Modellentwicklung mit R und git

Clemens Schmid

26. Juni 2016

git – Ausgangssituation

R – Datenvorbereitung

git – commit

R – Modellentwicklung

R – Kalibration

git – Geschichte

git – Ausgangssituation

status

Wir befinden uns in einem Subverzeichnis *nnmodel* im von git überwachten Verzeichnis *somethings*.

```
[clemens@clemens_asus nnmodel]$ pwd
/home/clemens/Rstats/somethings/nnmodel
```

git status ermöglicht es uns, den aktuellen Zustand von *somethings* aus der Perspektive von git abzufragen.

```
[clemens@clemens_asus nnmodel]$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

Wir befinden uns im Master-Zweig, der auf dem selben Stand wie das externe Repository `git@github.com:nevrome/somethings.git` ist. Bisher wurden keine Änderungen am letzten aktuellen Zustand vorgenommen.

R – Datenvorbereitung

Bibliotheken

Zunächst legen wir eine Datei data_prep.R an. Sie enthält folgendes R-Skript:

```
library(nnet)
library(dplyr)
library(quantaar)
```

- **nnet**: Feed-Forward Neural Networks and Multinomial Log-Linear Models
  - Sammlung verschiedener machine learning Algorithmen. Wir werden einen Algorithmus zur Konstruktion eines Neuronalen Netzes nutzen.
- **dplyr**: A Grammar of Data Manipulation
  - Umfassende data wrangling Funktionssammlung. Wir brauchen hier nur eine Filter- und eine Sample Funktion.
- **quantaar**: R Library for Quantitative Analysis in Archaeology
  - Sammlung von für bestimmte archäologische Anwendungen vorgesehene Methoden. Wir nutzen nur einen Testdatensatz daraus.

Datenvorbereitung

Der Datensatz *bs1* enthält eine Datentabelle für ein fiktives Gräberfeld. Das Gräberfeld umfasst 50 Bestattungen, für die jeweils Informationen zu Geschlecht, Bestattungssitte und Beigaben vorliegen.

```
graves <- bs1

colnames(graves)
```

```
##  [1] "sex_male"     "sex_female"   "pos_crouched" "pos_extended"
##  [5] "orient_N_S"   "orient_W_E"   "axe_1"        "axe_2"
##  [9] "adze_1"       "adze_2"       "pottery_1"    "pottery_2"
## [13] "pottery_3"    "pottery_4"    "goldring"     "goldbead"
## [17] "fibula_1"     "fibula_2"
```

Datenvorbereitung

Wir wollen uns nur auf die Relation von Geschlecht und Beigaben konzentrieren und entfernen alle anderen Attribute aus der Tabelle.

```
graves <- data.frame(graves[,-c(3,4,5,6)])

colnames(graves)
```

```
##  [1] "sex_male"   "sex_female" "axe_1"      "axe_2"      "adze_1"
##  [6] "adze_2"     "pottery_1"  "pottery_2"  "pottery_3"  "pottery_4"
## [11] "goldring"   "goldbead"   "fibula_1"   "fibula_2"
```

Datenvorbereitung

Für die weitere Auswertung, muss die Datentabelle geringfügig umgeformt werden. Statt den Spalten *sex_male* und *sex_female*, die die fiktive anthropologische Ansprache binär kodiert enthalten **(int)**, benötigen wir eine Spalte, die die Informationen bündelt **(factor)**.

```
sex <- as.factor(rep(0, nrow(graves)))
levels(sex) <- c("male", "female", "unknown")
for (i in 1:nrow(graves)){
  if (graves$sex_male[i] == 1){
    sex[i] <- "male"
  }
  if(graves$sex_female[i] == 1){
    sex[i] <- "female"
  }
  if((graves$sex_male[i] == 1 && graves$sex_female[i] == 1) ||
     (graves$sex_male[i] == 0 && graves$sex_female[i] == 0)){
    sex[i] <- "unknown"
  }
}
```

Datenvorbereitung

```
sex
```

```
##  [1] male    male    male    male    male    male    male    male
##  [9] male    male    male    male    male    male    male    male
## [17] male    male    male    male    unknown unknown unknown unknown
## [25] unknown unknown unknown unknown unknown unknown female  female
## [33] female  female  female  female  female  female  female  female
## [41] female  female  female  female  female  female  female  female
## [49] female  female
## Levels: male female unknown
```

Der erzeugte Vektor soll die ursprünglichen Spalten ersetzen.

