Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

Course structure

- **Statistical learning is the 2nd module of Optimization (3CFU) and Statistical Learning (6CFU)**
- **Optimization is not mandatory but suggested**
- **48 hours**
  - 32 hours of theoretical lessons
  - 16 hours of lab (coding)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Textbook

- Gareth, James, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Second Edition. Springer, 2021.
- Book website: www.statlearning.com
- R files and data sets: https://www.statlearning.com/resources-second-edition

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Software

- R 4.1.2 (https://cloud.r-project.org)
  - This is the engine
- R-Studio (https://www.rstudio.com/products/rstudio/download/#download)
  - This is the user interface

- R vs MATLAB
  - Free
  - Statisticians make new packages available every day
  - Requires better coding skills
  - Usually slower than MATLAB (not a problem for us)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Student evaluation

- Teamwork (groups of 1, 2 or 3 students max)
- You choose the data set to analyze
- The analysis is based on the methods seen during the course
- R or MATLAB
- The tutor will be available for help during the course
- The evaluation will be based on a report
- Mid and final evaluation (only the final evaluation gives the grade)
- Each group will present in front of the class (in English)

Why this course

- To provide advanced statistical tools that enable the extraction of useful information from data sets…
- In order to make decisions…
- Following a statistical inference approach…
- Thus, knowing the risk to make the wrong decision.

Why this course

- In many contexts, collecting large amount of data is an easy task.
- It is not uncommon to find data sets of few GB or TB.
- Large data sets do not imply useful information.
- We need tools to extract the useful information.

Prerequisites

- Statistical inference
  - Distributions
  - Estimators
  - Confidence intervals
  - Hypothesis testing
- Simple and multiple regression (Chapter 3)
- Validation and cross-validation techniques(?) (Chapter 5)
- Monte Carlo and Bootstrap(?) (Chapter 5)

What we won't address

- Generalized linear models
- Stochastic processes
- Time series techniques
- State-space models (Kalman filter)
- Space-time models (next year -> Statistics for high dimensional data)

What is statistical learning?

$$Y = f(X) + \epsilon$$

- Y is the output
- $X = (X_1, \dots, X_p)$ is the set of predictors
- f is fixed but unknown function

The function f can be:

- A simple equation
- A complex equation (e.g., neural network)
- An algorithm (from few lines to a large number of lines)

Why estimate f?

- Two main goals
  - Prediction: $\hat{Y} = f(\hat{X})$
  - Inference
    - *Which predictors are associated with the response?*
    - *What is the relationship between the response and each predictor?*
    - *Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?*

How to estimate f?

- We need data!
- For us a data set has this form:

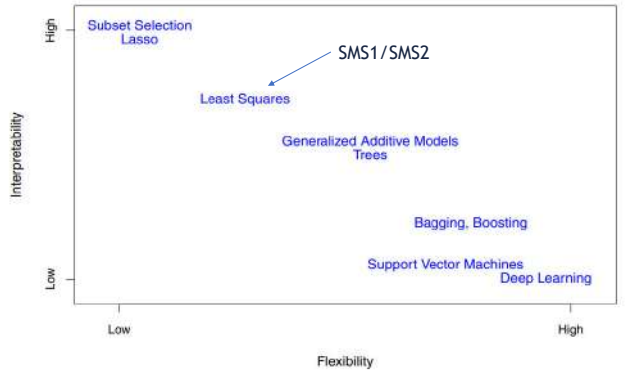
$$\{(x_1, y_1), \dots, (x_n, y_n)\}$$
$$x_i = (x_{i1}, \dots, x_{ip})', i = 1, \dots, n$$

- We will apply statistical learning methods to estimate the unknown function f such that $Y \approx \hat{f}(X)$.

Which methods?

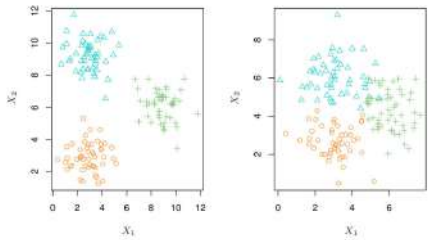
- Parametrics
  - We made assumptions on the functional form of f (e.g, linear, quadratic, etc.)
  - The function f is usually simple and has some parameters to be estimated (e.g., using least squares)
  - Once estimated, the model is easy to interpret
- Non-parametrics
  - No assumptions on f are made
  - We look for a f which is close to the data point (without overfitting)
  - Parameters are still there but usually they are in large number (e.g, splines)
  - Hard to interpret the model from its parameters

Flexibility vs interpretability



Which methods? (2)

- Supervised
  - When we have both X and Y (e.g, linear regression)
- Unsupervised
  - When we only have X (e.g, clustering)



Measuring the quality of fit

- In order to evaluate the performance of a statistical learning method on a given data set, we need some way to measure how well its predictions actually match the observed data.
- In the regression setting, the most commonly-used measure is the mean squared error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2, \qquad (2.5)$$

Measuring the quality of fit

- The MSE in (2.5) is computed using the training data that was used to fit the model and is called “training MSE”.
- But in general, we do not really care how well the method works on the training data.
- We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.
- We want to choose the method that gives the lowest **test MSE**, as opposed to the lowest ~~training MSE~~.
- If we had a large number of test observations, we could compute

$$\text{Ave}(y_0 - \hat{f}(x_0))^2, \qquad (2.6)$$

Overfitting problem

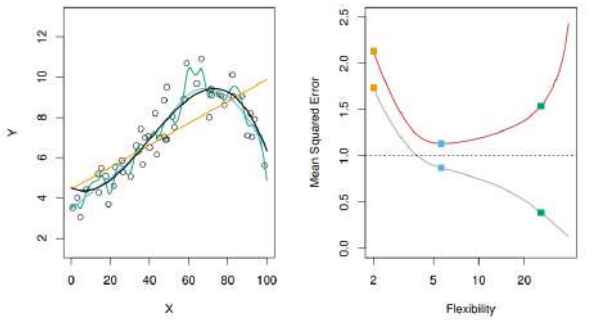


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

The Bias-Variance trade-off

- The U-shape observed in the test MSE curve is the result of two competing properties of statistical learning methods.
- The expected test MSE, for a given value x_0, can always be decomposed into the sum of three fundamental quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$ and the variance of the error term ϵ.

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon). \qquad (2.7)$$

- To minimize the expected test error, we need to select a statistical learning method that simultaneously achieves **low variance** and **low bias**.

The Bias-Variance trade-off (2)

- **Variance** refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set. If a method has high variance, then small changes in the training data can result in large changes in $\hat{f}$.
- In general, more flexible statistical methods have higher variance!
- **Bias** refers to the error that is introduced by approximating a real-life problem by a much simpler model.
- As a general rule, as we use more flexible methods, the variance will increase and the bias will decrease (trade-off).

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

Chapter 3

Resampling Methods

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

What is resampling?

- Given a data set it is often useful to (re)sample inside the training data set in order to estimate the model f multiple times.
- We can study how $\hat{f}$ changes when changing the training data.
- It is time consuming but with modern computing it is feasible (not always)
- We may obtain information which are not available if we fit the model only one time (is this totally true?)
- Computing time is what we pay for our math ignorance!

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Validation

- We have seen that test-MSE is better than training-MSE when it comes to model selection (e.g., selecting the model flexibility)
- Usually the test data set is not available (if it was we would use test+training data to fit the model)
- Instead, we define the test data set as a subset of the data set that we have.
- In practice, we hold out part of the observations and we use them for model validation and selection.

