Support Vector Machines

Statistical Learning

Master in Big Data. University of Santiago de Compostela

Manuel Mucientes

Introduction

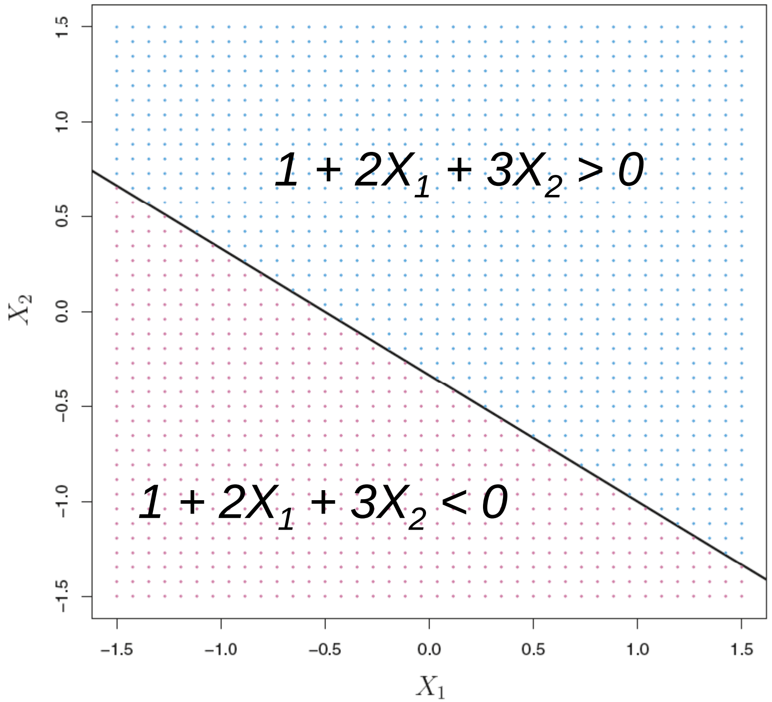
- Support Vector Machines (SVMs) are one of the best classifiers
- SVMs are a generalization of the maximal margin classifier
- Maximal margin classifiers require that the classes are separable by a linear boundary
- Support vector classifiers are an extension of maximal margin classifiers
- SVMs extend support vector classifiers to accommodate non-linear boundaries

Hyperplanes

- In a p-dimensional space, a hyperplane is a flat affine (needs not to pass through the origin) subspace of dimension p-1

- $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$

- A hyperplane divides a p-dimensional space into two halves
  - $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$
  - $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$



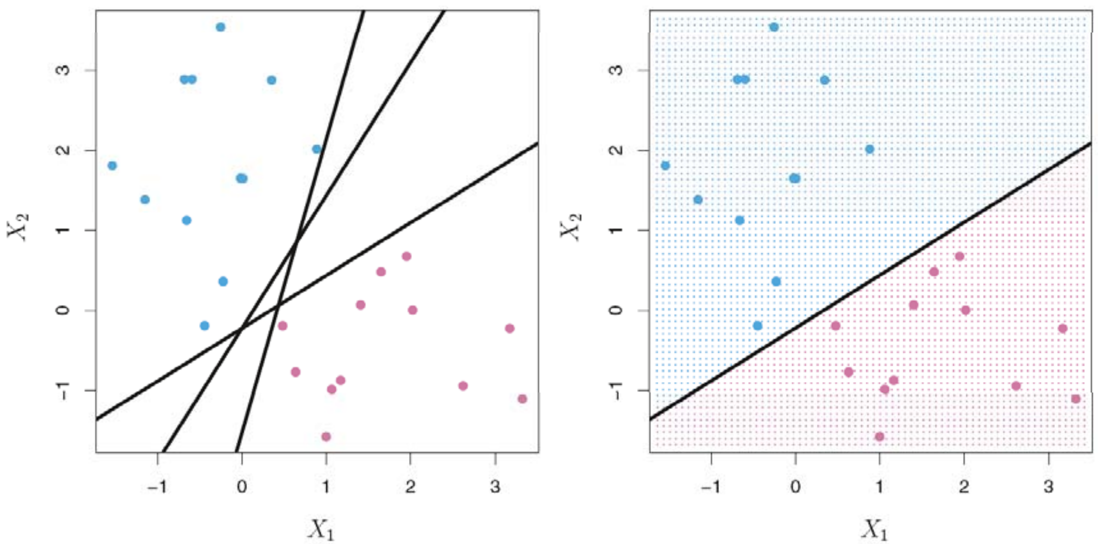
© Manuel Mucientes

Classification using a Separating Hyperplane

- Separating hyperplane:
  - $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0$ if $y_i = 1$
  - $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0$ if $y_i = -1$
  - $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$
- Classify a test observation based on the sign of:

