Predicting Interaction Quality in Online Social Networks

Ting Tang
tangt@stanford.edu

Aubrey Henderson
aubreyh@stanford.edu

Ding Zhao
zhaoding@stanford.edu

Abstract

The quality of interactions among users on social media sites often depends on both the properties intrinsic to the users themselves as well as on the users' past interactions on the sites. We study the Chatous random chat network, for which interaction quality is measured by the conversation length between users. By accounting for user properties and modeling user interactions as graph properties, we can accurately predict the quality of interactions between pairs of users in this network before they interact. We experiment with two methods of incorporating user and graph properties. First, we build a support vector machine (SVM) based multiclass classification system by combining user and graph properties as features in a principled manner. Second, we decompose the network into a bipartite graph using the most salient user property - gender - and implement collaborative filtering- and singular value decomposition- based prediction systems. We demonstrate that both approaches produce good results, with 62.2% accuracy for the feature-based classification approach and 59.2% accuracy for the bipartite approach.

1 Introduction

Since the inception of Friendster in 2002, online social networks have continued to proliferate and evolve as an increasingly popular medium of social interaction. In fact, current estimates reflect that nearly 60% of the world's population has participated in exchanges via Facebook, Twitter, Google+, and/or LinkedIn—and this number is expected to climb. According to [6], 98% of 18- to 24-year-olds have been categorized as active participants/consumers of social media.

For many of the aforementioned services and, in particular, online dating websites, an essential component to their continued success lies in the automatic recommendation of viable connections. Such is the case with Chatous, a text-based, random chat network that pairs users from over 180 countries. For Chatous, our goal is to match users such that they will interact in a high-quality manner, as measure by the length of their conversations.

In this paper, we describe our application of network analysis and machine learning (ML) techniques toward the identification of user and/or conversation characteristics that are predictive of high-quality social exchanges. This investigation is motivated by the theory that improvements at the level of individual conversations will lead to widespread increases in user retention, satisfaction, and engagement.

In this paper, we show that user interactions can be accurately predicted based on a combination of the properties intrinsic to users as well as on the users' past interactions. We model Chatous as a graph and user interactions on Chatous as graph properties, and illuminate some interesting observations with regards to the Chatous graph. Subsequently, we describe two approaches to user interaction prediction. First we illustrate a feature-based approach in which we build a multi-class support vector machine classification system by combining user properties and network properties in a principled way. Next, we show that the Chatous graph can be modeled as a bipartite graph using user gender properties, and apply collaborative filtering and singular value decomposition techniques to perform the prediction task. Finally, we evaluate our two approaches on a sample set of the Chatous network, and demonstate that both methods achieve high prediction accuracy.

2 Background

Guo et al. (2012) describes the creation of Chatous and several exploratory methods for predicting optimal conversation partners [2]. Initially, the authors sought to assess whether the quality of a conversation was linearly related to a particular set of features derived from user characteristics. A subsequent approach involved implementing a PageRank-inspired algorithm to rank users based on features such as weighted user ratings and conversation length. For their third approach, which produced the best predictive accuracy among the three, the authors leveraged the notion of triads between a participant and his/her candidate matches to predict which of two user pairings possessed the greater conversation length.

Guha et al. (2004) presents a model for predicting the trust relationship between a pair of users by applying trust propagation toward the construction of a web of trust. While explicit trust signals were propagated to neighboring nodes via four types of atomic propagations, one-step distrust was utilized to incorporate distrust signals. When evaluated on the Epinions dataset, this iterative method correctly predicted the trust/distrust relationship on masked edges in the graph with a prediction error rate of 6.4% for the entire dataset and a 14.7% error rate on a balanced dataset.

Leskovec, Huttenlocher, and Kleinberg (2010) discuss training a logistic regression binary classifier for predicting the sign associated with links in an online social network. Evaluating their method on the Epinions, Slashdot, and Wikipedia datasets, the researchers demonstrated that correct edge sign prediction was achieved with an accuracy of 93.4%, 93.5% and 80.2% on the three datasets, respectively. It was reported that predictive accuracy was enhanced by considering both degree- and triad-related features as well as the amount of local network structural context available (as represented by the embeddedness of the edge).

Wang et al. (2011) explored a novel technique for improving the prediction accuracy of online dating recommendations that addresses the initial assignment of potential dates to a target user, *t*. Their algorithm is based on the assumptions that two users share similar partner preferences if both are liked by the same users, and that interactions between similar users and *t*'s candidate matches, *C*, can predict *t*'s behavior toward members of *C*. When evaluated, Wang et al.'s method outperformed collaborative filtering and other traditional recommendation algorithms.

While a significant amount of prior work has focused on predicting the nature of relationships among one set of users in a social network based on the relationships that exist among the remaining set (termed link sign prediction), this positive/negative classification is binary [2][3][4]. Wang et al.'s method for generating a continuous compatibility score between a pair of users addresses this limitation, yet fails to demonstrate a theoretical understanding with regard to which dataset properties yield more accurate predictions. Furthermore, there is little indication that the authors' algorithms and analyses are applicable to other datasets. Finally, the majority of prior work approaches the task of online behavior prediction using either network structural data or non-structural metadata intrinsic to the users, ignoring the potential to combine the two [3][4][8].

3 Methods

3.1 Overview

It is highly probable that online relationships, much like real-world relationships, are non-discrete. When considering the profusion of structural data (e.g. friendship, conversations, etc.) and user characteristics present in the Chatous dataset, the development of a classifier that permits more granular prediction is warranted.

