(CS 6604) Recommending Games, Communities, Estimating Gameplay Time in Gaming Social Network: A Case Study of Steam

Sanket Lokegaonkar
Virginia Tech
sloke@vt.edu

Pratik Anand
Virginia Tech
pratik@vt.edu

Abstract

Gaming communities like Steam are one of the less-studied social networks. We intend to study this network from the point of relationship between friendship, communities and game ownership having an influence on addiction of gaming and ability to make predictions of game ownership and playtime, leading to prediction of addiction.

In this project, We propose JFactor, a joint matrix factorization approach, which predicts game-play times and recommended communities for users jointly for the Steam network. We optimize our method by coupling the factorization from the derived community and game co-occurrence matrix.

We perform extensive experiments to show that our joint matrix factorization model performs comparable to individually learnt matrix factorization models with game and community matrix.

1 Introduction

Steam is a popular social networking site and online webstore comprising of around 108.7 million active user accounts and 384.3 million games. Steam provides typical social networking features like profile pages, friends, groups, instant messaging, voice chat, and news feeds. In addition to that, it focuses on video game based interactions also like screenshots and video sharing, game achievement showcases, leaderboard statistics, and more. Because of its highly active usage worldwide, We also see highly active big and small communities among its users depending on the games they play, how frequently they play and which game they buy, and also geographical proximity.

Gaming networks, especially Steam network, are less studied than social media networks like Facebook and Twitter. They do share common features but there are certain aspects to Steam network which make it different than others (example : close interactions between non-friends users on ad-hoc basis in multi-player gaming). Thus, there is a marketing as well as social need of understanding this network. Estimating gaming playtime and game ownership can lead to applications like identifying key targets for viral marketing and early identification of addiction based on social aspects of gamers so that corrective measures can be taken. Recommending similar communities given the games user plays is an important application which can foster richer interactions and community growth.

In this project, we explore the recently released Steam Dataset from the point of view of recommending play-time(Interest) and predicting relevant communities. We propose a joint matrix factorization model (JFactor) for the dataset. We further explore two potential optimizations in the factorization space (Joint Optimization and Co-embedding) [3].

With this factorization framework, we are able to estimate game-play time. This has applications in finding addiction based on thresholds. This joint framework can be used to identify which users are likely to be heavy or addicted users based on the communities, friendships and the subset of the games they play, which users in this community are likely to play this game.

2 Formal Problem Definition

Our goal in this paper is to model a specific-types of social networks with the user products and community structure. Specifically in the context of Steam gaming network, we assume that the we have three entities, U (total users), C (total Communities), and G(total Games).

We model the gaming network as a tripartite network with 3 entities U (total users) , C (total Communities) , and G(total Games). There exists an edge between a user and community entity if user u_i is part of community(c_j). There exists an edge between user u_i and game g_j, if the user has bought this game. The edge between user u_i and the game is weighted and the weights models the hours played. The community membership edge is an unweighted graph.

Given this problem formulation, our objective in this paper is to develop a model which can jointly predict the missing edges between users and communities and predict the weighted missing edges between games and users (indicating interest and hours played.). We also interested in knowing latent connections between the communities and games as well.

3 Related Work

As far as we know, we are the first to consider the problem of recommending games and estimating gameplay time (essential for finding gaming addiction) from gaming social

