DEAKIN UNIVERSITY

Data Analysis with R
SIT718

Fitting Aggregation Functions

Just like with regression in statistics, we can find the best fitting parameters for an aggregation function. The idea is usually to minimise the sum of differences between predicted and observed values, with respect to the possible choices of weighting vector w.

$$\text{minimize}_{\mathbf{w}} \sum_{j=1}^{D} (A(\mathbf{x}_j) - y_j)^2$$

where j=1,2,…, D are the data points that we have observations for, yj is our observed output, xj is the input vector associated with that output and A(xj) is our predicted value

Fitting Aggregation Functions

we have restrictions on our weights.

$$w_i \geq 0, \text{for all} i$$

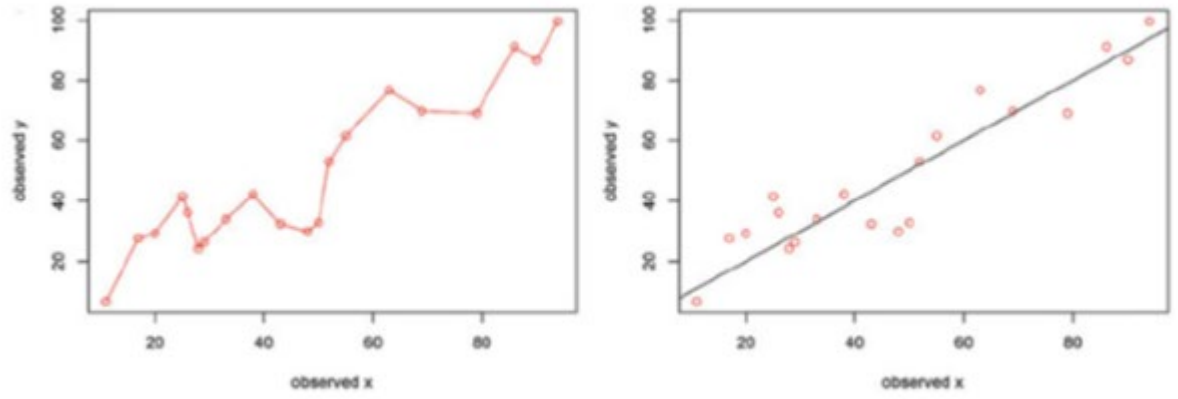
$$\sum_{i=1}^{n} w_i = 1$$

We can essentially use this process for one of two purposes:

- predicting the outputs for unknown/new data
- making inferences about the 'importance' of each variable based on the fitted weighting vector.

Fitting Aggregation Functions

The left graph is an example of overfitting: it shows a model flexible enough to fit every single data point, however any new values are unlikely to lie on this line. The right graph shows a line of best fit which does not necessarily match each data point individually, but would provide a prediction that is least likely to deviate from any new values.



Error Measures

We can measure a model's accuracy by comparing expected outputs with actual outputs in a few different ways.

- **Total Squared Error/Sum of Squared Error (SSE)**

$$SSE = \sum_{i=1}^{m} (A_w(x_i) - y_i)^2$$

- x_i are our input values,
- $A_w(x_i)$ is our expected output values based on the model,
- y_i are our observed output values.

Error Measures

After calculating the Sum of Squared Errors, we divide by the number of observations *n* and then take the square root of the result.

- **Root Mean Squared Error (RMSE)**

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(A_w(x_i) - y_i)^2}{n}}$$

- The RMSE can be interpreted as the average difference between each prediction and the output.
- It is also the quadratic mean, a power mean with $p = 2$, so it will be affected more by larger differences.
- If a fitted function performs better than another in terms of RMSE, then it will also have a lower total squared error.

