Building Recommender Systems for Video Games on Steam

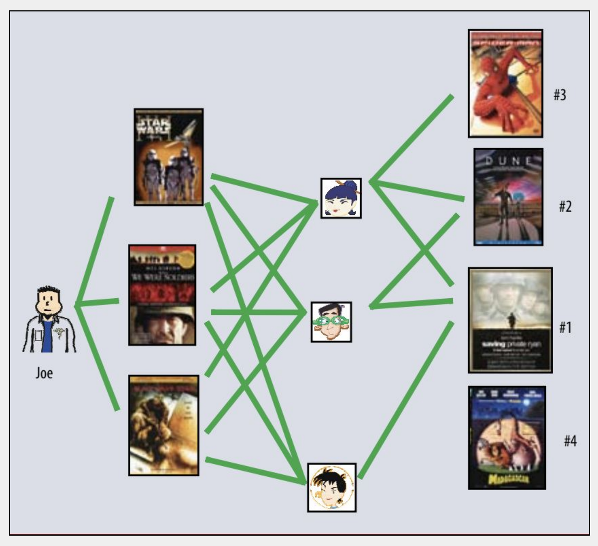
Team Members: Brian Choi, Dharmesh Panchal, Andrew Le, Misun Ryu
Advisors: Julian McAuley Ph.D, Ilkay Altintas Ph.D

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Introduction

Ubiquitous in e-commerce, entertainment, social media, and countless other domains, recommender systems have have allowed companies to target user specific tastes and help users decide what to buy. Collaborative filtering is a recommender system technique that models user preferences based on other user behavior and preferences. In contrast to content based filtering which uses item metadata to build a model, collaborative filtering relies purely on user-item interactions.

We construct a recommender system using game data on Steam that accurately captures user preferences. We then further analyze strengths and weaknesses of our model to build a more complete understanding of the data and the video game industry as a whole.



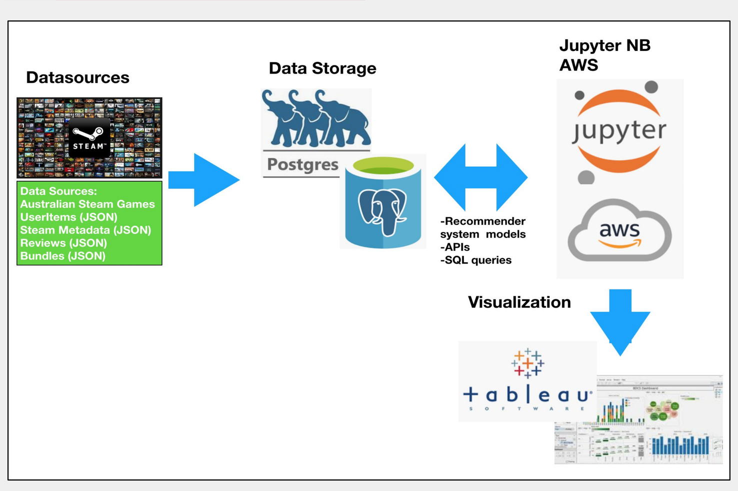
Primary Dataset and Data Preprocessing

Primary data set is user-item data containing list of users, the games they own, and how much time they have spent playing that game since bought. Additionally, we also files game metadata, bundle metadata, and user reviews. The game metadata contained information on a game's release date, average user sentiment, pricing, and genre, used in later analysis of results. The data set was static and collected from Steam's website using a web crawler..

Our main goal with data preprocessing is to generate an interaction matrix which is then used as input for our model. The matrix contains binary values indicating whether or not a user owns a game. The matrix was further filtered to remove of free games, remove of unplayed games for each user, and to only contain the top 1000 models.

item / user	Europa Universalis IV	Hurtworld	Interstellar Marines	ORION	Overlord: Raising Hell	Strike Suit Zero	Terraria
76561198040806617	0	0	0	0	0	0	0
76561198071059521	0	0	1	0	0	0	1
76561198077795250	0	0	0	0	0	0	0
76561198093085776	0	0	0	0	0	0	0
76561198096850767	0	0	0	1	0	0	1
Hueheueeheuee	0	0	0	0	0	0	0
Vladimirputinisgreatman	0	1	0	0	0	0	1
pinkie10	0	0	0	0	0	0	1
sonnymack	0	0	0	0	0	0	1
urami	0	0	0	0	0	0	0

