

---

---

# 作业 3: Aliens 游戏

孙嘉欣 (201240005、201240005@smail.nju.edu.cn)

(南京大学 匡亚明学院大理科班)

**摘要:** 使用监督学习,通过提取特征、收集数据、训练学习模型来模仿人玩游戏的动作.对于原有的特征提取方法,应用 KNN 算法、随机森林、朴素贝叶斯分类器、AdaBoost 等学习方法,观察训练结果,对比模型性能,修改特征提取方法,得到更好的学习性能.

**关键词:** 监督学习;KNN 算法;C4.5 算法;随机森林;朴素贝叶斯分类器

## 1 监督学习方法介绍

### 1.1 K-近邻算法 (K-Nearest Neighbors, KNN)

KNN 算法是最简单的机器学习算法之一,用于分类和回归,基于相近样本的结果往往也相近,从而通过查询相似样本得到预测结果. KNN 算法首先查询与待预测样本最相近的  $k$  个训练样本,再使用  $k$  个样本的“多数表决”结果决定分类结果(即  $k$  个最近邻居中最常见的分类赋予该对象);使用  $k$  个样本的均值决定回归结果.

KNN 中参数  $k$  的选择非常重要. $k$  较小时,决策更为接近数据本身的分布,但较容易收到噪音和异常数据的影响; $k$  较大时,能够减小噪声的影响,但会使类别之间的界限变得模糊,对数据变化的敏感程度下降.最佳  $K$  值的选择取决于数据,可通过启发式技术获取.<sup>[1]</sup>

KNN 是一种非参数、惰性(lazy learning)、基于实例的学习,前期没有显式学习过程,只是简单地储存好训练样本,不会对数据进行大量训练,后期需要对新样本进行判断时才开始分析新样本与已存在的训练样本之间的关系,据此确定新样本的取值<sup>[2]</sup>.因此几乎没有训练时间,但在新样本进入时做判断的计算开销大,评估时间长.更适用于增量学习,由于数据变化而提前执行的学习很快就会过时的情况<sup>[3]</sup>.

KNN 算法采用向量空间模型来分类,训练样本是多维特征空间向量,其中每个训练样本带有一个类别标签,用向量间的距离作为相似度的指标.一般情况下,连续变量将欧氏距离作为距离度量(Euclidean distance),也可采用曼哈顿距离(Manhattan Distance),闵可夫斯基距离(Minkowski Distance)等.处理高维数据时,需适当降维,减少计算开销.

此处对 KNN 的介绍参考了第二学期《人工智能设计基础》课程中黄书剑老师的 PPT 内容.

### 1.2 C4.5 决策树算法 (C4.5)

决策树是逻辑最简单的机器学习算法,用于分类和回归.它采用树形结构,通过训练数据构建决策树,对未知的数据进行预测.树的根节点包含样本的全集,每次分叉选择一个拆分样本能力的特征作为分裂标准,基于 if-then-else 规则产生分支,直到到达叶节点,得到结果,新样本出现时根据训练好的决策树模型即可给出对应的预测.<sup>[4]</sup>

不同算法使用不同的指标来进行特征选择: ID3 算法(J. Ross Quinlan 于 1986 年提出)采用信息增益(信息熵的差)最大的特征; C4.5 算法(J. Ross Quinlan 于 1993 年提出)采用“信息增益比”指标选择特征; CART 算法(Breiman 等人于 1984 年提出)使用了基尼系数,利用基尼指数最小化准则选择特征.<sup>[5]</sup>

此处着重介绍的 C4.5 算法是 ID3 算法 (Iterative Dichotomiser 3 迭代二叉树三代) 的改进版,与之一样使用了信息熵的概念来构建决策树.信息熵表示事务不确定性的程度,即信息量的大小,信息增益 (information gain) 是在划分数数据集之前之后信息发生的变化,计算每个特征引入后的信息增益,带来信息增益最大的特征,即为最优划分特征.然而信息增益会偏向那些取值较多 (产生分支较多) 的特征,因此 C4.5 算法做了延伸和优化,引入信息增益率来克服信息增益的这种不足.信息增益率考虑引入某种特征进行分裂时分支的数量和尺寸,作为特征的内在信息,同样由信息熵衡量.信息增益率=信息增益 / 内在信息,公式如图 1.特征  $j$  的可能取值数越多,  $IV(j)$  的值通常会越大,从而信息增益率对可取值数目较少的特征有所偏好.C4.5 算法中,先利用信息增益选出高于平均值的特征,再用信息增益率进行选择.<sup>[6]</sup>

