NBA 数据挖掘

组员：2120161002 姜晓健（数据爬取、算法设计、文档编写）

2120160986 冯博思（数据预处理、代码实现、算法改进、ppt 制作）

1 实验背景

NBA 作为世界上水平最高的篮球联赛，吸引了无数的球迷。每一场 NBA 比赛都会产生大量的数据信息，如果能够有效地运用这些数据，便可以充分发挥出其潜在价值。

在每年赛季开始之前，大量的媒体专家都会对本赛季 NBA 常规赛的情况进行预测，这其中球队战绩和明星球员的个人数据是大家着重讨论的话题。及时而准确的完成对这些数据的预测一方面有利于各球队管理层在赛季进行前采用合适的决策，另一方面可以最大化商业公司的利益。本实验采用机器学习的方法在赛季开始前完成对本赛季 NBA 球队战绩以及个人数据的预测。

2 实验设计

本次实验主要分为两个目的：

（1）预测新赛季（2016-2017 赛季）常规赛每场比赛的胜负情况，进而得出每支球队的胜负场数。

（2）预测新赛季（2016-2017 赛季）球员常规赛的各项数据情况（包括得分、篮板、助攻、抢断、盖帽、失误等若干项数据），进而得到各项数据的榜首球员名单。

2.1 预测新赛季（2016-2017 赛季）常规赛每场比赛的胜负情况

2.1.1 实验介绍

在新赛季开始之前，NBA 官网上对于新赛季各项指标的预测铺天盖地，ESPN、腾讯体育等多家媒体会对一些球队以及球员的表现展开讨论。其中，球队战绩是最常被讨论的话题之一。一方面，对于博彩公司而言，及时而准确的了解每场比赛的胜负情况可以最大化其收入。另一方面，对于球队管理层而言，了解球队战绩可以帮助他们更好的在赛季进行时作出相应的决策。因此，在赛季开始之前，预测每场比赛的胜负情况具有重要的意义。

我们将基于 2015-2016 年的 NBA 常规赛及季后赛的比赛统计数据，判断每个球队的战斗力，进而预测当下正在进行的 2016-2017 常规赛每场赛事的结果。

2.1.2 实验知识点

使用逻辑斯特回归。逻辑斯特回归是在线性回归模型的基础上，使用 sigmoid 函数，将线性模型 wTx 的结果压缩到[0,1] 之间，使其拥有概率意义。其本质仍然是一个线性模型，实现相对简单。在广告计算和推荐系统中使用频率极高，是 CTR 预估模型的基本算法。同时，逻辑斯特模型也是深度学习的基本组成单元。

逻辑斯特回归属于概率性判别式模型，之所谓是概率性模型，是因为逻辑斯特模型是有概率意义的；之所以是判别式模型，是因为逻辑斯特回归并没有对数据的分布进行建模，也就是说，逻辑斯特模型并不知道数据的具体分布，而是直接将判别函数，或者说是分类超平面求解了出来。

2.1.3 实验步骤

1）对互联网上的数据进行爬取，获取比赛的统计数据。

2）对爬取得到的数据进行预处理。

3）比赛数据分析，得到代表每场比赛每支队伍状态的特征表达。

4）利用机器学习方法学习每场比赛与胜利队伍的关系，并对 2016-2017 赛季的常规赛进行预测。

（1）数据爬取

在实验 1 中，我们将采用 Basketball Reference.com 中的统计数据。在该网站中，有不同球员、队伍、赛季和联盟比赛的基本统计数据，如得分，助攻，篮板等各项指标以及每支球队每个赛季的胜负情况。由于我们预测球队战绩，因此数据采用 2015-2016 NBA Season Summary 中数据。

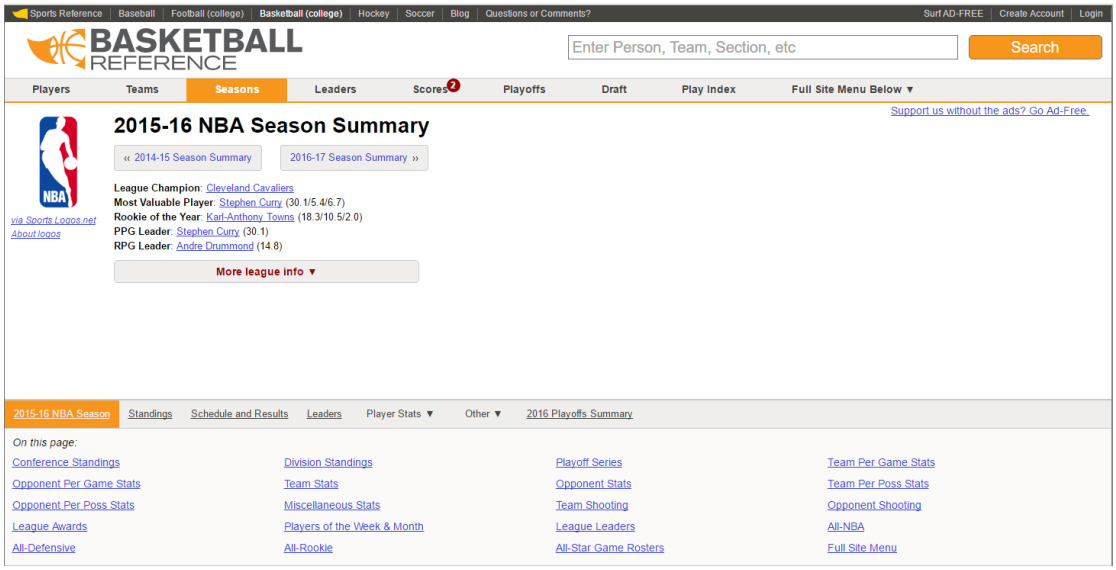


图 2-1　2015-2016 NBA Season Summary 数据

其中，2015-2016 球队的各项统计数据分别表示在 20 个表格中，我们将使用的表格为“Team Per Game Stats”、“Opponent Per Game Stats”、“Miscellaneous Stats”