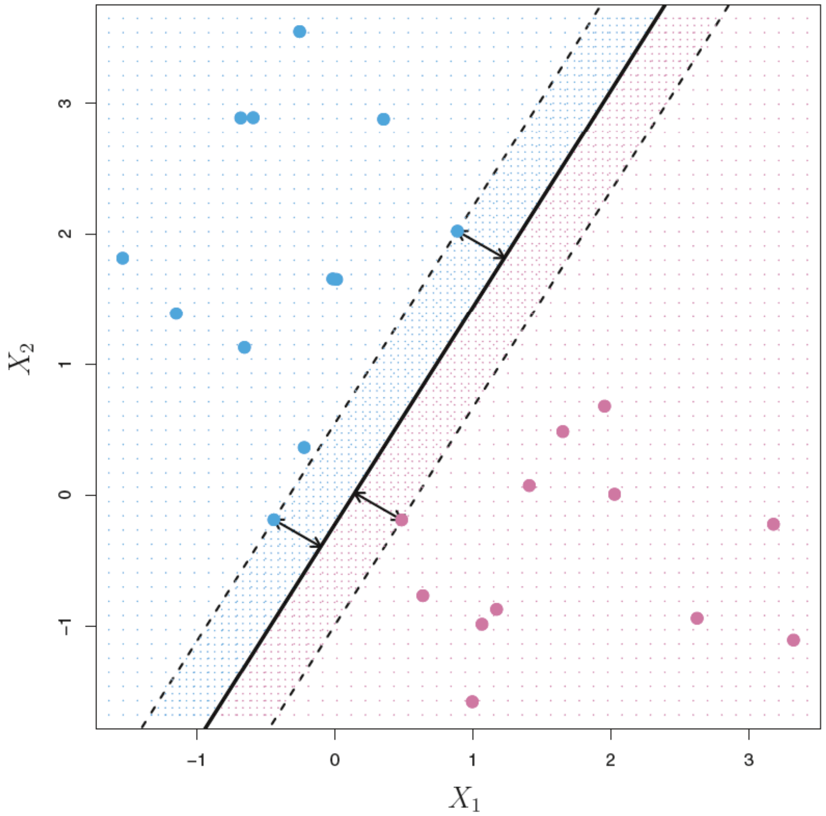
$$f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$$

- The magnitude of $f(x^*)$ gives the confidence
- This classifier leads to a linear decision boundary



Maximal Margin Classifier

- If data is separable by a hyperplane, there exist an infinite number of such hyperplanes
- Maximal margin hyperplane (optimal separating hyperplane): separating hyperplane farthest from the training observations
  - Maximal Margin Classifier (MMC)
- MMC can lead to overfitting when p is large
- Support vectors: observations in p-dimensional space that "support" the hyperplane
  - If they were moved the maximal margin hyperplane would move as well



© Manuel Mucientes

Maximal Margin Classifier (ii)

- Solution to the optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \quad \forall\, i = 1, \ldots, n$$

  - Second condition: each observation in the correct side, at least at a distance M
  - First condition: adds meaning to the second constraint; distance to the hyperplane

- Classification rule: $G(x) = \text{sign}[x^T\beta + \beta_0]$

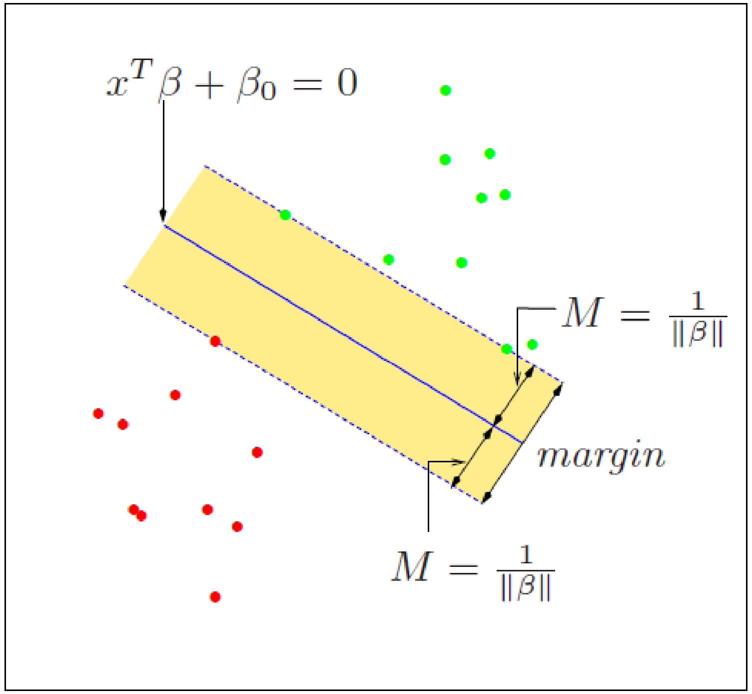
© Manuel Mucientes

Maximal Margin Classifier (iii)

- We can arbitrarily set $||\beta||=1/M$

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2$$

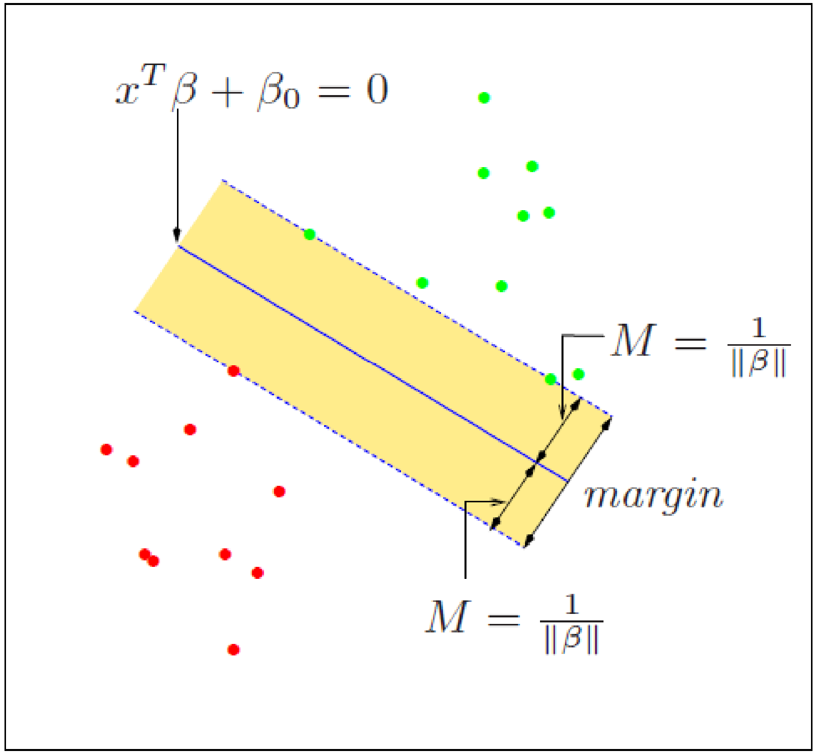
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1,\ i = 1, \dots,$$



