Recap 12

Support Vector Machines

ISLR 9

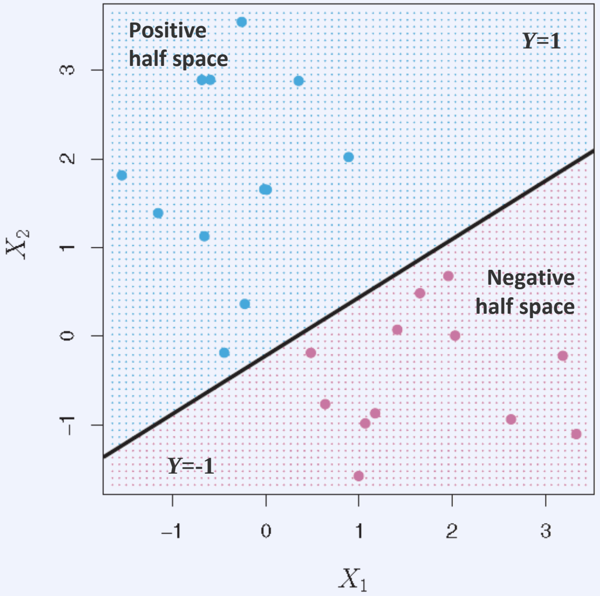
Jilles Vreeken
Krikamol Muandet

UNIVERSITÄT DES SAARLANDES
CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

The Maximal Margin Classifier

- a hyperplane that maximizes the **distance of the closest point** in the training set to it can be considered optimal

separating hyperplane and resulting classifier



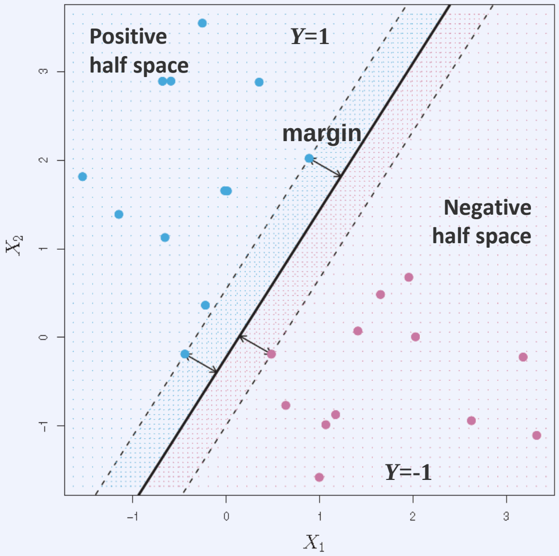
The Maximal Margin Classifier

- a hyperplane that maximizes the **distance of the closest point** in the training set to it can be considered optimal
- this distance is called the **margin**

The closest data points are called the **support vectors**

- only they determine the hyperplane
- can be a small subset of all points

separating hyperplane and resulting classifier



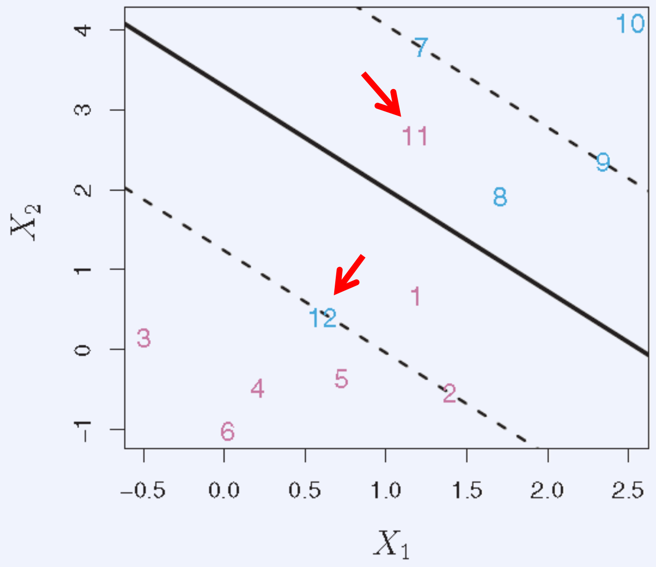
The Support Vector Classifier

Even if the dataset is separable, a separating hyperplane may not be desirable

Sometimes it may be preferable to have a classifier that misplaces a few points in the training set but has a large margin to the other data points

- the soft-margin classifier does exactly this

soft-margin classifier



points can be on the wrong side of the margin (misplaced but correct) or the hyperplane (misclassified)

Details of Soft-Margin Support Vector Classifier

The optimization problem is now

$$\max_{\beta_0,\beta_1,\ldots\beta_p,M} M$$

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

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

$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$

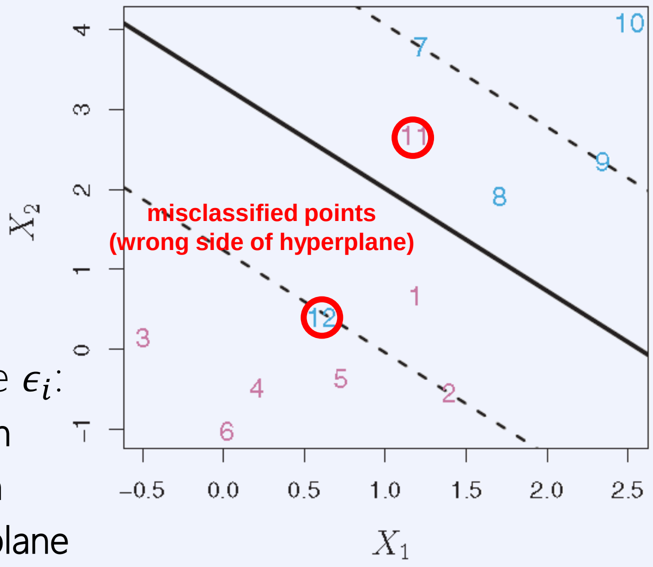
Slack variables allow for a fractional violation of the hard margin constraint