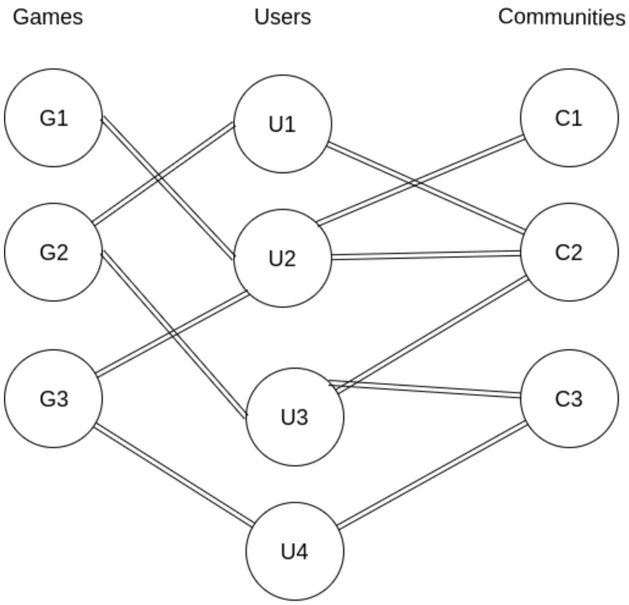

Figure 1. Illustration of tripartite graph of users, games and communities from Steam network

network perspective. Our method falls into the category of link prediction in graphs and product recommendation.
Product/Rating Recommendation for Bipartite graphs(users and products) have been very well investigated problem over the years. In the recent works, many approaches have started to take into account the social aspect of the problem (user-user connections for recommendation.) We follow the similar intuition of importance of social networks/graphs for the recommendation. But unlike the previous work, we use the ground-truth community-user bipartite graph as basis of social structure.
Link-prediction based approaches have either taken a matrix factorization based approaches or local feature-based link prediction approaches. De et.al proposed a classification -based approaches using local features like and global features perspective to the link prediction problem.The local features model the neighborhood (random walks, local similarity), the global features extraction is the co-clustering ie. (user-user collaborative filtering). SVM-based classifier is learnt to predict an edge between two nodes.

Our method uses the proposed idea of Liang et.al of using co-factorization model as an optimization for our joint matrix factorization. Liang et.al proposed co-factorization model which considers the item co-occurence matrix in its objective function.

Our method is similar to approaches proposed by Karatzoglou et.al[1] and Zhang et.al [5] . Kartazoglou et.al proposes a novel SGD-based formulation for learning link prediction parameters i.e the decomposition matrices. Training the prediction requires linear number of updates, in contrast to the previous approaches which require batch steps. Zhang et.al[5] proposes a recommendation algoirthm which has been applied to tripartite setting of recommending items based on the tags added by the user. Our work differs from this approach by extending the tripartite framework to predicting real-value in the tensors rather than binary values in the tensor.
In this paper, we focus exclusively on the STEAM network graph as our dataset. [4]. Our primary reason for this choice is the dataset is unique and has not been previously studied of recommendation or game-play estimation before.

4 Background

Steam Steam is an web based client for Windows, Mac and Linux which is used by large as well as small publishers to sell their games to the public. Its total user base is more than 108 million active users [4].
The social features of steam involve :

- **Friendship** Any steam user can send friend request to other users, much like any other social network.
- **Clans** Many users play online multiplayer games as part of teams called clans. The clan relationship is more ad-hoc than traditional friendship in Steam.
- **Community** Steam communities act as a hub for social interaction related to to a particular topic or game. It encompasses discussions, user-generated content (artwork, game modifications etc.) for the game and special gaming events.

Addiction in gaming Addiction in gaming can have a broad definition with multiple aspects [2]. Due to the limited nature of data, we are defining it as simply the amount of time spend in playing video games. We are also including hoarding as another aspect which is roughly the number of games owned vs played on Steam. Steam dataset shows the amount of time user has played a game as well overall game play time, the number of games owned and the daily gaming activity in the two weeks when the data was collected in [4].

5 Methodology

Data collection We are using the Steam dataset from *steam. internet. byu. edu* [4]. It is a 170 GB SQL dump with 11 tables covering the number of users, number of games with genres owned, community membership and friends network. Since, re-creating a SQL database with such a large data is difficult task without dedicated hardware, we decided to work with a sample of the data. A SQL dump is a collection of SQL queries. We split the large 170 GB file into 300 small files, preserving the SQL statement structure. A quick search revealed the files included the SQL queries which the 11 tables. Using this information. We sampled two different datasets using BFS sampling, one of size 10,000 users and another of size 100,000 users. We also gathered the games owned by these users as well as communities they are part of, for completing our data sample.

