Support Vector Machines

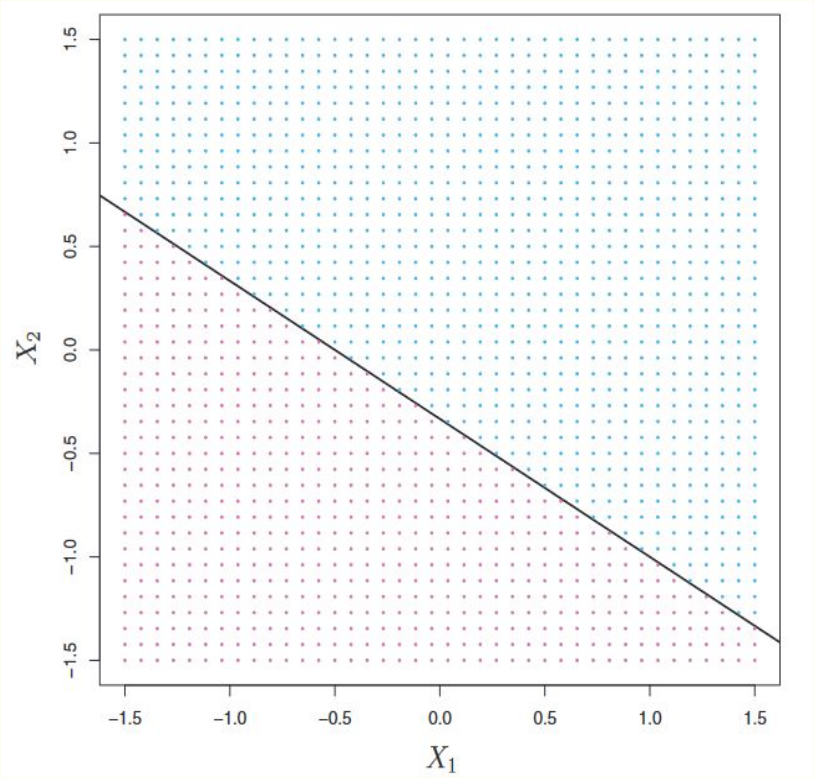
Abhinav Mishra, Kristin Köhler, Sanket Gosavi, Utkarsha Kandale
Group 4
Data Science in the Life Sciences

Table of Contents

1. Introduction: Hyperplane
2. Maximal Margin Classifier
3. Support Vector Classifier
4. Support Vector Machine
5. SVM (with more than two classes)
6. SVM vs Logistic Regression
7. Example in R

Hyperplanes

- Hyperplanes are a way to divide (classify) your data.
- For p-dimensional data the hyperplane will be of dimension p-1.



Mathematical definition

- For a two dimensional data the hyperplane would be defined as:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

- For n-dimensional data:

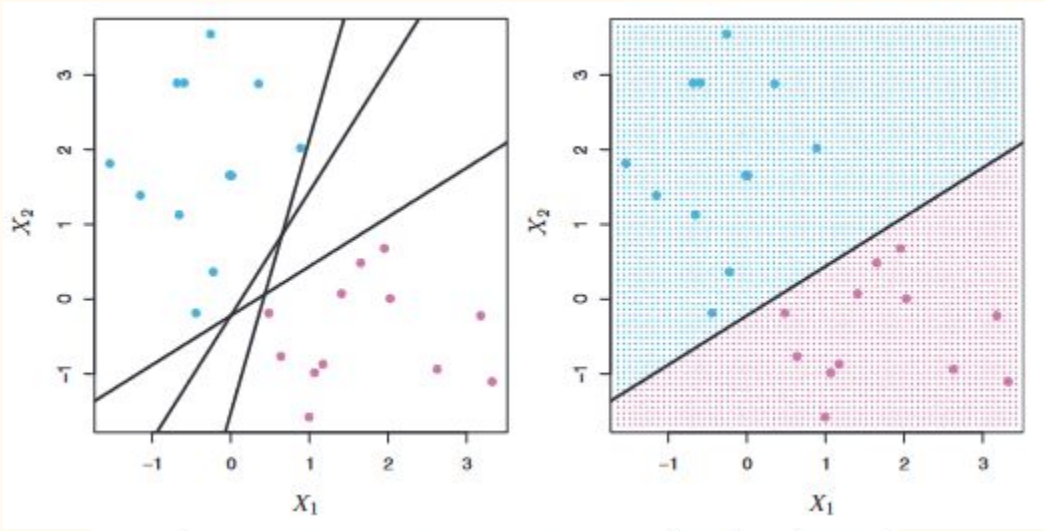
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

