# Data Mining With RWeka

An R package interfaces R to the Open-source machine learning toolbox Weka

刘思喆

@华东师范大学

2011 年 11 月 12 日

**Part1 : Weka and R**

**Part2 : ROCR**

# Weka 是什么

Weka (Waikato Environment for Knowledge Analysis) 是基于Java环境的数据挖掘软件，它的强项主要在分类(classification)领域，在这个领域几乎包含了几乎所有的机器学习的方法，而且它还集成了如回归、关联规则、clustering 等算法。

2005年8月，在第11届 ACM SIGKDD 国际会议上，怀卡托大学的 Weka 小组荣获了数据挖掘和知识探索领域的最高服务奖，Weka 系统得到了广泛的认可，被誉为数据挖掘和机器学习历史上的里程碑，是现今最完备的数据挖掘工具之一（已有11年的发展历史）。Weka 的每月下载次数已超过万次。

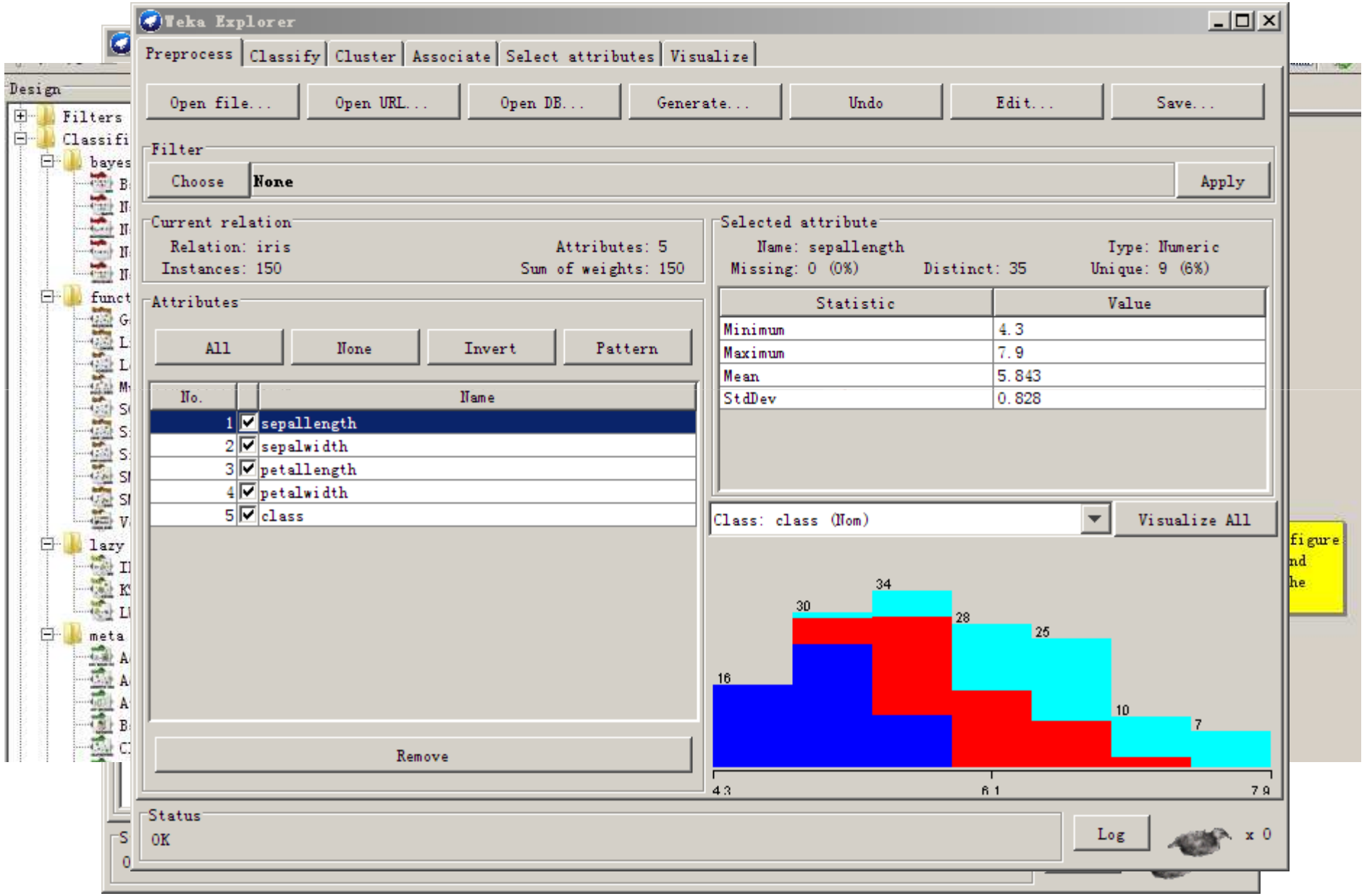
# Weka 的功能


- **Explorer：**在这个环境中，Weka 提供了数据的预处理，数据格式的转化，各种数据挖掘算法（包括分类与回归算法，聚类算法，关联规则等），并提供了结果的可视化工具。
- **Experimenter：**运行算法试验、管理算法方案之间的统计检验的环境。Experiment环境可以让用户创建，运行，修改和分析算法试验，这也许比单独的分析各个算法更加方便
- **KnowledgeFlow：**提供了“数据流”形式的界面。用户可以选择相应流程组件，把它们放置在面板上并按一定的顺序连接起来，组成一个“知识流”（knowledge flow）来处理和分析数据
- **SimpleCLI：**提供了一个简单的命令行界面

# Weka 的特点

- WEKA存储数据的格式是ARFF（Attribute-Relation File Format）文件，这是一种ASCII文本文件；
- 可通过 JDBC 协议读取数据库数据；
- 丰富的机器学习算法的集合；
- 提供了完整的数据挖掘流程，如数据源读取、数据预处理、回归、分类、聚类、关联规则、变量选择、交叉验证、模型评估、交互式可视化等；
- GPL协议，Java 实现