Budget for total admissible misclassification

The following holds if we also choose the smallest possible ϵ_i:

- $\epsilon_i = 0 \Rightarrow$ the observation is on the **correct** side of the **margin**
- $\epsilon_i > 0 \Rightarrow$ the observation is on the **wrong** side of the **margin**
- $\epsilon_i > 1 \Rightarrow$ the observation is on the **wrong** side of the **hyperplane**

soft-margin classifier



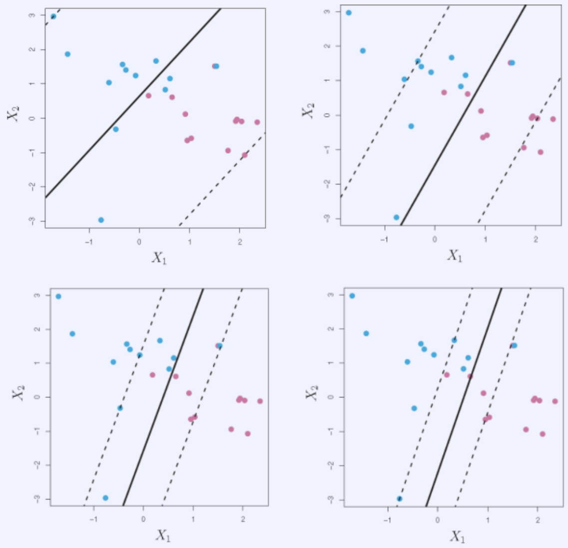
The Margin and the Support Vectors

We choose C via cross-validation

As C increases, we become more tolerant to violations

The fact that correctly classified points far away from the hyperplane do not affect the classifier is a property of the support-vector classifier

as C decreases we become less tolerant to violations



Nonlinear Decision Boundaries

Sometimes, data is inherently nonlinear

- then there is no soft margin that will do the trick
- we need a nonlinear version of support vector machines
- we could add nonlinear features to the feature space, e.g. $X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2$ instead of $X_1, X_2, \ldots, X_p$
- the resulting optimization program would become

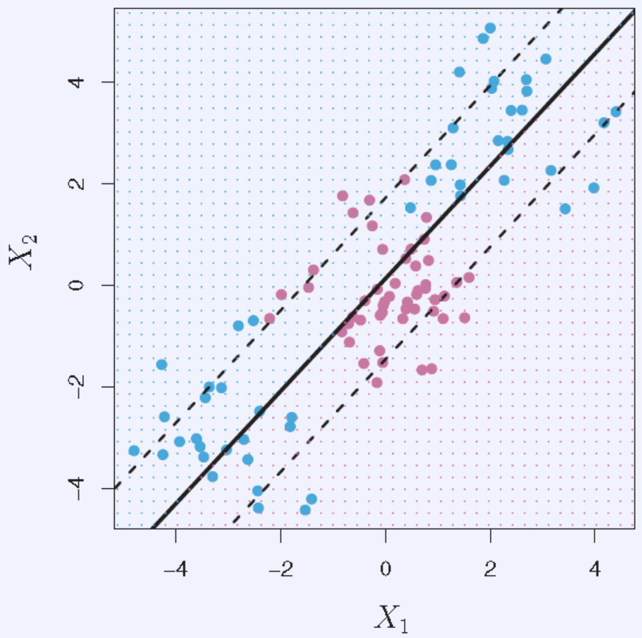
$$\max_{\beta_0, \beta_{11}, \beta_{12}, \ldots \beta_{p1}, \beta_{p2}, \epsilon_1, \ldots, \epsilon_n, M} M$$

$$\text{subject to } \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C, \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1$$

$$y_i \left(\beta_0 + \sum_j^p \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i),\ i = 1, \ldots n$$

- we could add higher-order, interaction terms, or use functions other than polynomials

the true boundary is non-linear



The Kernel Trick

With **support vectors machines (SVMs)** there is a different very elegant trick – the **kernel trick**

- builds on the optimization procedure for SVMs, which we will not detail
- it suffices to say that the linear support vector classifier can be rewritten as $f(x^*) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x^*, x_i \rangle$
- $\langle x^*, x_i \rangle = \sum_{j=1}^{p} x_j^* x_{ij}$ is the inner product,
- and the α_i are parameters that result from the training

set of support vectors

Important: Only the α_i for the support vectors are nonzero $f(x^*) = \beta_0 + \sum_{i \in S} \alpha_i \langle x^*, x_i \rangle$

Advantages of Kernels

To calculate the SVM you only need the kernel matrix for the pairs of training points

- in contrast, enlarging the feature space is computationally expensive

Can be applied to arbitrary observations that are not vectors: graphs, strings, molecules, etc.

The kernel trick can also be used with other statistical learning methods such as LDA or PCA

Summary

The main ideas behind SVMs is to find the max-margin hyperplane that separate the data

Hard SVM requires that all training data is correctly separated by can overfit

Soft SVM allows violations of the margin up to a budget C to get a better hyperplane overall

We can rewrite the SVM classifier only in terms of inner products – replacing those with a kernel is the kernel trick which allow us to efficiently introduce non-linearity

- the kernel trick is an important general idea that also applies to LDA, PCA and other models

Linear SVM is similar to logistic ridge regression but uses a hinge loss instead