

HW3: Aliens游戏实验报告

匡亚明学院 张子谦 191240076 191240076@smail.nju.edu.cn

本人承诺该实验全程由本人独立自主完成，无抄袭或给予他人抄袭行为，代码已上传至本人github

摘要：本次实验学习了解了Random Forest, Naive Bayes, Logistic Regression, AdaBoost四种学习方法，观察对比四种方法训练出模型的性能。对特征提取方法进行改进，让训练的模型有了更好的泛化性，在实际运行中也有了更好的行为。

1. 引言

Aliens 游戏通过操控 Avator 进行 向左，向右，射击，静止不动四项操作，躲避炸弹，发射子弹来击杀敌人以取得胜利。通过玩家进行游戏，提取游戏局面特征与玩家行为产生训练集，借助weka训练出学习模型，将模型导入，运行测试，让Avator通过学习模型和当前局面输出操作行为，进行游戏。

2. 实验内容

Task1: 学习方法介绍

I. Random Forest

在介绍RandomForest前先介绍决策树学习方法。决策树是一种基于树结构的根据输入状态给出输出决策的学习方法。根据《机器学习》书中内容进行理解：

给定一组数据集 $D = \{(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)\}$ ，其中 x_i 表示状态 i ， y_i 表示当前状态对应的决策（模型训练过程默认该决策就是正确的）。对于每个 x_i ，可以认为它是由多个特征构成的，**特征集** $A = \{a_1, a_2 \dots a_n\}$ ，则 x_i 等价于 $(a_{1i}, a_{2i} \dots a_{ni})$ ；其中 a_{ki} 表示 x_i 状态下 a_k 特征的取值。

在此基础上，从根节点出发，从特征集中选出最优划分特征 a_* ，根据数据集中每个数据的状态的 a_* 特征的不同取值，拓展子节点并依次对数据集进行划分。不断递归向下拓展直至达到特定条件，叶节点即决策结果。

此时输入一个新的状态 x ，决策树便会根据这个状态的不同特征的值输出一个决策。

但是一个决策树往往会带来过拟合的情况，训练数据的不同带来的变化很大。因此为了对模型进行改善，提出了将许多决策树进行组合的学习方法，称为随机森林（RandomForest）。

其核心是通过数据集 D ，特征集 A 训练出多个不同的决策树，输入一个状态后，每个决策树都各自输出一个决策，选择出现次数最多的决策作为最终决策输出。（*democracy*）

训练不同的决策树的方法也是**随机**一词的体现，假设训练 k 个决策树，那么每个决策树都从数据集 D 中随机抽样，假设有 M 个样本，则只抽取 $m(m < M)$ 个数据，得到 k 个数据集 $\{D_1, D_2, \dots, D_k\}$ ，同时对于特征集也同样的进行随机抽样，得到 k 个特征集 $\{A_1, A_2, \dots, A_k\}$ ；那么根据 D_i 与 A_i 就可以训练出决策树 i ，进而构建出随机森林。

II. Naive Bayes

朴素贝叶斯分类器基于**贝叶斯公式**

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}$$

其中 x 表示输入状态, y 表示输出决策; 通过利用先验概率 $P(y)$ 和条件概率 (似然) $P(x|y)$ 来计算决策在观测到状态 x 后的后验概率 $P(y|x)$ 。

给定数据集 $D = \{(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)\}$, 特征集 $A = \{a_1, a_2 \dots a_n\}$, 且 x_i 等价于 $(a_{1i}, a_{2i} \dots a_{ni})$; 其中 a_{ki} 表示 x_i 状态下 a_k 特征的取值。

则贝叶斯公式可改写为:

$$P(y|x) = P(y|a_1, a_2, \dots a_n) = \frac{P(y)P(a_1, a_2 \dots a_n|y)}{P(a_1, a_2 \dots a_n)}$$

由于对于规模有限的训练样本, $f(a_1, a_2 \dots a_n|y)$ 并不能视作 $P(a_1, a_2 \dots a_n|y)$, 所以无法单纯依赖统计得到条件概率。所以朴素贝叶斯分类器假设各特征之间相互独立, 在独立性假设的前提下, 公式再次改写为:

$$P(y|x) = P(y|a_1, a_2, \dots a_n) = \frac{P(y)P(a_1, a_2 \dots a_n|y)}{P(a_1, a_2 \dots a_n)} = \frac{P(y)P(a_1|y)P(a_2|y) \dots P(a_n|y)}{P(a_1)P(a_2) \dots P(a_n)}$$

