Multidimensional Scaling and Naive Bayesian Classifier

Week 10, Spring 2023

Learn about...

- Distance
- Dimensional reduction method
  - Multidimensional Scaling (MDS)
- Naïve Bayes Classifier for classification

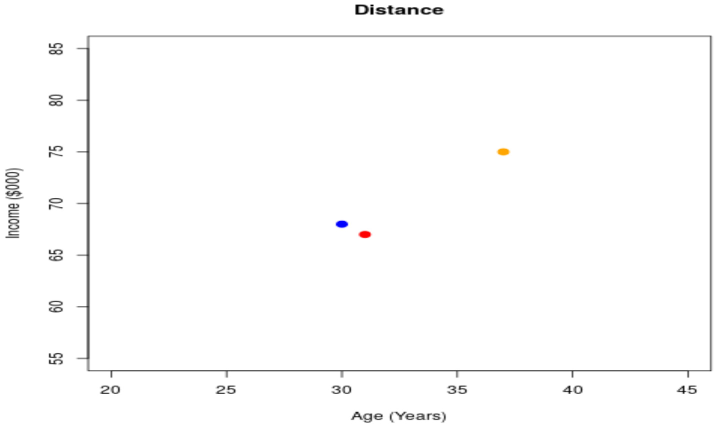
Why Distance?

- Many problems that involve thinking about how similar or dissimilar two observations are. For example:
  - We may use the same marketing strategy for people from similar demographic groups
  - We may lend money to applicants who are similar to those who pay the debts back.
- Distance could be one of the important concept in data science.

Simple Example

- Consider 3 individuals
  - Mr orange: 37 years of age earns $75k a year
  - Mr red: 31 years of age earns $67k a year
  - Mr blue: 30 years of age earns $68k a year
- Which two are the most similar?

On a scatterplot

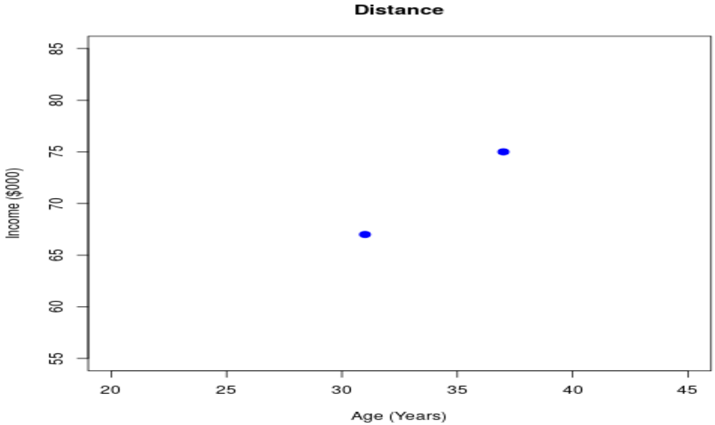


▶

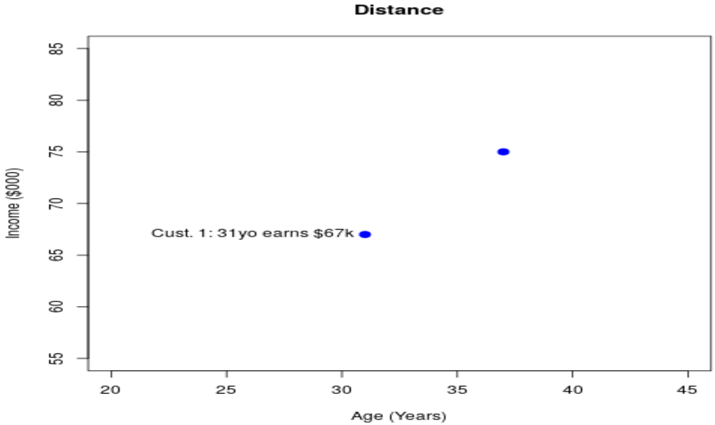
Distance as a number

- It is easy to think about three individuals but what if there are thousands of individuals?
  - In this case, it will be useful to attach some number to the distance between pairs of individuals
  - We will do it with a simple application of the Pythagoras theorem.