- 实现了很多标准算法，比如 Ross Quinlan 的 C4.5 和 M5

# Weka 的界面


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150
Selected attribute
Name: sepallength
Type: Numeric
Missing: 0 (0%)
Distinct: 35
Unique: 9 (6%)
Statistic
Value
Minimum
4.3
Maximum
7.9
Mean
5.843
StdDev
0.828
Attributes
All
None
Invert
Pattern
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Class: class (Nom)
Visualize All
Remove
Status
OK
Log
x 0

# Weka 同 R 语言的关系

- Weka 和 R 都是数据挖掘领域流行的平台环境，虽然二者实现的方式不同，Weka 是 Java 实现，R 语言的底层以R、C、Fortran为主；
- 两者都有很好的跨平台性
- 都是基于 GPL 协议的开源软件
- 他们都来自于新西兰
- R 强于统计分析，而 Weka 强于机器学习
- R 通过 RWeka 包搭建同 Weka 的连接，RWeka 实际使用了 rJava 做对 Weka 进行底层交互

# R 能够直接调用的 Weka 算法

`list_Weka_interfaces()`

| 分类 | 细分类 | 函数 | 释义 |
|---|---|---|---|
| filters | | Normalize() | 无监督的标准化连续性数据 |
| | | Discretize() | 有监督的离散化连续性数值数据 |
| clusters | | Cobweb() | Cobweb算法 |
| | | FarthestFirst() | 快速的近似的k均值聚类算法 |
| | | SimpleKMeans() | k均值 |
| | | XMeans() | 改进的k均值法，能自动决定类别数 |
| | | DBScan() | 基于密度的聚类方法 |
| associators | | Apriori() | Apriori算法 |
| | | Tertius() | Tertius算法 |