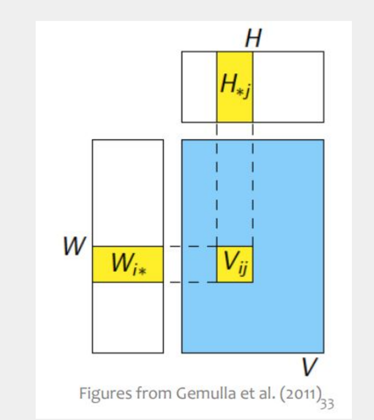
Data Pipeline



	Environment	Process Data	Derive Models	Analyze Results	Visualization
Datasource					
Python	Jupyter notebook version 3.6 installed on EC2 m5ad.12xlarge used to import/parse JSON data.	Create user-item matrix. Pre-process data joining user-item data to game metadata	Find optimal hyperparameter using 5-fold cross validation. Train final model.	Perform post-training analysis of model on: -Cold start problem -Games bought vs played -BPR vs WARP -Hybrid Model	
PostgreSQL	PostgreSQL v11.0 installed on EC2 m5ad.12xlarge used to import data and store temporary tables.		Save performance of hyperparameter sets using		

Modeling

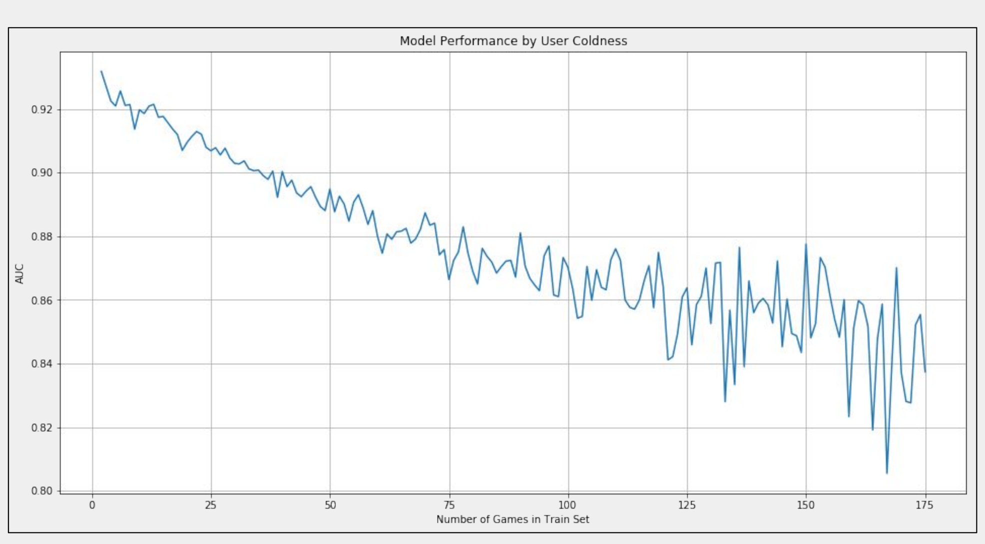
We used Bayesian Personalized Ranking (BPR) as the learning algorithm for our model. BPR was developed as collaborative filtering model to solve one class recommendation problem where user-item interaction values are 0 or 1. BPR generates scores for each user-item pair which can then be sorted and ranked to generate recommendations. Latent factor embeddings are found for each user and item using gradient descent and a user and item's embeddings multiplied to find the score.