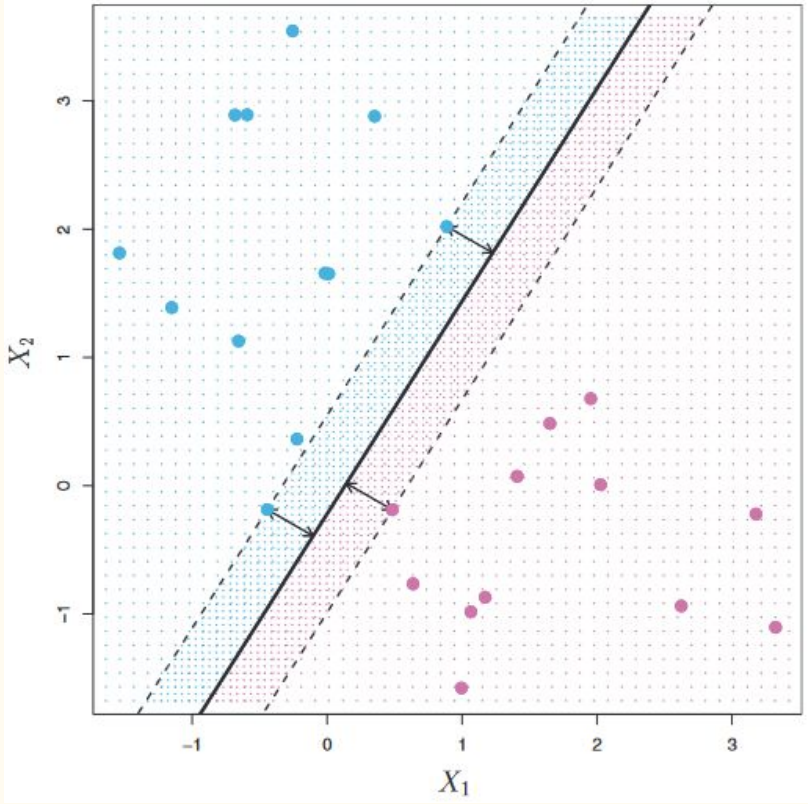
- For classification:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0. \qquad \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0,$$

Maximal Margin Classifier

- There are potentially infinite number of ways in which you can divide the data using a hyperplane.
- One way to find a good hyperplane is to use a Maximal Margin Classifier.
- The Maximal Margin Classifier is composed of three components:
  - Dividing hyperplane
  - Supporting vectors
  - Margins





Construction of the Maximal Margin Classifier

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

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

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

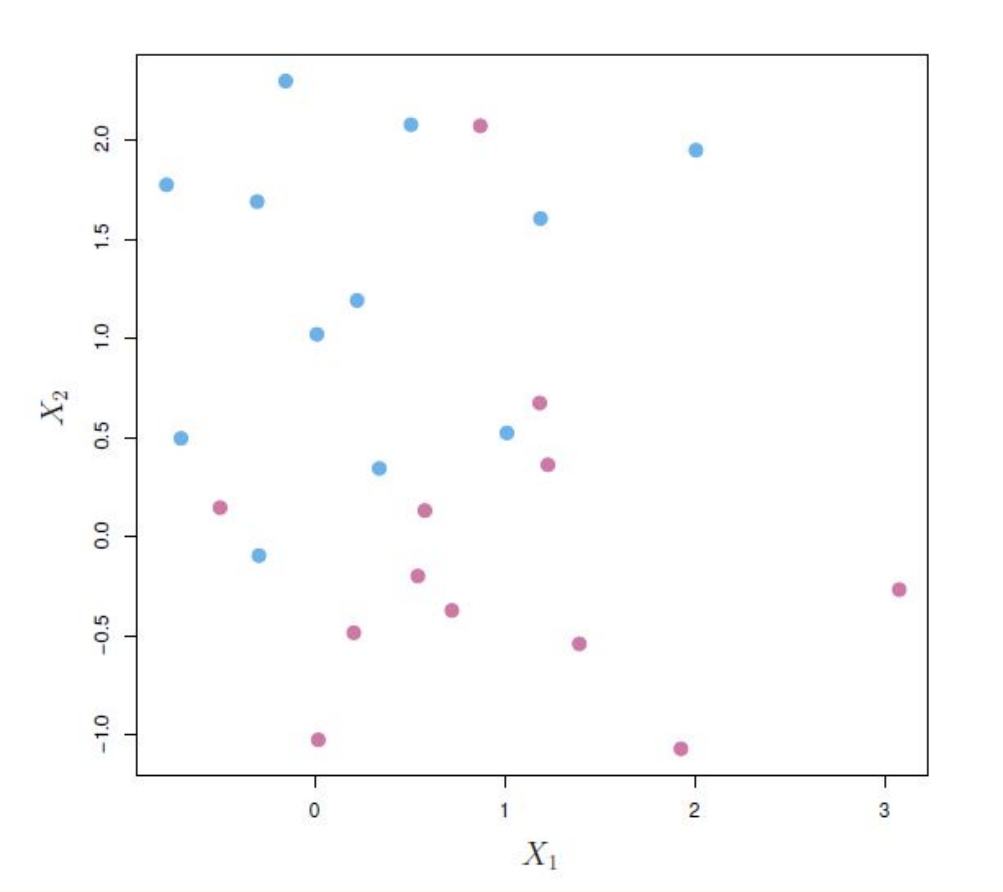
- Equation three ensures that each observation is classified correctly for all the observations provided that M is positive.
- The second equation allows one to calculate the perpendicular distance of an observation from the hyperplane using the equation:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}).$$

