Out-of-sample prediction from mixed models in R using 'mixoutsamp'

Example R code using repeated-measures data on pig weights

Ruth H Keogh, Jessica K Barrett, Mike Sweeting

There does not appear to be an existing way of obtaining out-of-sample predictions from mixed models in R. I created the R function 'mixoutsamp', which provides out-of-sample predictions from mixed models fitted using lme in R. With thanks to Jessica Barrett and Mike Sweeting for contributing to this work.

The code for 'mixoutsamp' is provided in a separate file (mixoutsamp_v2.R. This document provides some examples to illustrate how 'mixoutsamp' can be used, using a data set called pigs.weights. This data set is freely available in R (in the SemiPar package) and contains 9 repeated weight measurements on 48 pigs.

Preliminaries

We start by loading the libraries needed and by loading the data.

```
library(nlme)
library(Matrix)
library(SemiPar) #this package contains the data set pig.weights

data(pig.weights)
pigs=pig.weights #rename the data
names(pigs)=c("id","week","weight")

head(pigs)
```

```
##   id week weight
## 1  1    1   24.0
## 2  1    2   32.0
## 3  1    3   39.0
## 4  1    4   42.5
## 5  1    5   48.0
## 6  1    6   54.5
```

Next, run the code for 'mixoutsamp'.

```
source("./mixoutsamp_v2.R")
```

Separate the pigs data into an 'in-sample' part, which will be used to fit the mkxed models, and an 'out-of-sample' part, which will be used to illustrate the use of 'mixoutsamp'.

```
pigs.insample=pigs[pigs$id<=40,] #the first 40 pigs
pigs.outsample=pigs[pigs$id>40,] #the remaining 8 pigs
```

Example 1

In this example we fit a mixed model (with random intercept and slope) to the pigs.insample data. The model is used to obtain predicted values on the same pigs on which the model was fitted (pigs.insample)

using the standard 'predict' function and using our code 'mixoutsamp'. In this example we do not need to use 'mixoutsamp' because 'predict' gives in-sample predictions, but this is used to illustrate that 'mixoutsamp' gives the same results.

```
mixmod1=lme(weight~week,random=~1+week|id,data=pigs) # model with random intercept and slope

#---
#obtain fitted values
#---
pred.mixmod1=mixoutsamp(model=mixmod1,newdata=pigs) #using mixoutsamp
testpred.mixmod1=predict(mixmod1,newdata=pigs) #using predict

#---
#Compare fitted values
#---
head(pred.mixmod1$preddata) #using mixoutsamp
```

```
##   id week weight    fixed     random   fitted
## 1  1    1   24.0 25.56551 -0.1561810 25.40933
## 2  1    2   32.0 31.77541 -0.5525862 31.22282
## 3  1    3   39.0 37.98530 -0.9489914 37.03631
## 4  1    4   42.5 44.19520 -1.3453966 42.84980
## 5  1    5   48.0 50.40509 -1.7418018 48.66329
## 6  1    6   54.5 56.61499 -2.1382070 54.47678
```

```
head(testpred.mixmod1) #using predict
```

```
##        1        1        1        1        1        1
## 25.40933 31.22282 37.03631 42.84980 48.66329 54.47678
```

```
head(mixmod1$fitted) #directly from the model
```

```
##      fixed       id
## 1 25.56551 25.40933
## 2 31.77541 31.22282
## 3 37.98530 37.03631
## 4 44.19520 42.84980
## 5 50.40509 48.66329
## 6 56.61499 54.47678
```

```
#---
#Compare random effects (random intercept and slope)
#---
head(pred.mixmod1$random)#using mixoutsamp
```

```
##   id      reff1      reff2
## 1  1  0.2402242 -0.3964052
## 2  2 -1.5915189  0.5113588
## 3  3 -3.5374571  0.3218440
## 4  4  1.9744930 -0.7738018
## 5  5  1.3087411 -0.9259342
## 6  6 -1.1464335 -0.5451292
```

```
head(as.data.frame(mixmod1$coefficients$random$id)) #directly from the model
```

```
##   (Intercept)       week
## 1   0.2402242 -0.3964052
## 2  -1.5915189  0.5113588
```

```
## 3  -3.5374571  0.3218440
## 4   1.9744930 -0.7738018
## 5   1.3087411 -0.9259342
## 6  -1.1464335 -0.5451292
```

