Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Отчёт о работе с пакетом «Rminer»

Выполнила:
Морозова Дарья 317

2012

Описание пакета:

Пакет “Rminer” способствует более лёгкой работе с data mining алгоритмами в задачах классификации и регрессии. Особенно он подходит для NN (нейронных сетей) и SVM (Support Vector Machines). Здесь предлагается пользователю небольшой набор любопытных функций.

Список функций пакета “Rminer”

CasesSeries
crossvaldata
delevels
factorize
fit
holdout
Importance
imputation
lforecast
mgraph
mining
mmetric
predict.fit
savemining
sin1reg
vecplot

По сравнению с Rattle, Rminer может показаться упрощённым вариантом, так как его легче установить, и он требует меньше ранее загруженных пакетов R. Однако Rminer предоставляет больше NN и SVM возможностей (SVM можно использовать для задач регрессии).

Rminer выполняет отбор признаков (настраивает NN и SVM), вычисляет метрики и показывает результаты на графиках.

Этот пакет выполняет такие задания, как civil engineering, wine quality и spam email detection. Пользуется спросом у менеджеров, биологов и строителей.

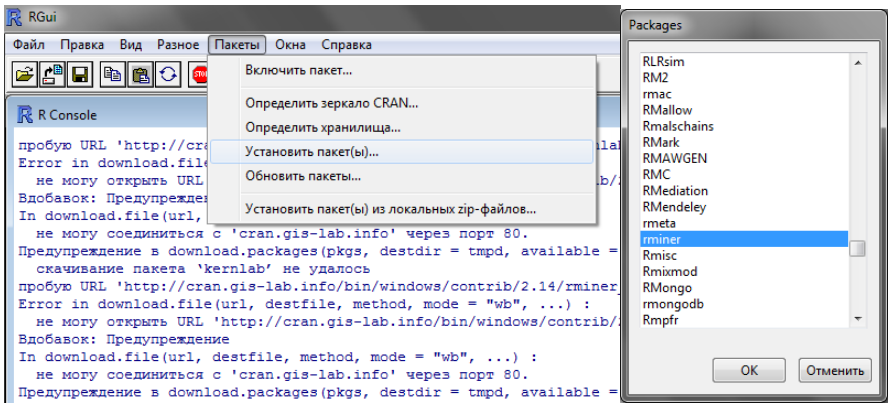
Инструкции по загрузке пакета Rminer

I. Если ещё не установлен R, загрузить из `http://www.r-project.org`.

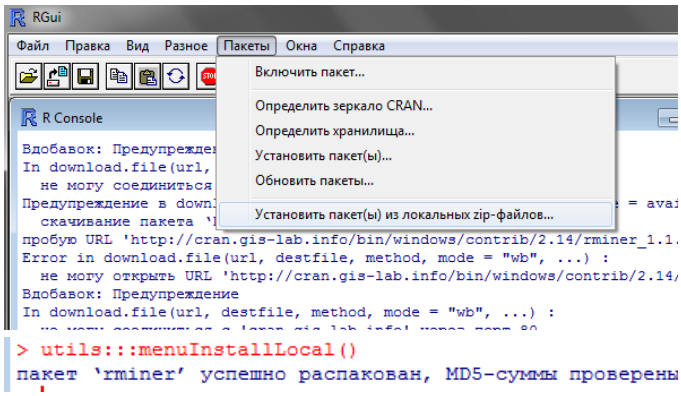
II. Установить некоторые пакеты R, которые потребуются для Rminer. Ему необходимы nnet, kknn, kernlab, rpart, lattice, methods. Большинство из этих пакетов уже загружены в R, поэтому поработать придётся только с пакетами kernlab и kknn.

III. Загрузить Rminer из http://cran.r-project.org/web/packages/rminer/index.html

IV. Набрать R команду `install.packages ("rminer")` и затем выбрать зеркало CRAN – “Russia”.
В результате R в папку “ downloaded_packages” загрузит пакеты igraph, kernlab, kknn и rminer (для справки набрать ?`install.packages`).
Или можно проделать это так (Windows):



V. Установить пакет из локальных zip-файлов (Windows):
(вспомнить, куда загрузили `rminer_1.0.zip`)



Набрать R команды:

```
install.packages (repos = NULL, "rminer_1.0.tar.gz")
library(rminer)
?fit
help(package = rminer)
getwd() - показывает текущую директорию
setwd(dir) - изменяет текущую директорию на dir
```

Пример классификации satellite-example:

Данные для этой задачи были сгенерированы с помощью многоспектральных изображений спутника Landsat. Цель исследования заключалась в том, чтобы классифицировать пиксели по 4 спектральным группам, основываясь на 36 количественных признаках.