- Solution:
  - $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i,$ (1)
  - $\alpha_i[y_i(x_i^T\beta + \beta_0) - 1] = 0\ \forall i.$ (2)
    - α_i: Lagrange multipliers obtained by solving the optimization problem

Maximal Margin Classifier (iv)

- $\alpha_i[y_i(x_i^T\beta + \beta_0) - 1] = 0 \; \forall i.$ (2)
- if $\alpha_i > 0$, then $y_i(x_i^T\beta + \beta_0) = 1$:
  - x_i is in the edge of the margin
- if $y_i(x_i^T\beta + \beta_0) > 1$: $\alpha_i = 0$
  - x_i is outside the margin

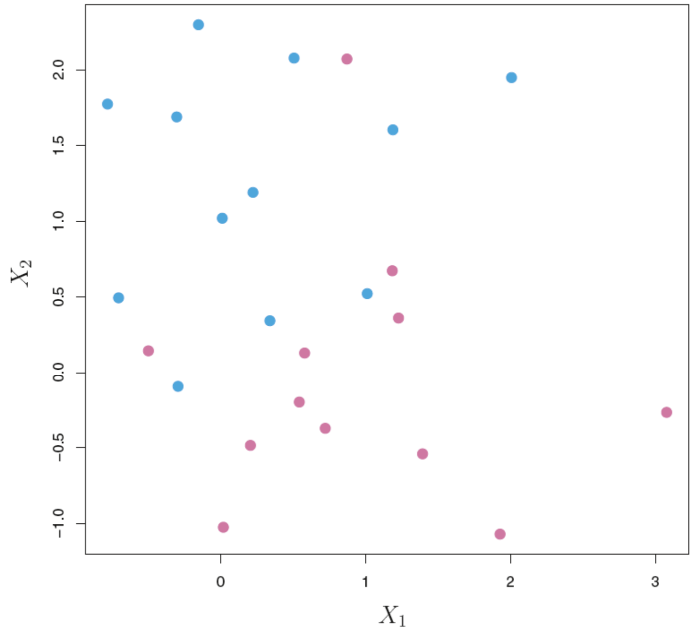


- Support vectors: x_i with $\alpha_i > 0$
- β is defined as a linear combination of the support vectors (eq. 1)
- β_0 is obtained solving eq. 2 for any support vector

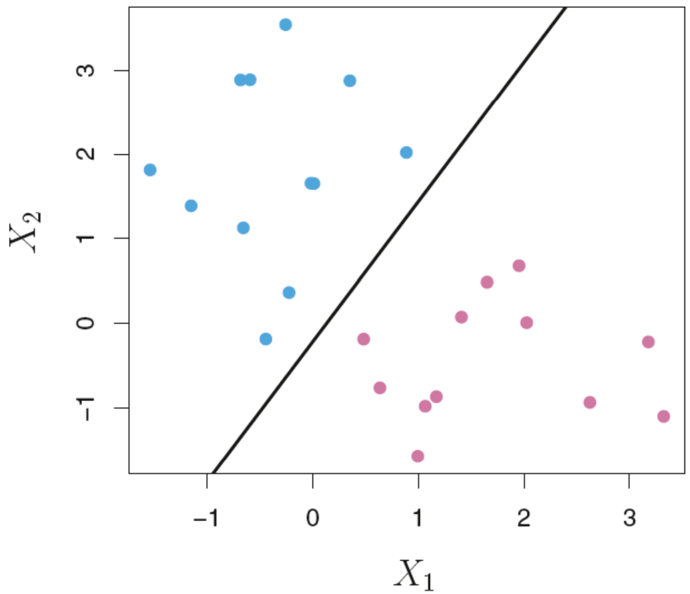
© Manuel Mucientes

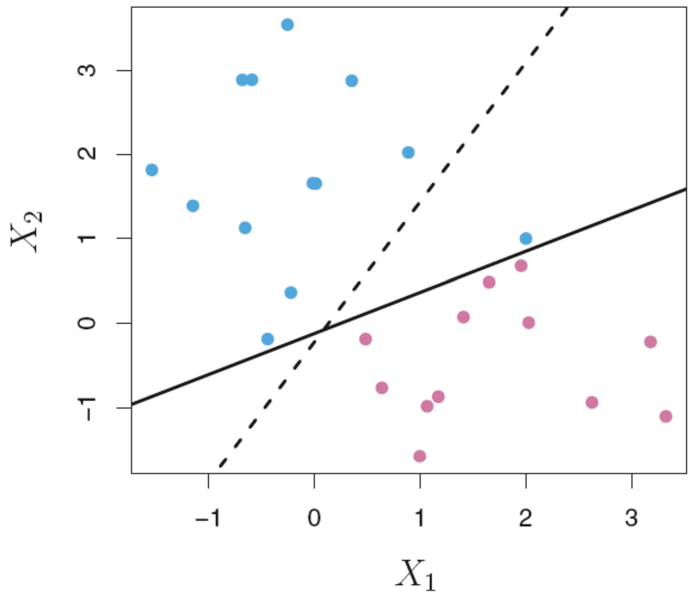
Support Vector Classifiers

- No separating hyperplane exists



- Sometimes, a classifier based on a separating hyperplane is not desirable
  - Extremely sensitive to one observation: overfitting