6 Proposed Model

In this section, we formally define our matrix factorization model for jointly predicting game-play time and recommending communities and recommending friends.

Data Description:
We first describe the STEAM-INTERACTION graph as tripartite undirected graph of users, games and communities. As show in the figure, each edge weight between a user and game node represents the logarithm of game play-time of the user for the specific game. The edge weight between a user and community node c indicates participation of the user in that specific community. We define STEAM-FRIENDSHIP Graph as undirected graph between the users. An edge exists between two users in the graph if they are connected via 'FRIEND' relation.

Matrix Descriptions:
We represent the two graphs with 3 matrices:

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

JFactor: Matrix Factorization
MF is standard in recommendation/link-prediction based approaches. Formally, we define the typical matrix factorization problem for single matrix in this case X as follow:

We say that e.g $X \in R^{U \times G}$: Sparse User-Game Interaction Matrix decomposes into the product of user and game latent factors denoted by $\Theta_u \in R^K$ $(u = 1, ...U)$ and $\beta_g \in R^K (g = 1, ...G)$ respectively. The optimization objective for the matrix decomposition can be defined as :

$$L_{mf} = \sum_{u,i}(x_{ui} - \theta_u^T \beta_i)^2 + \sum_u ||\theta_u||^2 + \sum_i ||\theta_i||^2 \qquad (1)$$

Additional regularization L1-L2 regularization is usually added to avoid model overfitting. The (global) optimum the function can be shown to be equivalent to the maximum a posteriori estimate of the probabilistic Gaussian matrix factorization model.

Similarly, we can individually solve the optimization objective for the sparse matrices Y and Z to learn the latent features for users , games and communities.

Coupling
Individual matrix factorization does not take advantage of the coupling/consistency that exists between these matrices. We propose a modification to the above optimization function, which tries to take advantage.
We assume that the latent vectors of U (defined by β_u) , latent vectors of G (defined by θ_i) and latent vectors of C (defined by α_j) compose together to form the interaction matrices : X, Y and Z. We formulate an optimization objective which combines the latent features and matrices in a joint framework as follows:

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

Coupling latent features together and learning them jointly provides additional regularization and allows us to enforce the consistency among the latent features with respect to multiple interaction and networks.

7 Additional Consistency via Embedding Loss:

Embedding:
Recently proposed embedding models like word2vec have seen excellent success in natural language tasks. The core idea of the word2vec is to learn to predict the context (surrounding words) given the current words using low-dimensional embedding space. We build on the idea proposed by Liang et.al [3] of extending the embedding objective to the implicit factorization of graphs. Liang et.al and Levy and Goldberg et.al show the equivalence between factorizing Point-wise mutual information matrix shifted by log k and the skip-gram word2vec trained with negative sampling. PMI between two items in a factorization matrix is defined as

$$PMI(i, j) = \log \frac{P(i, j)}{P(i)P(j)} \qquad (3)$$

.

JFactor with co-embedding:
We define a co-occurence matrix (Point-wise Mutual Information Matrix) U and W for the Games g and Communities U. Games Co-occurence matrix $U \in R^{G \times G}$ can be factorized into latent features of the Games g (β) and context features of the matrix ($p_i \in R^K$). Similarly,Communties co-occurence matrix $W \in R^{C \times C}$ can be factorized into latent features of the Communities c(α) and the context features of the matrix ($q_i \in R^K$).
We can modify our optimization objective to take into account the item co-occurence patterns. Our final objective function is defined as follows:

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

We also add regularization loss on the context features for the games and community co-occurence matrices.

8 Inference :

The final objective function can be minimized in multiple ways. Batch Co-ordinate descent based approaches are commonly used methods in Factorization history. In batch coordinate descent, one matrix parameters is optimized while the others are kept fixed and then the matrices are alternated. This style of methods provide fast convergence but require to be run in batch setting.

In this work, we chose to use Limited Memory-BFGS with dogleg trust region as our optimization routine to learn the parameters θ , β , α , p and q for our optimization objective. BFGS is a quasi-Newton optimization method and has near linear complexity with the number of examples with rank-two updates to the Hessian. It has also been known for faster convergence in contrast to the Conjugate Gradient Descent Methods.