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Validation set approach

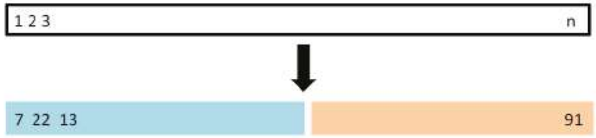


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

Validation set approach

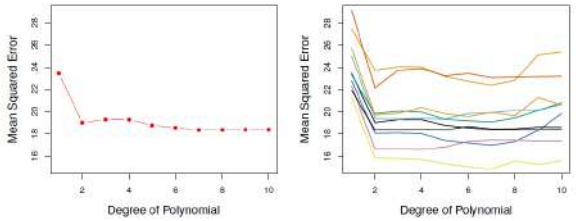


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

Validation set approach

- **Pros**
  - An independent test data set is not needed
  - Faster than other validation approaches
- **Cons**
  - Changing the validation set leads to different test MSEs
  - The test MSE is overestimated since we use less data (e.g., *n*/2) to fit the model
- **What about the full data set?**

Leave-One-Out Cross-Validation

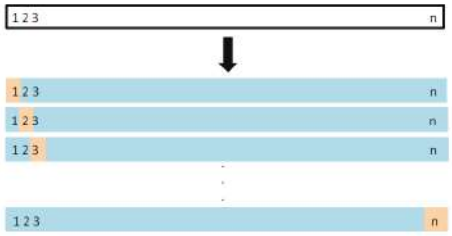


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

Leave-One-Out Cross-Validation

- For each model estimation (leaving one data point out) we can compute $MSE_i = (y_i - \hat{y}_i)^2$
- Which is a poor estimate of the test-MSE since based on one data point
- Therefore, we iterate over the *n* data points to get:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i. \qquad (5.1)$$

Leave-One-Out Cross-Validationn set approach

- **Pros**
  - LOO Test MSE is less biased than the test MSE computed using the validation set approach (because we always use *n-1* data points instead of *n/2* to fit the model).
  - There is no data randomization
- **Cons**
  - Very time consuming (the model is estimated *n* times)

k-Fold Cross-Validation

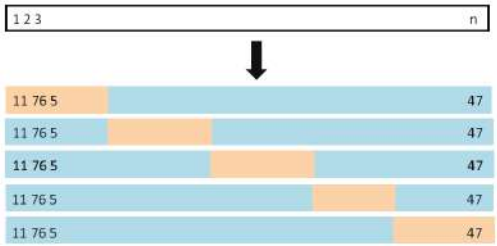


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

LOOCV vs k-Fold CV

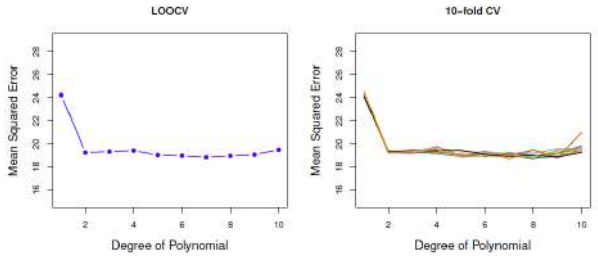


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`*. Left: The LOOCV error curve. Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

Bootstrap

- Validations approaches seen so far are based on a resampling scheme without re-insertion (data points used for training are not used for validation and vice-versa).
- The boostrap approach removes this «constraint». If we have a data set of *n* data points, we can generate new data sets of size *n* where a given data point may be repeated up to *n* times.
- Useful when *n* is small and splitting the data set is not viable.
- Bootstrap is used for statistical inference when statistical inference is not easy (because *n* is small)

Bootstrap

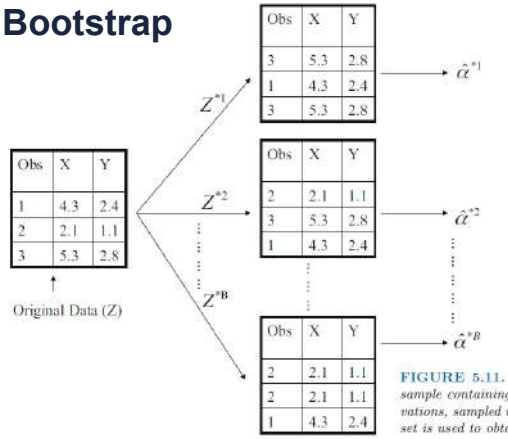


FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing n = 3 observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

Bootstrap example

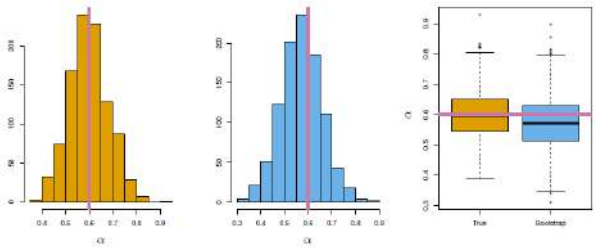


FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

Chapter 6

Linear Model Selection and Regularization

Where were we (after SMS1 and SMS2)

- **Multiple linear regression (Chapter 3 of the book)**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon, \qquad (3.19)$$

- **We know how to:**
  - Estimate the betas coefficients
  - Make hypothesis tests on the coefficients
  - Make predictions using the model

What happens when p is large?

- When $n \gg p$ the estimators of β have low variance. Everything is good.
- When $n \not\gg p$ the estimators of β have high variance. Predictions are bad.
- When $p > n$ the least squares approach cannot be used.
- Still, $n \not\gg p$ and $p > n$ are not uncommon in real-life important applications .
- We need a way to deal with these two cases.
- And even when $n \gg p$ we may want to keep p as small as possible (for model interpretability)

Alternatives to least squares

- **Subset Selection**
  - Least squares on a reduced set of variables (still least squares).
- **Shrinkage (regularization)**
  - Estimated coefficients are shrunken towards zero relative to the least squares estimates.
- **Dimension Reduction**
  - This approach involves projecting the p predictors into an M-dimensional subspace, where $M < p$.

Subset selection methods

- 3 options
  - Best subset selection
  - Forward Stepwise Selection
  - Backward Stepwise Selection

Best subset selection

- We fit a separate least squares regression for each possible combination of the p predictors. From models with only one predictor to models with 2, 3, 4… predictors.
- The total number of models to fit is 2^p.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Forward Stepwise Selection

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all the predictors are in the model.
- At each step the variable that gives the greatest additional improvement to the fit is added to the model.
- This amounts to a total of *1+p(p+ 1)/2* models (Compare stepwise and best selection when *p=20*)

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Backward Stepwise Selection

- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.
- It begins with the full least squares model containing all *p* predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Shrinkage methods

- We can fit a model containing all *p* predictors using a technique that shrinks the coefficient estimates towards zero.
- Shrinking the coefficient estimates can significantly reduce their variance.
- 2 options
  - Ridge regression
  - Lasso

Ridge regression

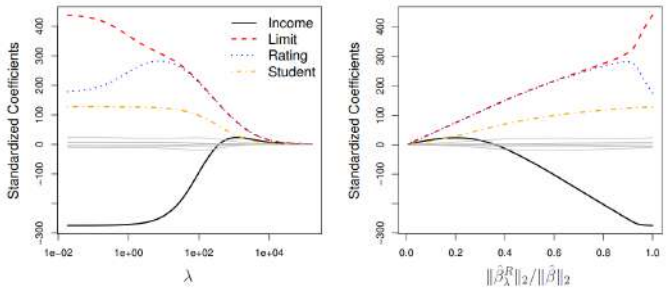
- Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2, \qquad (6.5)$$

- λ is a tuning parameter to be determined separately.
- Unlike least squares, which generates only one set of coefficient estimates, ridge regression will produce a different set of coefficient estimates, $\hat{\beta}_\lambda^R$, for each value of λ.

