

Predictive Analytics Part 1: Clustering, Classification, Association & Frequent Itemset Mining

0.1 Cluster Analysis in R

For the following exercise, load the *BigMac2003*¹ data set from the *alr4*-package. This data set contains for 69 cities world-wide the average working hours, price level and income level for the year 1991. Your task is to perform a cluster analysis of the cities based on selected variables. This data set contains the following variables (to list them, you might use, e.g., `attributes(BigMac2003)`):

BigMac Minutes of labor to purchase a Big Mac

Bread Minutes of labor to purchase 1 kg of bread

Rice Minutes of labor to purchase 1 kg of rice

FoodIndex Food price index (Zurich=100)

Bus Cost in US dollars for a one-way 10 km ticket

Apt Normal rent (US dollars) of a 3 room apartment

TeachGI Primary teacher's gross income, 1000s of US dollars

TeachNI Primary teacher's net income, 1000s of US dollars

TaxRate Tax rate paid by a primary teacher

TeachHours Primary teacher's hours of work per week

For the class label, generate vector with the cities' regions (aggregated to 4 classes) as follows:

```
region=as.factor(c("EU","EU","AUNZ","AS","EU","EU","EU","SA","EU","EU","EU","EU","SA","SA","NA",
  "EU","AF","EU","EU","EU","EU","AS","AF","AS","AF","AS","EU","AS","AF","SA","EU","EU","EU","NA",
  "EU","EU","EU","AF","AS","NA","NA","EU","NA","EU","AS","AF","NA","EU","EU","EU","EU","SA","EU",
  "SA","SA","AS","AS","AS","EU","EU","AUNZ","AS","EU","AF","AS","NA","EU","EU","EU"))
region4 = region;
levels(region4) = c("AF","AS","N","N","N","SA")
```

Now perform the following steps:

1. Select the variables *FoodIndex*, *Bus*, *Apt*, *TeachNI* and *TeachHours* and scale them (using the function `scale()`) to an arithmetic mean of 0 and a standard deviation of 1.

```
bm<-data.frame(FoodIndex=scale(BigMac2003$FoodIndex),Bus=scale(BigMac2003$Bus),
  Apt=scale(BigMac2003$Apt),TeachNI=scale(BigMac2003$TeachNI),
  TeachHours=scale(BigMac2003$TeachHours),row.names=row.names(BigMac2003))
summary(bm$FoodIndex)
```

2. Compute the distance matrices for distances based on $(1 - |\text{correlation}|)$, L_1 (manhattan) and L_2 (euclidean) norm:

```
dcor<-(1-abs(cor(t(bm[,]),method="pearson"))))
dcor<-as.dist(1-abs(cor(t(bm[,]),method="pearson"))) # correlation transformed to distance in [0;1]
d1<-dist(bm, method = "minkowski", diag = FALSE, upper = FALSE, p = 1) # same as "manhattan"
d2<-dist(bm, method = "minkowski", diag = FALSE, upper = FALSE, p = 2) # same as "euclidean"
dim(as.matrix(d1)) # show the number of rows and columns
```

¹Provided by the Union Bank of Switzerland.

3. Compute a divisive hierarchical clustering using the R function `diana()` from the package *cluster* and analyse the results by studying the resulting dendrograms with the `plot()` function.

```
library(cluster)
dg_cor<-diana(x=dcor,diss=TRUE)
plot(dg_cor)
dg_d1<-diana(x=d1,diss=TRUE)
plot(dg_d1)
dg_d2<-diana(x=d2,diss=TRUE)
plot(dg_d2)
\end{enumerate}
```

Comparing the dendrograms, it seems that using euclidean (or manhattan) distance produces more geographically related clusters than correlation-based similarity.

4. Compute an agglomerative hierarchical clustering using the R function `agnes()` from the package *cluster* and analyse the results by studying the resulting dendrograms with the `plot()` function.

```
library(cluster)
ag_single<-agnes(x=dcor,diss=TRUE,method="single")
# alt. method param. values: "single", "complete", "weighted", "ward"
plot(ag_single)
ag_complete<-agnes(x=dcor,diss=TRUE,method="complete")
plot(ag_complete)

ag_single<-agnes(x=d1,diss=TRUE,method="single")
plot(ag_single)
ag_complete<-agnes(x=d1,diss=TRUE,method="complete")
plot(ag_complete)
```

Single linkage with euclidean/manhattan distance produces geographically reasonably clustered results.

5. Using only the attributes *FoodIndex* and *Apt*, compute a k-means based clustering using the R function `kmeans()` for $k = 3$, $k = 4$, and $k = 5$, and analyse the results in a 2-dimensional plot of the clusters and their points. Using the functions `intCriteria()` and `extCriteria` from the library *clusterCrit*, compute the Silhouette Index as internal measure for all clustering. For $k = 4$, compute also the Rand Index as external measure (against the class label vector *region4*).

```
nclusters=4 # 3 4 5
km=kmeans(bm[,c("FoodIndex","Apt")],centers=nclusters)
par(new=F); plot.new()
xlim_ = c(-2.5,2.5);
ylim_ = c(-2.5,2.5);
for (k in sort(unique(km$cluster))) {
  plot(x=bm[km$cluster==k,c("FoodIndex","Apt")],type="p",
       col=rainbow(length(unique(km$cluster)))[k],xlim=xlim_,ylim=ylim_); par(new=T);
  text(x=km$centers[k,1],y=km$centers[k,2],labels=paste("Cluster ",k),
       col=rainbow(length(unique(km$cluster)))[k]); par(new=T);
}
library("clusterCrit")
km_int <- intCriteria(traj=as.matrix(bm[,c("FoodIndex","Apt")]),km$cluster,"silhouette")
if (nclusters==4) { km_ext <- extCriteria(km$cluster,as.integer(region4),"Rand") }
```

