[日期]

移动应用公司投诉数据分类分析

——基于 weka 的数据挖掘

汪文藻 141250131

南京大学软件学院

1 实验内容

本实验的基本内容是，根据现有的客服部门的投诉的纪录，使用 weka 中三种常见分类方法，KNN、朴素贝叶斯、决策树训练出分类模型，病使用校验数据对各个模型进行测试和评价，找出各个模型最优的参数值，并对三个模型进行全面评价比较，得到一个最好的分类模型以及该模型所有设置的最优参数。最后使用这些参数以及训练集和校验集数据一起构造出一个最优分类器，使用该分类器对用户投诉进行归类预测。

2 数据预处理

2.1 格式转换

在 mac 运行环境下，mac 链接不上 jdbc，所以使用 excel 转 arff 文件使用 waka。

转换方法：假如我们准备分析的文件为“tousu.xlsx”，则在 excel 中打开“tousu.xlsx”，选择菜单文件->另存为，在弹出的对话框中，文件名输入“tousu”，保存类型选择“CSV（逗号分隔）”，保存，我们便可得到“tousu.csv”文件；然后，打开 Weka 的 Exporler，点击 Open file 按钮，打开刚才得到的“filename”文件，点击“save”按钮，在弹出的对话框中，文件名输入“tousu”，文件类型选择“Arff data files（*.arff）”，这样得到的数据文件为“tousu.arff”。

2.2 建立数据训练集，校验集和测试集

通过统计数据信息 tousu 表，发现带有类标号的数据一共有 282 行，为了避免数据的过度拟合，必须把数据训练集和校验集分开，目前的拆分策略是校验集 82 行，训练集 200 行，。类标号为‘no-recurrence-events’（就是指投诉一次就被解决了的简单时间）的数据有 201

条，而类标号为‘recurrence-events’（多次投诉仍未解决）的数据有 81 条，为了能在训练分类模型时有更全面的信息，所以决定把包含 115 条 no-recurrence-events 类标号数据和 75 条 recurrence-events 类标号数据作为模型训练数据集，而剩下的 86 条类标号类 no-recurrence-events 的数据将全部用于校验数据集，这是因为在校验的时候，两种类标号的数据的作用区别不大，而在训练数据模型时，则更需要更全面的信息，特别是不同类标号的数据的合理比例对训练模型的质量有较大的影响。另外，我们为了做预测测试，我们将分类标号为 no-recurrence-events 的 86 行数据集的分类标号去掉，作为预测数据集。

3. 实验过程及结果截图

3.1 决策树分类

用“Explorer”打开刚才得到的“train-data.arff”，并切换到“Class”。点“Choose”按钮选择“tree (weka.classifiers.trees.j48)”，这是 WEKA 中实现的决策树算法。

选择 Cross-Validatioin folds=10，然后点击“start”按钮：

训练数据集训练决策树得出的结果

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         127               63.5    %
Incorrectly Classified Instances        73               36.5    %
Kappa statistic                          0.2057
Mean absolute error                      0.4375
Root mean squared error                  0.4971
Relative absolute error                 89.4681 %
Root relative squared error            100.5177 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.852     0.659      0.636     0.852      0.729      0.552    no-recurrence-events
                 0.341     0.148      0.63      0.341      0.443      0.552    recurrence-events
Weighted Avg.    0.635     0.442      0.634     0.635      0.607      0.552

=== Confusion Matrix ===

  a  b   <-- classified as
 98 17 |  a = no-recurrence-events
 56 29 |  b = recurrence-events
```

使用不同配置训练参数，得到的实验数据：

配置不同的叶子节点的实例个数					
实例数/叶节点	**2**	**3**	4	5	6
准确率	**63.5%**	**63.5%**	62.5%	62.5%	62.5%

结果分析：使用决策树时，每个叶子节点最优的实例个数为 2 或者 3。

校验数据集校验决策树得出的结果

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          74               86.0465 %
Incorrectly Classified Instances        12               13.9535 %
Kappa statistic                          0
Mean absolute error                      0.3956
Root mean squared error                  0.4464
Relative absolute error                 92.9218 %
Root relative squared error            104.8513 %
Total Number of Instances               86

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.86      0          1         0.86      0.925      ?        no-recurrence-events
                 0         0.14       0         0         0          ?        recurrence-events
Weighted Avg.    0.86      0          1         0.86      0.925      0

=== Confusion Matrix ===

  a  b   <-- classified as
 74 12 |  a = no-recurrence-events
  0  0 |  b = recurrence-events
```

