbf{y}) = \phi^T(\mathbf{x})\phi(\mathbf{y})$, **is a the kernel function**, and the **dual optimization problem** becomes:

$$\begin{aligned} \text{maximize} \quad & \sum_{n=1,\dots,N} a_n - \tfrac{1}{2} \sum_{n=1,\dots,N} \sum_{m=1,\dots,N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m), \\ \text{s.t.} \quad & \sum_{n=1,\dots,N} a_n t_n = 0 \\ & a_n \geq 0, \quad \text{for } n=1,\dots, N \\ \text{var} \quad & \mathbf{a} \in R^N \end{aligned}$$

SVM – non-overlapping classes. Final solution …

- Remembering that the Lagrangian is:

$$L(w_0, \mathbf{w}, a) = \tfrac{1}{2}\,||\mathbf{w}||^2 - \sum_n a_n \left(t_n (w_0 + \mathbf{w}^T \phi(\mathbf{x})) - 1 \right) = \tfrac{1}{2}\,||\mathbf{w}||^2 - \sum_n a_n (t_n y(\mathbf{x}_n,\mathbf{w}) - 1)$$

- we can apply the KKT conditions to the Lagrangian:

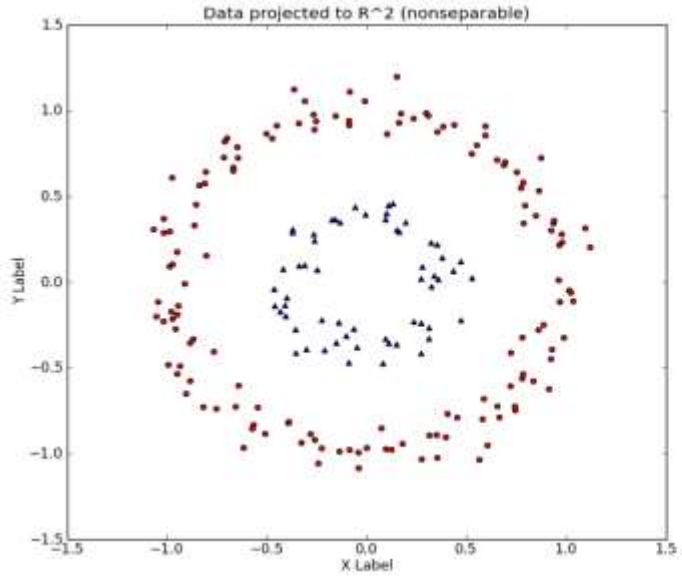
$$\mathbf{w} = \sum_{n=1,\ldots,N} a_n t_n y(\mathbf{x}_n,\mathbf{w})$$

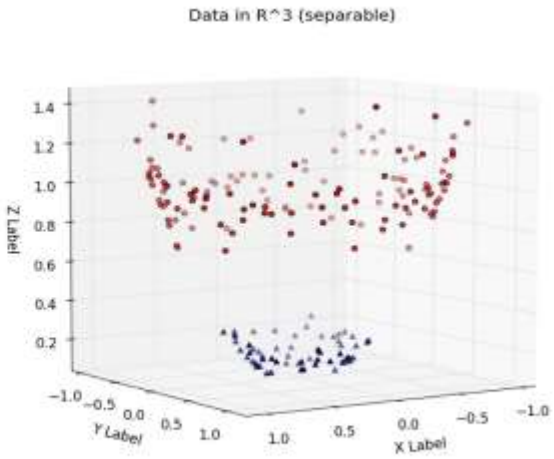
$$a_n = 0 \quad \text{or} \quad t_n y(\mathbf{x}_n,\mathbf{w}) = 1, \quad \text{for each point } n=1,\ldots, N,$$

- since $y(\mathbf{x},\mathbf{w}) = w_0 + \mathbf{w}^T\phi(\mathbf{x}) = w_0 + \sum_{n=1,\ldots,N} a_n t_n \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_n) = w_0 + \sum_{n=1,\ldots,N} a_n t_n k(\mathbf{x}_n, \mathbf{x})$, those points $a_n = 0$, don't participate in the sum and then do not take any role in the prediction of new data points,
- On the other hand, those points, for which $a_n > 0$ (those that satisfy $t_n y(\mathbf{x}_n,\mathbf{w})=1$), are called **support vectors**, and because they satisfy that $t_n y(\mathbf{x}_n,\mathbf{w})=1$, they correspond to points that lie on the maximum margin hyperplanes in the feature space,
- **SVM algorithm:**
  1. Compute the support vectors a_n, solving the dual problem,
  2. Compute w_0 as the average of support vectors: $w_0 = 1/N_S \sum_{n\in S} (t_n - \sum_{m\in S} a_m t_m k(\mathbf{x}_n,\mathbf{x}_m))$, where S is the set of support vectors and N_S is the number of support vectors,
  3. Make predictions for new points **x** using: $y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{n=1,\ldots,N} a_n t_n k(\mathbf{x}_n, \mathbf{x})$,

The dual problem has computational complexity N (maybe disadvantageous with respect small M), but allows the use of kernels (which is always advantageous).