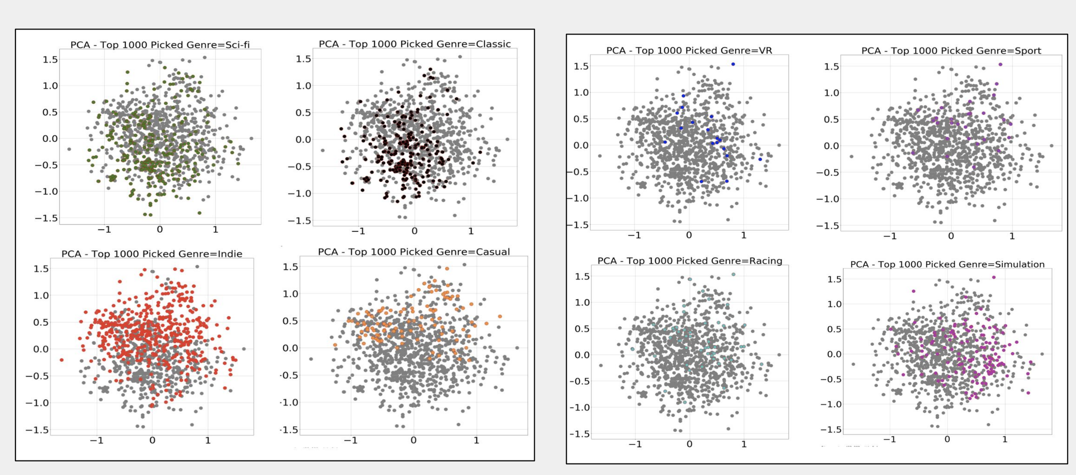
AUC is used as primary metric to evaluate model. During testing, AUC measures the probability that a positive example (user owns game) ranks higher than a negative example (user does not own game).

$$AUC = \frac{1}{|\mathcal{U}|}\sum_u \frac{1}{|E(u)|}\sum_{(i,j)\in E(u)} \delta(\widehat{x}_{u,i} > \widehat{x}_{u,j})$$

Key Findings and Results



- Cold start problem is a common issue facing collaborative filtering models.
- New ("cold") users generally harder to model because fewer interactions exist to train.
- Found counterintuitive result where cold users are outperforming hot users when comparing AUC of test sets.
- Analyzing further, found that cold users generally buy the most popular games on Steam. Model performs extremely strongly for popular games.
- Once taste/genre specialization occurs and most popular games are owned, model begins facing more difficulty.



- Using PCA, found BPR item embeddings can represent genres, confirming user taste correlated with specific genres.
- Plotting genres on top two principal components show certain genres occupy specific regions, showing clustering property.

		Test Set			
		No Filter	Only games played > 0 minutes	Only games played > 30 minutes	Only games played > 10th percentile/game
Model	No Filter	0.9151	0.9076	0.9106	0.9061
	Only games played > 0 minutes	0.881	0.9102	0.9179	0.9084
	Only games played > 30 minutes	0.8623	0.9015	0.9168	0.8995
	Only games played > 10th percentile/game	0.8812	0.9105	0.9185	0.9089

- With AUC given in table above, we trained four models using four different user-item data set.
- Filtering out games that users bought but did not play gave better recommendations for games that users will play.
- However, filtering data decreases size of data set and while trains stronger on own data set, weakens performance on recommendations for pure purchasing. No filtraton was used beyond free games for final output based on results here.