Example 2

In this example we fit a mixed model (with random intercept and slope) to the pigs.insample data. The model is used to obtain predicted values on a new set of pigs in the data set pigs.outsample.

```
mixmod2=lme(weight~week,random=~1+week|id,data=pigs.insample,na.action=na.omit)

#---
#first note that using predict doesn't work for the out-of-sample pigs
#---
testpred.mixmod2=predict(mixmod2,newdata=pigs.outsample)
head(testpred.mixmod2)
```

```
## 41 41 41 41 41 41
## NA NA NA NA NA NA
```

```
#---
#using mixoutsamp to obtain out-of-sample predictions
#---
pred.mixmod2=mixoutsamp(model=mixmod2,newdata=pigs.outsample)
pred.mixmod2$preddata #fitted values
```

```
##     id week weight    fixed      random   fitted
## 361 41    1   26.5 25.12139 -0.65507707 24.46631
## 362 41    2   30.5 31.30910 -1.71978386 29.58931
## 363 41    3   33.0 37.49681 -2.78449065 34.71231
## 364 41    4   39.0 43.68451 -3.84919744 39.83532
## 365 41    5   43.5 49.87222 -4.91390423 44.95832
## 366 41    6   49.5 56.05993 -5.97861102 50.08132
## 367 41    7   56.5 62.24764 -7.04331781 55.20432
## 368 41    8   61.0 68.43535 -8.10802460 60.32732
## 369 41    9   65.0 74.62306 -9.17273139 65.45032
## 370 42    1   24.0 25.12139  0.41093709 25.53233
## 371 42    2   32.0 31.30910  0.33694496 31.64604
## 372 42    3   39.0 37.49681  0.26295284 37.75976
## 373 42    4   44.5 43.68451  0.18896072 43.87347
## 374 42    5   50.0 49.87222  0.11496860 49.98719
## 375 42    6   56.0 56.05993  0.04097648 56.10091
## 376 42    7   63.0 62.24764 -0.03301564 62.21462
## 377 42    8   67.5 68.43535 -0.10700776 68.32834
## 378 42    9   74.0 74.62306 -0.18099988 74.44206
## 379 43    1   24.5 25.12139  0.12924826 25.25064
## 380 43    2   31.0 31.30910 -0.15025887 31.15884
## 381 43    3   37.5 37.49681 -0.42976600 37.06704
## 382 43    4   43.5 43.68451 -0.70927313 42.97524
## 383 43    5   48.0 49.87222 -0.98878026 48.88344
## 384 43    6   56.0 56.05993 -1.26828739 54.79164
## 385 43    7   62.5 62.24764 -1.54779452 60.69984
## 386 43    8   66.5 68.43535 -1.82730166 66.60805
```

```