- **SVM – non-overlapping classes. The kernel trick …**
  - **Kernel trick (kernel substitution):** data that is not separable can be separated at higher dimensional spaces,



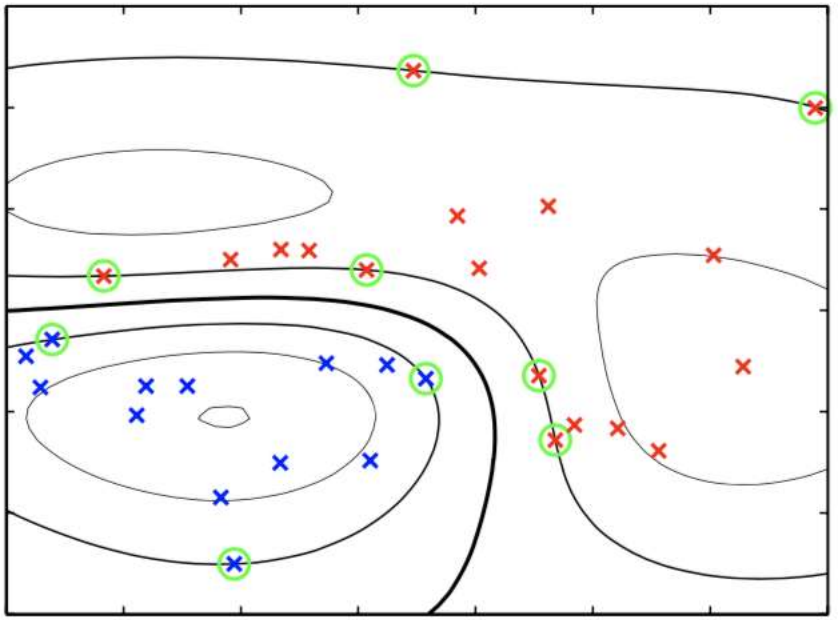


  - The general idea is that, if we have an algorithm formulated in such a way that the input vector x enters only in the form of scalar products, then we can **replace** in the dual problem where the lagrange multipliers are obtained as the scalar product with any choice of a kernel: $k(x_n, x_m) = \phi^T(x_n)\,\phi(x_m)$

1. **using inner product:** $L(a) = \Sigma_n a_n - \frac{1}{2} \Sigma_n \Sigma_m a_n a_m t_n t_m \phi^T(x_n)\,\phi(x_m)$,
2. **using a kernel:** $L(a) = \Sigma_n a_n - \frac{1}{2} \Sigma_n \Sigma_m a_n a_m t_n t_m k(x_n, x_m)$,

SVM – non-overlapping classes. An example …

Figure 7.2 Example of synthetic data from two classes in two dimensions showing contours of constant $y(\mathbf{x})$ obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



SVM hyperparameters: in the non-overlapping class problem, we do not have SVM own hyperparameters, however, we have to add the hyperparameters of the Kernel function, e.g. if RBF is used, we will have the hyperparameter that controls the distance between points and that is called gamma or sigma depending on the textbook. Use **cross-validation** to optimize these hyperparameters.

- **SVM - Overlapping class distributions …**
  - **Assumption made:** the training points were linearly separable in the feature space **ϕ(x)**, it is to say, all the points of the class lie in the side of the margin (see left figure). Then, the SVM will give an exact separation of the training data, even if the decision boundary is not linear,
  - However, many times, **the points are misclassified**, meaning, that some points of the class lie inside the margin (left) or both inside the margin or in the other class side (right),