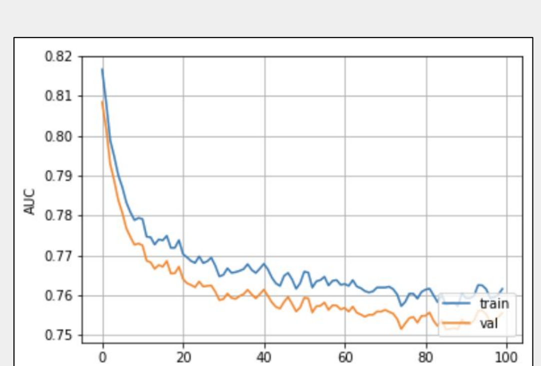


Figure 1: Train vs Validation AUC for BPR

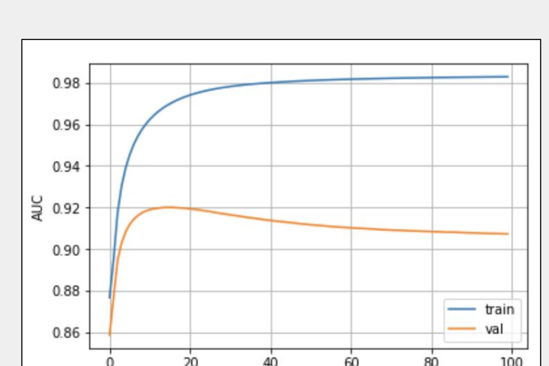
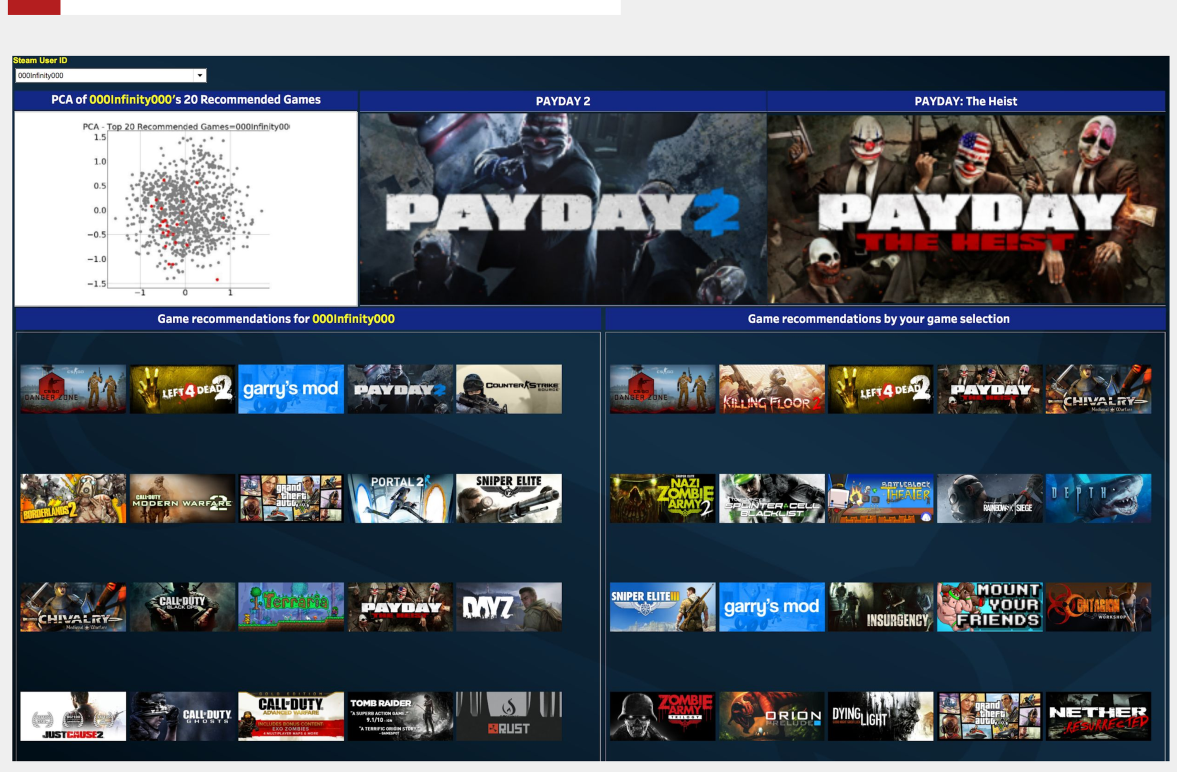


Figure 2: Train vs Validation AUC for WARP

- WARP is an alternative to BPR loss function where embeddings are only updated when a violation is found during training.
- The WARP loss function outperforms the classic BPR loss.
- BPR is prone to overfitting with more epochs while WARP maintains a steadier AUC with more epochs.
- We ran our final model using 15 epochs using these plots as reference.

Dashboard



We built a dashboard using Tableau to display the results of our model. We drew inspiration from similar recommender systems found on Netflix and Amazon. Given user input, our dashboard shows:

- Top 20 recommended games
- Similar games to one selected created
- PCA scatter plot showing how similar games in user's library are to one another. Shows if user has diverse or selective tastes.

Conclusion and Acknowledgements

Conclusions:

- Video games are less susceptible to cold start problem compared to other data sets normally used for recommender systems. Because of nature of video game industry, cold users can sometimes be given better recommendations with their smaller library.
- PCA is a powerful way to visualize natural grouping among BPR embeddings. Contrived groupings such as genre can be coincide with natural groupings based on collaborative filtering.
- Data filtration can sometimes improve performance of model. BPR and WARP perform well even after filtering data. However, more data is still king.
- Future work include obtaining more data and seeing how model performs across entire Steam system (millions of users).

We would like to thank our advisors, Professors Julian McAuley and Ilkay Altintas for the help they provided over the last six months in helping us complete our project. We would also like to thank our classmates, staff, and professors who have supported us over the last two years in the DSE MAS program.