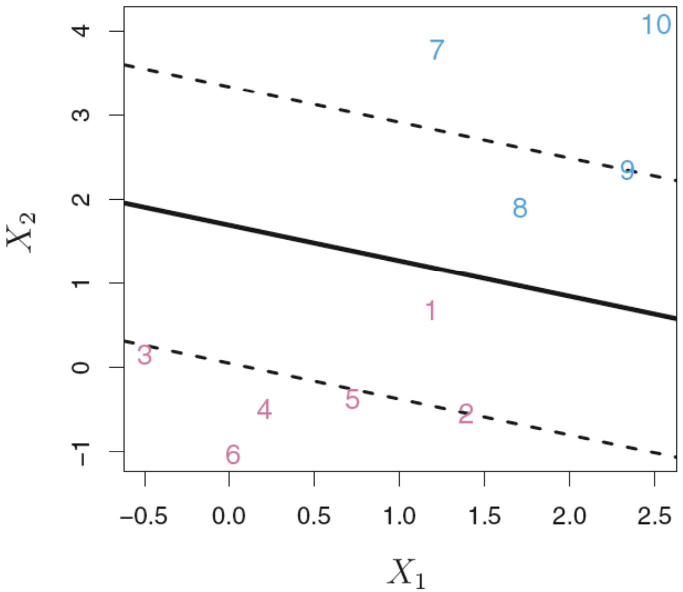


© Manuel Mucientes

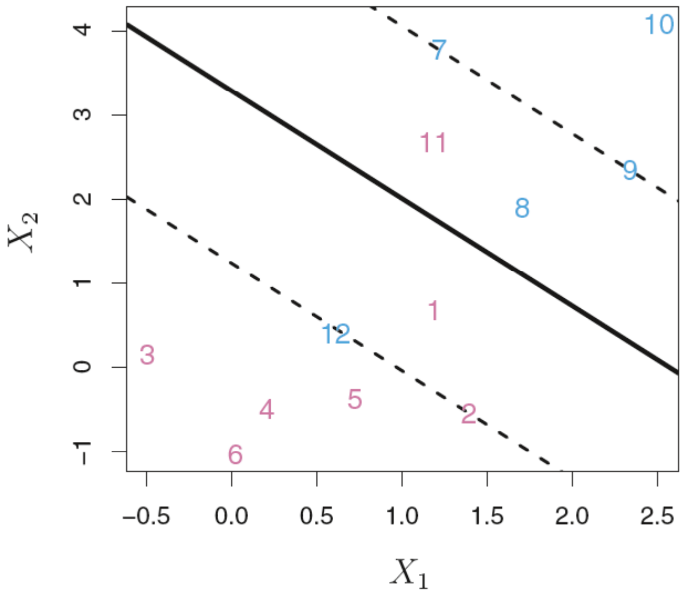
Support Vector Classifiers (ii)

- A classifier that does not perfectly separate the two classes
  - Greater robustness to individual observations
  - Better classification of most training observations
- Soft margin: allow some observations to be in the incorrect side of the margin, or even in the incorrect side of the hyperplane

On the margin: 2, 9
Wrong side of the margin: 1, 8



On the margin: 2, 7, 9
Wrong side of the margin: 1, 8
Wrong side of the hyperplane: 11, 12



© Manuel Mucientes

Support Vector Classifiers (iii)

- The hyperplane separates most of the training data, but may misclassify a few observations

- Optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

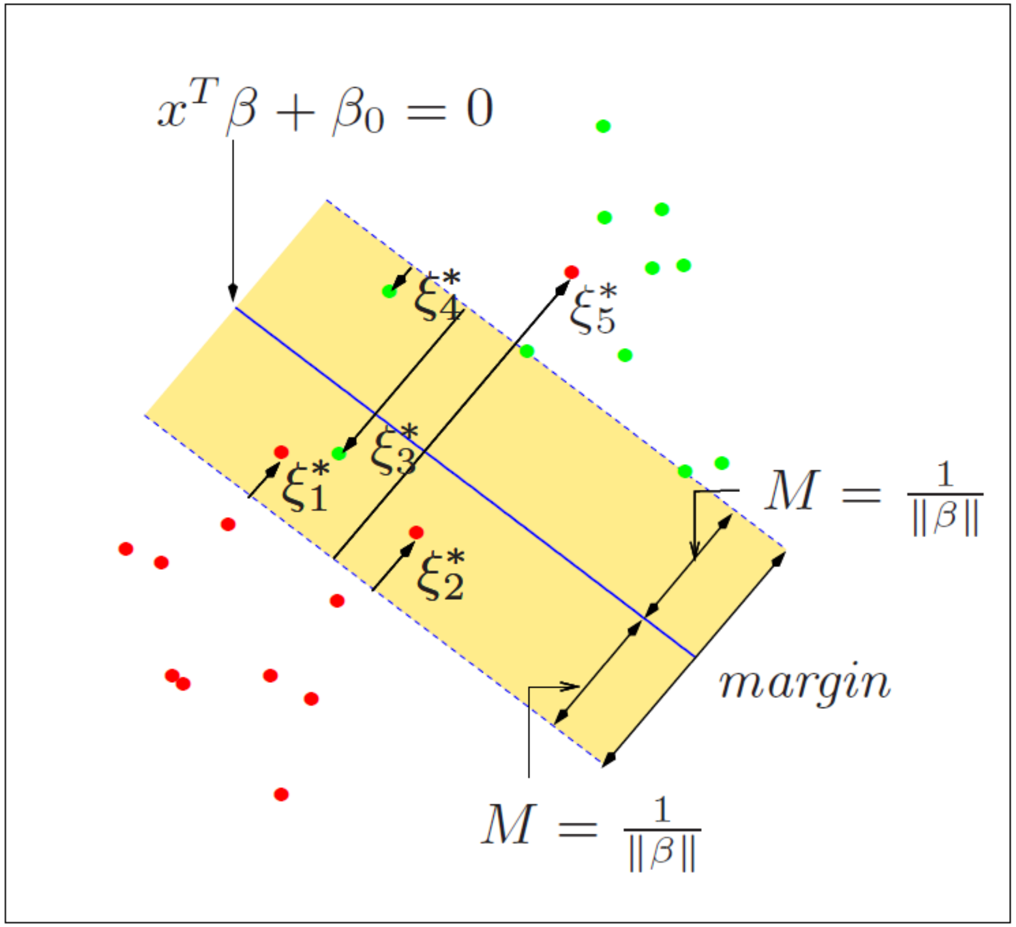
$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq \textit{constant}$$

