Show me your Friends and I'll tell you who you are

Exploiting Social Networks in Massively Multiplayer Online Role-Playing Games

Abhiram Ravi, Vaishnavh Nagarajan, Vageesh D C, Balaraman Ravindran
Indian Institute of Technology, Madras
{abhiram, vaishnavh, vageesh, ravi}@cse.iitm.ac.in

***Abstract*—Massively Multiplayer Online Role Playing Games (MMORPG) form a genre of games that significantly dominates today's interactive-entertainment industry. With over seventeen million subscribers, issues with large-scale deployment need to be dealt with, demanding novel models and distributed solutions. In this project, we show that social networks within the game can be exploited to make significantly better predictions about player behaviour. In particular, we are concerned with two problems, (i) Predicting when a specific player will be online given his history, and (ii) Predicting player churn. Though these problems are not completely disjoint, both of them are of practical interest, especially for large-scale game developers. For example, the former can be used to predict server load at a given time; and the latter could be used to re-engineer quality of service and to stay proactive about the game's status in the market.**

By working with the publicly available dataset of the ever-famous game World of Warcraft, we first build baseline models that solve the above problems. We then augment these models with social information inferred from the dataset, and analyze their performance in comparison to the baseline. We observe a significant improvement in prediction in terms of precision, recall and error in the augmented model, indicating that the underlying social network truly carries latent information about player behavior.

***Index Terms*—Churn Prediction, Time Series Prediction, Social Network Analysis, Graph Centrality, MMORPG**

I. Introduction

Massively Multiplayer Online Role-playing games form a billion-dollar industry in the field of interactive entertainment. With over 55% of the internet users also being online gamers[11], and with over seven million subscribers just for the one MMORPG, *World of Warcraft,* in the July of 2013[1], issues with large-scale deployment of such games crop up and need to be dealt with in novel ways. Managing and distributing server load is a typical issue in games with high subscription count, with QoS estimation being a major concern. Making accurate predictions of this load will help in better provisioning and allocation of resources, along with optimizing power consumption. Also, churners, i.e people who leave the game forever, are a huge problem to large scale game developers. Churn analysis plays a major role in helping game developers since it helps them understand the several factors that lead to players leaving their game. Personal interests, competing games, social influence, shifting interests are some of the several possible reasons for churning.

We are specifically interested in these two problems: (i) Given a specific player and his history of online activity, we would like to predict whether or not he would be online at a given future time instant. (ii) In addition, we would also like to predict if and when the player is likely to leave the game forever.

A. Our Contribution

A common approach to most of these problems is to analyze and estimate player behavior at the macro level i.e to work with the system of players as a whole[2]. Instead, in this discussion, we are concerned with making predictions about player behaviour at the micro level where we treat each player individually, the results of which can be consolidated into a macro output. The reason we have chosen such a bottom-up approach, as we shall explain in detail soon, is the power to augment the model with social information at this micro level. We first build baseline models for the given problems, and then augment those models with social networking features derived from the players' activity, to show that the models can be enhanced. We use the publicly available World of Warcraft Avatar History (WoWAH) dataset[7] to perform validation of our hypothesis. We would like to stress on the fact that we are not validating our *models* per se, but rather validating our hypothesis that augmenting social information *does infact enhance any* model that lacks it.

We infer a social network from the given data using a similarity metric that captures the session similarities among players in terms of time of gameplay and geographical zones within the game. Henceforth, we shall refer to this network as the *similarity matrix.*

For predicting whether or not a specific player will be online at a given time instant, we model the player's activity history as a binary time series, each bit representing whether the player was online at that time instant. We analyze the autocorrelation and partial autocorrelation curves of the time series and show that a linear model can be employed to model it. We then go ahead to apply AR and Neural Network based techniques to perform prediction and measure precision. Further, to augment this model with social information, we find *k* players most similar to the given player using the similarity matrix, and use their time series as an auxiliary series to better the prediction of the given player. This is done using the NARX network model where the auxiliary series are fed as external inputs. Our underlying hypothesis is that a player is more likely to be online at a given time instant if his friends are also likely to be online at that time instant. We observe that there is a significant improvement in performance, and also a strong correlation between this improvement and the average similarity of the given player to the *k* players whose time series we used as external input.

For predicting player churn, we formulate a simple linear regression problem where the oldest 10% of the history is used for generating the features for training, and the entirety is used for generating the output values. In the given dataset, this translates into using the first three months of data to predict possible churn in the next three years. The following three features are used: *Playtime density*, *Last seen*, *Mean of occurence times*, - all calculated over the first three months, and the same three values are used as output, but now calculated over a three year span. Since *Last seen* is more noisy, we instead work with the *third quartile* as a more stable estimator of gamer departure in the output. With the given setup, we obtains error measure in prediction. Further, to augment this model with social information, we introduce a new input feature which is a measure of Graph Centrality of the corresponding player within the similarity matrix. Our underlying hypothesis is that players with high centrality are more *attached* to the game and hence more unlikely to quit the game early. With this new feature, linear regression shows improved performance compared to the previous error, under three different similarity metrics. A neural network implementation of the same formulation also showed reduction in error upon augmentation. Several peripheral results and correlations were also observed during the process.

A more thorough treatment, analysis and justification of the above summarization can be found in the following sections, which are organized as follows. Section 2 reviews related work in the field. Section 3 gives a description of the dataset and presents some preliminary observations. Section 4 describes how we extract social information from the given dataset. Section 5 describes how we perform time series prediction, and shows how the extracted social information is used to better the prediction on baseline models that we use. Section 6 describes our approach to churn prediction and how centrality is used as an additional feature to improve a baseline model. Also presented is a short discussion with respect to observations made on the correlation plots. Section 7 presents another baseline model and augmentation procedure that we use to predict if a specific player will be online a given time instant. We finally wrap things up in the conclusion and present scope for future work.

