

5.0 - Statistical Learning

5.1 - Basic concepts

“Machine learning” ?

- Essentially, it is statistical learning
- Essentially, we're looking for a pattern (unseen up to now)
- If there is no pattern, then ML will be counter-productive as it is likely to produce one!
- Assumes availability of relevant data
- Examples: consumer taste/habits, online advertising, election forecasts, risk prediction
- Warning: linear regression is now referred to as “artificial intelligence” by a lot of people

Statistical learning: regression

- Variable of interest: **Wage** (continuous)

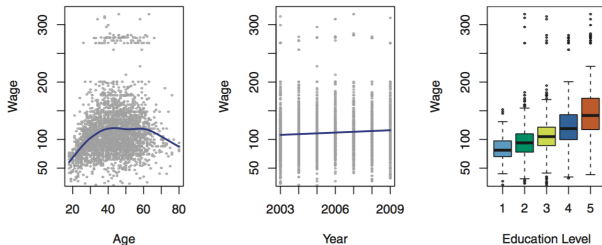


FIGURE 1.1. *Wage* data, which contains income survey information for males from the central Atlantic region of the United States. Left: *wage* as a function of *age*. On average, *wage* increases with *age* until about 60 years of age, at which point it begins to decline. Center: *wage* as a function of *year*. There is a slow but steady increase of approximately \$10,000 in the average *wage* between 2003 and 2009. Right: Boxplots displaying *wage* as a function of *education*, with 1 indicating the lowest level (no high school diploma) and 5 the highest level (an advanced graduate degree). On average, *wage* increases with the level of education.

Statistical learning: classification

- Variable of interest: **Default** (categorical)

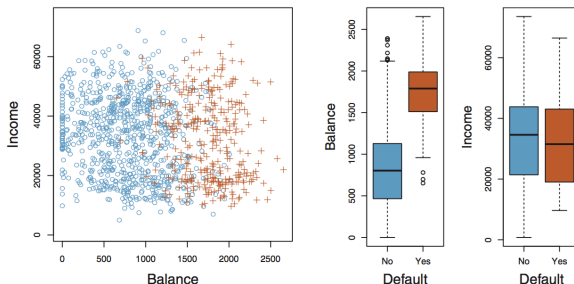
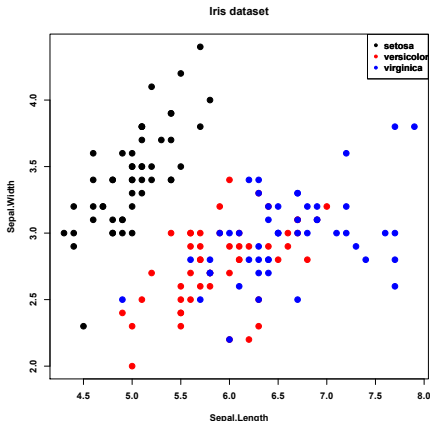


FIGURE 4.1. *The Default data set. Left: The annual incomes and monthly credit card balances of a number of individuals. The individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue. Center: Boxplots of balance as a function of default status.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

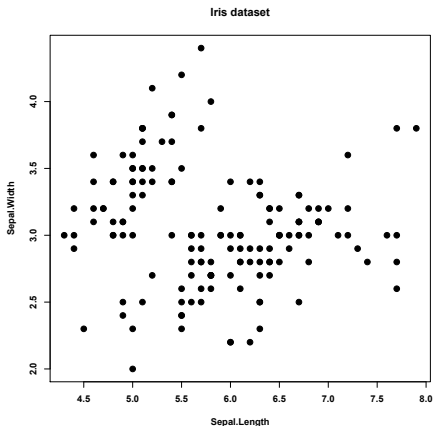
Statistical learning: supervised learning

- Variable of interest: **Species** (categorical)
- With prior experience available (**supervised classification**)



Statistical learning: unsupervised learning

- Variable of interest: **Species** (categorical)
- With no prior experience available (**clustering**)



5.2 - Learning framework

Data splitting

- A dataset is randomly split into a *training set* and a *test set*
- The training set is used to fit (or train) the model
- The test set is used to validate this model
- Model is both tuned and fitted during training
- We expect $MSE_{train} < MSE_{test}$
- **Overfitting** occurs when too much emphasis is put on training the data:
 - Training set yields a *much* smaller MSE than test set
 - Model is describing the training data too well and unable to adapt to new data
 - Yields poor prediction performance

General framework with large data

Example in model selection:

- Random split into training, validation and test sets (e.g. 50%, 25%, 25% resp.)
- Training set: used to fit the model
- Validation set: used to measure prediction error and choose best model
- Test set: used to measure generalization error of final model (i.e. ability to predict from new data)

[*The Elements of Statistical Learning*, T. Hastie, R. Tibshirani, J. Friedman, Springer]

General framework with small data

Cross-validation is usually used when dealing with small samples.