- Together equations two and three ensure that each observation is on the correct side of the hyperplane and at least a distance M from the hyperplane.
- M represents the margin of our hyperplane, and the optimization problem chooses $\beta_0, \beta_1, \ldots, \beta_p$ to maximize M

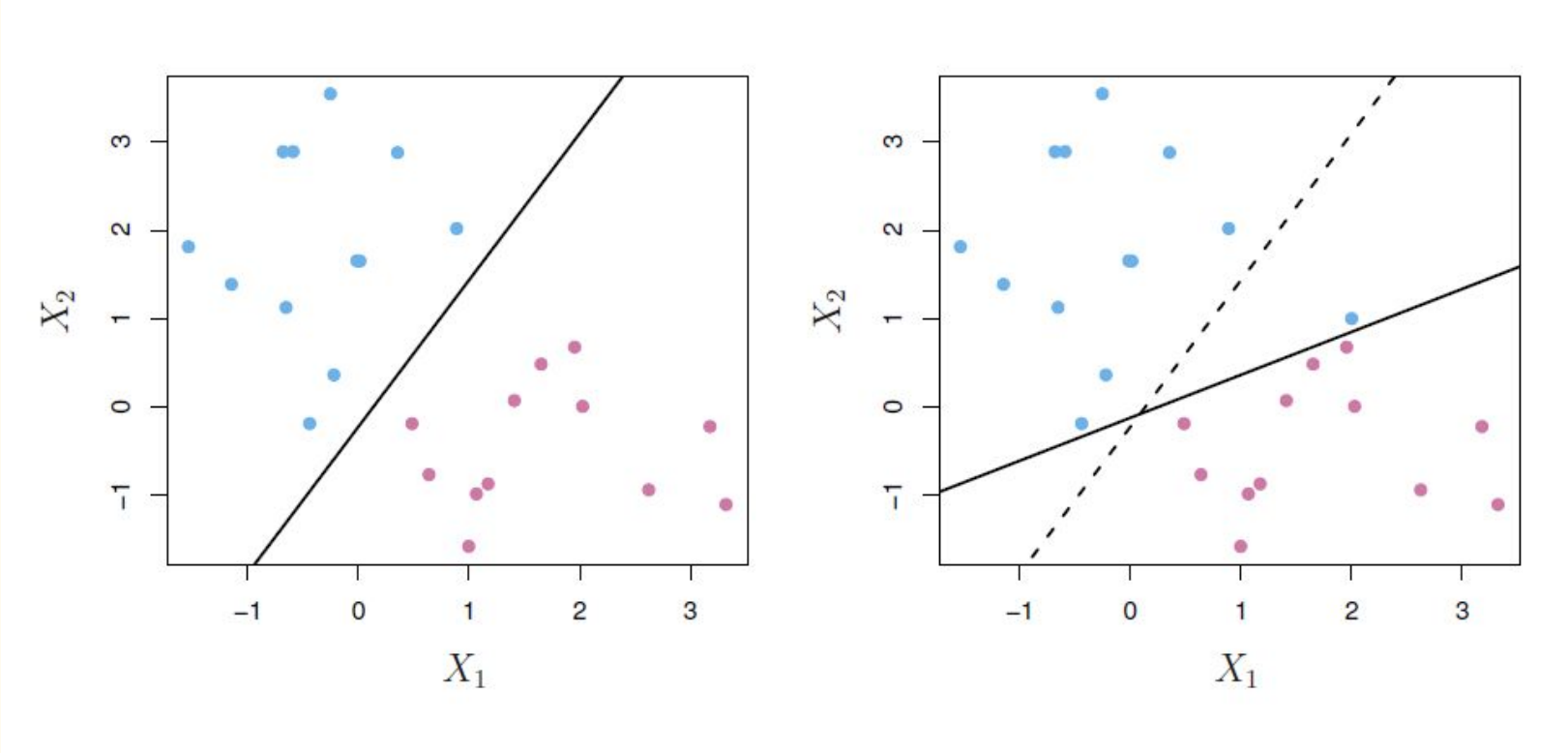
Support Vector Classifier

Problem 1: **Non-separable data**



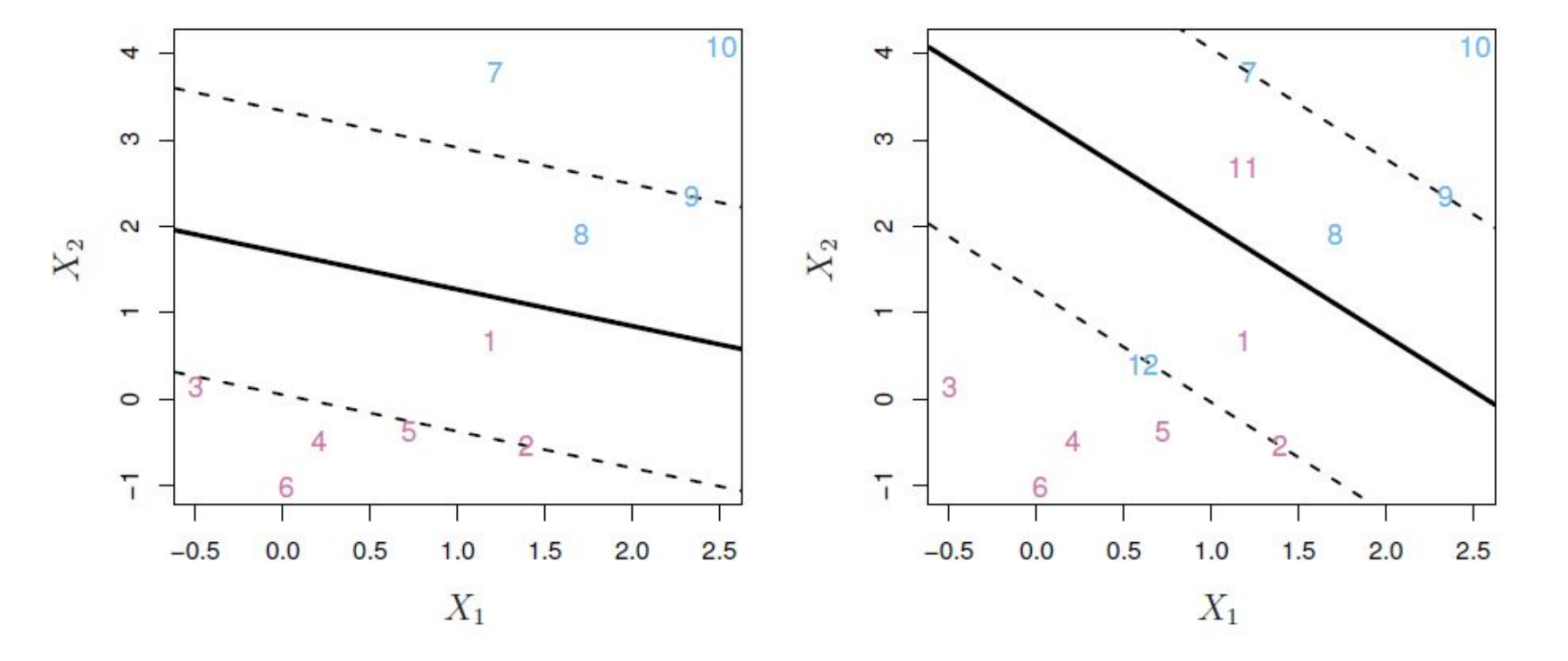
Support Vector Classifier

Problem 2: **Noisy data > overfitting to training data**