初步结果分析：

使用决策树进行分类，对于已知的 86 个类标号为 no-recurrence-events 的数据进行比较准确的分类，准确率达到 86%；该数据一般，并且有一定的缺陷，因为该结果是以训练集的低准确率作为前提的。

3.2 朴素贝叶斯分类

点“Choose”按钮选择“bayes”，这是 WEKA 中实现的决策树算法。

选择 Cross-Validatioin folds=10，然后点击“start”按钮：

训练数据集训练 Naïve Bayes 得出的结果

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         132               66      %
Incorrectly Classified Instances        68               34      %
Kappa statistic                          0.2891
Mean absolute error                      0.3717
Root mean squared error                  0.4799
Relative absolute error                 76.0099 %
Root relative squared error             97.0403 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.765     0.482      0.682     0.765     0.721      0.701    no-recurrence-events
                 0.518     0.235      0.62      0.518     0.564      0.701    recurrence-events
Weighted Avg.    0.66      0.377      0.656     0.66      0.654      0.701

=== Confusion Matrix ===

  a  b   <-- classified as
 88 27 |  a = no-recurrence-events
 41 44 |  b = recurrence-events
```

得出的准确率为 66%

校验数据集校验 Naïve Bayes 得出的结果

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          68               79.0698 %
Incorrectly Classified Instances        18               20.9302 %
Kappa statistic                          0
Mean absolute error                      0.3128
Root mean squared error                  0.437
Relative absolute error                 73.4687 %
Root relative squared error            102.6477 %
Total Number of Instances               86

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.791    0          1         0.791     0.883      ?        no-recurrence-events
                 0        0.209      0         0         0          ?        recurrence-events
Weighted Avg.    0.791    0          1         0.791     0.883      0

=== Confusion Matrix ===

  a  b   <-- classified as
 68 18 |  a = no-recurrence-events
  0  0 |  b = recurrence-events
```

初步结果分析：

评价结果中准确率仅仅达到 79%，结果不是非常让人满意。

3.3 K 最近邻算法分类

点“Choose”按钮选择“laze->ibk”，这是 WEKA 中实现的决策树算法。

选择 Cross-Validatioin folds=10，然后点击“start”按钮：

训练数据集训练 KNN 得出的结果

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         133               66.5    %
Incorrectly Classified Instances        67               33.5    %
Kappa statistic                          0.2872
Mean absolute error                      0.3747
Root mean squared error                  0.5274
Relative absolute error                 76.6302 %
Root relative squared error            106.6543 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.817    0.541      0.671     0.817     0.737      0.674    no-recurrence-events
                 0.459    0.183      0.65      0.459     0.538      0.674    recurrence-events
Weighted Avg.    0.665    0.389      0.662     0.665     0.653      0.674

=== Confusion Matrix ===

  a  b   <-- classified as
 94 21 |  a = no-recurrence-events
 46 39 |  b = recurrence-events
```

使用不同配置训练参数，得到的实验数据：

配置不同的叶子节点的实例个数										
K 值	1	2	3	**4**	5	6	7	8	9	10
准确率	66.5%	64%	65%	**68.5%**	67%	66.5%	66.5%	66%	66%	67%

结果分析：使用 KNN 算法分类时，K 最优值为 4。

校验数据集校验 KNN 得出的结果

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          76               88.3721 %
Incorrectly Classified Instances        10               11.6279 %
Kappa statistic                          0
Mean absolute error                      0.3275
Root mean squared error                  0.3756
Relative absolute error                 76.9223 %
Root relative squared error             88.2179 %
Total Number of Instances               86

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.884     0          1         0.884     0.938      ?        no-recurrence-events
                 0         0.116      0         0         0          ?        recurrence-events
Weighted Avg.    0.884     0          1         0.884     0.938      0

=== Confusion Matrix ===

  a  b   <-- classified as
 76 10 |  a = no-recurrence-events
  0  0 |  b = recurrence-events
```