Coefficients «trajectories»

- From the book. A higher λ implies smaller (in absolute value) coefficients.



Bias-variance trade-off

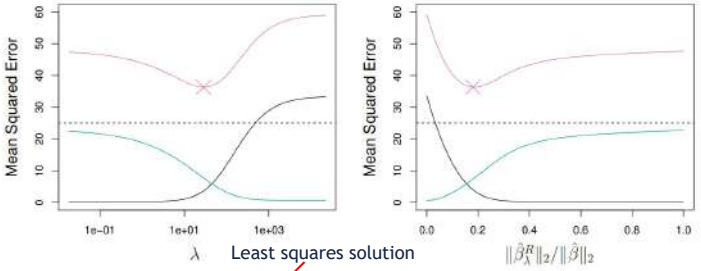


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

Ridge regression pros and cons

- When $p \cong n$ least squares estimates will be extremely variable (a small change in the training data can cause a large change in the least squares coefficient estimates)
- Ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance.
- Ridge regression also has substantial computational advantages over best subset selection.
- Ridge regression will include all p predictors in the final model (hard to interpret).

Lasso regression

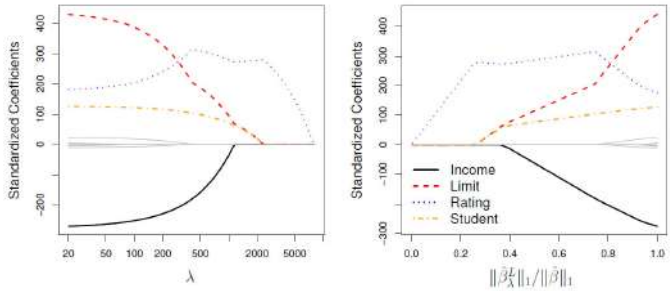
- The lasso is an alternative to ridge regression that overcomes the disadvantage of ridge regression:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|. \qquad (6.7)$$

- (6.7) is similar to ridge regression but the lasso uses an $\ell 1$ penalty instead of an $\ell 2$ penalty.
- The $\ell 1$ penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when λ is sufficiently large. We say that the lasso yields sparse models.

Coefficients «trajectories»

- From the book. A higher λ implies smaller (in absolute value) coefficients.
- When λ is large enough some coefficient estimates are zero.



Why lasso performs variable selection?

- One can show that the lasso and ridge regression coefficient estimates solve the problems

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j| \le s \qquad (6.8)$$

and

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}\beta_j^2 \le s, \qquad (6.9)$$

Why lasso performs variable selection?

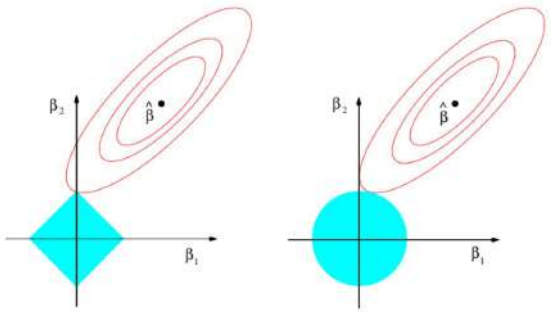


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

How to choose λ?

- λ is NOT a model parameter
- Cannot be jointly estimated with the model parameters β
- Solution: validation techniques
  - validation (data set split for model estimation and validation)
  - k-fold cross-validation
  - leave-one-out validation

Dimension reduction: principal components regression

- Principal components regression is an alternative to stepwise and shrinkage.
- The idea is to «create» *M*<*p* new predictors that most retain the information of the *p* predictors.
- This is done by linearly combining the *p* predictors.
- Useful for data reduction but we loose model interpretability.
- Better rely on shrinkage unless we only care for the model fitting capability and/or prediction capability.

Regression in High Dimensions

- What happens when *p* is large?

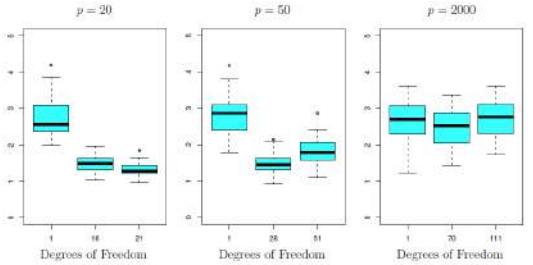


FIGURE 6.24. *The lasso was performed with* $n = 100$ *observations and three values of* p, *the number of features. Of the* p *features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter* λ *in (6.7). For ease of interpretation, rather than reporting* λ, *the* degrees of freedom *are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When* $p = 20$, *the lowest test MSE was obtained with the smallest amount of regularization. When* $p = 50$, *the lowest test MSE was achieved when there is a substantial amount of regularization. When* $p = 2{,}000$ *the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.*

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

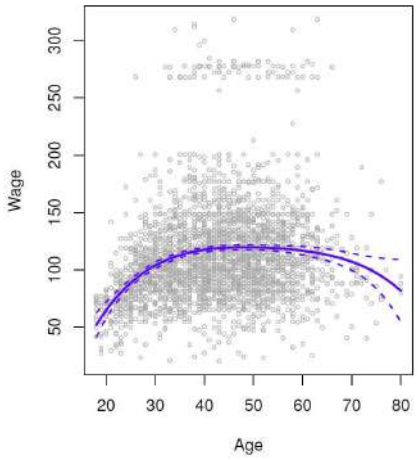
Chapter 7

Moving Beyond Linearity

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

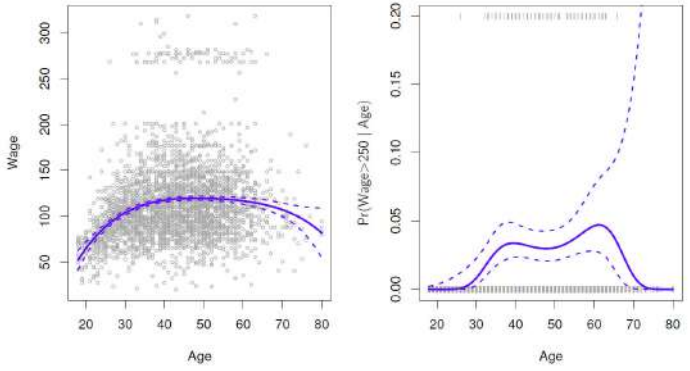
Non-linear regression

- The relationship between X and Y may be non-linear. Which options do we have?



