

实验三 Aliens 游戏

银琦 (141220132、141220132@smail.nju.edu.cn)

(南京大学 计算机科学与技术系, 南京 210093)

摘要: 本次作业目的是使用监督学习来模仿人玩游戏的动作。运行 Train.java 玩游戏 (须成功), 得到数据文件, 然后使用机器学习包训练模型, 运行 test.java 测试模型的游戏完成情况。本文尝试了 naivebeyes 朴素贝叶斯算法, Logistic 对率回归算法, RBFNetwork 算法, J48 (C4.5 决策树) 算法等学习方法训练模型, 并进行测试。

关键词: 监督学习 naivebeyes 朴素贝叶斯算法 Logistic 对率回归 RBFNetwork J48 (C4.5 决策树)

1 学习方法介绍及性能对比

1.1 Naivebeyes朴素贝叶斯算法

1.1.1 算法介绍

朴素贝叶斯 (Naive Bayes) 算法的基本思路是计算文本属于类别的概率, 文本属于类别的概率等于文本中每个词属于类别的概率的综合表达式。主要流程如下: 计算各个独立特征在各个分类中的条件概率; 计算各类别出现的概率; 对于特定的特征输入, 计算其相应属于特定分类的条件概率; 选择条件概率最大的类别作为该输入类别进行返回。

1.1.2 结果统计

首先成功完成游戏, 得到了数据, 然后通过 naivebeyes 训练出了模型, 在 lv10-lv14 上进行测试, 统计了相关数据用于比较。模型及数据如下:

模型评估:

```
== Evaluation on training set ==
== Summary ==

Correctly Classified Instances      364          57.8696 %
Incorrectly Classified Instances    265          42.1304 %
Kappa statistic                    0.1704
Mean absolute error                 0.2107
Root mean squared error             0.4274
Relative absolute error             128.6323 %
Root relative squared error         149.9659 %
Total Number of Instances          629

== Detailed Accuracy By Class ==
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.583	0.376	0.862	0.583	0.696	0.645	0
	0.526	0.264	0.306	0.526	0.387	0.699	1
	0.75	0.061	0.073	0.75	0.133	0.961	2
	1	0.071	0.137	1	0.241	0.978	3
Weighted Avg.	0.579	0.35	0.748	0.579	0.631	0.66	

```
== Confusion Matrix ==

 a  b  c  d  <- classified as
294 136 31 43 | a = 0
47 60 7 0 | b = 1
0 0 3 1 | c = 2
0 0 0 7 | d = 3
```

lv10 测试结果

Round	Result	Score	TimeSteps
1	Win	53	692
2	Lose	31	373
3	Lose	6	151
4	Lose	31	331
5	Win	57	735
Average	40%	35.6	456.4

lv11 测试结果

Round	Result	Score	TimeSteps
1	Win	44	599
2	Lose	33	678
3	Lose	41	770
4	Lose	9	175
5	Lose	17	355
Average	20%	28.8	515.4

lv12 测试结果

Round	Result	Score	TimeSteps
1	Win	41	427
2	Win	41	427
3	Win	41	427
4	Win	41	427
5	Win	41	427
Average	100%	41	427

lv13 测试结果

Round	Result	Score	TimeSteps
1	Lose	13	217
2	Lose	31	319
3	Win	40	705
4	Lose	21	298
5	Win	40	604
Average	20%	29	428.6

lv14 测试结果

Round	Result	Score	TimeSteps
1	Lose	35	805
2	Lose	36	805
3	Lose	39	901
4	Lose	28	343
5	Win	44	437
Average	10%	36.4	658.2

1.2 Logistic 对率回归

1.2.1 算法介绍

Logistic 回归为概率型非线性回归模型，是研究二分类观察结果与一些影响因素之间关系的一种多变量分析方法。在分类情形下，经过学习后的 LR 分类器是一组权值，当测试样本的数据输入时，这组权值与测试数据按照线性加和，通过极大似然法来估计权值，给定数据集，对率回归模型最大化“对数似然”。

1.2.2 结果统计

通过 Logistic 训练出模型，在 lv10-lv14 上进行测试，统计了相关数据用于比较。模型及数据如下：
模型评估：

```

===== Evaluation on training set =====
===== Summary =====

Correctly Classified Instances      544          86.4865 %
Incorrectly Classified Instances    85           13.5135 %
Kappa statistic                    0.5251
Mean absolute error                 0.0948
Root mean squared error             0.2169
Relative absolute error             57.8797 %
Root relative squared error         76.09 %
Total Number of Instances          629

===== Detailed Accuracy By Class =====

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      -----  -
      0.956    0.504    0.884    0.956    0.919    0.894    0
      0.465    0.039    0.726    0.465    0.567    0.895    1
      0.75     0       1       0.75    0.857    0.999    2
      0.857    0.003    0.75    0.857    0.8     0.999    3
Weighted Avg.  0.865    0.411    0.855    0.865    0.853    0.896

