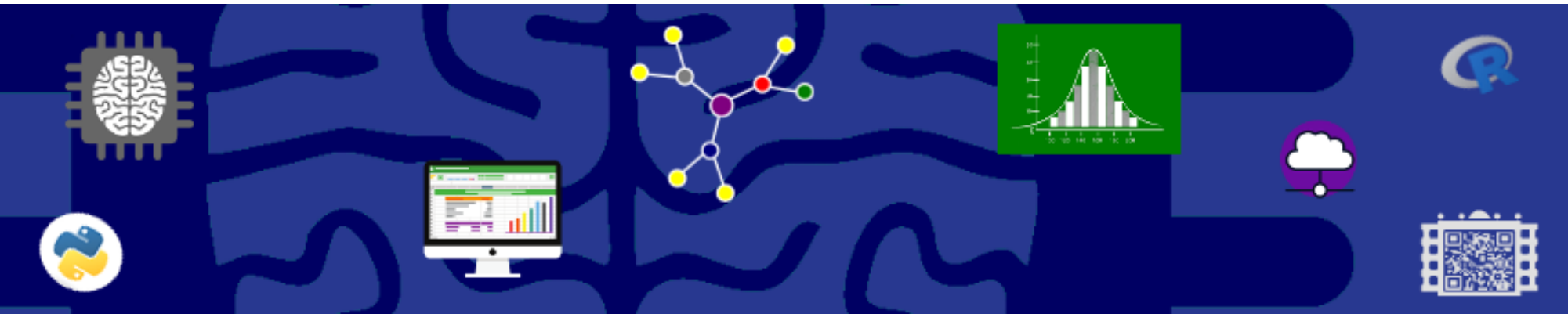
Introduction to Data Science

Winter Semester 2018/19

Oliver Ernst

TU Chemnitz, Fakultät für Mathematik, Professur Numerische Mathematik

Lecture Slides



Contents I

Contents II

Contents III

Contents

Contents

Support Vector Machines

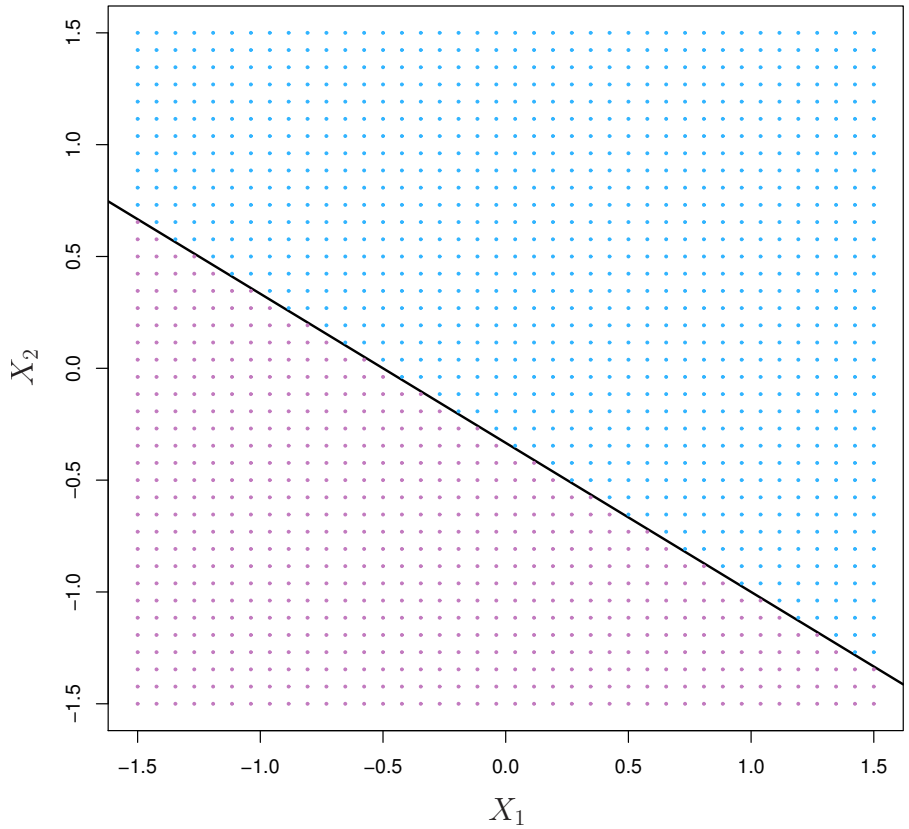
Hyperplanes for classification

- Assume a (true) bivariate classification model has a linear decision boundary.
- The points **on** the decision boundary are characterized by an equation of the form

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

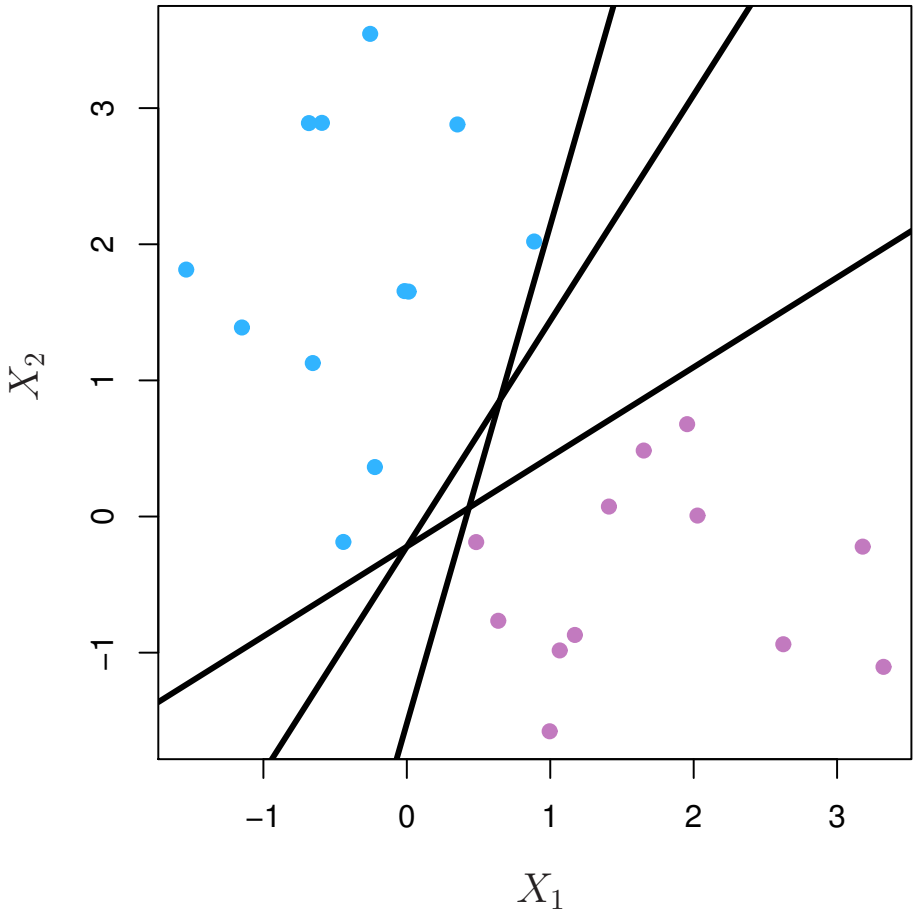
- The points on either side are characterized by