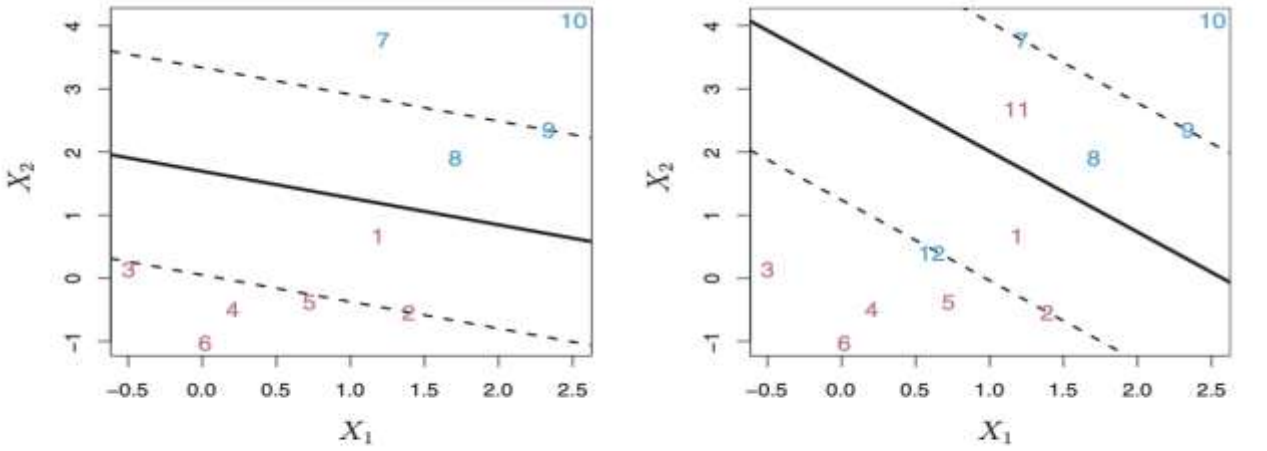


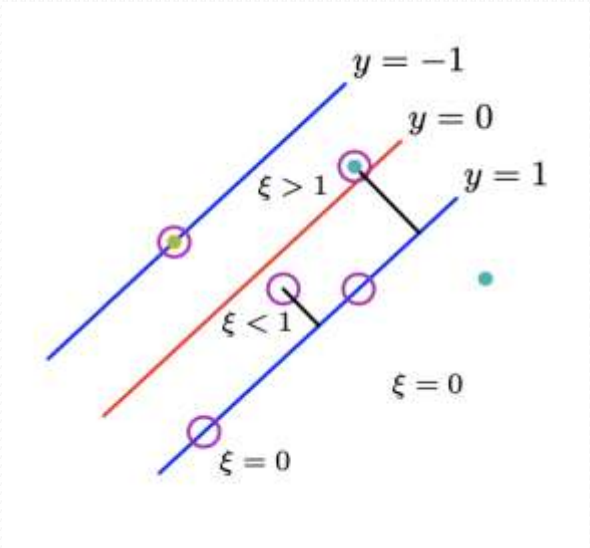
FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation* 1 *is on the wrong side of the margin.* Blue observations: *Observations* 7 *and* 10 *are on the correct side of the margin, observation* 9 *is on the margin, and observation* 8 *is on the wrong side of the margin. No observations are on the wrong side of the hyperplane.* Right: *Same as left panel with two additional points,* 11 *and* 12. *These two observations are on the wrong side of the hyperplane and the wrong side of the margin.*

SVM - Overlapping class distributions …

We modify the previous approach, to include this possibility, adding a penalty that increases with the distance to the boundary. Thus, we **introduce a slack variable $\xi_n \geq 0$** for each training point *i*=1, …, N.

- Slack variables are defined as $\xi_n = 0$ for training points that are inside the correct boundary margin, and
- are defined as $\xi_n = |t_n - y(x_n)|$ for other points. That means that a data point that is on the decision boundary, ($y(x_n)=0$ and $t_n = \{-1, 1\}$) will have $\xi_n=1$, and points with $\xi_n>1$ will be misclassified.
- The classification constraint $t_n y(\mathbf{x_n},\mathbf{w}) = t_n(w_0 + \mathbf{w}^T\mathbf{x}) \geq 1$, becomes:

$$t_n y(\mathbf{x_n},\mathbf{w}) = \geq 1 - \xi_n \quad \text{with } \xi_n \geq 0$$



Data points in circles are support vectors.

1. Data points for which **$\xi_n = 0$** are correctly classified and are either on the margin or on the correct side of the margin,
2. data for which **$0 < \xi_n \leq 1$** lie inside the margin, but on the correct side of the decision boundary, and
3. Data points for which **$\xi_n > 1$** lie on the wrong side of the decision boundary and are misclassified, with a penalty that increases linearly with ξ.

- **SVM - Overlapping class distributions …**
  - Now, the objective is instead to minimize $\frac{1}{2}\,||\mathbf{w}||^2$, to minimize $C\sum_{n=1,\ldots,N}\xi_n + \frac{1}{2}\,||\mathbf{w}||^2$, where the constant C>0, controls the trade-off between the slack variable penalty and the margin, if C is large, then ξ_n will be small (or cero), and we have the SVM for separable data (previous slides). The optimization problem becomes:

$$\begin{aligned} &\text{minimize} && C\sum_{n=1,\ldots,N}\xi_n + \tfrac{1}{2}\,||\mathbf{w}||^2 \\ &\text{s.t.} && t_n(w_0 + \mathbf{w}^T\phi(\mathbf{x})) = t_n y(\mathbf{x_n},\mathbf{w}) \geq 1-\xi_n, \quad \text{for } n=1,\ldots,N \\ & && \xi_n \geq 0 \\ &\text{var} && w_0,\ \mathbf{w} \end{aligned}$$

