



DeepTennis: CS230-2019 年秋季网球比赛

中期预测

斯文-勒纳
国际博物馆理事会
斯坦福大学
svenl@stanford.edu

迪皮卡-巴德里
斯坦福大学电子工程系
dipika98@stanford.edu

凯文-莫诺格
国际博物馆理事会
斯坦福大学
kmonogue@stanford.edu

摘要

在过去十年中, 体育分析领域经历了重大发展, 其中包括比赛预测领域。与赛前预测相比, 赛中实时预测仍是一个有待深入研究的领域。我们研究了利用长短期记忆 (LSTM) 模型基础结构的递归神经网络, 从比赛中期数据计算网球比赛的 "实时" 胜率。使用详细的逐点数据集 [1], 我们的模型旨在学习连续数据之间的深层关系, 并对每个选手在任何给定点之后赢得比赛的概率进行分类。结合传统的赛前预测先验, 我们的模型在所有给定点上获得了 79.5% 的平均准确率, 比传统的逐点统计模型高出约 3 个百分点 [2]。

1 引言

在过去的十年中, 应用于体育世界的分析方法层出不穷, 令人难以置信。与此同时, 体育博彩业也在迅速发展。据估计, 美国的体育博彩市场带来了数千亿美元的收入 [3]。现在, 大多数博彩公司都提供实时投注的机会, 即在整个比赛过程中更新赔率, 让用户可以根据比赛的进展情况下注。在罗杰-费德勒 (Roger Federer) 和拉斐尔-纳达尔 (Rafael Nadal) 的一场比赛中, Betfair 博彩交易所在比赛中期的投注额就高达 5000 万欧元 [4]。

更丰富的体育数据和更多的投注机会的出现, 自然而然地产生了比赛预测模型。网球因其得分离散的特性, 为这种分析提供了一个很好的缩影。大多数已发表的网球预测研究都侧重于使用赛前数据。此外, 现有关于比赛中期预测模型的研究大多是通过更新输入参数, 而不是在逐点数据之间建立时间关系。我们建议, 通过更好地结合逐点数据的时间性质,

并从数据中提取更复杂的特征（如疲劳或动量），RNN 应能提供超越传统赛中预测方法的优势。

我们算法的输入是一系列特征向量，代表网球比赛中的每个点。每个点都包含当前比赛得分、击球速度等数据，以及赛前

CS230：深度学习，2018 年冬季，加州斯坦福大学。(LateX 模板借自 2017 年 NIPS)。

中奖概率作为锚点。然后，我们在数据上训练 LSTM，将给定比赛的点数作为序列数据。该模型会输出 "玩家 2" 在比赛过程中每一点的获胜概率。

2 相关工作

如前所述，网球预测方法多种多样。Klaasen 和 Magnus 的论文[6]提供了一种使用分层马尔可夫模型解决这一问题的开创性方法，在以后的工作中经常被复制或修改。该模型将每位选手在发球时赢得一分的概率作为输入，然后利用该概率构建比赛结果树，得出每位选手赢得比赛的概率。其他研究对这种方法进行了扩展，例如在比赛过程中更新单点概率 [9] [13]。

Kovalchik (2016) [8]对已发表的结果进行了分析，同时还使用了另外两种方法：回归法和配对比较法。回归模型试图利用一组输入特征来预测胜率，例如 Clark 和 Dyte (2000 年) [10] 使用的模型。基于比较的模型使用之前的比赛结果来更新棋手的 "等级"，并使用这些等级来预测胜者--一个显著的例子就是 ELO 排名法。预测结果从 59% 到 72% 不等，基于比较的模型表现最好。Gollub (2017 年) 对这些模型以及几种改进的实现方法进行了另一次有力的评述。这篇论文在赛前和赛中数据上测试了修改后的基于点数的模型、逻辑回归模型和修改后的 ELO 模型。结果发现，赛前模型的准确率为 63% 至 69%，赛中模型的准确率为 71% 至 76.5%。