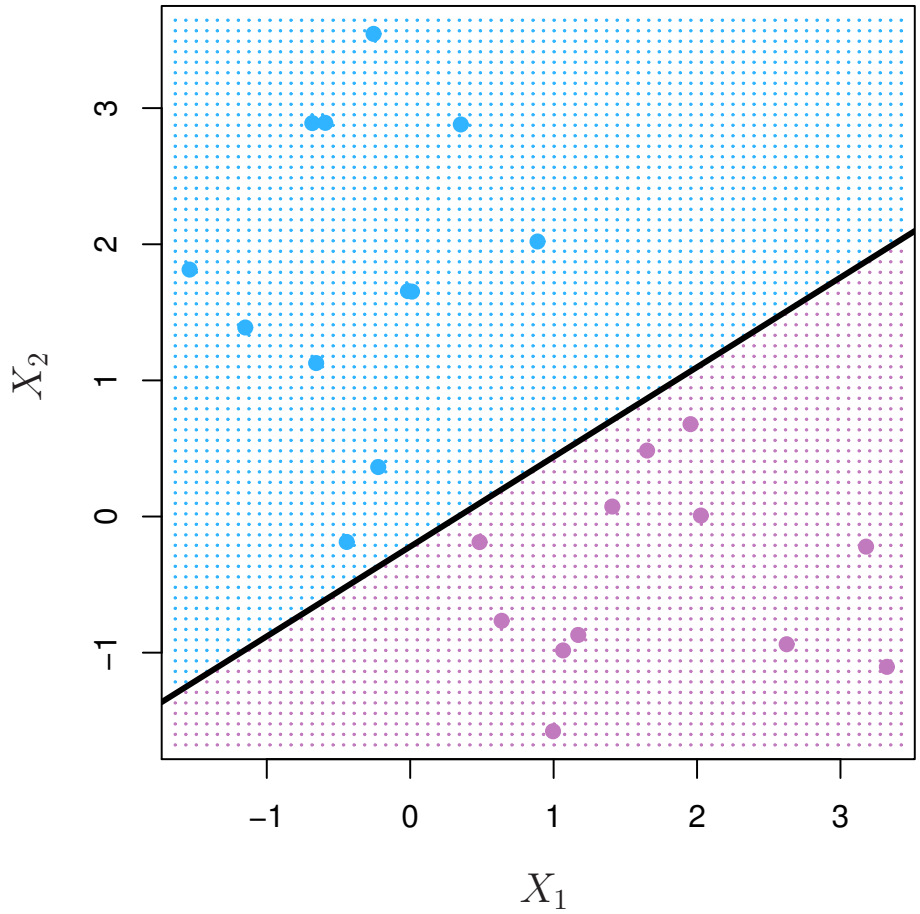
$$\beta_0+\beta_1 X_1+\beta_2 X_2 \begin{cases} > 0 & \text{one side,} \\ < 0 & \text{other side.} \end{cases}$$



Support Vector Machines

Hyperplanes for classification





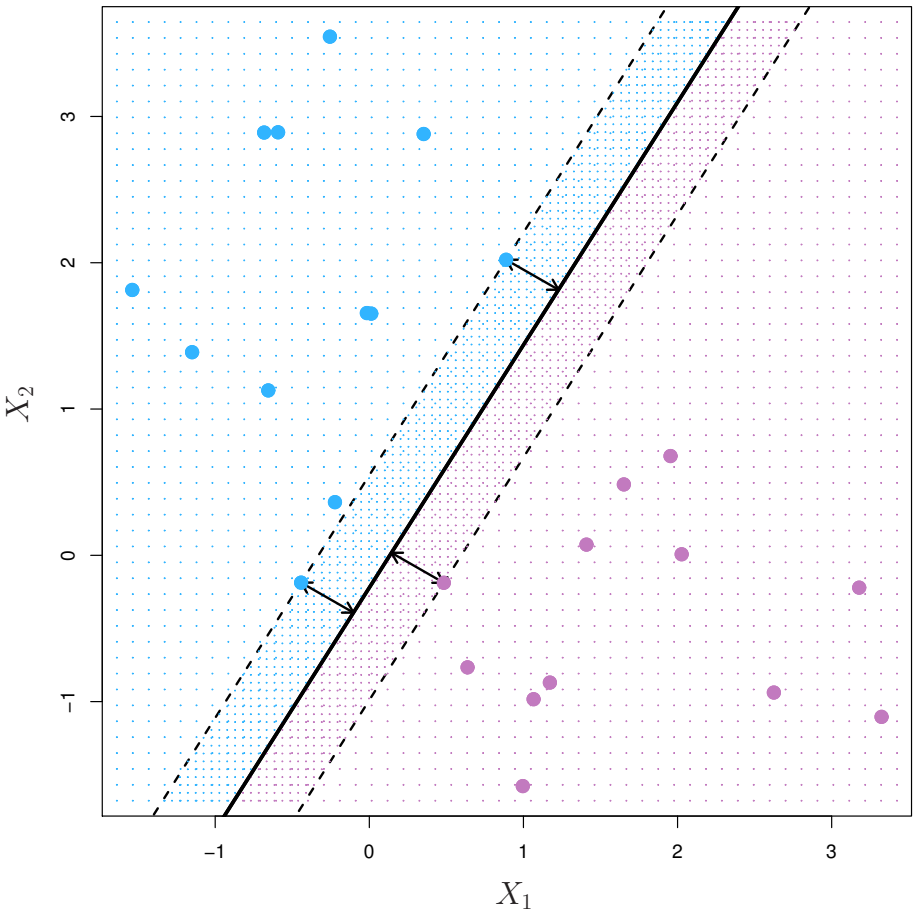
Left: Given a 2D data set of binary observations that can be split by a line, the line is in general not unique.

Right: Once a line has been fixed, this defines an associated classification model.