初步结果分析：

对使用 k=4 训练出来的分类模型进行校验的结果，准确率达到 88.3%，算是一个比较合理的分类结果。

3.4 三类分类方法的校验结果比较

	决策树	K 最近邻	朴素贝叶斯
校验准确率	86%	88.3%	79%
训练 混淆矩阵	a b <-- classified 98 17 | a = no-recurre 56 29 | b = recurrence	a b <-- classified 94 21 | a = no-recurr 46 39 | b = recurrenc	a b <-- classified 88 27 | a = no-recurre 41 44 | b = recurrence
校验 混淆矩阵	a b <-- classified 74 12 | a = no-recurr 0 0 | b = recurrenc	a b <-- classified 76 10 | a = no-recurr 0 0 | b = recurrenc	a b <-- classified 68 18 | a = no-recurr 0 0 | b = recurrenc
标准误差	0.4464	0.3756	0.437

比较结果分析：

根据上述数据，虽然决策树有比较好的准确率和相对较好的标准误差，但是在这背后，很有可能是以较大错误率作为代价，这点可以从训练混淆矩阵中得到印证；而朴素贝叶斯分类算法的准确率相对较低，而标准误差也较高，综合评价可以得知，当前最好的分类算法是 KNN 算法，并且它是最优设置参数为 k=4。

3.5 训练最优模型

使用预处理中的 buildmodel_data.arff 数据文件训练分类模型，算法为 k=4 的 KNN。

数据集训练 KNN 得出的结果

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         213               74.4755 %
Incorrectly Classified Instances        73               25.5245 %
Kappa statistic                          0.236
Mean absolute error                      0.3383
Root mean squared error                  0.4381
Relative absolute error                 80.8482 %
Root relative squared error             95.851  %
Total Number of Instances              286

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.965     0.776      0.746     0.965     0.842      0.669    no-recurrence-events
                 0.224     0.035      0.731     0.224     0.342      0.669    recurrence-events
Weighted Avg.    0.745     0.556      0.742     0.745     0.693      0.669

=== Confusion Matrix ===

   a   b   <-- classified as
 194   7 |   a = no-recurrence-events
  66  19 |   b = recurrence-events
```

使用最终模型对测试集进行预测结果

inst#	actual	predicted	error	probability distribution
1	?	1:no-recur	+	*0.999 0.001
2	?	1:no-recur	+	*0.667 0.333
3	?	1:no-recur	+	*0.727 0.273
4	?	1:no-recur	+	*0.999 0.001
5	?	1:no-recur	+	*0.833 0.167
6	?	1:no-recur	+	*0.864 0.136
7	?	1:no-recur	+	*0.999 0.001
8	?	1:no-recur	+	*0.625 0.375
9	?	1:no-recur	+	*0.999 0.001
10	?	1:no-recur	+	*0.999 0.001
11	?	1:no-recur	+	*0.846 0.154
12	?	1:no-recur	+	*0.667 0.333
13	?	1:no-recur	+	*0.857 0.143
14	?	1:no-recur	+	*0.999 0.001
15	?	1:no-recur	+	*0.65 0.35
16	?	1:no-recur	+	*0.875 0.125
17	?	1:no-recur	+	*0.733 0.267
18	?	1:no-recur	+	*0.999 0.001
19	?	1:no-recur	+	*0.857 0.143
20	?	1:no-recur	+	*0.8 0.2
21	?	1:no-recur	+	*0.875 0.125
22	?	1:no-recur	+	*0.857 0.143
23	?	2:recurren	+	0.334 *0.666
24	?	1:no-recur	+	*0.583 0.417
25	?	1:no-recur	+	*0.625 0.375
26	?	1:no-recur	+	*0.999 0.001
27	?	1:no-recur	+	*0.75 0.25
28	?	1:no-recur	+	*1 0
29	?	2:recurren	+	0.4 *0.6
30	?	1:no-recur	+	*0.786 0.214
31	?	1:no-recur	+	*0.999 0.001
32	?	1:no-recur	+	*0.857 0.143