人们已经探索了使用现代机器学习方法进行赛前预测的方法。Cornman、Spellman 和 Wright (2017 年) 的前 CS 229 项目[11]测试了包括支持向量机、随机森林甚至神经网络在内的各种模型，发现准确率在 65% 到 70% 之间。Sipko (2015 年) 探索了用于赛前预测的基本逻辑回归和神经网络建模，发现比博彩市场预测有所改进。

3 数据集和特征

我们使用了杰夫-萨克曼 [1] 提供的 2011 - 2019 年网球大满贯（温布尔登、美国网球公开赛、法国网球公开赛和澳大利亚网球公开赛）逐点数据集。该数据包含比赛中每一分的特征向量，包括球员跑动距离、是否出现失误、获胜分类型等。我们从整个数据集的现有数据中，或者通过用字段的全局平均值替换空白值，提取了其中大约 41 个特征。我们还创建了新的字段，记录每位棋手需要再赢多少局或多少盘才能从该点赢下比赛。图 1 中显示了一些示例特征。

我们还使用了 Gollub [15] 中使用的 13 个模型所产生的赛前预测。我们将这些数据与萨克曼数据相匹配，并将赛前预测附加到每个点向量上。我们的目的是利用 Gollub 提供的各种方法，将稳健的赛前预测作为赛中预测的锚点。

图 1：单点特征向量样本

match_id	2019-usopen-1101
player1	Novak Djokovic
player2	Roberto Carballes Baena
winner	1
SetNo	2
P2Winner	0
P1DoubleFault	0
P2DoubleFault	0
P1UnfErr	0
P2UnfErr	0
P1NetPoint	0
P2NetPoint	0
P1NetPointWon	0
P2NetPointWon	0
P1BreakPoint	1
P2BreakPoint	0
P1BreakPointWon	1
P2BreakPointWon	0
Speed_MPH	116
RallyCount	4
P1DistanceRun	14.894
P2DistanceRun	17.513
p1_sets_to_win	2
p2_sets_to_win	3
p1_games_to_win	7
p2_games_to_win	17

我们将数据按年份分成 3373 个训练示例（比赛）、693 个开发集示例和 799 个测试示例。开发集是所有 2017 年的比赛，测试集是所有 2014 年的比赛，也就是 Gollub 中预测模型所测试的年份，这是目前最好的比较指标。

4 方法

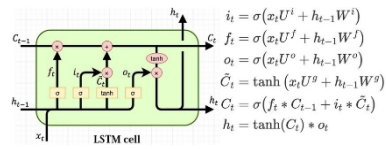
如前所述，我们方法的核心是将序列模型应用于网球比赛的逐点数据，以预测每位选手在比赛中每一分之后的获胜概率。我们使用 GRU 和 LSTM 模型进行了实验，但是 LSTM 模型的表现始终优于基于 GRU 的模型，因此我们在本报告中介绍了它们的结果。

长短期记忆（LSTM）网络是递归神经网络（RNN）的一个特殊类别。与普通神经网络不同，RNN 处理的是训练实例中时间上关联的数据点。长短期记忆（LSTM）网络利用门或过滤器结构来控制时间点（数据点）之间的信息流。

LSTM 的常见架构包括

构建模块称为单元（图 2）。每个单元都有两条输入/输出管道（上图中的 C_t 和 h_t ）。 C_{t-1} 负责将以前的单元向外传播，而 h_{t-1} 则将以前的单元向内传播。

图 2：单个 LSTM 单元 [16]



h_{t-1} 通过一个输入门 (i_t) 和一个遗忘门 (f_t)，以控制哪些信息被传递到

并与当前输入数据 x_t 结合，生成新的建议输出 C_t 。

C_t 然后通过使用

遗忘门和输入门将先前的输出 C_{t-1} 与新的拟议输出 C_t 结合起来。

最后，输出门 o_t 用于管理隐藏状态中流向下一个单元 h_t 的信息流，该信息流由 C_t 的 \tanh 激活产生。 C_t 和 h_t 水平传递给下一个时间单元，而 h_t 也传递给多层 LSTM 的下一层（图 3）。或者，如果这是最后一层，则在每个时间点 t 使用激活输出来产生 \hat{y}_t - 在我们的例子中，即玩家 2 获胜的概率。请注意， U 和 W 分别代表模型各组成部分的可训练参数集。