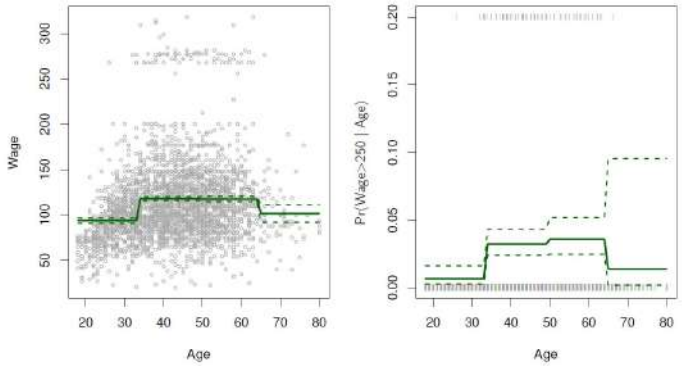
UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Polynomial regression

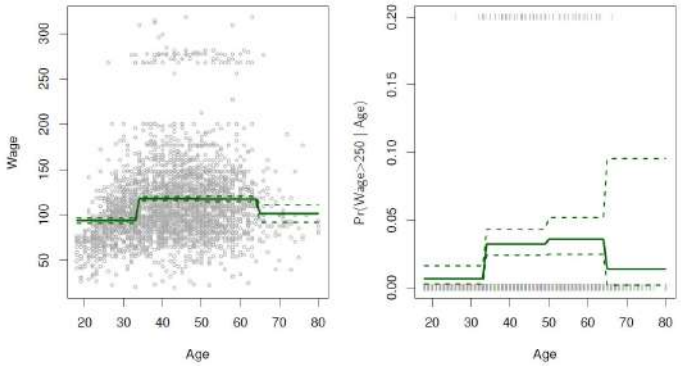


UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

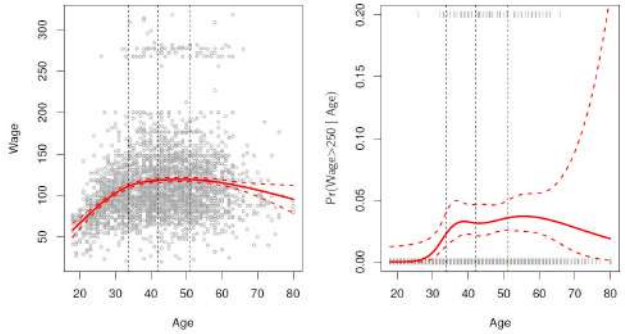
Step functions



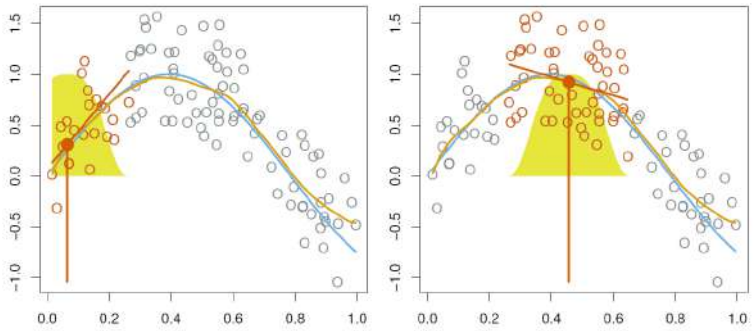
Step functions



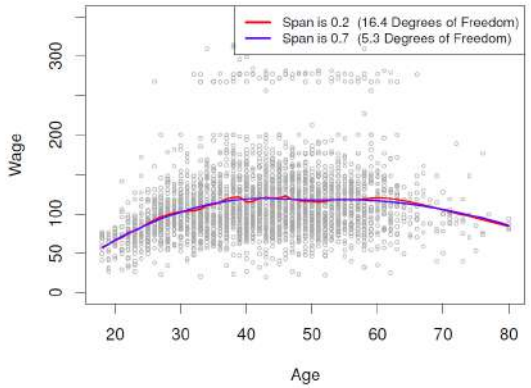
Basis functions (splines)



Local regression



Local regression



Generalized additive models (GAMs)

- Methods seen so far allow do deal with only one regressor (*p=1*)
- GAMs extend those methods to multiple regressors
- From

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$$

- To

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\ &= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i. \end{aligned} \qquad (7.15)$$

GAM example

- From the book:

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon \qquad (7.16)$$

- We look for a GAM which explains wage as a function of some regressors.
- Each regressor has a different function and it is not linear!
- **Why we do not want lineary on year, age and education?**

Fitted GAM

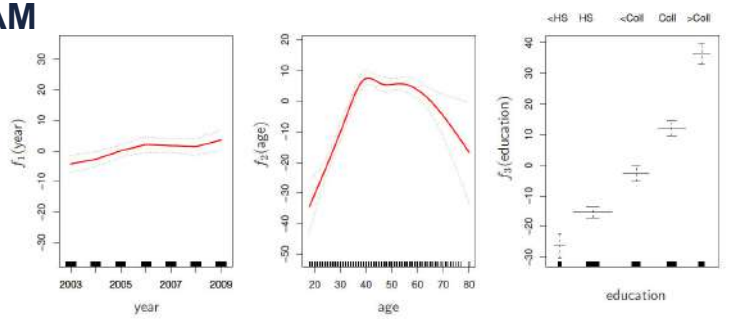


FIGURE 7.11. *For the* `Wage` *data, plots of the relationship between each feature and the response,* `wage`*, in the fitted model (7.16). Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in* `year` *and* `age`*, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable* `education`*.*

When to use GAMs

- When the relationship between a regressor X and the Y is non-linear.
- When regressors contribute to the Y additively.
- GAMs do not automatically handle interactions between regressors but…
- If an interaction is needed, simply add it to the model as a new regressor.

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Chapter 8

Tree-Based Methods

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Tree-Based Methods

- Main idea
  - The space of the predictors $(X_1, \dots, X_p)$ is partitioned into a number of simple regions (what is a partition?)
  - For each region, the mean or the mode of the response variable *Y* is computed, say $\hat{Y}$
  - $\hat{Y}$ is the prediction for all observations falling in the same region
- Problems:
  - Which is the best space partitioning?
  - Is there a model at all?
  - Can we do statistical inference?

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Regression tree - Example from the book

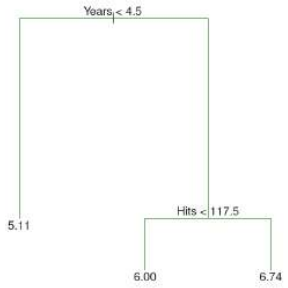


FIGURE 8.1. *For the* Hitters *data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year.*

Regression tree – Space partitioning

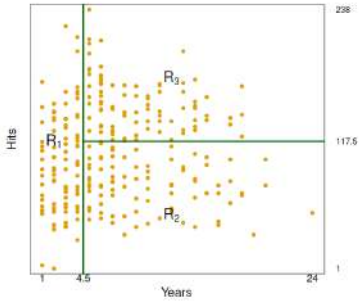


FIGURE 8.2. *The three-region partition for the* Hitters *data set from the regression tree illustrated in Figure 8.1.*

Prediction via partitioning of the feature space

1. We divide the predictor space — that is, the set of possible values for $X_1, X_2, \ldots, X_p$ — into J distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$.
2. For every observation that falls into the region R_j, we make the same prediction, which is simply the mean of the response values for the training observations in R_j.

Best partitioning

- The goal is to find regions that minimize the RSS given by

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2, \qquad (8.1)$$

- It is computationally infeasible to consider every possible partition of the feature space into J regions
- So, we take a top-down greedy approach that is known as recursive binary splitting

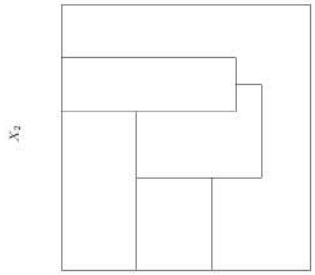
Recursive binary splitting

- The recursive binary splitting approach is top-down because it begins at the top of the tree (where all observations belong to a single region) and then successively splits the predictor space.
- It is greedy because at each step of the tree-building process, the best split is made at that step, rather than looking ahead and picking a split that will lead to a better tree in some future step (easy and fast but suboptimal!)

