Hibernia College

Kevin O'Brien

May 13, 2013

Contents

1 MAY 2012 : Research Notes

Roy (2009) proposes a suite of hypothesis tests for assessing the agreement of two methods of measurement, when replicate measurements are obtained for each item, using a LME approach. (An item would commonly be a patient). Two methods of measurement can be said to be in agreement if there is no significant difference between in three key respects. Firstly, there is no inter-method bias between the two methods, i.e. there is no persistent tendency for one method to give higher values than the other. Secondly, both methods of measurement have the same within-subject variability. In such a case the variance of the replicate measurements would consistent for both methods. Lastly, the methods have equal between-subject variability. Put simply, for the mean measurements for each case, the variances of the mean measurements from both methods are equal.

1.1 Testing for Inter-method Bias

Firstly, a practitioner would investigate whether a significant inter-method bias is present between the methods. This bias is specified as a fixed effect in the LME model. For a practitioner who has a reasonable level of competency in `R` and undergraduate statistics (in particular simple linear regression model) this is a straight-forward procedure.

1.2 Reference Model (Ref.Fit)

Conventionally LME models can be tested using Likelihood Ratio Tests, wherein a reference model is compared to a nested model.

```
> Ref.Fit = lme(y ~ meth-1, data = dat,   #Symm , Symm#
+     random = list(item=pdSymm(~ meth-1)),
+     weights=varIdent(form=~1|meth),
+     correlation = corSymm(form=~1 | item/repl),
+     method="ML")
```

Roy(2009) presents two nested models that specify the condition of equality as required, with a third nested model for an additional test. There three

formulations share the same structure, and can be specified by making slight alterations of the code for the Reference Model.

1.3 Nested Model (Between-Item Variability)

```
> NMB.fit  = lme(y ~ meth-1, data = dat,   #CS , Symm#
+     random = list(item=pdCompSymm(~ meth-1)),
+     correlation = corSymm(form=~1 | item/repl),
+     method="ML")
```

2 Big Data

2.1 Apache Hadoop

Apache Hadoop is an open source software framework this supports data-intensive distributed applications licensed under the Apache v2 license. It enables applications to work with thousands of computational independent computers and petabytes of data. Hadoop was derived from Google's MapReduce and Google File System (GFS) papers. Hadoop is a top-level Apache project being built and used by a global community of contributors, written in the Java programming language. Yahoo! has been the largest contributor to the project, and uses Hadoop extensively across its businesses.

2.2 bigmemory

This project extends the R statistical programming environment. Package bigmemory supports the creation, storage, access, and manipulation of massive matrices. These matrices are allocated to shared memory and may use memory-mapped files. The associated packages provide advanced functionality.

- biganalytics: A library of utilities for big.matrix objects of package bigmemory.
- bigtabulate,
- synchronicity: The R package synchronicity can be useful on it's own for streaming data analyses, but exists primarily to complement the shared-memory capabilities of bigmemory.
- bigalgebra

The Bigmemory Project was awarded the 2010 John M. Chambers Statistical Software Award by the ASA Sections on Statistical Computing and Statistical Graphics.

2.3 Compute Unified Device Architecture (CUDA)

Compute Unified Device Architecture (CUDA) is a parallel computing architecture developed by Nvidia for graphics processing. CUDA is the computing engine in Nvidia graphics processing units (GPUs) that is accessible

to software developers through variants of industry standard programming languages.

3 MapReduce

MapReduce is a framework for processing embarrassingly parallel problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogenous hardware). Computational processing can occur on data stored either in a filesystem (unstructured) or in a database (structured).

Map step The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

Reduce step The master node then collects the answers to all the sub-problems and combines them in some way to form the output the answer to the problem it was originally trying to solve.

MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel though in practice it is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase - provided all outputs of the map operation that share the same key are presented to the same reducer at the same time.

While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than "commodity" servers can handle a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled assuming the input data is still available.

4 Parallel Computing

The simultaneous use of more than one CPU to execute a program. Ideally, parallel processing makes a program run faster because there are more engines (CPUs) running it. In practice, it is often difficult to divide a program in such a way that separate CPUs can execute different portions without interfering with each other.

Most computers have just one CPU, but some models have several. There are even computers with thousands of CPUs. With single-CPU computers, it is possible to perform parallel processing by connecting the computers in a network. However, this type of parallel processing requires very sophisticated software called distributed processing software. Note that parallel processing differs from multitasking, in which a single CPU executes several programs at once. Parallel processing is also called parallel computing.

5 MA4003 Question 9

5.1 Fourier Coefficient

Useful Identities

$$b_n$$

Simply compute b_n for n=3
sin(nt) cos(nt)

$$\int tsin(t)dt$$

$$\int sin(nt)dt = \frac{cos(nt)}{n}$$

$$\int cos(nt)dt = \frac{sin(nt)}{n}$$

$$cos(n\pi) = (-1)^n \qquad (-1^0 = 1)$$

$$sin(n\pi) = 0$$

Lets consider the integral (I) specifically then, divide it at the end. $a_0 = \frac{I}{\pi}$

$$\int_{-\pi}^{\pi} |t|dt = \int_{\pi}^{0} -t|dt + \int_{0}^{\pi} tdt$$

The function is even - so both component values have equal size.

$$I = 2 \times \int_{0}^{\pi} tdt$$

$$I = \left[\frac{t^2}{2}\right]_0^{pi} = \frac{\pi^2}{2}$$

$$a_0 = \frac{I}{\pi} = \frac{\pi^2}{2}$$

- The LOGIT transformations Use of the logit transformation precludes confidence interval boundaries outside the 0-1 interval.
-
-
-

5.2 Two Options

- Wald Type CIs
- PL Type CIs

5.3 Profile Likelihood Confidence Intervals

The Profile-likelihood based confidence intervals methods is described in Venzon and Moolgavkar, Journal of the Royal Statistical Society, Series C vol 37, no.1, 1988, pp. 87-94.

Profile likelihood confidence intervals can be computed for real parameter estimates.

The default confidence intervals for real parameter estimates in the 0-1 interval are based on the standard error and the logit transformation. That is, a 95% confidence interval is computed on the logit estimate, and then these intervals are transformed to the real scale.

6 Google Analytics

6.1 AdWords

6.2 Search Engine Optimisation (SEO)

7 Audience Research

7.1 Audience Retention

- Media Toolkits
- Media Monitoring

7.2 Survival Analysis

7.2.1 3funnell NDRC Des Farrell

8 Quantitative Ecology

8.1 Similarity Measures

- Euclidean Distance (and Squared Euclidean Distance)
- Manhattan Distance (also known as City Block Distance)
- Jaccard Distance
- Cosine Distance
- Chebyshev's Distance
- Mahalanobis Distance

9 Time and Dates

- The builtin `as.Date()` function handles dates (without times);
- the contributed package `chron` handles dates and times, but does not control for time zones;

- `POSIXct` and `POSIXlt` classes allow for dates and times with control for time zones.

10 Databases : SQL

The most important SQL command is SELECT. Since queries are performed using single statements, the syntax of the SELECT command can be quite daunting:

```
SELECT columns or computations
FROM table
WHERE condition
GROUP BY columns
HAVING condition
ORDER BY column [ASC | DESC]
LIMIT offset,count;
```

11 Character Manipulation

R is usually thought of as a language designed for numerical computation, it contains a full complement of functions which can manipulate character data.

- The `nchar()` function can be used to find the number of characters in a character value.
- Character values will be displayed when they are passed to the `print()` function.
- The `cat()` function will combine character values and print them to the screen or a file directly. The `cat()` function coerces its arguments to character values, then concatenates and displays them. This makes the function ideal for printing messages and warnings from inside of functions.

11.1 Regular Expressions in R

Regular expressions are a method of expressing patterns in character values which can then be used to extract parts of strings or to modify those strings in some way. Regular expressions are supported in the R functions `strsplit()`, `grep()`, `sub()`, and `gsub()`, as well as in the `regexp()` and `gregexpr()` functions which are the main tools for working with regular expressions in R.

11.2 Breaking Apart Character Values

The `strsplit()` function can use a character string or regular expression to divide up a character string into smaller pieces. The first argument to`strsplit()` is the character string to break up, and the second argument is the character value or regular expression which should be used to break up the string into parts.

12 Matrices with R

Matrices are a very useful data structure. An inputted vector can be transformed into matrix form before further constructing it as a vector.

Suppose the following vector was scanned in as a vector, but intended as a data frame.

```
2 1.15
3 1.56
5
2

DAT = c(2
```

13 Reading in Data

14 A Brief Introduction to fitting Linear Models

A very commonly used statistical procedure is **simple linear regression**

- `lm()`
- `summary()`

```
Y <- c( )
X <- c( )

plot(X,Y)
cor(X,Y)
lm(Y~X)
```

```
FitA =lm(Y~X)
summary(FitA)
```

Let's look at this summary output in more detail, to see how it is structured. Importantly this object is structur3ed as a list of named components.

```
names(summary(FitA))
class(summary(FitA))
mode(summary(FitA))
str(summary(FitA))
```

The summary of `FitA` is a data object in it's own right. We will save it under the name `Sum.FitA` (N.B. The dot in the name has no particular meaning).

```
Sum.FitA=summary(FitA)
Sum.FitA[1]
Sum.FitA$pvalue
```

Suppose we wish require the p-value for the slope estimate only.(

```
class(Sum.FitA$pvalue)
mode(Sum.FitA$pvalue)
dim(Sum.FitA$pvalue)
```

15 Programming Language Statements

- for
- while
- if
- break
- switch

```
for (i in 1:5)
    {
    cat("This is loop",i, "\n")
    cat("The square of ", i, "is ",i^2,"\n\n")
    }
```

```
> for (i in 1:5)
+     {
+     cat("This is loop",i, "\n")
+     cat("The square of ", i, "is ",i^2,"\n\n")
+     }
This is loop 1
The square of  1 is  1

This is loop 2
The square of  2 is  4

This is loop 3
The square of  3 is  9

This is loop 4
The square of  4 is  16

This is loop 5
The square of  5 is  25
```

15.1 While

```
X=sample(1:20,5)
i=0
while( mean(X) != floor(mean(X)) )
 {

 cat("This is attempt",i,"\n")
 cat("The mean value of X is",mean(X),"\n \n")

 X=sample(1:20,6)
 i = i+1
 }

cat(
 "First Successful Attempt \n
 This is attempt",i,"\n
 The data set is", X , "\n
 The mean value of X is",mean(X),"\n")
```

```
...

This is attempt 10
The mean value of X is 9.166667

This is attempt 11
The mean value of X is 8.833333

This is attempt 12
The mean value of X is 12.33333

This is attempt 13
The mean value of X is 10.16667
```

```
First Successful Attempt

 This is attempt 14

 The data set is 10 15 4 17 18 2

 The mean value of X is 11
```

1) MA4704 Midterms - finalize
2) MA4128

Missing Data
Pairwise Deletion
Listwise Deletion
Multiple Imputation
Replace with the mean value
Bayesian Approach

Missing at Random (MCAR)
Missing Not At Random (MNAR)
Missing Completely at Random (MCAR)
Censored Data
Left Censored Data (Not Part of Course)
Right Censored Data (Not Part of Course)

Classification
Misclassification
Training and Validation
False Positive and False Negative
Confusion Matrix
Specifity and Sensitivity
Accuracy
Recall
Precision
True Error Rate
Apparent Error Rate

---

http://www.jstor.org/discover/10.2307/1266219?uid=2&uid=4&sid=21102180625907
OC or ROC diagram for 2-class problems. (OC = operating characteristic; ROC= recei
Detection, false alarm.

15.2 What Is Discriminant Analysis?

Discriminant analysis is a **classification** method.

- To train (create) a classifier, the fitting function estimates the parameters of a Gaussian distribution for each class.
- To predict the classes of new data, the trained classifier finds the class with the smallest misclassification cost.

15.3 Recall and Precision

In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

Cost

There are two costs associated with discriminant analysis classification: the true misclassification cost per class, and the expected misclassification cost per observation. True Misclassification Cost per Class. Cost(i,j) is the cost of classifying an observation into class j if its true class is i. By default, Cost(i,j)=1 if i =j, and Cost(i,j)=0 if i=j. In other words, the cost is 0 for correct classification, and 1 for incorrect classification. B is a square matrix of size K-by-K when there are K classes.

15.4 Expected Misclassification Cost per Observation

. Suppose you have Nobs observations that you want to classify with a trained discriminant analysis classifier obj. Suppose you have K classes. You place the observations into a matrix Xnew with one observation per row.

Cross Validation

As in all statistical procedures it is helpful to use diagnostic procedures to asses the efficacy of the discriminant analysis.

We use cross-validation to assess the classification probability.

Typically you are going to have some prior rule as to what is an acceptable misclassification rate.

Those rules might involve things like, "what is the cost of misclassification?"

This could come up in a medical study where you might be able to diagnose cancer.

There are really two alternative costs. The cost of misclassifying someone as having cancer when they don't.

This could cause a certain amount of emotional grief.

Sensitivity and specificity

Sensitivity and specificity are statistical measures of the performance of a binary classification test, also known in statistics as classification function. Sensitivity (also called the true positive rate, or the recall rate in some fields) measures the proportion of actual positives which are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition). Specificity measures the proportion of negatives which are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition, sometimes called the true negative rate).

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

Confusion Matrix

In predictive analytics, a table of confusion (sometimes also called a confusion matrix), is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. This allows more detailed analysis than mere proportion of correct guesses (accuracy). Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (that is, when the number of samples in different classes vary greatly). For example,

if there were 95 cats and only 5 dogs in the data set, the classifier could easily be biased into classifying all the samples as cats. The overall accuracy would be 95%, but in practice the classifier would have a 100% recognition rate for the cat class but a 0% recognition rate for the dog class.

There is also the alternative cost of misclassifying someone as not having cancer when in fact they do have it!

The cost here is obviously greater if early diagnosis improves cure rates.

We can evaluate error rates by means of a training sample (to construct thevdiscrimination surface) and a test sample.

An optimistic error rate is obtained by reclassifying the design set: this is known as the **apparent error rate**.

If an independent test sample is used for classifying, we arrive at the true error rate.

The leaving one out method attempts to use as much of the data as possible: for every subset of $n-1$ objects from the given objects, a classifier is designed, and the object omitted is assigned. This leads to the overhead of discriminant analyses, and to tests from which an error rate can be derived.

Another approach to appraising the results of a discriminant analysis is to determine a confusion matrix which is a contingency table (a table of frequencies of cooccurrence) crossing the known groups with the obtained groups.

Calculating precision and recall is actually quite easy. Imagine there are 100 positive cases among 10,000 cases. You want to predict which ones are positive, and you pick 200 to have a better chance of catching many of the 100 positive cases. You record the IDs of your predictions, and when you get the actual results you sum up how many times you were right or wrong. There are four ways of being right or wrong:

- TN / True Negative: case was negative and predicted negative
- TP / True Positive: case was positive and predicted positive
- FN / False Negative: case was positive but predicted negative
- FP / False Positive: case was negative but predicted positive

Makes sense so far? Now you count how many of the 10,000 cases fall in each bucket, say:

Now, your boss asks you three questions:

What percent of your predictions were correct?
You answer: the “accuracy” was (9,760+60) out of 10,000 = 98.2%
What percent of the positive cases did you catch?
You answer: the ”recall” was 60 out of 100 = 60%
What percent of positive predictions were correct?
You answer: the ”precision” was 60 out of 200 = 30%

16 Missing Data

17 What is multiple imputation?

Imputation, the practice of 'filling in' missing data with plausible values, is an attractive approach to analyzing incomplete data. It apparently solves the missing-data problem at the beginning of the analysis. However, a naive or unprincipled imputation method may create more problems than it solves, distorting estimates, standard errors and hypothesis tests, as documented by Little and Rubin (1987) and others.

- `roots()` roots of a polynomial
- `poly()` characteristic equation
- `primes()` generate a sequence of prime numbers
- `isprime()` check for prime numbers
- `any()` check if any value fulfils a logical condition
- `all()` check if all value fulfils a logical condition

Grubbs Test

p: value

The null hypothesis is that both sampmes are drawn from the same population of values.

The test is a non parametric test. Non Parametric tests are a family of inference tests that do not require the assumption of normality.

17.1 Grubbs's Test for Outliers

Grubbs Testnis is used to determine whether or not a particular value in the data set is an ***outlier***. There is no standard definition of what defines an outlier. The definition of an outlier used for this procedure is a value that is numerically distance from the rest of the values of the data set. (We will refer to such an outlier as a ***Grubbs outlier***.

Another type of outlier is a point identified as such by a boxplot. We will refer to this type of outlier as a ***boxplot outlier***. An outlier may be either one or the other type, but not necessarily both.

Some outliers may be due to incorrectly recorded data. Other outliers are correctly recorded, but very unusual, values.

null H_0 : There are no outliers present in the data set.

alt H_1 : There is an outlier present in the data set.

17.1.1 Implementation using `R`

To implement the Grubbs test, we require the package ***outliers***.

17.2 Kolmorgorov Smirnov Test

Two sample KS test

The Kolmorogorov Smirnov Test (aka The KS test) is used to determine whether or not two data sets are from the same distribution. The test is implemented in `R`n using the command `ks.test()`. No packages are required to run this procedure.

```
x=c(3,4,5,1,6)
library(outliers)
grubbs.test(X,two.tailed=T)
```

18 Section 3 Logic

18.1 Logical Operations

- $\neg p$ the negation of proposition p.
- $p \wedge q$ Both propositions p and q are simultaneously true (Logical State AND)
- $p \vee q$ One of the propositions is true, or both (Logical State : OR)
- $p \otimes q$ Only one of the propositions is true (Logical State : exclusive OR (i.e XOR)

p	q	$p \vee q$	$q \wedge p$	$p \otimes q$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

19 Conditional Connectives

Construct the truth table for the proposition $p \rightarrow q$.

p	q	$p \rightarrow q$	$q \rightarrow p$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

20 Section 4 Functions

20.1 Invertible Functions

A function is invertible if it fulfils two criteria

- The function is ***onto***,
- The function is ***one-to-one***.

State the conditions to be satisfied by a function $f : X \leftarrow Y$ for it to have an inverse function $f^{-1} : Y \leftarrow X$.
$\lceil \frac{x^2+1}{4} \rceil$ where $f : A \rightarrow \mathbf{Z}$

(i) Find $f(4)$ and the ancestors of 3.

(ii) Find the range of f.

(iii) Is f invertible? Justify your answer

Given $f : \mathbf{R} \rightarrow \mathbf{R}$ where f(x) =3x-1,define fully the inverse of the function f ,i.e.f^{-1}. State the value of $f^{-1}(2)$

20.2 Precision Functions

- Absolute Value Function $|x|$
- Ceiling Function $\lceil x \rceil$
- Floor Function $\lfloor x \rfloor$

Question1.2: State the range and domain of the following function

$$F(x) = \lfloor x - 1 \rfloor$$

20.3 Powers

$$2^4 = 2 \times 2 \times 2 \times 2 = 16$$

$$5^3 = 5 \times 5 \times 5 = 125$$

20.3.1 Special Cases

Anything to the power of zero is always 1

$$X^0 = 1 \text{ for all values of X}$$

Sometimes the power is a negative number.

$$X^{-Y} = \frac{1}{X^Y}$$

Example

$$2^{-3} = \frac{1}{2^3} = \frac{1}{8}$$

20.4 Exponentials Functions

$$e^a \times e^b = e^{a+b}$$

$$(e^a)^b = e^{ab}$$

20.5 Logarithmic Functions

20.5.1 Laws for Logarithms

The following laws are very useful for working with logarithms.

1. $\log_b(X) + \log_b(Y) = \log_b(X \times Y)$
2. $\log_b(X)$ - $\log_b(Y) = \log_b(X/Y)$
3. $\log_b(X^Y) = Y\log_b(X)$

Question1.3 Compute the Logarithm of the following

- $\log_2(8)$
- $\log_2(\sqrt{128})$
- $\log_2(64)$
- $\log_5(125) + \log_3(729)$
- $\log_2(64/4)$

21 graph theory

Given the following definitions for simple, connected graphs:

- K_n is a graph on n vertices where each pair of vertices is connected by an edge;
- C_n is the graph with vertices $v_1, v_2, v_3, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots \{v_n, v_1\}$;
- W_n is the graph obtained from C_n by adding an extra vertex,v_{n+1}, and edges from this to each of the original vertices in C_n.

(a) Draw K_4, C_4, and W_4.

22 Digraphs and Relatiosn

Given a flock of chickens, between any two chickens one of them is dominant. A relation, R, is defined between chicken x and chicken y as xRy if x is dominant over y. This gives what is known as a pecking order to the flock. Home Farm has 5 chickens: Amy, Beth, Carol, Daisy and Eve, with the following relations:

- Amy is dominant over Beth and Carol
- Beth is dominant over Eve and Carol
- Carol is dominant over Eve and Daisy
- Daisy is dominant over Eve, Amy and Beth
- Eve is dominant over Amy.

23 Counting

Given S is the set of all 5 digit binary strings, E is the set of a 5 digit binary strings beginning with a 1 and F is the set of all 5 digit binary strings ending with two zeroes.

(a) Find the cardinality of S, E and F.

(b) Draw a Venn diagram to show the relationship between the sets S, E and F. Show the relevant number of elements in each region of your diagram.

MONDAY 15 April
1. MS4024 MATLAB (Project Euler) - DONE
2. MS4024 R (Complete Course) - Week 12 DONE
3. MA4128 - Paper Submit (Joe Lynch) - DONE
4. MA4128 - Practical Exam Sample Paper FORWARD
5. MA4704 11A
   Update Sample Papers(DONE)
   Re-publish 10B/C (DONE)
   HTs and CIs for Proportions (DONE)
   sample size estimation for proportions (Corrections)
6. HibColl Review Sessions 3 and 4
7. Corrections for MTs

TUESDAY 16 April
1. MA4128 Discriminant Analysis + Missing Data
2. HibColl Review Sessions 5 and 6
3. Update Sample Questions for MA4128

WEDNESDAY 17 April
1. Dentist (09:30) DONE
2. Dublin R (17:30) and Python Ireland (18:30) DONE
3. MA4704 Tutorials Prep (Proportions + p-value prcoedures)
4. MT2 MA4704 Corrections (Some Done)
5. MS4024 Payments
6. Method Comparison Studies

FRIDAY 19 April
1. Kubrick Night (Cant Make It)
2.Julia Language - Learn
3. Coursera Downloads
4. Prof. Nial Friel (4pm)

24 Useful MATLAB Functions

- `unique()`
- `primes()`
- `size()`
- `factor()`

```
factor(223)
```

24.1 For Loops in MATLAB

```
for( )
```

24.2 While Loops in MATLAB

`while` operator allows a set of commands to be repeated until a specific logical condition is met.

```
while( )
```