$$j = \arg \max_{j,s} \frac{I(X) - I(X|j)}{IV(j)}$$

$$IV(j) = - \sum_{i=1}^n \frac{|X_i|}{|X|} \log \frac{|X_i|}{|X|}$$

Fig.1 Formula of Information Gain Ratio  
图 1 信息增益率公式

### 1.3 随机森林 (Random Forest)

随机森林是一种通过构建大量决策树形成森林的集成学习算法,基本构成是多个没有关联的决策树.

随机森林算法将集成学习算法-装袋算法 (Bootstrap aggregating, Bagging) 应用于决策树.根据 Bagging 算法,从一个样本容量为  $N$  的样本中均匀、有放回地 (即自助抽样法) 选取  $m$  个样本容量为  $N$  的新样本,以选出的  $m$  个新样本分别构建决策树.决策树中的每个节点需要分裂时,随机选取一些特征 (需满足选出的特征数  $m \ll$  每个样本的总特征数  $M$ ), 再从选取出的特征中采取某种策略 (如 ID3 算法采取信息增益) 选出该节点的分裂特征,从而构建出一棵决策树.整个决策树形成过程中不进行剪枝.这样就建立出大量决策树,组成随机森林.新的输入样本进入时,让森林中的每一棵决策树分别进行预测,决策树预测“多数投票”决定分类结果,决策树预测的均值决定回归结果.

一棵可视的决策树可以向使用者形象地展示决策的结果和过程,但生长很深的树容易过学习,学习到高度不规则的模式,呈现低偏差和高方差<sup>[7]</sup>.随机森林平均了多个决策树以降低方差,不容易过拟合,但在某些噪音较大的分类或回归问题上还是会过拟合<sup>[7]</sup>.且训练速度比较快,易并行方法;可以处理大量的输入变量;在决定类别时还能评估特征的重要性,判断出不同特征之间的相互影响;如果有很大一部分的特征遗失,仍可以维持准确度<sup>[8]</sup>.

### 1.4 朴素贝叶斯分类器 (Naive Bayes classifier, NBC)

贝叶斯分类法是基于贝叶斯定理的统计学分类方法,朴素贝叶斯法在此基础上加以特征条件独立假设,实际特征间可能相互依赖或有些特征由其他特征决定,故称这一假设为“朴素” Naive.朴素贝叶斯分类器是利用朴素贝叶斯概率模型的简单概率分类器,通过预测一个给定的元组属于一个特定类的概率,来进行分类.

具体模型原理解释如下<sup>[9]</sup>:

首先,对应的分类可看作一个独立的类变量  $C$ ,其条件依赖于若干特征变量  $F_1, F_2, \dots, F_n$ ,于是我们目标是计算在不同特征变量条件下发生不同类别的  $C$  的条件分布  $p(C|F_1, \dots, F_n)$ . 根据贝叶斯定理

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \text{ 可将原式化为 } p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}.$$

其中,分

式中的分母部分只由特征  $F_i$  给出,与  $C$  无关,取不同的  $C$  分母都不变,故可以看作一个常数;只需关心分子部分

$p(C) p(F_1, \dots, F_n|C)$ , 其等价于联合概率  $p(C, F_1, \dots, F_n)$ . 对于联合概率,重复使用链式法则,可将该式写成一系列条件概率乘积的形式:

$$\begin{aligned}
p(C, F_1, \dots, F_n) & \\
&\propto p(C) p(F_1, \dots, F_n | C) \\
&\propto p(C) p(F_1 | C) p(F_2, \dots, F_n | C, F_1) \\
&\propto p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3, \dots, F_n | C, F_1, F_2) \\
&\propto p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3 | C, F_1, F_2) p(F_4, \dots, F_n | C, F_1, F_2, F_3) \\
&\propto p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3 | C, F_1, F_2) \dots p(F_n | C, F_1, F_2, F_3, \dots, F_{n-1}).
\end{aligned}$$