  - The **corresponding Lagragian** is then:

$$L(w_0, \mathbf{w}, a) = C\sum_{n=1,\ldots,N}\xi_n + \tfrac{1}{2}\,||\mathbf{w}||^2 - \sum_{n=1,\ldots,N} a_n\,(t_n\,(w_0 + \mathbf{w}^T\phi(\mathbf{x})) - 1 + \xi_n) - \sum_{n=1,\ldots,N}\mu_n\,\xi_n,$$

Where a_n and μ_n are Lagrange multipliers.

- **SVM - Overlapping class distributions …**
  - Now, the KTT conditions are (for n=1, …, N):

$$t_n y(\mathbf{x_n}) - 1 + \xi_n \geq 0 \quad \Rightarrow \quad t_n(w_0 + \mathbf{w}^T\phi(\mathbf{x})) - 1 + \xi_n \geq 0$$
$$a_n(t_n y(\mathbf{x_n}) - 1 + \xi_n) = 0 \quad \Rightarrow \quad a_n(t_n(w_0 + \mathbf{w}^T\phi(\mathbf{x})) - 1 + \xi_n) = 0$$
$$\mu_n \xi_n = 0$$
$$a_n \geq 0$$
$$\mu_n \geq 0$$
$$\xi_n \geq 0$$

  - Now, we optimize the Lagrangian with respect to **w**, w_0 and ξ_n:

$$\partial L/\partial \mathbf{w} = 0 \quad \Rightarrow \quad \mathbf{w} = \Sigma_{n=1,\ldots,N}\, a_n t_n\, \phi(\mathbf{x}_n)$$
$$\partial L/\partial w_0 = 0 \quad \Rightarrow \quad \Sigma_{n=1,\ldots,N}\, a_n t_n = 0$$
$$\partial L/\partial \xi_n = 0 \quad \Rightarrow \quad a_n = C - \mu_n \geq 0 \rightarrow\ C \geq \mu_n$$

  - Using all these equations, we can see that the Lagrange function can be expressed as:

$$g(\mathbf{a}) = \text{minimize}_{\{\mathbf{w},w0\}}\, L(\mathbf{w},w_0,\mathbf{a}) = \Sigma_{n=1,\ldots,N}\, a_n - 1/2\, \Sigma_{n=1,\ldots,N}\, \Sigma_{m=1,\ldots,N}\, a_n a_m t_n t_m k(x_n,x_m)$$

That is the same that in the separable case, but with different constraints

- **SVM - Overlapping class distributions …**
  - The **dual problem** becomes, then:

$$\begin{aligned} &\text{maximize} && \sum_{n=1,\dots,N} a_n - 1/2 \sum_{n=1,\dots,N} \sum_{m=1,\dots,N} a_n a_m t_n t_m k(x_n, x_m) \\ &\text{s.t.} && 0 \le a_n \le C \\ & && \sum_{n=1,\dots,N} a_n t_n = 0 \\ &\text{var} && a \in R^N \end{aligned}$$

This problem again is a quadratic optimization problem. We operate as in the separable problem, remembering the Lagrangian form and the KTT conditions. It is to say:

- A subset of data points may have $a_n = 0$, and do not participate in the predictive model $y(x) = w_0 + \sum_{-n} a_n t_n k(x_n, x)$,
- The points that contribute have $a_n \geq 0$, and then satisfy $t_n y(x_n) = 1 - \xi_n$,
  - If a point $a_n < C$ (and since $a_n = C - \mu_n \geq 0$), then implies that $\mu_n > 0$, that requires that $\xi_n = 0$ (see KKT where $\mu_n \xi_n = 0$), and then **these points lie on the margin**,
  - Points which $a_n = C$, **can lie inside the margin** and can either be correctly classified if $\xi_n \leq 1$, or misclassified if $\xi_n > 1$,
- To calculate w_0 note that if $0 < a_n < C$, $\xi_n = 0$, and then $t_n y(x_n) = 1 - \xi_n = 1$, that means that:

$$t_n y(x_n) = t_n(w_0 + \sum_{m \in S} a_m t_m k(x_m, x_n)) = 1 \quad \Rightarrow \quad w_0 = 1/N_M \sum_{n \in S} (t_n - \sum_{m \in S} a_m t_m k(x_n, x_m))$$

Where S is the set of support vectors, and N_M is the number of support vectors that satisfy the condition $0 < a_n < C$. **Support vectors** are those ones that lie on the margin ($a_n < C$) or inside the margin ($a_n = C$).