Example: model selection for classification with $N = 50, p = 5000$

- 1 Randomly divide samples into K cross-validation folds
- 2 For each fold $k = 1, 2, \dots, K$:
 - a Find a subset of predictors with higher (univariate) correlation with class labels, using all data except fold k
 - b Using just this subset of predictors, build a multivariate classifier, using all data except fold k
 - c Use the classifier to predict the class labels for fold k and compute corresponding prediction error

[*The Elements of Statistical Learning*, T. Hastie, R. Tibshirani, J. Friedman, Springer]

General framework: summary

- Randomly split training and test set
- Training set:
 - Model calibration (tuning)
 - Fit model on whole training set
- Test set:
 - Model validation (test)

Key aspects of the data (typical challenges)

- Data scales
 - Beware of scale effects in heterogeneous data
 - Illustrative example: iris dataset
- Dimensionality (too many covariates)
 - Many variables may be “aligned” / redundant
 - Data pre-filtering: must be done independently of class labels
 - Illustrative example: iris dataset
- Dimensionality (too many dimensions in the observed data)
 - Pre-process data using dimension reduction techniques
 - Factor analysis, PCA are predominant and common choices
 - Illustrative examples: iris and EuStockMarkets datasets

5.3 - Performance assessment

Performance indicators for regression

- Mean Square Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{f}(X_i))^2$$

- LOO CV test MSE:

$$MSE_{(n)} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{f}^{-i}(X_i))^2$$

- k-fold CV test MSE:

$$MSE_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i^{-i}$$

Performance indicators for classification

- Prediction accuracy (error rate)

$$Err = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq \hat{Y}_i)$$

- LOO CV error rate:

$$Err_{(n)} = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq \hat{Y}_i^{-i})$$

- k-fold CV test MSE:

$$Err_{(k)} = \frac{1}{k} \sum_{i=1}^k Err_i^{-i}$$

Performance indicators for classification

Recall: one seeks to retain or reject a null hypothesis H_0 on the basis of evidence. Let us denote H_1 the alternative hypothesis.

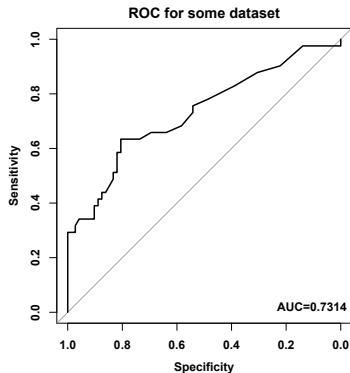
	H_0 is true	H_1 is true
H_0 is accepted	Correct decision	Type II error
H_1 is accepted	Type I error	Correct decision

- The Null hypothesis *can never be proven*
- Type I error occurs when H_0 is true but rejected
- $P(\text{Type I error}) = \text{significance level}$ of the test
- $P(\text{Type II error}) = \text{false negative rate}$
- $1 - P(\text{Type II error}) = \text{statistical power}$ of the test (*sensitivity*)

Performance indicators for classification

- False Positive rate = FP/N = specificity (I)
- True Positive rate = TP/N = sensitivity = recall ($1-II$)
- AUC of the ROC

		Prediction outcome	
		+	-
Actual value	+	TP	FN
	-	FP	TN
TOTAL		P	N



5.4 - Some techniques of reference

Logistic regression

- Use a scrambled subsample x of the iris dataset:
`is = sample(1:150); x = iris[is[1:100],]`
- Recode `x$Species` into `is.virginica` with values in $(0; 1)$
(we're changing the problem formulation slightly)
- Fit model:
`fit <- glm(Species~., data=x,
family=binomial(logit))`
- Use fit to predict Species of remaining data points:
`testset = iris[-is,]
y <- testset[,1:4]
pred <- predict(fit, newdata=y, type='response')`
- Assess prediction performance

Naive Bayes classification

- Recall: find unknown true label l_i for each observation Y_i , given the observed predictor vector x_0
- Naive Bayes Classifier: for all $i = 1, \dots, n$

$$\hat{l}_i = \arg \max_j Pr(Y_i = j | X_i = x_0)$$

- Example: for a two-class problem,

$$\hat{l}_i = 1 \text{ if } Pr(Y = 1 | X = x_0) > 0.5$$

Regression and Classification Trees

Ex: ISLR::Hitters (<http://www-bcf.usc.edu/~gareth/ISL>)

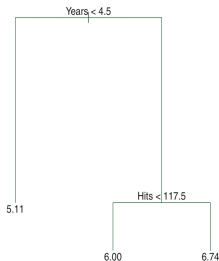
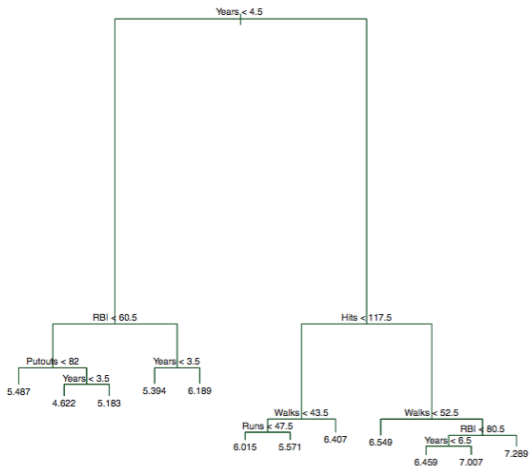


FIGURE 8.1. For the `Hitters` data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year. At a given internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to `Years < 4.5`, and the right-hand branch corresponds to `Years >= 4.5`. The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.