表 2-1　Team Per Game Stats：每支队伍平均每场比赛的表现统计。

数据名	含义
Rk – Rank	排名
G – Games	参与的比赛场数（都为 82 场）
MP – Minutes Played	平均每场比赛进行的时间
FG – Field Goals	投球命中次数
FGA –Field Goal Attempts	投射次数
FG% -- Field Goal Percentage	投球命中率
3P--3-Point Field Goals	三分球命中次数
3PA--3-Point Field Goal Attempts	三分球投射次数
3P%--3-Point Field Goal Percentage	三分球命中率
2P--2-Point Field Goals	二分球命中次数
2PA--2-point Field Goal Attempts	二分球投射次数
2P%--2-Point Field Goal Percentage	二分球命中率
FT--Free Throws	罚球命中次数

FTA--Free Throw Attempts	罚球投射次数
FT%--Free Throw Percentage	罚球命中率
ORB--Offensive Rebounds	进攻篮板球
DRB--Defensive Rebounds	防守篮板球
TRB--Total Rebounds	篮板球总数
AST--Assists	助攻
STL--Steals	抢断
BLK -- Blocks	盖帽
TOV -- Turnovers	失误
PF -- Personal Fouls	个犯
PTS -- Points	得分

Opponent Per Game Stats：所遇到的对手平均每场比赛的统计信息，所包含的统计数据与 Team Per Game Stats 中的一致，只是代表的该球队对应的对手的。

表 2-2 Miscellaneous Stats：**综合统计数据。**

数据名	**含义**
Rk (Rank)	排名
Age	队员的平均年龄
W (Wins)	胜利次数
L (Losses)	失败次数
PW (Pythagorean wins)	基于毕达哥拉斯理论计算的赢的概率
PL (Pythagorean losses)	基于毕达哥拉斯理论计算的输的概率
MOV (Margin of Victory)	赢球次数的平均间隔
SOS (Strength of Schedule)	用以评判对手选择与其球队或是其他球队的难易程度对比。
ORtg (Offensive Rating)	每 100 个比赛回合中的进攻比例
DRtg (Defensive Rating)	每 100 个比赛回合中的防守比例
FTr (Free Throw Attempt Rate)	罚球次数所占投射次数的比例
3PAr (3-Point Attempt Rate)	三分球投射占投射次数的比例

TS% (True Shooting Percentage)	二分球、三分球和罚球的总共命中率
eFG% (Effective Field Goal Percentage)	有效的投射百分比（含二分球、三分球）
TOV% (Turnover Percentage)	每 100 场比赛中失误的比例
ORB% (Offensive Rebound Percentage)	球队中平均每个人的进攻篮板的比例
FT/FGA	罚球所占投射的比例
eFG% (Opponent Effective Field Goal Percentage)	对手投射命中比例
TOV% (Opponent Turnover Percentage)	对手的失误比例
DRB% (Defensive Rebound Percentage)	球队平均每个球员的防守篮板比例
FT/FGA (Opponent Free Throws Per Field Goal Attempt)	对手的罚球次数占投射次数的比例

毕达哥拉斯定律：

$$win\% = \frac{runs\ scored^2}{runs\ scored^2 + runs\ allowed^2} \qquad (2\text{-}1)$$

我们将使用以上三个表格来评估球队过去的战斗力，另外还需 2015-16NBA Schedule and Results 中的 2015-2016 年的 nba 常规赛及季后赛的每场比赛的比赛数据，用以评估 Elo score 值。在预测时，我们同样也需要在 2016-17 年的 NBA 常规赛比赛安排数据。

经过数据爬取，我们获得了 Team Per Game Stats，Opponent Per Game Stats 和 Miscellaneous Stats（之后简称为 T、O 和 M 表）这三个表格的数据。

（2）数据预处理

直接爬取到的数据含有较多的噪音（如缺失值、重复值，无关属性等），因此我们要对数据进行预处理，对于缺失值，用最高频率值来填补缺失值。同时将“比赛时间”、“比赛场数”等无关属性剔除。将重复值去重。最终将规则的数据存储到.csv 文件中。

（3）数据分析

在获取到数据之后，我们将利用每支队伍过去的比赛情况和 Elo 等级分来判断每支比赛队伍的可胜概率。在评价到每支队伍过去的比赛情况时，我们将使用到 T、O 和 M 表作为代表比赛中某支队伍的比赛特征。我们最终将实现针对每场比赛，预测比赛中哪支队伍最终将会获胜，但并不是给出绝对的胜败情况，而是预判胜利的队伍有多大的获胜概率。因此我们将建立一个代表比赛的**特征向量**。由两支队伍的以往比赛情况统计情况（**T、O 和 M** 表），和两个队伍各自的 Elo 等级分构成。