Support Vector Classifier

The support vector classifier *maximizes a soft margin*.



Support Vector Classifier

The support vector classifier *maximizes a soft margin*.

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

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

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

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

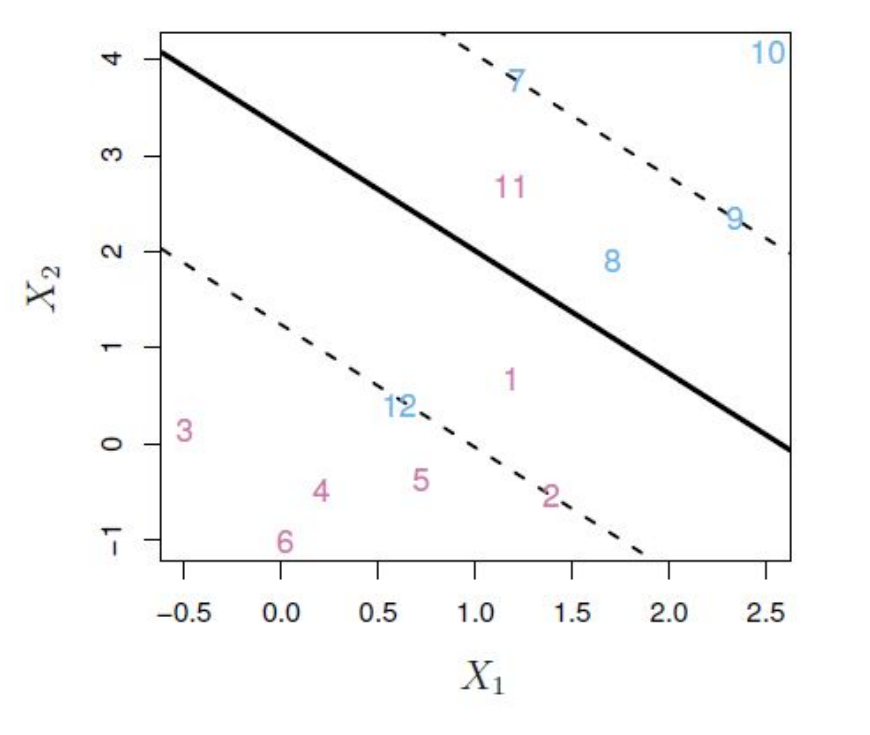
ε- slack variable allowing misclassifications
C- tuning parameter