在此公式基础上, 分类器便能根据统计给定数据, 计算出在输入状态 x 下, 对应得概率值最高的决策 y 并输出。

III. Logistic

首先介绍二分类的 Logistic Regression。

输出决策 $y \in \{0, 1\}$, 理想的映射函数 $f: X \rightarrow \{0, 1\}$, 其中 $X = \{x_1, x_2 \dots x_n\}$ 表示所有可能观测到的状态构成的集合, 但是这样一个阶跃函数不连续, 所以作为替代引入 Sigmoid/Logistic 函数:

$$y = \frac{1}{1+e^{-x}}。$$

所以在 Logistic 函数中, 假设函数 $y = \frac{1}{1+e^{\theta^T x}}$; 其中 $x = \{a_1, a_2 \dots a_n\}$ 为观测状态,

$\theta = \{\theta_1, \theta_2 \dots \theta_n\}$ 为待求参数。

最终输出决策 $y_* = 1$ if $y > 0.5$; 0 else。

对于多分类问题, 引入 Softmax 函数 $P(y = j) = \frac{e^{\theta_j^T x}}{\sum_{i=1}^K e^{\theta_i^T x}}$, $y \in \{1, 2 \dots K\}$, $\theta_i = \{\theta_1^i, \theta_2^i \dots \theta_n^i\}$

是对于第 i 个分类的参数; 在经过这样的归一化处理后, 选取 i 使得

$\max\{P(y = k) | k = 1, 2 \dots K\} = P(y = i)$ 作为最后决策输出。

IV. Adaboost

“集成学习”通过构建并结合多个学习器来完成学习任务”, 将每个学习器的输出加权结合得出最终决策。先前介绍的随机森林便是一种集成学习的方式, 构造多个彼此之间“不存在强依赖关系, 可同时生成”的个体学习器, 并行化运行**投票**输出决策。而另一种集成学习的方式, 是“串行生成彼此之间存在强依赖关系的多个个体学习器”, 代表是AdaBoost算法。

通过数据集 D 训练出一个**基学习器**在根据它的表现对样本进行调整优化, 不断迭代直到数目达到预期设定, 最后进行加权结合。

具体过程为, 给定数据集 $D = \{(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)\}$, 目标学习器个数 T ;

则先设定数据集中每个样本权重相等, 记为 $D_1 = (1/m, 1/m, \dots 1/m)$, 同时每个学习器的权重记为 $\alpha = (\alpha_1, \alpha_2 \dots \alpha_T)$, 根据 D_1 和 D 利用基学习器进行学习得到学习器 h_1 , 统计学习器 h_1 的误差率 ϵ_1 , 并给予这个学习器权重 $\alpha_1 = \frac{1}{2} \ln(\frac{1-\epsilon_1}{\epsilon_1})$; 此时再依次更新样本权重为 D_2 , 如此迭代运算直至产生 T 个学习器。最后得到 $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$ 。

Task2: 性能对比

I. Random Forest

由一百个决策树，且每个决策树由9个特征构成。模型在训练集上评估如下

Correctly Classified Instances	470	91.6179 %
Incorrectly Classified Instances	43	8.3821 %
Kappa statistic	0.8609	
Mean absolute error	0.1128	
Root mean squared error	0.1884	
Relative absolute error	37.5479 %	
Root relative squared error	48.6595 %	
Total Number of Instances	513	

II. Naive Bayes

Correctly Classified Instances	270	52.6316 %
Incorrectly Classified Instances	243	47.3684 %
Kappa statistic	0.2798	
Mean absolute error	0.2386	
Root mean squared error	0.4454	
Relative absolute error	79.4373 %	
Root relative squared error	115.0236 %	
Total Number of Instances	513	

III. Logistic

Correctly Classified Instances	388	75.6335 %
Incorrectly Classified Instances	125	24.3665 %
Kappa statistic	0.5985	
Mean absolute error	0.1539	
Root mean squared error	0.2781	
Relative absolute error	51.2397 %	
Root relative squared error	71.8304 %	
Total Number of Instances	513	

IV. AdaBoost

Correctly Classified Instances	238	46.3938 %
Incorrectly Classified Instances	275	53.6062 %
Kappa statistic	0.0432	
Mean absolute error	0.2985	
Root mean squared error	0.3863	
Relative absolute error	99.3453 %	
Root relative squared error	99.7679 %	
Total Number of Instances	513	