此时,由于“朴素”(Naive)的条件独立假设:假设每个特征  $F_i$  对于其他特征  $F_j, i \neq j$  是条件独立的,可知

$p(F_i | C, F_j) = p(F_i | C)$ , 因此可将条件概率的形式化简,得到:

$$\begin{aligned}
p(C | F_1, \dots, F_n) &\propto p(C, F_1, \dots, F_n) \\
&\propto p(C) p(F_1 | C) p(F_2 | C) p(F_3 | C) \dots \\
&\propto p(C) \prod_{i=1}^n p(F_i | C).
\end{aligned}$$

故,类变量  $C$  的条件分布可以表达如下, $Z$  是依赖于  $F_1, F_2, \dots, F_n$  的缩放因子,当特征变量的值已知时为常数:

$$p(C | F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i | C)$$

朴素贝叶斯分类器就是利用如上条件概率模型,计算出给定的元组属于某个类的概率分布,再选出最有可能的那个(即最大后验概率决策准则),就给出了分类的预测.分类公式可表达为:

$$\text{classify}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c).$$

从分类公式可见,只要保证正确类的后验概率比其他类要高,分类器就可以得到正确的分类.因此即使概率模型无法算出精确的概率估计,也不会影响分类结果的正确性,从而 NBC 分类器可以有足够的鲁棒性去忽略朴素贝叶斯概率模型存在的缺陷.<sup>[9]</sup>

和决策树模型相比,朴素贝叶斯分类器发源于古典数学理论,有着坚实的数学基础,以及稳定的分类效率.对大量训练和查询时具有较高的速度,每个特征的分布都可以独立地被当做一维分布来估计,当样本的特征个数增加时就不需要使样本规模呈指数增长,减轻了维数灾难带来的阻碍;所需估计的参数很少,对缺失数据不太敏感.<sup>[10]</sup>

但是特征独立假设在现实中常常是不准确的,这给 NBC 模型的正确分类带来了一定影响,分类决策存在一定的错误率.在属性个数比较多或者属性之间相关性较大时,NBC 模型分类效率比不上决策树模型;而属性相关性较小时,NBC 模型的性能良好.

## 2 基于原特征提取方法的不同学习模型对比

### 2.1 现有的特征提取方法及训练数据特征

现有的特征提取方法是记录当前游戏画面上每个位置的信息(即每个坐标上的物体种类 `itype`),以及 4 个游戏状态信息(游戏时间 `GameTick`、Avatar 的速度 `AvatarSpeed`、血量 `AvatarHealthPoints` 和状态 `AvatarType`).

将每一时刻提取到的游戏画面的特征和玩家按下的按钮（左、右、空格）作为训练数据保存,玩家按下的按钮通过一个分类变量 `class` 表示,4 种取值: 0-无操作,1-空格（发射子弹）,2-左,3-右。

收集 4 个游戏状态信息的代码部分有错误,数组下标重复,4 个信息保存到的是同一个位置,这里一开始就做了修正。

收集数据时选择的关卡为 level 3.采取的游戏策略不是一直保持发射子弹或不停左右移动,而是根据初始位置适当调整到方便击杀的位置后就尽量保持不动,根据子弹发射速度和怪兽前进速度,算准时间发射,在怪物刚出现在最上方一排时就直接击杀,只有在子弹下落且上方无遮挡物时做少量移动.采取这样的策略基本可以使怪物刚出现,一到达 Avatar 上方时就被子弹击杀,迅速获胜.因此在玩 level 3 时收集到的数据只有开局适当左移和适当发射子弹的操作,对应收集数据中 `class` 特征的分布如图 2。

## 2.2 训练结果

### 2.2.1 KNN

如图 3,KNN 算法的分类准确率为 88.2521%; 发射子弹的操作会与无操作混淆,移动操作也较容易与无操作混淆.性能表现还算不错,开局后会适当左移,但移动到位后,子弹发射虽然有短暂间隔,但和判断位置后的射击相比,比较贴近于持续子弹射击,不能很好根据怪兽的位置预测射击。

