

Previsão de estratégias vencedoras para LOL

Rama Alvim Sales Schiavo*
PUC-Minas
Belo Horizonte, Minas Gerais, Brasil
alvimrama@gmail.com

Rithie Natan Carvalhaes Prado
PUC-Minas
Belo Horizonte, Minas Gerais, Brasil
rithienatan@gmail.com

Ian Rodrigues dos Reis Paixão
PUC-Minas
Belo Horizonte, Minas Gerais, Brasil
ian.paixao@sga.pucminas.br

Luiz Gustavo Bragança dos
Santos
PUC-Minas
Belo Horizonte, Minas Gerais, Brasil
luiz.braganca@sga.pucminas.br

Guilherme de Andrade Moura
PUC-Minas
Belo Horizonte, Minas Gerais, Brasil
guileon64@gmail.com

Abstract

The rise on popularity of competitive MOBA-style online games (multiplayer online battle arena) promotes a need to analyze various aspects of the matches, aiming to create strategies after and before the match started. Through this article, the authors identified the most impactful decisions on a game match of LOL and how to chose most relevant characters to win a game through techniques of machine learning.

Keywords: datasets, neural networks, MOBA

1 Introdução

Com a ascensão da popularidade dos jogos online competitivos do estilo *MOBA* (*multiplayer online battle arena*), surgiu a necessidade de analisar diversos aspectos das partidas para obter cada vantagem possível em decisões tomadas durante e antes da partida começar. Por meio desse artigo, os autores identificaram as decisões mais impactantes durante uma partida do jogo *LOL* e a escolhas dos personagens mais relevantes ganhar uma partida através de *machine learning*.

2 Conhecendo o Jogo

O *MOBA* é um conceito de jogo online com partidas que se iniciam e finalizam durante a sessão em que o jogo começa, durando alguns minutos e no máximo pouco mais de algumas horas. Esse estilo de jogo vem ganhando cada vez

mais popularidade e é considerado altamente rentável pelos padrões de jogos online. Esse interesse não surgiu de forma súbita, uma vez que o primeiro jogo do estilo *MOBA* conhecido surgiu em 1998, mas desde então vem ganhando popularidade rapidamente.

Explicação do jogo é bem simples: cada partida é realizada entre dois times, cada time composto por cinco jogadores. Cada time começa em lados opostos do mapa e cada jogador controla um personagem escolhido previamente ou durante logo no começo da partida.

Existem recursos no jogo que podem ser revertidos para comprar equipamentos ou reviver o personagem mais rapidamente. Esses recursos são limitados e a definição de como esses recursos são utilizados dependem do tipo de personagem que o jogador escolhe. Definir estratégias diante de tantas variáveis é uma tarefa extremamente complexa, mas é possível obter as tomadas de ação que foram mais decisivas em partidas vencedoras. Essas tomadas de ação podem ser entendidas como prioridades momentâneas e atualizadas sempre que finalizadas.

3 Ferramentas utilizadas

O intuito dos autores não é apenas determinar o time vencedor, dado determinados parâmetros. O interesse maior é definir quais dados, quais parâmetros são realmente relevantes. E para isso os autores optaram por utilizar métodos caixa branca. Os autores utilizaram árvore de decisão e o *GridSearch* para buscar os atributos mais relevantes.

4 Dataset

A base de dados consiste em uma serie histórica de 51490 partidas ranqueadas de LOL, obtidas a partir de [dataset](#). A série inclui informações como personagens escolhidos e metas conquistadas ao longo da partida. Ao todo são contabilizados 61 atributos.

Os atributos a seguir foram selecionados seguindo o critério de metas/objetos que podem ser alterados ou conquistados durante a partida e que portanto devem influenciar a escolha de uma estratégia vencedora.