6. As above, use the attributes *FoodIndex* and *Apt*, but now compute an EM-based clustering using the R functions `init.EM()` and `emcluster()` for $n_{class} = 3$ and $n_{class} = 5$ from the library *EMCluster*, and analyse the results in a 2-dimensional plot of the clusters and their points. As above, compute the Silhouette and Rand Indices.

```
library("EMCluster")
nclusters=4 # 3 4 5
em=init.EM(x=bm[,c("FoodIndex","Apt")],nclass=nclusters, method="em.EM")
em=emcluster(x=bm[,c("FoodIndex","Apt")],em,assign.class=TRUE)
par(new=F); plot.new()
xlim_ = c(-2.5,2.5);
ylim_ = c(-2.5,2.5);
for (k in sort(unique(em$class))) {
  plot(x=bm[em$class==k,c("FoodIndex","Apt")],type="p",
       col=rainbow(length(unique(em$class)))[k],xlim=xlim_,ylim=ylim_); par(new=T);
  text(x=em$Mu[k,1],y=em$Mu[k,2],labels=paste("Cluster ",k),
       col=rainbow(length(unique(em$class)))[k],xlim=xlim_,ylim=ylim_); par(new=T);
}
library("clusterCrit")
em_int <- intCriteria(traj=as.matrix(bm[,c("FoodIndex","Apt")]),as.integer(em$class),"silhouette")
if (nclusters==4) { em_ext <- extCriteria(as.integer(em$class),as.integer(region4),"Rand") }
```

7. Generate a knee-plot for determining the optimal number of clusters in your kmeans or EM-clusterings based on the Silhouette Index.

```
km_int=matrix(nrow=10,ncol=100)
km_avint=vector(length=10)
for (nclusters in 1:10){
  for (repIdx in 1:100){
    km=kmeans(bm[,c("FoodIndex","Apt")],centers=nclusters)
    km_int[nclusters,repIdx] <- as.numeric(intCriteria(traj=as.matrix(bm[,c("FoodIndex","Apt")]),
    km$cluster,"silhouette"))
  }
  km_avint[nclusters] = mean(km_int[nclusters,],na.rm=TRUE)
}
par(new=F); plot.new()
plot(x=c(1:10),y=km_avint,xlab="nClusters",ylab="Silhouette",type="l")
```

This should show “knees” at $k = 2$ and $k = 4$.

0.2 How would you distinguish between flat and hierarchical partitioning?

hierarchy: there exists a hierarchy between the instances based on their proximity

partitioning: clusters are simple partitions (no hierarchy between instances)

0.3 What is probabilistic clustering?

A soft clustering that provides membership probabilities for instance:cluster pairs that indicate the instance's probability to belong to the cluster.

0.4 What is the difference between soft and hard partitioning?

hard: every instance belongs to one and only one cluster

soft: an instance belongs to all clusters with varying membership probability

0.5 What are the two main approaches of flat partitioning?

Shape-based and density-based

0.6 What are the two main approaches of hierarchical partitioning?

Agglomerative and divisive

0.7 Discuss the Minkowski-Distances for $r = 1$ and $r = 2$, their differences and their alternative names.

- $r = 1$ L1-Norm, City-Block-Metrik, Simple addition of the absolute distances (on each dimension) between the instances
- $r = 2$ L2-Norm, Euclidean Distance, Basically least squares, Minimizing the sum of the square of the differences between the target value and the estimated values

0.8 On which of the following datasets would you use (1) K-Means Clustering (or EM), (2) single-linkage agglomerative hierarchical clustering, (3) density-based clustering? Provide arguments for your recommendation.