II. Related Work

In this section, we briefly survey related work, without restricting ourselves to the MMORPG domain. In the field of predicting player gameplay time, [4] fit a Weibull distribution to the player session time cdf generated from a trace. [10] took the macro approach discussed earlier and worked on generalized web servers, presenting the performance of various machine learning algorithms on the server load trace. [6] performs a variability and predictability analysis of players and exploits the results to propose a server-consolidation strategy.

A variety of approaches exist for churn analysis in several domains (Eg. CHAMP [8]). In the context of multiplayer games, [3] builds a model for player motivation and after generating features for it, performs classification using ensemble methods. [9] uses average daily playtime and playing density of the players as features and performs SVM regression to predict time of churn. [5] takes a social influence based approach and models the positive and negative influences between the players in the game to predict churn.

Except for [5] , none of the approaches incorporate social interactions among the players in the game into their prediction model. We would like to extend this approach, in some sense, and show that augmenting any existing model of player behaviour with information regarding social interaction/features will better the existing model.

III. Dataset Description and Analysis

The World of Warcraft Avatar History Dataset[7] is a publicly available dataset that has been used in several publications in game research. Spanning over three gigabytes of data, the dataset is a log of three years of WoW gameplay. At sampling intervals of ten minutes each for the entire span of three years, the authors of the dataset have logged information regarding all the players that were online during that time sample, including their geographical zone of presence within the game (one among the 229 zone divisions) and their guild membership. A brief description of the dataset is given in the table shown. With data for over ninety one thousand players, the dataset covers the *Light Realm*[1], which is a common realm for Chinese players.

Avatar ID	Guild ID	Sampling Time	Zone	Skill level
2202	34	00:13:45 06/30/2008	Ashenvale	24

Table I: A partial entry in the WoWAH dataset

A. Pruning the Dataset

Since we are concerned with time series and churn prediction, we are uninterested in players who do not play a significant role i.e players who haven't played much. We term such players as *Visitors*. We prune them out by removing from the dataset, players who have not appeared online for more than seven days. It is interesting to observe the cumulative distribution of the number of players who have appeared for *x* days. A power law is fairly evident from the plot i.e it is clearly observed that there are a large number of players who have played only for a short amount of time.

IV. Inferring the Social Network

In this section, we discuss how we infer social network relations among the game players. The methodology described here is limited to the World of Warcraft framework but can easily be extended, as we shall explain, to other MMORPG frameworks as well. Within World of Warcraft, players can form in-game associations called *guilds* to make grouping and raiding easier. Guilds offer several benefits including easy access to group formation (or raids), free items, trading of

[1] A realm is an instance of the WoW game world.

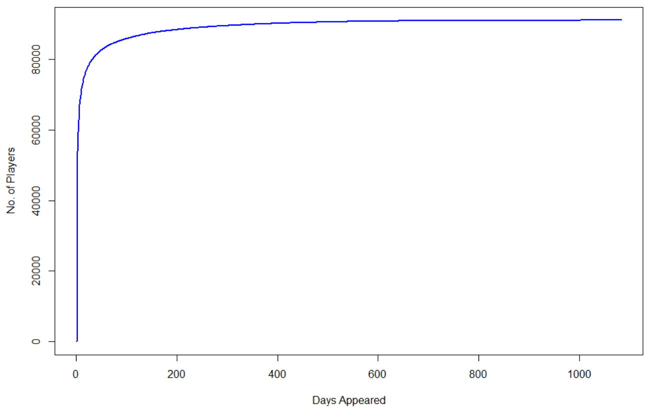


Figure 1: The CDF of the number of users who appeared online in $\leq x$ days

skill masters, etc. Becoming a part of a guild is considered as becoming one among the family of players that the guild represents, where each player strives to better the experience level and quality of the guild. The notion of guilds immediately introduces cliques into the social network graph of the players in the game. Although there is a significant amount of inter-guild interaction among the game players, we make a simplifying assumption that this is negligible when compared to intra-guild interactions. Since we will only be interested in *k* players most similar to the given player, and since these players are almost always intra-guild, the assumption stays valid. Within a guild, we would now like to estimate the similarity measure (weight of the edges) between every pair of players. Although we would ideally like to have some sort of a chat history or interaction history that we can infer this similarity from; in order to stay within the constraints of our dataset, we infer similarity in a constrained structural setting. Our similarity measure is a modification of the Jacard Coefficient. The measure essentially calculates the ratio of the number of (weighted) time samples when both of them were online *together* to the number of time samples when either of them was online. The weights that we introduce to the samples in the numerator is a penalty on the spatial non-locality of the players in concern. The underlying hypothesis is that two players playing simultaneously in the same geographical zone within the game are in essence more similar than two players simultaneously online, but playing in different zones.