Accurately predicting the length of a conversation two users will engage in is an essential prerequisite to the optimal assignment of matches, as longer conversations are theorized to reflect increased user satisfaction. Prior attempts to predict conversation length in similar networks have typically relied on simple models, minimal features, and/or the exclusive application of either user characteristics or network structural properties. Our construction of a comprehensive model permits a more thorough understanding of the Chatous network's structure and allows the informed application of both network structural properties and intrinsic user characteristics toward the derivation of an optimal algorithm for user matching.

We hypothesize that electing to assign weights to various social interactions on a continuous scale will enable superior modeling and result in more accurate social relationship classification. This motivates us to model the network using a series of directed/undirected graphs to represent different relationship signals between nodes, and implement a multiclass SVM to identify the weight each type of signal contributes toward edge strength prediction.

3.2 Dataset

The dataset supplied by Chatous consists of two tables: user profiles and conversations. While the former associates an identification number, screen name, age, gender, location, creation timestamp, and short biography field with each user, the latter describes the collection of interactions between pairs of users over a two week time period. Relevant columns in the conversation table include the participants' identification numbers, timestamps indicating a conversation's initiation and conclusion, which user (if any) disconnected from the chat, which user (if any) reported his/her partner as abusive, the number of lines submitted by each participant, word vectors comprising all words typed by each participant, and friendship status. Chats are categorized as "Finished," "Long," or "Short" depending upon their length and termination status. The "Long" distinction further denotes that a friendship link has been established between a pair of users. (Note: General statistics appear in the Table 3.1).

Users (Nodes)	332,888
Conversations (Edges)	9,050,713
Nodes in largest SCC	293,673 (0.88)
Edges in largest SCC	6,458,818 (0.99)
Average clustering coefficient	0.24
Number of triangles	45,244,826
Diameter (longest shortest path)	11

Table 3.1 General dataset statistics associated with the complete graph.

3.3 Preprocessing

Initially, conversation entries were partitioned into a training set and testing set. Our testing set was constructed by randomly sampling 20% of the conversations from the complete dataset; the remaining data was reserved for training.

We represented each participant in the user profiles table as a node, and each conversation (irrespective of length) as an edge in the graph. With the intention of assigning numerical weights to user interaction signals such as conversation length, conversation termination, user friendship, user reports, etc., a complete graph, *C*, and three subgraphs were created to model these relationships.

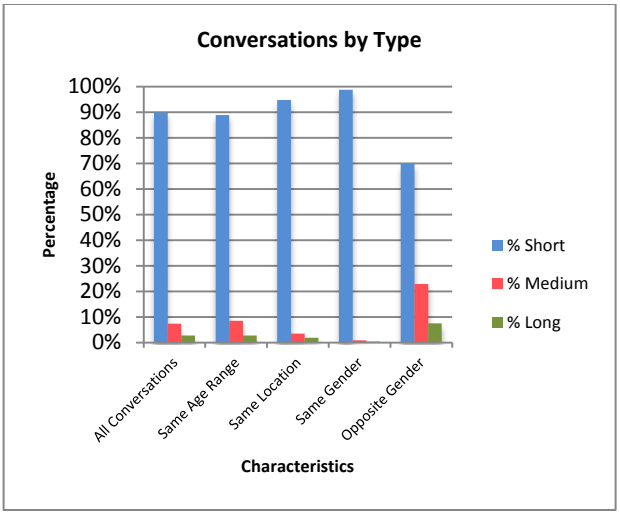
Graph	Type	Description
Complete	Undirected	Each edge represents a conversation between two users.
Chat+	Undirected	Conversations that exceed 5 lines or are currently in progress.
Chat−	Directed	Conversations that are fewer than 6 lines and have been terminated.
Report	Directed	Represents which users have flagged others as abusive.

Table 3.3 Graphs created to enable the extraction of granular network features.

Table 3.4 depicts the fraction of "Short," "Medium," and "Long" conversations that were captured between participants who share an age range, location and/or gender and those between participants of the opposite gender. While "Short" conversations describe those with less than 5 lines, "Medium" chats fall between 5 and 20 lines; any exchange exceeding 20 lines is categorized as "Long." We consider participants whose average age is below X years to fall within the same age group if their age difference is no more than Y years. When $X \leq 20$, Y is 3. For $20 < X \leq 30$, Y is 5. Otherwise, if $X > 30$, the value of Y is 10. To reduce

data sparsity, location was normalized by omitting the city/state for all non-U.S. locations.

Figure 3.4 Fraction of conversations (by type) in which participants share an age range, gender, and/or location, and those between opposite-gender participants.

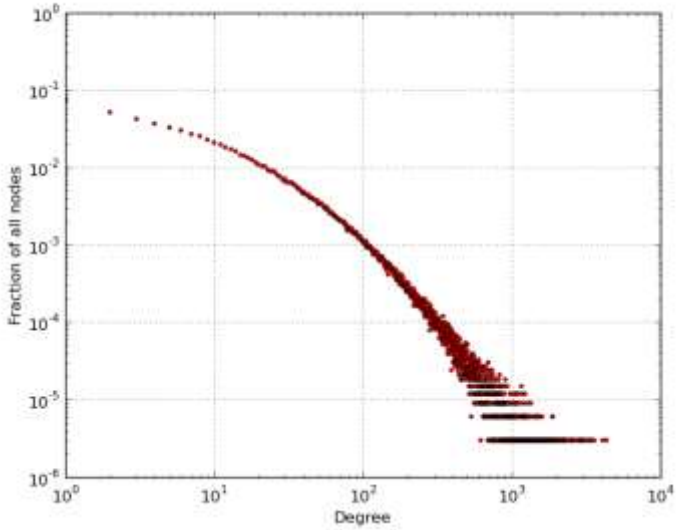


3.4 Network Analysis

We observed a correlation between conversation length and several user profile characteristics, the most salient being gender (see Figure 3.4 above). In particular, users of the opposite gender tend to engage in conversations that extend well beyond the duration typically observed between same-gender participants. Somewhat unexpectedly, geographic location and age similarity are rather poor indications of conversation length.