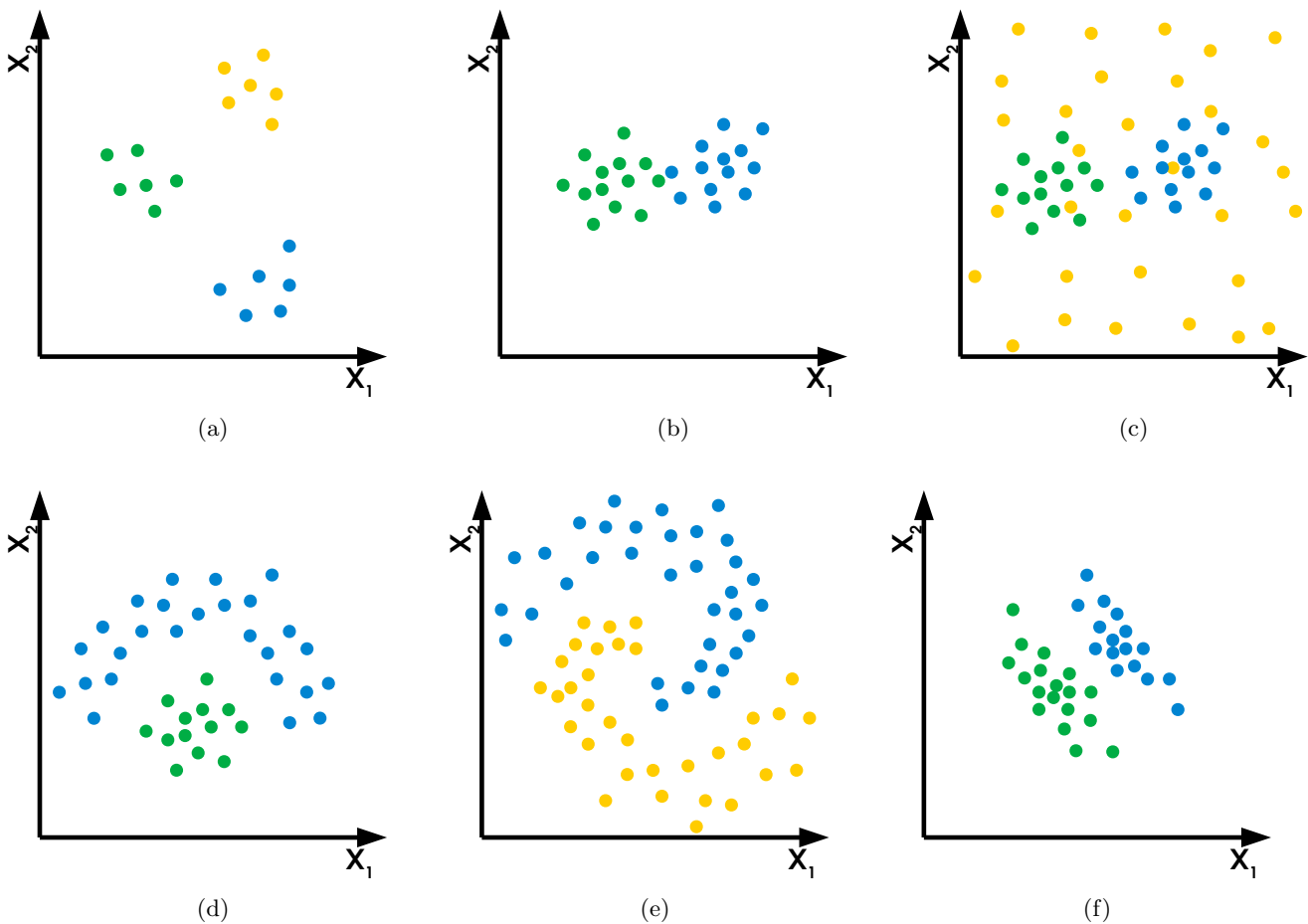


Figure 1: Scatter plots of different data sets.

- a 1 (spherical), 3 (well-separated)
- b 1 (spherical), not 2 or 3 (because clusters are overlapping)
- c 1 (EM with mixture of two Gaussians and a uniform)
- d 2 or 3
- e 2 or 3
- e 2 or 3, maybe EM (with large covariance)

0.9 Which clustering(s) tend(s) to form clusters that are chains of points?

- ☒ Single Linkage Agglomerative Hierarchical Clustering
- ☐ Complete Linkage Agglomerative Hierarchical Clustering
- ☐ K-Means Clustering
- ☐ Expectation Maximisation with Mixture of Gaussians

0.10 Which clustering(s) tend(s) to form spherical clusters?

- ☐ Single Linkage Agglomerative Hierarchical Clustering
- ☒ K-Means Clustering
- ☒ Expectation Maximisation with Mixture of Gaussians

0.11 Which of the following similarity/distance measures is appropriate for comparing instances based on their relative attribute values?

- ☐ Euclidean Distance
- ☐ L1 norm or City-Block Metric
- ☐ Mahalanobis distance
- ☒ Q-Correlation Coefficient

0.12 Which of the following similarity/distance measures is appropriate for comparing instances based on the absolute distance on their attribute values?

- ☒ Euclidean Distance
- ☒ L1 norm or City-Block Metric
- ☒ Mahalanobis distance
- ☐ Q-Correlation Coefficient

0.13 Which of the following statements are correct?

- ☒ (X) The Silhouette Index is an internal cluster quality measure.
- ☐ () Internal cluster quality measures compare the cluster quality based on ground-truth class labels.
- ☒ (X) External cluster quality measures compare the cluster quality based on ground-truth class labels.
- ☒ (X) Clustering aims to maximise similarity between instances within clusters and dissimilarity between instances in different clusters.
- ☒ (X) The Silhouette Index considers the intra-cluster and inter-cluster distances.

0.14 You want to compare two clusterings, one with three and one with five clusters. Which external performance measures can you use?

