
Recommending Games, Communities and Estimating Gameplay time in Gaming Social Network

A case study of Steam

Pratik Anand

Sanket Lokegaonkar

Outline

- Introduction
 - Big Picture!
 - Gaming Social Networks
 - Analyzing Steam
 - Standard Matrix Factorization
 - Co-occurrence Based Factorization
 - Proposed Method
 - Empirical Analysis
 - Challenges We faced
 - Conclusion & Future Work
-

Big Picture

Given:

Tripartite graph of users, games , communities (Steam)

Our Goal:

How do we learn a model which can predict games and communities while staying consistent to each other?

Why is it important:

Few datasets with such a unique-structure (Tripartiteness)

Large real-world dataset unexplored (Sounds like fun!)

Our Technical Framework:

Matrix Factorization shown to work well empirically.

How can we extend matrix factorization methods to this framework?

How can we do better than just applying Matrix Factorization?

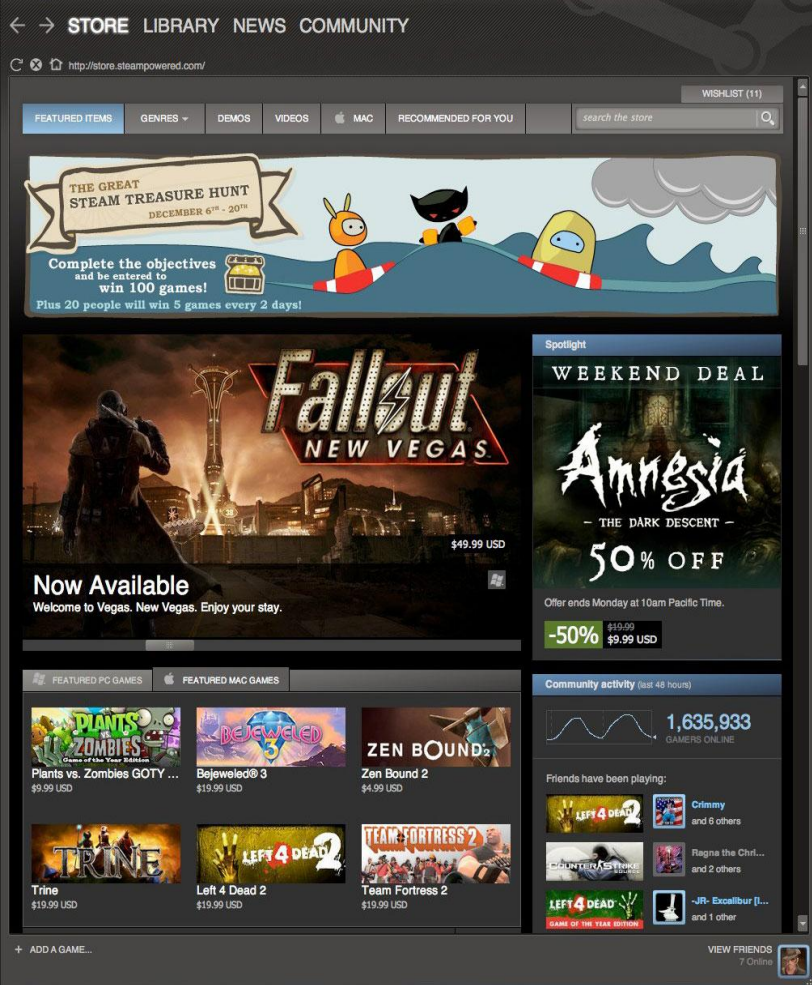
Gaming Social Network

- Not as well studied as other social network like Twitter or Facebook
- Unique in representation as centred around video games
- Different kind of relations like friends, followers, fans, publishers, game recommenders etc
- Different kind of gaming networks :