As a result of the random nature in which users are assigned conversation partners, the Chatous network fails to possess characteristics associated with those following a power-law distribution. In particular, it is evident that users involved in few conversations do not comprise as large of a fraction as a straight-line log-log power-law graph would indicate (see Figure 3.1).

Figure 3.1 Degree distribution of all nodes.



4 Implementation

4.1 Feature-Based Approach

4.1.1 Feature Extraction

Considering that both network structural properties and intrinsic user properties are most likely valuable for predicting user compatibility, our classifier was provided with four feature types.

1. Features that capture user-specific information
   i. Participant A's characteristics– age, gender, location, average conversation length, etc.
   ii. Participant B's characteristics– age, gender, location, average conversation length, etc.
2. Cross features involving user characteristics
   i. Participants share an age range, location, etc.
3. Features that capture user-specific network characteristics
   i. Participant A's degree in the Chat+, Chat–, and Report graphs, and PageRank based status
   ii. Participant B's degree in the Chat+, Chat–, and Report graphs, and PageRank based status
4. Cross features involving network characteristics
   i. Participant A's and B's common neighbors in the Chat+, Chat–, and Report graphs
   ii. Triad formations (i.e. the 16 proposed in [4]) in the Chat+, Chat–, and Report graphs

For SVM compatibility, it was necessary to represent each data instance as a vector of real numbers. After converting non-numeric features into numeric data, we modeled each *m*-category attribute using a vector of *m* values. Within this vector, a single entry takes on the value "1", while the rest remain zeros. To provide a concrete example, suppose we are representing a three-category attribute such as gender (male, female, unspecified). Each feature vector will be assigned one of the following configurations: (0,0,1), (0,1,0), and (1,0,0). Next, each column of the of

the $m \times n$ feature matrix, X, is scaled, such that the mean and unit variance is zero when provided to the formula

$$X_{ij} = \frac{X_{ij} - mean(X_{1j}, X_{2j}, \dots, X_{mj})}{stdv(X_{1j}, X_{2j}, \dots, X_{mj})}$$

where *mean* and *stdv* are computed as

$$mean(X_{1j}, X_{2j}, \dots, X_{mj}) = \frac{1}{m}\sum_{i=1}^{m} X_{ij}$$

$$stdv(X_{1j}, X_{2j}, \dots, X_{mj}) = \sqrt{\frac{\sum_{i=1}^{m}(X_{ij} - mean(X_{1j}, X_{2j}, \dots, X_{mj}))^2}{m}}$$

The construction of bit vectors for the remaining user/network characteristics are depicted in the following table.

Table 3.5 Features selected for SVM input.

Feature	ID	Description
Gender	1	Set to 1 if both users are of the opposite gender and 0 otherwise.
	2	Set to 1 if both users are male.
	3	Set to 1 if both users are female.
	4	Set to 1 if at least one user's gender is unspecified.
Age	1	Set to 1 if at least one user's age is unspecified.
	2	One 6-bit vector representing the age difference between the participants.
	3	Two 6-bit vectors describing the age range of the participants.
Location	1	Set to 1 if both users share the same geographic location.
	2	Two 20-bit vectors representing each user's location.
Graph	1	Two 4-bit vectors representing the in-degree of each participant.
	2	Two 4-bit vectors representing the out-degree of each participant.
	3	One 4-bit vector reflecting the number of neighbors participants have in common.
Conversation Length	1	Each user's average conversation length
PageRank	1	PageRank values computed from the Chat– graph.
	2	PageRank values computed from the Report graph.
Triads	1	One 16-bit vector representing the count of each of the triad configuration.

4.1.2 Principal Component Analysis and Feature Selection

Principal component analysis (PCA) is a statistical procedure that utilizes orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables (principal components). PCA was leveraged to reduce the dimension of features, automatically detect and eliminate noise in the feature set, minimize the complexity of the hypothesis class considered, avoid overfitting, and optimize our algorithm's performance. In conjunction with PCA, we applied a whitening transformation to ensure that our feature vector contained unit component-wise variances.

4.1.3 Classification

Given two users, we attempt to classify them into one of three categories: "Highly Compatible," "Compatible," or "Incompatible." "Highly Compatible" suggests that they will engage in a conversation that exceeds 20 lines. On the opposite end of the spectrum is "Incompatible," reserved for conversations predicted to be of less than 5 lines.

