Predicting Edge Properties in a Bipartite Network

Jessie Duan, Sean Scott, Yongxing Deng (Group 17)

I. Introduction

Chatous is a text-based, one-on-one anonymous chat network begun in October 2012. In this project, we will examine a data set consisting of 333,000 users and 9 million conversations between these users over a period of 2 weeks. Users are paired randomly for conversations, may leave the conversation at any time, and may carry on multiple conversations simultaneously.

This random pairing creates a *de novo* online social network (i.e. not based on a real world social network), and provides the opportunity to examine conversation compatibility between any two users. We observe, however, that there are multiple differences between same-gender and opposite-gender conversations. Namely, same-gender conversations are much more likely to have 0 lines, signifying that a user left a conversation based on profile data only, which includes gender and interests. Thus, the issue of conversation compatibility can be divided into two parts: single-gender conversations and opposite-gender conversations.

In this project, we attempt to predict male-female conversation compatibility on the *Chatous* network by modeling the network as a bipartite graph based on gender. We hope that focusing on male-female conversations will give us insights that allow us to address a major subproblem of conversation compatibility.

I. Prior Work

The *Chatous* platform, and more broadly, random online social networks, have not received much attention. Guo et al. is the first and thus far only paper that examines the *Chatous* network, and examines scoring metrics for measuring conversation quality as well as methods of predicting compatibility. Although it was not able to find a compatibility algorithm that performed significantly above average, it did determine that conversation length is a useful predictor of conversation compatibility.[3]

To predict unseen links based on existing edges in the network, Liben-Nowell et al. examines edges between a pair of nodes' neighbors as well as nodes incident to edges in both the training set and test set; Leskovec et al. uses triads, node metadata, and status and balance theories.[1][2]

Although Guo et al. attempted sign prediction on triads and did not find a successful algorithm, it is possible that the inclusion of both same-gender and opposite-gender conversations clouded the data. While Guo et al. made an excellent start in broadly examining *Chatous* data, their algorithms examined all conversations generally; we attempt to separate the conversation data into subsets and apply prediction algorithms to male-female conversations only.[3]

II. The Model

I. Assumptions

In order to build a mathematical model on the *Chatous* network, we make four assumptions about the network.

i. Similarity

The first assumption we have about *Chatous* is that similar users like to talk to similar sets of users. This is to be distinguished from "similar users like to talk to each other." In other words, if user x and y are intrinsically similar, then if user x enjoys talking to user a, then user y will enjoy talking to a as well. Conversely, if x does not enjoy talking to b, then y will not either. This is important to making predictions, because we will base some of our models on how similar users interact with other groups of users.

ii. Modeling the *Chatous* network as a bipartite graph

We observe that conversations on *Chatous* network are mostly between males and females. As a result, we filter out all conversations between two users of the same gender. We hope this way some features of the graph will be clearer.

Gender is also self-reported, so we will assume that users are reporting the correct gender. Users that do not report any gender, as well as their conversations, are removed.

iii. Conversation length represents conversation quality, and in general, compatibility between two users

There are other measures that could measure conversation quality, such as users "friending" each other or engaging in a video chat. However, that data may be sparse, since two users can have a good conversation without doing either of the above. Further, the concept of "quality" is somewhat unclear, since different users may have different goals in their use of *Chatous*. Some may try to find a date, while others may just be looking for people with similar interests, such as music, online games, or sports. However, *Chatous* found that scoring based on user ratings performed more poorly than conversation length, so we will use length as an indicator of quality.

iv. Profiles represent users

The interface of *Chatous* allows users to change any parts of their profile after they have started using the site. Therefore one person may have several profiles associated with them, during different points in time. They may also have several logins, in which case they are represented by several "user" designations on the site. However, for our purposes, conversations happen between two profiles, since there is no metadata associated with users (or real people, for that matter). In other words, we will predict conversation lengths between two profiles, at the risk of losing some data by not considering the evolution of profiles for one user (or different users for one person). This is reasonable both because it makes more sense from a data perspective, but also because profile changes may affect a conversation partner's willingness to begin a conversation (especially gender and age).