# R 能够直接调用的 Weka 算法

| | | | |
|---|---|---|---|
| classifiers | Lazy | IBk() | k最近邻 |
| | | LBR() | naive Bayes |
| | Trees | J48() | C4.5决策树算法，Quinlan (1993) |
| | | LMT() | 组合树结构的Logistic回归模型，Landwehr (2005) |
| | | M5P() | 组合了树结构的线性回归的模型，Quinlan (1992) |
| | | DecisionStump() | 单层决策树算法 |
| | Meta | AdaBoostM1() | Adaboost M1方法，Freund and Schapire (1996) |
| | | Bagging() | Bagging算法，Breiman (1996) |
| | | LogitBoost() | 可加logistic回归，Friedman, Hastie and Tibshirani (2000) |
| | | Stacking() | Stacking，Wolpert (1992) |
| | | MultiBoostAB() | AdaBoost 方法的改进，Webb (2000) |
| | Functions | LinearRegression() | 线性回归 |
| | | SMO() | 支持向量机 |
| | | Logistic() | logistic回归 |
| | Rules | JRip() | RIPPER方法，Cohen (1995) |
| | | M5Rules() | 用M5方法产生回归问题的决策规则，Hall, Holmes and Frank (1999) |
| | | OneR() | 简单的1-R分类法，Holte (1993) |
| | | PART() | 产生PART决策规则，Frank and Witten (1998) |

# 算法简介——J48

```
library(RWeka)
data(spam, package = 'ElemStatLearn')
s <- sample(1:nrow(spam), round(nrow(spam)/3,0))
M.J48 <- J48(spam ~ . , data = spam[-s,],
                        control = Weka_control(M = 100))
```

结果的展示：

```
J48 pruned tree
------------------

A.7 <= 0
|   A.53 <= 0.053
|   |   A.23 <= 0.25
|   |   |   A.16 <= 0.2
|   |   |   |   A.52 <= 0.106: email (1477.0/81.0)
|   |   |   |   A.52 > 0.106
|   |   |   |   |   A.55 <= 2.655: email (269.0/43.0)
|   |   |   |   |   A.55 > 2.655: spam (102.0/40.0)
|   |   |   A.16 > 0.2
|   |   |   |   A.52 <= 0.198: email (117.0/28.0)
|   |   |   |   A.52 > 0.198: spam (103.0/18.0)
|   |   A.23 > 0.25: spam (41.0/3.0)
|   A.53 > 0.053: spam (432.0/74.0)
A.7 > 0: spam (526.0/30.0)

Number of Leaves  :     8

Size of the tree :      15
```

# 查看算法参数

Weka Option Wizard

| | WOW(J48) |
|---|---|
| -U | Use unpruned tree. |
| -O | Do not collapse tree. |
| -C | Set confidence threshold for pruning. (default 0.25) |
| -M | Set minimum number of instances per leaf. (default 2) |
| -R | Use reduced error pruning. |
| -N | Set number of folds for reduced error pruning. One fold is used as pruning set. (default 3) |
| -B | Use binary splits only. |
| -S | Don't perform subtree raising. |
| -L | Do not clean up after the tree has been built. |
| -A | Laplace smoothing for predicted probabilities. |
| -J | Do not use MDL correction for info gain on numeric attributes. |
| -Q | Seed for random data shuffling (default 1). |

# 模型预测

predict 是一个泛型函数，在 RWeka 下，它代表了

- predict.Weka_classifier
- predict.Weka_clusterer

predict 函数(classifier)支持 type 参数，其两个选项为：

- Class：输出预测类别
- Probability：输出预测概率

```
p.J48 <- predict(M.J48, spam[s,], type = ‘class’)
table(spam$spam, p.J48)
```