We would like to point out that the similarity between two players can be calculated over various time *intervals,* and hence we will work with similarity between two zone series $\overrightarrow{u}$ and $\overrightarrow{v}$ instead. The pool of zones is the set $Z = \{z_0, z_1, \ldots z_G\}$ where G is the total number of zones, and z_0 is the *null* zone: the default zone value used when the player is not online at the corresponding time sample. The similarity $\sigma(\overrightarrow{u}, \overrightarrow{v})$ is given by

$$\sigma(\overrightarrow{u}, \overrightarrow{v}) = \frac{\sum_{t=1}^{T} \mu(u_t, v_t)}{\sum_{t=1}^{T} (1 - \mathbb{I}(u_t = z_0, v_t = z_0))}$$

where $\overrightarrow{u} = (u_1, u_2, \ldots u_T)$ and $\overrightarrow{v} = (v_1, v_2, \ldots v_T)$ represent the geographical zone series of players *a* and *b* respectively for a time interval of size *T*, with

$$u_i \in Z \ \ and \ \ v_i \in Z \ \ \forall \ i = 1, \ldots T$$

$$\mu(u_t, v_t) = \begin{cases} 1 & u_i = v_i \ \ and \ \ u_i \neq z_0 \\ 0 & u_i = z_0 \ \ or \ \ v_i = z_0 \\ w & u_i \neq z_0 \ \ and \ \ v_i \neq z_0 \end{cases}$$

We experiment with three different penalty values *w* = 0, 0.5, 1. Higher penalties for spatial non-locality correspond to lower values for *w*. Using this measure, we obtain similarity matrices for each guild. Based on the situation, we extract graph centrality measures or top *k* similar players using this matrix when required. The penalty weights *w* on spatial non-locality in this case is what is specific to WoW. It is quite possible that there exist games where spatial locality within the game is not a necessary condition for social interaction. In such situations, we will have to resort to explicit interaction logs to infer the desired similarity.

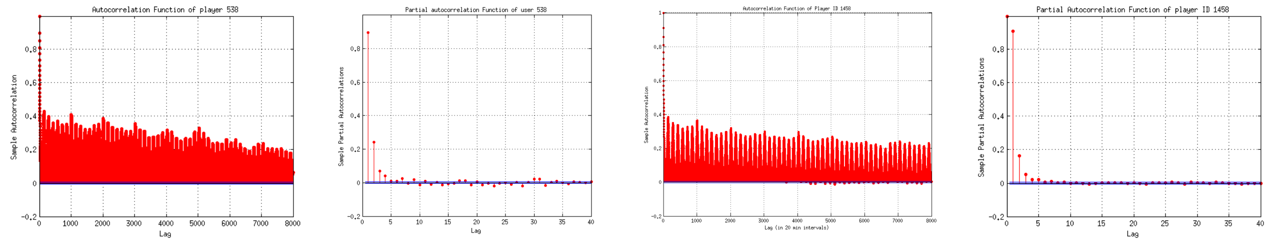
V. Time Series Prediction

In the given data set, we extract the behaviour of each individual player and model it as a time series with multiple parameters. The player activity history is a binary time series in which each bit represents whether or not the player was online at the corresponding time sample.

With this set up, and given the player's history, we would like to predict the player's activity in a given future time slot. In essence, we would like to perform time series prediction. In addition to doing this, we also explain the applicability of linear time series models for player prediction in this scenario, which can be viewed as an orthogonal contribution of this work.

A. Applicability of Linear Models

To choose the right time series model, we analyze the autocorrelation function to check for stationarity of the time series. In addition, this also helps us identify seasonality, if present, in the given time series. The lag denotes the difference in the epoch from where we correlate the same signals. This is normalized so that the value ranges between [-1,1]. A positive value nearing one indicates that the signal structure nearly repeats itself while a value nearing -1 indicates that the signal is out of phase, but yet has a repeating structure. A value close to 0 indicates that the signal has negligible structural relation among its values. Seasonal variation of the autocorrelation plots on different offsets implies seasonal variation of the original signal. To ascertain the nature of this seasonality, we employ Partial Autocorrelation. With this plot, we identify the time signals from history that influence the value of the time series at present. For a given offset, if we were to observe a fixed number of contiguous spikes initially from the origin while the rest of the points were well within the error interval, we can conclude that a linear model can be employed; ascertaining the influential points as the significant spikes in the partial autocorrelation plot. These plots exhibit



(a) Player ID 538 - Autocorrelation (b) Player ID 538 - Partial Autocorrelation (c) Player ID 1458 - Autocorrelation (d) Player ID 1458 - Partial Autocorrelation

Figure 2: The autocorrelation and partial autocorrelation plots for the players with the given ID. We observe a fixed number of contiguous spikes initially from the origin in the partial autocorrelation, while the rest of the points were well within the error interval. We can hence conclude that a linear model can be employed.

such behaviour for all players, thus allowing for the use of linear models in this scenario[2].

We observe that most players show strong patterns in their gameplay time, with periodicity of around one day. The autocorrelation plot backs up this intuitive observation.

B. Predicting user patterns

We employ the following models to predict the user behaviour

1) AutoRegressive (AR) model: This is one of the linear time series model. An AR is defined by its order ρ for our case we have take $\rho = 10$ using the partial correlation plots. One of the prerequisites is that the time series is wide sense stationary.

2) Neural Network: We employ here 2 variants of the closed loop feed forward neural networks in predicting the time series assuming a non linear behaviour of the time series. We use these tools to see if they provide extra performance over the linear models. They are the following:

Non-linear Autoregressive Network (NAR) The network predicts the future output based on the history of the previous outputs.

Non-linear Autoregressive Network with external input (NARX) The network predicts the future output based on the history of the previous outputs and by using an external time series.