Error Measures

The sum of all the absolute differences between predicted and observed outputs

- **Total Least Absolute Deviation (LAD) / Sum of Absolute Errors (SAE)**

$$SAE = \sum_{i=1}^{m} |A_w(x_i) - y_i|$$

- **Average L1 error / Average Absolute Error**

This is the SAE divided by the number of observations,

$$Av.\ AE = \sum_{i=1}^{n} \frac{|A_w(x_i) - y_i|}{n}$$

Error Measures

In statistics, the Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations.
The Pearson Correlation returns a value between −1 and 1 that indicates how much of a linear relationship there is between two variables, where

- **Pearson Correlation (r)**
  - $r = -1$ indicates a perfect negative relationship,
  - $r = 0$ indicates no relationship at all,
  - $r = 1$ indicates a perfect positive relationship.

we’re looking for our Pearson Correlation to be close to 1.

Error Measures

The Spearman Correlation is similar to the Pearson Correlation but assesses monotonic relationships instead of linear relationships. In other words, it looks at whether the ranking/ordering of values relative to each other are consistent between the two variables.

- **Spearman Correlation (ρρ - pronounced 'rho')**

- $\rho = -1$ indicates a monotonic decreasing relationship between the two variables.
- $\rho = 0$ indicates a lack of montonicity between the two variables (i.e. a line of best fit has a few twists and turns).
- $\rho = 1$ indicates a monotonic increasing relationship between the two variables.

For example, if our observed outputs were $(0.5, 0.95, 0.3, 0.72)$ and our predicted outputs were $(0.7, 0.89, 0.66, 0.72)$, then $\rho = 1$ since the relative rankings/orderings between the two sets are the same.

Example of Error measures

```
a = c(90, 40, 69, 31, 39, 44, 21, 81, 25, 52)
b = c(64, 42, 65, 4, 55, 42, 18, 79, 46, 62)
# Total Squared Error
sum((a-b)^2)

# Root Mean Squared Error
sqrt(sum((a-b)^2)/length(a))

# Sum of Absolute Errors
sum(abs(a-b))

# Average Absolute Error
sum(abs(a-b))/length(a)

# Correlation
cor(a,b)
#rho
cor(a,b,method = "spearman")
```

Result:

> Correlation

[1] 0.7711654

Near perfect positive relationship

> Spearman correlation

[1] 0.8024353

A monotonic increasing

Using Formulas in R

The variable on the left-hand side of a tilde(~) is called the "dependent variable", while the variables on the right-hand side are called the "independent variables“

you use these R objects (formulas) to express a relationship between variables.

```
x = c(73, 41, 61, 78)
y = c(56, 65, 56, 58)
z = x ~ y
z
class(z)
```

KeiHotel Data

Hotel ID	Kei's reviews	Similar user ID								
		220	817	751	265	656	231	289	345	171
159	56	65	18	56	69	58	53	61	70	50
508	73	41	31	61	78	78	75	83	73	78
457	81	60	100	71	69	66	91	74	100	90
215	83	73	84	90	90	86	81	54	96	89
343	56	80	76	40	43	49	62	38	52	86
299	79	83	76	59	67	75	80	79	87	35
277	92	67	58	80	90	95	93	100	100	100
242	99	69	99	91	100	84	100	100	92	96
2		98	50	62	63	81	72	82	96	51
826		94	99	63	58	96	70	91	59	86
977		59	85	66	55	78	91	87	94	73

Using Plot() in R

We can add a title to our plot with the parameter main. Similarly, xlab and ylab can be used to label the x-axis and y-axis respectively

- We can change the plot type with the argument type
- We can define the colour using “col”.
- PCH symbols can be used for r plots and get values of 1-25.

```
plot(x, sin(x), main="The Sine Function", ylab="sin(x)", type="l", col="blue")
```