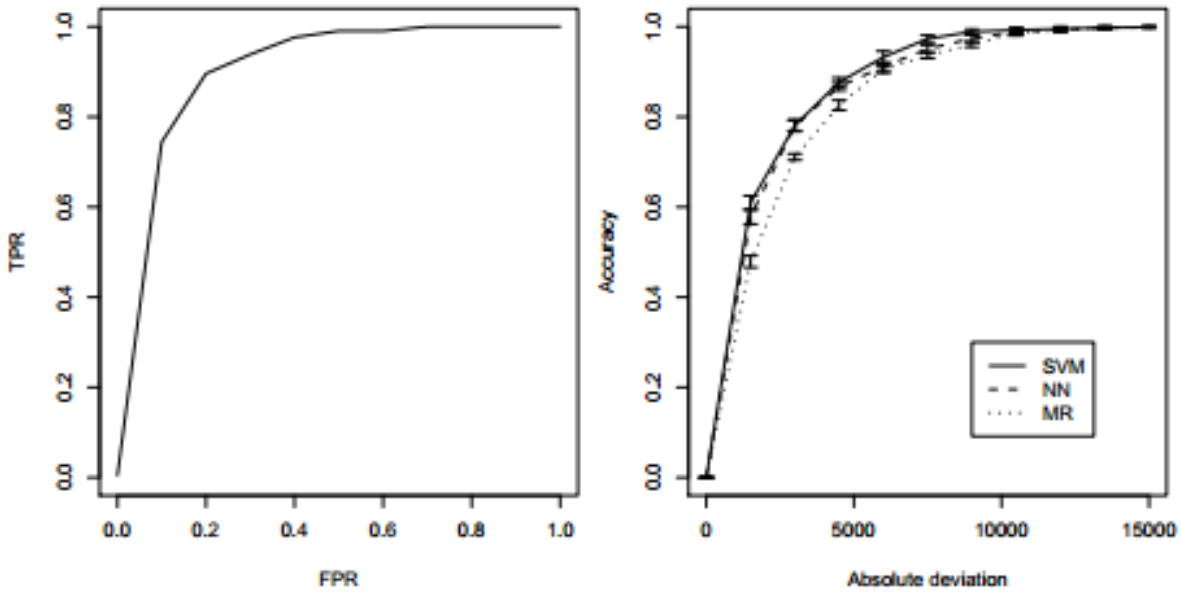
Даны 2 выбоки: обучающая – sat.trn с 4435 объектами, и тестовая – sat.tst с 2000 объектами.

```
library(rminer)  # load the library
# --- modeling block: ---
# read the training and test sets:
mypause("Modeling block (requires some computation):")
# rminer function that pauses the console until enter is pressed
tr=read.table("sat.trn",sep=" "); ts=read.table("sat.tst",sep=" ")
tr$V37=factor(tr$V37); ts$V37=factor(ts$V37) #convert V37 to factor
DT=fit(V37~.,tr,model="dt") # fit a Decision Tree with tr
NN=fit(V37~.,tr,model="mlp",search=10) # fit a NN with H=10
SV=fit(V37~.,tr,model="svm",search=2^c(-5,-3)) # fit the SVM
print(DT); print(NN); print(SV) # show and save the trained DM models
savemodel(DT,"sat.dt"); savemodel(NN,"sat.nn"); savemodel(SV,"sat.sv")
savemodel(SV,"sat.sv")

# get the predictions:
PDT=predict(DT,ts); PNN=predict(NN,ts); PSVM=predict(SV,ts)
P=data.frame(ts=ts$V37,dt=PDT,nn=PNN,svm=PSVM) # create a data.frame
write.table(P,"sat.res",row.names=FALSE) # save output and predictions

# --- evaluation block: ---
mypause("Evaluation block (computes some metrics and ROC graph into
roc4.pdf):")
# rminer function that pauses the console until enter is pressed
P=read.table("sat.res",header=TRUE); P$ts=factor(P$ts); # read the results
# compute the test errors:
EDT=metrics(P$ts,P[,2:7]); ESV=metrics(P$ts,P[,14:19]);
ENN1=metrics(P$ts,P[,8:13]); ENN2=metrics(P$ts,P[,8:13],D=0.7,TC=4)
print(EDT); print(ESV); print(ENN1); print(ENN2) # show errors
mgraph(P$ts,P[,8:13],graph="ROC",PDF="roc4",TC=4) # plot ROC
NN=loadmodel("sat.nn") # load best model
```