XBOX
LIVE

Humble Bundle



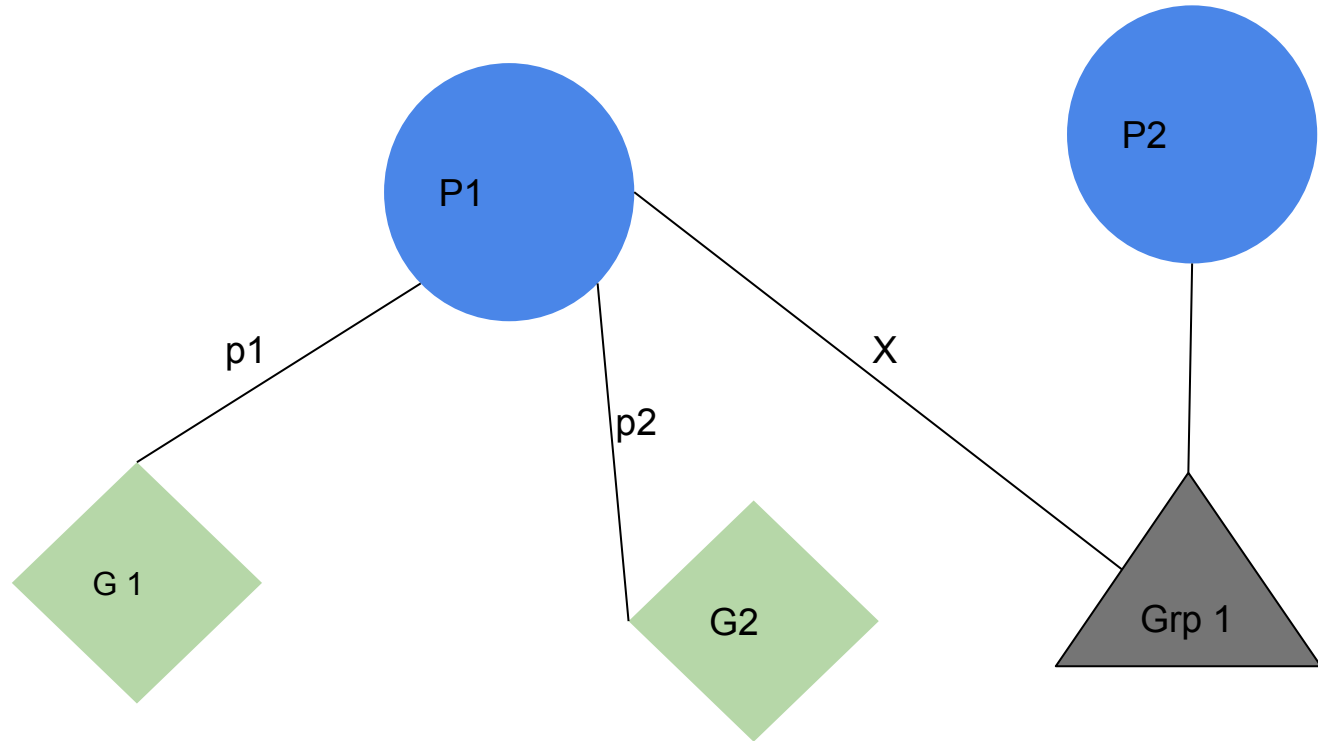
Steam

- Largest online gaming marketplace and social network
- 108.7 million user accounts and 384.3 million owned games
- Different social features :
 - Friendships
 - Communities
 - Clans
 - Steam Workshop
- Tracks user playtime session

Dataset Sampling

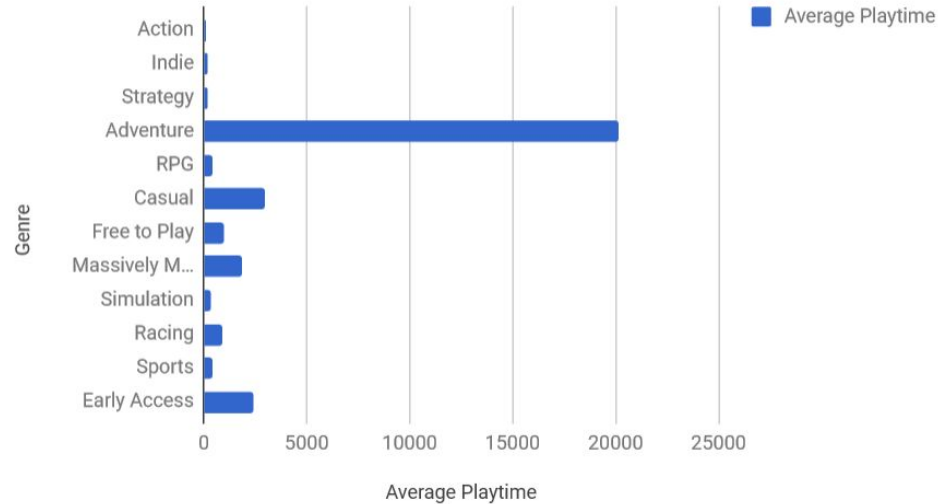
- 170GB Steam dataset compiled by <http://steam.internet.byu.edu/>
(Condensing Steam: Distilling the Diversity of Gamer Behavior)
 - 11 tables with user data retrieved in period of 2 weeks
 - Data consisted of friendship, game ownership, playtime as well as community membership
 - Gameplay time recorded over the course of two weeks
 - Sampled two data sets : 10,000 users and 100,000 users
-