从通过率看，Random Forest通过率最高，其次是Logistic Regression。在实际运行中发现Logistic Regression模型下 Avator 虽然经常躲避，但是发射子弹的命中率很低。AdaBoost模型下 Avator 不会移动，只是不断发射子弹，但是命中率很高，如果敌人不向中间位置抛掷炸弹，那么往往能获胜。Random Forest模型下 Avator 的行为较好，命中率较高，也会进行一些有意义的移动。Naive Bayes 模

型下 Avator 会一开始就移动到最右端，然后不断发射子弹，且命中率较低。

不过由于模型对Database的依赖性过强，且本人轻度手残，所能提供的Database并不能为模型的训练提供足够优秀的训练样本，所以基于此对模型性能做出的评价并不能作为准确的参考。

Task3: 修改特征提取方法

在原基础上添加了：(1)左、右、正上方是否有保护物；(2)左、右、正上方是否有炸弹；(3)左、右、正上方是否有敌人；(4)敌人总数；(5)正上方炸弹距自己距离，正上方敌人距自己距离。

但效果并不理想，结合游戏本身来看，原有的特征提取方法将整个地图每一个位置的物体信息作为特征传入，这样的特征提取方式会带来较多的噪声，且不同关卡的地图差异较大，一方面不具有泛化的效果，另一方面可能出现过拟合的问题。

于是将原本的特征提取方法删除，只保留添加的特征。但这带来了新的问题，对于添加的特征：左、右、正上方是否有保护物，将level4的游戏过程作为训练集传入训练出的模型在level3（一个几乎没有保护物的关卡）中表现时不会做出左右移动的动作。所以再度考虑将其删除。

最终保留特征：(1)左、右、正上方是否有炸弹；(2)左、右、正上方是否有敌人；(3)敌人总数；(4)正上方炸弹距自己距离，正上方敌人距自己距离。(5)原有的4项特征。

```
feature[index++] = bomb_up*8;
feature[index++] = bomb_left*4;
feature[index++] = bomb_right*4;
feature[index++] = enemy_up*16;
feature[index++] = enemy_left*8;
feature[index++] = enemy_right*8;
feature[index++] = enemyNum*32;
feature[index++] = enemyDis*32;
feature[index++] = bombDis*32;
// 4 states
feature[index++] = obs.getGameTick()/100;
feature[index++] = obs.getAvatarSpeed();
feature[index++] = obs.getAvatarHealthPoints();
feature[index++] = obs.getAvatarType();
```

对于不同特征我们根据经验赋予初始权重，将时间权重调整除以100，避免出现玩家在500ticks内完成游戏，训练模型在500ticks后不采取行动的情况出现。

虽然最终训练结果显示，学习器在训练集上的准确率有所降低，但是当将模型运用在实际游戏中产生了更好的效果。个人认为之前的准确率比现在要高更多的是因为过拟合的情况使得模型在训练集上正确率高，但面对新的情况难以采取正确行为。

Random Forest 模型中 Avator面对炸弹会有躲避行为，但不是一定能躲开。

Naive Bayes 模型中 Avator射击命中率提高，且出现了：当只剩两个敌人时，会移动到一端，停止发射子弹，当敌人即将达到上方，发射子弹，且命中率极高。

Logistic 模型对于炸弹的躲避行为效果有所下降，但也出现了只剩一个敌人时会停止不停发射子弹而是伺机发射子弹的行为。

Adaboost 模型开局会移动，但到达一端后就会一直发射子弹，仍然不会出现躲避的行为。

综上，特征提取函数修改后，各个模型都有了一些提升，但相应的部分模型也出现了一些行为的退化。

认为一方面数据量较少，仅能提供几百个数据；另一方面，数据集由玩家一次游戏产生，玩家的行为很大程度上决定了模型训练的好坏。

3. 结束语

本次实验学习了解并观察了利用四种学习方法进行监督学习生成模型的运行情况。从算法的差异化使得产生的模型运行的效果也有极大的差异。改进特征提取方法后，让训练的模型有了更好的泛化性，在实际运行中也有了更好的行为。但针对本次实验仍存在以下问题：

- 1、训练集过小，而模型对数据的依赖性很强，一次玩家游戏能提供的样本好坏上下限差距较大且效率较低。
- 2、或许存在更加适合此游戏的分类模型，仅尝试了4个学习模型，不够充分。

References:

《机器学习》 -- 周志华

https://blog.csdn.net/qq_18500245/article/details/95789288

<https://zhuanlan.zhihu.com/p/105722023>