Elo Score 等级分制度：对于比赛的双方 A 和 B，假设 A 和 B 的当前等级分为 R_A 和 R_B，则 A 对 B 的胜率期望值为：

$$E_A = \frac{1}{1+10^{(R_B-R_A)/400}} \tag{2-2}$$

B 对 A 的胜率期望值为：

$$E_B = \frac{1}{1+10^{(R_A-R_B)/400}} \tag{2-3}$$

如果 A 在比赛中的真实得分 S_A（胜 1 分，和 0.5 分，负 0 分）和他的胜率期望值 E_A 不同，则他的等级分要根据以下公式进行调整：

$$R_A^{new} = R_A^{old} + K(S_A - R_A^{old}) \tag{2-4}$$

其中，根据等级分的不同 K 值也会做相应的调整：

- ≥2400，K=16
- 2100~2400 分，K=24
- ≤2100，K=32

因此我们将会用以表示某场比赛数据的**特征向量**为（加入 A 与 B 队比赛）：

[A 队 Elo score, A 队的 T,O 和 M 表统计数据，B 队 Elo score, B 队的 T,O 和 M 表统计数据]

（4）构建模型：

Logistic Regression 是一种经常用来解决分类问题的机器学习方法。它不仅可以预测样本的类别，还可以计算出分类的概率信息。对于一个 c 类分类问题，假设有 n 个训练样本 $\{x_1,...,x_n\}$， x_i 是 d 维向量，其类别标签是 $\{y_1,...,y_n\}$，其中 $y_i \in \{1,2,...,c\}$。Logistic 回归学习这样一个函数：

$$f(x)=g(\theta^T x)=\frac{1}{1+e^{-\theta^T x}}，\quad 其中\ g(z)=\frac{1}{1+e^{-z}} \qquad (2\text{-}5)$$

本次实验采用 Logistic Regression 进行二分类，类别标签 $y_i \in \{1,0\}$。对于给定的样本 x，其属于类别 1 的概率为 $f(x)$，属于类别 0 的概率是 $1-f(x)$。我们定义训练的代价函数为：

$$J(\theta)=-\frac{1}{n}\sum_{i=1}^{n} y_i \log f(x_i)+(1-y_i)\log(1-f(x_i))。 \qquad (2\text{-}6)$$

目标是使得代价函数 $J(\theta)$ 最小化，这里采用梯度下降的方法，利用公式（2-7）每次向当前点的梯度方法移动 α 的距离，经过多次计算之后就能得到其最小值。

$$\theta_j=\theta_j-\alpha\frac{\partial}{\partial\theta_j}J(\theta),(j=0...n) \qquad (2\text{-}7)$$

经过多次迭代达到终止条件后，当前的 $\theta_j(j=0...n)$ 即是我们最终求得的参数值。这样，我们便使用 Logistic Regression 建立分类模型完成了对数据的划分，最终利用训练好的模型在 16-17 年的常规赛数据中进行预测。

2.2 预测新赛季（2016-2017 赛季）球员常规赛数据

2.2.1 实验介绍

对于很多球迷而言，观看 NBA 比赛的一个重要原因是源于其对 NBA 著名球

星的热爱。因此，在新赛季开始前，一些 NBA 球星尤其是超级巨星的表现也就会备受期待。本次实验着重于预测球员在新赛季的个人数据（得分、篮板、助攻、抢断、盖帽、失误）。从而为后续的球员个人荣誉的预测奠定基础。

2.2.2 实验框架图

本次实验采用的框架图如图 2-1 所示，首先使用 k-means 算法对所有球员进行聚类，然后对于待预测球员，找到该球员所属类别下的所有球员，提取它们的数据作为 RNN 神经网络的训练集，最终利用训练好的模型对该球员本赛季数据进行预测。

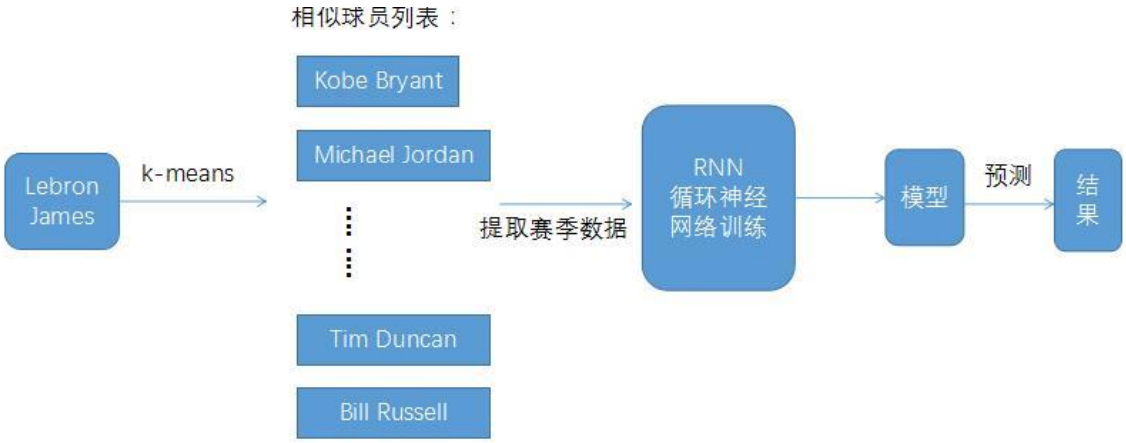


图 2-2 球员个人数据预测框架图

2.2.2 实验知识点

K-Means 聚类。K-Means 算法是硬聚类算法，是典型的基于原型的目标函数聚类方法的代表，它是数据点到原型的某种距离作为优化的目标函数，利用函数求极值的方法得到迭代运算的调整规则。K-Means 算法以欧式距离作为相似度测度，它是求对应某一初始聚类中心向量 V 最优分类，使得评价指标最小。算法采用误差平方和准则函数作为聚类准则函数。