C. Baseline Results

We choose a particular set of players with ID 358 and 1458 where one played less frequently and other who is a frequent player. The AR model is trained with 80-20 train-test split while the NAR is trained with 10 hidden nodes and 70-30 train-test split

The prediction of the various models is as follows:

[2]By this scenario, we refer to the generic scenario of MMORPGs, the kind of games where we can expect strong pattern in gameplay time of non-visitors

Model	accuracy(%)	precision (%)
AR	43.24	89.97
Neural network (NAR)	99.57	85.67

Table II: Player 358

Model	accuracy(%)	precision (%)
AR	98.00	92.16
Neural network (NAR)	98.02	87.50

Table III: Player 1458

D. Augmented series

The main hypothesis here is that given the history of players who are socially close the given player, we can improve prediction using the underlying latent information about the given player's playing pattern, that the similar player carries. When we augment the above time series with a similar player series we see that we encounter a better performance in terms of precision and in accuracy. This tells that the additional series is instrumental in giving more insights of the current player and predict this trend. Here we employ NARX and it is trained with 10 hidden nodes and 70-30 test - train split. The results obtained are as follows:

User ID	accuracy(%)	precision (%)
538	99.54	89.98
1458	98.02	92.39

Table IV: Prediction using NARX with the auxiliary series of similar players as external input

When using NARX, we trained the players time series with the auxiliary time series of the most similar player with the given user. For player 358 and 1458 its players 359 and 1450 respectively. They have a highest similarity score in all variants with each other. Once can clearly see a increase in precision which indicates that using similar users obtained using the user similarity measure does indeed improve the prediction.

VI. Player Churn Prediction

In this section, using the player's history, we predict several values that quantify player churn. At the outset, we wanted to understand the significance of the centrality measure derived from the *similarity matrix*. Does it contain any *latent* information? This boils down to questioning the legitimacy of the inferred social network.

A. Modelling the Features

We make a reasonable assumption that we are given the oldest 10% of the history of the player, and are asked to predict churn in the entire time span. This 10% translates to around three months in the given dataset. Given the entire player history in the first three months, we set out to predict the approximate time at which he would leave the game. For this, as part of the baseline model we first select a set of features that characterize the player's playtime behaviour in the first three months. As a naive set of options we chose the following as the *input* features:

- *Playtime density*: how frequent has the player come online during the first three months?
- *Mean occurrence*: what is the average of the times at which the player was present?
- *Last seen*: what is the last date on which the player was seen?
- *Third quartile*: the median between the median and the last occurrence.

Note that the above features, in some sense, summarize the binary time series of a player's online presence in the first three months. Also, we can see the *mean*, *third quartile* and the *last seen* time as a spectrum of his end-points in the three-month span. *Last seen* is a sharp value that tells us when the player left (within those three months) exactly. Clearly, if the last seen time is *too* early, the model will be encouraged to predict an early churn (though this might not be an accurate thing to do). It could also be a noisy value if there is significant different between say, the *last seen* time and the *second last seen* time. The third quartile is a moderately shaded-down value of the last-seen point. We expect this data to be a considerably less sharper and less noisier and more representative of the user's actual playtime behaviour. The *mean* however summarizes a much larger span of his playtime and is a generously shaded-down value.

B. Predicted Feature

Since we used a spectrum of input features characterizing a player's end-point, we followed suit for prediction too i.e., we tried to learn parameters for models that predict the mean, the last seen and third quartile for a span of three years. The basic idea is to use the first three months of playtime data to predict churn in the next three years.

C. Linear Regression Model

The aim of this exercise was not to come up with the best features possible. Rather, we wished to analyse how the prediction model behaved when it was provided with an extra feature - the degree centrality measure. Since the degree centrality was evaluated considering a single guild as a separate community of interacting players, we chose different guilds of users and constructed input matrices corresponding to each guild *individually* and performed cross-validation on them.

D. Performance

The following is the discussion of the first experiment we tried as a preliminary step in understanding the various similarity measures and testing their utility in our prediction models.

We chose 40 different 85-15 train-test splits of the data to test the error of the regression model that predicts *the last seen* value for the span of three years [3]. We consider four different cases. One of them is devoid of the centrality measure (which essentially is a piece of information from the inferred social network). The remaining three are augmented with the centrality measure derived from various levels of penalty imposed on spatial non-locality while calculating similarity between two users in the three month span i.e., *low, medium and high*. Also, the centrality measure here is the sum of the *top 5* similarities of a user.[4]

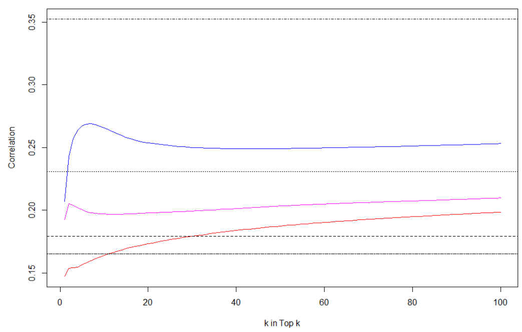
Cases	Variance	Mean Error
Without Centrality Measure	0.2269562	4.446092
Low Penalty	0.2072139	4.404702
Medium Penalty	0.1998799	4.147091
High Penalty	0.1977435	4.147091

1) Observations: Firstly, we point out that adding a feature will *always* reduce the error on the *training* data. That is, the parameters become richer and hence have a greater degree of freedom to settle themselves in a configuration that better matches the seen data. Unfortunately, this can run into trouble when the model is too restrictive and *overfits* the data, which is exactly why one would run cross-validation. Fortunately though, adding any kind of similarity measure to the features ends up giving a model that better fits the *unseen* data too, i.e, the 15% fold from the validation run. This can be observed from the **Mean Error** column.

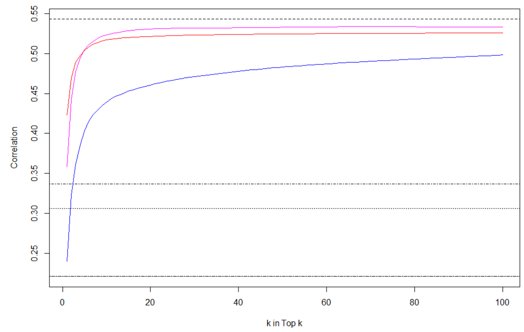
The second observation is that it's not just the error, but it is the **variance too that has reduced** on adding the centrality features! Often, the bias-variance tradeoff is such that, when the bias in the model is reduced its variance increases and vice versa. That is, when the model tries to fit better on an average is exhibits more variance in its accuracy. It turns out that adding information about similarity has helped improve the prediction along two orthogonal dimensions i.e., the bias and the variance!