- ☐ () Purity
- ☒ (X) Normalised Mutual Information

0.15 Based on the elbow method, determine in the plots in Fig. 2 below which number of clusters k to chose.

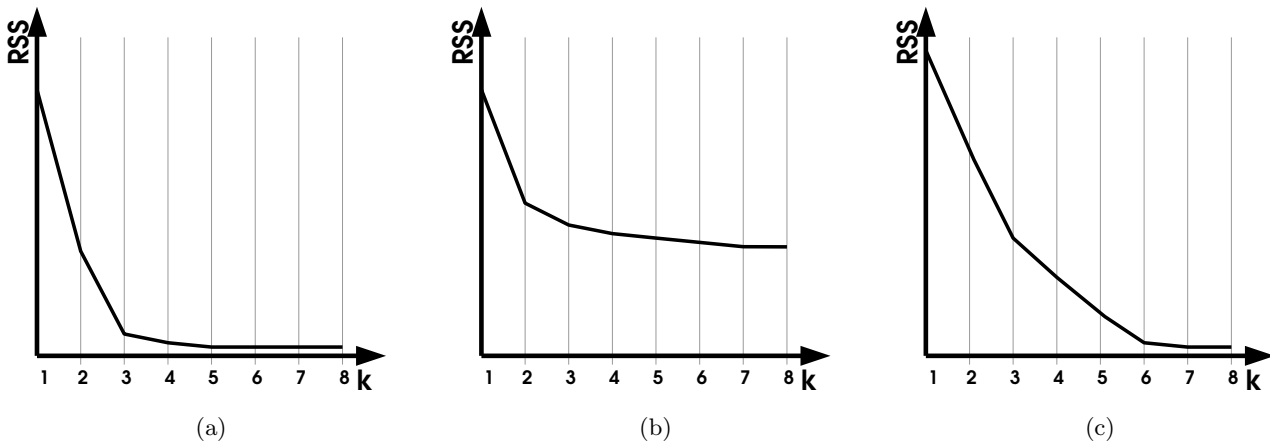


Figure 2: Elbow plots for three different data sets.

- (a) $k = 3$, (b) $k = 2$, (c) $k = 3$ or $k = 6$ (both reasonable)

0.16 Which of the following statements are correct?

- (X) In classification, the task is to predict a dependent variable (class label) based on a set of explanatory (feature) variables.
- (X) A probabilistic classifier returns also estimates of the posterior class probability $\Pr(y|x)$.
- (X) A lazy learner does not construct an (abstract) model from the data.
- (X) A Bayes classifier requires an estimate of either the joint probability $\Pr(x, y)$, or estimates of the class prior probability $\Pr(y)$ together with the class-conditional feature probability $\Pr(x|y)$.
- () The (unconditional) feature probability $\Pr(x)$ can be computed by summing (marginalising) $\Pr(x, y)$ over the different values of x .
- (X) The (unconditional) feature probability $\Pr(x)$ can be computed by summing (marginalising) $\Pr(x, y)$ over the different values of y .
- (X) In contrast to a multivariate Bayes classifier, a Naive Bayes classifier assumes conditional independence.
- () A predictive classifier also provides a model of the underlying data distribution, which can be used to generate data.
- (X) For k-fold cross validation, the data set is partitioned into k subsets, and each of them is used once for testing and the other times for training.
- () For obtaining a reliable estimate of the performance on new data, we need to evaluate a classifier on its *training* data.
- () False positives are instances that are classified as negative, but actually are positive.
- (X) In a region, where we classify all instances as positive, our false negative rate is zero.

0.17 Curse of Dimensionality: If you have three feature variables X_1, X_2 , and X_3 , which you divide each into two intervals (low vs. high values), in how many cells does this split your feature space? Alternatively, how many training instances do you need to place exactly one in each cell? How does this change with four features X_1, X_2, X_3 and X_4 ?

For three features, we have $2^3 = 8$ cells (or instances). For four, we have $2^4 = 16$ cells. In general, for D features we have 2^D cells.

0.18 In Fig. 3, the scatterplots of two data sets (a) and (b) are given. In which of these two is conditional independence assumption clearly violated?

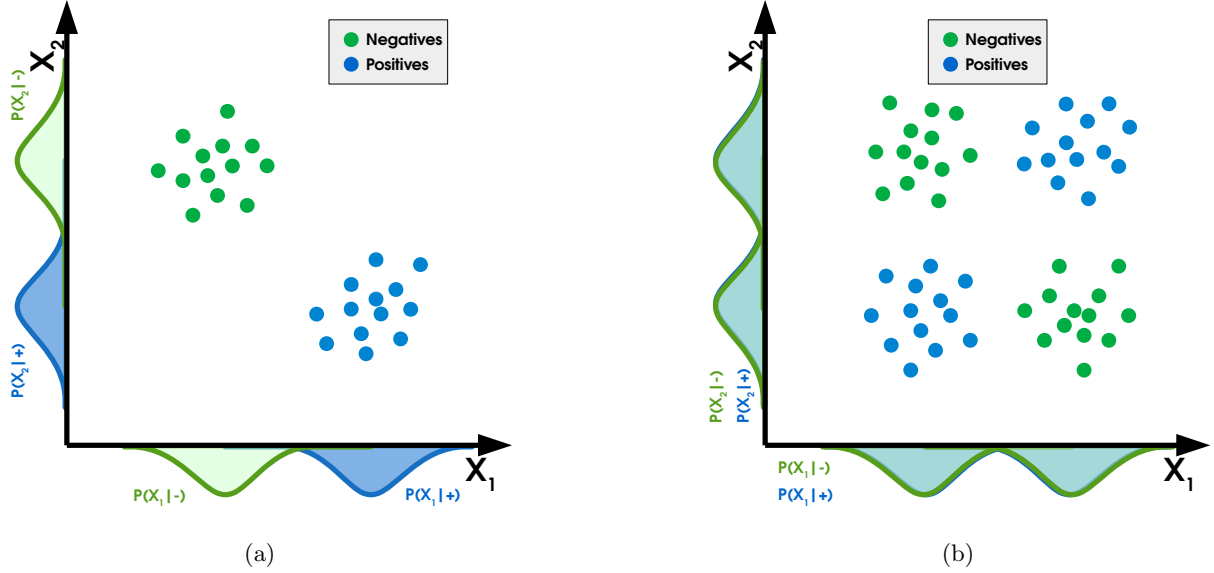


Figure 3: Scatterplots of two datasets.