Support Vector Classifiers (iv)

- ε_i tells where the i-th observation is located: percentage of M
  - $\varepsilon_i=0$: observation in the correct side of the margin
  - $\varepsilon_i>0$: observation in the wrong side of the margin
  - $\varepsilon_i>1$: observation in the wrong side of the hyperplane (misclassification)
- Bounding $\sum\varepsilon_i$ to a constant bounds the total number of training misclassifications



© Manuel Mucientes

Support Vector Classifiers (v)

- Rephrasing the problem:

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad \xi_i \geq 0,\ y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i\ \forall i$$

- *C* replaces the constant (proportional to the inverse of the constant)
  - Can be interpreted as the inverse of a regularization parameter
  - Separable case: *C*=∞

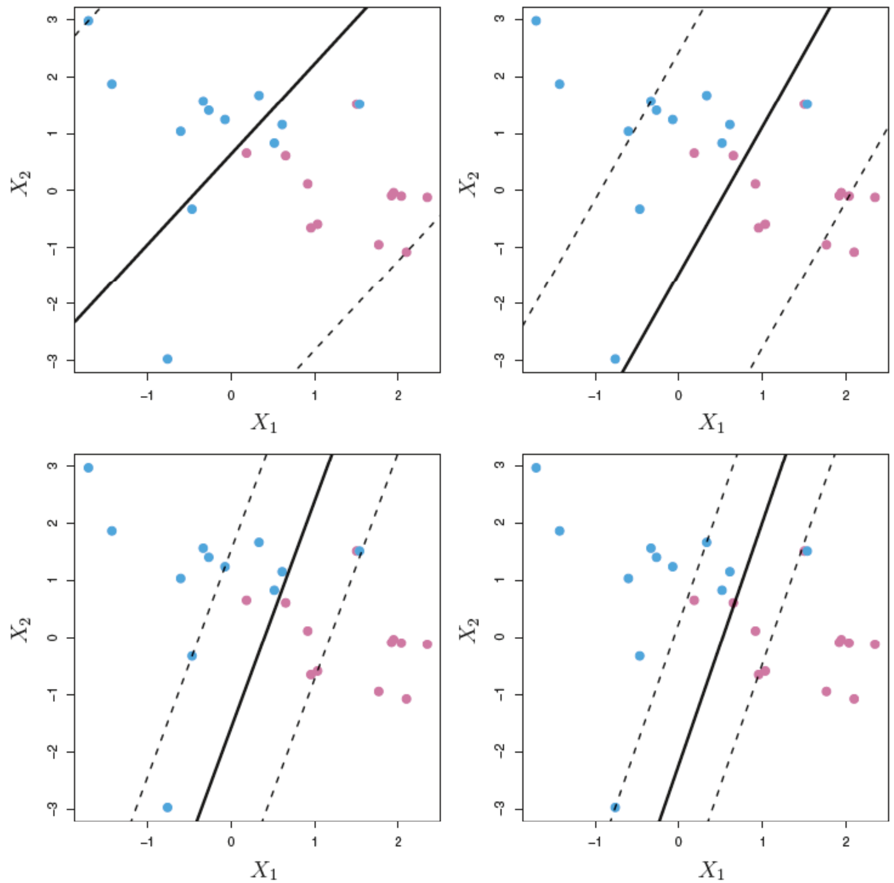
Support Vector Classifiers (vi)

- Solution: $\alpha_i,\ \mu_i,\ \xi_i \geq 0\ \forall i$
  - $\alpha_i[y_i(x_i^T\beta + \beta_0) - (1-\xi_i)] \;=\; 0,$ (1)
  - $\mu_i\xi_i \;=\; 0,$ (2)
  - $y_i(x_i^T\beta + \beta_0) - (1-\xi_i) \;\geq\; 0,$ (3)
  - $\beta \;=\; \sum_{i=1}^{N}\alpha_i y_i x_i$ (4) $\alpha_i \;=\; C - \mu_i,\ \forall i$ (5)
- Support vectors: $\alpha_i>0$ (eq. 1)
  - Support vectors in the edge: $\varepsilon_i=0$, $0<\alpha_i<C$ (eqs. 2, 5)
    - From eq. 1 we can use any of these margin points to solve for β_0
    - Typically use an average of all the solutions for numerical stability
  - The remainder support vectors: $\varepsilon_i>0$, $\alpha_i=C$ (eqs. 2, 5)
- Decision function: $G(x) = \text{sign}[x^T\beta + \beta_0]$

Support Vector Classifiers (vii)

- *C* is the tuning parameter
  - Controls the bias-variance trade-off
    - *C* large: lower number of support vectors (narrower margin)
      - Low bias, high variance
    - *C* small: higher number of support vectors (wider margin)
      - High bias, low variance
  - Choose the value of *C* via cross-validation
- **<u>Note</u>**: in James et al. the *C* parameter is not the standard one, but inversely proportional!!!