Finding a distance

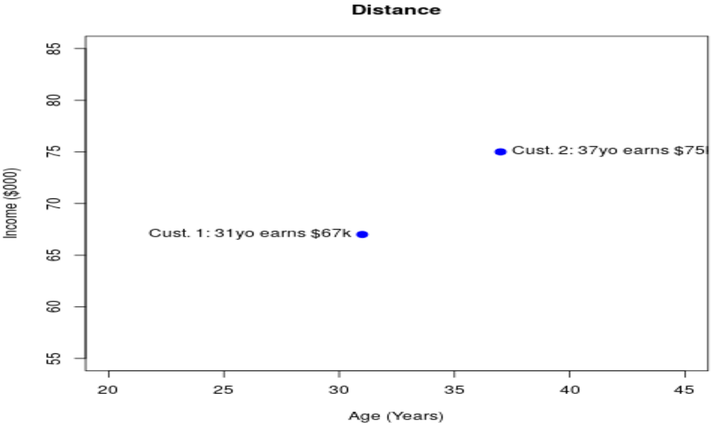


Finding a distance



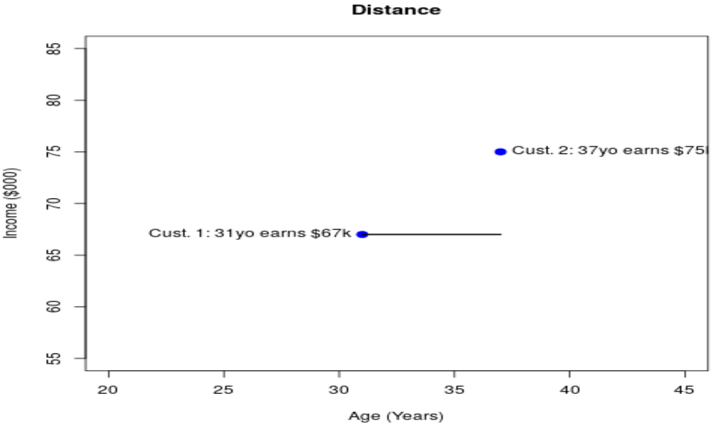
▶

Finding a distance



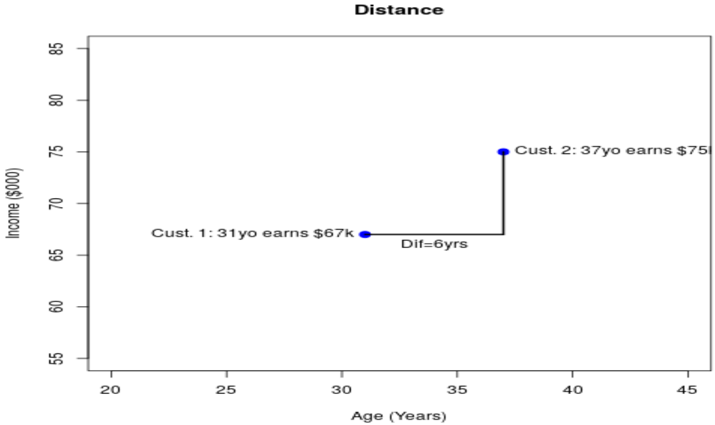
▶

Finding a distance



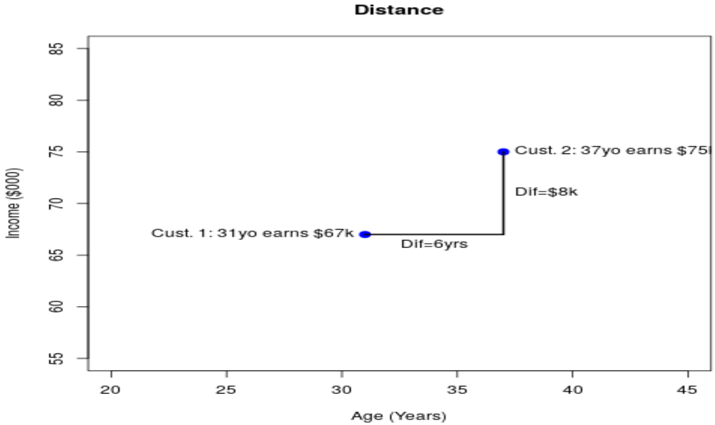
▶

Finding a distance



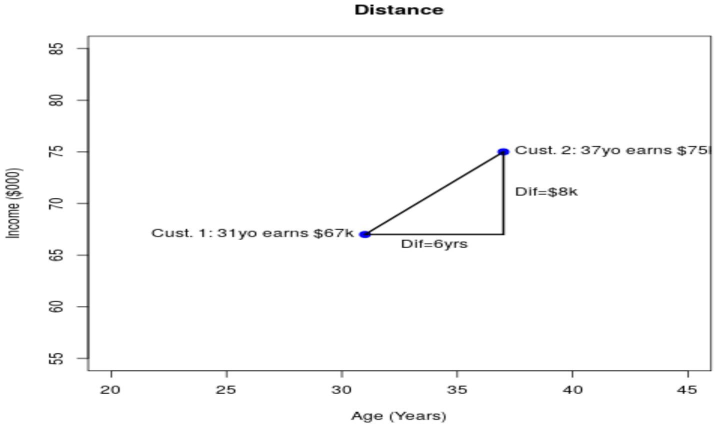
▶

Finding a distance



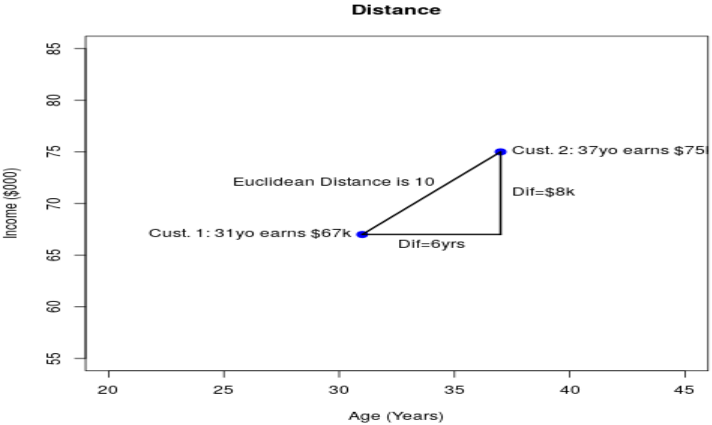
▶

Finding a distance



▶

Finding a distance



Euclidean Distance

- In general, there are more than 2 variables
- Is there a way to apply our intuition in 2
  - Pythagoras theorem can be generalised to higher dimensions.
  - This results in a concept of distance called Euclidean distance.