算法的过程如下：

（1）从 N 个点随机选取 K 个点作为质心。

（2）对剩余的每个点测量其到每个质心的距离，并把它归到最近的质心的类。

（3）重新计算已经得到的各个类的质心。

（4）迭代 2-3 步，直至新的质心与原质心相等或小于指定阈值，算法结

束。

RNN 循环神经网络。RNN 目的是用来处理序列数据。在传统的神经网络模型中，是从输入层到隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNNs 之所以称为循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNNs 能够对任何长度的序列数据进行处理。但是在实践中，为了降低复杂性往往假设当前的状态只与前面的几个状态相关，图 2-2 和图 2-3 便是一个典型的 RNNs：

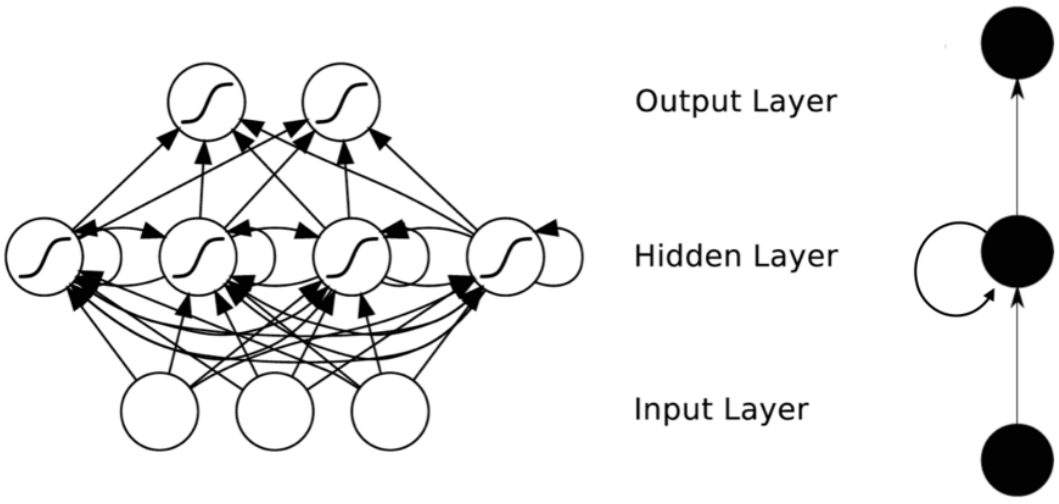


图 2-3 RNNs 的结构

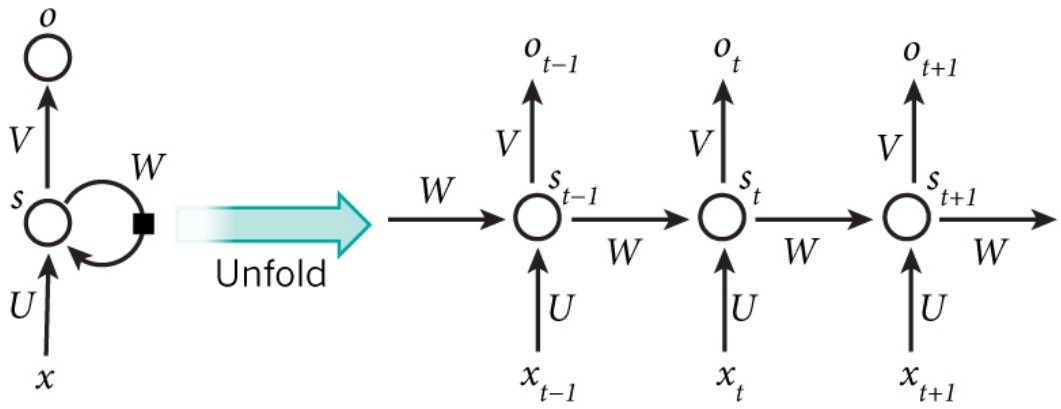


图 2-4 RNNs 的层与层之间传递情况

RNN 包含输入单元，输入集标记为$\{x_0, x_1, x_2, ..., x_t, x_{t+1}, ...\}$，而输出单元的输出集则被标记为$\{y_0, y_1, y_2, ..., y_t, y_{t+1}, ...\}$。RNN 还包含隐藏单元，我们将其输出集标记为$\{s_0, s_1, s_2, ..., s_t, s_{t+1}, ...\}$。这些隐藏单元完成了最为主要的工作。在图中：有一条单向流动的信息流是从输入单元到达隐藏单元的，与此同时另一条单向流动的信息流从隐藏单元到达输出单元。在某些情况下，RNNs 会打破后者的限制，引导信息从输出单元返回隐藏单元，这些被称为“Back Projections”，并且隐藏层的输入还包括上一隐藏层的状态，即隐藏层内的节点可以自连也可以互连。

2.2.3 实验步骤

1) 首先从互联网上爬取 NBA 球员的各项统计数据。
2) 对爬取到的数据进行预处理，得到形式较为规则统一的数据。
3) 根据球员数据对 NBA 球员进行聚类，将球风相近的球员聚在一起。
4) 当需要预测某个球员的各项数据指标时，将与该球员相似的球员的数据作为训练集，使用 RNN 循环神经网络进行训练，根据训练得到的模型预测该球员新赛季的数据。
5) 对实验结果进行分析。

（1）数据爬取

本次任务的目的是预测新赛季球员的各项数据，在进行机器学习训练时，我们需要大量的以往各个赛季各球员常规赛数据情况，因此，我们选择 http://www.stat-nba.com/index.php 进行数据爬取。该网站显示了历赛季各球员常规赛以及季后赛的数据情况。