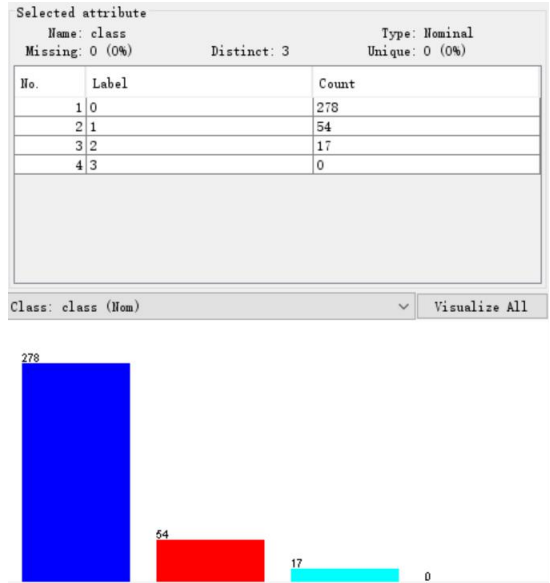


Fig.2 Distribution of attribute class

图 2 特征 class 分布

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

Correctly Classified Instances	308	88.2521 %
Incorrectly Classified Instances	41	11.7479 %
Kappa statistic	0.6469	
Mean absolute error	0.0631	
Root mean squared error	0.2351	
Relative absolute error	36.6593 %	
Root relative squared error	80.7239 %	
Total Number of Instances	349	

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.932	0.31	0.922	0.932	0.927	0.84	0
	0.685	0.058	0.685	0.685	0.685	0.853	1
	0.706	0.006	0.857	0.706	0.774	0.8	2
	0	0	0	0	0	?	3
Weighted Avg.	0.883	0.256	0.882	0.883	0.882	0.84	

== Confusion Matrix ==

a	b	c	d	<- classified as
259	17	2	0	a = 0
17	37	0	0	b = 1
5	0	12	0	c = 2
0	0	0	0	d = 3

Fig.3 Cross-validation of KNN

图 3 KNN 模型的交叉检验结果



Fig.4 Game results of KNN model  
图 4 KNN 模型游戏结果

尝试模型在其他地图中的性能,发现开局的移动还是有些死板,初始位置在中央时仍会左移,甚至一度移动到最左端,但对于不同图还是做出了不同的移动距离调整,如图 4.推测原因是收集的特征中未考虑 Avatar 自身的位置,所以只是根据图中物体的分布决策开局左移操作的距离。

### 2.2.2 C4.5

C4.5 算法的分类准确率如图 5,90.8309%,准确率比 KNN 更高,能很好地复刻左移操作,但基本是持续一直发射子弹。

从图 6 的决策树上可见,精准复刻来源于决策树主要采用了游戏时长 GameTick 作为分裂特征。这也使得 C4.5 决策树过拟合,泛化能力较差,游戏中移动死板,灵活性差。按照游戏时长决定左移,则无论换到哪张图,开局左移的长度都固定。保持不动,开始射击子弹后,由于初始位置不同,地图不同,只有[1,2]和[9,2]两个位置的特征用于对怪兽位置的预判也并不佳。

— Stratified cross-validation —  
— Summary —

Correctly Classified Instances	317	90.8309 %
Incorrectly Classified Instances	32	9.1691 %
Kappa statistic	0.7206	
Mean absolute error	0.0698	
Root mean squared error	0.2006	
Relative absolute error	40.5464 %	
Root relative squared error	68.8754 %	
Total Number of Instances	349	

— Detailed Accuracy By Class —

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.953	0.268	0.933	0.953	0.943	0.835	0
0.667	0.041	0.75	0.667	0.706	0.806	1
0.941	0.003	0.941	0.941	0.941	0.969	2
0	0	0	0	0	?	3
Weighted Avg.	0.908	0.22	0.905	0.908	0.906	

— Confusion Matrix —

a	b	c	d	← classified as
265	12	1	0	a = 0
18	36	0	0	b = 1
1	0	16	0	c = 2
0	0	0	0	d = 3