Euclidean Distance

- We measure p variables for two observations x_j is the measurement of variable j for observation x, y_j is the measurement of variable j for observation y.
- Euclidean distance between x and y

$$D(x, y) = \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2}. \tag{1}$$

Distance and Standardising Data

- We must be careful about the units of measurement.
- Euclidean distance change for variables measured in different units
- For this reason, it is common to calculate distance after standardising data
  - Z scores. Values are standardized to Z scores, with a mean of 0 and a standard deviation of 1.
- If the variables are all measured in the same units, then this standardisation is unnecessary.

Non-Metric Data

- Can we define distance when the variables are non-metric?
- The answer is yes
  - For example: Jaccard Similarity/Distance

First a motivation

- Many people use music streaming services like spotify.
- One of the attractions of these services is they recommend artists based on the favourite artists of other users who have similar taste in music.
- The data in this case is in the form of a list of favourite artists

Distance in musical taste

- Suppose there are three customers with the following favourite artists
  - Customer A: Maroon 5, Ariana Grande, Ed Sheeran, Cardi B
  - Customer B: Maroon 5, Ed Sheeran, and BTS
  - Customer C: Cardi B, Drake, Future
- How do we measure which customers have similar taste and which have different taste?

Distance in musical taste

- Jaccard similarity gives us a measure how close two sets are, in this case the set of each customers favourite musician.
- The formula is
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$
- Where $|A \cap B|$ is the number of elements in both set A and set B and $|A \cup B|$ is the number of elements in either set A or set B

Jaccard Similarity and Distance

- In our example:
  - $A \cap B = \{Maroon5, EdSheeran\}$ and $|A \cap B| = 2$
  - $A \cup B = \{Maroon5, Ariana, EdSheeran, CardiB, BTS\}$ and $|A \cup B| = 5$
- The Jaccard similarity will be $J = 2/5 = 0.4$ and the Jaccard distance is $d_J = 1 - J = 1 - 0.4 = 0.6$.

Outline

Multidimensional Scaling

Bayesian Networks

Multidimensional Scaling: Introduction

- Previously we looked at the concept of distance between observations
- We looked at our usual understanding of distance known as Euclidean distance
- We also looked at higher dimensional versions of Euclidean distance
- Other distance metrics including Jaccard distance can be used for categorical data.

Multidimensional Scaling: Introduction

- Suppose that we have n observations and the distance between each possible pair of observations
- A scatterplot shows whether observations are close together or far apart.
- This works nicely when there are 2 variables.

Multidimensional Scaling: Introduction

- Suppose that we have p variables where p is large
- Consider p-dimensional Euclidean distances
- Can we represent these using 2 dimensions?
- Unfortunately the answer is no
- but we can still get good approximation

Multidimensional Scaling: Introduction

- Multidimensional Scaling (MDS) finds a low (usually 2) dimensional representation
- The pairwise 2D Euclidean distance in this representation should be as close as possible to the original distances
- MDS always begins with a matrix of distances and ends with a low dimensional representation that can be plotted.

Multidimensional Scaling: Introduction

- Multidimensional scaling (MDS) is a technique for dimensionality reduction
- General idea:
  - Find a projection of the data (e.g. p-vectors $x_1, ..., x_n$) to a lower dimensional space (e.g. q-vectors $z_1, ..., z_n$) such that the pairwise distances are preserved as well as possible.

Classic MDS

- Let $d_{i,j}$ denote the dissimilarity ("Euclidean" distance) between original variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

$$d_{i,j} = ||\boldsymbol{x}_i - \boldsymbol{x}_j|| = \sqrt{\sum_{k=1}^{p} (x_{i,k} - x_{j,k})^2}$$

- Let

$$d'_{i,j} = ||\boldsymbol{z}_i - \boldsymbol{z}_j|| = \sqrt{\sum_{k=1}^{q} (z_{i,k} - z_{j,k})^2}$$

  denote the corresponding q-dimensional distance after transforming the data to q-dimensions.
- *Classic* MDS can be applied when all data ($\boldsymbol{x}_i$s) are numeric.