## 387 43    9   70.5 74.62306 -2.10680879 72.51625
## 388 44    1   27.0 25.12139  3.14957793 28.27097
## 389 44    2   34.5 31.30910  3.23159358 34.54069
## 390 44    3   42.0 37.49681  3.31360922 40.81041
## 391 44    4   48.5 43.68451  3.39562486 47.08014
## 392 44    5   53.0 49.87222  3.47764050 53.34986
## 393 44    6   60.0 56.05993  3.55965614 59.61959
## 394 44    7   67.0 62.24764  3.64167179 65.88931
## 395 44    8   73.0 68.43535  3.72368743 72.15903
## 396 44    9   76.0 74.62306  3.80570307 78.42876
## 397 45    1   31.0 25.12139  6.94231937 32.06371
## 398 45    2   39.0 31.30910  7.07538744 38.38448
## 399 45    3   47.5 37.49681  7.20845551 44.70526
## 400 45    4   51.0 43.68451  7.34152358 51.02604
## 401 45    5   57.0 49.87222  7.47459165 57.34681
## 402 45    6   64.0 56.05993  7.60765972 63.66759
## 403 45    7   71.0 62.24764  7.74072779 69.98837
## 404 45    8   77.0 68.43535  7.87379586 76.30914
## 405 45    9   80.5 74.62306  8.00686393 82.62992
## 406 46    1   27.0 25.12139  1.68718378 26.80857
## 407 46    2   33.5 31.30910  2.09071888 33.39982
## 408 46    3   40.0 37.49681  2.49425399 39.99106
## 409 46    4   46.5 43.68451  2.89778909 46.58230
## 410 46    5   53.0 49.87222  3.30132420 53.17355
## 411 46    6   60.0 56.05993  3.70485930 59.76479
## 412 46    7   66.5 62.24764  4.10839441 66.35603
## 413 46    8   72.5 68.43535  4.51192951 72.94728
## 414 46    9   80.0 74.62306  4.91546462 79.53852
## 415 47    1   29.5 25.12139  5.01288788 30.13428
## 416 47    2   37.0 31.30910  6.19988237 37.50898
## 417 47    3   46.0 37.49681  7.38687687 44.88368
## 418 47    4   52.5 43.68451  8.57387136 52.25839
## 419 47    5   60.0 49.87222  9.76086585 59.63309
## 420 47    6   67.5 56.05993 10.94786034 67.00779
## 421 47    7   76.0 62.24764 12.13485483 74.38249
## 422 47    8   81.5 68.43535 13.32184933 81.75720
## 423 47    9   88.0 74.62306 14.50884382 89.13190
## 424 48    1   28.5 25.12139  2.99183904 28.11323
## 425 48    2   36.0 31.30910  3.93511546 35.24421
## 426 48    3   42.5 37.49681  4.87839187 42.37520
## 427 48    4   49.0 43.68451  5.82166829 49.50618
## 428 48    5   55.0 49.87222  6.76494471 56.63717
## 429 48    6   63.5 56.05993  7.70822112 63.76815
## 430 48    7   72.0 62.24764  8.65149754 70.89914
## 431 48    8   78.5 68.43535  9.59477395 78.03012
## 432 48    9   85.5 74.62306 10.53805037 85.16111
```

```
pred.mixmod2$random #random effects
```

```
##   id     reff1       reff2
## 1 41 0.4096297 -1.06470679
## 2 42 0.4849292 -0.07399212
## 3 43 0.4087554 -0.27950713
## 4 44 3.0675623  0.08201564
## 5 45 6.8092513  0.13306807
```

```
## 6 46 1.2836487  0.40353510
## 7 47 3.8258934  1.18699449
## 8 48 2.0485626  0.94327642
```