Support Vector Classifier

$\varepsilon = 0$: ith observation on the correct side of the margin

$\varepsilon > 0$: ith observation on the wrong side of the margin

$\varepsilon > 1$: ith observation on the wrong side of the hyperplane



Support Vector Classifier

$\varepsilon = 0$: ith observation on the correct side of the margin

$\varepsilon > 0$: ith observation on the wrong side of the margin

$\varepsilon > 1$: ith observation on the wrong side of the hyperplane

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

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

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

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

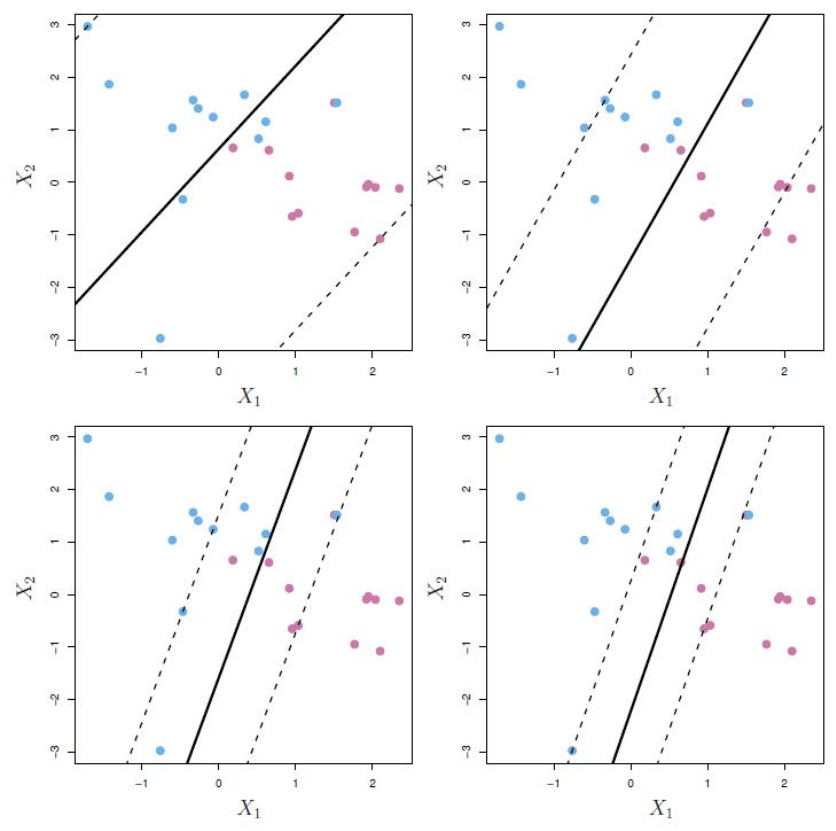
Support Vector Classifier

large C: wider margin, more misclassifications allowed

> *high bias-low variance*

small C: small margin, less misclassifications allowed

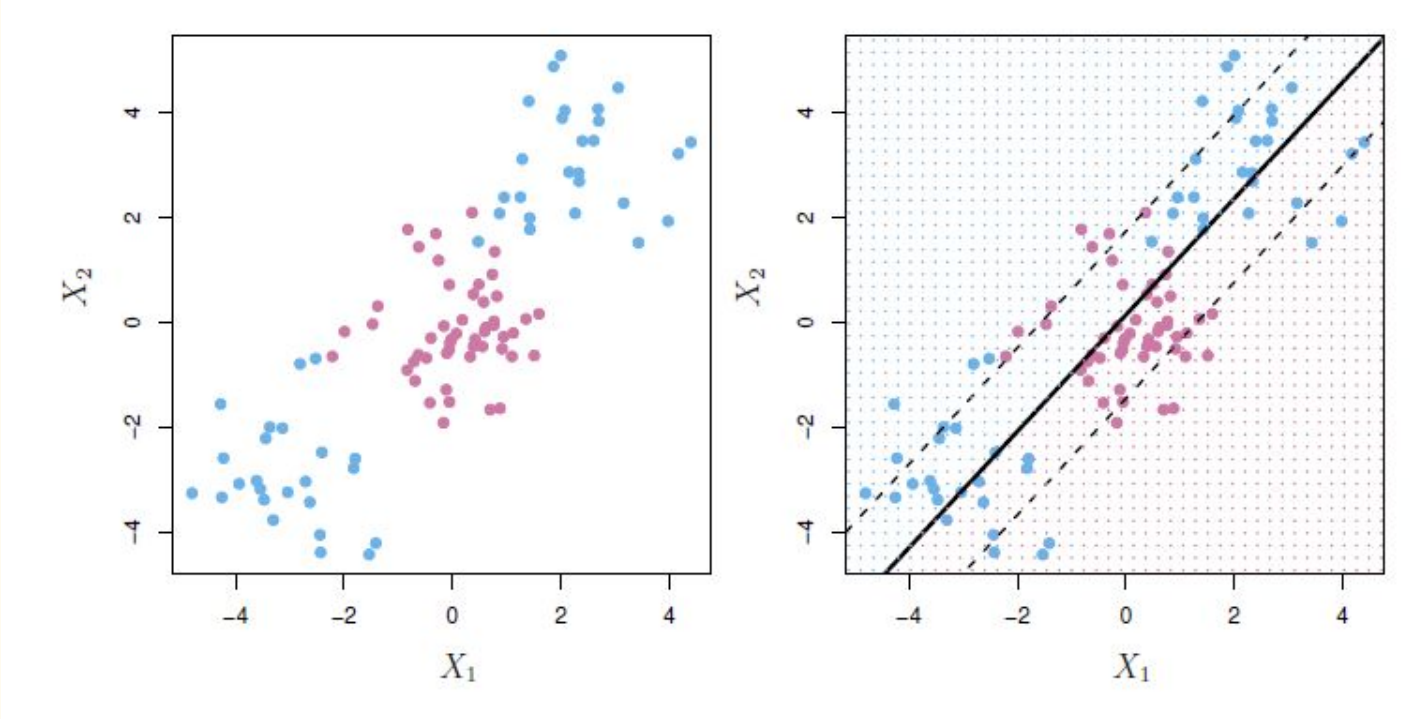
> *low bias-high variance*



only observations that either lie on the margin or that violate the margin will affect the hyperplane => *support vectors*

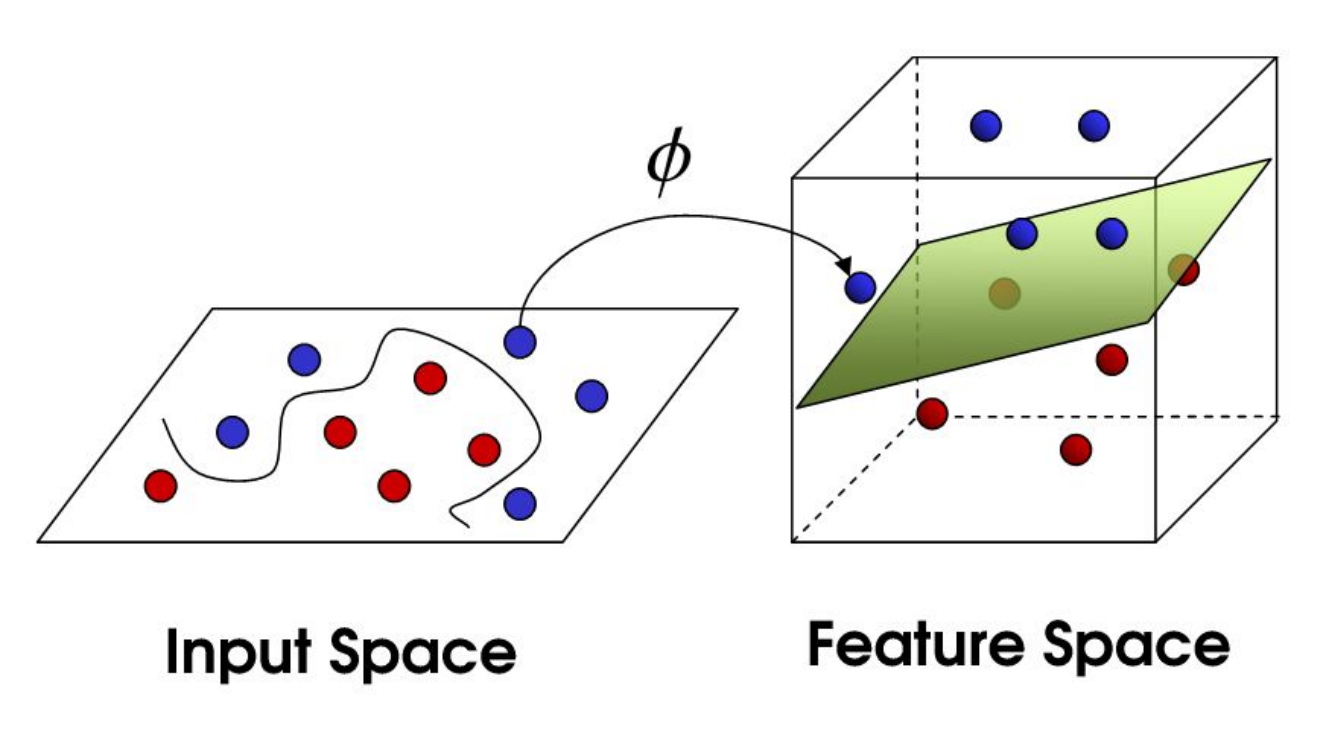
Support Vector Machines

Problem: **Non-linear decision boundaries**



Support Vector Machines

Idea: enlarge the feature space to get a separating hyperplane in a higher dimension