```
graves <- data.frame(sex, graves[,-c(1,2)])
```

Datenvorbereitung

Der Trainingsdatensatz für den Algorithmus, der unser Modell konstruieren soll, darf keine unklaren Werte enthalten.

```
graves <- filter(
  graves,
  sex != "unknown"
)
```

Außerdem möchten wir **bootstrapping** zur Anwendung bringen, um unser Modell später validieren und kalibrieren zu können. Der Trainingsdatensatz soll also nur 20 der 40 geschlechtsbestimmten Gräber umfassen.

```
graves.training <- sample_n(graves, 20)
```

Export

Speichern der vorbereiteten Daten ins Dateisystem.

```
save(
  graves,
  file = "/home/clemens/Rstats/somethings/nnmodel/graves.RData"
)

save(
  graves.training,
  file = "/home/clemens/Rstats/somethings/nnmodel/graves-training.RData"
)
```

Blick ins Dateisystem:

```
[clemens@clemens_asus nnmodel]$ ls
data_prep.R  graves.RData  graves-training.RData
```

git – commit

status

git hat das Anlegen des neuen Verzeichnisses und der Dateien darin bemerkt.

```
[clemens@clemens_asus nnmodel]$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    ./

nothing added to commit but untracked files present (use "git add" to track)
```

add

Um den erarbeiteten Zustand zu speichern, müssen wir die Änderungen mit **git add** bestätigen.

```
[clemens@clemens_asus nnmodel]$ git add --all

[clemens@clemens_asus nnmodel]$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   data_prep.R
    new file:   graves-training.RData
    new file:   graves.RData
```

commit

Die Änderung wird nun in Form eines Commits über **git commit** dokumentiert und archiviert.

```
[clemens@clemens_asus nnmodel]$ git commit -m
"Datenvorbereitung für das NN-Modell"
Warning: commit message did not conform to UTF-8.
You may want to amend it after fixing the message, or set the config
variable i18n.commitencoding to the encoding your project uses.
[master 58d1991] Datenvorbereitung fÃ¼r das NN-Modell
 3 files changed, 40 insertions(+)
 create mode 100644 nnmodel/data_prep.R
 create mode 100644 nnmodel/graves-training.RData
 create mode 100644 nnmodel/graves.RData
```

push

Gegebenenfalls können wir die Änderung mit **git push** auch in ein externes Archiv überführen.

```
[clemens@clemens_asus nnmodel]$ git push origin master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.79 KiB | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
To git@github.com:nevrome/somethings.git
   52dd0f2..58d1991  master -> master
```

R – Modellentwicklung

Modellentwicklung

Wir legen ein neues R-Skript model.R an, um die Arbeit am Modell von der Datenvorbereitung zu trennen.

Zur Modellentwicklung kommt ein Algorithmus zur Konstruktion eines sehr simplen **Neuronalen Netzwerks** zum Einsatz.

```
sex.nnet <- nnet(sex~., data = graves.training, size = 1, decay = 0)
```

```
## Warning in nnet.formula(sex ~ ., data = graves.training, size = 1, decay =
## 0): group 'unknown' is empty
```

```
## # weights:  17
## initial  value 14.564486
## iter  10 value 0.080327
## final  value 0.000079
## converged
```

Wichtig: Der Algorithmus ist **nondeterministisch**.

Modell

```
summary(sex.nnet)
```

```
## a 12-1-2 network with 17 weights
## options were - softmax modelling
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##   -0.83   -4.24   -1.58   -6.86    0.03   -0.83    1.20    2.16    0.69
##  i9->h1 i10->h1 i11->h1 i12->h1
##   -0.16    0.79    0.48    0.80
##  b->o1 h1->o1
##  11.91 -17.92
##  b->o2 h1->o2
## -11.47  17.33
```

Validierung

Durch Vergleich der Modellvorhersage mit den Ausgangsdaten können wir eine Abschätzung über die Güte unseres Modells treffen.

```
table(
  actual = graves$sex,
  predict = predict(sex.nnet, newdata = graves, type = "class")
)
```

```
##          predict
## actual    female male
##   male         2   18
##   female      18    2
##   unknown      0    0
```

Export

Speichern des Modells ins Dateisystem.