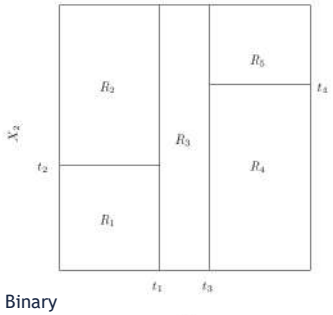
Recursive binary splitting

- When to end the splitting?
- One option is to stop the splitting when the RSS stops to decrease significantly but…
- Since the algorithm is greedy, it is hard to tell if a future split will bring to a significant improvement in the RSS
- The solution is:
  - Build the largest tree T_0 (stop when the leaves have a small number of observations)
  - Prune the tree

Binary splitting

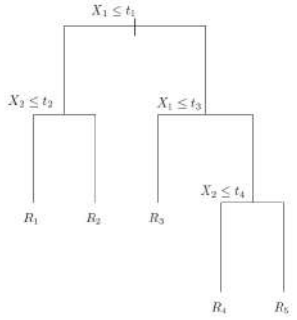


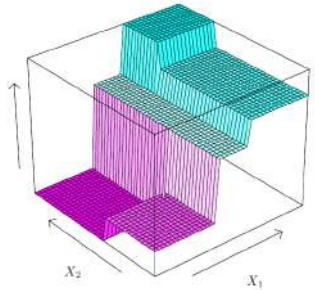
Non-binary splitting



Binary splitting

The corresponding tree





Tree pruning

- How to prune the tree? Which is the best sub-tree?
- The number of sub-trees is extremely large. We need a way to select a small number of sub-trees for consideration.
- *Cost complexity pruning*: for each value of α there corresponds a subtree $T \subset T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \qquad (8.4)$$

is minimum. Note that $|T|$ is the number of terminal nodes (complexity)

Tree pruning

- As we increase α from zero in (8.4), branches get pruned from the tree in a nested and predictable fashion, so obtaining the whole sequence of subtrees as a function of α is easy.
- We can select a value of α using a validation set or using cross-validation.
- We then return to the full data set and obtain the subtree corresponding to α.

Example
Full tree

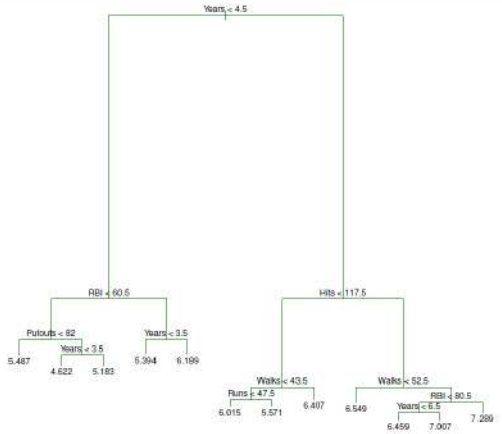


FIGURE 8.4. *Regression tree analysis for the* Hitters *data. The unpruned tree that results from top-down greedy splitting on the training data is shown.*

Example
Pruning

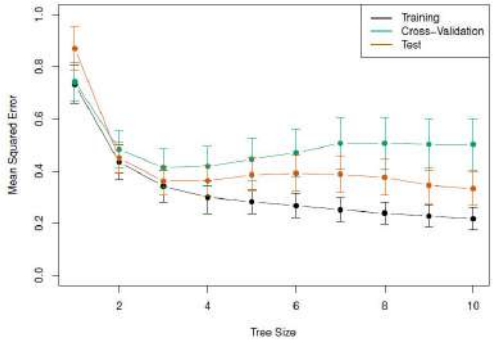


FIGURE 8.5. *Regression tree analysis for the* Hitters *data. The training, cross-validation, and test MSE are shown as a function of the number of terminal nodes in the pruned tree. Standard error bands are displayed. The minimum cross-validation error occurs at a tree size of three.*

Example
Pruned tree

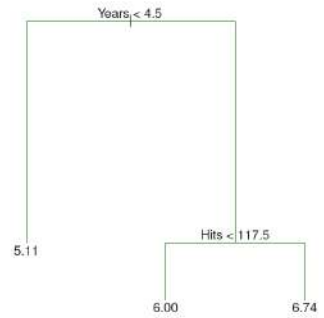


FIGURE 8.1. *For the* Hitters *data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year.*

Trees vs linear models

- Linear model

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j, \tag{8.8}$$

- Tree

$$f(X) = \sum_{m=1}^{M} c_m \cdot 1_{(X \in R_m)} \tag{8.9}$$

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Trees vs linear models

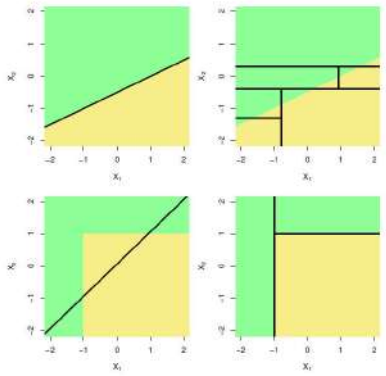


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Advantages and disadvantages of trees

- Trees are very easy to explain to people.
- Decision trees more closely mirror human decision-making process.
- Trees can be displayed graphically and are easily interpreted even by a non-expert.
- Trees generally do not have the same level of predictive accuracy of previously seen regression approaches.
- Trees can be very non-robust. A small change in the data can cause a large change in the final estimated tree.

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

Chapter 8.2

Ensemble methods

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Ensemble methods

- Ensemble method is an approach that combines many simple "building ensemble block" models in order to obtain a single and potentially more powerful model.
  - Bagging
  - Random forests
  - Boosting
  - Bayesian learners additive regression trees
- All these methods has a regression tree as building block

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Bagging (bootstrap aggregation)

- Regression trees have high variance (if we split the data set and fit 2 trees, they can be quite different)
- If we had many training data sets we could fit multiple trees and take the "average" (but this is not the case)
- Instead, we can use bootstrap to generate "new" training data sets from the one we have.
- What we actually average are the predictions (not the tree object)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Bagging

- With bagging, trees are not pruned so they have low bias (but high variance).
- Taking the prediction average reduces the variance.
- How many trees? Could be any number above 100.
- Usually, the number of trees is increased until the test MSE stabilizes (no overfitting).

Bagging

- What we lose with bagging is model interpretability (since each tree could be very different)
- We can however obtain an overall summary of the importance of each predictor using the RSS
- We record the total amount that the RSS (8.1) is decreased due to splits over a given predictor, averaged over all the trees
- A large value means an important predictor

Random forests

- If the data set has few (very) import predictors, all the bootstrapped trees will tend to use those predictors for splitting
- Averaging similar trees limits the variance reduction
- Random forests method is similar to bagging, but splits are constrained: **a random sample of m predictors is chosen as split candidates from the full set of p predictors.**
- Typically, $m \approx \sqrt{p}$
- We can think of this process as decorrelating the trees

Random forests

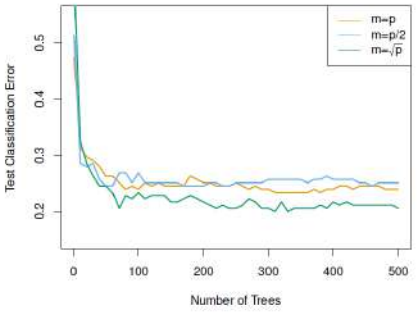


FIGURE 8.10. *Results from random forests for the 15-class gene expression data set with $p = 500$ predictors. The test error is displayed as a function of the number of trees. Each colored line corresponds to a different value of m, the number of predictors available for splitting at each interior tree node. Random forests ($m < p$) lead to a slight improvement over bagging ($m = p$). A single classification tree has an error rate of 45.7%.*

Boosting

- Boosting is another approach for improving the predictions resulting from a decision tree
- Trees are grown sequentially: each tree is grown using information from previously grown trees
- Boosting does not involve bootstrap sampling: each tree is fit on a modified version of the original data set
- Given the current model, we fit a decision tree to the residuals from the model. That is, we fit a tree using the current residuals (rather than the outcome Y, as the response)