本次实验所需要的数据有三种：

1） 所有 NBA 球员生涯场均数据

2） 所有球员各个赛季的数据

3） 新赛季需要预测的球员名单

第 1 类的数据爬取结果如图 2-5 所示，它显示了所有已退役的球员的生涯场均数据，一共有 4153 条记录。

	球员	出场	首发	时间	投篮	命中	出手	三分	命中	出手	罚球	命中	出手	篮板	前场	后场	助攻	抢断	盖帽	失误	犯规	得分	胜	负
1	迈克尔-乔丹	1072	1039	38.3	49.7%	11.4	22.9	32.7%	0.5	1.7	83.5%	6.8	8.2	6.2	1.6	4.7	5.3	2.3	0.8	2.7	2.6	30.1		
2	威尔特-张伯伦	1045		45.8	54.0%	12.1	22.5				51.1%	5.8	11.4	22.9			4.4				2.0	30.1		
3	埃尔金·贝勒	846		40.0	43.1%	10.3	23.8				78.0%	6.8	8.7	13.5			4.3				3.1	27.4		
4	凯文-杜兰特	703	703	37.4	48.8%	9.2	18.9	37.9%	1.8	4.7	88.2%	7.0	8.0	7.2	0.8	6.4	3.8	1.2	1.0	3.2	1.9	27.2	425	278
5	勒布朗-詹姆斯	1061	1060	38.9	50.1%	9.8	19.6	34.2%	1.4	4.0	74.0%	6.1	8.2	7.3	1.2	6.0	7.0	1.6	0.8	3.4	1.9	27.1	711	350
6	杰里-韦斯特	932		39.2	47.4%	9.7	20.4				81.4%	7.7	9.4	5.8			6.7				2.6	27.0		
7	阿伦-艾弗森	914	901	41.1	42.5%	9.3	21.8	31.3%	1.2	3.7	78.0%	7.0	8.9	3.7	0.8	2.9	6.2	2.2	0.2	3.6	1.9	26.7	466	448
8	鲍勃-佩蒂特	792		38.8	43.6%	9.3	21.3				76.1%	7.8	10.3	16.2			3.0				3.2	26.4		
9	乔治-格文	791		33.5	51.1%	10.2	19.9				84.4%	5.7	6.8	4.6	1.5	3.1	2.8	1.2	0.8		2.9	26.2		
10	奥斯卡-罗伯特森	1040		42.2	48.5%	9.1	18.9				83.8%	7.4	8.8	7.5			9.5				2.8	25.7		
11	卡尔-马龙	1476	1471	37.2	51.6%	9.2	17.8	27.4%	0.1	0.2	74.2%	6.6	8.9	10.1	2.4	7.7	3.6	1.4	0.8	3.1	3.1	25.0	952	524
12	科比-布莱恩特	1346	1198	36.1	44.7%	8.7	19.5	32.9%	1.4	4.1	83.7%	6.2	7.4	5.2	1.1	4.1	4.7	1.4	0.5	3.0	2.5	25.0	836	510
13	多米尼克-威尔金斯	1074	995	35.5	46.1%	9.3	20.1	31.9%	0.7	2.1	81.1%	5.6	6.9	6.7	2.7	3.9	2.5	1.3	0.6	2.5	1.9	24.8		
14	卡梅罗-安东尼	976	976	36.2	45.2%	8.8	19.5	34.6%	1.2	3.5	81.3%	5.9	7.2	6.6	1.8	4.8	3.1	1.1	0.5	2.8	2.9	24.8	533	443
15	卡里姆-贾巴尔	1560		36.8	55.9%	10.2	18.1				72.1%	4.3	6.0	11.2			3.6				3.0	24.6		
16	拉里·伯德	897	870	38.4	49.6%	9.6	19.3	37.6%	0.7	1.9	88.6%	4.4	5.0	10.0	2.0	8.0	6.3	1.7	0.8	3.1	2.5	24.3		
17	阿德里安-丹特利	955		35.8	54.0%	8.6	15.8				81.8%	7.2	8.7	5.7	2.3	3.4	3.0	1.0	0.2		2.7	24.3		
18	皮特-马拉维奇	658		37.0	44.1%	9.4	21.3				82.0%	5.4	6.6	4.2			5.4				2.8	24.2		
19	沙奎尔-奥尼尔	1207	1197	34.7	58.2%	9.4	16.1	4.5%	0.0	0.0	52.7%	4.9	9.3	10.9	3.5	7.4	2.5	0.6	2.3	2.7	3.4	23.7	819	388
20	德维恩-韦德	915	904	35.4	48.4%	8.5	17.5	28.7%	0.5	1.6	76.8%	5.9	7.7	4.8	1.2	3.6	5.7	1.6	0.9	3.3	2.3	23.3	539	376