Leveraging the Python programming language and its compatible scikit-learn ML library, we utilized SVMs to perform the classification task. A SVM constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space to maximize the distance to the nearest training data points of any class. Provided with training vectors $x_i \in R^p, i = 1, \dots, n$, in two classes, and a vector $y \in R^p$, such that $y_i \in \{-1, 1\}$, an SVM solves the following primal problem

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1,n} \xi$$
$$subject\ to\ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, i = 1, \dots n$$

Its dual is

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$subject\ to\ y^T \alpha = 0$$
$$0 \leq \alpha_i \leq C, i = 1, \dots, l$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an $n \times n$ positive semi-definite matrix, $Q_{ij} = K(x_i, x_j)$ and $\emptyset(x_i)^T \emptyset(x_j)$ is the kernel. Here, training vectors are mapped into a higher (possibly infinite) dimensional space by the function $\emptyset$ [5].

Since a SVM is only capable of binary classification, the "one-against-one" approach will be applied toward multi-class classification [as cited in 5]. As we have three classes, $3 * \frac{3-1}{2} = 3$ classifiers were constructed and each trained data from two classes. Each SVM is assigned a weight of 1, and predictions are made based on the largest number of votes the SVMs collectively assign to a particular class.

Two important parameters requiring adjustment are the error term, C, and the kernel type. C controls the relative weighting between the dual goals of minimizing$||w||^2$ and of ensuring that most examples possess a functional margin at least 1.0. In other words, large values of C can result in a high variance and overfitting, while small values of C can lead to a high bias and underfitting. Following several adjustments, we concluded that $C = 0.1$ is the optimal value for our model. We selected a Gaussi-

an kernel since it constructs a hyper-plane in an infinite dimensional space and is reputed to be among the most powerful kernel functions.

Due to the large number of zero-length conversations, we observed that classification failed to operate correctly on the unbalanced dataset (i.e. "Incompatible" was always predicted). Resolution of this issue involved creating both a balanced training and testing dataset.

Although other classification algorithms we tested (Naïve Bayes, decision trees, and random forest) improve performance, they produced significantly worse prediction results.

4.1.4 Grid Search and K-fold Cross Validation

The kernel type K, the error term C and kernel coefficient γ, were three important SVM input parameters requiring adjustment. We selected a Gaussian kernel as our kernel function, since it constructs a hyper-plane in an infinite dimensional space and is reputed to be among the most powerful kernel functions available. As explained earlier, C balances between the dual goals of minimizing$||w||^2$ and of ensuring that most examples possess a functional margin at least 1.0. Larger values of γ correspond to worse balanced classification, which suggests that the deflective classifier model is causing lower accuracy on one side of the classification destination and higher accuracy on the opposite side.

As opposed to using the default values for C and γ, we searched for a (C, γ) pair that maximized accuracy, which was achieved by setting $C = 0.5$ and $\gamma = 0.1$. During this process, we applied a two-step grid-search method that tests every combination of possible C and γ values ($C = 10^{-5}, 10^{-4}, \dots, 10^{5}$, $\gamma = 10^{-10}, 10^{-9}, \dots, 10^{1}$) to find the optimal combination. After conducting a search with a coarse grid to locate a better region, the method constructs a finer grid within that region.

To evaluate the accuracy of each combination, we applied K-fold cross validation (where K = 3) on the training data. This was to avoid biasing our model on the test data.

4.2 Bipartite Graph Approach

Our initial network analysis showed that that the majority of same-gender conversations (i.e. both participants are of the same gender) is short. Fewer than 0.7% of such conversations exceed 5 lines. This observation motivates us to model Chatous as a bi-partite graph in which the set of male users and the set female users have good conversation edges between them, but not amongst themselves. With this formulation, we can approach the conversation length prediction task by always predicting that a conversation will be short (i.e. class 0) if the two users are of the same gender (as shown above, we will be wrong for only 0.7% of same-gender conversations). As for opposite-gender conversations, we can apply two prediction techniques - Collaborative Filtering and Singular Value Decomposition.

4.2.1 User-based Collaborative Filtering

We select Collaborative Filtering due to its high scalability, ease of implementation, and proven effectiveness in practice for the task of bi-partite link prediction [7], especially in the domain of online dating prediction [8]. We implemented a memory-based collaborative filtering system by following the approach described in [7] for our prediction task.

4.2.1.1 Model

We first construct a sparse matrix, $L \in \mathbb{R}^{f \times m}$ (where $f = 102{,}469$ and $m = 146{,}352$ are the number of female and male users, respectively), that contains all existing female-to-male user conversation length classes (where class 0 corresponds to 0 - 5 lines, class 1 corresponds to 6 - 20 lines, and class 2 corresponds to 20+ lines) using the conversation data in the training set. This matrix is extremely sparse, with only about 0.02% of the entries are non-empty. Considering that about 3% of Chatous users fail to specify their gender, we pre-process the data to resolve gender by inspecting conversation history (e.g. we will label a participant as "Female" if he/she has a greater number of long conversations with known male users than known female users).

Next, we construct a matrix, $W \in \mathbb{R}^{f \times f}$, of female-to-female user similarity scores. For each female user, u, we obtain a vector $\vec{u}$ that is representative of the user's conversation preference with male users by indexing into L and retrieving all of her past conversation lengths with male users from the u^{th} row. The i^{th} entry of the vector $\vec{u}$ is the length class of u's past conversation with the male user i, or empty if u has never chatted with i. If we let $\vec{v}$ be the conversation vector of another female user, v, we can subsequently calculate the similarity between u and every other v by computing the vector cosine similarity between $\vec{u}$ and $\vec{v}$, and populating the similarity score between the two female users into W. The vector cosine similarity is given by:

$$w_{u,v} = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|\|\vec{v}\|}$$

Following the same approach, we construct another matrix, $Z \in \mathbb{R}^{m \times m}$, of male-to-male user similarity scores.