In (b), because there $\Pr(X_i, X_j|Y) = \Pr(X_i|Y) \cdot \Pr(X_j|Y)$ does not hold.

Hint: Imagine X_1 and X_2 each being discretised into two attribute values *low* and *high*, and then calculate the according probabilities.

For (a), these are equal:

$$\begin{aligned}\Pr(X_1 = l, X_2 = l|Y = +) &= 0 \\ \Pr(X_1 = l, X_2 = h|Y = +) &= 0 \\ \Pr(X_1 = h, X_2 = l|Y = +) &= 1 \\ \Pr(X_1 = h, X_2 = h|Y = +) &= 0 \\ \Pr(X_1 = l, X_2 = l|Y = -) &= 0 \\ \Pr(X_1 = l, X_2 = h|Y = -) &= 1 \\ \Pr(X_1 = h, X_2 = l|Y = -) &= 0 \\ \Pr(X_1 = h, X_2 = h|Y = -) &= 0\end{aligned}$$

$$\begin{aligned}\Pr(X_1 = l|Y = +) \cdot \Pr(X_2 = l|Y = +) &= 0 \cdot 1 = 0 \\ \Pr(X_1 = l|Y = +) \cdot \Pr(X_2 = h|Y = +) &= 0 \cdot 0 = 0 \\ \Pr(X_1 = h|Y = +) \cdot \Pr(X_2 = l|Y = +) &= 1 \cdot 1 = 1 \\ \Pr(X_1 = h|Y = +) \cdot \Pr(X_2 = h|Y = +) &= 1 \cdot 0 = 0 \\ \Pr(X_1 = l|Y = -) \cdot \Pr(X_2 = l|Y = -) &= 1 \cdot 0 = 0 \\ \Pr(X_1 = l|Y = -) \cdot \Pr(X_2 = h|Y = -) &= 1 \cdot 1 = 1 \\ \Pr(X_1 = h|Y = -) \cdot \Pr(X_2 = l|Y = -) &= 0 \cdot 0 = 0 \\ \Pr(X_1 = h|Y = -) \cdot \Pr(X_2 = h|Y = -) &= 0 \cdot 1 = 0\end{aligned}$$

For (b), these are not:

$$\begin{aligned}\Pr(X_1 = l, X_2 = l|Y = +) &= 1/2 \\ \Pr(X_1 = l, X_2 = h|Y = +) &= 0 \\ \Pr(X_1 = h, X_2 = l|Y = +) &= 0 \\ \Pr(X_1 = h, X_2 = h|Y = +) &= 1/2 \\ \Pr(X_1 = l, X_2 = l|Y = -) &= 0 \\ \Pr(X_1 = l, X_2 = h|Y = -) &= 1/2 \\ \Pr(X_1 = h, X_2 = l|Y = -) &= 1/2 \\ \Pr(X_1 = h, X_2 = h|Y = -) &= 0\end{aligned}$$

$$\begin{aligned}\Pr(X_1 = l|Y = +) \cdot \Pr(X_2 = l|Y = +) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = l|Y = +) \cdot \Pr(X_2 = h|Y = +) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = h|Y = +) \cdot \Pr(X_2 = l|Y = +) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = h|Y = +) \cdot \Pr(X_2 = h|Y = +) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = l|Y = -) \cdot \Pr(X_2 = l|Y = -) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = l|Y = -) \cdot \Pr(X_2 = h|Y = -) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = h|Y = -) \cdot \Pr(X_2 = l|Y = -) &= 1/2 \cdot 1/2 = 1/4 \\ \Pr(X_1 = h|Y = -) \cdot \Pr(X_2 = h|Y = -) &= 1/2 \cdot 1/2 = 1/4\end{aligned}$$

Note that for the alternative formula $\Pr(X_i|X_j, Y) = \Pr(X_i|Y)$, the conditional probabilities of the left side $\Pr(X_i|X_j, Y)$ involves for some X_j, Y probabilities $P(X_j, Y)$ that are zero, and is therefore not useful here.