The third, more interesting observation is regarding the behaviour of the three kinds of similarity measures. Recall that we had defined three different models that help infer a social network between the players. The most conservative model where we considered two players to be interacting *if and only if* they had played at the same time *and* at the same zone, turns out to add the greatest information. This tells us that the other similarity measures were noisier. That is, they

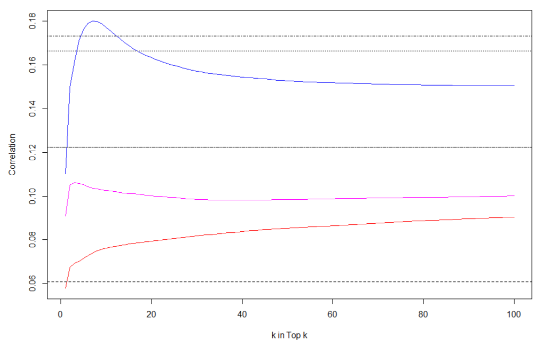
[3] Note that this is not exactly churn but a good approximation of the same.
[4] All values reported here correspond to normalized feature vectors.



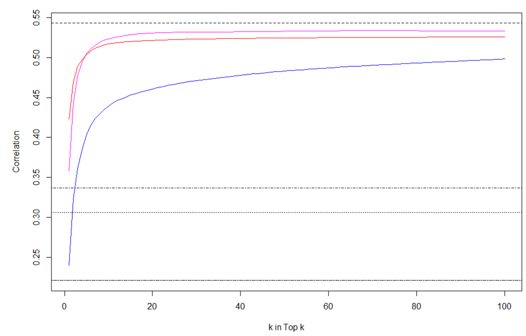
(a) Plot 1: Correlation with *last seen* time



(b) Plot 2: Correlation with the *third quartile*



(c) Plot 3: Correlation with the *mean time of occurence*



(d) Plot 4: Correlation with the *playtime density*

Figure 3: Correlation of the degree centrality measure with different output features for varying similarity measures over the entire three year span.

misrepresented interactions between users who happened to play roughly during the same time zones but never actually knew each other for they were playing in different zones altogether. Since we did not want to make assumptions about the effect of zone matches, we decided to experiment with various similarity measures and as the experiment has shown, when zones match, it is more likely that the players are often interacting players.

One must remember that these models aren't really great per se. They are naive attempts at trying to predict the *last seen* feature. We repeat ourselves in saying that the aim of this exercise was to observe and analyze the significance of the centrality measures, and not to construct the best model.

E. Correlation

Our observations in the first experiment led us to try and explore the features better. Here we show a number of plots, depicting the correlation between the features.

All the following four plots show the behaviour of the degree centrality measure with the number of top similarity values that are included in it *versus* the correlation it exhibits with respect to the *output* features measured over the span of three years. The readings are specific to *guild 4*. Experiments on the other guilds were also performed and showed similar results. We present the results for only one guild for the sake of brevity

.

Legend:

- **X axis**: The value k in the definition of the centrality measure i.e., the top k similarity values that are summed up to generate the degree centrality. We limit ourselves to the top 100 players in a guild of about 382 people.
- **Y axis**: Correlation with the following features measured over the whole *three years*:
  - Plot 1 : Last seen time
  - Plot 2 : Third quartile
  - Plot 3 : Mean time of occurrence
  - Plot 4 : Playtime density

  Essentially, the motivation is to see how the degree centrality value that is the *social network information* gathered in the *first three months*, seems to be carried forward as a pattern over three whole years!
- The **blue line** corresponds to the highest penalty similarity measure i.e., high penalty is ascribed to users who are playing in different zones at some given instance.
- The **magenta line** corresponds to a medium penalty.
- The **red line** corresponds to no penalty at all.
- The **uniformly dashed** line is the correlation of the feature considered for the plot with the **playtime density** measured over the *first three months* of the user. This, and the following three lines that are being described are

used as baselines to compare the correlation value of the degree centrality.

- The **uniformly dotted** line is the correlation of the feature with the **third quartile** of the playtime measured in the first three months.
- The **dot-dash** line is the correlation of the feature with the **last seen time** of occurrence in the first three months.
- The **two-dash** line is the correlation of the feature with the **mean occurrence** in the first three months.

Plot 1: It is not surprising that the dot-dash line, or the correlation between the last seen time over the first three months and that over all the three years is much higher than any other correlation values. We could instead focus on the lower part of the graph. Especially the blue curve or the high-penalty, strict degree centrality measure. Our centrality measure, derived from the behaviour of the inferred social network in the first three months, correlates with the *last-seen* measure of the three year span, much better than a) any *other* individual characteristics of the player b) any other lower-penalty centrality measure. This re-iterates the fact that this centrality measure is a much better candidate for prediction than many other measures.

The other observation worthy of noting is how the blue lines behaves with increasing k. The initial increase in the correlation tells us that the centrality measure is becoming *richer*, trying to understand how well-situated a player is in the context of about four to five of his closest players. However, as we increase the value of k this correlation dips (and weirdly increases a little bit later, which we haven't been able to explain). The dip however, is expected because when we try to include a lot of players in the degree centrality measure, we are including *spurious* values i.e., stray similarity values with possibly unknown players with whom the user co-incidentally has exhibited some similarity. It is interesting how this dip due to inclusion of spurious values begins at a value of $k = 5$. We can hypothesize that this is a property of the underlying social network. That is, on an average only about five players are relevant to your life in the game.