Support Vector Machines

Maximal margin classifier

- Compare with: KNN (Chapter 2), logistic regression, LDA, QDA (Chapter 4), decision trees (Chapter 8).
- Given binary observations, which linear decision boundary to choose?
- Separating line farthest from all training observations: **maximal margin hyperplane** or **optimal separating hyperplane**.
- Minimal distance from this line to closest observation: **margin**.



- Margin: half-width of largest slab separating the two observation set classes.
- Associated classification model: **maximal margin classifier**.

Support Vector Machines

Maximal margin classifier

- Large margin instills confidence in the classifier.
- In previous example: 3 points at marginal distance to separating hyperplane (there will always be at least 2, why?) These observations are called the **support vectors** of the maximal margin hyperplane, as moving these would move the latter.
- "Although the maximal margin classifier is often successful, it can also lead to overfitting when p is large."

- **Given:** training set $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^n$, $\boldsymbol{x}_j \in \mathbb{R}^p$, $y_j \in \{-1, 1\}$.
- **Goal:** determine coefficients $\beta_0, \beta_1, \ldots, \beta_p$ such that

$$\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} \begin{cases} > 0 & \text{if } y_i = 1, \\ < 0 & \text{if } y_i = -1, \end{cases} \qquad i = 1, \ldots, n,$$

  or, equivalently, $y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) > 0$, $i = 1, \ldots, n$, such that the margin between the resulting hyperplane and the data set is maximal.
- Formally:

  maximize M as a function of $\beta_0, \beta_1, \ldots, \beta_p, M$ such that

$$\sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.1a}$$

$$y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq M, \quad i = 1, \ldots, n. \tag{9.1b}$$

Support Vector Machines

Constructing the maximal margin classifier

- (9.1) is a constrained optimization problem.
- Constraint (9.1b) ensures all observations on correct side of separating hyperplane if $M > 0$.
- Constraint (9.1a) not strictly necessary, merely scales M to coincide with margin.
- For details on solving the optimization problem see the reading list or a class on constrained optimization.
- Optimal separating hyperplane produces function $\hat{f}(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}, \boldsymbol{\beta} := (\beta_1, \ldots, \beta_p)^\top$ from which the classifier is derived as

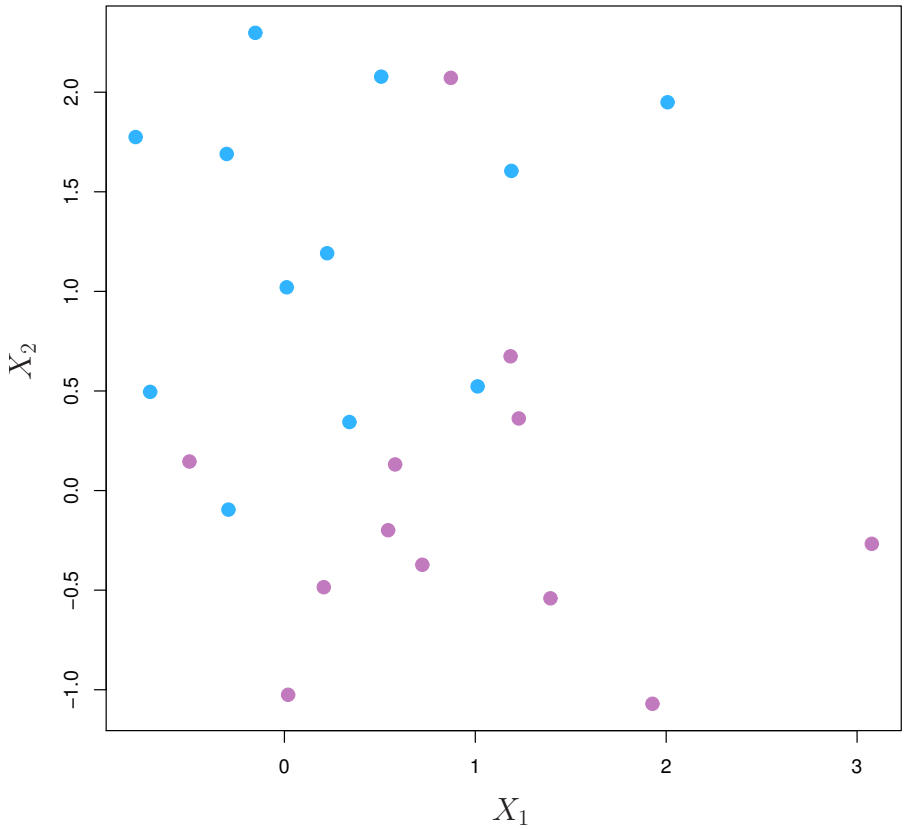
$$\hat{F}(\boldsymbol{x}) := \operatorname{sign} \hat{f}(\boldsymbol{x}).$$

- All training observations outside margin, not necessarily so for test observations. Intuition: large margin on training data will result in good separation of test data.

Support Vector Machines

Non-separable case

- A maximal margin classifier (or even a separating hyperplane) need not exist for a given data set.
- In this case optimization problem (9.1) has no solution with $M > 0$.
- Will extend concept of separating hyperplane using so-called **soft margin** to almost separate the classes.
- This leads to **support vector classifier**.