Example 3

In this example we illustrate the use of 'mixoutsamp' when an exponential within-person correlation structure is specified (see the mixoutsamp file for other within-person correlation structures that are supported).

```
mixmod3=lme(weight~week,random=~1+week|id,data=pigs.insample,corr=corExp(form = ~ week|id),na.action=na
            control=lmeControl(maxIter=1000,msMaxIter=1000,opt="optim"))
#---
#---
#first obtaining in-sample predictions for pigs.insample
#---
#---

pred.mixmod3=mixoutsamp(model=mixmod3,newdata=pigs.insample) #using mixoutsamp
testpred.mixmod3=predict(mixmod3,newdata=pigs.insample) #using predict

#---
#Compare fitted values
#---
head(pred.mixmod3$preddata) #using mixoutsamp
```

```
##   id week weight    fixed     random   fitted
## 1  1    1   24.0 24.75184 -0.4380224 24.31382
## 2  1    2   32.0 31.00994 -0.6631228 30.34682
## 3  1    3   39.0 37.26805 -0.8882233 36.37982
## 4  1    4   42.5 43.52615 -1.1133237 42.41283
## 5  1    5   48.0 49.78425 -1.3384241 48.44583
## 6  1    6   54.5 56.04236 -1.5635246 54.47883
```

```
head(testpred.mixmod3) #using predict
```

```
##        1        1        1        1        1        1
## 24.31382 30.34682 36.37982 42.41283 48.44583 54.47883
```

```
head(mixmod3$fitted) #directly from the model
```

```
##      fixed       id
## 1 24.75184 24.31382
## 2 31.00994 30.34682
## 3 37.26805 36.37982
## 4 43.52615 42.41283
## 5 49.78425 48.44583
## 6 56.04236 54.47883
```

```
#---
#Compare random effects (random intercept and slope)
#---
head(pred.mixmod3$random)#using mixoutsamp
```

```
##   id       reff1       reff2
## 1  1 -0.21292197 -0.22510043
```

```
## 2  2  0.25260166  0.30148460
## 3  3 -0.01498913  0.03538531
## 4  4 -0.46365701 -0.51768873
## 5  5 -0.62060311 -0.68472344
## 6  6 -0.45808686 -0.46452938
```

```
head(as.data.frame(mixmod3$coefficients$random$id)) #directly from the model
```

```
##   (Intercept)        week
## 1 -0.21292197 -0.22510043
## 2  0.25260166  0.30148460
## 3 -0.01498913  0.03538531
## 4 -0.46365701 -0.51768873
## 5 -0.62060311 -0.68472344
## 6 -0.45808686 -0.46452938
```

```
#---
#---
#also obtaining out-of-sample predictions for pigs.outsample
#---
#---

pred.outsample.mixmod3=mixoutsamp(model=mixmod3,newdata=pigs.outsample) #using mixoutsamp
```

Example 4

In this example we illustrate using mixoutsamp for a setting with multivariate response. For illutration, we created a new variable 'height', which is measured alongside weight at each week (this is artificial variable - I do not really know how tall pigs are).

```
#---
#create a stacked data set with two response variables (weight and height)
#---
pigs.multivar=rbind(pigs,pigs)
pigs.multivar$response.type=rep(1:2,each=dim(pigs)[1])
pigs.multivar$height=rep(rnorm(dim(pigs)[1],70+0.5*pigs$weight,5),2) #generating 'height'
pigs.multivar$response=ifelse(pigs.multivar$response.type==1,pigs.multivar$weight,pigs.multivar$height)

#---
#create in-sample and out-of-sample data sets as before
#---
pigs.multivar.insample=pigs.multivar[pigs.multivar$id<=40,] #the first 40 pigs
pigs.multivar.outsample=pigs.multivar[pigs.multivar$id>40,] #the remaining 8 pigs
```