图 2-5 所有已退役的球员的生涯场均数据

第 2 类数据爬取的结果如下，图 2-6 显示了所有球员每个赛季的数据，一共有 26838 条记录：

赛季	球队	出场	首发	时间	投篮	命中	出手	三分	命中	出手	罚球	命中	出手	篮板	前场	后场	助攻	抢断	盖帽	失误	犯规	得分	胜	负
16-17	克里夫兰骑士	74	74	37.8	54.8%	9.9	18.2	36.3%	1.7	4.6	67.4%	4.8	7.2	8.6	1.3	7.3	8.7	1.2	0.6	4.1	1.8	26.4	51	23
15-16	克里夫兰骑士	76	76	35.7	52.0%	9.7	18.6	30.9%	1.1	3.7	73.1%	4.7	6.5	7.4	1.5	6.0	6.8	1.4	0.6	3.3	1.9	25.3	56	20
14-15	克里夫兰骑士	69	69	36.1	48.8%	9.0	18.5	35.4%	1.7	4.9	71.0%	5.4	7.7	6.0	0.7	5.3	7.4	1.6	0.7	3.9	2.0	25.3	50	19
13-14	迈阿密热火	77	77	37.7	56.7%	10.0	17.6	37.9%	1.5	4.0	75.0%	5.7	7.6	6.9	1.1	5.9	6.3	1.6	0.3	3.5	1.6	27.1	52	25
12-13	迈阿密热火	76	76	37.9	56.5%	10.1	17.8	40.6%	1.4	3.3	75.3%	5.3	7.0	8.0	1.3	6.8	7.3	1.7	0.9	3.0	1.4	26.8	61	15
11-12	迈阿密热火	62	62	37.5	53.1%	10.0	18.9	36.2%	0.9	2.4	77.1%	6.2	8.1	7.9	1.5	6.4	6.2	1.9	0.8	3.4	1.5	27.1	45	17
10-11	迈阿密热火	79	79	38.8	51.0%	9.6	18.8	33.0%	1.2	3.5	75.9%	6.4	8.4	7.5	1.0	6.5	7.0	1.6	0.6	3.6	2.1	26.7	57	22
09-10	克里夫兰骑士	76	76	39.0	50.3%	10.1	20.1	33.3%	1.7	5.1	76.7%	7.8	10.2	7.3	0.9	6.4	8.6	1.6	1.0	3.4	1.6	29.7	60	16
08-09	克里夫兰骑士	81	81	37.7	48.9%	9.7	19.9	34.4%	1.6	4.7	78.0%	7.3	9.4	7.6	1.3	6.3	7.2	1.7	1.1	3.0	1.7	28.4	66	15
07-08	克里夫兰骑士	75	74	40.4	48.4%	10.6	21.9	31.5%	1.5	4.8	71.2%	7.3	10.3	7.9	1.8	6.1	7.2	1.8	1.1	3.4	2.2	30.0	45	30
06-07	克里夫兰骑士	78	78	40.9	47.6%	9.9	20.8	31.9%	1.3	4.0	69.8%	6.3	9.0	6.7	1.1	5.7	6.0	1.6	0.7	3.2	2.2	27.3	47	31
05-06	克里夫兰骑士	79	79	42.5	48.0%	11.1	23.1	33.5%	1.6	4.8	73.8%	7.6	10.3	7.0	0.9	6.1	6.6	1.6	0.8	3.3	2.3	31.4	47	32
04-05	克里夫兰骑士	80	80	42.4	47.2%	9.9	21.1	35.1%	1.4	3.9	75.0%	6.0	8.0	7.4	1.4	6.0	7.2	2.2	0.7	3.3	1.8	27.2	41	39
03-04	克里夫兰骑士	79	79	39.5	41.7%	7.9	18.9	29.0%	0.8	2.7	75.4%	4.4	5.8	5.5	1.3	4.2	5.9	1.6	0.7	3.5	1.9	20.9	33	46

图 2-6 所有球员每个赛季的数据

第 3 类数据爬取的结果如下，图 2-7 显示了新赛季我们要预测的球员名单。

 Abrines, Alex	亚历克斯-艾布里恩,阿莱克斯
 Acy, Quincy	埃希,昆西
 Adams, Steven	亚当斯,斯蒂文
 Afflalo, Arron	阿夫拉罗,阿隆
 Ajinca, Alexis	阿金萨,亚历克西斯
 Aldrich, Cole	阿尔德里奇,科尔
 Aldridge, LaMarcus	阿尔德里奇,拉马库斯

图 2-7　待预测的球员名单

（2）数据预处理

数据预处理主要从以下两个方面进行：

1) 由于早期 NBA 统计工具的不完备，所以很多早期球员的数据并没有精确的记录。因此在图 2-5 和图 2-6 的爬取结果中，含有很多的空白数据。我们需要采用适当的策略填补这些空白数据。

2) 图 2-5 和图 2-6 的爬取结果中含有较多的属性，而我们在进行球员数据预测时，只关心其中的若干个属性（如得分、篮板、助攻、抢断、盖帽等），多余的属性一方面增加了数据的维度，降低了神经网络训练的速度。另一方面，多余的属性对我们结果的预测不会产生太大的作用，甚至会降低结果预测的准确率。因此我们需要去除多余属性。

综合考虑以上两个方面的因素，我们采用以下的处理策略：根据一些专业信息，从这些特征中再选取一些相对重要的特征，包括：球员，赛季，时间，篮板，助攻，抢断，盖帽，失误，得分，胜，负。其中，部分网站只给出了球员姓名的中文信息，为方便统一信息，我们调用百度翻译的工具包，将其转化为英文处理，之后我们删除一些翻译不合理的噪声数据，因为它们对整体训练效果意义不大，而且为训练增加难度。

另外，从网站上爬取的数据并不完整，许多数值属性存在缺失值，根据网站信息排版规则，我们发现，排版位置相近的数据之间相似度也更高，所以在这里，我们用缺失值附近数据来进行填补，具体做法就是，把距离缺失值最近的前后两条非缺失数据的平均值作为填补值。除此之外，我们再手动添加一列

特征，胜负率，就是球员胜场次数与胜负总次数的比值，最后一共得到 12 个特征。最终处理完的数据分别存储在.csv 文件中。

（3）球员聚类

经过大量的统计得知，尽管 NBA 球员数量众多，但大体可以归为若干类。通过相关特征完成对球员的聚类，我们发现，位于同一类的球员，他们的球风相似，而且职业生涯发展轨迹也有共同点。当我们想要预测一名球员新赛季的数据时，根据与他类似的球员相同年龄段或相同身体状态下的表现可以基本得到该球员的数据。因此，我们首先要寻找与待预测球员相似的其他球员。本次实验采用 K-Means 聚类算法完成对 NBA 球员的聚类。