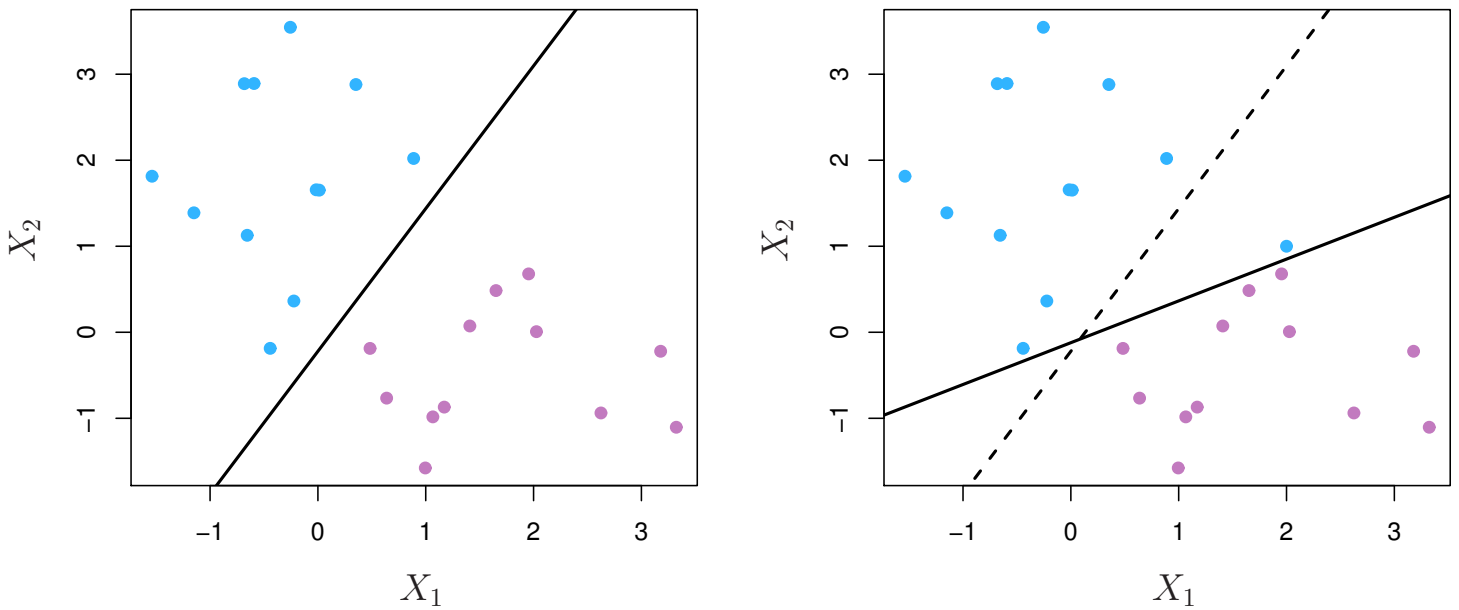


Contents

Support Vector Classifiers

Shortcomings of exact hyperplane classifiers

- Even if separating hyperplane exists, associated classifier possibly undesirable.
- Example: single additional blue point dramatically changes maximal margin hyperplane, leading to tiny margin.

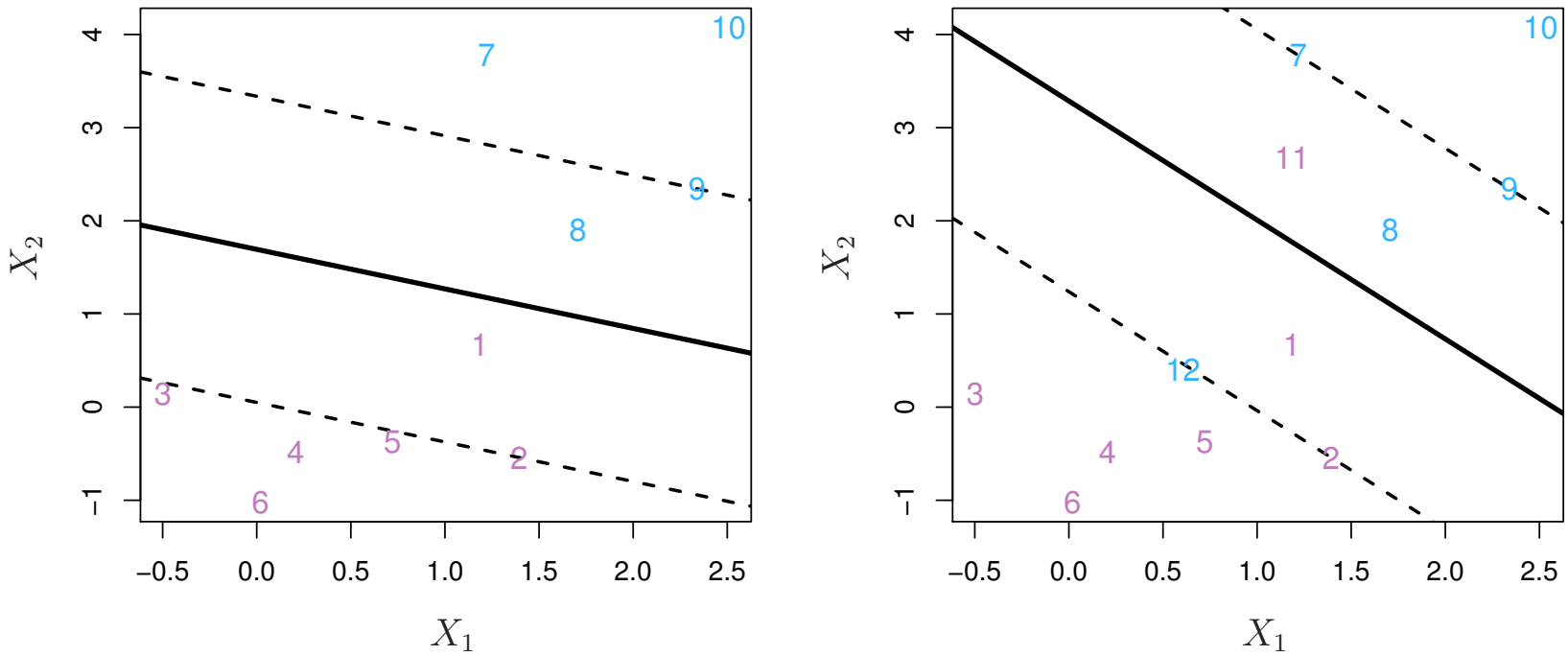


- Reasonable to accept hyperplane classifier which does **not** perfectly separate training data in the interest of greater robustness to individual observations and better classification of **most** training observations.

Support Vector Classifiers

Support vector classifiers

- **Support vector classifier** (**soft margin classifier**): allow some observations to be on wrong side of margin or hyperplane. (Soft margin: can be violated by some training observations).
- Example: hyperplane (solid), margins (dashed) and observations.



Observations 11 (violet) and 12 (blue) are added in right panel.

- Support vector classifier solves following optimization problem:

$$\begin{aligned}
&\text{maximize } M \text{ as a function of } \beta_0, \beta_1, \ldots, \beta_p, M, \epsilon_1, \ldots, \epsilon_n \\
&\text{subject to} \\
&\sum_{j=1}^{p} \beta_j^2 = 1, && (9.2a) \\
&y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq M(1 - \epsilon_i), && (9.2b) \\
&\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \qquad i = 1, \ldots, n, && (9.2c)
\end{aligned}$$

  where $C \geq 0$ is a tuning parameter.
  $\epsilon_1, \ldots, \epsilon_n$ are known as **slack variables**, allowing some observations to be on the wrong side of the margin or hyperplane.
- Classify test observation $\boldsymbol{x}$ based on sign of $\hat{f}(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$.