# 模型评估

```
e <- evaluate_Weka_classifier(M.J48, newdata = spam[s,],
                      cost = matrix(c(0,2,1,0), ncol = 2),
                       numFolds = 10, complexity = TRUE,
                       seed = 123, class = TRUE)
e$details
```

```
> e$details
             pctCorrect            pctIncorrect          pctUnclassified                   kappa
             86.8318123              13.1681877               0.0000000               0.7233885
      meanAbsoluteError    rootMeanSquaredError relativeAbsoluteError rootRelativeSquaredError
              0.2086155               0.3256297              44.0108071              66.8901385
```

| 评估子项 | 解释 |
|---|---|
| details | 基本统计信息，如正确率等 |
| string | 性能统计信息，以字符串形式 |
| detailsCost | Cost统计信息 |
| detailsComplexity | 基于entropy的统计 |
| detailsClass | 各类的统计信息，如recall、precision、AUC等 |
| confusionMatrix | 混淆矩阵 |

# 算法简介——PART

```
library(RWeka)
data(spam, package = 'ElemStatLearn')
M.PART <- PART(spam ~ . , data = spam,
                          control = Weka_control(M = 100))
```

结果的展示：

```
> M.PART
PART decision list
------------------

A.7 <= 0 AND
A.53 <= 0.055 AND
A.23 <= 0.25 AND
A.52 <= 0.378: email (2697.0/246.0)

A.25 <= 0.4 AND
A.56 > 9: spam (1557.0/103.0)

A.25 <= 0.21 AND
A.21 <= 0.33: email (139.0/36.0)

A.25 <= 0.21: spam (105.0/38.0)

: email (103.0/10.0)

Number of Rules  :      5
```

# 算法简介——M5P

M5P算法同J48算法结构上类似，都是构建了树，只不过M5P算法在树的叶子上使用了线性回归模型。

```
library(RWeka)
DF3 <- read.arff(system.file("arff", "cpu.arff",
                                    package = "RWeka"))
m3 <- M5P(class ~ ., data = DF3)
```

```
M5 pruned model tree:
(using smoothed linear models)

CHMIN <= 7.5 : LM1 (165/12.903%)
CHMIN >  7.5 :
|   MMAX <= 28000 :
|   |   MMAX <= 13240 :
|   |   |   CACH <= 81.5 : LM2 (6/18.551%)
|   |   |   CACH >  81.5 : LM3 (4/30.824%)
|   |   MMAX >  13240 : LM4 (11/24.185%)
|   MMAX >  28000 : LM5 (23/48.302%)

LM num: 1
class =
        -0.0055 * MYCT
        + 0.0013 * MMIN
        + 0.0029 * MMAX
        + 0.8007 * CACH
        + 0.4015 * CHMAX
        + 11.0971

LM num: 2
class =
        -1.0307 * MYCT
        + 0.0086 * MMIN
        + 0.0031 * MMAX
        + 0.7866 * CACH
        - 2.4503 * CHMIN
        + 1.1597 * CHMAX
        + 70.8672

LM num: 3
class =
        -1.1057 * MYCT
        + 0.0086 * MMIN
        + 0.0031 * MMAX
        + 0.7995 * CACH
        - 2.4503 * CHMIN
        + 1.1597 * CHMAX
        + 83.0016

LM num: 4
class =
        -0.8813 * MYCT
        + 0.0086 * MMIN
        + 0.0031 * MMAX
        + 0.6547 * CACH
        - 2.3561 * CHMIN
        + 1.1597 * CHMAX
        + 82.5725

LM num: 5
class =
        -0.4882 * MYCT
        + 0.0218 * MMIN
        + 0.003 * MMAX
        + 0.3865 * CACH
        - 1.3252 * CHMIN
        + 3.3671 * CHMAX
        - 51.8474

Number of Rules : 5
```