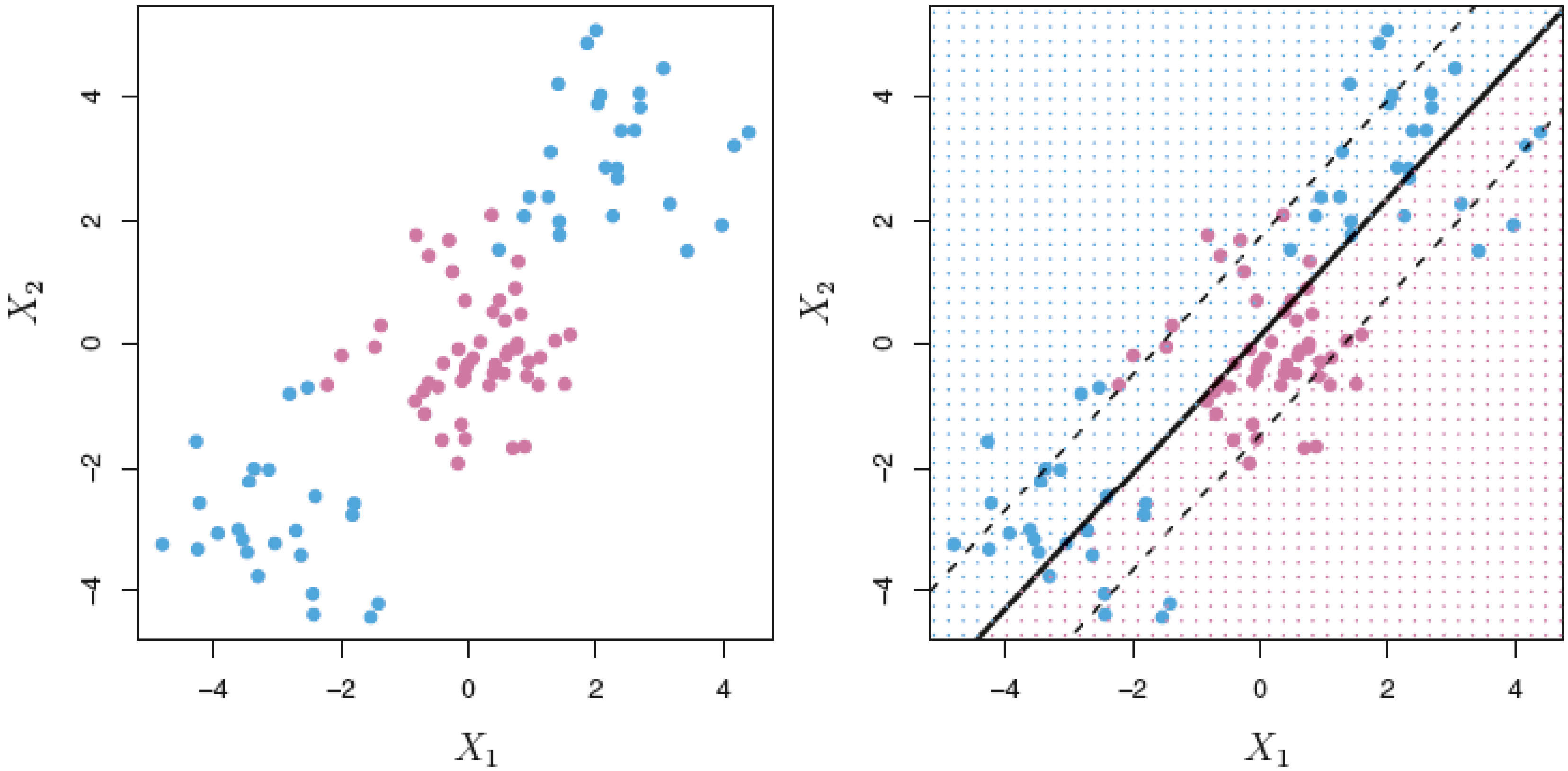
Small C



Large C

Support Vector Machines

- Non-linear class boundaries
  - Support vector classifier is useless
- Enlarge the feature space



© Manuel Mucientes

Support Vector Machines (ii)

- Feature space enlarged with functions of the predictors
  - Huge number of possible features
- SVM enlarge the feature space in a way that leads to efficient computations: kernels
- The solution to the support vector classifier problem involves only the inner products of the observations

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j} \qquad\qquad f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle y_i$$

- Transformed feature vectors $h(x)$:
  - Cheap computations of the inner products for particular choices of h

Support Vector Machines (iii)

- Solution function:
$$f(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i)\rangle + \beta_0$$
  - To evaluate *f*(x), compute the inner product of x with each support vector
- All we need are inner products
  - To represent the linear classifier *f*(x)
  - To compute its coefficients
- Need not to specify *h*(*x*), but the kernel function:
$$K(x, x') = \langle h(x), h(x')\rangle$$
  - The kernel measures the similarity between two observations
- Solution:
$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0$$

Support Vector Machines (iv)