Support Vector Machines

The linear support vector classifier can be represented as:

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

To estimate the parameters $\alpha_1, \ldots, \alpha_n$ and β_0, all we need are $\binom{n}{2}$ inner products $\langle x_i, x_{i'} \rangle$ between all pairs of training observations. α_i is nonzero only for the support vectors in the solution

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle,$$ with S as indices of support points

Support Vector Machines

Replace the inner product with a generalization of the form $K(x,x_i)$ called kernel function

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i).$$

Linear kernel

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

Polynomial kernel of degree 'd'

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

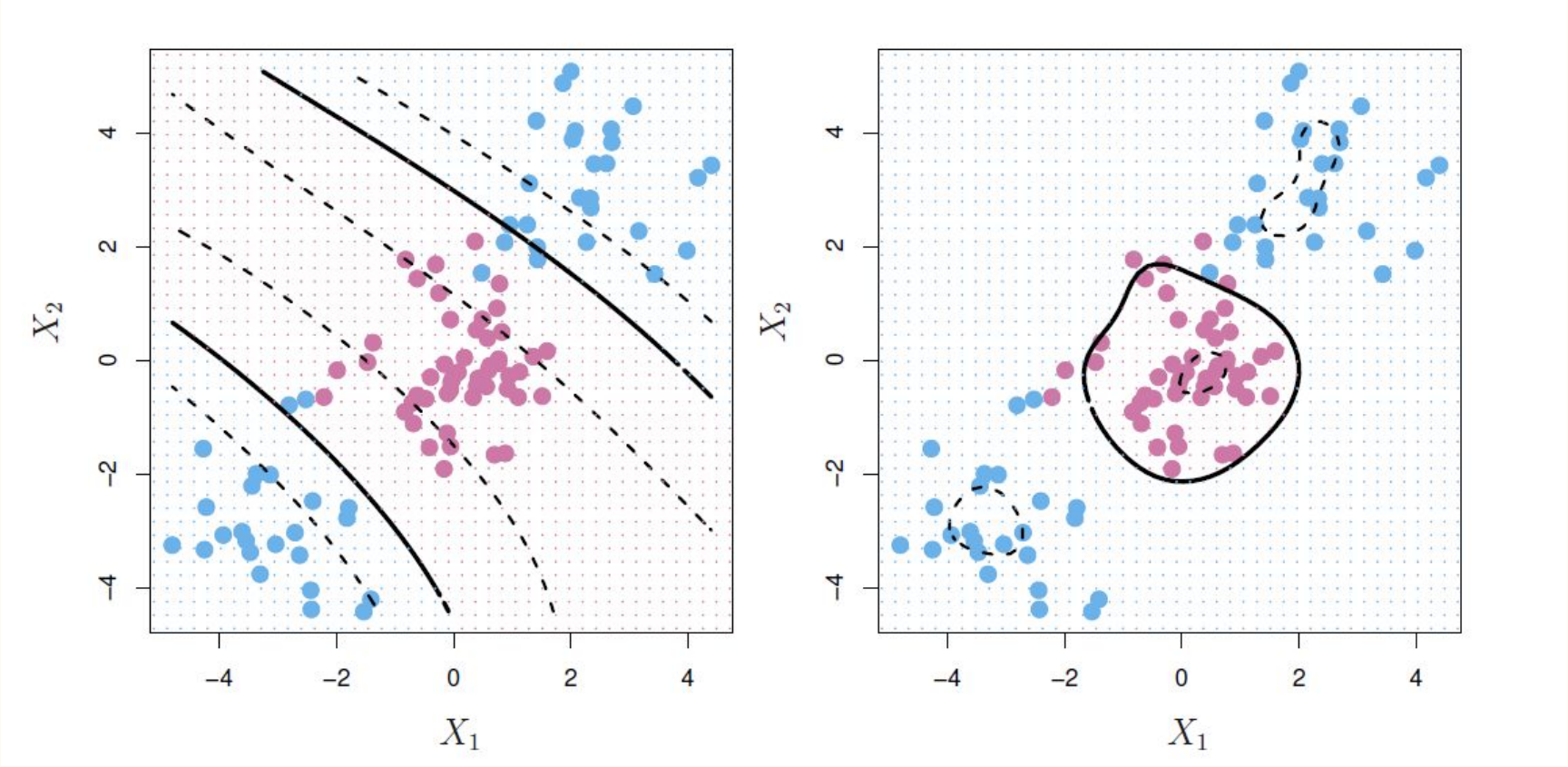
Radial kernel

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2).$$

Support Vector Machines

Polynomial kernel (degree = 3)

Radial kernel



SVM (> 2 Classes)

As defined before, it works for K > 2 classes.