Fig.5 Cross-validation of C4.5  
图 5 C4.5 模型的交叉检验结果

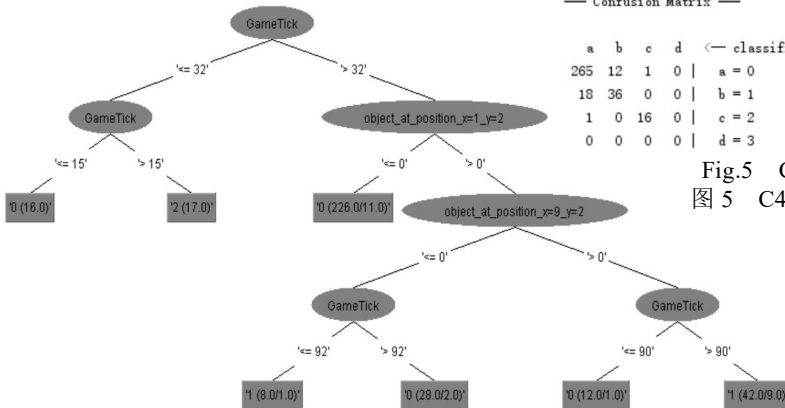


Fig.6 C4.5 decision tree  
图 6 C4.5 决策树

2.2.3 Random Forest

随机森林的交叉检验结果如图 7,分类的准确率为 88.2521%,总体和 KNN 算法相似.但更换不同地图,每次开局都会直接移动到最左侧,如图 8.持续一直发射子弹的学习结果和 C4.5 没太多差别.

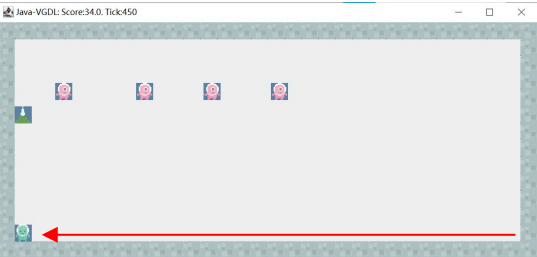


Fig.8 Game result of Random forests  
图 8 随机森林模型的游戏结果

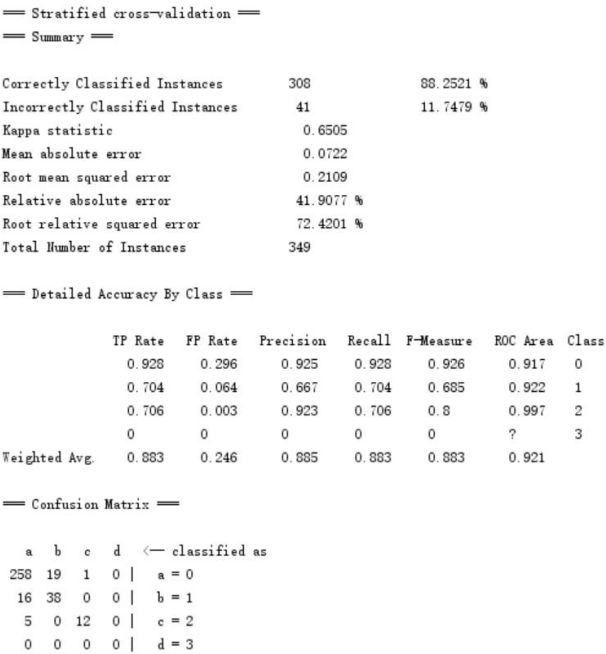


Fig.7 Cross-validation of Random forests  
图 7 随机森林模型的交叉检验结果

2.2.4 NBC

朴素贝叶斯模型的分类准确率如图 9,77.6504%,4 个模型中最低,错误主要是将大量无操作的情况判断为发射子弹.

实际游戏中模型的性能极差.发射子弹有一点断续,但也没有对怪兽的位置做出较好的预判,几乎无法杀死怪兽.且移动模式奇怪,开局会移动到最左侧,当怪兽下降到第三行后又会移动到最右侧,保持无操作状态,直到游戏失败结束,如图 10.这种怪兽下降到第三行后的现象在不同的图中都会出现.