我们通过最小化损失函数来优化我们的模型，该函数将 \hat{y}_t 与真实值 y_t 进行比较。我们尝试了几种损失函数的方法。我们问题的一个有趣方面是，我们试图预测特定时间点的获胜概率，但这个值并没有真正的“基本事实”。作为替代，我们使用了比赛的最终获胜者，促使我们的模型朝着尽可能准确地预测获胜者的方向进行优化。我们曾尝试将问题视为回归问题（预测球员的获胜概率）和分类问题（预测哪位球员会获胜），结论是前者更适合我们的问题，并实施了二元交叉熵损失函数。我们还考虑根据每个点在数据示例中的时间位置对其损失进行加权。在尝试了不同的加权机制后，我们发现线性加权方案产生的结果最强。该方案减少了对早期数据点的惩罚，同时强调在后期阶段正确预测匹配。根据这些准则，我们的损失函数（ k 为匹配中的点数）为

$$L = - \sum_{i=1}^k \frac{1}{k} * (y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)) \quad (1)$$

我们尝试的另一种有趣的加权方法是让模型预测它对每个点的置信度，并以此为基础进行加权。现在的输出是二维的，即获胜概率和置信度值，损失也是这样计算的，置信度值为

c_i :

$$L = \lambda ||I - c||_2 - \sum_{i=1} c_i * (y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)) \quad (2)$$

我们发现，当模型开始训练时， c 输出一开始接近于一个 0 向量，但随着模型的训练，它逐渐趋近于一个 1 向量。这种损失计算方法与上述更简单的损失函数相比，精确度提高了不到 1%，但最终证明效果较差。

5 实验/结果/讨论

使用的主要评估指标是所有点的总准确率。精确度和召回率对我们的问题没有直观的意义（因为我们的真正目标是预测概率），因此我们还对校准进行了监测。下面将介绍优化过程，然后是结果。

我们探索了模型的几种超参数选择。我们使用 ADAM 优化器对学习率进行了简短搜索，结果是 $\alpha = 0.003$ ，并在第 20 和 30 个历元上切去 10%。我们还使用了 1 的批量大小，因为训练效率很高，而且这简化了对不同大小序列的损失计算。

我们探索的其他超参数包括模型深度

（堆叠单元数）和宽度（每个单元的隐藏状态维度）

。与一个模块相比，我们通过堆叠两个模块获得了 5% 左右的显著收益。然而，如果我们将深度增加到两个以上，尽管训练性能进一步降低（引入方差），但我们实际上注意到评估性能略有下降。我们通过在 LSTM 模块之间使用 dropout，在一定程度上缓解了方差的增加，但两个模块的深度仍然更优。就宽度而言，我们在隐藏状态大小为 50 时取得了最佳结果，这与 Hagan（2014 年）[18] 的经验法则基本一致：

$$N_h = \frac{N_s}{\alpha * (N_i + N_o)} \quad (3)$$

其中， N_h 是隐藏神经元的数量， N_s 是样本的数量， N_i 是输入的维度， N_o 是输出的维度， α 是缩放因子。对于深度增加的模型，减少堆栈中每层的宽度并不能提高性能。

我们利用 "放弃" 和 "提前停止" 来防止大型模型的过度拟合。我们确信，我们没有过度拟合训练数据集，因为我们的准确率在训练集、评估集和测试集上紧密匹配（79%）。

我们在表 1 中列出了测试结果和比较。"K-M Logit Elo" 是 Gollub [2] 使用的基于点的模型 [6]，利用逐点数据预测结果。"逻辑回归" 是一个简单的分类器，使用 13 个 Gollub 模型作为输入特征向量。我们的模型在测试集上的准确率为 79.5%，比 Gollub 最佳中期模型提高了 3%，比基本赛前预测器和 Kovalchik [8] 评述的最佳赛前结果提高了 6%。我们的模型还提供了每组数据后的数据点结果。

图 3：多层 LSTM 网络 [17]

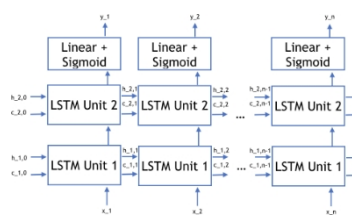


表 1：模型结果（百分比）

模型	净精度	第 1 套	第二套	第三套	第 4 套	第 5 套
DeepTennis	79.5	84	85	93	85	90
K-M Logit Elo (Gollub)	76.5	无	无	无	无	无
逻辑回归	73.4	无	无	无	无	无

表 2：按匹配进度分列的准确率（%）： 1.