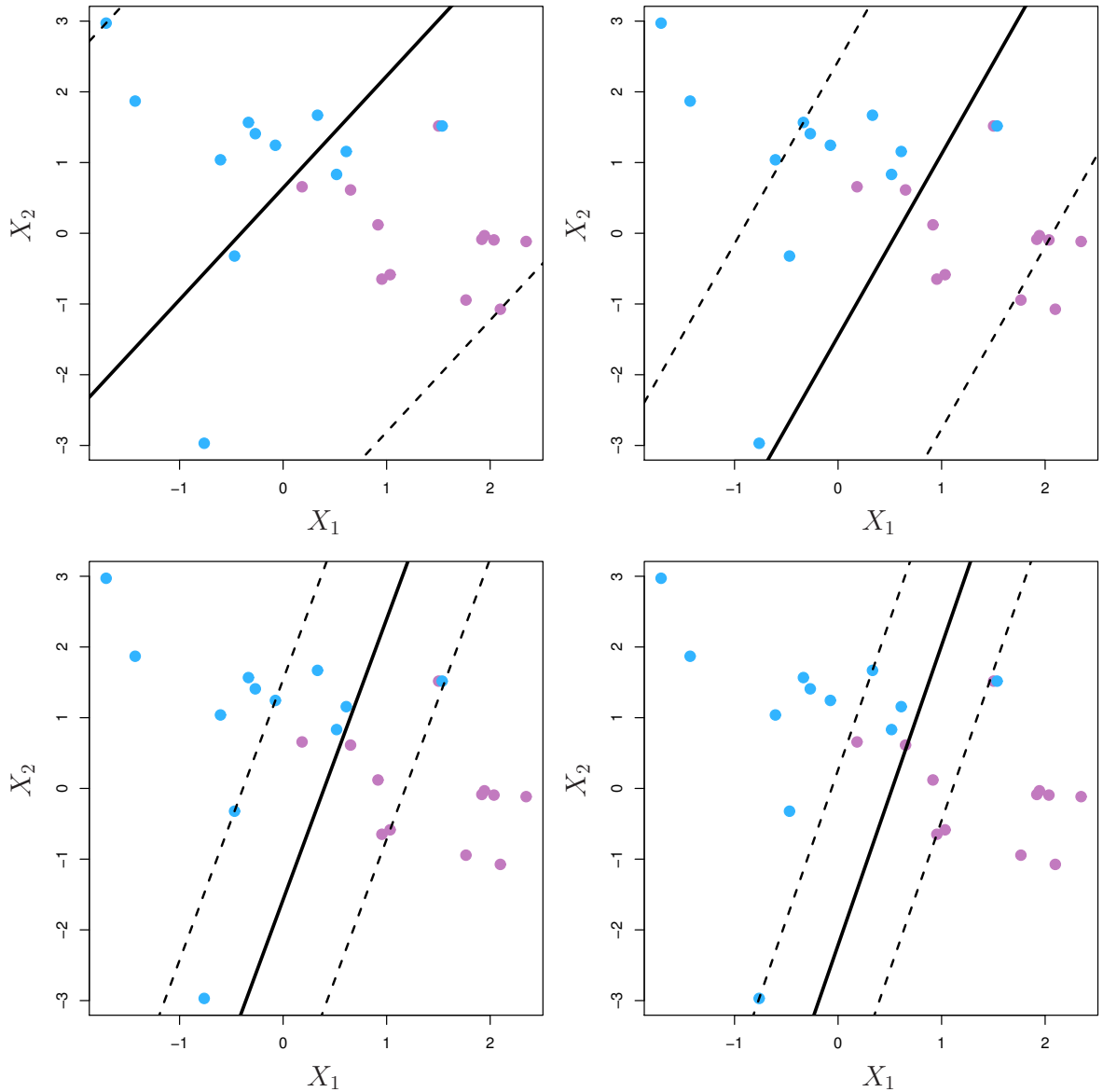
Support Vector Classifiers

Remarks on optimization problem

- Slack variable ϵ_i indicates location of $\boldsymbol{x}_i$ relative to margin:
  $\epsilon_i = 0$: $\boldsymbol{x}_i$ on correct side of margin,
  $\epsilon_i > 0$: $\boldsymbol{x}_i$ on wrong side of margin,
  $\epsilon_i > 1$: $\boldsymbol{x}_i$ on wrong side of hyperplane.
- C: bounds total permitted violation, budget for amount margin can be violated. $C = 0$ implies $\epsilon_i = 0$ for all i, i.e., recover maximal margin hyperplane optimization.
  C controls bias-variance trade-off.
- Example on next slide: Support vector classifier fit using 4 different values of C (value decreases from top left to bottom right).