An optical illusion with Beyonce

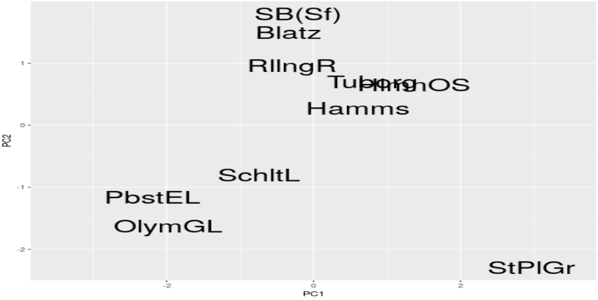
An optical illusion with Beyonce

- The photo is a 2D representation of a 3D reality
- In reality the distance between Beyonce's hand and the Eiffel tower is large
- In the 2D photo, this distance is small.
- This is a misleading representation to understand the distance between Beyonce's hand and the Eiffel tower.

Why do we care?

- An important issue in business is to profile the market. For example:
    - Which products do customers perceive to be similar to one another?
    - Who is my closest competitoR?
    - etc...
- Multidimensional scaling can help us to produce a simple visualisation (usually in 2D) that can address these questions.

Beer Example



Beer Example

- The plot on the previous slide is an MDS solution for the beer dataset
- The data are 5-dimensional so we cannot use a scatter plot.
- MDS shows Olympia Gold Light and Pabst Extra Light are similar
- This also suggests that there is a low number of competitors with St. Pauli Girl.
- This may also reflect that the attributes of St. Pauli Girl are not desired by Customers.

Classic MDS

- In classic MDS, the objective is to minimise the objective function called strain (the exact formula is not needed for this course).
- The above problem has a tractable solution when Euclidean distance is used
- This solution depends on an Eigenvalue decomposition
- We try to get a 2D view that represents the true distances as accurately as possible.

Non-Metric Example

- The following example comes from 'Multidimensional Scaling of Sorting Data Applied to Cheese Perception', Food Quality and Preference,6, pp.91-98. The purpose of this study was to visualise the difference between types of cheese.

Non-Metric Example: Cheese

- The motivation is to investigate the similarities and differences between types of cheese.
- In principle one could measure attributes of the cheese.
- However the purpose of this study was to ask customers about their perceptions
- How do we ask customers about distances?
- Could you walk out on to the street and ask someone about the Euclidean distance between Brie and Camembert?

Constructing the survey

- Customers can be asked:
- On a scale of 1 to 10 with 1 being the most similar and 10 being the most different, how similar are the following cheeses
  - Brie and Camembert
  - Brie and Roquefort
  - Camembert and Roquefort
- The dissimilarity scores can be averaged over all customers and used in an MDS
- This is not a good method when there is a large number of products.

A more feasible approach

- In the study there are 16 cheeses therefore 120 possible pairwise comparisons
- It is not practical to ask survey participants to make 120 comparisons!
- Instead of being asked to make so many comparisons, customers were asked to put similar cheeses into groups.
- Proportion of customers with two cheeses in same group is a similarity score.
- Proportion of customers with two cheeses in different groups is a dissimilarity score.

Consider four customers

- Suppose there are four customers sorting cheeses
  - Customer A: Brie and Camembert together, Roquefort and Blue Vein together
  - Customer B: Roquefort and Blue Vein together, all others separate
  - Customer C: All cheeses in their own category
  - Customer D: All cheeses in one category

Consider four customers

- Customer A and D have Brie and Camembert in the same group, customers B and C have them in different groups.
  - The distance between Brie and Camembert is 0.5.
- Customer A, B and D have Roquefort and Blue Vein in the same group, customer C has them in different groups.
  - The distance between Roquefort and Blue Vein is 0.25.

Non-Metric MDS

- The study on cheese did not use classical MDS but something called Kruskals algorithm.
- Kruskal's algorithm is implemented in R using the isoMDS function from the MASS package.
- Kruskal's algorithm is invariant to monotone transformations of the distances.

Monotone tranformations

- By monotone transformation we mean any function of the distance that is either constantly increasing or decreasing.
  - Exponential function is monotone
  - Sine function is not monotone
- By invariant we mean that the solution provided by Kruskal's does not change if we transform the input distances.

Non-Metric MDS

- The study on cheese did not use classical MDS but something called Kruskals algorithm.
- Kruskal's algorithm is implemented in R using the isoMDS function from the MASS package.
- In some cases, the distance themselves are not metric but ordinal.
- Suppose we only know that

$$d_{Bri,Cam} < d_{Roq,Cam} < d_{Roq,Bri}.$$

- Brie and Roquefort are more different compared to Brie and Camembert.
- We do not know how big the distance between Brie and Roquefort is compared to the distance between Brie and Camembert.

Non-Metric MDS

- In this case we minimise the objective function using only the information:

$$d_{Bri,Cam} < d_{Roq,Cam} < d_{Roq,Bri}.$$

- Taking the ranks is an example of a monotone transformation.
- Therefore the solution of isoMDS only requires the ranks of the distances and not the distances themselves.
- This is a very useful algorithm for marketing since survey participants cannot easily and reliable assign numbers to the difference between products.