# 如何扩展 RWeka 中的算法

R 中构建 Weka 算法的连接函数：

- make_Weka_associator
- make_Weka_classifier
- make_Weka_clusterer

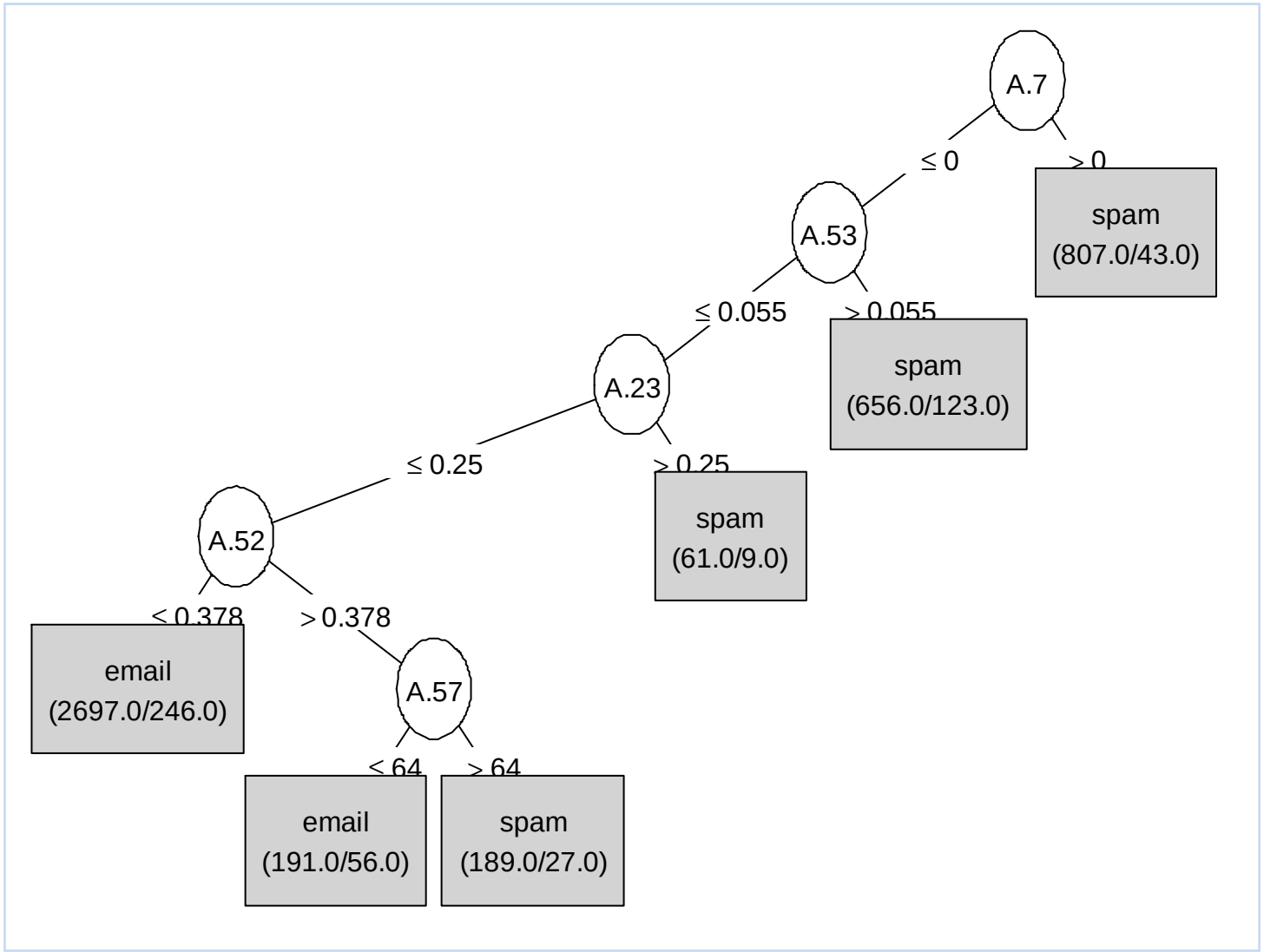
```
## Random Forest
RF <-
make_Weka_classifier("weka/classifiers/trees/RandomForest")

## Multilayer Perceptron
MLP <-
make_Weka_classifier("weka/classifiers/functions/Multilaye
rPerceptron")

FC <-
make_Weka_clusterer("weka/clusterers/FilteredClusterer")
```