The model is a multivariate mixed model with height and weight as two responses. We allow a random intercept and slope for each response type.

```
#---
#fit the mixed model with interactions by response type and separate residual variance by response type
#---
mixmod4=lme(response~week*response.type,random=~1+week*response.type|id,weights=varIdent(form=~1|respons
            control=lmeControl(maxIter=1000,msMaxIter=1000,opt="optim"))

#---
#first obtaining in-sample predictions for pigs.insample
```

```
#---

pred.mixmod4=mixoutsamp(model=mixmod4,newdata=pigs.multivar.insample) #using mixoutsamp
testpred.mixmod4=predict(mixmod4,newdata=pigs.multivar.insample) #using predict

#Compare fitted values
head(pred.mixmod4$preddata) #using mixoutsamp
```

```
##   id week weight response.type   height response    fixed     random
## 1  1    1   24.0             1 81.34412     24.0 25.12139  0.1811245
## 2  1    2   32.0             1 86.37267     32.0 31.30910 -0.1659776
## 3  1    3   39.0             1 93.42815     39.0 37.49681 -0.5130798
## 4  1    4   42.5             1 92.89611     42.5 43.68451 -0.8601820
## 5  1    5   48.0             1 97.17808     48.0 49.87222 -1.2072841
## 6  1    6   54.5             1 98.97584     54.5 56.05993 -1.5543863
##     fitted
## 1 25.30251
## 2 31.14312
## 3 36.98373
## 4 42.82433
## 5 48.66494
## 6 54.50554
```

```
head(testpred.mixmod4) #using predict
```

```
##        1        1        1        1        1        1
## 25.30251 31.14312 36.98373 42.82433 48.66494 54.50554
```

```
head(mixmod4$fitted) #directly from the model
```

```
##      fixed       id
## 1 25.12139 25.30251
## 2 31.30910 31.14312
## 3 37.49681 36.98373
## 4 43.68451 42.82433
## 5 49.87222 48.66494
## 6 56.05993 54.50554
```

```
#Compare random effects (random intercepts and slopes)
head(pred.mixmod4$random)#using mixoutsamp
```

```
##   id     reff1         reff2      reff3      reff4
## 1  1 -1.808318  0.0005116774  2.3365446 -0.3476138
## 2  2  2.226680 -0.0007734317 -2.9717537  0.4472604
## 3  3 -1.890472 -0.0037814658 -0.8092307  0.2705478
## 4  4 -2.147001  0.0019494024  3.7451846 -0.6042836
## 5  5 -4.181523  0.0013388895  5.5036538 -0.8243778
## 6  6 -4.853057 -0.0015969793  4.0183998 -0.4946581
```

```
head(as.data.frame(mixmod4$coefficients$random$id)) #directly from the model
```

```
##   (Intercept)          week response.type week:response.type
## 1   -1.808318  0.0005116774     2.3365446         -0.3476138
## 2    2.226680 -0.0007734317    -2.9717537          0.4472604
## 3   -1.890472 -0.0037814658    -0.8092307          0.2705478
## 4   -2.147001  0.0019494024     3.7451846         -0.6042836
## 5   -4.181523  0.0013388895     5.5036538         -0.8243778
```

```
## 6   -4.853057 -0.0015969793     4.0183998         -0.4946581
```

```
#---
#also obtaining out-of-sample predictions for pigs.outsample
#---

pred.outsample.mixmod4=mixoutsamp(model=mixmod4,newdata=pigs.multivar.outsample)
head(pred.outsample.mixmod4$preddata)
```

```
##     id week weight response.type   height response    fixed    random
## 361 41    1   26.5             1 85.62422     26.5 25.12139 -1.186687
## 362 41    2   30.5             1 87.45276     30.5 31.30910 -2.089630
## 363 41    3   33.0             1 87.58644     33.0 37.49681 -2.992573
## 364 41    4   39.0             1 93.14827     39.0 43.68451 -3.895517
## 365 41    5   43.5             1 99.26311     43.5 49.87222 -4.798460
## 366 41    6   49.5             1 98.42601     49.5 56.05993 -5.701403
##       fitted
## 361 23.93470
## 362 29.21947
## 363 34.50423
## 364 39.78900
## 365 45.07376
## 366 50.35853
```

```
head(pred.outsample.mixmod4$random)
```

```
##   id     reff1         reff2      reff3       reff4
## 1 41 -7.089550 -0.0011691744  6.8058067 -0.90177415
## 2 42  0.200584  0.0010339937  0.6182732 -0.12935815
## 3 43 -1.425570  0.0004877494  1.9392392 -0.29110488
## 4 44  4.944497  0.0048857949 -1.6119902  0.02165414
## 5 45 10.710321  0.0099677472 -3.9314416  0.13388713
## 6 46  4.821134  0.0027527114 -3.0598016  0.31270803
```