II. Data Collection

Data was provided by *Chatous* in a CSV format. We have 9.05 million conversations and 332,887 profiles. Conversations include profile IDs, length in lines, who disconnected, a conversation word vector, and any abuse reports, among other fields. Profiles represent different iterations of a user's own description, including age, gender, screenname, and a description. Note that any user-entered words (including screennames, descriptions, and conversations themselves) were replaced by integers to protect users' privacy.

In this project, we will examine a subset of this data, 104,629 conversations and 7,915 profiles. This subset was obtained by selecting all conversations and involved profiles that occurred in a time period of 5.5 hours, to maintain a similar density to the already-sparse original graph. We note that in some instances, a pairing had two separate conversations or a profile was duplicated; in the interest of simplification, we use the first conversation and first profile that occurred.

III. Scoring Framework

We attempt to solve a real problem *Chatous* faces - given a list of users in a queue who are waiting to chat with a random stranger, how do we quickly match the users up such that they are more likely to enjoy their conversations with their partners? In order to create an accurate matching, we need to be able to predict edge weights well. Based on our assumption above, edge weights represent the conversation length in lines. As described in the findings section below, over three quarters of all conversations end without a word being said, so being able to make a binary prediction on whether the conversation will be successful or not would be an accomplishment in itself. We deem 10 lines to be a "successful" conversation, as that allows two users to interact and engage with one another. However, in order to provide flexibility in pairing, we also attempt to predict actual conversation length.

III. Initial Findings

In order to design a fair metric to compare our model to baseline models, we need to first understand the dataset. In particular, we want to understand how lengths of conversations are distributed:

Conversation length	Appearances	Percentage
0	79933	76.397
1	9785	9.3521
2	3501	3.3461
3	2281	2.1801
4	1500	1.4336
5	1173	1.1211
6	844	0.80666
7	719	0.68719
8	584	0.55816
9	495	0.4731
≥ 10	3814	3.6453

As one can see, more than three quarters of the conversations initiated by the application were terminated before either person talks to the other, and only 3.6% of the conversations had a meaningful length (≥ 10).

Next, we verified our assumption that male-female conversations were longer than male-male or female-female conversations by breaking down the number of conversations of length 0 (i.e. conversations where one user disconnects immediately after observing the other user's profile information, which includes gender). Using that same random sample of conversations, we observed the following rates of immediate disconnects:

Genders in conversation	Percentage of immediate disconnects
Male-Male	97.2%
Female-Female	86.4%
Male-Female	45.2%
Male-Unspecified	86.8%
Female-Unspecified	57.2%
Unspecified-Unspecified	67.4%

As we can see, although the rate of immediate disconnects is high in all conversations, conversations where males and females are paired has the lowest rate of immediate disconnects - indeed, less than half the immediate disconnect rate of male-male conversations. This confirms our assumption that users are more likely to converse with users of the opposite gender, and that the *Chatous* network can be modeled as a bipartite graph.

I. Original vs. Bipartite Graph Comparison

To further validate our decision to make the graph bipartite, we observe specific graph properties both before and after bipartition to note that key properties do not change drastically. The latter 7 properties in the below table are of a graph where all conversations of length 0 have been removed.

Property	Original Graph	Bipartite Graph
Total Nodes (Users)	7878	6886
Total Edges (Conversations)	92199	33408
Nodes that have had Non-Zero Length Conversations	6828	6287
Non-Zero Length Conversations	23081	19014
Average Degree	3.380	3.024
Longest Diameter	12	13
Number of Connected Components	65	64
Average Clustering	0.0344	0.0
Density	0.0009903	0.0009622

Looking at the number of nodes and edges, it seems at first that we are removing almost 66% of edges through bipartition. However, we can see that most of these removed conversations are immediate disconnects, fitting our observation that most single-gender conversations are immediate disconnects. When examining only conversations of length greater than 0, which give us more information about compatibility based on the conversation itself, we see that we are only removing 7.9% of users who have had at least one non-zero-length conversation and 17.6% of non-zero-length conversations. The average degree, when only considering non-zero conversations, only decreases by .376. Other characteristics, such as the longest diameter and the number of connected components, stay similar; note that clustering is already very low in the original graph, and becomes 0.0 in the bipartite graph because it is impossible for a node's neighbors to be neighbors with each other in a bipartite graph.