Boosting

- We then add this new decision tree into the fitted function in order to update the residuals
- Each of these trees can be rather small, with just a few terminal nodes
- By fitting small trees to the residuals, we slowly improve f in areas where it does not perform well
- The shrinkage parameter λ slows the process allowing more and different shaped trees to "attack" the residuals
- In general, statistical learning approaches that learn slowly tend to perform well

Boosting

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \quad (8.10)$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \quad (8.11)$$
3. Output the boosted model,
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \quad (8.12)$$

Boosting tuning parameters

- The number of trees B. Unlike bagging and random forests, boosting can overfit if B is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select B
- The shrinkage parameter λ that controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small λ can require using a very large value of B in order to achieve good performance
- The number d of splits in each tree, which controls the complexity of the boosted ensemble. Often d = 1 works well, in which case each tree has a single split

Boosting performance

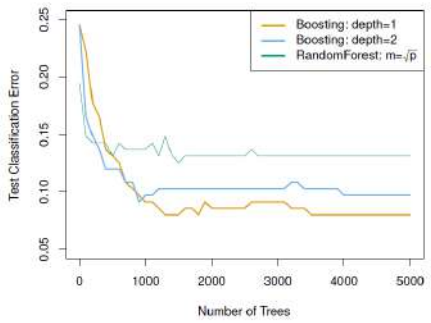


FIGURE 8.11. *Results from performing boosting and random forests on the 15-class gene expression data set in order to predict* cancer *versus* normal. *The test error is displayed as a function of the number of trees. For the two boosted models,* $\lambda = 0.01$. *Depth-1 trees slightly outperform depth-2 trees, and both outperform the random forest, although the standard errors are around 0.02, making none of these differences significant. The test error rate for a single tree is 24 %.*

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Chapter 4

Classification

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Why do we go back? (In the book)

- In many situations, the response variable *Y* is qualitative rather than quantitative
- Regression approaches are (usually) not suitable for predicting a quantitative variable
- We now study approaches for predicting qualitative responses, a process known as **classification**
- Often methods used for classification first predict the probability that the observation belongs to each of the categories as the basis for making the classification (**so they are similar to regression approaches**)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Classification overview

- Just as in the regression setting, in the classification setting we have a set of training observations $(x_1, y_1), \ldots, (x_n, y_n)$
- We use training data to fit a model (classifier)
- We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier (as in regression)

Example from the book

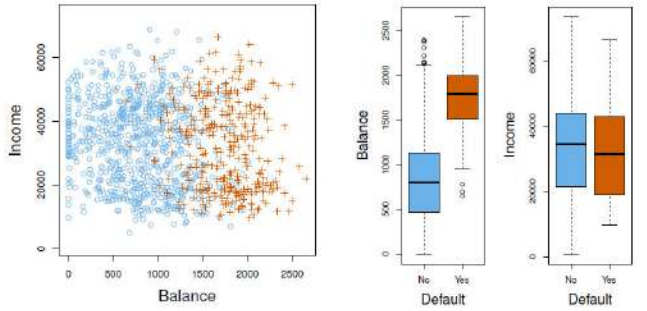


FIGURE 4.1. *The* `Default` *data set.* Left: *The annual incomes and monthly credit card balances of a number of individuals. The individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.* Center: *Boxplots of* `balance` *as a function of* `default` *status.* Right: *Boxplots of* `income` *as a function of* `default` *status.*

Why not linear regression?

- Suppose we are trying to predict the medical condition of a patient in the emergency room on the basis of her symptoms
- We could consider encoding these values as a quantitative response variable, *Y*, as follows

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

- Using this coding, least squares could be used to fit a linear regression model to predict *Y* on the basis of a set of predictors

Why not linear regression?

- Problems:
  - What happens if we change the ordering *Y*?
  - What is the "difference" between drug overdose (2) and stroke (1)?
  - And the difference between epileptic seizure (3) and drug overdose (2)?
  - Should this difference be the same? (numerically)
- Each ordering would produce fundamentally different linear models that would ultimately lead to different sets of predictions on test observations
- In general, **there is no natural way** to convert a qualitative response variable with more than two levels into a quantitative response that is ready for linear regression

Why not linear regression?

- For a binary (two level) qualitative response, the situation is better. For instance, when the patient's medical condition are stroke and drug overdose

$$Y = \begin{cases} 0 & \text{if stroke;} \\ 1 & \text{if drug overdose} \end{cases}$$

- We could then fit a linear regression to this binary response and predict drug overdose if Y>0.5 and stroke otherwise. In the binary case, even if we flip the above coding, linear regression will produce the same final predictions

Why not linear regression?

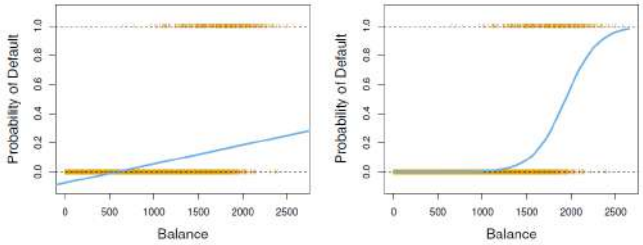


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default` *(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

Logistic regression

- Rather than modeling the response Y directly, logistic regression models the probability that Y belongs to a particular category
- For example, the probability of default given balance can be written as

$$\Pr(\texttt{default} = \texttt{Yes}|\texttt{balance})$$

- The probability will range from 0 to 1
- For any given value of balance, a prediction can be made for default

Logistic regression

- Rather than modeling the response Y directly, logistic regression models the probability that Y belongs to a particular category
- For example, the probability of default given balance can be written as

$$\Pr(\texttt{default} = \texttt{Yes}|\texttt{balance})$$

- The probability will range from 0 to 1
- For any given value of balance, a prediction can be made for default

The logistic model

- We now consider the binary case. For convenience we use the generic 0/1 coding for the response
- How should we model the relationship between Pr(Y=1|X) and X?
- If we use a linear regression model:

$$p(X) = \beta_0 + \beta_1 X. \qquad (4.1)$$

probabilities are not restricted to be in [0, 1]

The logistic model

- To avoid this problem, we must model p(X) using a function that gives outputs between 0 and 1 for all values of X
- Many functions meet this description. In logistic regression, we use the logistic function

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \qquad (4.2)$$

- The logistic function will always produce an S-shaped curve and regardless of the value of X we obtain predictions in [0, 1]

The logistic model

- After a bit of manipulation of (4.2), we find that

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X. \qquad (4.4)$$

- The left-hand side is called the log odds or logit. We see that the logistic regression model (4.2) has a logit that is linear in X.
- Because the relationship between p(X) and X in (4.2) is not a straight line, the slope does not correspond to the change in p(X)
- The amount that p(X) changes due to a one-unit change in X depends on the current value of X

Multiple logistic regression

- We can generalize (4.4) as follows

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p, \qquad (4.6)$$

or

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}. \qquad (4.7)$$

Multinomial logistic regression

- The logistic regression approach allows for K = 2 classes for Y.
- It is possible to extend the two-class logistic regression approach to the setting of K > 2 classes. This extension is known as multinomial logistic regression
- We first select a single class to serve as the **baseline**; without loss of generality, we select the Kth class for this role

Multinomial logistic regression

- Then we replace the model (4.7) with the model

$$\Pr(Y = k|X = x) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}} \qquad (4.10)$$

for k=1,…,K−1, and

$$\Pr(Y = K|X = x) = \frac{1}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}}. \qquad (4.11)$$