Modern MDS

- Methods for finding a low dimensional representation of high-dimensional data continue to be used today.
- These mostly go by the name of manifold learning methods
- For example: Local Linear Embedding (LLE), ISOMAP, Laplacian Eigenmap, etc....

R Implementation

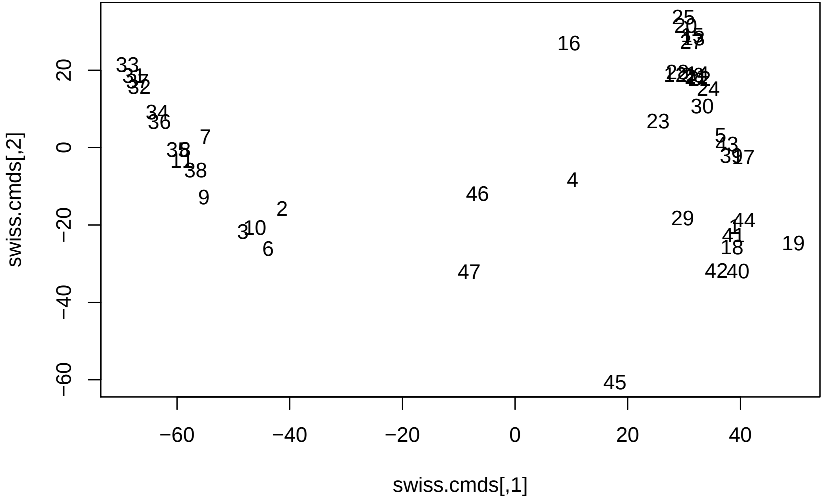
```r
library(MASS)
data(swiss)
# Leave out Fertility:
swiss.x <- as.matrix(swiss[, -1])
# Compute pairwise distances between provinces:
as.matrix(swiss.dist <- dist(swiss.x))[1:4, 1:4]

##              Courtelary Delemont Franches-Mnt Moutier
## Courtelary         0.00    80.54        87.35   31.38
## Delemont          80.54     0.00        11.11   52.21
## Franches-Mnt      87.35    11.11         0.00   60.16
## Moutier           31.38    52.21        60.16    0.00
```

R Implementation (continued)

```r
# Obtain classic MDS:
head(swiss.cmds <- cmdscale(swiss.dist), 5)

##                 [,1]    [,2]
## Courtelary     38.96 -20.405
## Delemont      -41.36 -15.751
## Franches-Mnt  -48.30 -21.660
## Moutier        10.21  -8.270
## Neuveville     36.49   3.219

# Obtain ISOMDS estimates:
head((swiss.mds <- isoMDS(swiss.dist, trace = FALSE))$points,
    5)

##                 [,1]    [,2]
## Courtelary     39.98 -18.665
## Delemont      -42.14 -15.835
## Franches-Mnt  -49.18 -23.502
## Moutier        10.64  -7.795
## Neuveville     35.71   4.224
```

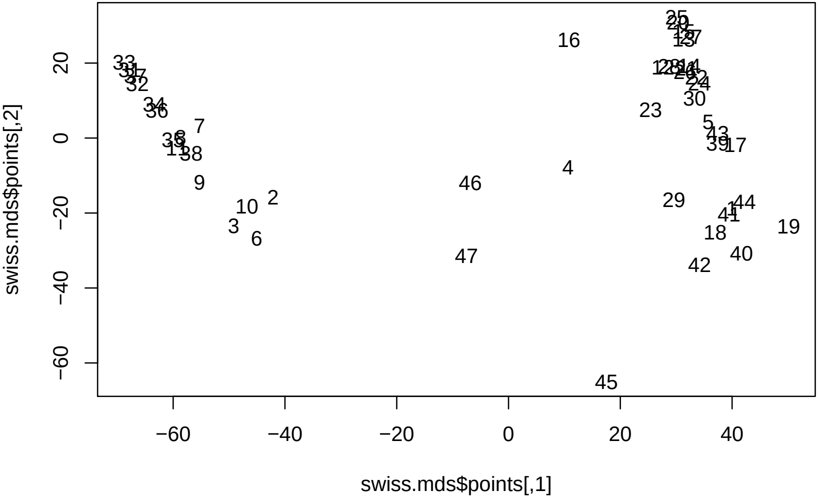
Plotting Results (Classic)

```r
n <- nrow(swiss.x)
plot(swiss.cmds, type = "n")  # type='n' => set up axes only
text(swiss.cmds, labels = as.character(1:n))  # put text at locations
```



Plotting Results (ISOMDS)

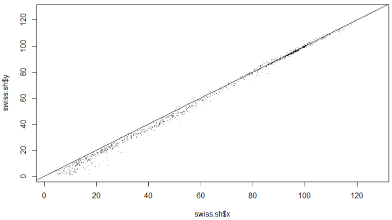
```r
n <- nrow(swiss.x)
plot(swiss.mds$points, type = "n")
text(swiss.mds$points, labels = as.character(1:n))
```