Support Vector Classifiers

Example



Support Vector Classifiers

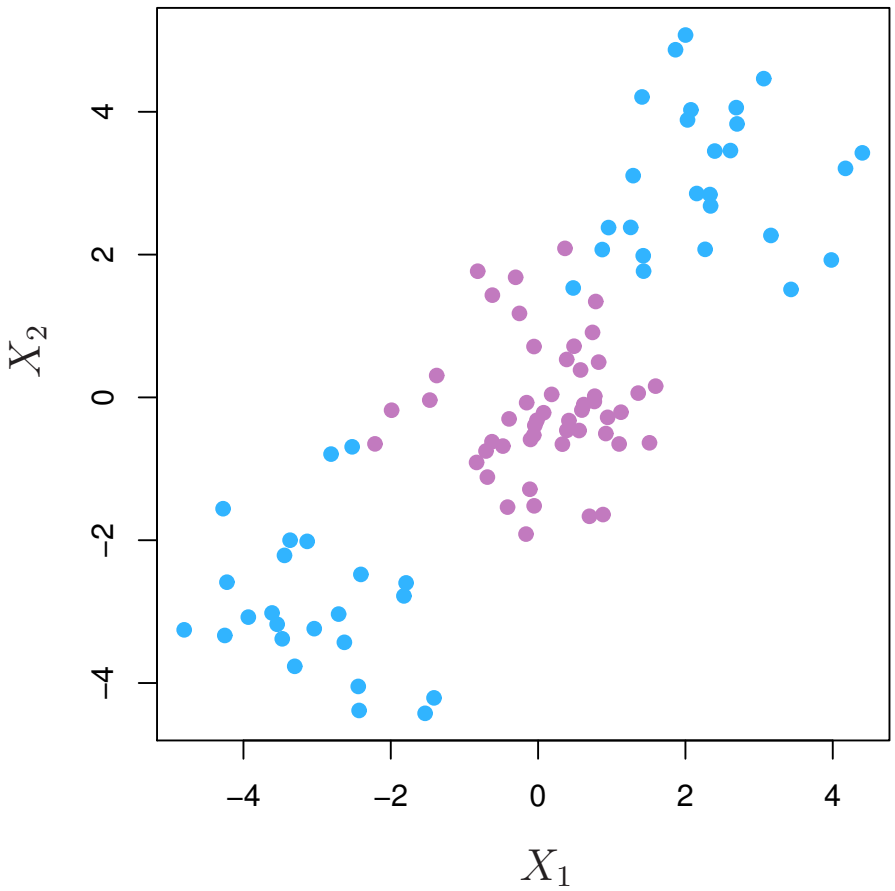
Support vectors

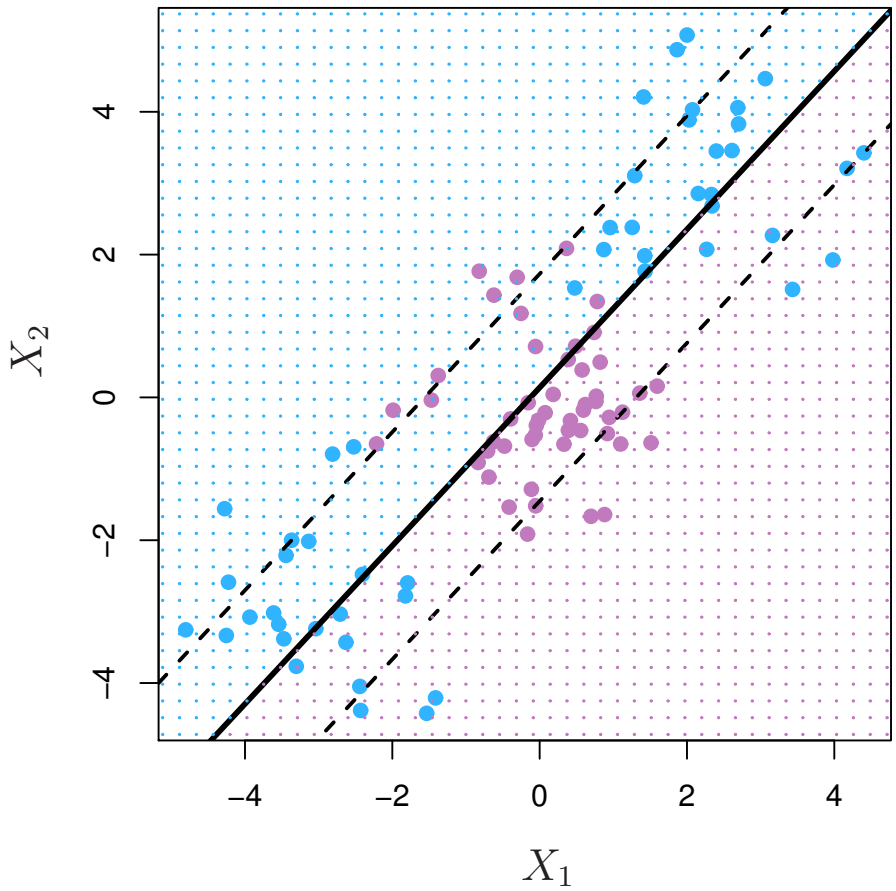
- Only observations on or violating margin will affect hyperplane/classifier.
- Observations on correct side orf margin have no effect on support vector classifier: changing their position without violating the margin would not change the classifier.
- The remaining observations (on or in violation of margin) are called **support vectors**.
- C large: margin wide, many observations violate margin, many support vectors, many observations involved determining hyperplane, classifier has low variance, potentially high bias (top left, pevious figure)
- C small: fewer support vectors, lower bias, higher variance (bottom right, pevious figure).
- Compare LDA classifier: depends on mean of **all** observations within each class as well as within-class covariance matrix computed using all observations.
- By contrast: logistic regression, unlike LDA, has low sensitivity to observations far from decision boundary. (Will see later: logistic regression and support vector classifier closely related.)

Contents

Support Vector Machines

Nonlinear decision boundaries





Some data sets require decision boundaries which are curved, hence a support vector classifier will perform poorly.

Support Vector Machines

Nonlinear decision boundaries

- In Chapter 7: extended model flexibility by adding nonlinear terms.
- Analogously: in addition to features $X_1, \ldots, X_p$, add $X_1^2, \ldots X_p^2$. Then (9.2) becomes

maximize M as a function of $\beta_0, \beta_{1,1}, \beta_{1,2} \ldots, \beta_{p,1}, \beta_{p,2}, M, \epsilon_1, \ldots, \epsilon_n$

subject to

$$\sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{j,k}^2 = 1, \tag{9.3a}$$

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j,1} x_{i,j} + \sum_{j=1}^{p} \beta_{j,2} x_{i,j}^2 \right) \geq M(1 - \epsilon_i), \tag{9.3b}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \qquad i = 1, \ldots, n, \tag{9.3c}$$

- Could also add interaction terms $X_j X_k$ in this way.

Support Vector Machines

New idea

- Enlarging the feature space in this way quickly makes the computations unmanageable.
- The key computational ingredient involves using **kernels**.
- Details of solving optimization problem (9.2) for support vector classifier involve **inner product** of observations

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_{i'} \rangle = \sum_{j=1}^{p} x_{i,j} x_{i',j}$$

rather than observations themselves.

- Can show that linear support vector classifier has representation

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle \tag{9.4}$$

in terms of n parameters $\{\alpha_i\}_{i=1}^{n}$ (one per training observation).

Support Vector Machines

New idea

- Estimating parameters $\beta_0, \alpha_1, \ldots, \alpha_n$ involves all $n(n-1)/2$ pairwise inner products of training observations.
- It turns out that in (9.4) α_i is only nonzero if $\boldsymbol{x}_i$ is a support point.
- Denoting the index set of the support points by $\mathscr{S}$, (9.4) becomes

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i \in \mathscr{S}} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle. \tag{9.5}$$

- This typically involves far fewer points than (9.4).
- Expanding the inner products in (9.5) establishes the relationship between the parameters α_i and the original coefficients β_j.
- **Summary:** computation and evaluation of linear classifier rests on evaluating innter products of point in feature space.
- Replace inner products $\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle$ with different function $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ referred to as a **kernel function** or simply **kernel**.

Support Vector Machines

Kernels

- Kernels quantify the degree of similarity or strength of relationship between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ of feature space.
- Could simply take inner product

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sum_{j=1}^{p} x_{i,j} x_{i',j},$$

(**linear kernel**), recovering support vector classifier. Here similarity of observations quantified using standard correlation.

- Alternatively,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \left(1 + \sum_{j=1}^{p} x_{i,j} x_{i',j}\right)^d$$

known as **polynomial kernel** of degree $d \in \mathbb{N}$. Leads to nonlinear decision boundary for support classifier, involves higher order terms in original features.