# RWeka 的绘图（1）

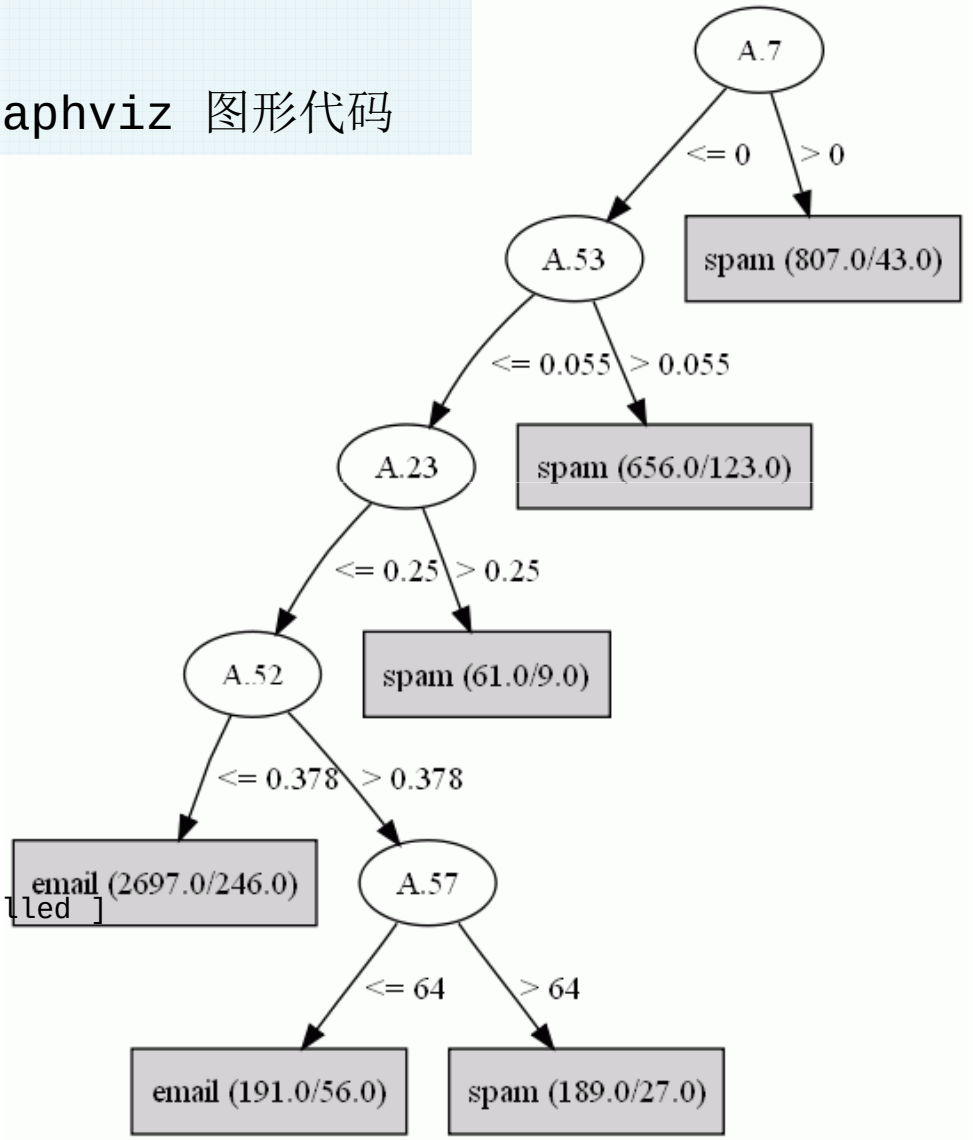
```
plot(M.J48) # 调用party包，来实现绘图
```



# RWeka 的绘图（2）

```
## 借助 Graphviz 环境转化图形
write_to_dot(M.J48) # 生成 Graphviz 图形代码
```

```
digraph J48Tree {
N0 [label="A.7" ]
N0->N1 [label="<= 0"]
N1 [label="A.53" ]
N1->N2 [label="<= 0.055"]
N2 [label="A.23" ]
N2->N3 [label="<= 0.25"]
N3 [label="A.52" ]
N3->N4 [label="<= 0.378"]
N4 [label="email (2697.0/246.0)" shape=box
style=filled ]
N3->N5 [label="> 0.378"]
N5 [label="A.57" ]
N5->N6 [label="<= 64"]
N6 [label="email (191.0/56.0)" shape=box
style=filled ]
N5->N7 [label="> 64"]
N7 [label="spam (189.0/27.0)" shape=box
style=filled ]
N2->N8 [label="> 0.25"]
N8 [label="spam (61.0/9.0)" shape=box style=filled ]
N1->N9 [label="> 0.055"]
N9 [label="spam (656.0/123.0)" shape=box
style=filled ]
N0->N10 [label="> 0"]
N10 [label="spam (807.0/43.0)" shape=box
style=filled ]
}
```



# 思考

- 既然 R 中已经实现了诸如 logistic 回归、CART、RandomForest、SVMs等算法，连接 Weka 获取 Weka 内置算法是否必要？
- 甚至在 R 中的很多算法要比 Weka 中的优秀；
- 假如我们需要在系统级嵌入数据挖掘环境，内置 R 还是内置 Weka？
- 交互式可视化可由 GGobi 等工具来代替
- 使用 RWeka 环境并不能直接输出跨平台的数据挖掘结果，比如pmml。由于在 R 环境中调用了 Java，其挖掘的规则（rules）提取以及系统级嵌入不太容易实现
- Weka 中 Classification 的评估体系（Experimenter）虽然完整，但在 RWeka 中仍有不足