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1. winner -> classificação
2. firstBlood
3. firstTower
4. firstInhibitor
5. firstBaron
6. firstDragon
7. firstRiftHerald
8. t1towerKills
9. t1inhibitorKills
10. t1baronKills
11. t1dragonKills
12. t2towerKills
13. t2inhibitorKills
14. t2baronKills
15. t2dragonKills

5 Metodologia

A forma que encontramos para produzir os resultados foi direta, exigindo poucos tratamentos. A grande quantidade de amostras no *dataset* contribuiu para facilitar a obtenção dos resultados. Inicialmente lemos o *dataset*, eliminamos as linhas vazias, criamos uma árvore de decisão e em seguida buscamos os atributos reelevantes. Utilizamos o *cross validation* com 5 *folds* ao treinar a árvore.

5.1 Árvore de decisão

Utilizamos o método *DecisionTreeClassifier* contido na biblioteca *sklearn* do *Python 3.8* e separamos 30% das amostras para teste e 70% para o treinamento.

6 Resultados

6.1 Matriz de confusão

Classification report :				
	precision	recall	f1-score	support
1	0.97	0.97	0.97	7877
2	0.97	0.97	0.97	7570
accuracy			0.97	15447
macro avg	0.97	0.97	0.97	15447
weighted avg	0.97	0.97	0.97	15447

Figure 1. Fonte: autor

6.2 Encontrando estratégias

Utilizamos o método *GridSearchCV* contido na biblioteca *sklearn* do *Python 3.8* e geramos uma árvore com os melhores atributos. Essa árvore, partindo da raiz até as folhas contém os melhores atributos, partindo da maior relevância (raiz) até as metas menos impactantes (nas folhas). O esboço dessa estrutura pode ser visualizada a seguir:

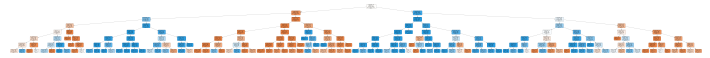


Figure 2. Fonte: autor

7 Conclusão

Como pretendíamos, obtivemos uma estratégia viável de vitória no *MOBA* utilizando o *dataset* pretendido. As diretrizes obtidas através da árvore representam ainda uma contra estratégia, uma vez que estratégias para os dois times ganharem são listadas na estruturas.

A métrica precisão observada na matriz de confusão está em 97% representando um alto índice de acerto.

7.1 Trabalhos Futuros

A próxima etapa é encontrar uma forma de encontrar condições favoráveis antes mesmo da partida começar. Isso inclui a escolha de heróis em cada time, o impedimento de escolha de um determinado herói do time adversário (os atributos *Ban-Heroes*) e a modelagem desse problema visando antagonizar o time rival pela escolha de personagens dele.

Infelizmente, estamos limitados ao uso de valores numéricos e não nominais pelos próprios métodos implementados na biblioteca *SKLearn*.

Tais alterações significariam drásticas mudanças no cerne dos algoritmos contidos nas bibliotecas utilizadas, e portanto não continuamos seguindo essa abordagem.

References

- [1] Joshua Eaton and David Mendonça. 2019. Testing an Integrated Team Effectiveness Framework in League of Legends. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 63 (11 2019), 240–246. <https://doi.org/10.1177/1071181319631466>
- [2] Hannah R. Gerber, Kevin Sweeney, and Erica Pasquini. 2019. Using API Data to Understand Learning in League of Legends: A Mixed Methods Study. *Educational Media International* 56, 2 (2019), 93–115. <https://doi.org/10.1080/09523987.2019.1614250> arXiv:<https://doi.org/10.1080/09523987.2019.1614250>
- [3] Marçal Mora-Cantalops and Miguel Ángel Sicilia. 2018. Player-centric networks in League of Legends. *Social Networks* 55 (2018), 149 – 159. <https://doi.org/10.1016/j.socnet.2018.06.002>
- [4] Kirk St.Amant (Ed.). 2018. *Commun. Des. Q. Rev* 6, 2 (2018).