9 Empirical Study

In this section, We study the performance of our model (JFactor) qualitatively and quantitatively. We provide insights/intuition on the clustering capabilities by random sampling. We highlight the following results.

- On the steam graph, Our combined model (including the co-embedding optimization and Joint factorization) does not achieve the optimal performance. We hypothesize this due to the difficulty in optimization and overly-constrained system.
- In quantitative analysis, Co-embedding optimization individually shows positive effect and improves the performance of the game. Joint optimization(Coupling) individually performs close to that of the individual matrix factorization models for User-Game and User-Community.
- We explore the resultant embeddings learnt by our model qualitatively for correlation between the addiction and genres. We find the latent factors does preserve the information

9.1 Evaluation Metrics

We follow the evaluation protocol of Liang et.al[3] and follow ranking-based metrics. We report the following metrics:

- Recall@M
- Truncated normalized discounted cumulative gain (NDCG@M)
- Mean Average Precision@M

For each user, all the metrics performs compare the predicted rank of (unobserved) items with their true rank. While Recall@M considers all the items with first rank M, NDCG@M and MAP@M use a monotonically increasing discount to emphasize the importance of higher ranks versus lower ones. CoFactor predicts ranking π for each user by sorting the predicted preference $\theta_u^T \beta_i$ for $i = 1, \ldots I$.

Recall@M for user u is :

$$Recall@M(u, \pi) = \sum_{i=1}^{M} \frac{\mathbf{1}u(\pi(i)) = 1)}{min(M, \sum_i^1, \mathbf{1}u(\pi(i')) = 1))}$$

The expression in the denominator evaluates to the minimum between M and the number of items consumed by user u. This normalizes Recall@M to have a maximum of 1, which corresponds to ranking all relevant items in the top M positions.

DCG@M for user u is

$$DCG@M(u, \pi) = \sum_{i=1}^{M} \frac{2^{\mathbf{1}\{u(\pi(i))=1\}} - 1}{\log(i + 1)}$$

NDCG@M is the DCG@M normalized to [0,1] where one signifies perfect ranking.

Mean average precision (Map@M) calculates the mean of users' average precision (AP). The average precision AP@M for a user u is

$$AP@M(u, \pi) = \sum_{i=1}^{M} \frac{Precision@Mi(u, \pi)}{min(M, \sum_i^1, \mathbf{1}u(\pi(i')) = 1))}$$

[3]

9.2 Ablation Study

We compare an ablation study comparing different optimizations proposed in our model against each other (see Table 1). We report the ranking metrics and perform evaluations on two datasets: BFS Sampled data of 10000 users , 3310 Games and 1618 Communities/Groups. We compare the following settings:

- 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)

We see that Joint Optimization achieves similar performance as that of individually trained on the User-Game Matrix and User-Community Matrix. We cam also see that the co-embedding increases performance for the Game Matrix and decreases performance for the Community Matrix. The Combined objective function achieves moderate performance on all metrics but never outperforms the R+C matrix. We believe the potential reason for this is overly-constraining the problem.

9.3 Qualitative Analysis on multiple samples of 10k users

The playtime (in mins) per user as well as playtime (in mins) per game show the power law 2 3 i.e. a small number of users are playing a small number of games for much longer

Table 1. Ablation Study comparing effects of different optimizations on the 10K BFS-Sampled dataset. We perform evaluation on 10k datast

	User-Game Matrix				User-Community Matrix			
	Recall@20	Recall@50	NDCG@100	MAP@100	Recall@20	Recall@50	NDCG@100	MAP@100
Games Only Factorization	0.23	0.313	1.634	0.113	-	-	-	-
Communities Only Factorization	-	-	-	-	0.0121	0.018	0.0104	0.004
G+C (Joint Factorization)	0.237249	0.313843	1.674253	0.117684	0.012154	0.018569	0.010437	0.004059
Games + Game co-occurrence Factorization	0.375371	0.492089	2.482255	0.190134				
Community + Community co-occurrence Factorization					0.007765	0.023858	0.010219	0.002573
(Joint Factorization + Co-occurence)	0.190743	0.268247	1.300759	0.085098	0.009453	0.021832	0.012558	0.002898