Shepard Plot

- The *Shepard* plot is a scatterplot of the original distance against the distance when the data is transformed.
- A narrow scatter around a 1:1 line indicates a good fit of the distances when the data is transformed to the original distance, while a large scatter or a nonlinear pattern indicates a lack of fit.

```
swiss.sh <- Shepard(swiss.dist, swiss.mds$points)
plot(swiss.sh, pch = ".")  # Plot the points
abline(swiss.sh, pch = ".")  # Plot the points
```



Outline

Conditional Probability

- The *conditional* probabiity of B given A tells us how likely an event B is when we know that A has occurred:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

  - The *intersection* $A \cap B$ denotes the event that both A *and* B occur.

$$P(A \cap B) = P(B)P(A|B).$$

Independence

- The events A and B are said to be *independent* if

$$P(A \cap B) = P(A) \times P(B)$$

  or equivalently $P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{\cancel{P(A)} \times P(B)}{\cancel{P(A)}} = P(B)$.

  - *Conditional independence of A and B given C* means that

$$P(A \cap B|C) = P(A|C) \times P(B|C)$$

Bayes's Rule

- For events A and B, where $B^{\complement}$ denotes the *complement* "not B", *Bayes's Rule* provides a way to reverse the order of events in a conditional probability:

$$
\begin{aligned}
P(B|A) = \frac{P(B \cap A)}{P(A)} &= \frac{P(B \cap A)}{P(B \cap A) + P(B^{\complement} \cap A)} \\
&= \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B^{\complement})P(B^{\complement})}
\end{aligned}
$$

Bayes's Rule

- More generally, for discrete random variables $Y \in \mathcal{Y}$ and X,

$$P(Y = y \mid X = x) = \frac{P(X = x \mid Y = y)P(Y = y)}{\sum_{y' \in \mathcal{Y}} P(X = x \mid Y = y')P(Y = y')}$$

- Here, $\mathcal{Y}$ is all possible values of Y.
- I will be using $P(y')$ as shorthand for $P(Y = y')$, and similarly for $P(x)$, $P(y|x)$, etc..

Optimal Bayes Classifier

- Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a set of predictor variables and Y be the response (for just one "random" observation).
- According to the *Bayes classifier*, our best guess for Y is

$$\hat{y} = \arg\max_{y} P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{\sum_{y' \in \mathcal{Y}} P(\boldsymbol{x}|y')P(y')}$$

- Unfortunately, we usually can't estimate $P(\boldsymbol{x}|y)$ with any degree of accuracy.
  - i.e., *every possible combination of predictors* for each level of response.

Outline

Conditional Independence

- Let $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ be a sets of variables.
- $\boldsymbol{X}$ is said to be *conditionally independent* of $\boldsymbol{Z}$ given $\boldsymbol{Y}$ if

$$P(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{z}) = P(\boldsymbol{x}|\boldsymbol{y}).$$

- If (a big "if") for a $\boldsymbol{X} = (X_1, X_2)$, X_1 cond. ind. X_2 given Y,

$$\begin{aligned} P(\boldsymbol{x} \mid y) &= P(x_1|y) \times P(x_2 \mid x_1, y) \\ &= P(x_1|y) \times P(x_2|y). \end{aligned}$$

$\Longrightarrow$ Much easier, since it can be done one dimension at a time.

Naïve Bayes Classifier

- The *Naïve Bayes* Classifier is a probability classifier based on an assumption of conditional independence between attributes:

$$P(\boldsymbol{x} \mid y) = \prod_{i=1}^{p} P(x_i|y)$$

- This leads to the following approximation based on prior probabilities $P(Y = y)$:

$$P(y|\boldsymbol{x}) \approx \frac{P(y)\prod_{i=1}^{p} P(x_i|y)}{\sum_{y' \in \mathcal{Y}} P(y')\prod_{i=1}^{p} P(x_i|y')}.$$

Advantages and Disadvantages

+ Despite the implausible assumption, Naïve Bayes has been found to work surprisingly well in a broad range of applications.
+ Training and classification are very fast.
− Can behave very poorly if the predictors are strongly dependent on each other.

Categorical Attributes

- Estimate $P(X_i = x_i \mid Y = y)$ within training set by

$$\frac{\text{no. of instances with class } Y = y \text{ and attribute } X_i = x_i}{\text{no. of instances with class } Y = y}$$

Continuous attribute

- Assume the form of *probability distribution* for each continuous attribute.
  - Often a Gaussian (Normal) distribution is assumed, with parameters μ and σ^2.
  - In this case, estimate μ and σ^2 by the sample mean $\overline{x}_{y,i}$ and sample variance $s^2_{y,i}$ of attribute x_i for observations in class y.

Handling Gaussian approximation

- So just proceed with Naïve Bayes as for the discrete case, using

$$P(x_i \mid y) \approx \frac{1}{s_{y,i}\sqrt{2\pi}} \exp\left(-\frac{(x_i - \overline{x}_{y,i})^2}{2s_{y,i}^2}\right),$$

  to replace $P(X_i = x_i \mid Y = y)$ for any continuous attribute X_i.
- We often define $\phi(x) \equiv (2\pi)^{-1/2}\mathrm{e}^{-x^2/2}$, so $P(x_i \mid y) \approx \phi\left(\frac{x_i - \overline{x}_{y,i}}{s_{y,i}}\right)/s_{y,i}$.
- In R, $\phi(x) =$`dnorm(x)`, and $\phi\left(\frac{x-m}{s}\right)/s =$`dnorm(x, m, s)`.