Multinomial logistic regression

- It is not hard to show that for k=1,…,K−1

$$\log\left(\frac{\Pr(Y = k|X = x)}{\Pr(Y = K|X = x)}\right) = \beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p. \qquad (4.12)$$

- Equation 4.12 indicates that once again, the log odds between any pair of classes is linear in the features
- It turns out that in (4.10)-(4.12), the decision to treat the Kth class as the baseline is unimportant
- Nonetheless, interpretation of the coefficients in a multinomial logistic regression model must be done with care, since it is tied to the choice of baseline

Generalized linear models

- In some cases, Y is neither qualitative nor quantitative, and so neither linear regression nor classification approaches are applicable
- This happens when Y is a non-negative integer value or counts
- With linear regression we may predict negative values and in general the prediction is not an integer (this is not a big problem actually)

Example: bike sharing data

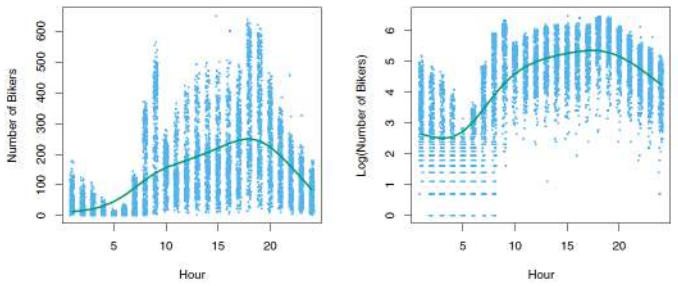


FIGURE 4.14. Left: *On the* Bikeshare *dataset, the number of bikers is displayed on the y-axis, and the hour of the day is displayed on the x-axis. Jitter was applied for ease of visualization. For the most part, as the mean number of bikers increases, so does the variance in the number of bikers. A smoothing spline fit is shown in green.* Right: *The log of the number of bikers is now displayed on the y-axis.*

Generalized linear models

- In the bike sharing data set, it is reasonable to suspect that when the expected value of bikers is small, the variance of bikers should be small as well. At 2 AM we expect that extremely few people will use a bike, and moreover that there should be **little variance** associated with the number of users
- The mean-variance relationship is displayed in the left-hand panel of Figure 4.14. This is a major violation of the assumptions of a linear model, which state that ϵ is a mean-zero random error term with a **constant variance** (not a function of the covariates)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Generalized linear models

- Some of the problems that arise when fitting a linear regression model to count data can be overcome by transforming the response; for instance, we can fit the model

$$\log(Y) = \sum_{j=1}^{p} X_j \beta_j + \epsilon$$

- Transforming the response avoids the possibility of negative predictions, and it overcomes much of the heteroscedasticity in the untransformed data, as is shown in the right-hand panel of Figure 4.14 (but is not totally satisfactory)

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Poisson regression

- To overcome the inadequacies of linear regression, we make use of an alternative approach called Poisson regression
- Suppose that a random variable Y takes on nonnegative integer values, i.e. $Y \in \{0,1,2,\ldots\}$. If Y follows the Poisson distribution, then

$$\Pr(Y = k) = \frac{e^{-\lambda}\lambda^k}{k!} \quad \text{for } k = 0, 1, 2, \ldots. \tag{4.35}$$

- Here, $\lambda > 0$ is the expected value of Y, i.e. E(Y). It turns out that λ also equals the variance of Y, i.e. $\lambda = E(Y) = Var(Y)$

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Poisson regression

- The Poisson distribution is typically used to model counts; this is a natural choice for a number of reasons, including the fact that counts, like the Poisson distribution, take on nonnegative integer values
- In a Poisson distribution, λ is constant. However, we expect the mean number of users of the bike sharing program, λ = E(Y), to vary as a function of the hour of the day, the month of the year, the weather conditions, and so forth

Poisson regression

- We consider the following model for the mean λ = E(Y), which we now write as $\lambda(X_1,\ldots,X_p)$ to emphasize that it is a function of the covariates $X_1,\ldots,X_p$

$$\log(\lambda(X_1,\ldots,X_p)) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \qquad (4.36)$$

- or equivalently

$$\lambda(X_1,\ldots,X_p) = e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}. \qquad (4.37)$$

- Note that λ is always positive
- What about model estimation?

Statistical Learning
a.a. 2022/2023 (2nd edition)

Prof. Francesco Finazzi francesco.finazzi@unibg.it
Dott. Frank Massoda frank.massodatchoussi@unibg.it

Chapter 9

Support Vector Machine

Introduction

- The support vector machine is a generalization of a simple and intuitive classifier called the **maximal margin classifier**
- We will see that this classifier cannot be applied to most data sets, since it requires that the classes be separable by a linear boundary
- We will then introduce the support vector classifier which is an extension of the maximal margin classifier that can be applied in a broader range of cases
- **Support vector machine** is a further extension of the support vector classifier in order to accommodate non-linear class boundaries

What is a hyperplane?

- In a p-dimensional space, a hyperplane is a flat subspace of hyperplane dimension p − 1
- In two dimensions, a hyperplane is a flat one-dimensional subspace (a line).
- In three dimensions, a hyperplane is a flat two-dimensional subspace (a plane).
- In p > 3 dimensions, it is hard to visualize a hyperplane but the notion of a (p − 1)-dimensional flat subspace still applies

What is a hyperplane?

- In two dimensions a hyperplane is defined by the equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \tag{9.1}$$

- Equation 9.1 can be easily extended to the p-dimensional setting:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{9.2}$$

What is a hyperplane?

- Suppose that X does not satisfy (9.2) but rather:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0. \tag{9.3}$$

- This tells us that X lies to one side of the hyperplane. On the other hand

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0, \tag{9.4}$$

- We can think of the hyperplane as dividing the p-dimensional space into two sub-spaces

What is a hyperplane?

- Suppose that X does not satisfy (9.2) but rather:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0. \tag{9.3}$$

- This tells us that X lies to one side of the hyperplane. On the other hand

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0, \tag{9.4}$$

- We can think of the hyperplane as dividing the p-dimensional space into two sub-spaces

What is a hyperplane? Example

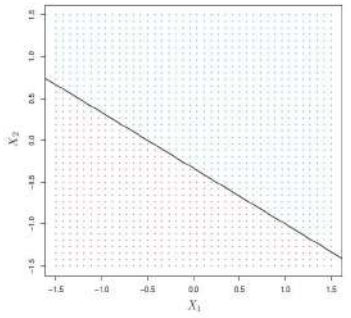


FIGURE 9.1. *The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.*

Classification using a separating hyperplane

- Suppose that we have a n×p data matrix X that consists of n training observations in p-dimensional space

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}, \tag{9.5}$$

- and that these observations fall into two classes {-1,1}
- Our goal is to develop a classifier based on the training data that will correctly classify the test observations using their feature measurements

Classification using a separating hyperplane

- Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels (namely y is -1 or 1)
- Then a separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 \text{ if } y_i = 1, \tag{9.6}$$

and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 \text{ if } y_i = -1. \tag{9.7}$$

Classification using a separating hyperplane

- If a separating hyperplane exists, we can use it to construct a very natural classifier: a test observation is assigned a class depending on which side of the hyperplane it is located

