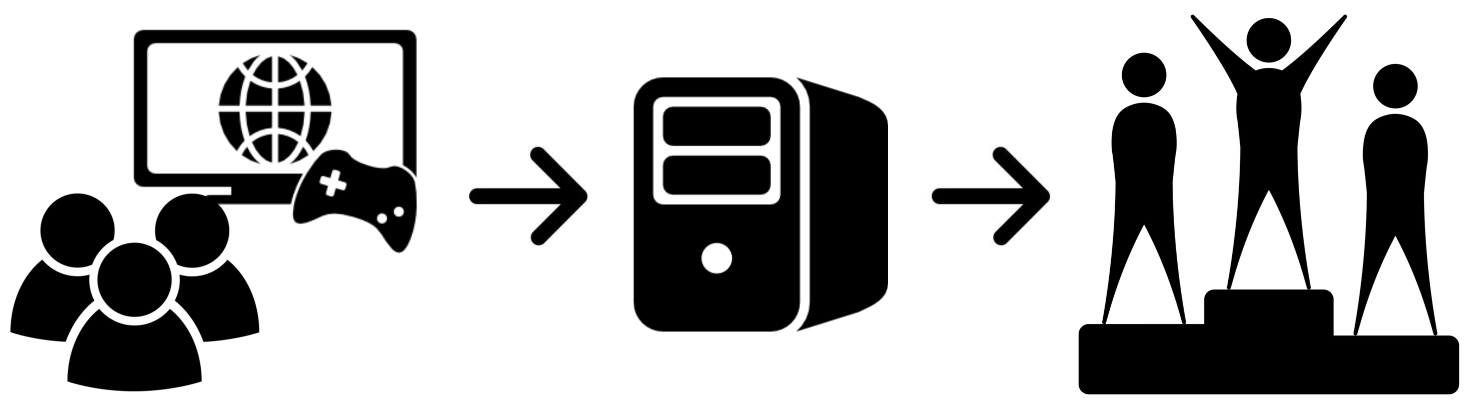
Player Behavior and Optimal Team Compositions in Online Games

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

Numerical Simulation

- **Target Game: League of Legends**
  - Multiplayer battle arena game with 27 million plays per day
  - Free online API to retrieve recorded game data
  - Official guide provides clustering information for players based on in-game character choices (e.g., character with good defense)
- **Data samples** Total of 120,000 training and 12,000 test samples
- **Implementation**
  - Clustering and classification algorithms in MATLAB 2014b
  - Data processing and feature selection in Python 2.7
- **Hardware** All simulations on 2.7 GHz Intel Core i7, 8 GB RAM

Baseline Outcome Predictor

- **Features** are team compositions for all teams in a game based on official guide's clustering information (characters are mapped to 1 of 5 play styles)
- **Logistic regression** with 10% hold-out cross validation
- **Poor accuracy** of 55.1% on training samples, 54.4% on test samples

Behavioral Clustering

- **Features** are normalized player statistics (damage dealt, money earned,...)
- Clustering algorithms (unsupervised learning)
  - **k-means** with 10-fold cross validation over parameter k gave 12 clusters
  - **DP-means** is a nonparametric expectation-maximization algorithm derived using a Dirichlet process mixture model (Kulis and Jordan, 2012)
  - Intuitively, a new cluster is formed whenever a point is sufficiently far away from all existing centroids, as determined by some threshold distance λ
  - We ran it with 10% hold-out cross validation with $\lambda = 3.3$, giving 8 clusters

given training set of size N, threshold distance λ
repeat
For $n = 1, \ldots, N$
1. Assign sample n to the closest cluster if the contribution to objective from the squared distance is at most λ^2
2. Otherwise, form a new cluster with just sample n

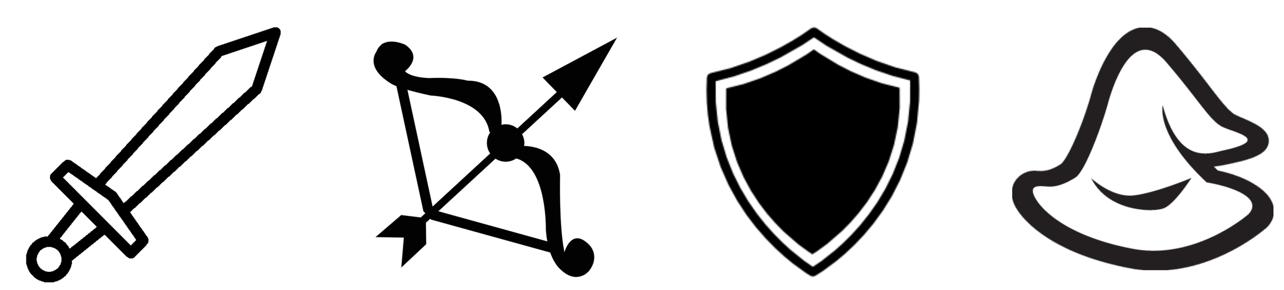
until clusters converge

Summary: Play style clustering algorithm results (10 trials)

	cross validation method	no. of clusters	cpu time (s)
k-means	k-fold (k = 10)	12	154.1
DP-means ($\lambda = 3.3$)	10% hold-out	8	65.4