To achieve system scalability, we implemented two optimizations. First, during the computation of W, after computing the similarity score between a female user u and every other female user, we only retain and store the top 20 similarity scores for u (and likewise during the computation of Z, we only store the top 100 similarity scores for each male user). This optimization drastically reduced the size of our data model (for which we implement using a dictionary of counters in Python) without having a noticeable impact to our prediction quality, since female users with lower similarity scores contribute negligibly in our "Weighted Sum of Others" approach. Second, we shard users into 100 shards by taking the mod of their ids, and compute the similarity matrix of each shard in parallel by running parallel jobs on Farmshare clusters. This parallelization successfully reduced our model computation time from 36 hours to 40 minutes.

4.2.1.2 Prediction

When predicting the length class of a conversation a female user, u, will have with a male user, i, we index into the female-to-female similarity matrix, W, to retrieve u's K most like-minded female users, ordered in descending order of their vector cosine similarity score to u. The predicted conversation length class is then the weighted sum of these K neighbors' conversation length classes with i, given by the following formula [4]:

$$L_{u,i} = \bar{l}_u + \frac{\sum_{v \in V}(l_{v,i} - \bar{l}_v) w_{u,v}}{\sum_{v \in V} |w_{u,v}|}$$

Where:

- V is the neighborhood set of u's K most like-minded female users.
- If u is a cold start female user (i.e. she has not chatted with anyone), we have no female-female similarity information for this user. In this case we use the average conversation length across all conversations in the training set.
- K is a hyper-parameter that we tune. Higher K value increases the likelihood of finding a like-minded female user who has chatted with i, but at the cost of significantly longer runtime.
- $w_{u,v}$ is the vector cosine similarity between female users u and v.
- $l_{v,i}$ is the length class of a conversation between female user v and male user i. We ignore all neighbor v that have not chatted with i.
- $\bar{l}_u$ and $\bar{l}_v$ are the complementary conversation length classes of female users u and v (i.e. their average conversation length class with male users other than i).
- If none of u's top K like-minded female users has chatted with the male user i, we base our prediction on u's past conversations with i's K most like-minded male users (J is the neighborhood set of these like-minded male users):

$$L_{u,i} = \bar{l}_u + \frac{\sum_{j \in J}(l_{u,j} - \bar{l}_j) w_{i,j}}{\sum_{j \in J} |w_{i,j}|}$$

- If u has not chatted with i's K most like-minded male users, we base our prediction on u's average conversation length class with all male users.

Intuitively, the above formulation takes female user u's average conversation length class with all male users, and adjust it by taking into account how like-minded female users prefer male user i compared to their own average conversation length class. This is a powerful formulation because rather than using user conversation lengths as if they all follow a single distribution, it recognizes the fact that some users are better conversationalists than others, and account for this fact by using the difference between a female user's incidental conversation length class (i.e. for a particular male user) and her average complementary conversation length class.

Note that in the formulation above we take a female-centric approach to Collaborative Filtering - to predict the conversation length class between a female user and a male user, we find like-minded female users and observe their conversation history with the male user. Alternatively, we can also take a male-centric approach. We have in fact also implemented a male-centric Collaborative Filtering system and shown its result in the Results section later.

4.2.2 Singular Value Decomposition

4.2.2.1 Model

We notice that our female-male conversation length class matrix, L, is extremely sparse, and only less than 0.1% of the entries are populated. This limits the efficiency of the Collaborative Filtering method, because the model can only apply the neighborhood-based prediction approach for a small fraction of conversations (and have to fall back to using user average for the rest). To overcome this issue, we apply Singular Value Decomposition to reduce matrix L into lower dimensionality. From a high-dimensionality sparse matrix, this factorization attempts to remove noise in the sparse matrix and discover the hidden correlations and latent features in low dimension space. The factorization is given by [5]:

$$\hat{L}_r = U \Sigma V^T$$

Where:

- $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix whose diagonal entries are the r largest eigenvalues of L sorted in descending order.
- $U \in \mathbb{R}^{f \times r}$ captures the response of each female user to the latent features. f is the number of female users.
- $V \in \mathbb{R}^{r \times m}$ captures the amount each latent feature is present in the male users. m is the number of male users.
- U_{ik} can be interpreted as female user i's reaction to latent feature k, Σ_{kk} as the importance of the feature k, and V_{jk} as the amount feature k is present in male user j.
- $\hat{L}_r$ is the rank-r low-dimensionality approximation of the original L matrix. Specifically, $\hat{L}_r$ is the best rank-r approximation of L in that it minimizes the Frobenius norm $\|L - \hat{L}_r\|$ over all rank- r matrices.

We first pre-process L to fill in the missing entries using the approach proposed by Sarwar et al [9]:

- Compute the average of each row $\bar{r}_i$ (i.e. the average conversation length class of a female user) and each column $\bar{c}_j$ of L.
- Replace all missing values with the corresponding column average $\bar{c}_j$. This produces a new dense matrix, L_C.
- Subtract the corresponding row average, $\bar{r}_i$, from L_C, and obtain the row-centered matrix L_R in which the row mean is 0.

After this pre-processing step, we feed L_R to Equation 5 to obtain $\hat{L}_r$, U, Σ, and V.