Функции read.table и write.table загружают и сохраняют данные из текстового файла (“.csv”). Объекты tr и ts – data.frames. Выходной параметр V37 – набор классов, ему соответствуют числовые значения.
В примере тестируются дерево решений (model="dt"), NN ("mlp") и SVM ("svm"). Предсказание для каждой модели – матрица, где каждый столбец обозначается за p_c для данного $c \in$ {“1”, ”2”, “3”, ”4”, “5”, ”7”} (нет 6ого класса). Функция metrics возвращает список с оценками качества работы. Лучший результат был получен NN с ACC = 86%, затем был SVM со своими 81%, а далее DT с 78%. Что касается глобальной AUC, то тут тоже победил NN, получивший 98%.



Примеры ROC (слева) и REC (справа) кривых

Некоторые результаты, которые выдаёт R при получении такого кода:

```
Slot "time" elapsed: 19.56
Slot "object": n= 4435
1) root 4435 3363 1 (0.24 0.11 0.22 0.094 0.11 0.23)
  2) V17>=79.5 1107  207 3 (0.018 0 0.81 0.13 0.0009 0.042) *
  3) V17< 79.5 3328 2276 1 (0.32 0.14 0.018 0.083 0.14 0.3)
    6) V20>=73.5 1924  920 1 (0.52 0.25 0.029 0.062 0.094 0.047)
     12) V18>=65 1403  401 1 (0.71 0.026 0.039 0.086 0.07 0.065)
       24) V21< 73 1123  165 1 (0.85 0.027 0.008 0.023 0.073 0.016) *
       25) V21>=73 280  186 4 (0.16 0.025 0.16 0.34 0.057 0.26)
         50) V20>=89 56   17 1 (0.7 0 0.089 0 0.21 0) *
         51) V20< 89 224  130 4 (0.022 0.031 0.18 0.42 0.018 0.33) *
     13) V18< 65 521   85 2 (0.0038 0.84 0 0 0.16 0)
       26) V20>=90.5 428    8 2 (0 0.98 0 0 0.019 0) *
```

```
      27) V20< 90.5 93   18 5 (0.022 0.17 0 0 0.81 0) *
    7) V20< 73.5 1404  504 7 (0.034 0.0043 0.0043 0.11 0.21 0.64)
     14) V18< 61.5 227    8 5 (0.0088 0.0044 0 0 0.96 0.022) *
     15) V18>=61.5 1177  282 7 (0.039 0.0042 0.0051 0.13 0.059 0.76) *

SV type: C-svc  (classification)
 parameter : cost C = 3

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.03125

Number of Support Vectors : 1303

Objective Function Value : -32.3417 -139.8064 -70.0573 -225.7086 -33.0775 -
22.2247 -62.6081 -82.2142 -44.4222 -801.7633 -56.2032 -297.1046 -185.1081 -
1210.698 -422.2368
Training error : 0.073281
Probability model included.

$conf
       pred
target  FALSE TRUE
  FALSE  1742   47
  TRUE    123   88
$acc [1] 91.5
$kappa [1] 46.45922
$acclass [1] 91.5 91.5
$tpr [1] 97.37283 41.70616
$tnr [1] 41.70616 97.37283
$precision [1] 93.40483 65.18519
$tauc [1] 0.9249495
$auc [1] 0.9249495 0.9249495
```

Список используемой литературы:

- “Анализ данных, обучение по прецедентам, логические игры, системы WEKA, RAPIDMINER и MATLAB” Дьяконов А.Г.
- страничка “Обсуждение”
- http://www.machinelearning.ru/
- http://www3.dsi.uminho.pt/pcortez/readme.txt инструкции по загрузке
- http://www3.dsi.uminho.pt/pcortez/rminer.html
- http://repositorium.sdum.uminho.pt/bitstream/1822/10828/1/2010-rminer.pdf статья по Rminer