The other observation that we're yet to convincingly explain is the fact that the centrality measure based on no penalty - the red line -shows better correlation with any increasing k. It is however, fair to expect that the point at which the (red) graph makes a turn-around is at a higher value of k. In some sense, this measure is a noisier, impurer way of inferring the social network. Hence, for small values of k, the degree centrality measure exhibits a lot of impurity and hence a low correlation value. It would require a greater k - greater than the four or five that we saw for the blue line - to gather sufficient information about the centrality of the person in the graph to exhibit the *best performance*. However, it is a mystery as to why it is ever-increasing in the range that we have seen. It is probable that the measure is so impure that it requires a very high k, that is beyond the limits of our exploration. We must however acknowledge the fact that there *is* a break in all the three lines for $k \approx 5$.

Plot 2: The red, magenta and the blue lines show a *very* similar increase-and-then-dip behaviour and at nearly the same values of k. Hence, it reinforces the fact that this $k*$, or possibly the *average number of relevant friends in the network* is a characteristic of the network that will manifest itself in many observations. The most interesting part of this plot is that the correlation of the third quartile value over the three years with that over the three months is less than the correlation with the degree centrality measured, evaluated around the region of $k*$.

Plot 3: The most exciting part of this plot is that the correlation value that the blue line - or the strictest similarity measure - exhibits with the mean occurrence of the players, is higher than any other possible correlation value.

An observation across the above plots that we're yet to explain is that despite the "last seen" value perceived as being noisy, it shows the highest correlation values with the other features.

Plot 4: The weirdest plot that we have seen is the correlation measured with the playtime density over the three years. Unsurprisingly this is the most correlated with the playtime density measured over the first three months. However, the blue line shows much lesser correlation than the other two lines! The only intuitive understanding we could get out of this was the following: remember, the zero penalty similarity measure is *merely* a function of the bit vectors of the users and hence placing this value *closer* to the playtime density of the users. However, when a penalty was introduced, we effectively introduce *more* information into the system in the form of zones. This in some sense disturbs the close relation that the similarity measure had been following with the playtime density. The addition of penalty can be seen as a perturbation in the relation of the similarity measure with the playtime density.

We also tried to observe the plot of the degree centrality measure (with $k = 5$) measured over the first three months against the playtime density for the span of three years. The similarity measures were min-max normalized for the sake of comparison. This is because the *high penalty* measure will always return a very low similarity measure. Greater the penalty, lower the similarity value. In the following graph, the green, blue and red points represent various penalties in the increasing order.

F. Second Round of Prediction

Motivated by the above results, we ran another round of experiments using linear regression. We report the observations after running a cross-validation that uses a random set of 50 folds. Note that the same folds were used across all the experiments shown below. The degree centrality measure was limited to $k = 5$.

The following table presents the results for prediction of the *last seen* measure.

Cases	Variance	Mean Error
Without Centrality Measure	0.3573	4.3901
Low Penalty	0.3898	4.389
Medium Penalty	0.4015	4.2779
High Penalty	0.3700	4.133

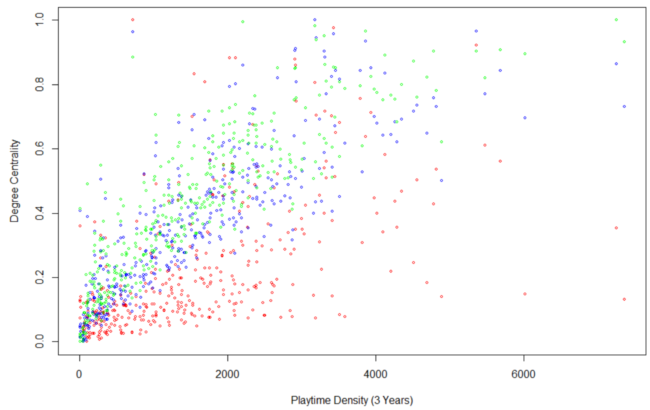


Figure 4: Variation of degree centrality

The following is the record of errors for the prediction of the mean occurrence over the three years.

Cases	Variance	Mean Error
Without Centrality Measure	0.2090	0.9659
Low Penalty	0.2190	0.9711
Medium Penalty	0.2169	0.9570
High Penalty	0.2029	0.9328

The following is the set of errors and variances recorded for the prediction of the third quartile for the span of three years.

Cases	Variance	Mean Error
Without Centrality Measure	0.3830	2.2110
Low Penalty	0.3958	2.212
Medium Penalty	0.3888	2.1816
High Penalty	0.3622	2.1409

The most striking observation, apart from the fact that adding the strict similarity measure has done the best in all cases, is that the error in the prediction of the mean, the quartile measure and the last seen values (or the maximum) is increasing in that order. This is not however surprising as we already stated that we perceive them as increasingly becoming noisier in that order. Moreover, the variance also follows *nearly* the same pattern.

Increasing k: We repeat the above experiment for a degree centrality measure that takes the first hundred similarities. That is $k = 100$.

The following table represents the results for the prediction of the "last seen" measure.

Cases	Variance	Mean Error
Without Centrality Measure	0.3944	4.4077
Low Penalty	0.4240	4.3401
Medium Penalty	0.4276	4.300
High Penalty	0.4252	4.1892

The following table represents the results for the prediction of the mean occurrence.

Cases	Variance	Mean Error
Without Centrality Measure	0.1865	0.9428
Low Penalty	0.2013	0.9372
Medium Penalty	0.2024	0.9309
High Penalty	0.1917	0.9066

The following table represents the results for the prediction of the third quartile measure.