Example: Iris data

- Predict species y from sepal measurements $(x_1, \ldots, x_4)$.
- Function `naiveBayes` in package `e1071`.
  - Takes regular formula interface.
  - Handles quantitative predictors using normal approximation.
- Outputs:
  - *a priori probabilities* $P(y)$ for $y \in \mathcal{Y}$
  - for categorical predictors x_i, a table $P(x_i|y)$ for each $y \in \mathcal{Y}$
  - for quantitative predictors, a mean $\overline{x}_{y,i}$ and a standard deviation $s_{y,i}$ for each $y \in \mathcal{Y}$

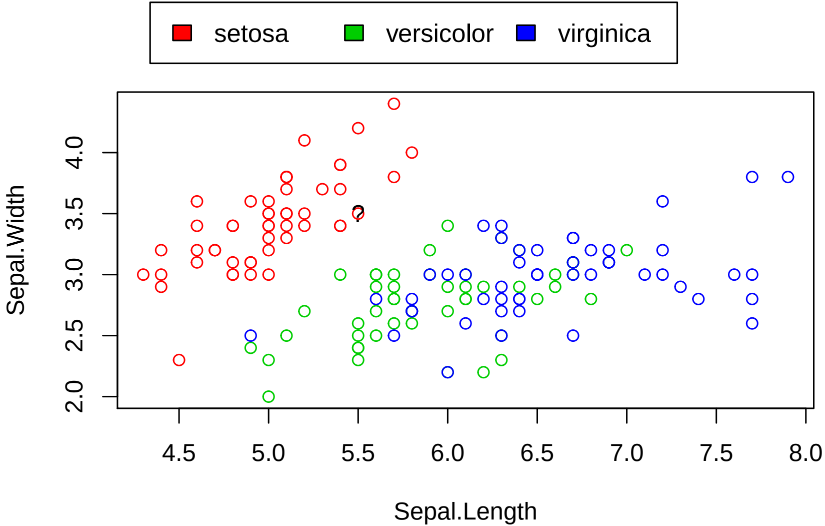
```
data(iris)
library(e1071)
(m <- naiveBayes(Species ~ Sepal.Length + Sepal.Width,
    data = iris))
```

R output

```
## Naive Bayes Classifier for Discrete Predictors
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
## A-priori probabilities:
## Y
##     setosa versicolor  virginica
##     0.3333     0.3333     0.3333
## Conditional probabilities:
##             Sepal.Length
## Y              [,1]   [,2]
##   setosa     5.006 0.3525
##   versicolor 5.936 0.5162
##   virginica  6.588 0.6359
##             Sepal.Width
## Y              [,1]   [,2]
##   setosa     3.428 0.3791
##   versicolor 2.770 0.3138
##   virginica  2.974 0.3225
```

Naive Bayes with quantitative predictors: scatterplot

Predict species for sepal length 5.5 cm and sepal width 3.5 cm:



Naive Bayes with quantitative predictors

$$
\begin{aligned}
&P(Y = \text{setosa} | X_1 = 5.5, X_2 = 3.5)\\
&\quad \propto P(Y = \text{setosa}) P(X_1 = 5.5 | Y = \text{setosa}) P(X_2 = 3.5 | Y = \text{setosa})\\
&\quad = 0.3333 \times \phi\left(\frac{5.5 - 5.0060}{0.3525}\right)/0.3525 \times \phi\left(\frac{3.5 - 3.4280}{0.3791}\right)/0.3791\\
&\quad = 0.3333 \times 0.4239 \times 1.0336 = 0.1461
\end{aligned}
$$

$$
\begin{aligned}
&P(Y = \text{versi.} | X_1 = 5.5, X_2 = 3.5)\\
&\quad \propto P(Y = \text{versi.}) P(X_1 = 5.5 | Y = \text{versi.}) P(X_2 = 3.5 | Y = \text{versi.})\\
&\quad = 0.3333 \times \phi\left(\frac{5.5 - 5.9360}{0.5162}\right)/0.5162 \times \phi\left(\frac{3.5 - 2.7700}{0.3138}\right)/0.3138\\
&\quad = 0.3333 \times 0.5410 \times 0.0849 = 0.0153
\end{aligned}
$$