- Kernel vs. enlarging the feature space using functions
  - Computational advantage: n(n-1)/2 inner products
  - Without explicitly working in the enlarged feature space
    - we do not care about those functions
- Linear kernel: SVC

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

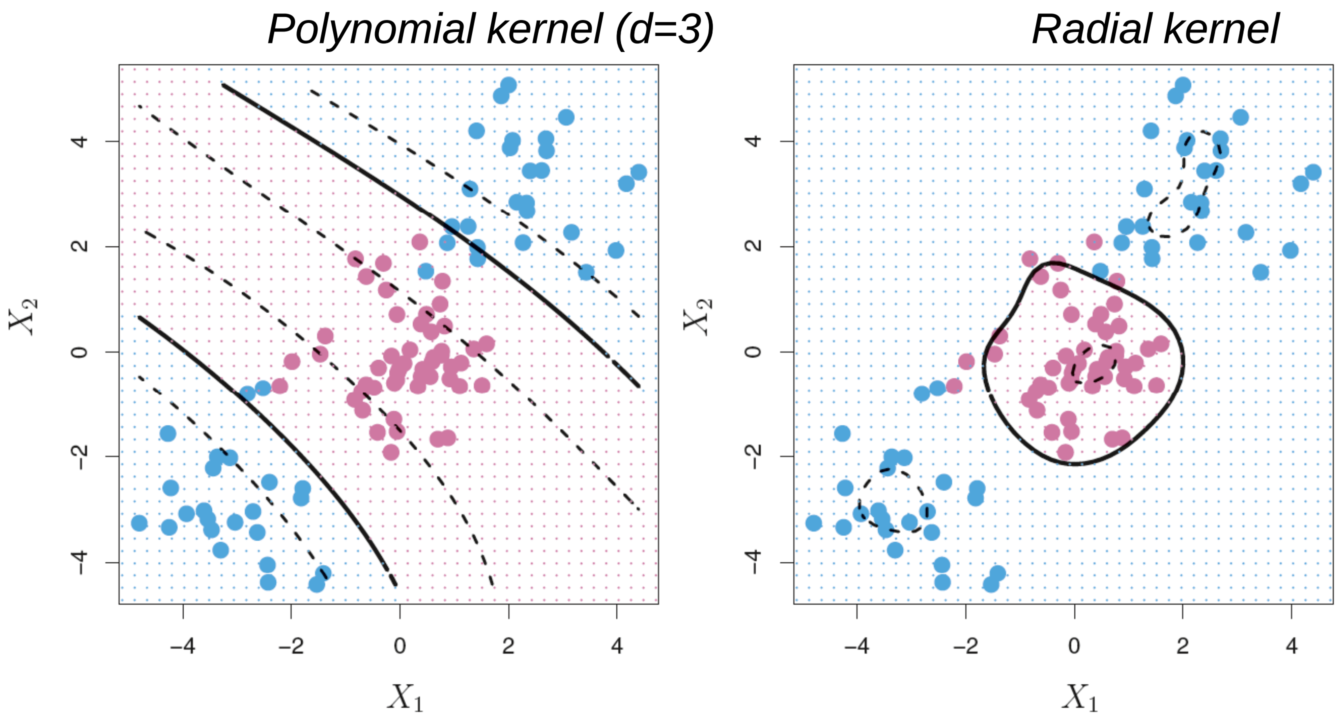
- Combination of a support vector classifier with a non-linear kernel: SVM
- Polynomial kernel of degree *d*:

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$

  - If *d*>1: non-linear decision boundary with degree *d* polynomials in a higher dimensional space

Support Vector Machines (v)