Cases	Variance	Mean Error
Without Centrality Measure	0.3361	2.1532
Low Penalty	0.3493	2.1341
Medium Penalty	0.3507	2.1212
High Penalty	0.3459	2.0859

We again see a similar pattern here. What however is intriguiing is that the error ***and*** the variance values have reduced considerably despite the fact that we didn't choose the *optimum* k. This is another observation we're yet to explain.

Increasing timespan: What happens if the similarity measure was measured over a larger time-span? Say, over the first six months? We perform this experiment for the high-penalty case. (w/o - without)

Cases	Predict	Variance	Mean error
Last Seen	W/o Cent. Meas.	0.4223	4.4716
Last Seen	With Cent. Meas.	0.3850	4.1303
Mean	W/o Cent. Meas.	0.2193	0.9655
Mean	With Cent. Meas.	0.2000	0.9096
Quartile	W/o Cent. Meas.	0.4210	2.236
Quartile	With Cent. Meas.	0.3796	2.1134

As expected, roughly, almost all mean squared errors have now decreased. What is to be noted is that, the dip in the error in the case where the information from the social network is *not* used is smaller in compared to the dip in the error in the case where the degree centrality measure is used.

VII. Online or not?

In this section, we present an alternative approach to predicting whether or not a player will be online at a given time instant. We were interested in testing the social network we inferred by conducting another experiment: using the *friends* that this similarity measure tells us, is it possible to determine whether a person is about to come online, with merely the knowledge of whether his friends are online? We expect this to be a model that could have a lot of scope as it will help us make online decisions.

On a related note, this model sounds like that of an epidemic. A person comes online if at least a minimum number of his friends come online. However, the relation might not be as simple. We define the following methodology to perform predictions and verify the social network.

Procedure

We require an input matrix from which we learn patterns of what combinations of a set of friends being online *excites* a user into coming online. Thus, we would want an exhaustive $T \times k$ matrix that tells us the playtime behaviour of k friends across a contiguous set of T samples. In our case T spans three months. The output that we will train on will be a bit-vector of size T. In order to test the model we could either perform cross-validation or extract samples from a later date, that falls beyond the range of our T.

What is the drawback of the first approach? We already know that these players are similar to the user considered. Hence, it won't be a surprise if this already-recorded information manifests as a good result in the cross-validation as these time samples are from within the range of the similarity measurement. Cross-validation can only probably tell us whether this model is actually a sufficiently valid model that models the data well or not. However, there is a hurdle: there is *too* much data present in the timespan of three months. The matrix $T \times k$ will be quite large. How do we overcome this? Instead of learning from the exhaustive record of data, we instead sample about 5000 points.

The approach we adopt to sample is that, we keep a record of the times at which the user came online, along with the times at which at least one of his friends came online. We set a probability value which defines the number of times a time instance is sampled from the list of times at which the user is online. This is to make sure that there is no *class imbalance*. Remember, the classes here are being offline and being online. More the time samples drawn from the list of times that the user was actually online, more is the proportion of "being" online class.

Another hurdle in the above process is that, the sampling times are not simultaneous across people. We overcome this by softening the "is-online?" function of the user. Thus, a user is online with a truth value of 1 within an ϵ-time neighbourhood around the time at which he was recorded to be online. At other times, it is modelled as a linear function decreasing with respect to the distance of the closest recorded time of him being online.

A. Observations

Having sampled data, we can either try to fit a linear regression model or a logistic regression model or even an ANN to classify datapoints. We measured the precision and recall values too. The first observation we made is that the threshold we choose to classify a point as being online-or-not is difficult to choose. We're yet to model this. Since we have not measured this against a similar baseline model, it turns out to be difficult to evaluate this procedure. Nevertheless, we report the values here.

A simple linear regression model trained on 3000 points for a user from guild 4, gave 61.3% accuracy on test data sampled from the next six months. Note that this data had a class imbalance, biased in number towards being online.

	Precision	**Recall**
Offline	0.496	0.363
Online	0.658	0.769

As expected, this accuracy actually decreases when the test data are sampled from a later period. This is because, our underlying assumption in these models is that the similarity we measure at some time, tends to stay on for some more time. Given sufficient time, such similarities might become meaningless.

As stated previously, *cross-validatio*n on hundred random folds of train data which consisted of 33% off-line time gave the following results which turn out to be considerably good:

	Precision	**Recall**
Offline	0.741	0.855
Online	0.657	0.479

We performed *cross-validation* on the data sampled ***after*** the time range during which the similarity measure. This should tell us how fast the similarities between players might decay. The sampled *test* data contained equal amounts of off-time and on-time.

	Precision	**Recall**
Offline	0.484	0.311
Online	0.629	0.777

The neural network model fit on the train data, returned the following results on the test data:

	Precision	**Recall**
Offline	0.484	0.436
Online	0.646	0.689

VIII. Conclusion

By taking a micro-level approach to predicting user behavior, we have shown that, for several baseline models, augmenting social structure information decreases the prediction error in various scenarios. In the interest of the MMORPG game developer community, we believe that this approach can significantly help improve resource management and churn analysis. We observe that spatial locality of players within the game space is a critical parameter in evaluating player-player similarity in World of Warcraft-type MMORPGs. By employing time series prediction tools coupled with linear modality of the data, we were able to achieve significant improvement in precision for predicting player gameplay time. We were also able to improve churn prediction by exploiting centrality measures on the player similarity matrix. We believe that this latent information that is carried by the underlying social structure can be used to obtain richer insights on behavorial patterns.

IX. Future Work

A. Better Characterizations of Player Behaviour

In these experiments, we have confined our discussions to experiments that were made using simple models as our preliminary aim was to test the hypothesis that there is some latent information existing in a network of players that helps predict player behaviour in general. Player behaviour, in turn, was characterized by simple parameters of the bit vector corresponding to the user's playtime - namely the last seen, the mean and the third quartile. We would like to however, experiment with a richer set of variables and also understand how "easier" they are to predict. For example, amongst the parameters that we have dealt with, we saw that predicting the mean time of occurrence gave us the lowest error as it would be the least noisy.