Naive Bayes quantitative predictors (continued)

$$
\begin{aligned}
&P(Y = \text{virgi.} | X_1 = 5.5, X_2 = 3.5) \\
&\quad \propto P(Y = \text{virgi.}) P(X_1 = 5.5 | Y = \text{virgi.}) P(X_2 = 3.5 | Y = \text{virgi.}) \\
&\quad = 0.3333 \times \phi\left(\frac{5.5 - 6.5880}{0.6359}\right) / 0.6359 \times \phi\left(\frac{3.5 - 2.9740}{0.3225}\right) / 0.3225 \\
&\quad = 0.3333 \times 0.1452 \times 0.3271 = 0.0158
\end{aligned}
$$

$$
\begin{aligned}
P(Y = \text{setosa} | X_1 = 5.5, X_2 = 3.5) &= \tfrac{0.1461}{0.1461 + 0.0153 + 0.0158} = 0.8242 \\
P(Y = \text{versi.} | X_1 = 5.5, X_2 = 3.5) &= \tfrac{0.0153}{0.1461 + 0.0153 + 0.0158} = 0.0864 \\
P(Y = \text{virgi.} | X_1 = 5.5, X_2 = 3.5) &= \tfrac{0.0158}{0.1461 + 0.0153 + 0.0158} = 0.0893
\end{aligned}
$$

```r
predict(m, newdata = data.frame(Sepal.Length = 5.5,
    Sepal.Width = 3.5), type = "raw")

##      setosa versicolor virginica
## [1,] 0.8242    0.08644   0.08932
```

Naive Bayes with categorical predictors

- Discretise by taking the integer part of the measurment (e.g., $3.5 \rightarrow 3$):

```
iris$SL.int <- factor(floor(iris$Sepal.Length))
iris$SW.int <- factor(floor(iris$Sepal.Width))
# Show a sample of rows:
(iris[, -(3:4)][sample.int(nrow(iris), 7), ])

##     Sepal.Length Sepal.Width    Species SL.int SW.int
## 58           4.9         2.4 versicolor      4      2
## 70           5.6         2.5 versicolor      5      2
## 3            4.7         3.2     setosa      4      3
## 132          7.9         3.8  virginica      7      3
## 8            5.0         3.4     setosa      5      3
## 122          5.6         2.8  virginica      5      2
## 89           5.6         3.0 versicolor      5      3
```

```
(m <- naiveBayes(Species ~ SL.int + SW.int, data = iris))
```

R output

```
## Naive Bayes Classifier for Discrete Predictors
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
## A-priori probabilities:
## Y
##     setosa versicolor  virginica
##     0.3333     0.3333     0.3333
## Conditional probabilities:
##             SL.int
## Y               4    5    6    7
##   setosa     0.40 0.60 0.00 0.00
##   versicolor 0.02 0.50 0.46 0.02
##   virginica  0.02 0.12 0.62 0.24
##             SW.int
## Y               2    3    4
##   setosa     0.04 0.88 0.08
##   versicolor 0.68 0.32 0.00
##   virginica  0.42 0.58 0.00
```

Naive Bayes with categorical predictors

Predict species for sepal length 5.5 cm and sepal width 3.5 cm:

$$P(Y = \text{setosa}|X_1 = 5.5, X_2 = 3.5) = P(Y = \text{setosa}|X_1 = 5, X_2 = 3)$$
$$\propto P(Y = \text{setosa})P(X_1 = 5|Y = \text{setosa})P(X_2 = 3|Y = \text{setosa})$$
$$\propto 0.3333 \times 0.6000 \times 0.8800 = 0.1760$$

$$P(Y = \text{versi.}|X_1 = 5.5, X_2 = 3.5) = P(Y = \text{versi.}|X_1 = 5, X_2 = 3)$$
$$\propto P(Y = \text{versi.})P(X_1 = 5|Y = \text{versi.})P(X_2 = 3|Y = \text{versi.})$$
$$\propto 0.3333 \times 0.5000 \times 0.3200 = 0.0533$$

Naive Bayes with categorical predictors (continued)

$$
\begin{aligned}
P(Y = \text{virgi.}|X_1 = 5.5, X_2 = 3.5) &= P(Y = \text{virgi.}|X_1 = 5, X_2 = 3) \\
&\propto P(Y = \text{virgi.})P(X_1 = 5|Y = \text{virgi.})P(X_2 = 3|Y = \text{virgi.}) \\
&\propto 0.3333 \times 0.1200 \times 0.5800 = 0.0232
\end{aligned}
$$

$$
\begin{aligned}
P(Y = \text{setosa}|X_1 = 5.5, X_2 = 3.5) &= \frac{0.1760}{0.1760+0.0533+0.0232} = 0.6969 \\
P(Y = \text{versi.}|X_1 = 5.5, X_2 = 3.5) &= \frac{0.0533}{0.1760+0.0533+0.0232} = 0.2112 \\
P(Y = \text{virgi.}|X_1 = 5.5, X_2 = 3.5) &= \frac{0.0232}{0.1760+0.0533+0.0232} = 0.0919
\end{aligned}
$$

```r
# We need to "tell" R that floor(5.5) -> 5 is a categorical
# variable, whose possible levels are those of that column in the
# iris dataset.
nd <- data.frame(
  SL.int=factor(floor(5.5), levels=levels(iris$SL.int)),
  SW.int=factor(floor(3.5), levels=levels(iris$SW.int)))
predict(m, newdata=nd,type="raw")

##      setosa versicolor virginica
## [1,] 0.6969     0.2112   0.09187
```