基于 K-Means 算法，我们根据 NBA 球员的数据完成对所有球员的聚类。算法的伪代码如下：

输入：NBA 所有球员生涯场均数据 player_career.csv。

输出：聚类完成后每个类别下球员名单。

Step1：从 player_career.csv 文件中读取数据，共 4148 条数据，设数据集规模 N=4148。每　　个数据包含 11 个属性特征。

Step2：类别数目 K=5。从所有点中随机选择 5 个点分别作为每个类的质心。

Step3：计算每个点到 5 个质心的最短距离（采用公式（2-8）欧式距离计算方法），并将该点归到最近的质心的类。

$$\mathrm{d}(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \qquad (2\text{-}8)$$

Step4：重新计算每个类的质心。质心采用公式（2-9）进行计算。

$$P_i(x,y)=\frac{1}{n_i}\sum_{j=1}^{n_i}p_{i,j}\quad (\mathrm{i}=1,2,3,4,5) \qquad (2\text{-}9)$$

Step5：如果新质心与原质心相等，转到 Step6，否则，重复 Step3-Step4。

Step6：输出每个类别下球员名单。

（4）球员预测

在 NBA 球员数据统计中，我们发现，一名球员当赛季的数据与其之前三个赛季的数据有较强的关系，在不考虑伤病等特殊情况下，相似球员的数据变化趋势大致相同。因此，在进行当前赛季球员数据预测时，我们可以根据其前三个赛季数据来预测当前赛季的数据。这也恰好符合 RNN 神经网络的特点，当前

节点的输出不仅与输入有关，还跟上个节点的输出有关系。本实验中，我们选择 RNN 神经网络完成对球员数据的预测。算法实现伪代码如下：

输入：待预测球员名单。

输出：2016-17 赛季该球员的场均数据（上场时间、得分、篮板、助攻、抢断、盖帽、失误）

Step1：从 player_list.csv 中导入待预测球员名单。

Step2：使用 K-Means 聚类算法完成对所有球员的聚类。

Step3：对于待预测球员 player，找出其所属的类别 class。

Step4：提取出 class 中所有球员的名单列表 player_set。

Step5:根据 player_set 中球员名称从 player_season.csv 中提取出每个赛季的数据（包括上场时间、得分、篮板、助攻、抢断、盖帽、失误），存储到 list 数组 player_data_list 中。

Step6:将每名球员连续三个赛季的数据作为训练集一条记录的特征值（共 21 个特征），将第四个赛季的数据作为该条记录的结果。

Step7：依据上述规则构建训练集，其中输入特征有 21 个，输出特征为 7 个。

Step8：使用 RNN 循环神经网络训练已经构建好的训练集，RNN 的各项参数如下：

```
learn_rate=0.001;
training_iters=100000;
batch_size=128;
n_input=7;
n_steps=3;
n_hidden=128;
n_classes=7。
```

Step9：对于待预测的球员，提取出该球员之前三个赛季的数据，利用之前训练好的模型预测当前赛季该球员的数据。

3 实验结果及分析

本次实验共完成了两项任务：

（1）在赛季初预测球队战绩。

（2）在赛季初预测球员常规赛数据。

针对实验任务（1），我们采用 Logistic Regression 的方法，利用训练集学习获胜球队与落败球队间的关系，之后对 2016-17 赛季的常规赛每场比赛的胜负进行预测。最终每场比赛的胜负情况如图 2（其中 probability 代表胜方赢的概率，1 代表预测正确，0 代表预测错误）所示，同时我们根据图 2 统计了每支球队的获胜场数以及落败场数，结果如图 1 所示。图 1 仅显示了部分球队的胜场情况，与 2016-2017 赛季常规赛情况相比，我们发现，“圣安东尼奥马刺”和“俄克拉荷马雷霆”的预测战绩都偏高。究其原因，主要是因为在 2016-17 赛季，两队都缺少了球队的头号球星，导致实际战绩比预想的要差。

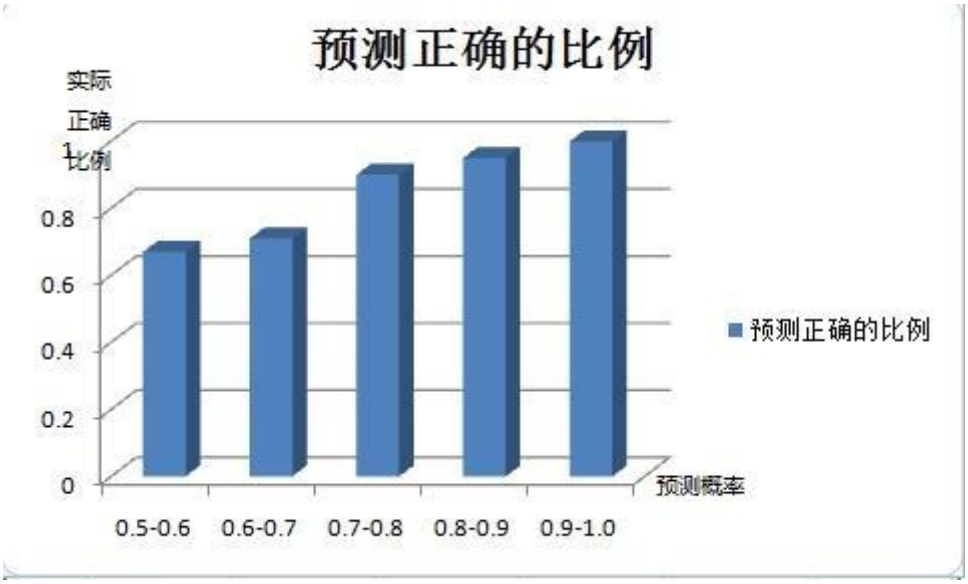
team	win	lose
Memphis Grizzlies	23	59
Los Angeles Clippers	66	16
San Antonio Spurs	77	5
Boston Celtics	59	23
Utah Jazz	42	40
Milwaukee Bucks	20	62
Sacramento Kings	23	59
Los Angeles Lakers	4	78
Charlotte Hornets	59	23
Indiana Pacers	52	30
Oklahoma City Thunder	73	9
Philadelphia 76ers	2	80
Portland Trail Blazers	58	23
Miami Heat	59	23
New York Knicks	19	63