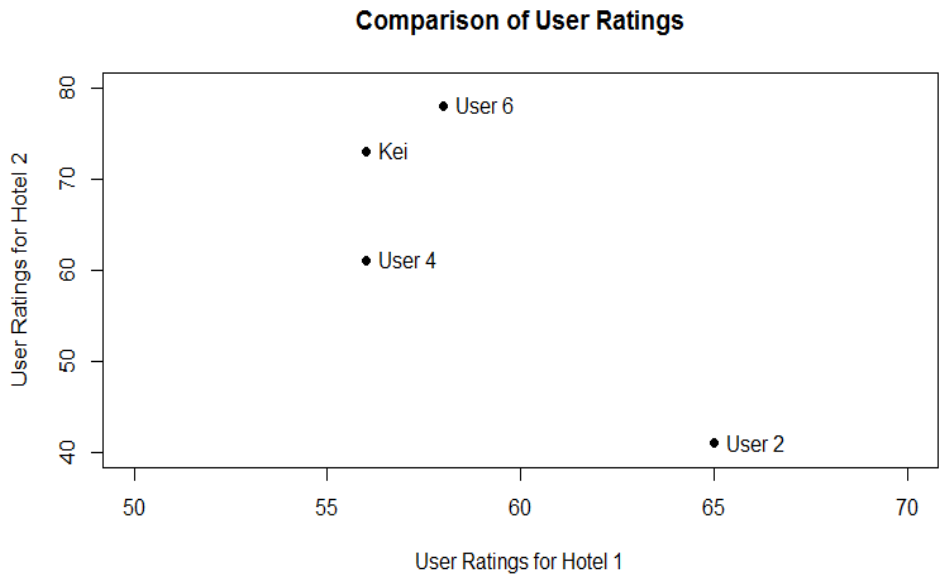
```
plot(z, xlab="User Ratings for Hotel 1", ylab="User Ratings for Hotel 2",
main="Comparison of User Ratings", xlim=c(50,70), ylim=c(40,80), pch=16)
```

Using text() in R

The R function text() allows you to place a specified text string in a particular location on your R plot.

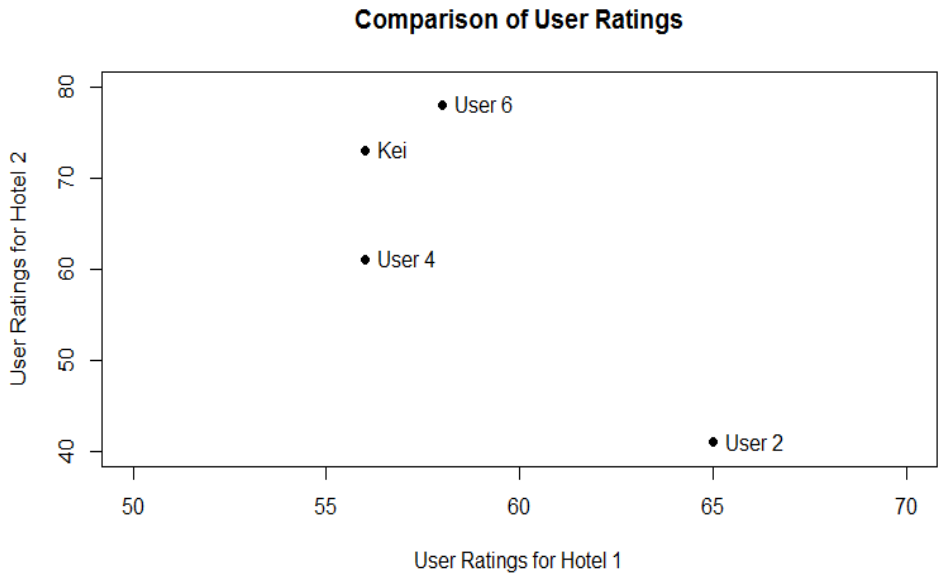
- “cex=1” means that the text will have default size, cex = 1.5 means that the text will have text size that is 1.5 times larger than the default text size in R
- “Pos” can take one of the following values: 1(below), 2(left), 3(above) or 4(right).

```
text(z, label=c("Kei", "User 2", "User 4", "User 6"), pos=4)
```



Plot in R example

```
x = c(73, 41, 61, 78)
y = c(56, 65, 56, 58)
z = x ~ y
z
class(z)
plot(z, xlab="User Ratings for Hotel 1", ylab="User Ratings for Hotel 2", main="Comparison of User Ratings", xlim=c(50,70), ylim=c(40,80), pch=16)
text(z, label=c("Kei", "User 2", "User 4", "User 6"), pos=4)
```



Exploring the Data

Since all variables range between 0 and 100 and there is a positive linear relationship between Kei and all other users, we can divide all values by 100.