===== Confusion Matrix =====

  a  b  c  d  <-- classified as
482 20  0  2 | a = 0
 61 53  0  0 | b = 1
  1  0  3  0 | c = 2
  1  0  0  6 | d = 3

```

lv10 测试结果

Round	Result	Score	TimeSteps
1	Win	43	677
2	Lose	42	659
3	Win	47	510
4	Lose	39	805
5	Lose	38	370
Average	40%	41.8	604.2

lv11 测试结果

Round	Result	Score	TimeSteps
1	Lose	31	684
2	Lose	41	809
3	Lose	27	524
4	Lose	21	451
5	Win	44	521
Average	20%	32.8	597.8

lv12 测试结果

Round	Result	Score	TimeSteps
1	Win	55	728
2	Win	55	721
3	Lose	57	869
4	Win	61	633
5	Lose	8	148
Average	60%	47.2	619.8

lv13 测试结果

Round	Result	Score	TimeSteps
1	Win	40	617
2	Win	40	617
3	Lose	27	337
4	Win	40	633
5	Win	40	617
Average	80%	37.4	564.2

lv14 测试结果

Round	Result	Score	TimeSteps
1	Lose	24	697
2	Lose	24	406
3	Lose	7	355
4	Lose	39	601
5	Win	49	508
Average	20%	28.6	513.4

1.3 RBFNetwork

1.3.1 算法介绍

RBF 神经网络属于前向神经网络类型，它能够以任意精度逼近任意连续函数，**RBF** 网络的结构与多层前向网络类似，它是一种三层前向网络。输入层由信号源结点组成；第二层为隐含层，隐单元数视所描述问题的需要而定，隐单元的变换函数是 **RBF** 径向基函数，它是对中心点径向对称且衰减的非负非线性函数；第三层为输出层，它对输入模式的作用作出响应。从输入空间到隐含层空间的变换是非线性的，而从隐含层空间到输出层空间变换是线性的。

RBF 网络的基本思想是：用 **RBF** 作为隐单元的“基”构成隐含层空间，这样就可将输入矢量直接（即不需要通过权连接）映射到隐空间。根据 **Cover** 定理，低维空间不可分的数据到了高维空间会更有可能是可分的。换句话说，**RBF** 网络的隐层的功能就是将低维空间的输入通过非线性函数映射到一个高维空间。然后再在这个高维空间进行曲线的拟合。它等价于在一个隐含的高维空间寻找一个能最佳拟合训练数据的表面。

当 **RBF** 的中心点确定以后，这种映射关系也就确定了。而隐含层空间到输出空间的映射是线性的，即网络的输出是隐单元输出的线性加权和，此处的权即为网络可调参数。由此可见，从总体上看，网络由输入到输出的映射是非线性的，而网络输出对可调参数而言却又是线性的。这样网络的权就可由线性方程组直接解出，从而大大加快学习速度并避免局部极小问题。

1.3.2 结果统计

通过 RBFNetwork 训练出模型，在 lv10-lv14 上进行测试，统计了相关数据用于比较。模型及数据如下：
模型评估：

```

== Evaluation on training set ==
== Summary ==

Correctly Classified Instances      508           80.7631 %
Incorrectly Classified Instances    121           19.2369 %
Kappa statistic                    0.0787
Mean absolute error                 0.1425
Root mean squared error            0.2669
Relative absolute error             87.0077 %
Root relative squared error        93.6504 %
Total Number of Instances          629

== Detailed Accuracy By Class ==

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.996    0.952    0.808    0.996    0.892    0.701    0
      0         0         0         0         0         0.712    1
      0.75      0         1         0.75    0.857    0.994    2
      0.429     0.003    0.6       0.429    0.5       0.98     3
Weighted Avg.  0.808    0.763    0.661    0.808    0.726    0.708