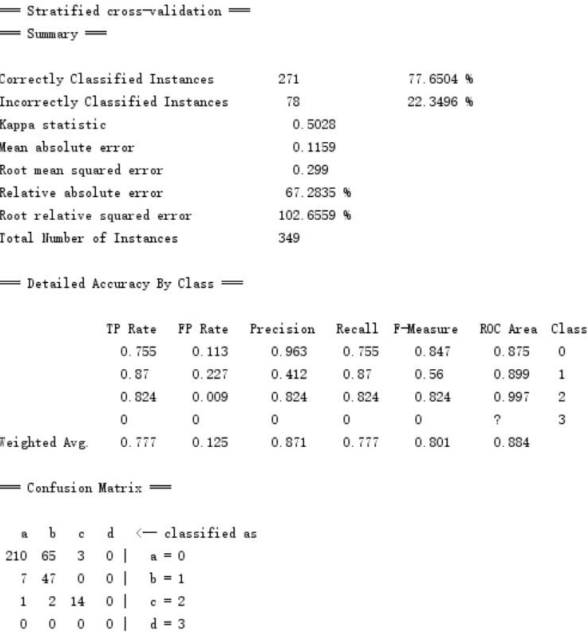


Fig.9 Cross-validation of NBC  
图 9 朴素贝叶斯分类器的交叉检验结果

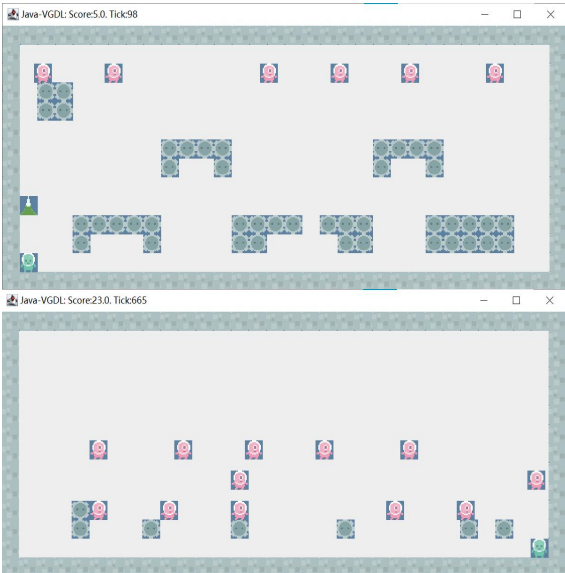


Fig.10 Game result of NBC  
图 10 朴素贝叶斯分类器的游戏结果



### 2.2.5 性能比较

KNN 的分类准确率不是最高,但性能相对最好.C45 的分类准确率最高,但存在过拟合,模型行为死板,性能一般.Random forest 和 KNN 的分类准确率并列第二,但性能较差.NBC 的分类准确率和性能都是最差,无法完成任务.

## 3 特征提取方法的改进

### 3.1 尝试的方案

因为人玩游戏的时候采取的行为模式简单,改进的目标是发射子弹时有准确的预判、针对怪兽的位置做适当追杀的移动.因此,改进方案的关键是让模型学会判断 Avatar 和怪兽间的相对位置.

选择改进方案时做了以下思考:

- 1) 4 个游戏状态中的 Avatar 的速度 AvatarSpeed、血量 AvatarHealthPoints 和状态 AvatarType 三项,在游戏中完全没有影响,收集到的数据中也显示完全没有变化,是废物特征,可以删除.原版有用的信息也可以看作只有游戏画面上每个位置上的物体种类 itype,和游戏时间 GameTick.
- 2) 如果不追求游戏时间,时间参数 GameTick 或许可以不用考虑,仅考虑空间参数.
- 3) 当前游戏画面上每个位置的信息(即每个坐标上的物体种类 itype)有意义,可以保留.但如果不考虑绝对位置,只考虑相对位置是否可行.
- 4) 要判断 Avatar 和怪兽间的相对位置,首先需要得到 Avatar 的位置,因为 Avatar 只在最下面一行活动,因此只需要提取位置的横坐标.其次是怪兽位置,无论是开局向怪兽移动,方便击杀,还是射击时的预判,针对的都是距离最近的怪兽,因此可以通过曼哈顿距离找到距离 Avatar 的怪兽,将两者间的曼哈顿距离作为特征.相对距离对追杀移动有意义,而子弹发射则最好考虑两者位置连线的斜率.

因此,改进时提取了以下五种特征:时间参数 GameTick(T)、游戏画面上每个位置上的物体种类 itype(P)、Avatar 的横坐标 Avatar\_X(X)、Avatar 与最近的怪兽间的曼哈顿距离 MinNPC\_AvatarDis(D)以及两者位置连线的斜率 NPC\_AvatarK(K),并做了不同组合的尝试.