**Part1 : Weka and R**

**Part2 : ROCR**

# ROCR 是做什么的

- 模式识别中分类（classification）算法的地位非常重要
- 而在模型评估时，我们会绘制或计算：
  - ① ROC 曲线
  - ② AUC 值
  - ③ lift 曲线
  - ④ precision/recall 曲线


在 R 中可视化分类器的性能只需要3个命令：
prediction -> performance -> plot

# ROCR 的原理及特性

原理：当提升其中一个指标 X 的性能时，另外一个指标 Y 的性能会随着变差，那么构建二维的 x_y plot 则是首选；

- 除了ROC曲线, precision/recall曲线，lift曲线，cost曲线，或者根据用户需要通过自定义 x 轴和 y 轴的性能指标；
- 可以将 threshold 值绘制在曲线上，或者用渐变颜色表示；
- 或者使用交叉验证、bootstrapping这些方法做曲线平均(水平平均、垂直平均或根据threshold), 以及standard error bars, box plots

# 混淆矩阵（confusion matrix）

我们假定：

二元分类器中， Y 为实际变量，Yhat 为预测变量，用 + 和 – 来代替正例和负例(positive and negative class)；

或者，用另外一种表现方式：

预测类别为 positive(p)和 negative(n)，那么根据实际类别的 p 和 n，则可以将两列数据分为：

- TP (true positives)：预测为 p 且实际为 p
- TN (true negatives)：预测为 n 且实际为 n
- FP (false positives)：预测为 p 且实际为 n
- FN (false negatives). 预测为 n 且实际为 p

```
> table(actual = spam[s,]$spam, pred = predict(M.J48, spam[s,]))
        pred
actual   email spam
  email    880   39
  spam     141  474
```

| | |
|---|---|
| True Positive(TP) | False Negative(FN) |
| False Positive(FP) | True Negative(TN) |

http://en.wikipedia.org/wiki/Receiver_operating_characteristic

help(performance )

# performance measures

| 指标/绘图 | 缩写 | 解释 |
|---|---|---|
| Accuracy | acc | *(TP+TN)/(P+N)*. |
| Error rate | err | *(FP+FN)/(P+N)* |
| False positive rate | fpr | *FP/N*. |
| True positive rate | tpr | *TP/P*. |
| Recall(Sensitivity) | rec(sens) | True positive rate |
| False negative rate | fnr | *FN/P*. |
| Miss | miss | False negative rate |
| True negative rate | tnr | *P(Yhat = - \| Y = -)*. |
| Specificity | spec | True negative rate |
| Positive predictive value | ppv | *TP/(TP+FP)* |
| Precision | prec | Positive predictive value |
| Negative predictive value | npv | *TN/(TN+FN)*. |
| Lift value | lift | *P(Yhat = + \| Y = +)/P(Yhat = +)*. |
| …… | | …… |
| ROC curves | | measure="tpr", x.measure="fpr". |
| Precision/recall graphs | | measure="prec", x.measure="rec". |
| Sensitivity/specificity plots | | measure="sens", x.measure="spec". |
| Lift charts | | measure="lift", x.measure="rpp".(rpp = *P(Yhat = +)*) |
| auc | auc | Area under the ROC curve |

# ROC（receiver operating characteristic）曲线

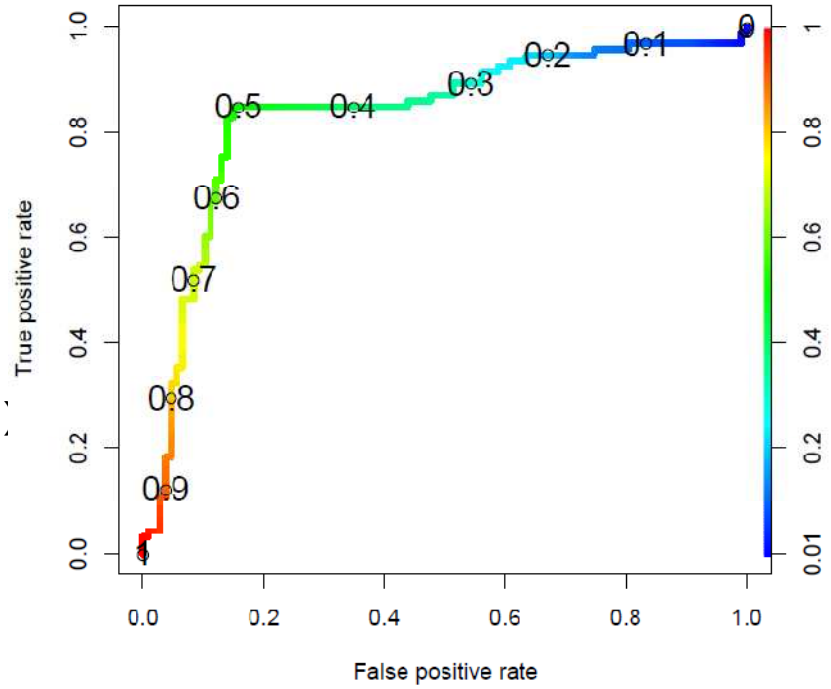
- 预测值和实际类别：

```
> head(as.data.frame(ROCR.simple))
  predictions labels
1   0.6125478      1
2   0.3642710      1
3   0.4321361      0
4   0.1402911      0
5   0.3848959      0
6   0.2444155      1
```

- 预测值的变化，根据cut-off分割
  - f(x) ≥ c ➔ spam
  - f(x) < c ➔ email

- ROC Curve：

```
pred <- prediction( scores, labels )
(pred: S4 object of class prediction)
perf <- performance( pred,
                     measure.Y, measure.X)
(pred: S4 object of class performance)
plot( perf )
```



# 参考文献

K. Hornik, C. Buchta, and A. Zeileis. Open-source machine learning: R meets Weka. Computational Statistics, 24(2):225-232, 2009.

I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco, 3rd edition, 2011.

T. Sing, O. Sander, N. Beerenwinkel and T. Lengauer: ROCR: Visualizing the performance of scoring classifiers, 2009

# 关于作者

主页：http://bjt.name

邮件：sunbjt@gmail.com

微博：@刘思喆

统计之都 R 语言板块：http://cos.name/cn/forum/15

感谢聆听！