图 1 球队胜负场数

win	lose	probability	
Cleveland Cavaliers	New York Knicks	0.898987187	1
Golden State Warriors	San Antonio Spurs	0.540894162	0
Portland Trail Blazers	Utah Jazz	0.656152799	1
Boston Celtics	Brooklyn Nets	0.906426791	1
Indiana Pacers	Dallas Mavericks	0.614560944	1
Houston Rockets	Los Angeles Lakers	0.788878587	0
Memphis Grizzlies	Minnesota Timberwolv	0.656860185	1
Charlotte Hornets	Milwaukee Bucks	0.720910835	1
New Orleans Pelicans	Denver Nuggets	0.544635926	0
Miami Heat	Orlando Magic	0.656771044	1
Oklahoma City Thunder	Philadelphia 76ers	0.950924212	1
Sacramento Kings	Phoenix Suns	0.566174449	1
Toronto Raptors	Detroit Pistons	0.711886128	1
Atlanta Hawks	Washington Wizards	0.659969548	1
Boston Celtics	Chicago Bulls	0.60102178	0
Los Angeles Clippers	Portland Trail Blazers	0.517598713	1
San Antonio Spurs	Sacramento Kings	0.881779626	1
Indiana Pacers	Brooklyn Nets	0.798311068	0
Dallas Mavericks	Houston Rockets	0.52712786	0
Detroit Pistons	Orlando Magic	0.683226136	1
Miami Heat	Charlotte Hornets	0.525541205	0
Golden State Warriors	New Orleans Pelicans	0.929979704	1
Oklahoma City Thunder	Phoenix Suns	0.935345966	1
Cleveland Cavaliers	Toronto Raptors	0.615032511	1
Utah Jazz	Los Angeles Lakers	0.879427911	1
Indiana Pacers	Chicago Bulls	0.547460388	0
Charlotte Hornets	Boston Celtics	0.523962284	0
Cleveland Cavaliers	Orlando Magic	0.868154154	1
Portland Trail Blazers	Denver Nuggets	0.707948246	1
Milwaukee Bucks	Brooklyn Nets	0.77077289	1
Memphis Grizzlies	New York Knicks	0.503407217	0
Atlanta Hawks	Philadelphia 76ers	0.90209412	1
Sacramento Kings	Minnesota Timberwolv	0.626749186	1
San Antonio Spurs	New Orleans Pelicans	0.929569486	1
Detroit Pistons	Milwaukee Bucks	0.755285227	1
Houston Rockets	Dallas Mavericks	0.561268084	1
Los Angeles Clippers	Utah Jazz	0.682314899	1
Washington Wizards	Memphis Grizzlies	0.525996288	0
San Antonio Spurs	Miami Heat	0.753721696	1
Oklahoma City Thunder	Los Angeles Lakers	0.956375015	1
Golden State Warriors	Phoenix Suns	0.951320259	1
Atlanta Hawks	Sacramento Kings	0.780516495	1
Chicago Bulls	Brooklyn Nets	0.722908974	1
Los Angeles Clippers	Phoenix Suns	0.895172151	1
Toronto Raptors	Denver Nuggets	0.844611255	1
Cleveland Cavaliers	Houston Rockets	0.791963512	1
Detroit Pistons	New York Knicks	0.744582321	1
Indiana Pacers	Los Angeles Lakers	0.901312131	1
Miami Heat	Sacramento Kings	0.779238568	1
Minnesota Timberwolves	Memphis Grizzlies	0.523616305	1

图 2 每场比赛胜方赢的概率

与此同时，我们对于在不同概率区间内预测正确的比例进行了分析，如图 3 所示：



由图 3 可知，当我们预测获胜的概率在 0.7 以上时，我们有很大的把握说明我们预测结果的准确性很高。

4 实验总结

本次实验共完成了两项任务：

（1）在赛季初预测球队战绩；

（2）在赛季初预测球员常规赛数据。

在（1）中，我们使用了 Logistic Regression 来学习获胜球队与落败球队之间的关系，从而完成对每场比赛球队胜率的预测。（2）中，我们首先使用 k-means 聚类算法将球员分为 5 类，每类中球员球风相似，职业生涯数据也相近。之后，当预测某一个球员的数据时，将与其类似球员的数据作为训练集，使用 RNN 神经网络进行训练，并利用训练好的模型对该球员进行预测。

参考文献

[1] Graves A. Supervised sequence labelling[M]//Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012: 5-13.

[2] 李航. 统计学习方法[J]. 清华大学出版社, 北京, 2012.

[3] Walker S H, Duncan D B. Estimation of the probability of an event as a function of several independent variables[J]. Biometrika, 1967, 54(1-2): 167-179.

[4] MacKay D. An example inference task: clustering[J]. Information theory, inference and learning algorithms, 2003, 20: 284-292.