OVA One versus All. Fit K different 2-class SVM classifiers $\widehat{f}_k(x)$, k = 1, . . . , K ; each class versus the rest. Classify x^* to the class for which $\widehat{f}_k(x^*)$ is largest.

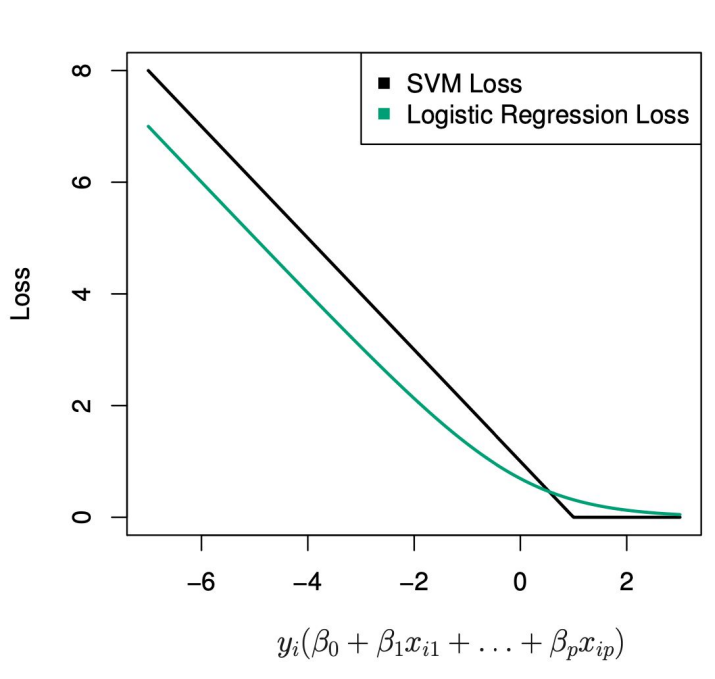
OVO One versus One. Fit all $\binom{K}{2}$ pairwise classifiers $\widehat{f}_{kl}(x)$. Classify x^* to the class that wins the most pairwise competitions.

Which to choose? If K is not too large, use OVO.

Support Vector versus Logistic Regression ?

With $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$, can be written as:

$$\underset{\beta_0, \beta_1, \ldots, \beta_p}{minimize} \left\{ \sum_{i=1}^{n} max\left[0, 1 - y_i f(x_i)\right] + \lambda \sum_{j=1}^{p} \beta^2{}_j \right.$$



This has the form *loss plus penalty*.

The loss is known as the *hinge loss*.

Very similar to "loss" in logistic regression

(negative log-likelihood).

Which to use: SVM or Logistic Regression ?

- When classes are (nearly) separable, SVM does better than LR. So does LDA.
- When not, LR (with ridge penalty) and SVM very similar.
- If you wish to estimate probabilities, LR is the choice.
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.

Example in R

The e1071 library contains implementations for a number of statistical learning methods. In particular, the svm() function can be used to fit a support vector classifier when the argument kernel = "linear" is used.

```
> set.seed (1)
> x <- matrix(rnorm (20 * 2), ncol = 2)
> y <- c(rep(-1, 10), rep(1, 10))
> x[y == 1, ] <- x[y == 1, ] + 1
> plot(x, col = (3 - y))
```

Next, we fit the support vector classifier.

```
> dat <- data.frame(x = x, y = as.factor(y))
> library(e1071)
> svmfit <- svm(y ~ ., data = dat , kernel = "linear", cost = 10,
scale = FALSE)
```

We can now plot the support vector classifier obtained

```
> plot(svmfit , dat)
```

The support vectors are plotted as crosses and the remaining observations are plotted as circles; we see here that there are seven support vectors. We can determine their identities as follows:

```
> svmfit$index


[1] 1 2 5 7 14 16 17
```

We can obtain some basic information about the support vector classifier fit using the summary() command:

```
> summary(svmfit)
Call:
svm(formula = y ~ ., data = dat , kernel = "linear", cost = 10,
scale = FALSE)

Parameters:
        SVM -Type: C- classification
      SVM -Kernel: linear
                  cost: 10
Number of Support Vectors: 7
      ( 4 3 )
Number of Classes: 2
Levels:
   -1 1
```

References

- James, Gareth, Daniela Witten, Trevor Hastie, und Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Second edition. Springer Texts in Statistics. New York NY: Springer, 2021. https://doi.org/10.1007/978-1-0716-1418-1.
- Noel Bambrick, *SVM: Feature Selection and Kernels,* https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c,

Questions?