比赛百分比	25	50	75	100
精确度	73.2	84.9	89.2	99

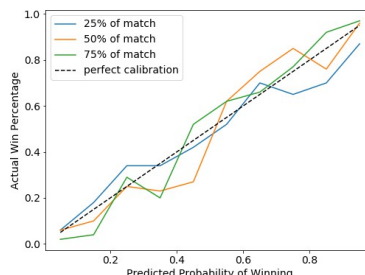
值得注意的是，我们的模型在第 4 盘之后的表现（85%）明显高于比赛开始时 或整体表现。顾名思义，比赛的比分在第 4 盘后打成平手，这表明我们的模型已经掌握了比赛的重要特征，而不仅仅是比分。我们的

准确率在第 3 盘后达到峰值，因为这包括许多以 3-0 结束的比赛。表 2 显示，我们模型的预测能力随着比赛的进行而增强。

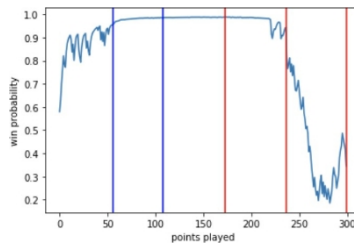
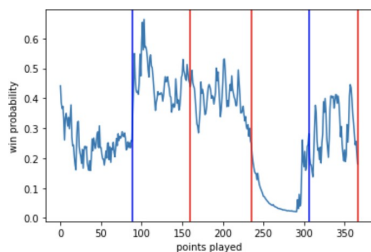
最后，我们研究了模型校准作为精确度或召回率数字的替代方法。图 4 显示了模型预测的获胜概率（以 10% 大小的桶为单位）和每个桶中比赛的实际获胜率。事实证明，模型的校准效果相对较好。

对网络中的单场比赛结果进行定性研究很有意义。参考图 5（蓝色垂直线为棋手 2 的胜局，红色为棋手 1 的胜局）：

图 4：校准



(a) 德约科维奇对费德勒，2014 年温布尔登网球公开赛 (b) 蒂姆对古尔比斯，2014 年美国网球公开赛



(a) 诺瓦克-德约科维奇鏖战五盘击败罗杰-费德勒

2014 年温布尔登网球锦标赛男子决赛。诺瓦克-德约科维奇以最被看好的姿态进入比赛，我们可以看到模型已经学会在比赛开始时纳入赛前先验。费德勒赢下第一盘后，他的获胜概率急剧上升。在德约科维奇连赢两盘后，比赛似乎已胜券在握，直到费德勒连赢 5 局，将比赛拖入第五盘。最终，德约科维奇取得了胜利。

(b) 2014 年美国网球公开赛的这场比赛就是我们的模型在训练损失方面表现非常糟糕的一个例子。我们看到这场比赛的获胜者是多米尼克-蒂姆（球员 1），但几乎在整场比赛中，我们的模型都预测他的对手欧内斯特-古尔比斯会获胜。事实上，这是一场备受关注的失利，20 岁的蒂姆首次参加美国网球公开赛，就在先失两盘的情况下扳回一盘[14]。

无论这些比赛中的故事情节多么出人意料，我们的模型似乎都能很好地捕捉到大热门的概念（古尔比斯持续的高胜率）或势头（德约科维奇连赢两盘并在第四盘险胜）。

6 结论/未来工作

与现有的网球比赛中期预测方法相比，采用双层、50 个隐藏节点的 LSTM 模型似乎提高了性能。与传统统计模型相比，我们的模型显示了网球积分之间的学习联系（势头、赛前信息的一致性、第 4 盘后的准确性），并更好地纳入了逐点数据的关系。