0.19 Based on the AUC/ROC plotted in Fig. 4, which of the following statement(s) is/are correct?

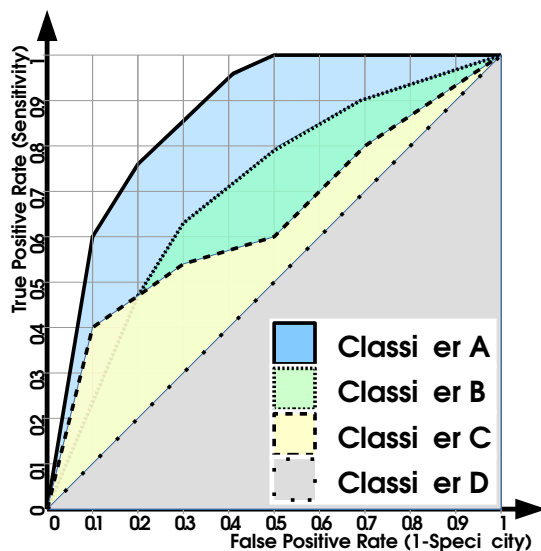
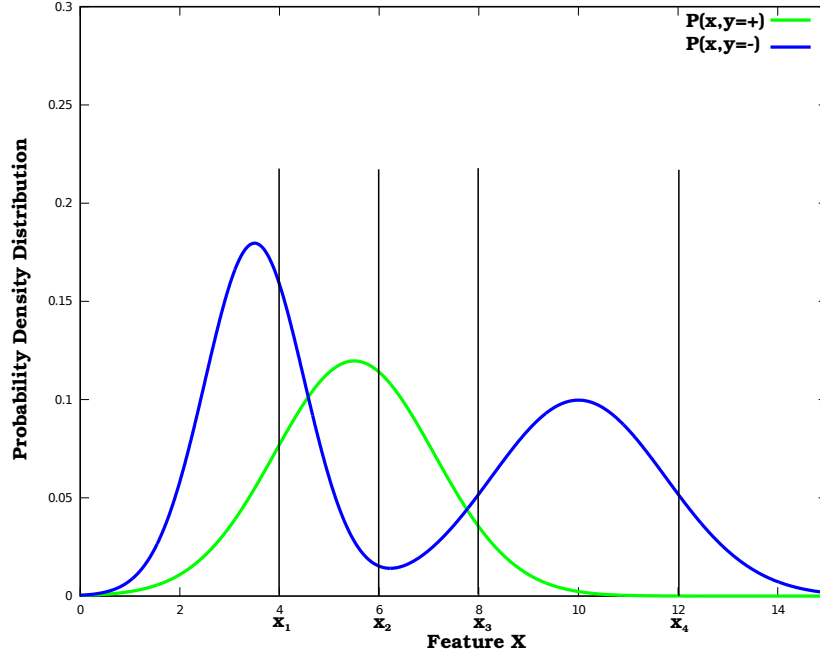


Figure 4: Area Under the Receiver Operating Characteristics (ROC) Curve (AUC) Plots.

- (X) Classifier A performs best.
- () Classifier A is a *perfect classifier*.
- () Classifier D is a *perfect classifier*.
- () Classifier A performs like a *randomly guessing classifier*.
- (X) Classifier D performs like a *randomly guessing classifier*.
- (X) Classifier A dominates classifiers B,C,D.
- () Classifier B dominates classifiers C,D.
- () Classifier D dominates classifiers A,B,C.
- (X) Classifier D is dominated by classifiers A,B, and C.
- (X) Neither classifier B dominates classifier C, nor classifier C dominates classifier B.

0.20 (Univariate) Bayesian Classification

Consider a binary classification problem, with class variable $Y \in \{pos, neg\}$ and feature variable $X \in (0, 15)$. The joint distributions of $P(X, Y)$ are given in the plot below ($P(X, Y = pos)$ is plotted in green, $P(X, Y = neg)$ in blue).



$P(X)$	$P(X, Y = +)$	$P(X, Y = -)$	Sum
$x_1 = 4$	0.077122	0.158710	0.235830
$x_2 = 6$	0.113980	0.015205	0.129180
$x_3 = 8$	0.035309	0.051915	0.087224
$x_4 = 12$	0.000031	0.051907	0.051939

- a) Plot the feature distribution $P(X)$

We know that $P(X)$ can be obtained from the joint distribution $P(X, Y)$ by marginalising over Y , i.e. $P(X) = \sum_{y \in \mathcal{Y}} P(X, Y) = P(X, Y = +) + P(X, Y = -)$

Thus, for example for $x_1 = 4$, $P(X = 4) = P(X = 4, Y = +) + P(X = 4, Y = -) \approx 0.077122 + 0.158710 = 0.235830$

- b) Can you derive the posterior class distributions $P(Y = pos|X)$ and $P(Y = neg|X)$ from the presented plot? Sketch them in the plot.

Using the Bayes' theorem, we know that $P(Y = +|X) = \frac{P(X|Y=+) \cdot P(Y=+)}{P(X)} = \frac{P(X, Y=+)}{P(X)}$

Thus, for example for $x_1 = 4$, $P(Y = +|X = 4) = \frac{P(X=4, Y=+)}{P(X=4)} \approx \frac{0.077122}{0.235830} = 0.32702$

- c) Mark the Bayes' optimal decision boundary/boundaries. How would you classify the following instances: $x_1 = 4$, $x_2 = 6$, $x_3 = 8$, $x_4 = 12$

Using our calculations from above, we classify as positive, when $P(Y = +|X) \geq P(Y = -|X)$ (or first derive the decision boundaries, where $P(Y = +|X) = P(Y = -|X)$, and classify accordingly in the intervals between). Thus, for example for $x_1 = 4$, $P(Y = +|X = 4) = 0.32702 \leq 0.67298 = P(Y = -|X = 4)$, therefore x_1 is classified as negative.

- d) Indicate the Bayes' error rate. Assuming a Bayes' optimal decision, annotate the number of false negatives, and the number of true negatives in the plot.

At each position of X , the Bayes' error rate there is indicated by the lower of $P(Y = +|X)$ and $P(Y = -|X)$, i.e. by $\min_{y \in \{+, -\}} P(Y = y|X)$. Integrating over X (i.e. the area under the lower of the two) yields the Bayes' error on this data set.