== Confusion Matrix ==

  a  b  c  d  <- classified as
502  0  0  2 |  a = 0
114  0  0  0 |  b = 1
 1  0  3  0 |  c = 2
 4  0  0  3 |  d = 3

```

lv10 测试结果

Round	Result	Score	TimeSteps
1	Win	45	592
2	Win	45	592
3	Win	45	592
4	Win	45	592
5	Win	45	592
Average	100%	45	592

lv11 测试结果

Round	Result	Score	TimeSteps
1	Win	48	447
2	Win	48	447
3	Win	47	534
4	Win	45	528
5	Win	45	529
Average	100%	46.6	497

lv12 测试结果

Round	Result	Score	TimeSteps
1	Win	41	427
2	Win	41	427
3	Win	41	427
4	Win	41	427
5	Win	41	427
Average	100%	41	427

lv13 测试结果

Round	Result	Score	TimeSteps
1	Lose	37	885
2	Lose	11	256
3	Lose	37	885
4	Lose	9	223
5	Lose	37	885
Average	0%	26.2	626.8

lv14 测试结果

Round	Result	Score	TimeSteps
1	Lose	18	253
2	Lose	36	442
3	Win	43	510
4	Win	43	510
5	Win	43	510
Average	60%	36.6	445

1.4 J48决策树

1.4.1 算法介绍

J48(C4.5)决策树算法是机器学习算法中一个分类决策树算法,是基于 ID3 算法进行改进的。相较于 ID3, J48 用信息增益率选择属性,克服了信息增益选择属性时偏向选择取值不多的属性的不足;同时在决策树构造的过程中剪枝;能够处理连续属性的离散化,能够处理不完整数据。

J48 有分类规则易理解、准确率高的优点。但同时在构造决策树的过程中,需要重复顺序扫描同一数据集,因此算法效率低。决策树的构造方法如下:

创建节点 N

If 训练集为空

 返回节点标记为 failure

If 训练集中所有记录属于同一类别

 以该类别标记节点 N

If 候选属性为空

 返回 N 作为叶节点,标记为训练集中最普通的类

For each 候选属性 attribute_list

 If 候选属性是连续的

 对该属性进行连续化

 选择候选属性 attribute_list 中具有最高信息增益率的属性 D

 标记节点 N 为属性 D

For each 属性 D 的一致值 d

 由节点 N 长出一个条件为 $D=d$ 的分支

 设 s 是训练集中 $D=d$ 的训练样本的集合

 if s 为空

 加上一个树叶,标记为训练集中最普通的类

 Else 加上一个有 C4.5 (R - {D},C, s) 返回的点

1.4.2 结果统计

通过 RBFNetwork 训练出模型，在 lv10-lv14 上进行测试，统计了相关数据用于比较。模型及数据如下：
模型评估：

```

== Evaluation on training set ==
== Summary ==

Correctly Classified Instances      701          83.5518 %
Incorrectly Classified Instances    138          16.4482 %
Kappa statistic                    0.2822
Mean absolute error                 0.1345
Root mean squared error             0.2594
Relative absolute error             81.4389 %
Root relative squared error         90.5166 %
Total Number of Instances          839

== Detailed Accuracy By Class ==

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      -----  -
      0.997    0.805    0.831    0.997    0.906    0.644    0
      0.149    0.003    0.92     0.149    0.257    0.647    1
      0.6       0       1       0.6     0.75    0.991    2
      0.7       0       1       0.7     0.824    0.993    3
Weighted Avg. 0.836    0.643    0.85     0.836    0.785    0.651

== Confusion Matrix ==

  a  b  c  d  <- classified as
668  2  0  0 | a = 0
131 23  0  0 | b = 1
  2  0  3  0 | c = 2
  3  0  0  7 | d = 3

```

lv10 测试结果

Round	Result	Score	TimeSteps
1	Lose	21	624
2	Win	45	586
3	Win	42	455
4	Lose	35	821
5	Lose	35	805
Average	40%	35.6	658.2

lv11 测试结果

Round	Result	Score	TimeSteps
1	Win	42	751
2	Win	42	631
3	Win	42	373
4	Win	43	483
5	Lose	35	809
Average	80%	40.8	609.4

lv12 测试结果

Round	Result	Score	TimeSteps
1	Win	41	427
2	Lose	32	349
3	Win	41	427
4	Win	41	427
5	Win	41	427
Average	80%	39.2	411.4

lv13 测试结果

Round	Result	Score	TimeSteps
1	Lose	27	337
2	Lose	37	885
3	Lose	37	885
4	Lose	37	885
5	Lose	27	456
Average	0%	33	689.6

lv14 测试结果

Round	Result	Score	TimeSteps
1	Win	45	518
2	Lose	25	648
3	Win	45	551
4	Lose	37	789
5	Lose	3	361
Average	40%	31	573.4

1.5 最终结果分析

比较了四种算法，其中 RBFNetwork 的成功率较高，但是过拟合风险较高，贝叶斯算法的成功率较低，相应花费的时间也较少，J48 成功率相对较高，耗时也较长。

2 修改特征提取方法

思路：原来的特征提取记录屏幕上每个位置的信息，以及 4 个游戏状态信息，我认为将记录的信息应当改进为，记录精灵的坐标，记录离精灵最近的外星人的坐标，记录和精灵纵坐标相同的外星人的坐标，记录和精灵纵坐标相同的子弹的坐标，记录精灵的移动方向，在搜索时仅搜索相关记录的坐标即可。

References:

- [1] http://blog.sina.com.cn/s/blog_5ae7a1de0100g8at.html 文本分类算法
- [2] <http://blog.csdn.net/moodytong/article/details/9633661> 朴素贝叶斯算法
- [3] 机器学习 周志华著
- [4] <http://blog.csdn.net/xuxurui007/article/details/18045943> C4.5 决策树