**Table 1** Trial combination of features  
**表 1** 尝试的特征组合

组合	准确率
T+P (原版)	88.9831 %
T+P+X+D+K	88.9831 %
P+X+D+K	89.548 %
T+X+D+K	88.1356 %
T+P+D+K	88.9831 %
T+P+X+K	88.9831 %
T+P+X+D	88.9831 %
X+D+K	87.8531 %
T+D+K	88.1356 %
T+X+K	87.8531 %
T+X+D	87.8531 %
P+D+K	89.548 %
P+X+D	89.2655 %
P+X+K	89.2655 %
T+P+D	88.9831 %
T+P+K	88.9831 %
T+P+X	89.2655 %
P+X	89.548 %
P+K	89.548 %
P+D	89.548 %
T+X	86.1582 %
T+D	88.1356 %
T+K	88.1356 %
X+D	87.5706 %
X+K	87.5706 %
D+K	87.8531 %

以 KNN 算法为例,比较不同组合的准确率,如表 1,结果显示:

- 1) Avatar 的横坐标 (X)、Avatar 与最近怪兽间的曼哈顿距离(D)、两者位置连线的斜率(K),三个特征的效果相近,尤其是 D 和 K 几乎等价,原因是收集的数据里怪兽在第一排刚出现就被击杀,此时曼哈顿距离和连线斜率一一对应.
- 2) 时间参数 GameTick 确实可以不用考虑,去除特征 T 准确率反而上升.
- 3) 考虑绝对位置上的信息依旧有意义,只考虑相对位置准确率不佳.当前游戏画面上每个位置的信息 (即每个坐标上的物体种类 itype) 还包含着一些未被挖掘的信息.

尽管改进后准确率提高不大,最终还是选择了其中准确率最优 (89.548%) 的 P+X+D+K 组合的特征提却方法,即游戏画面上每个位置上的物体种类 itype、Avatar 的横坐标以及 Avatar 与最近怪兽间的曼哈顿距离和斜率,尝试去观察模型具体的游戏表现.

3.2 改进后模型间学习性能对比

Table 2 Performances of improved models  
表 2 改进后的模型表现

模型	准确率	表现	
		移动	射击
C4.5	88.7006 %	移动表现较好.开局左移,到位后保持不动开始射击.后期剩少数怪兽后会移动追击.	怪兽离得很远时不发射,一定距离时开始发射.但准确率一般,混淆矩阵也显示,模型会将许多发射 (1) 判断为无操作 (0). 且开始发射时怪兽还有一定距离,提前将一些 0 判断为 1.
KNN	89.548 %	开局向左移动,逼近怪兽.开始射击中,间断性短距离移动到最左.	近乎一直在发射,即将很多无操作状态 (0) 判断为发射 (1), 但命中率最高,相比于原先的提取方式有提升,很快即可杀死全部怪兽,不是无脑一直发射.
Random Forest	89.8305 %	移动行为介于 KNN 和 C45 之间.开局左移有些过,到位开始射击后也有短距离左移倾向,但数量少.后期剩少数怪兽时也会追击,幅度没 C45 大.	怪兽离得远时不发射,等到怪兽距离较近时才开始发射.虽然不是等到怪兽完全到位时发射,但在这一行为上学习得最好.但准确率没有 KNN 高,比 C45 稍好.
NBC	70.339 %	开局移动到最左侧,怪兽下降到第三行后移动到最右侧,保持不动.	移动到最左侧后会开始射击,射击数量极少.移动到最右侧后就不再射击,无操作保持到游戏失败.

总体上,改进后的模型在移动和射击两方面都有提高,行为更加规律.学会了较好的开局移动,一定程度上的追击; 射击从原先的一直保持发射,命中率较差,提升到等待怪兽靠近、有判断的射击.

其中,C45 模型的游戏性能提升最明显,虽然判断准确率下降,但这是对过拟合的缓解.行为表现明显比原先的版本灵活性提高,在各个图上都有比较好的表现.因为是基础的决策树模型,所以相对其他模型可能活动更少.混淆矩阵显示,模型不会将无操作状态 (0) 判断为左移 (2),但会将 2 误判为 0,所以没有 KNN 和 Random forest 在射击中期的移动倾向,不过应对关卡也够用.