Examining the degree distribution of the original and bipartite graph also yields similar shapes and frequencies:

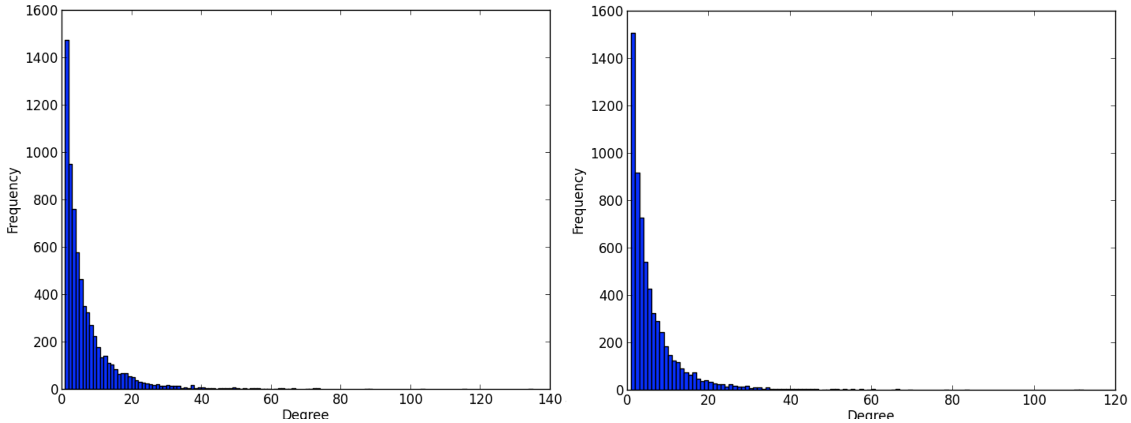


Figure 1: *Left: Degree distribution of original graph; Right: Degree distribution of bipartite graph*

Thus, we can conclude that making the graph bipartite does not significantly change the graph structure. We can then continue our analysis of the bipartite graph with the assumption that it reflects properties of the original graph.

II. Models

However, since the distribution of conversation lengths is skewed towards 0, many commonly used evaluation metrics fail to represent the accuracy of a model. For example, if we use mean squared error as our evaluation metric, then the following algorithm

```
def predict(user1, user2):
  return 0
```

will likely achieve a much "better" result than most models. In general, the dataset will penalize a model for over-estimating much more than under-estimating.

Indeed, attempts at using Weka, a machine learning tool, to predict pairings yields exactly this. Both logistic regression and multilayer perceptron methods, after training on ages, genders, and profile IDs, predict all conversations to have length less than 3. Since 3 is the absolute minimum number of lines for there to be any sort of interaction and participation of both users, we must use more refined methods to predict compatibility.

Instead, we currently use a diagram to denote the prediction of a model. For each testing example, we plot a point $(x + \epsilon_1, y + \epsilon_2)$ on the graph where x is the actual conversation length, y is the predicted conversation length, and both ϵ_1 and ϵ_2 are small random variables so that a same length/prediction pair will not stack on one another (and as a result become invisible). A point on the line $y = x$ denotes an accurate prediction. The farther away a point is from the line $y = x$, the less accurate the prediction is.

Model 1. Random. This model records all the conversation lengths. To predict a conversation, it randomly samples one length from the distribution.

```
def train(profiles, convos):
  return convos
```

```
def predict(user1, user2, convos):
  return random.choice(convos).length
```

Model 2. Global average. This model considers no information about a user, and just predicts the length of a conversation between two users based the known global average.

```
def train(convos):
  return average(convos)
def predict(user1, user2, average):
  return average
```