4.2.2.2 Prediction

Similar to the Collaborative Filtering model above, we predict the length class of a conversation to be 0 if the participants of the conversation are of the same gender. For opposite gender conversations, the predicted length class of a conversation between a female user u and a male user i is given by:

$$L_{u,i} = \bar{r}_u + \sum_{k=1}^{r} U_{uk} \Sigma_{kk} V_{ik}$$

Where the second term gives the (u, i) entry in the rank- r approximation matrix $\hat{L}_r$.

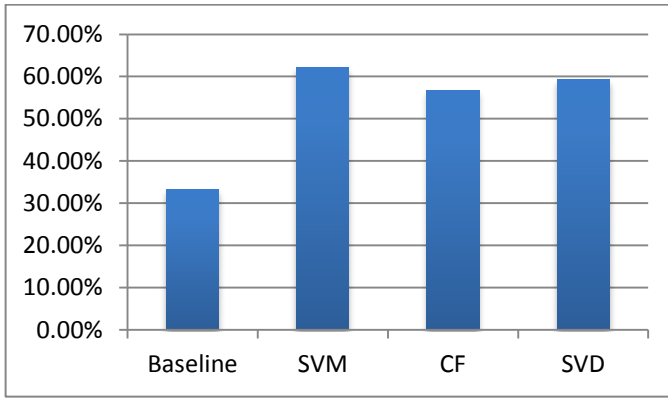
If the male user i is a cold start user, we predict using the female user u's average conversation length class. If the female user u is a cold start user, we predict using the average length class across all conversations in the training set.

5 Results and Analysis

5.1 Summary

We tested all of our models on the balanced test in which each conversation length class is represented equally. We choose this balanced test set so that our models will not be able to achieve high accuracy by simply predicting the label of the most representative class. The figure below shows the prediction accuracy of our three main models - Support Vector Machine, Collaborative Filtering and Singular Value Decomposition. For comparison purpose, we have included a baseline model that predicts each of the three conversation length class with equal probability.

Figure 5.1. Prediction accuracy comparison of the different models



On the balanced test set, we find that all three models perform quite well, and significantly better than the baseline model. Among the three models, the Support Vector Machine model produces the best result. This validates our hypothesis that the feature based classification approach, as the most principled way to combine user and network properties, would produce the best result.

5.2 Feature-Based Approach

Using the method previously outlined, we achieved a 62.2% predictive accuracy on the test data. This amounts to an 86.8% increase when compared to baseline random tri-class classification, which yields an expected accuracy of 33.3%. In the following sections, we offer a detailed explanation regarding the methods by which this high accuracy rate was achieved, along with the shortcomings of our model and likely future improvements.

Theoretically, test error should decrease as the SVM is provided with additional training examples; however, increasing the number of examples from 9000 to 30000 results in a significant increase in execution time with negligible improvement in predictive accuracy. As a result, our final training/testing run was conducted on a balanced dataset of 9,000 conversations (3000 conversations per class), while testing was performed on a total of 3,000 conversations (1000 conversations per class). Accuracy was computed by dividing the number of correct predictions by the number of conversations in the testing set.

5.2.1 The Effect of Scaling Features

Enormous improvement was achieved by scaling features--50.0% accuracy in 20.96 seconds became 62.2% accuracy in 10.37 seconds. This was due to the fact that scaling disallows attributes in greater numeric ranges (i.e. degree in the Chat–graph) from dominating values within a smaller numeric range (i.e. users' PageRank values). In fact, the introduction of scaling resulted in accelerated calculations and the avoidance of computational difficulties, such as overflow/underflow caused by the kernel values' dependence on the inner products of feature vectors.

5.2.2 Principal Component Analysis and Feature Selection

Principal component analysis was one technique applied to reduce the feature space. To identify the optimal number of features to incorporate, we evaluated our model's performance while varying the number of components retained. Our results appear below.

Figure 5.2.1. Prediction accuracy vs the number of components

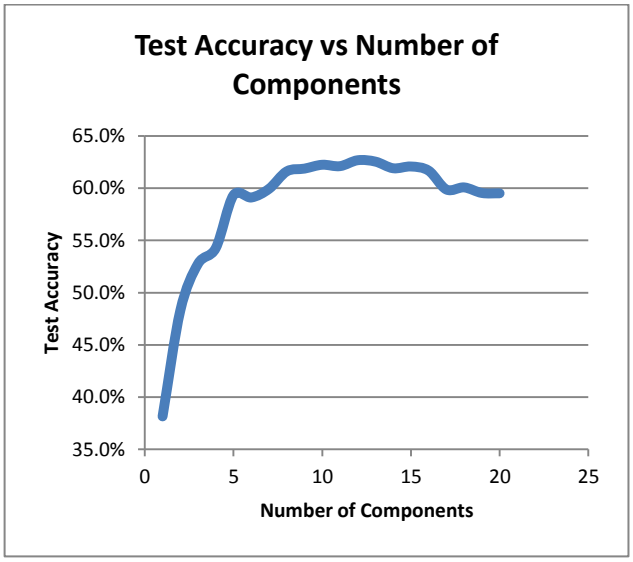
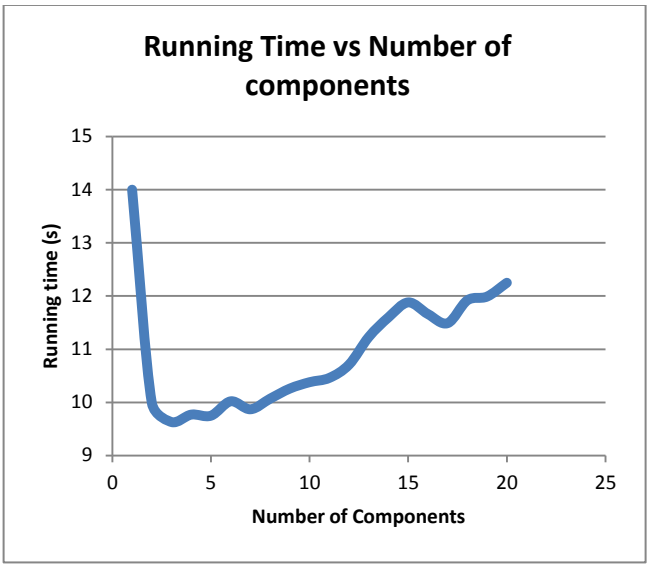