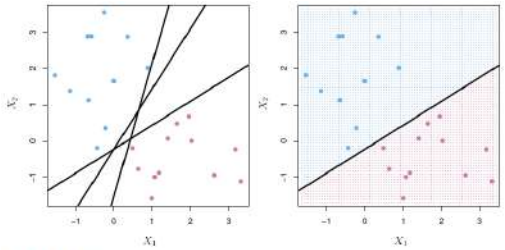


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

The Maximal Margin Classifier

- If our data can be perfectly separated using a hyperplane, then there will exist an infinite number of such hyperplanes
- In fact, a given separating hyperplane can usually be shifted a tiny bit up or down, or rotated, without coming into contact with any of the observations
- In order to construct a classifier based upon a separating hyperplane, we must have a reasonable way to decide which of the infinite possible separating hyperplanes to use
- A natural choice is the maximal margin hyperplane which is the separating hyperplane that is farthest from the training observations

The Maximal Margin Classifier

- We can compute the (perpendicular) distance from each training observation to a given separating hyperplane; the smallest such distance is the minimal distance from the observations to the hyperplane, and is known as the margin
- The maximal margin hyperplane is the separating hyperplane for which the margin is largest
- We can then classify a test observation based on which side of the maximal margin hyperplane it lies
- We hope that a classifier that has a **large maximal margin on the training data** will also have a **large margin on the test data** (and hence will classify the test observations correctly)

The Maximal Margin Classifier: Example

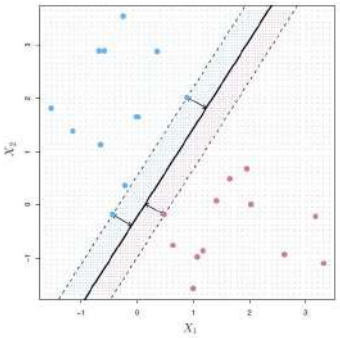


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the hyperplane is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

The Maximal Margin Classifier

- Examining Figure 9.3, we see that three training observations are equidistant from the maximal margin hyperplane and lie along the dashed lines indicating the width of the margin
- These three observations are known as **support vectors**, since they are vectors in p-dimensional space and they "support" the maximal margin hyperplane (in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well)

Construction of the Maximal Margin Classifier

- The maximal margin hyperplane is the solution to the optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \; M \tag{9.9}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n. \tag{9.11}$$

- M represents the margin of the hyperplane, and the optimization problem chooses the betas to maximize M

The Non-separable Case

- The maximal margin classifier is a very natural way to perform classification, **if a separating hyperplane exists**
- In many cases no separating hyperplane exists, and so there is no maximal margin classifier. In this case, the optimization problem (9.9)–(9.11) has no solution with M >0
- We can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called soft margin.
- The generalization of the maximal margin classifier to the non-separable case is known as the **support vector classifier**

The Non-separable Case

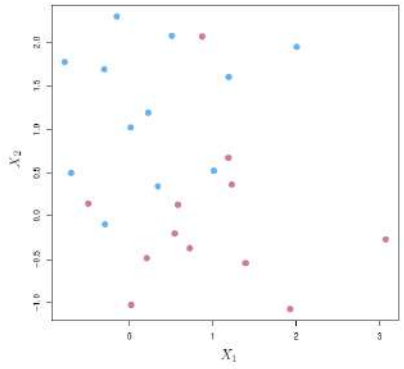


FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Support Vector Classifiers

- Even if a separating hyperplane exists, there are instances in which a classifier based on a separating hyperplane might not be desirable
- The maximal margin hyperplane is extremely sensitive to a change in single observations (suggesting that it may overfit the training data)
- We might be willing to consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of
  - Greater robustness to individual observations, and
  - Better classification of most of the training observations

Support Vector Classifiers

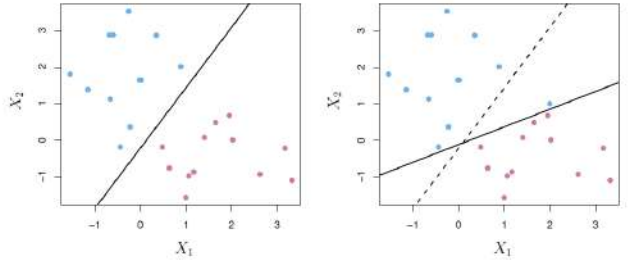


FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.*

Support Vector Classifiers

- Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, the support vector classifier allows some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.
- The hyperplane is chosen to correctly separate most of the training observations into the two classes, but may misclassify a few observations

Support Vector Classifiers

- It is the solution to the optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M \tag{9.12}$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.13}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{9.14}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \tag{9.15}$$

C is a non-negative tuning parameter and the ϵ_i are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane

Slack variables

- The slack variable ϵ_i tells where the i-th observation is located relative to the hyperplane and relative to the margin.
- If $\epsilon_i = 0$, then the i-th observation is on the correct side of the margin. If $\epsilon_i > 0$, then the i-th observation is on the wrong side of the margin. If $\epsilon_i > 1$, then it is on the wrong side of the hyperplane.
- C bounds the sum of the ϵ_i's, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate
- For C > 0 no more than C observations can be on the wrong side of the hyperplane, because if an observation is on the wrong side of the hyperplane then $\epsilon_i > 1$.

Support Vector Classifiers: example

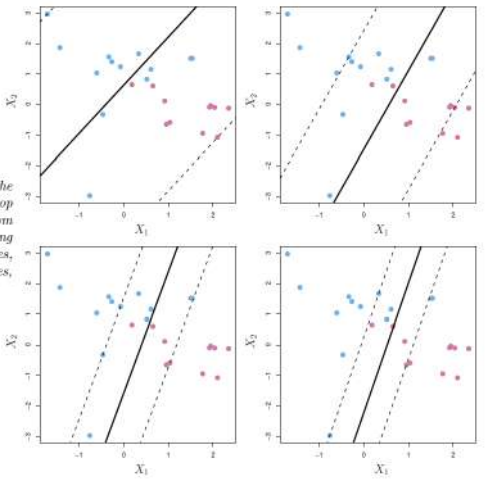


FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.*

Support Vector Machines

- The support vector classifier is a natural approach for classification in the two-class setting, **if the boundary between the two classes is linear**

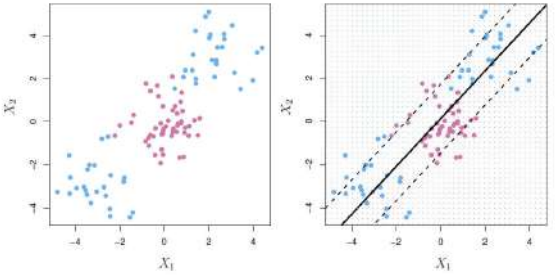


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Support Vector Machines

- Similarly to what we do in regression, we **could** address the problem of possibly non-linear boundaries between classes by enlarging the feature space using quadratic, cubic, etc. functions of the predictors. The optimization becomes:

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M \tag{9.16}$$

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

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

Support Vector Machines

- It is not hard to see that there are many possible ways to enlarge the feature space, and that unless we are careful, we could end up with a huge number of features
- Then computations would become unmanageable
- The support vector machine allows to **enlarge the feature space** used by the support vector classifier in a way that leads to **efficient computations**
- Unfortunately, the SVM technicality is quite complex (inner product generalization and kernels)
- SVM proposes an efficient computational approach for enlarging the feature space

Support Vector Machines: example

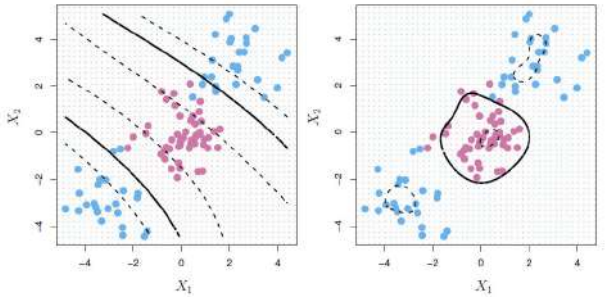


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione