Chapter 1

influence.ME

Introduction to Influence Analysis

Outliers and detection of influential observations is an important step in the analysis of a data set. There are several ways of evaluating the influence of perturbations in the data set and in the model given the parameter estimates.

The basic rationale behind measuring influential cases is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.

1.0.1 About influence.ME

influence.ME is an R package for detecting influential data in multilevel regression models (or, mixed effects models as they are referred to in the R community). The application of multilevel models has become common practice, but the development of diagnostic tools has lagged behind. Hence, we developed influence.ME, which calculates standardized measures of influential data for the point estimates of generalized multilevel models, such as DFBETAS, Cooks distance, as well as percentile change and a test for changing levels of significance. influence.ME calculates these measures of influence while accounting for the nesting structure of the data. A paper detailing this package was published in the R Journal (available from the R Journal (.PDF) and my researchgate.net profile).

influence.ME depends on lme4. As the authors of lme4 have completely revised the inner workings of lme4 and are currently releasing version 1.0,

Usage

```
influence(model, group=NULL, select=NULL, obs=FALSE,
gf="single", count = FALSE, delete=TRUE, ...)
```

The `influence()` function was known as the `estex()` command in previous versions of the influence.ME pacakge

1.1 influence.ME: Tools for detecting influential data in mixed effects models

- influence.ME provides a collection of tools for detecting influential cases in generalized mixed effects models.
- It analyses models that were estimated using lme4.
- The basic rationale behind identifying influential data is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.
- To standardize the assessment of how influential a (single group of) observation(s) is, several measures of influence are common practice, such as DFBETAS and Cook's Distance. In addition, we provide a measure of percentage change of the fixed point estimates and a simple procedure to detect changing levels of significance.
- Despite the increasing popularity of multilevel regression models, the development of diagnostic tools lagged behind. Typically, in the social sciences multilevel regression models are used to account for the nesting structure of the data, such as students in classes, migrants from origin-countries, and individuals in countries. The strength of multilevel models lies in analyzing data on a large

number of groups with only a couple of observations within each group, such as for instance students in classes.

- Nevertheless, in the social sciences multilevel models are often used to analyze data on a limited number of groups with per group a large number of observations. A typical example would be the analysis of data on individuals nested within countries. By nature, only a limited number of countries exists. In practice, typical country-comparative analyses are based on about 25 countries. With such a small number of groups (e.g. countries), observations on a single group can easily be overly influential to the outcomes. This means that the conclusions based on the multilevel regression model could no longer hold when a single group is removed from the data.

- In our recent publication in the R Journal, we introduce influence.ME, software that provides tools for detecting influential data in multilevel regression models (or: in mixed effects models, as these are commonly referred to in statistics). influence.ME is a publically available R package that evaluates multilevel regression models that were estimated with the lme4.0 package. It calculates standardized measures of influential data for the point estimates of generalized mixed effects models, such as DFBETAS, Cooks distance, as well as percentile change and a test for changing levels of significance. influence.ME calculates these measures of influence while accounting for the nesting structure of the data. The package and measures of influential data are introduced, a practical example is given, and strategies for dealing with influential data are suggested.

- With this publication, and of course with the software that was available for quite some time, we hope to contribute to a better usage of multilevel regression models. The provided example and guidelines were geared towards applications in the social sciences, but are applicable in all disciplines.

1.2 Influence Measures with *influence.ME*

influence.ME allows you to compute measures of influential data for mixed effects models generated by lme4.

influence.ME provides a collection of tools for detecting influential cases in generalized mixed effects models. It analyses models that were estimated using lme4. The basic rationale behind identifying influential data is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.

To standardize the assessment of how influential a (single group of) observation(s) is, several measures of influence are common practice, such as DFBETAS and Cook's Distance. In addition, we provide a measure of percentage change of the fixed point estimates and a simple procedure to detect changing levels of significance.

`influence()` is the workhorse function of the influence.ME package. Based on a priorly estimated mixed effects regression model (estimated using lme4), the `influence()` function iteratively modifies the mixed effects model to neutralize the effect a grouped set of data has on the parameters, and which returns returns the fixed parameters of these iteratively modified models. These are used to compute measures of influential data.

- `influence.ME` provides a collection of tools for detecting influential cases in generalized mixed effects models.
- It analyses models that were estimated using lme4. The basic rationale behind identifying influential data is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.
- To standardize the assessment of how influential a (single group of) observation(s) is, several measures of influence are common practice, such as DFBETAS and Cook's Distance.
- In addition, we provide a measure of percentage change of the fixed point estimates and a simple procedure to detect changing levels of significance.

influence.ME calculates measures of influence for mixed effects models estimated with the lme4 R package. The basic rationale behind measuring influential cases is that when iteratively single units are omitted from the data, models based on these data should not produce substantially different estimates.

Calculating measures of influential data for an LME model requires the re-estimation of this model for each set of potentially influential data separately. The `estex(` function does this, and returns the altered estimates resulting from each re-estimation.

The main function in the influence.ME package is the `influence()`.

> Based on a priorly estimated mixed effects regression model (estimated using lme4), the `influence()` function iteratively modifies the mixed effects model to neutralize the effect a grouped set of data has on the parameters, and which returns returns the fixed parameters of these iteratively modified models. These are used to compute measures of influential data. (Nieuwenhuis et al, 2014)

The `estex()` function requires the specification of two parameters:

1. a mixed effects model is to be specified,
2. the grouping factor of which the influence of the nested observations are to be evaluated.

1.2.1 Functionality of the influence.ME pacakge

The R package influence.ME allows for the calculation measures of influential data for mixed effects models generated by lme4. To standardize the assessment of how influential an observation (or group of observations)is, several commonly encountered measures of influence are used by **influence.ME**.

- DFBETAS is a standardized measure of the absolute difference between the estimate with a particular case included and the estimate without that particular case.
- Cooks distance provides an overall measurement of the change in all parameter estimates, or a selection thereof.

The `estex()` command computes revised estimates can subsequently be entered to the `cooks.distance` and `dfbetas` commands, to calculate Cooks Distance and the DFBETAS (standardized difference of the beta) measures.

Example

```
library(lme4)
model <- lmer(mpg ~ disp + (1 | cyl), mtcars)

#The function influence is the
# basis for all further steps:

library(influence.ME)
infl <- influence(model, obs = TRUE)

# Calculate Cook's distance:
cooks.distance(infl)

# Can Plot Cook's distance:

plot(infl, which = "cook")
```

The `pchange` command

The `pchange` command computes the percentile change, as a measure of influential data. This unstandardized measure can serve to help interpret the magnitude of the influence single or combined grouping levels exert on mixed effects models.

The percentage change in parameter estimates between an LME model based on a full set of data, and a model from which a (potentially influential) subset of data is removed. A value of percentage change is calculated for each parameter in the model separately, based on the information returned by the `estex()` function.

sigtest

The `sigTest()` function can test for changes in the level of statistical significance resulting from the deletion of potentially influential observations

The `plot.estex` command

This is a wrapper function to the `dotplot()` function in the **lattice R** package.

1.3 Computing DFBETAs with `R`

- This function computes the DFBETAS based on the information returned by the estex() function.
- The dfbeta refers to how much a parameter estimate changes if the observation or case in question is dropped from the data set.
- Cook's distance is presumably more important to you if you are doing predictive modeling, whereas dfbeta is more important in explanatory modeling.
- The DFBETAS statistics are the scaled measures of the change in each parameter estimate and are calculated by deleting the th observation:

  Missing Formula

  where is the th element of . In general, large values of DFBETAS indicate observations that are influential in estimating a given parameter.
- **Belsley, Kuh, and Welsch (1980)** recommend 2 as a general cutoff value to indicate influential observations and as a size-adjusted cutoff.

The `R` programming language has a variety of methods used to study each of the aspects for a linear model. While linear models and GLMS can be studied with a wide range of well-established diagnostic technqiues, the choice of methodology is much more restricted for the case of LMEs.

For an `lme` object, such as our fitted model `JS.roy1`, the predicted values for each subject can be determined using the `coef.lme` function.

```
> JS.roy1 %>% coef %>% head(5)
methodJ   methodS
74     84.31724  91.08404
```

```
36     91.54994  97.05548
3      81.16581  96.48653
62     92.09493  90.89073
31     88.41411 103.38802
```

The `CookD` fucntion , from the predictmeans R package, produces Cooks distance plots for an LME model (***predictmeans***)

```
library(predictMeans)
CookD(model, group=method, plot=TRUE, idn=5, newwd=FALSE)
```

1.4 DFbetas for Blood Data

```
plot(JS.ARoy20091.dfbeta$all.res1[1:255],JS.ARoy20091.dfbeta$all.res2[256:510],
pch=16,col="blue")
abline(v=JS.ARoy20091.dfbeta$all.res1[256],col="red")
abline(h=JS.ARoy20091.dfbeta$all.res2[1],col="red")
```

1.5 The `logLik` Function

`logLik.lme` returns the log-likelihood value of the linear mixed-effects model represented by object evaluated at the estimated coefficients. It is also possible to determine the restricted log-likelihood, if relevant, using this function. For the Blood Data Example, the loglikelihood of the JS.roy1 model can be computed as follows.

```
> logLik(JS.roy1)
'log Lik.' -2030.736 (df=8)
```

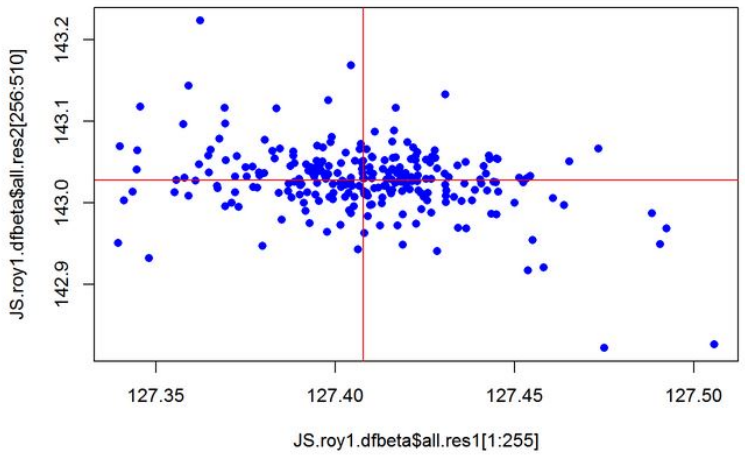

Figure 1.1:

1.5.1 Identifying outliers with a LME model object

The process is slightly different than with standard LME model objects, since the ***influence*** function does not work on lme model objects. Given ***mod.lme***, we can use the plot function to identify outliers.

1.6 Partitioning Matrices

Without loss of generality, matrices can be partitioned as if the $i-$th omitted observation is the first row; i.e. $i = 1$.

Using the influence.ME package

Influence Analysis can onnly be carried out with LME models fitted using the functions in the **lme4** package. Such models are known as `mer` objects. Hence the `estex()` function only works on LME models of class `mer`. The package developers advise that it is required that the `mer` model was estimated using a factor variable to indicate group levels. When using something similar to `+ (1 | as.factor(variable))`, the function is not able of identifying the correct grouping factors, and returns an error.

Executing this procedure can be computationally highly demanding, because `estex()` entails the re-estimation of the provided mixed effects model for each level of the specified grouping factor (after alteration of the data).

1.7 Leave-One-Out Diagnostics with `lmeU`

Galecki et al provide a brief the matter of LME influence diagnostics in their book.

The command `lmeU` fits a model with a particular subject removed. The identifier of the subject to be removed is passed as the only argument

A plot ofthe per-observation diagnostics individual subject log-likelihood contributions can be rendered.

To apply the same logic to mixed effects models one has to measure the influence of a particular higher level unit on the estimates of a higher level predictor.

This means that the mixed effect model has to be adjusted to neutralize the units influence on that estimate, while at the same time allowing the units lower-level cases to help estimate the effects of the lower-level predictors in the model.

This procedure is based on a modification of the intercept and the addition of a dummy variable for the cases that might be influential.

Influence.ME provides several measures of influential cases, and is specifically designed for use with mixed effects regression models using the afore mentioned modified intercept and dummy approach.

Using both real and simulated data from Social Science applications of mixed effects models, five tools to detect influential cases which are available in the package will be discussed:

- Cooks Distance
- DFBETAS
- Percent change of the estimated parameter magnitude
- Changes in statistical significance of parameter estimates
- Changes in the sign of parameter estimates

In contrast with other algorithms for detecting influential cases, influence.ME is capable to uncover groups of cases that are influential. Since this rapidly becomes computationally highly intensive, additional script functions are provided that assist in manually dividing the computation into multiple sessions, or to possibly to share the computations between different computers.

Other Material

1.7.1 The school23 Example

In a published tutorial, Nieuwenhuis et al provide an example using a data set that is provided with the **influence.ME** `R` package. The **school23** data contains information on students performance on a math test, as well as several explanatory variables. These data are subset of the NELS-88 data (National Education Longitudinal Study of 1988). Both a selected number of variables and a selected number of observations are given.

The *school23* data contains information on a math test performance of 519 students, who are nested within 23 schools. For this example, analysts will be interested in the relationship between class structure (in this data measured at the school level) and students performance on a math test. The research question is: *To what extend does the classroom structure determine the students math test outcomes?*

Initially, we will estimate the effect of class structure on the result of the math performance test, without any further covariates. We do take into account the nesting structure of the data, however, and allow the intercept to be random over schools. This model is estimated using the following syntax, and is assigned to an object we call model.

```
model <- lmer(math ~  structure + (1 | school.ID), data=school23)
summary(model)
```

The call for a `summary` of the model results in the output shown below. In this summary, the original model formula is shown, as well as the data on which this model was estimated. Both random and fixed effects are summarized. The amount of intercept variance associated with the nesting structure of students within schools is considerably large (23.8 compared with 81.2 + 23.8 = 104 in total). The effect of interest is that of the structure variable, which is -2.343 and statistically insignificant by most reasonable standards (t=-1.609).

```
Linear mixed model fit by REML
Formula: math ~ structure + (1 | school.ID)
Data: school23
AIC  BIC logLik deviance REMLdev
3802 3819  -1897     3798    3794

Random effects:
Groups    Name        Variance Std.Dev.
school.ID (Intercept) 23.884   4.8871
Residual              81.270   9.0150
Number of obs: 519, groups: school.ID, 23

Fixed effects:
Estimate Std. Error t value
(Intercept)   60.002      5.853  10.252
structure     -2.343      1.456  -1.609

Correlation of Fixed Effects:
(Intr)
structure -0.982
```

In the syntax example below, the original object 'model' is specified, and 'school.ID' is the relevant grouping factor. school.ID is the name of the variable used to indicate the grouping factor when the original model was specified. The estex() function works perfectly when more than a single grouping is present in the model, but only one grouping factor can be addressed at once.

```
data(school23)
model <- lmer(math ~ structure + SES + (1 | school.ID), data=school23)
alt.est <- influence(model, group="school.ID")
cooks.distance(alt.est)
```

1.8 Permutation Test, Power Tests and Missing Data

This section explores topics such as dependent variable simulation and power analysis, introduced by Galecki & Burzykowski (2013), and implementable with their ***nlmeU*** `R` package. Using the ***predictmeans*** `R` package, it is possible to perform permutation t-tests for coefficients of (fixed) effects and permutation F-tests.

The matter of missing data has not been commonly encountered in either Method Comparison Studies or Linear Mixed Effects Modelling. However Roy (2009) deals with the relevant assumptions regrading missing data. Galecki & Burzykowski (2013) approaches the subject of missing data in LME Modelling. The ***nlmeU*** package includes the `patMiss` function, which "*allows to compactly present pattern of missing data in a given vector/matrix/data frame or combination of thereof*".

Contents

1.9 Profile Function with "lmer"

The profile() function for lmer models is now available in the latest version of lme4, to be installed by typing:

install.packages("lme4",repos="http://r-forge.r-project.org")

also

The mle function from the stats4 package is a wrapper of optim, which makes it quite easy to produce profile likelihood computations.

See help("profile,mle-method", package = "stats4") for more information.

http://people.upei.ca/hstryhn/stryhn208.pdf

The profile likelihood (or likelihood or likelihood ratio) methid is applicable to all likelihood based statstical analysis and is generally less sensitive to the difficulties encountered by walkd-Tyoe CIs.