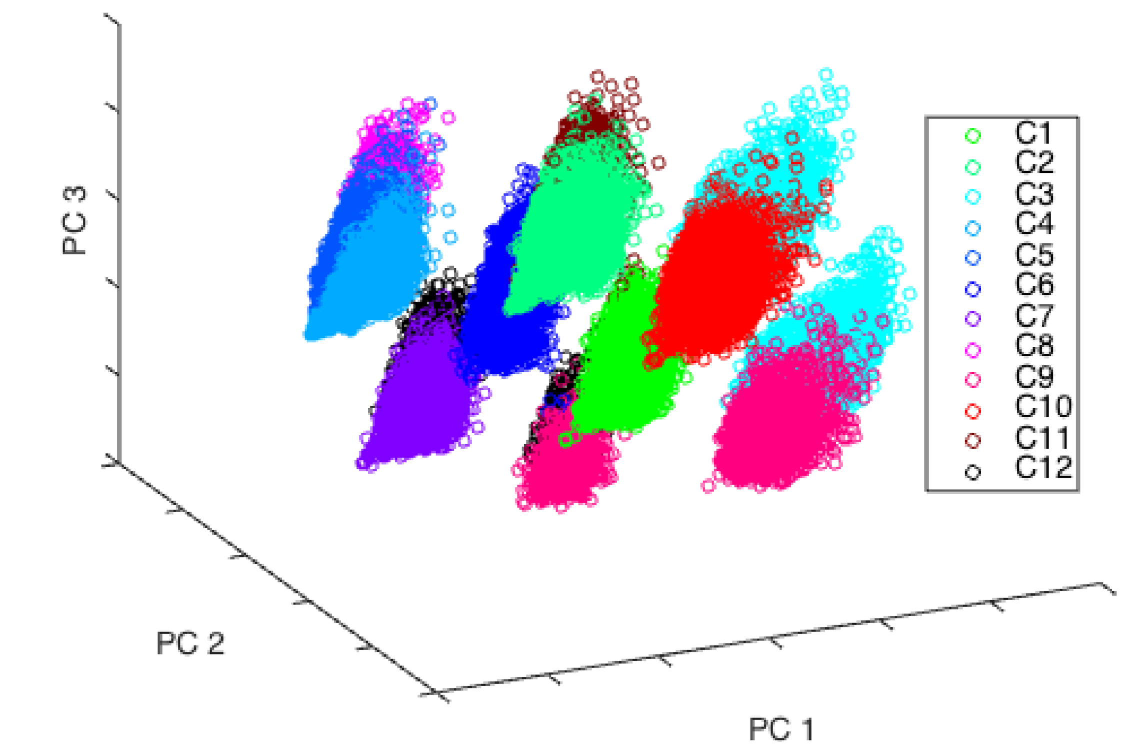
Cluster Intepretation

- Surprisingly accurate intuition behind groups (with input from expert players)
  - **Physical damage attacker**
    - Clusters 1, 7, and 9
    - Differ in risk attitudes
  - **Ambusher**
    - Clusters 3, 8, 11, and 12
    - Team oriented vs. lone wolf
    - Includes "hybrid" roles with other play style clusters
  - **Team support**
    - Cluster 5: Many assists in kills
  - **Magic attacker**
    - Clusters 6 and 10: Many kills
    - Ranged- vs. close-combat
  - **Miscellaneous**
    - Cluster 2: All-around average
    - Cluster 4: Novice player



Cluster Visualization

- **Principal component analysis** helps us visualize our dataset in 3D



Game Outcome Prediction

- **Features** are team compositions for all teams in a game based on behavioral groupings generated from k-means and DP-means clustering
- **Labels** are win/loss for each game (if team 1 *beats* team 2, label = 1)
- Classification algorithms (supervised learning)
  - **Logistic regression** generalized linear model with Bernoulli distribution
  - **Gaussian discriminant analysis** assuming data is Gaussian-distributed
  - **Support vector machine** assuming data is separable by soft-margins
  - **Cross validation** 10% hold-out for all methods
- **Best predictor** uses features based on k-means clustered team compositions for each game, trained on an SVM; 70.4% accuracy (vs. 55.1% baseline)

Summary: Outcome prediction algorithm results (10 trials)

	k-means (%)		DP-means (%)		cpu time (s)	
	train acc.	test acc.	train acc.	test acc.	k-means	DP-means
LR	72.25	68.75	69.67	67.11	7.4	7.1
GDA	74.79	70.14	70.88	68.39	7.7	7.1
SVM	74.75	70.39	71.71	69.21	91.2	41.6

Conclusion and Extensions

- k-means and DP-means provided clusters separated by highly intuitive play style differences, as confirmed by expert players
- LR, GDA, and SVM all provided better predictors with our new team composition features than baseline LR with official character classes
- More features, such as time-identified statistics (early vs. late game actions)

Acknowledgements

We thank Professor Andrew Ng and the instructor team.