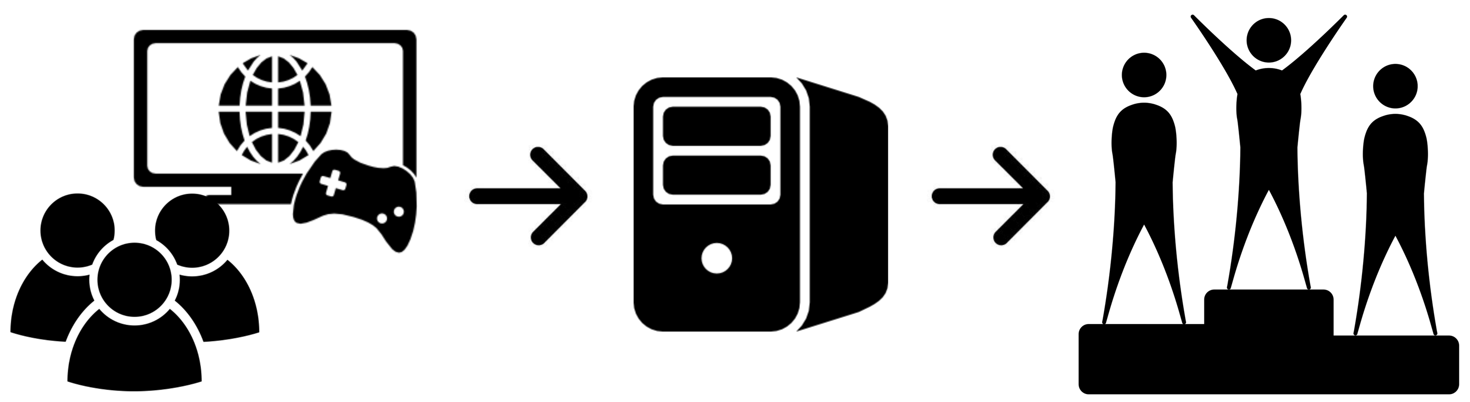
Player Behavior and Optimal Team Compositions in Online Games

Hao Yi Ong, Sunil Deolalikar, and Mark Peng
CS 229: Machine Learning Class Project

Introduction

- In online role-playing games, players work in teams to accomplish a common objective (e.g., defeating an opposing team)



- In a game, given the teams' player compositions and their player statistics, we want to predict the players' play style and forecast the game outcome
- **Player behavior**
  - In-game play style; e.g., prefers more offense-oriented strategies
  - Also encompasses skill level
  - Predict from player statistics
- **Team composition**
  - Types of players on a team, each classified by their play styles
  - Predict from game server database's match histories

Problem Description

- Given
  - **Match histories** containing participant IDs and match statistics
  - **Player statistics** containing player histories and overall game statistics
- Output
  - **Play style classifier** that groups players by their in-game tendencies given their game histories
  - **Outcome predictor** that guesses which team will win given the various team compositions
- In order to
  - *Gain insight* on player behaviors and game strategies
  - *Maximize accuracy* on predicting game outcomes

Target Game

- **League of Legends**
  - Multiplayer battle arena game with 27 million plays per day
  - Free online API to retrieve de-identified game data
  - Official guide provides clustering information for players based on in-game character choices (e.g., character with good defense)

Baseline Outcome Predictor

- **Features** are team compositions for each team in a game based on official game guide's clustering information (each character is mapped to a play style)
- **Logistic regression** with 10% hold-out cross validation
- **Data samples** with 120,000 training samples and 12,000 test samples
- **Poor accuracy** of 55.1% on training samples, 54.4% on test samples

Behavioral Clustering

Cluster Visualization

Results and Analysis

- **Best structure** We picked the top 8 structures based on the Bayesian score, ran parameter learning on them, and tested their prediction accuracies based on classifying test sets using inference
- **Error rates** Varying the number of missing variables, the error rates were 15–19% on 1,000 test samples, which are competitive with the 16% baseline
- **Runtime** Depending on the number of missing variables, a single prediction can take up to 30 seconds—much more expensive than our baseline

Summary: Classification results with varying number of missing features

#missing	Error rate	False positive	False negative	Pred time (sec)
3	0.1662	0.0363	0.1299	3.754
5	0.1511	0.0302	0.1208	6.437
8	0.1339	0.0272	0.1067	9.672
10	0.1299	0.0252	0.1047	11.859

Conclusion and Extensions

- Our model is suitable for diabetes prediction on existing databases (even with incomplete records) to detect patients at risk for diabetes
- Incorporate expert knowledge in feature selection and deciding the structure
- Consider different inference methods for runtime and accuracy improvement

Acknowledgements

We thank Professor Andrew Ng, the instructor team, and fellow students for their help with and feedback on our work.