B. More measures of Centrality

While characterizing player behaviour is one dimension that has scope for exploration, we would be more interested in developing and analysing more measures of centrality. We have restricted ourselves to a measure of the degree by summing up the top k similarities. Are there more interesting measures? For example, this measure in some sense estimates centrality upto one level of friends in the network and the assumption is that "if a user has “thicker” friends, he is more likely to be well-situated in the network and in the game, and is less likely to leave the game earlier". We would like to reformulate this ***recursively***, thus: *if a user has thicker friends who are well-situated in the network and in the game, he must be well-situated in the network too (and is less likely to leave the game earlier)*. This hints at a *PageRank/SimRank* kind of centrality measure.

Apart from such a (complex) formulation, we could also explore more standard measures applicable to a weighted complete graph. But note that, if we would like to apply a measure that does not apply to an edge-weighted graph [5], we can always *prune* the complete graph by retaining the top k edges falling on every user or using some kind of a minimum-spanning tree approach and apply such a centrality measure. As an example, we could have run MST (or a variant that produces any subgraph, not necessarily a tree) on the clique considered on the players of a specific guild, and used the number of degrees a user has in this tree as the measure of centrality. We might expect the *leaf* players to be less attached to the game and have an earlier churn.

C. More measures of Similarity

Taking a step behind, we would also like to re-analyse the formulation of our similarity model. This similarity measure faced the biggest hurdle of the experiment in the sense that, the dataset we used did not explicitly convey information about the interaction between the players. Firstly, it will be interesting to run the same experiments on another dataset that gives us a description of player interactions from where measuring similarity would be more straightforward. Secondly, even in this dataset, is there a way of defining better similarity measures? We have covered a spectrum of similarity measures varying over zone-matching penalty. It would be interesting to experiment possibly other measures of similarity too.

[5]The edge weights are the similarities thus calculated. The graph is complete as we have measures of similarity for every pair of users.

D. Better models

As stated already, we confined our experiments to simple models that were intended to understand the relevance of the social network information. There has been work done previously that used support vector machines to predict churn for each of the following sixty days at any given point of time. Moreover, this model used the input data measured over k equally large time intervals. For example, in our case, we would want to split the three months' data into 12 periods each a week long and estimate the playtime density for each week. This would better capture the dynamics of the player with time and hence add more information to the model.

The same idea is applicable to the centrality measure: measure the similarities between the user for every week and produce degree centrality measures for each of the twelve weeks. How will this help? One case that we expect to be interesting in this regard is when we observe the degree centrality of the user to be decaying with every subsequent week. This would probably be a good representation of the user "leaving" the network of users due to decaying interest. There could be many other patterns that are not simple to be explain verbally, that, say, a support vector machine can spot and learn from. Thus, this direction has a lot of scope for exploration too.

E. Easier computation

The computation of similarities and fitting the models might be expensive to perform for all pairs of players. As has already been seen in this project, we had stuck to considering only intra-guild interactions for the same reason. In real life, we would like to develop a model that is scalable and is adaptable to the dynamics of the network too.

References

[1] Mmorpg, http://en.wikipedia.org/wiki/massively_multiplayer_online_role-playing_game.

[2] Mauro Andreolini and Sara Casolari. Load prediction models in web-based systems. In *Proceedings of the 1st International Conference on Performance Evaluation Methodolgies and Tools*, valuetools '06, New York, NY, USA, 2006. ACM.

[3] Z. Borbora, J. Srivastava, Kuo-Wei Hsu, and D. Williams. Churn prediction in mmorpgs using player motivation theories and an ensemble approach. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 157–164, Oct 2011.

[4] Francis Chang and Wu-chang Feng. Modeling player session times of on-line games. In *Proceedings of the 2Nd Workshop on Network and System Support for Games*, NetGames '03, pages 23–26, New York, NY, USA, 2003. ACM.

[5] Jaya Kawale, Aditya Pal, and Jaideep Srivastava. Churn prediction in mmorpgs: A social influence based approach. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, CSE '09, pages 423–428, Washington, DC, USA, 2009. IEEE Computer Society.

[6] Yeng-Ting Lee and Kuan-Ta Chen. Is server consolidation beneficial to mmorpg? a case study of world of warcraft. In *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, CLOUD '10, pages 435–442, Washington, DC, USA, 2010. IEEE Computer Society.

[7] Yeng-Ting Lee, Kuan-Ta Chen, Yun-Maw Cheng, and Chin-Laung Lei. World of warcraft avatar history dataset. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 123–128, New York, NY, USA, 2011. ACM.

[8] Brij Masand, Piew Datta, D.R. Mani, and Bin Li. Champ: A prototype for automated cellular churn prediction. *Data Mining and Knowledge Discovery*, 3(2):219–225, 1999.

[9] Pin-Yun Tarng, Kuan-Ta Chen, and Polly Huang. On prophesying online gamer departure. In *Proceedings of the 8th Annual Workshop on Network and Systems Support for Games*, NetGames '09, pages 16:1–16:2, Piscataway, NJ, USA, 2009. IEEE Press.

[10] Suthee Chaidaroon (unsuthee@stanford.edu) Joon Yeong Kim (kim64@stanford.edu) Jonghan Seo (jonghan@stanford.edu). Server load prediction, http://cs229.stanford.edu/proj2009/chaidaroonkimseo.pdf.

[11] Z. Z. Eric Wan and 2006 In Pacific Epoch Red Innovation Report Series. X. Xu. 2006 online game report.