Modelling the relations

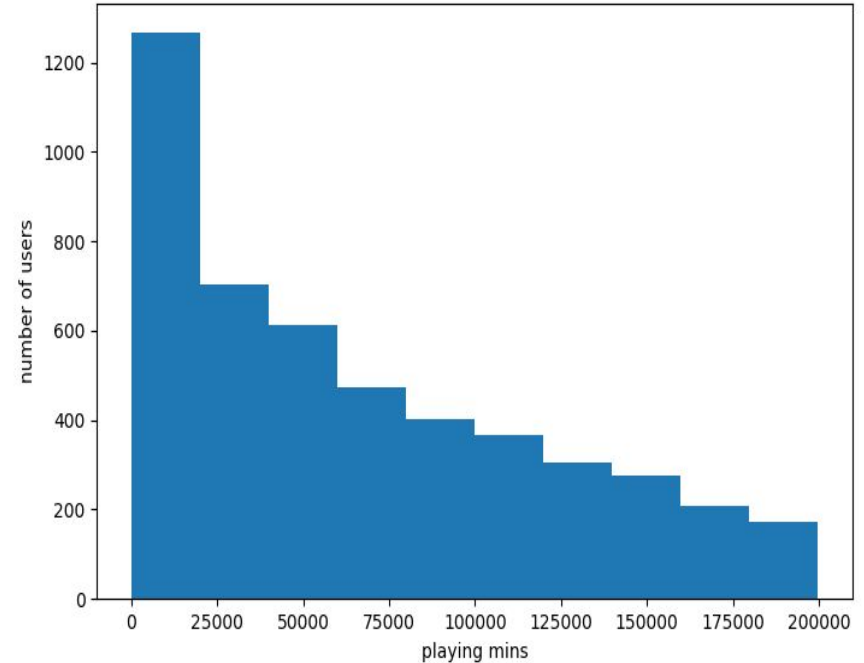


Analyzing Steam

Average Playtime vs. Genre



Users



Standard Matrix Factorization

- Given Users U and Items I and User-Item Interaction Matrix
- Find latent features for U and I which reconstruct the matrix
- Solve Least Squares Optimization

Joint Matrix Factorization

- Given 3 interacting entities: Users U , Items I and Communities C
 - Interaction Matrices:
 - User-Community Matrix
 - User-Game Matrix
 - User-User Matrix*
 - Alternating Least-Squares Optimization
 - Solving jointly, might have a synergistic effect improving the performance on both,
-

Co-occurrence Based Matrix Factorization

- Proposed by Liang et.al RecSys'16
- Given Users U and Items I
- Does not utilize core-property of similar items are bought together
- Construct a Co-occurrence Matrix(similar to word2vec)

$$PMI(i, j) = \log \frac{P(i, j)}{P(i)P(j)}$$

- Perform Joint Optimization which conforms with the co-occurrence matrix
 - Alternating Least -Squares based Optimization
 - Intuitively , can have a synergistic effect improving the performance on both,
-

JFactor Matrix Factorization (Proposed Method)

Math: (Joint Optimization Part)

$$L_{mf} = \sum_{u,i} (x_{ui} - \theta_u^T \beta_i)^2 + \sum_{u,j} (x_{uj} - \theta_u^T \alpha_j)^2 \\ + \sum_{u1,u2} (x_{u1,u2} - \theta_{u1}^T \theta_{u2})^2 + \sum_u \|\theta_u\|^2 + \sum_i \|\beta_i\|^2 + \sum_j \|\alpha_j\|^2$$

$X \in R^{U \times G}$: Sparse User-Game Interaction Matrix from
U Users and G Games

$Y \in R^{U \times C}$: Sparse User-Community Interaction Ma-
trix from U Users and C Communities

$Z \in R^{U \times U}$: Sparse User-User Interaction Matrix from
U Users

θ = User Latent Features

β = Game Latent Features

α = Community Latent Features

JFactor : Combined Optimization Model

$$L_{modified} = L_{mf} + \sum_{ij} (U_{ij} - \beta_i^T p_i) + \sum_{ij} (W_{ij} - \alpha_i^T q_i) + \sum_i \|p_i\|^2 + \sum_j \|q_j\|^2$$

In a gist, Joint Factor + Co-occurrence = JFactor
Implemented in TensorLab

Notation:

- L_{mf} = Joint Matrix Factorization Objective ;
 - U_{ij} = Co-occurrence Matrix for Games ;
 - W_{ij} = Co-occurrence Matrix for Communities
 - p_i = Context features of Users
 - q_j = Latent features of Communities
-

Empirical Analysis: Evaluation Metrics

Absolute Error:

- Mean Square Error:

Ranking Metrics:

- Recall @ M
 - NDCG @ M
 - MAP @ M
-

Empirical Analysis: Ablation Study

Settings we compare:

- Trained on Games Only (G)
 - Trained on Communities Only (C)
 - Trained on Games + Games Co-occurrence Matrix (G + Embedding G)
 - Trained on Community + Community Co-occurrence Matrix (C + Embedding C)
 - Games + Community (Joint Factorization)
 - Games + Community + Community Co-occurrence Matrix + Game Co-Occurrence Matrix (All Combined)
-