Support Vector Machines

Kernels

- Combination of support vector classifier with nonlinear kernel referred to as **support vector machine** (SVM).
- In this case the model has the form

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i \in \mathscr{S}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i). \tag{9.6}$$

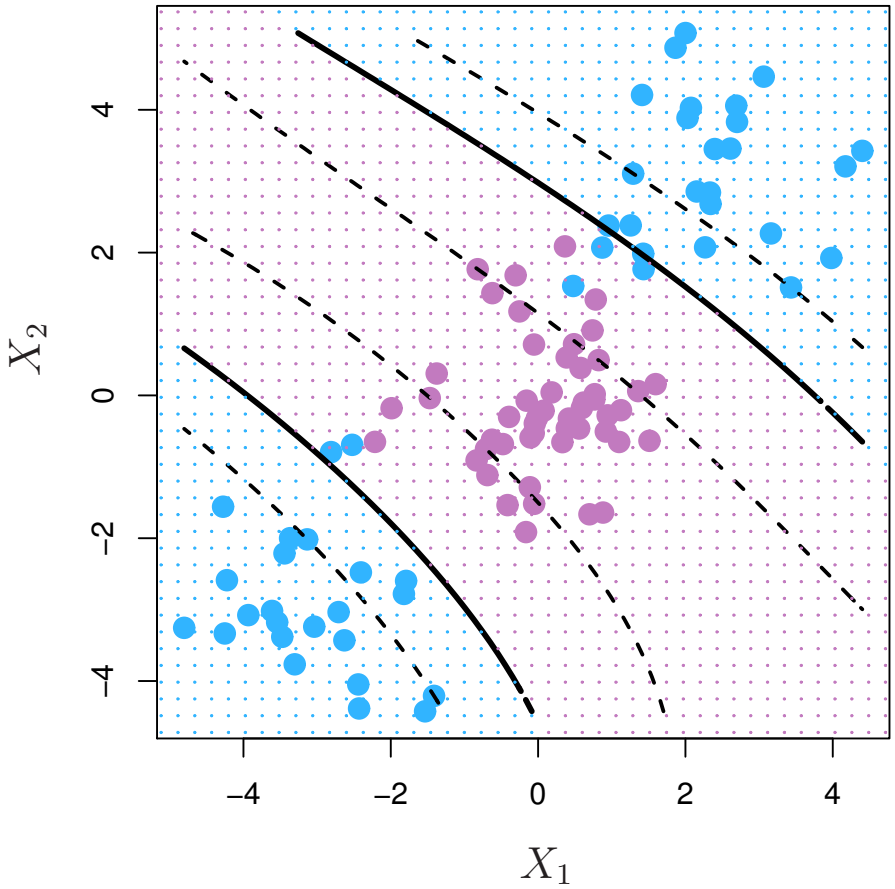
- Next picture: same data set as previous picture, SVM classifier using polynomial kernel (left panel). Right panel shows SVM using **radial kernel**

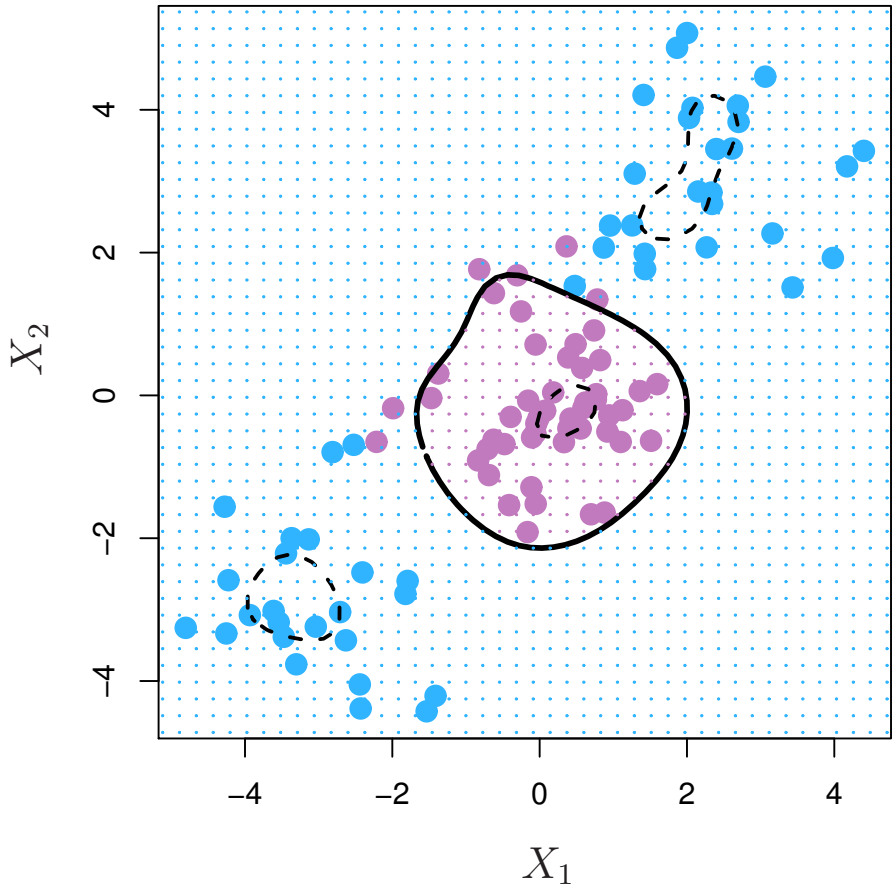
$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2\right) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2), \tag{9.7}$$

$\gamma > 0$ a parameter.

Support Vector Machines

Kernels





Support Vector Machines

Kernels

- If test observation $\boldsymbol{x} \in \mathbb{R}^p$ far from training observation $\boldsymbol{x}_i$ in the sense that $\|\boldsymbol{x} - \boldsymbol{x}_i\|_2$ large, then $K(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}_i\|_2^2)$ will be very small.
- Therefore in (9.6) observation $\boldsymbol{x}_i$ will have almost no influence on $\hat{f}(\boldsymbol{x})$.
- Recall: class label prediction depends on sign of $\hat{f}(\boldsymbol{x})$.
- Therefore, observations far away from $\boldsymbol{x}$ have little influence in class prediction for $\boldsymbol{x}$. (Radial kernel has very **local** behavior.)
- Advantage of using kernel over simply enlarging feature space with nonlinear expressions in predictor variables: explicit enlarging of feature space avoided, leading to great computational savings (particularly if feature space infinite-dimensional).