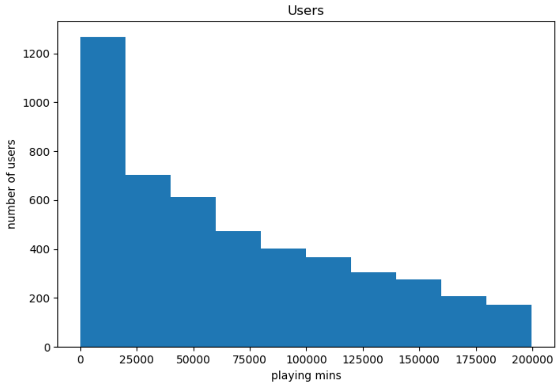

Figure 2. Number of users vs playtime in minutes.

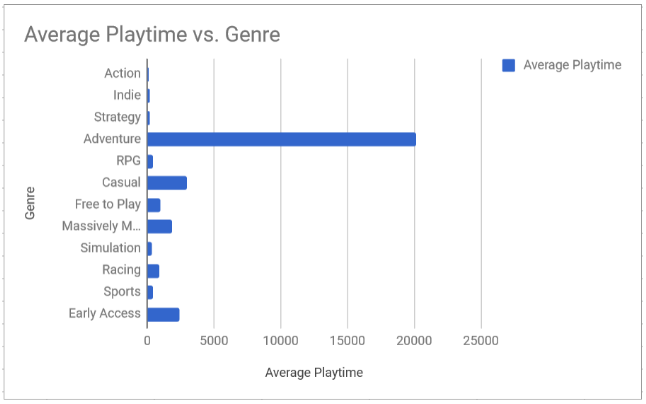

Figure 3. Number of games vs playtime in minutes

period. The playtime of the games for user follow a power law distribution. Given that, we can assume that the users on the higher end with significant (more) playtime are most likely addicted.

Nearest Neighbor sampling of Addicted Users: For qualitative analysis of the User Latent Space, we choose 20 random samples of the addicted users having playtime of more than 2000 minutes. We then find the 10 nearest neighbors in the latent space β. We find that around 3-4 data points of addicted users among the random sampled points.

Nearest Neighbor sampling of Games: For qualitative analysis of the Games Latent Space, we choose random samples of the games. We then find the 5 nearest neighbors games in the latent space α. We see 3-4 data points of games which share the same genre.

10 Future Work and Conclusion

In this paper, We proposed JFactor a joint matrix factorization approach which jointly predicts the game-play times and recommends communities for Steam Gaming Network. We evaluated the proposed approach and found interesting findings.

- We find that our implementation of Joint Factorization + Co-occurrence based embedding objective does not perform optimally and at times worse than the our Joint Factorization only. We further found that this analysis was extremely sensitive to initialization. We hypothesize this maybe due to overly constraining the parameter space.
- The features of the users behavior are well preserved in the latent space obtained by Factorization

Future work in this direction could be to learning a way to model the co-occurrence matrix without overly constraining the matrix. We also tried to include the Friendship Graph information within our objective function, but found it significantly dropped the performance. Future work, can also explore a potential approach to incorporating friendship graph knowledge.

References

[1] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 79–86.

[2] D. J. Kuss and M. D. Griffiths. 2012. Internet gaming addiction: A systematic review of empirical research. *International Journal of Mental Health and Addiction* 1, 2 (2012), 278–296.

[3] Charlin Liang, Altosaar and Blei. 2016. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-occurrence. *ACM* 389, 1 (2016), 179–186.

[4] Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. 2016. Condensing Steam: Distilling the Diversity of Gamer Behavior. In *Proceedings of the 2016 ACM on Internet Measurement Conference*. ACM, 81–95.

[5] Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. 2010. Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications* 389, 1 (2010), 179–186.