Empirical Analysis: Ablation Study

10K Users , 1.9K Communities, 3.1K Games

	User-Game Matrix					User-Community Matrix				
	Recall@20	Recall@50	NDCG@100	MAP@100	MSE	Recall@20	Recall@50	NDCG@100	MAP@100	MSE
Games Only Factorization	0.242547	0.322987	1.675945	0.117363	0.4953	-	-	-	-	-
Communities Only Factorization	-	-	-	-	-	0.151137	0.186248	0.092616	0.055346	0.95
G+C (Joint Factorization)	0.190743	0.268247	1.300759	0.085098	0.5153	0.009453	0.021832	0.012558	0.002898	1.622
Games + Game co-occurrence Factorization	0.254986	0.342827	1.747526	0.123401	0.4937	-	-	-	-	-
Community + Community co-occurrence Factorization	-	-	-	-	-	0.007765	0.023858	0.010219	0.002573	2.6075
(Joint Factorization + Co-occurrence)	0.237249	0.313843	1.674253	0.117684	0.4744	0.069097	0.127842	0.054317	0.023398	0.5032

Empirical Analysis: Ablation Study

100K Users , 15K Communities, 3.9K Games

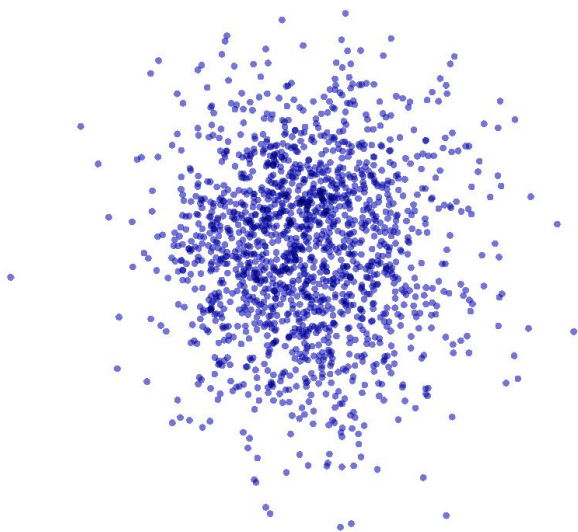
	User-Game Matrix					User-Community Matrix				
	Recall@20	Recall@50	NDCG@100	MAP@100	MSE	Recall@20	Recall@50	NDCG@100	MAP@100	MSE
Games Only Factorization	0.047	0.129	0.5624	0.012	0.3374	-	-	-		
Communities Only Factorization	-	-	-	-		0.0108	0.015	0.005	0.002612	2.2618
G+C (Joint Factorization)	0.009180	0.033939	0.205335	0.002952	0.3461	0.000444	0.001710	0.000701	N/A	3.8844
Games + Game co-occurrence Factorization	0.229253	0.349671	1.511421	0.087371	0.333					
Community + Community co-occurrence Factorization						0.000804	0.002311	0.000905	0.000174	4.49
(Joint Factorization + Co-occurrence)	0.000924	0.012138	0.149904	0.001536	0.3615	0.000499	0.001793	0.000683	0.000126	4.30

Empirical Study: Qualitative



- Random Sampling:
 - Games with similar genres are clustered together -
 - Users with high scores are clustered together
 - Users who hoard/or are addicted clustered together.
- Visualizing the latent features(PCA)
 - PCA- Games

Empirical Study: Qualitative



- Random Sampling:
 - Games with similar genres are clustered together -
 - Users with high scores are clustered together
 - Users who hoard/or are addicted clustered together.
- Visualizing the latent features(PCA)
 - PCA- Communities

Empirical Study: Qualitative



- Random Sampling:
 - Games with similar genres are clustered together -
 - Users with high scores are clustered together
 - Users who hoard/or are addicted clustered together.
- Visualizing the latent features(PCA)
 - PCA- Users

Critical Analysis: What went wrong?

- No strong synergistic signal between User-Community and User-Games
 - Sparsity of User-Community
 - Joint optimization ?
 - Over-constrained System
 - Longer training times?
-

Challenges We faced

- Lots of changes to the core idea:
 - Initially , planned to do tensor factorization-based approach
 - Lots of changes towards the end
 - Did not explore details on why the results are weird
 - Tried integrating User-User Friendship Graph.
 - Got worse performance.
 - Tried our own implementation in Python.
 - Had bugs. Was not working as expected.
-

Conclusion & Future Work

- We developed a model for jointly recommending communities and games together.
 - First to explore this unique dataset from the recommendation perspective
 - Future Work:
 - Find a way to avoid over-constraining
 - Model sensitive to initialization. Further investigation needed
 - Incorporating User-Friendship Graph
-

Acknowledgements

Prof. Aditya Prakash

Thanks!
Questions?