0.21 Multivariate Bayesian Classification

Consider the following, simplified golf-player classification example²:

Temp	Humidity	Windy	Play
hot	high	false	no
hot	high	true	no
hot	high	false	yes
mild	high	false	yes
cool	normal	false	yes
cool	normal	true	no
cool	normal	true	yes
mild	high	false	no
cool	normal	false	yes
mild	normal	false	yes
mild	normal	true	yes
mild	high	true	yes
hot	normal	false	yes
mild	high	true	no

a) Calculate and **interpret** the following probabilities:

- $\Pr(X_{\text{temp}} = \text{hot}) = \frac{4}{14}$
- $\Pr(y = \text{yes}) = \frac{9}{14}$
- $\Pr(X_{\text{temp}} = \text{hot}, y = \text{yes}) = \frac{2}{14}$
- $\Pr(X_{\text{temp}} = \text{hot} | y = \text{yes}) = \frac{\frac{2}{14}}{\frac{9}{14}} = \frac{2}{9}$
- $\Pr(y = \text{yes} | X_{\text{temp}} = \text{hot}) = \frac{\frac{2}{14}}{\frac{4}{14}} = \frac{2}{4}$

b) Explain the classification of a **multivariate Bayes-optimal** classifier for an instance with $X_{\text{temp}} = \text{cool}$ and $X_{\text{hum}} = \text{normal}$ and calculate the necessary probabilities.

$$\Pr(Y = \text{yes} | X_{\text{temp}} = \text{cool}, X_{\text{hum}} = \text{normal}) = \quad (1)$$

$$= \frac{\Pr(Y = \text{yes}, X_{\text{temp}} = \text{cool}, X_{\text{hum}} = \text{normal})}{\Pr(X_{\text{temp}} = \text{cool}, X_{\text{hum}} = \text{normal})} = \frac{\frac{3}{14}}{\frac{4}{14}} = \frac{3}{4} \quad (2)$$

c) Explain the classification of a **Naive Bayes** classifier for an instance with $X_{\text{temp}} = \text{cool}$ and $X_{\text{hum}} = \text{normal}$ and calculate the necessary values.

$$\Pr(Y = \text{yes} | X_{\text{temp}} = \text{cool}, X_{\text{hum}} = \text{normal}) \quad (3)$$

$$\propto \Pr(Y = \text{yes}) \cdot \Pr(X_{\text{temp}} = \text{cool} | Y = \text{yes}) \cdot \Pr(X_{\text{hum}} = \text{normal} | Y = \text{yes}) \quad (4)$$

$$= \frac{9}{14} \cdot \frac{3}{9} \cdot \frac{6}{9} = \frac{1}{7} \quad (5)$$

$$\Pr(Y = \text{no} | X_{\text{temp}} = \text{cool}, X_{\text{hum}} = \text{normal}) \quad (6)$$

$$\propto \Pr(Y = \text{no}) \cdot \Pr(X_{\text{temp}} = \text{cool} | Y = \text{no}) \cdot \Pr(X_{\text{hum}} = \text{normal} | Y = \text{no}) \quad (7)$$

$$= \frac{5}{14} \cdot \frac{1}{5} \cdot \frac{1}{5} = \frac{1}{70} \quad (8)$$

²Based on a fictitious dataset from [Witten et al., 2011, chapter 11]

0.22 Classification and Regression in R

In R, replicate the Golf-Player data set from the exercise above with the following command:

```
golf_train<-data.frame(
  temp=c("hot","hot","hot","mild","cool","cool","cool","mild","cool","mild","mild","mild","hot","mild"),
  hum=c("h","h","h","h","n","n","n","h","n","n","n","n","h","n"),
  windy=c("f","t","f","f","f","t","t","f","f","f","t","t","f","t"),
  play=c("n","n","y","y","y","n","y","n","y","y","y","y","y","n"))
```

Next, generate a test set with the following command:

```
golf_test<-data.frame(temp=c("cool","cool","hot"),hum=c("n","n","h"),
  windy=c("t","f","t"),play=c("n","y","n"))
```

- a) In R, use the `naiveBayes()` and `predict()` functions from the package *e1071* to train a Naive Bayes classifier on the attributes *temp* and *hum*, and to predict the class label of the test instances. Compare the results with your calculation in the exercise above!

```
library("e1071")
nb<-naiveBayes(play~temp+hum,data=golf)
predict(nb,golf_test)
```

- b) In R, use the `rpart()` and `predict()` functions from the package *rpart* to train a decision tree on the attributes *temp* and *hum*, and to predict the class label of the test instances. Compare the results with above!

Note: Due to the small training set size, you will have to adjust some of *rpart*'s control paramters:

minsplit = 2, *minbucket* = 1, and *xval* = 3 (for reasonably large data sets you should use higher settings!).

Finally, use the `HMeasure()` function from the package *hmeasure* to compute the error rate and AUC.

```
library("rpart")
dectree_control=rpart.control(minsplit = 2, minbucket = 1, xval=3)
dectree<-rpart(play~temp+hum,data=golf,control=dectree_control)
dectree_pred<-data.frame(class=predict(dectree,golf_test,type="class"),
  prob=predict(dectree,golf_test,type="prob"))
dectree_hm<-HMeasure(golf_test$play, dectree_pred$prob.y)
print(paste("Error Rate: ",dectree_hm$metrics$ER))
print(paste("AUC: ",dectree_hm$metrics$AUC))
```

Remark: Please show also what happens when fitting *rpart* without the parameter adjustments - we get a tree trunc with only the root note, which classifies all instances the same way! This might be a good learning experience to **always** also check the model produced by the algorithms.