Figure 5.2.2. Model running vs the number of components



Initially, as the number of components increases, predictive accuracy dramatically improves. This result can be explained by the increased preservation of useful features. However, when the number of features retained exceeds 10, noise is subsequently introduced and we begin incorporating irrelevant features or those that overlap with existing features.

5.2.3 The Effect of Individual Features

Through the gradual introduction of additional features, both the accuracy and performance of our model improved. This result is illustrated in the table below.

Features	Accuracy	Performance (in seconds)
Age + Gender + Location	53.7%	14
All above + Basic Graph Properties	54.7%	13.57
All above + PageRank	62.0%	11.54
All above + Triad + Average Conversation Length	62.2%	10.37

Table 5.2.3. Prediction accuracy with different features

To more thoroughly understand how each feature affects predictive accuracy, we consider the weights assigned to each by the SVM. However, interpreting these SVM-assigned weights is particularly challenging, as weights are assigned a very high dimension. Additionally, PCA will alter the feature vector, so we cannot apply PCA if we want to know the weight for each feature. One option is to use SVM with a linear kernel, without PCA optimization, to obtain an approximate weight for each bit of the feature vector (73 bits in total).

While PageRank values provide significant improvements in predictive accuracy, basic graph properties (e.g. in/out degree, number of common neighbors, etc.) and triad structure provide minimal value. Unexpectedly, two bits in the "Age" feature are assigned greater significance than gender. Assigned a weight nearly equal to zero, we conclude that the "Location" feature has little relevance to predicting conversation length. We observed that bits from PageRank are assigned moderate weights. Since every node has a PageRank value, even small PageRanks are able to greatly impact the prediction. As a result, we consider PageRank one of our most important features.

Despite the fact that large weights are assigned to some triad structures, incorporating triad features fails to provide any significant improvement. In particular, large weights are given to triad structures involving two users who share a common neighbor with whom both have experienced an unpleasant interaction (i.e. a short conversation or report). It is quite possible that the failure of triads to significantly improve predictive accuracy stems from the lack of triads present in the small subset of interaction data.

Since the majority of participants on Chatous engage in zero-length conversations, we discovered that average conversation length is a poor predictor of user quality. This was contrary to our original assumption that average conversation length would be a useful metric in predicting which two users would be likely to engage in an extended conversation. Alternatively, we decided to count the number of long conversations (the out-degree in the Chat+ graph) to obtain the same information.

We also explored how accuracy is affected when matching new users versus established users. We define new users as nodes that were not present in the training set, while established users our program has seen previously. The accuracy for the set is 64.5% for established users. This number declines to 59.5% when at least one of the users being matched is a new user.

Finally, we observed our model's performance when provided with a set of users who are of the same gender compared to a set of opposite-gendered participants. The accuracy for the set of same gender is 86.1% while the accuracy for other set is 53.9%. Therefore, for future improvement, we should focus more on the accuracy for opposite gender conversations.

5.3 Bipartite Graph

5.3.1 Collaborative Filtering

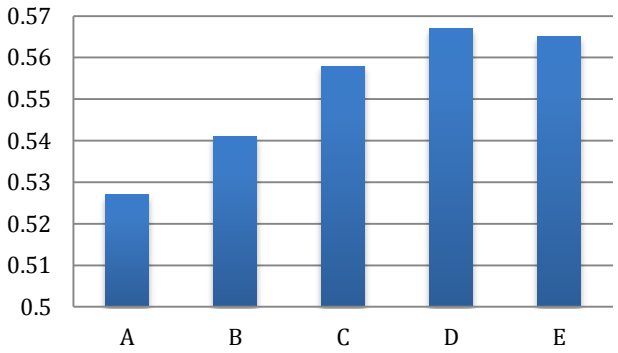
We trained the model on the complete training set of 6,462,054 conversations, and tested the model on a balanced test set of 122,097 examples. The test set has exactly 40,699 conversations of each conversation length class (i.e. {0, 1, 2}). We developed our model in the following iterative manner:

- Model A: Use female-similarity only. Use $K = 10$ most like-minded female users. Use the female user's average when none of the like-minded female users has chatted with the male user i.
- Model B: Incorporate male similarity. When none of the like-minded female users has chatted with the male user i, base prediction on female user u's conversation history with i's K most like-minded male users.
- Model C: Increase K to 20.
- Model D: Increase K to 100.

- Model E: Male-centric version of Model D. I.e. when predicting the length class of the conversation between female user u and male user i, first find i's K most like-minded male users and use the weighted average of their conversation length classes with u.