Correctly Classified Instances	314	88.7006 %					
Incorrectly Classified Instances	40	11.2994 %					
Kappa statistic	0.6631						
Mean absolute error	0.0803						
Root mean squared error	0.2088						
Relative absolute error	44.0152 %						
Root relative squared error	69.5665 %						
Total Number of Instances	354						
===== Detailed Accuracy By Class =====							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.953	0.346	0.907	0.953	0.929	0.847	0
	0.59	0.044	0.735	0.59	0.655	0.837	1
	0.882	0	1	0.882	0.938	0.97	2
	0	0	0	0	0	?	3
Weighted Avg.	0.887	0.278	0.882	0.887	0.882	0.852	
===== Confusion Matrix =====							
a	b	c	d	<-- classified as			
263	13	0	0	a = 0			
25	36	0	0	b = 1			
2	0	15	0	c = 2			
0	0	0	0	d = 3			

Fig11 Cross-validation of C45  
图 11 C45 的交叉检验结果



KNN 模型的性能也有提高,从游戏得分中看,是其中最好的模型.射击准确率大有提高,虽然比不上人工射击,但也能很快赢得游戏.或许是前期怪兽连续出现时射击预判的表现不明显,由于很快取胜,也没有出现后期有漏网之鱼离得很远的情况,因此观察不到等待靠近的行为.如果用更复杂的关卡测试,或许可以观察到.至于移动行为的改变,原因主要是训练样本的问题.射击中本无需移动,但模型想要向右移动逼近最下层的怪兽.而样本中只有左移,模型无法修错,所以都被判断为左移,出现间断性向左移的行为.可以通过特征提取方法中判断怪兽左右,或使用多组不同样本进一步修正,KNN 模型的性能还能进一步提高.

Random forest 模型的准确率最高,从行为学习上看,也是学习得最好的模型.开局直接移动到最左侧的问题得到解决,尽管还是有些过.但模型体现了 Random forest 可以缓解决策树模型过拟合的优点,射击和移动上的问题都没有 KNN 和 C4.5 明显.

NBC 模型准确率有下降,且表现更差.原因应该是去除了相对独立的时间参数,加入了关联性很强的三个空间参数,使游戏中各特征独立的条件更加不满足.也说明朴素贝叶斯分类器并不适合这个游戏.

## 4 结束语

实验尝试了四种监督学习方法,不同的算法都展现了各自的特点,应用较多、发展完善的算法确实有其长处,较为基础的算法有一定的缺陷,但通过特征提取的优化,一定情况下也够用,概率模型的贝叶斯模型不太适合这一应用.

尝试修改特征提取方法,学习性能得到一定的提高,但不明显.原因主要是训练数据的问题.一是收集的样本较少,二是人工玩游戏迅速取胜,只有前期的样本,AI 会出错进入游戏后期,此时没有样本参考.但我技术可能太好,故意打不中会给予错误的样本,一个可能的策略是前期故意不射击,让怪兽向下跑,进入后期再射击,收集样本后,处理数据,把前期的样本去除;多收集几组数据后,再将样本合并.这也说明给予模型有代表性的训练样本的重要性,过拟合的模型泛化性会不太好.总之,特征提取方法和样本收集都有很大的提升空间.

## References:

- [1] k-nearest neighbors algorithm - Wikipedia
- [2] 机器学习中的急切学习方法(Eager Learning)和惰性学习方法(Lazy Learning) - 知乎 (zhihu.com)
- [3] Lazy learning - Wikipedia
- [4] 决策树 - Decision tree (3 个步骤+3 种典型算法+10 个优缺点) (easyai.tech)
- [5] 随机森林\_百度百科 (baidu.com)
- [6] 信息增益、信息增益率以及基尼指数详解 - 知乎 (zhihu.com)
- [7] Random forest - Wikipedia
- [8] 随机森林 - Random Forest (4 个实现步骤+10 个优缺点) (easyai.tech)
- [9] 朴素贝叶斯分类器 - 维基百科,自由的百科全书 (wikipedia.org)
- [10] 朴素贝叶斯 NBC (附: 5 个优点+4 个缺点) (easyai.tech)