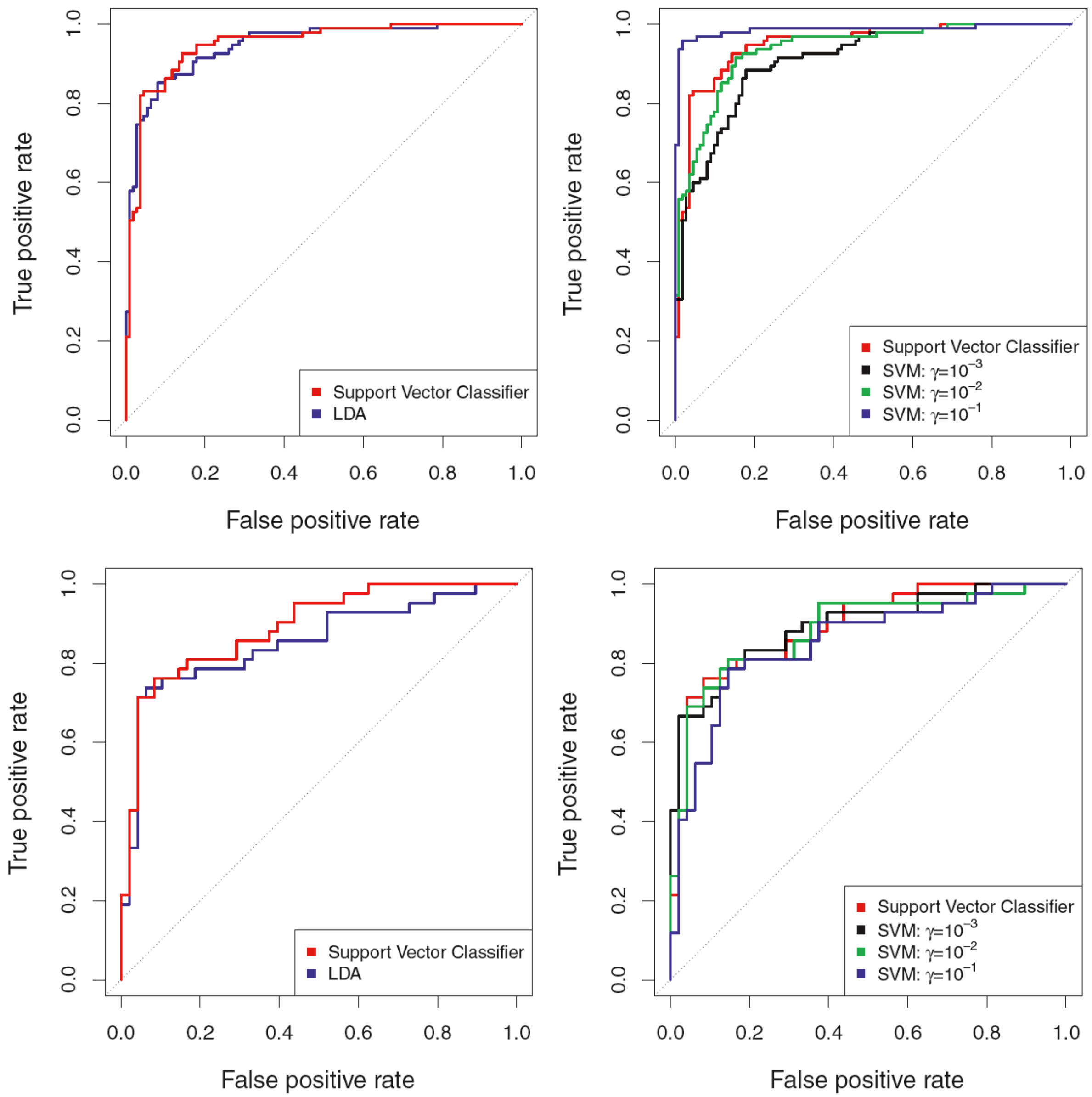
- Radial kernel: $$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$
  - Training observations far from x (test observation) play no role in the predicted class label
  - Very local behavior



© Manuel Mucientes

Example: Heart dataset

- Training:
  - 207 observations
  - High γ: more non-linear
  - Best: SVM-$\gamma=10^{-1}$
- Test:
  - 90 observations
  - Best: SVC, SVM-$\gamma=10^{-2}$, SVM-$\gamma=10^{-3}$
- The best type of kernel depends on the problem



SVMs with more than Two Classes

- K classes
- One vs. One
  - Learn K(K-1)/2 (all the pairs) of classifiers
  - Each classifier compares the k-th class (coded +1) with the k'-th class (coded -1)
  - Test:
    - Count the number of times that the observation is assigned to each of the K classes
    - Assign the class most frequently selected
- One vs. All
  - Learn K classifiers: k-th is coded +1, and the remaining K-1 classes are coded -1
  - Test: assign the observation to the class with largest $f(x)$ (highest level of confidence)

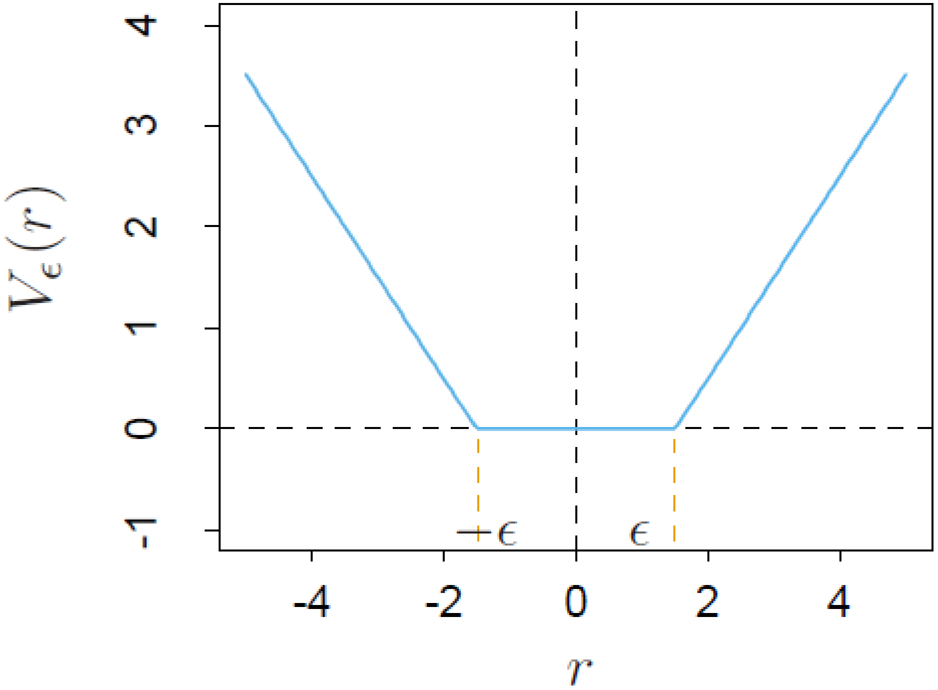
Support Vector Regression

- SVMs for regression
- To learn the parameters of f(x), minimize:

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2}\|\beta\|^2$$

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise} \end{cases}$$

  - λ: regularization parameter
    - Estimate by cross-validation
- SVR not as good for regression as SVMs for classification



© Manuel Mucientes

Bibliography

- G. James, D. Witten, T. Hastie, y R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer, 2013.
  - Chapter 9
- T. Hastie, R. Tibshirani, y J. Friedman, The elements of statistical learning. Springer, 2009.
  - Chapter 12