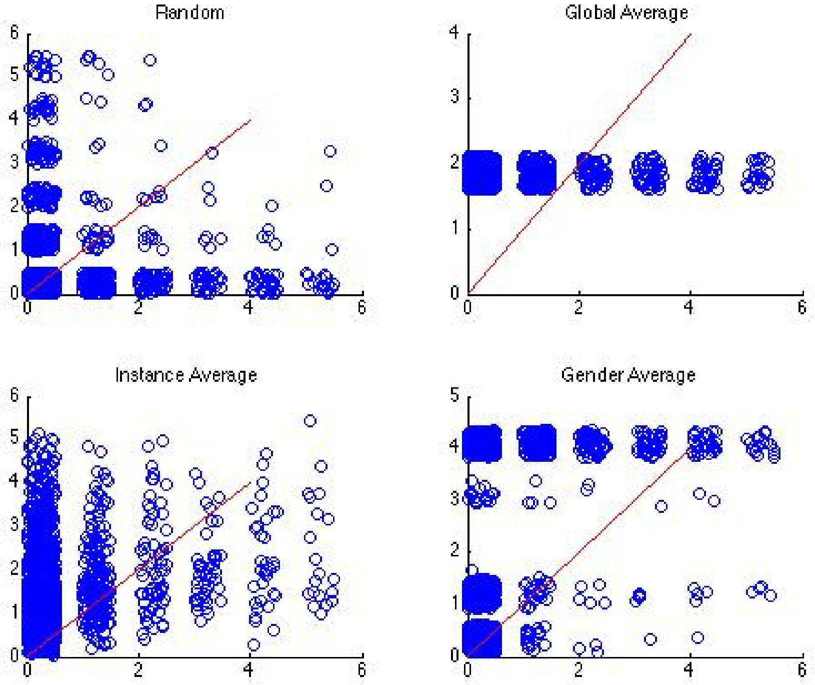
As we can see, the model perform poorly on the dataset.

Model 3. Instance average. This model computes the average conversation length for each user. To predict the length of a conversation, use the average of the averages of the two users.

```
def train(convos):
  average_dict = dict()
  for user in users:
     average_dict[user] = compute_average(convos, user)
def predict(user1, user2, average_dict):
  return (average_dict[user1] + average_dict[user2]) / 2
```

Model 4. Gender-specific average. This model considers all possible pairs between the three gender categories: **male**, **female**, and **unspecified**. For each pair, compute the average conversation length, and use that to predict the length of an unknown conversation.

```
def train(convos):
  for (g1, g2) in combinations([Male, Female, Unspecified], 2):
    average_dict[(g1, g2)] = compute_average(g1, g2, convos)
def predict(user1, user2, average):
  return average_dict[(user1.gender, user2.gender)]
```



From the four basic models that involves no learning process, one can see that the gender average model captures at least some information about the data. For example, when the conversation length is zero, the gender average model is likely to predict zero (77%); when the conversation length is non-zero, the algorithm is likely to predict non-zero values (79%). This verifies our assumption that gender plays a significant role in conversation lengths so we should model the graph as a bipartite graph.

IV. Modeling Bipartite Graph as a Recommender System

After the graph is converted into a bipartite graph, we need to estimate values of unknown edges, using values of known edges. Since the graph is bipartite, we can use a matrix to represent the graph.

	Male 1	Male 2	Male 3	...	Male m
Female 1	1			...	
Female 2		3	?	...	4
Female 3	2		5	...	
...	...	...	...	...	...
Female n	10	3		...	

Figure 2: *Another way to represent the bipartite graph*

This representation reminds us that the problem can be considered to be the well-known model of recommender systems.[5]

I. Collaborative Filtering

A recommender system is a setting with many users and items, and users are asked to rate items. The goal is to used the known ratings to predict unknown items, and to recommend to each user items that they will rate highly. The intuition behind the model is that in order to predict a given user's rating on an item, we can use the ratings on this item given by other similar users. More technically, let r_{xi} be the rating that user x gives item i. Then we estimate an unknown r_{xi} to be

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

Here, $s_{ij} \in [0, 1]$ is the similarity score computed between user i and user j, where j is selected from $N(i; x)$, the set of users rated by user x most similar to user i. The more similiar a pair of users is, the higher their similarity score is (and r_{xj} is weighted higher as a result). Some similarity measures include Jaccard similarity, cosine similarity, and Pearson correlation coefficient.