Plot histograms for each User (including Kei) to survey their tendencies in terms of distributing ratings.

```
hotelDatanorm = hotelData/100
plot(hotelDatanorm[,1])
typeof(hotelDatanorm)
class(hotelDatanorm)
h1 = hist(hotelDatanorm[,1])
h2 = hist(hotelDatanorm[,2])
plot(h1, col = rgb(1,0,0,1/20))
plot(h2, , col = rgb(0,0,1,1/20), add = T)

summary(hotelDatanorm)
```

Exploring the Data

We measure Euclidean, Manhattan distance along with Pearson and Spearman correlation to identify the which of the users is most similar to Kei?

Euclidean Distance:
distance between each set of coordinates

Manhattan distance:
The **distance** between two points measured along axes at right angles.

```
kei.data = as.matrix(read.table("KeiHotels.txt"))
minkowski <- function(x,y,p=1) (sum(abs(x-y)^p))^(1/p)
manhattan = array(0,9)
euclid = array(0,9)
pearson = array(0,9)
spearman = array(0,9)
for (i in 1:9){
euclid[i]= minkowski(kei.data[,1],kei.data[,i+1],2)}
for (i in 1:9){
  manhattan[i]= minkowski(kei.data[,1],kei.data[,i+1],1)}
for (i in 1:9){
  pearson[i]= cor(kei.data[,1],kei.data[,i+1]) }
for (i in 1:9){
  spearman[i]= cor(kei.data[,1],kei.data[,2],method = "spearman")}
all = rbind(euclid,manhattan,pearson,spearman)
all
```

Using AggWaFit718.r:

```
install.packages("lpSolve")
library(lpSolve)
packageDescription("lpSolve")
source("AggWaFit718.R")
```

The fitting functions in `AggWaFit718.r` takes in data as its input where it treats the last column as our intended output values, and the other columns as input values. It then attempts to find weights for each input by minimising the error measure, which requires optimisation knowledge (not assessed) of which the basics are covered in Course 4.

Weighted Arithmetic Mean:

```
fit.QAM(hotelData[,c(2:10,1)],"WAMoutput.txt",
"WAMstats.txt")
```

Some observations:

- We can't make any comment on the errors until we compare them to other models.
- Pearson and Spearman Correlation are very close to 1, which is promising.
- Highest weights are given to User 5 and User 6 which is consistent with our observations in 2.5 Exploring and Transforming the Data.
- User 8 having zero weight is a little strange, but given how low the weights are for User 2 and User 3, it can be considered reasonably accurate. We'll be aiming to see similar relative results in the other models.

Using AggWaFit718.r:

Power Mean

Using fit.QAM, we find the weights for a weighted arithmetic mean that best approximates Kei's ratings from those of the other users.

```
fit.QAM(hotelData[,c(2:10,1)],"PMoutput.txt", "PMstats.txt", g=PM05, g.inv = invPM05)
```

Quadratic Mean (Power Mean with p=2):

```
fit.QAM(hotelData[,c(2:10,1)],"QMoutput.txt", "QMstats.txt", g=QM, g.inv = invQM)
```

Ordered Weighted Average:

```
fit.OWA(hotelData[,c(2:10,1)],"OWAoutput.txt", "OWAstats.txt")
```

Using AggWaFit718.r:

Choquet Integral:

```
fit.choquet(hotelData[,c(2:10, 1)], "Choquetoutput2.txt", "Choquetstats2.txt")
```

it’s optimising for 2^9=512 variables instead of 9!

Discussions on reliability

Every step of the process can be questioned in several ways.

- Are we evaluating an approach, or just the parameters of the model?
- Do we have a ‘ground truth’ dataset we can use?
- Do we have enough data to make reliable models?
- How can we account for overfitting