The prediction accuracies of the above models is shown below:

Figure 1: Prediction accuracy for different Collaborative Filtering models.



We notice that Model A suffers significantly from data sparsity problem as in less than 1% of opposite-gender conversations have any of u's top K like-minded female users chatted with i. Model B seeks to alleviate this problem somewhat by falling back to use u's past conversations with i's like-minded male users. This increases the percentage of opposite-gender test conversations for which we can apply neighborhood-based prediction (i.e. the coverage of the model) to 1.37%. Model C and Model D further increase the coverage of the model by enlarging the neighborhood set of u and i. With Model D, we can make neighborhood-based prediction in 3.5% of all test conversations. Model E shows that the male-centric approach to Collaborative Filtering produces results comparable to the female-centric approach. We expected this result due to the symmetric nature of female-male conversations.

The final model we selected is Model D, which produces an overall prediction accuracy of 0.567008. As shown in the table below, this model works fairly well for conversations for which we have neighborhood information (with a prediction accuracy exceeding 0.624), but performs poorly when we do not have neighborhood information and have to fall back to using u's average conversation length class.

Note that this model's prediction accuracy for same-gender conversations is 0.809, which is significantly lower than our empirical observation that 99.3% of same-gender conversations in the training set are short. This is due to the fact that we test all of our models on a balanced test set to reduce model bias, and as a result a disproportionately large number of medium/long same-gender conversations are pulled into this set.

Conversation type	Composition	Accuracy
u's neighbors have chatted with i	2.49%	0.643721
u's neighbors have not chatted with i but i's neighbors have chatted with u	1.01%	0.623953
Neither u's neighbors have chatted with i nor i's neighbors have chatted with u	37.0%	0.478665
u has not chatted with anyone	29.8%	0.427705
Same gender conversations	29.7%	0.808727

Table 1: Break-down of prediction accuracy of Collaborative Filtering Model D

5.3.2 Singular Value Decomposition

With Singular Value Decomposition we achieved an overall prediction accuracy of 0.592124 with $r = 10$. We ran our model training/testing with different r values, and found that the prediction accuracy of the model is little changed with $r > 10$. This is due to the fact that the 10 largest eigenvalues of L_R are significantly larger than the smaller eigenvalues (the largest three eigenvalues are 83594.55, 33601.56, 1526.17, respectively).

As expected, the SVD model produces higher prediction accuracy than the Collaborative Filtering model as it mitigates the data sparsity problem. As shown in the table below, the SVD model has a coverage of 33% as compared to Collaborative Filtering Model D's coverage of 3.5%. However, the SVD model still suffers from significant data sparsity problem, as 53.1% of opposite-gender conversations in the test set involve a new user for which we have no record in the low-rank approximation matrix $\hat{L}_r$, and therefore have to fall back to using u's average conversation length class. As a result, the SVD model lags the SVM model, as the SVM model offers significantly better prediction accuracy for such cold start conversations (with 59.5% accuracy).

Conversation type	Composition	Accuracy
Both female and male users are existing users	33.0%	0.559333
Either female or male user is a new user	37.3%	0.448665
Same gender conversations	29.7%	0.808727

Table 2: Break-down of prediction accuracy of Singular Value Decomposition model

6 Conclusion

In this report, we discussed our application of various network analysis and machine learning techniques to predict the quality of user interactions between Chatous users as measured by conversation length. We propose that user interaction quality can be accurately predicted using a combination of user properties and network structural properties, and demonstrate that these two types of properties can be combined to produce good results by applying three approaches: multi-class SVM classification, collaborative filtering, and singular value decomposition (the latter methods are based on modeling the network as a bipartite graph). All three techniques allowed us to accurately predict the conversation length metric with a high accuracy rate, 62.2%, 56.7%, and 59.2%, respectively.

7 References

[1] Backstrom, L., & Leskovec, J. (2011). Supervised Random Walks: Predicting and Recommending Links in Social Networks. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 635-644. Retrieved from http://cs.stanford.edu/people/jure/pubs/linkpred-wsdm11.pdf

[2] Guo, K., Bhakta, P., Narayen, S., & Loke, Z. K. (2012). *Predicting Human Compatibility in Online Chat Networks*. Unpublished manuscript, Department of Computer Science, Stanford University, Stanford, California.

[3] Guha, R., Kumar, R., Raghavan, P., & Tomkins, A. (2004). Propagation of Trust and Distrust. *Proceedings of the 13th international conference on World Wide Web Pages*, 403-412. doi:10.1145/988672.988727

[4] Leskovec. J., Huttenlocher, D., & Kleinberg, J. (2010). Predicting Positive and Negative Links in Online Social Networks. *Proceedings of the 19th international conference on World wide web*, 641-650. doi:10.1145/1772690.1772756

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2001). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830. Retrieved from http://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[6] Statistic Brain. Social Networking Statistics. Retrieved from http://www.statisticbrain.com/social-networking-statistics/

[7] Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*. doi:10.1155/2009/421425

[8] Wang, T., Liu, H., He, J., Jiang, X., & Du, X. (2011). Predicting New User's Behavior in Online Dating Systems. *Proceedings of the 7th international conference on Advanced Data Mining and Applications*, *2*, 266-277. doi:10.1007/978-3-642-25856-5_20

[9] Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2000). Application of Dimensionality Reduction in Recommender System -- A Case Study. *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 82-90. Retrieved from http://robotics.stanford.edu/~ronnyk/WEBKDD2000/papers/sarwar.pdf