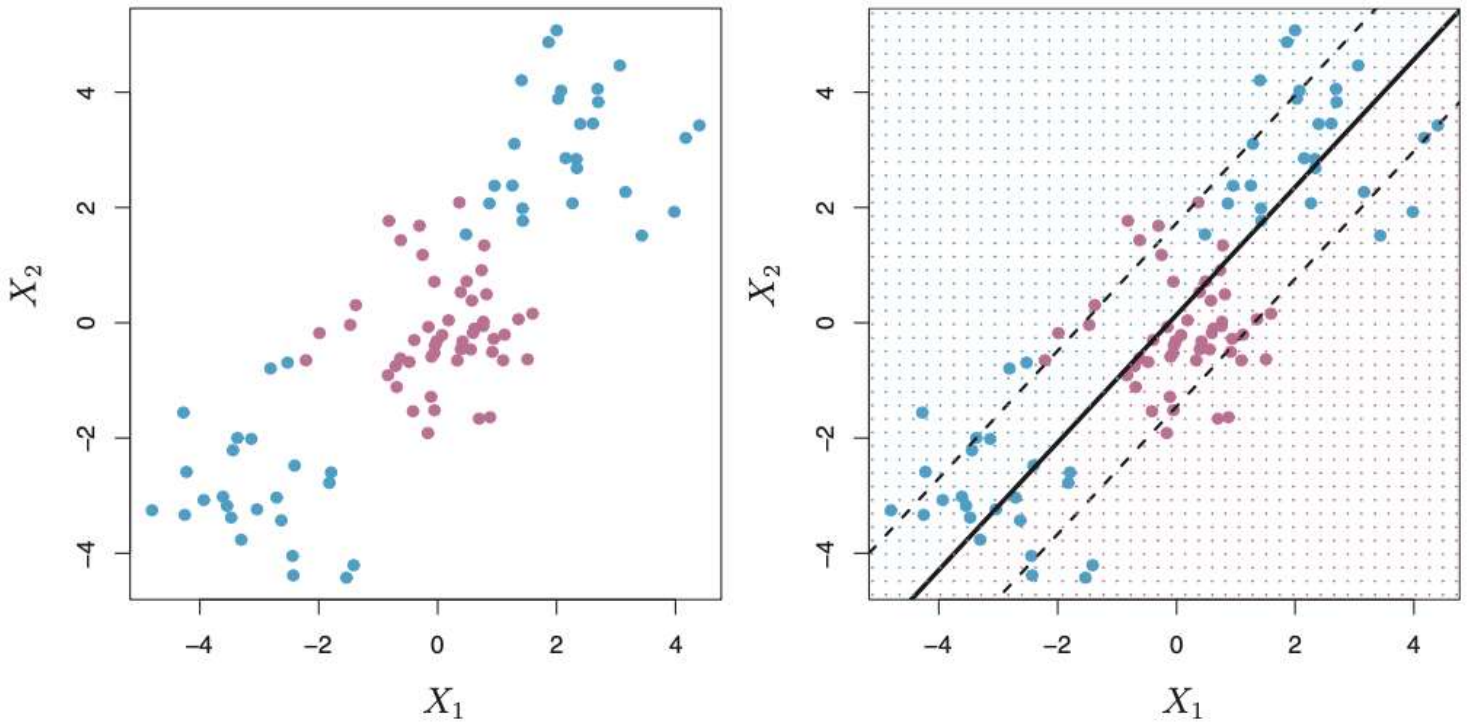


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

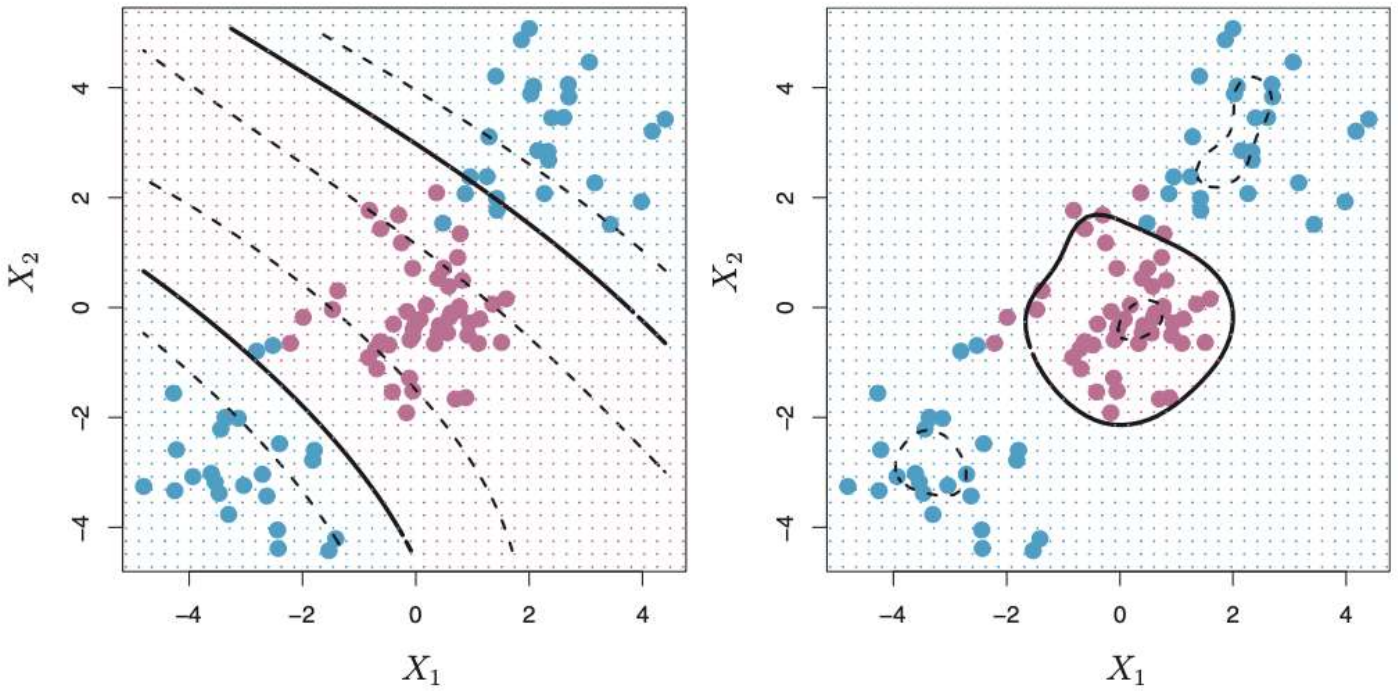


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

SVM – non-overlapping classes. An example …

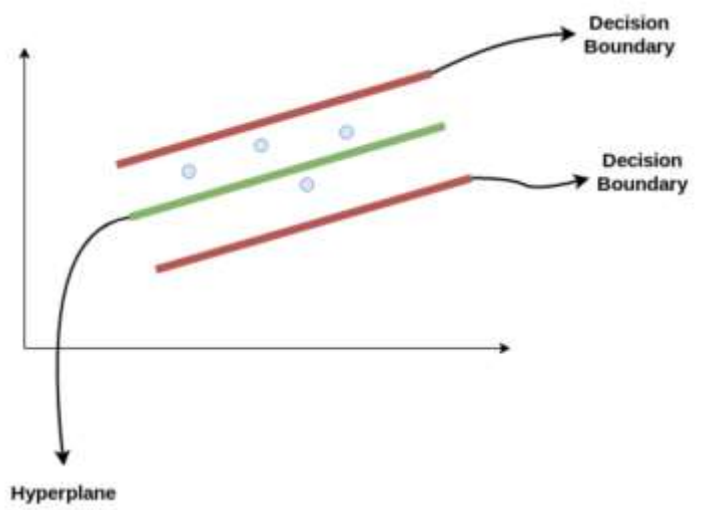
- **SVM hyperparameters:** C is the hyperparameter that belongs to SVM, while we have to add the hyperparameters of the Kernel function, e.g. if RBF is used we will have the hyperparameter that controls the distance between points and that is called gamma or sigma depending on the textbook. Use **cross-validation** to optimize these hyperparameters.
- **Interpretation of tuning parameter C:** instead of having $C\Sigma_{n=1,\ldots,N}\ \xi_n$ in the objective function, there are versions that have it in the constraints: $\Sigma_{n=1,\ldots,N}\ \xi_n \leq C$. The problem is the same but since C bounds $\Sigma_{n=1,\ldots,N}\ \xi_n$, then it determines the number and severity of the violations to the margin and to the hyperplane that we will tolerate,
  - If C=0, then the budget for violations is 0, and ξ_n=0 for all n=1,…,N, which accounts for the non-overlapping case,
  - For C>0, then no more than C violations (observations on the wrong part pf the hyperplane) are allowed, since then ξ_n>1, and $\Sigma_{n=1,\ldots,N}\ \xi_n \leq C$,
  - As **C increases**, we become more tolerant with violations and the margin widens, conversely, if **C decreases**, we become less tolerant with violations and the margin narrows.
  - Then, **C small** means narrow margins, with low number of violations, which makes the training data well fitted (**low bias and high variance**) → low number of support vectors,
  - However, when **C is large**, means that the margin is wider, and the are more violations, which makes the classifier with **larger bias but less variance** → large number of support vectors.