Regression and Classification Trees

Ex: ISLR::Hitters



Clustering

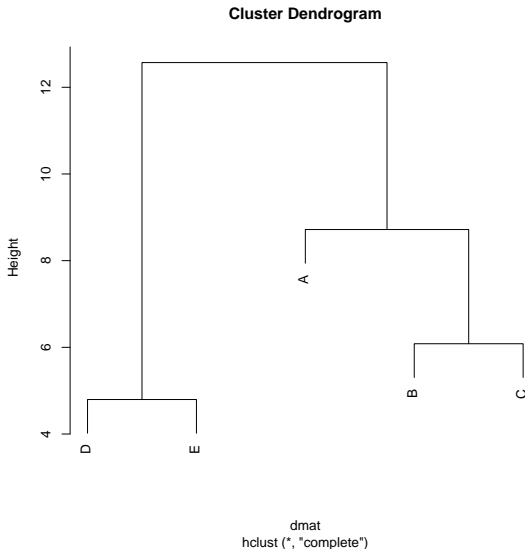
- Idea: arrange n individuals into groups wrt a set of measures
- The choice of measure is key and determines classification
- Variables should be rescaled first (and weighted)
- Highly dimensional data may require prior reducing (PCA not necessarily pertinent here)
- <http://cran.r-project.org/web/views/Cluster.html>

Hierarchical clustering

Hierarchical clustering: split-and-merge to construct a dendrogram

```
M = matrix(c(0,3,5,8,4,  
             3,0,2,6,8,  
             5,2,0,3,4,  
             8,6,3,0,1,  
             4,8,4,1,0),nrow=5)  
dM = data.frame(M, row.names=c("A","B","C","D","E"))  
dmat = dist(dM)  
plclust( hclust(dmat) )
```

Hierarchical clustering



Hierarchical clustering

- Example: `data(eurodist)`
- Creating a dendrogram:
`hc = hclust(eurodist, method="ward")`
- Plotting dendrograms:
`plot(hc) # (or plclust)`
`plot(hc, hang=-1)`
`rect.hclust(hc, k=3)`

k -means clustering

- k -means clustering is another popular clustering method
- It is comparable to an Expectation-Maximisation algorithm
- ?kmeans...
- Initialize at k clusters (use hierarchical to choose k ?)
- Move an individual to another cluster if criterion is optimized
- Risk of convergence to local solution

Principal component analysis

- Project data according to highest variance components
- Linear orthogonal transformation:
each component is uncorrelated with preceding ones
- Reveals internal structure of the data, explaining its variance
- PC1 = greatest variance by any projection of the data
- Theoretically optimal transform for a given data in LS terms
- PCA is widely used e.g. to reduce problem dimensionality
- ?prcomp...

Principal component analysis

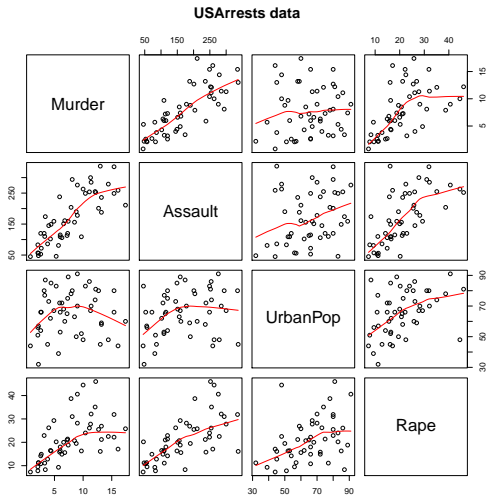
Example: ?USArrests

- Statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973
- Also contains the % of the population living in urban areas

```
plot(USArrests, main="USArrests data") # scatterplots
```

```
pairs(USArrests, panel = panel.smooth,  
      main = "USArrests data")
```

Principal component analysis



Principal component analysis

```
data(USArrests)

cor(x) # correlation matrix
eigen(cor(x)) # eigenvalue decomposition

# compare result with prcomp:
prcomp(USArrests, scale = TRUE)    # same vectors!

prcomp(~Murder+Assault+Rape, data=USArrests, scale=T)
plot(prcomp(USArrests, scale = TRUE))
# equiv.  cov(prcomp(USArrests, scale = TRUE)$x)
# i.e.   (prcomp(USArrests, scale = TRUE)$sdev)^2,
# i.e.   the eigenvalues of the cov/correl matrix

summary(prcomp(USArrests, scale = TRUE))
```

Principal component analysis