```
save(sex.nnet, file = "/home/clemens/Rstats/somethings/nnmodel/model.RData")
```

Blick ins Dateisystem:

```
[clemens@clemens_asus nnmodel]$ ls
data_prep.R  graves.RData  graves-training.RData  model.R  model.RData
```

git – commit

Archivieren und Dokumentieren des Arbeitsfortschritts mit git.

```
[clemens@clemens_asus nnmodel]$ git status
[clemens@clemens_asus nnmodel]$ git add --all
[clemens@clemens_asus nnmodel]$ git commit -m "Modellentwicklung"
[clemens@clemens_asus nnmodel]$ git push origin master
```

R – Kalibration

Modellentwicklung

Neuberechnung des Modells mit veränderten Parametern.

```
sex.nnet <- nnet(sex~., data = graves.training, size = 10, decay = 0.1)
```

```
## Warning in nnet.formula(sex ~ ., data = graves.training, size = 10, decay =
## 0.1): group 'unknown' is empty
```

```
## # weights:  152
## initial  value 18.653615
## iter  10 value 2.120026
## iter  20 value 1.485212
## iter  30 value 1.467246
## iter  40 value 1.465540
## iter  50 value 1.465483
## final  value 1.465483
## converged
```

Da die potentielle Komplexität des Neuronalen Netzes gegenüber dem ersten Versuch oben vergrößert wurde, ist das Ergebnismodell wahrscheinlich etwas besser. Aber: Nondeterminismus!

Modell

```
summary(sex.nnet)
```

```
## a 12-10-2 network with 152 weights
## options were - softmax modelling  decay=0.1
##    b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##     0.03    0.22    0.34    0.37   -0.17    0.01   -0.08   -0.34   -0.09
##   i9->h1 i10->h1 i11->h1 i12->h1
##    -0.11   -0.02   -0.17   -0.05
##    b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##     0.03    0.22    0.34    0.37   -0.17    0.01   -0.08   -0.34   -0.09
##   i9->h2 i10->h2 i11->h2 i12->h2
##    -0.11   -0.02   -0.17   -0.05
##    b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##     0.03    0.22    0.34    0.37   -0.17    0.01   -0.08   -0.34   -0.09
##   i9->h3 i10->h3 i11->h3 i12->h3
##    -0.11   -0.02   -0.17   -0.05
##    b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##    -0.03   -0.19   -0.29   -0.32    0.14   -0.01    0.07    0.29    0.07
##   i9->h4 i10->h4 i11->h4 i12->h4
##     0.10    0.02    0.14    0.04
##    b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    -0.03   -0.19   -0.29   -0.32    0.14   -0.01    0.07    0.29    0.07
```

Validierung

```
table(
  actual = graves$sex,
  predict = predict(sex.nnet, newdata = graves, type = "class")
)
```

```
##           predict
## actual     female male
##   male          1   19
##   female       20    0
##   unknown       0    0
```

Export

Gegebenenfalls Speichern des Modells ins Dateisystem

```
save(sex.nnet, file = "/home/clemens/Rstats/somethings/nnmodel/model.RData")
```

git – commit

Archivieren und Dokumentieren des Arbeitsfortschritts mit git.

```
[clemens@clemens_asus nnmodel]$ git status
[clemens@clemens_asus nnmodel]$ git add --all
[clemens@clemens_asus nnmodel]$ git commit -m "Modellkalibration"
[clemens@clemens_asus nnmodel]$ git push origin master
```

git – Geschichte

log

git hat alle Änderungen archiviert. Wir können uns die Versionsgeschichte mit **git log** ansehen und mit **git revert** zu jedem beliebigen Arbeitsstand zurückspringen.

```
[clemens@clemens_asus nnmodel]$ git log
commit 20668fa758698d610d7a8c4f2a19f5286dc74e0b
Author: nevrome <clemens@nevrome.de>
Date:   Mon Jun 27 08:48:53 2016 +0200

    Modellkalibration

commit 6ad3a276e6f6d770594ee59b2a49be9bff57641d
Author: nevrome <clemens@nevrome.de>
Date:   Mon Jun 27 08:40:17 2016 +0200

    Modellentwicklung

commit 58d1991dd3e2fb2a59030e48767a7784c7ba9e6e
Author: nevrome <clemens@nevrome.de>
Date:   Mon Jun 27 08:09:58 2016 +0200

    Datenvorbereitung fÃ¼r das NN-Modell
```