Contents

Support Vector Machines

SVMs with more than two classes

- Up to now: maximal margin/support vector classifiers/machines for binary classification.
- Concept rests on separating hyperplane idea, does not readily generalize to $K > 2$ classes.
- Of many proposals for this problem, two most popular are **one-versus-one** and **one-versus-all** approaches.

Support Vector Machines

One-versus-one classification

- For $K > 2$ classes, there are $\binom{K}{2} = K(K-1)/2$ possible class pairs.
- For each pair, construct separate SVM to compare just these two classes, e.g., for pair (k_1, k_2), code k_1 as $+1$ and k_2 as -1.
- To each test observation, apply all $K(K-1)/2$ SVMs, record how often observation assigned to each class.
- Finally, classify observation as belonging to class to which it was assigned most often.

Support Vector Machines

One-versus-all classification

- For $K > 2$ classes, fit K SVMs, each comparing one class with remaining $K - 1$ classes.
- Denote by $\beta_{0,k}, \beta_{1,k}, \ldots, \beta_{p,k}$ the coefficients resulting from fitting SVM to determine membership in k-th class, coded as $+1$, against membership in one of the $K - 1$ remaining classes, coded as -1.
- For test observation $\boldsymbol{x} = (x_1, \ldots, x_p)^\top$, assign to class k for which

$$\beta_0 + \beta_{1,k} x_1 + \cdots + \beta_{p,k} x_p$$

is largest, indicating high level of confidence that $\boldsymbol{x}$ belongs to class k.

Contents

Support Vector Machines

Relationship to logistic regression

- SVMs introduced in mid-1990s.
- Immediately successful due to performance, marketing, novelty of approach.
- Approach notably different from previously established classification methods such as logistic regression, LDA.
- Introduced kernel idea for expanding feature space and allowing nonlinear decision boundaries.
- In the meantime, deep connections between SVMs and classical statistical learning methods have been discovered.

Support Vector Machines

Relationship to logistic regression

- Can show: optimization problem (9.3) for fitting support vector classifier $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ can be reformulated as

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{arg\,min}} \left\{ \sum_{i=1}^{n} \max\{0, 1 - y_i f(x_i)\} + \lambda \sum_{j=1}^{p} \beta_j^2 \} \right\}, \tag{9.8}$$

  with $\lambda \geq 0$ a tuning parameter.

- λ large: $\beta_q 1, \ldots \beta_p$ small, more margin violations tolerated, low variance, high bias.
- λ small: fewer violations tolerated high variance, low bias.
  Corresponds to low value of C in (9.3c).
- Term $\lambda \sum_{j=1}^{p} \beta_j^2$ is exactly the ridge regression penalty term, provides regularization, controls bias-variance trade-off for support vector classifier.

Support Vector Machines

Relationship to logistic regression

- Formulation (9.8) has form "Loss + Penalty" (data misfit + regularization)

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\arg\min}\ \left(L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\right) \qquad \begin{array}{l} L: \text{loss function} \\ P: \text{ penalty function} \\ (\boldsymbol{x},\boldsymbol{y}): \textit{data}. \end{array} \tag{9.9}$$

  Loss function measures degree to which model fits data.
  Penalty function weighted by regularization parameter λ.

- Ridge regression and the lasso both employ loss function

$$L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2,$$

  with $P(\boldsymbol{\beta}) = \sum_{j=1}^{p}\beta_j^2$ for ridge regression and $\sum_{j=1}^{p}|\beta_j|$ for the lasso.

Support Vector Machines

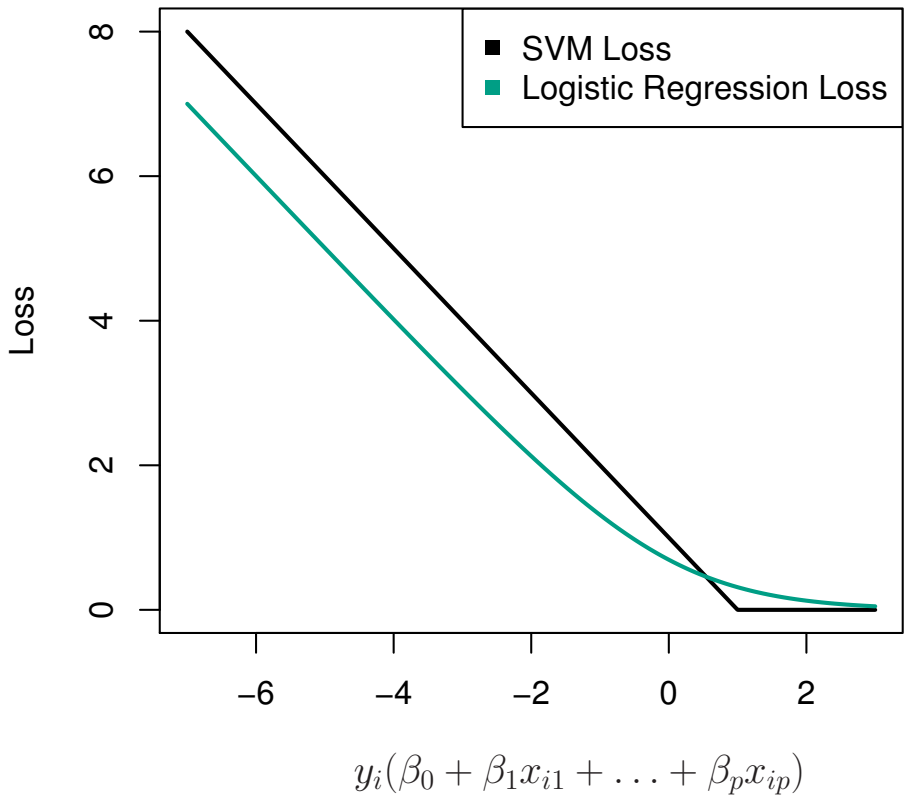
Relationship to logistic regression

- In (9.9), loss function has form

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}) = \max\{0, 1 - y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}\}$$

  known as **hinge loss**.

- Hinge loss function closely related to logistic regression loss function.



Support Vector Machines

Relationship to logistic regression

- In support classifier, only support vectors determine classifier, not observations on correct side of margin.
- Reason: loss function zero whenever $y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq 1$. Here margin cooresponds to value 1 and $\sum \beta_j^2$ determines margin width.
- Logistic regression loss function dies not vanish anywhere, but very small away from decision boundary.
- SVM's behave better for well-separated classes, logistic regression preferred when more overlap present.
- Importance of parameter C not initially realized.
- Nonlinear kernels possible also in logistic regression, but not as common in practice.
- Extension of SVM to regression: **support vector regression**. Uses different loss function than LS regression, where only residuals above positive threshold contribute (extension of margin concept from classification to regression).