- c) In R, use the build-in `lm()` function to fit a linear regression model on the build-in *iris* data set. This model should use *Sepal.Length* as dependent (response) variable, and *Petal.Length* as well as *Petal.Width* as explanatory variables. Print a summary of the model and plot a histogram of the residuals. Based on your model, discuss the relationship between *Petal.Length*, *Petal.Width* and *Sepal.Length*. What is the predicted *Sepal.Length* for a plant with (*Petal.Length* = 4.3, *Petal.Width* = 1.3)?

```
summary(iris)
linrm<-lm(data=iris,formula=Sepal.Length~Petal.Length+Petal.Width)
summary(linrm)
hist(residuals(linrm))
iris_test<-data.frame(Petal.Length=c(4.3),Petal.Width=c(1.3))
predict(linrm,newdata=iris_test)
# 4.19058 + 0.54178*4.3 - 0.31955*1.3 = 6.104819
```

There is a positive association between *Petal.Length* and *Sepal.Length*, but a negative association between *Petal.Width* and *Sepal.Length*. The residuals are approximately normal distributed (which is good).

0.23 Frequent Itemset Mining

Apply the apriori algorithm to the basket data in Table 1, using as threshold for support $\sigma \geq \frac{3}{8}$ and for confidence $c \geq \frac{1}{2}$. Based on the algorithm, which recommendations (in descending order of relevance; also state the recommendations' confidences!) should be made to the following customers:

a) Customer I, who has bought B (single item recommendation suffices).

1. Iteratively, create in the i -th iteration candidate itemsets (CIs) of cardinality i as combinations of (frequent) itemsets FIs (of lower cardinality $i - 1$). Calculate for each candidate set its support as

$$\text{support}(\{B\}) = \frac{\text{transactions containing } B}{\text{number of all transactions}}$$

and selecting those that are as frequent as required by the support threshold σ . If for a cardinality i no FI exists, stop.

2. In the following, consider only the subset of the frequent itemsets, i.e. those that have a support $\sigma \geq 3/8$ and that contain item B .
3. For each FI in this subset, create the set difference $FI \setminus \{B\}$, that is, the subset of the items in the FI that are different than B . For those, calculate the confidence of recommending them:

$$\text{Confidence}(B \rightarrow C) = \frac{\text{support}(\{B, C\})}{\text{support}(\{B\})}$$

4. Sort and recommend the items based on their decreasing confidence.

		Itemset	Support			
		$\{A, B\}$	2/8			$\text{conf}(B \rightarrow C) = \frac{5/8}{7/8} = 5/7$
Itemset	Support	$\{A, C\}$	2/8	Itemset	Support	$\text{conf}(B \rightarrow D) = \frac{3/8}{7/8} = 3/7$
		$\{A, D\}$	2/8			$\text{conf}(B \rightarrow G) = \frac{6/8}{7/8} = 6/7$
		$\{A, G\}$	2/8			$\text{conf}(\{B, C\} \rightarrow G) = \frac{4/8}{5/8} = 4/5$
		$\{B, C\}$	5/8			
		$\{B, D\}$	3/8			
		$\{B, G\}$	6/8			
		$\{C, D\}$	2/8			
		$\{C, G\}$	4/8			
		$\{D, G\}$	3/8			
$\{A\}$	3/8			$\{B, C, D\}$	2/8	
$\{B\}$	7/8			$\{B, C, G\}$	4/8	
$\{C\}$	5/8			$\{B, D, G\}$	3/8	
$\{D\}$	4/8			$\{C, D, G\}$	2/8	
$\{G\}$	6/8					
$\{H\}$	2/8					

Figure 5: Solution: Item G with confidence $\frac{6}{7}$, item C with confidence $\frac{5}{7}$

b) Customer II, who has bought B and C (single item recommendation suffices). Item G with confidence $\frac{4}{5}$

IID	Item Name	TID	Items in Baskez
A	Grey's Anatomy	1	A,B,C,D,G,H
B	The Big Bang Theory	2	A,B,C,G
C	Castle	3	A,D,H
D	Downton Abbey	4	B,C
G	Game of Thrones	5	B,C,D,G
H	How I Met Your Mother	6	B,C,G
		7	B,D,G
		8	B,G

(a) List of Items

(b) Transactions / Baskets

Table 1: Items and Baskets for Frequent Itemset Mining

0.24 Extra: Frequent Itemset Mining in R

In R, use the function `apriori()` from the *arules*-package to mine the *Groceries*-data set for frequent itemsets. As parameter values, use 0.5 for support and 0.9 for confidence.

Note: This requires a version of *R* $\geq 3.4.0$

```
library(arules)
data(Groceries)
summary(Groceries)
ap<-apriori(data=Groceries, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))
summary(ap)
```

0.25 Extra: Text Mining:

In R, use the function `sentiment()` from the *sentimentr*-package to perform sentiment analysis on the text in the *course_evaluation*-data set.

Note: This requires a version of *R* $\geq 3.4.0$

See the `sentiment()`-function example on pages 27–28 in the documentation of the *sentimentr*-package.

Appendix

References

[Witten et al., 2011] Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3 edition.