- **SVM - K>2 classes …**
  - SVM is basically a K=2 class classifier. There are modifications to work with K>2 classes, but they don't work at all well,

  - Several options are the following:
    1. **One versus the rest/all**: construct K separate SVM's, each using a C_k class (positive class) against the other K-1 classes (negative class), but in general poses inconsistent results,
    2. **One versus one:** organize K(K-1)/2 SVM's (all pairs), but again it can give inconsistent results, and for large K, it can be increase the computational cost,
    3. **Others:** organize pairwise classifiers in directed acyclic graphs (DAGSVM, Platt 2000), or Crammer & Singer (2000) use an optimization method for solving multiclass SVM.

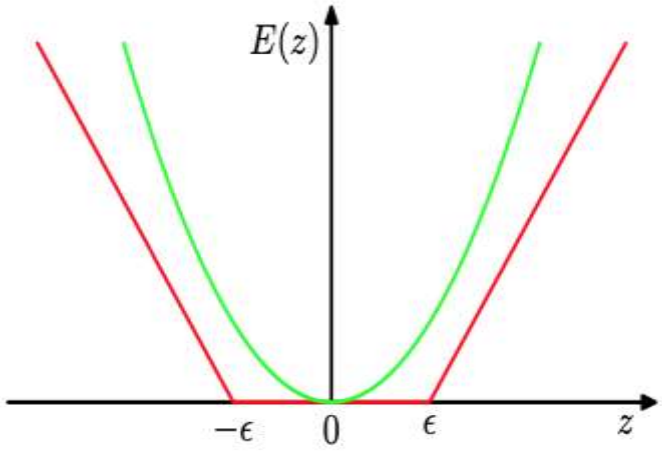
Regression (non-linear): Support-Vector Regression

Problem: find a function $y(\mathbf{x}_n)$ with at ε-most (ε-tube) deviation from target t_n. Solve a modified version of the SVM considering errors ε.



If $y(\mathbf{x},\mathbf{w})=\mathbf{w}^T\mathbf{x}+b$ (linear function), the system can be solved using an optimization problem, where we want to obtain the coefficients of the hyperplane that determines that $\mathbf{w}^T\mathbf{x}_n+b$ has a margin ε with respect the target data. Condition to be inside the tube is: $y(\mathbf{x}_n,\mathbf{w})- \varepsilon \leq t_n \leq y(\mathbf{x}_n,\mathbf{w})+ \varepsilon$.

minimize	$C\sum_{n=1,\dots,N} E[y(\mathbf{x}_n,\mathbf{w})-t]+ 1/2\|\|\mathbf{w}\|\|^2$
s.t.	$t_n - \mathbf{w}^T\mathbf{x}_n - b \leq \varepsilon$ (above boundary)
	$\mathbf{w}^T\mathbf{x}_n + b - t_n \leq \varepsilon$ (below boundary)
var	$\mathbf{w}$, b



The idea is that instead of using a quadratic error function, we use a ε-insensitive error function defined as:

$E_\varepsilon[y(\mathbf{x}_n,\mathbf{w})-t]= |y(\mathbf{x}_n,\mathbf{w})-t_n|-\varepsilon$ if $|y(\mathbf{x}_n,\mathbf{w})-t_n|\geq \varepsilon$,

$= 0$ otherwise

In green → a quadratic error function, and

In red → ε-insensitive error function that increases linearly with respect the insensitive region.

Regression (non-linear): Support-Vector Regression

Problem: find a function $y(\mathbf{x}_n)$ with at ε-most deviation from target t_n. Solve a modified version of the SVM considering errors ε. Find the hyperplane that minimizes the error by maximizing the margin tolerating certain error ε. ξ and ξ^* are **slack variables** to account for unfeasibe data (data that is out of the ε margins). Then, the slack variables meaning is:

$\xi_n > 0$ if $t_n > y(\mathbf{x}_n,\mathbf{w}) + \varepsilon$ (above the margin → gives an error)

$\xi^*_n > 0$ if $t_n < y(\mathbf{x}_n,\mathbf{w}) - \varepsilon$ (below the margin → gives an error)

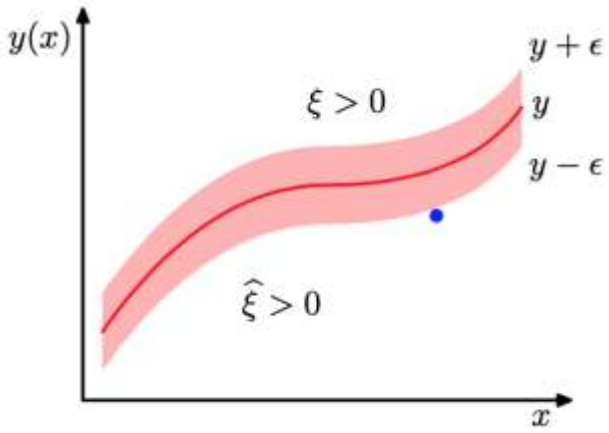
Now, the condition for a target point to be inside the ε-tube is $y(\mathbf{x}_n,\mathbf{w}) - \varepsilon \le t_n \le y(\mathbf{x}_n,\mathbf{w}) + \varepsilon$. Introducing the slack variables $y(\mathbf{x}_n,\mathbf{w}) - \varepsilon - \xi^*_n \le t_n \le y(\mathbf{x}_n,\mathbf{w}) + \varepsilon + \xi_n$. Then, we will have the following constraints:

$t_n \le y(\mathbf{x}_n,\mathbf{w}) + \varepsilon + \xi_n$ → $t_n - y(\mathbf{x}_n,\mathbf{w}) \le \varepsilon + \xi_n$ (above constraint)

$y(\mathbf{x}_n,\mathbf{w}) - \varepsilon - \xi^*_n \le t_n$ → $y(\mathbf{x}_n,\mathbf{w}) - t_n \le \varepsilon + \xi^*_n$ (below constraint)

And the error function is $E_\varepsilon[y(\mathbf{x}_n,\mathbf{w})-t] = |y(\mathbf{x}_n,\mathbf{w})-t| - \varepsilon$ if $|y(\mathbf{x}_n,\mathbf{w})-t| \ge \varepsilon$, and we can substitute the expression $C\sum_{n=1,\dots,N} E[y(\mathbf{x}_n,\mathbf{w})-t]$ by $C\sum_{n=1,\dots,N} (\xi_n + \xi^*_n)$.

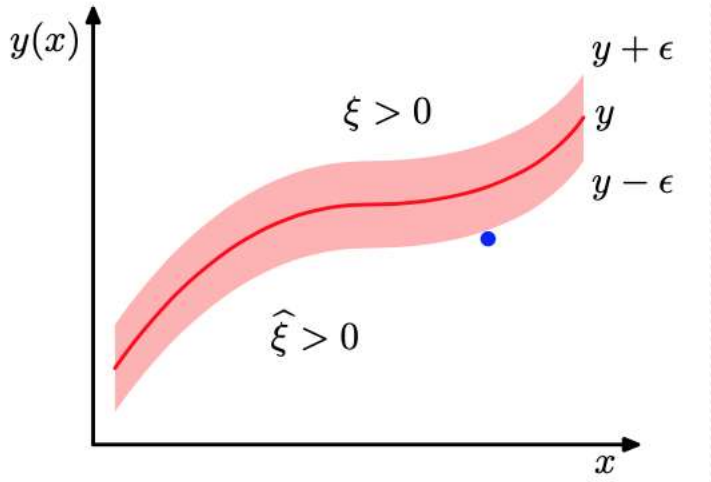
Illustration of SVM regression, showing the regression curve together with the ϵ-insensitive 'tube'. Also shown are examples of the slack variables ξ and $\widehat{\xi}$. Points above the ϵ-tube have $\xi > 0$ and $\widehat{\xi} = 0$, points below the ϵ-tube have $\xi = 0$ and $\widehat{\xi} > 0$, and points inside the ϵ-tube have $\xi = \widehat{\xi} = 0$.



Regression (non-linear): Support-Vector Regression

Problem: find a function f(x) with at ε-most deviation from target y. Solve a modified version of the SVM considering errors ε. Find the hyperplane that minimizes the error by maximizing the margin tolerating certain error ε. ξ and ξ^* are **slack variables** to account for unfeasibe data (data that is out of the ε margins). Then, the problem becomes:

minimize	$C \sum_{n=1,\ldots,N} (\xi_n + \xi^*_n) + 1/2 \, \|\|\mathbf{w}\|\|^2$
s.t.	$t_n - \mathbf{w}\cdot\mathbf{x_n} - b \leq \varepsilon + \xi_n$
	$\mathbf{w}^T\cdot\mathbf{x_n} + b - t_n \leq \varepsilon + \xi^*_n$
	$\xi_n, \xi^*_n \geq 0$
var	$\mathbf{w}$, b



- If the function is non-linear use other kernel functions.
- Look to support material Alex. J. Smola & B. Scholkopf, "**A tutorial on SVR**" for seeing the dual problem solution and support vectors interpretation,
- **Hyperparameters**, e.g. C, ε and others that appears with the kernels and have to be obtained using cross-validation.
- **Relevance Vector Machines (RVM):** Bayesian formulation of the SVM/SVR that allows to obtain the posterior distribution. Obtains sparser solutions and then in general is faster than SVM.

- **Regression (non-linear): Support-Vector Regression**

  Problem: regression of data that does not follow a linear pattern.