我们相信，如果加入更多数据或更详细的特征，我们的模型还能得到进一步改进。例如，更抽象的赛前特征（不仅仅是获胜概率）可能会提供与未来积分数据独特互动的信息，或者每个积分的可视化数据可能会很有用。再加上更详细的数据，模型结构的复杂性（尤其是宽度或深度）可能会比我们的实验更能提高性能。

如果能更多地联系实际应用，我们的项目也会得到改进。与实时投注市场数据的比较或功能增强将有助于分析，同时，为了正确实施我们的模型，需要对实时数据的实际收集进行研究。

7 捐款

所有团队成员都为该项目做出了同等贡献。具体来说，斯文专注于模型设计和培训，凯文专注于数据处理/功能工程和研究，迪皮卡专注于模型架构。

8 致谢

特别感谢郝胜在研究过程中提供的帮助和指导。我们还要感谢杰夫-萨克曼（Jeff Sackman）提供的数据集，以及雅各布-戈鲁布（Jacob Gollub）提供的网球预测综合资源，我们充分利用了这些资源。

9 代码库

<https://github.com/sven-lerner/DeepTennis>

参考资料

- [1] Jeff Sackman 提供: https://github.com/JeffSackmann/tennis_slam_pointbypoint
- [2] Gollub, Jacob. 从任何得分中得出职业网球比赛的获胜概率。Diss. 2019.
- [3] <https://www.legalsportsbetting.com/how-much-money-do-americans-bet-on-sports/>
- [4] Huang, Xinzhuo, William Knottenbelt 和 Jeremy Bradley. "从赛中投注赔率推断网球比赛进程"。毕业设计，伦敦帝国学院，南肯辛顿校区，伦敦，SW7 2AZ（2011 年）。
- [5] <https://ftw.usatoday.com/2017/02/super-bowl-espn-win-probability-atlanta-falcons-new-england-patriots-stats-tom-brady>
- [6] Klaassen, Franc JGM, and Jan R. Magnus. "预测网球比赛的获胜者"。European Journal of Operational Research 148.2 (2003): 257-267.
- [7] Easton, Stephen 和 Katherine Uylangco. "利用赛内投注市场预测网球比赛结果"。International Journal of Forecasting 26.3 (2010): 564-575.
- [8] Kovalchik, Stephanie Ann. "寻找网球胜负预测的GOAT"。体育定量分析期刊》12.3（2016 年）：127-138.
- [9] Madurska, Agnieszka M. "用于预测职业网球单打比赛结果的逐盘分析方法"。伦敦帝国学院计算机系统，Tech.Rep. (2012).
- [10] Clarke, Stephen R. and David Dyte. "利用官方评分模拟大型网球赛事"。International transactions in operational research 7.6 (2000): 585-594.
- [11] Cornman, Andre, Grant Spellman 和 Daniel Wright. "职业网球比赛预测和投注的机器学习"。(2017).
- [12] Sipko, Michal 和 William Knottenbelt. "预测职业网球比赛的机器学习"。伦敦帝国理工学院计算工程硕士毕业设计（2015 年）。

- [13] Bevc, Martin."从逐点数据预测网球比赛结果"。(2015).
- [14] <https://www.nytimes.com/2014/08/30/sports/tennis/dominic-thiem-tops-ernests-gulbis-to-reach-us-opens-third-round.html>
- [15] https://github.com/jgollub1/tennis_match_prediction
- [16] Varsamopoulos, Savvas, Koen Bertels, and Carmen G. Almudever."Designing neural network based decoders for surface codes." arXiv preprint arXiv:1811.12456 (2018).
- [17] Nicholas、Lee等人 "基于流量的网络入侵检测系统中的长短期记忆研究"。Journal of Intelligent & Fuzzy Systems Preprint (2018): 1-11.
- [18] Demuth、Howard B.等, 《神经网络设计》。马丁-哈根, 2014 年。
- [19] 导入 python 模块 pytorch、pandas、numpy、ipython、jupyter、torchvision、matplotlib。