II. Applying collaborative filtering to the bipartite graph

We can apply similar techniques to the converted bipartite. We first consider female users to be the "users" and male users to be "items," and then reverse the roles. In each scenario we use lengths of conversations (or edge weights) to be the "ratings", and use collaborative filtering to predict the length of the conversation. [1] We then use the arithmetic mean of the two as the final prediction of the length of an unknown edge.

III. Similarity score

What is the best way to indicate how similar two users (of the same gender) are? Since the graph itself is sparse, computing similarity based on the graph itself, though possible, will not give accurate results. Instead, we use user profiles given in the dataset. We divide the similarity score into three parts.

[1] If there is a new user in the conversation, then there will no edges from that node. As a result, both the numerator and the denominator of the equation will be zero. In this case, we simply use the global average of conversation length as the our prediction.

1. Location Similarity score: Sim_{loc}. Using the location data of user profiles, and the following table:

Description	Sim_{loc}
Same country	1
Same region (e.g. South-east Asia)	.8
Same continent	.6
Same language	.3
Other	.1

If multiple rows are satisfied, use the first one among them.

2. Age Similarity score: Sim_{age}. If the age of the two users are x and y, then age similarity score is:

$$\text{Sim}_{\text{age}}(x, y) = \max(1 - |\log_2 \frac{x}{y}|, 0.1)$$

Notice that this definition guarantees that the similarity score is symmetric, that $\text{Sim}_{\text{age}}(x, y) = \text{Sim}_{\text{age}}(y, x)$.

3. "About" (self-description) Similarity score: $\text{Sim}_{\text{about}}$. Let A and B be the set of words used in two users' self descriptions. About similarity score is the Jaccard similarity between the two users:

$$\text{Sim}_{\text{about}}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The final similarity score is the weighted average of the three, with weights determined empirically:

$$s_{ij} = .3 \cdot \text{Sim}_{\text{loc}} + .3 \cdot \text{Sim}_{\text{age}} + .4 \cdot \text{Sim}_{\text{about}}$$

IV. Results

The following is the results of Instance Average (our baseline) and Recommender Systems.

Converted bipartite graph

	$L \geq 10$	$L < 10$
$L_{\text{predict}} \geq 10$	.012	.031
$L_{\text{predict}} < 10$	.067	.890

Instance Average

	$L \geq 10$	$L < 10$
$L_{\text{predict}} \geq 10$	.035	.136
$L_{\text{predict}} < 10$	.043	.795

Collaborative filtering

Original graph[2]

	$L \geq 10$	$L < 10$
$L_{\text{predict}} \geq 10$	.00175	.01050
$L_{\text{predict}} < 10$	.02858	.95917

Gender-based Instance Average

	$L \geq 10$	$L < 10$
$L_{\text{predict}} \geq 10$	.01128	.04530
$L_{\text{predict}} < 10$	.01906	.92436

Collaborative filtering

If we use "Length $\geq$ 10" as an indicator of a meaningful conversation, in the converted bipartite graph, collaborative filtering produces a result of 20% precision (compared to 27%) and 44% recall (compared to 15%). In other words, collaborative filtering can capture many more meaningful conversations without losing too much precision.

In the original graph, as the baseline, we compute each user's average conversation given different genders. Collaborative filtering, which simply returns 0 for a same-sex edge, still produces a result of the same precision[3] (compared to 14%) and 37% recall (compared to 6%). It out-performs our baseline in both precision and recall.

[2] Collaborative filtering cannot predict an edge between two users of the same gender. It simply returns 0 in that case. As one can see, it still produces a decent result.

[3] because the set of positive predictions is the same.

V. Conclusions

Our analysis consistently shows that bipartition and gender-based methods improve conversation length prediction.

Our examination of the dataset reveals a sparse graph, in which most edges consist of immediate disconnects rather than real conversations. Examining conversations by gender, male-female conversations have by far the lowest percentage of immediate disconnects, confirming our assumption that the network can be modeled as a bipartite graph by gender. Bipartition then removes 66% of all conversations but only 17.6% of non-zero-length conversations. We also showed that the resulting bipartite graph maintains a very similar structure to the original in terms of density and node degrees; thus, findings on the bipartite graph will relate easily to the original network.

Next, we found that basic models without learning perform poorly. Although the gender average captures the most information, correctly predicting 77% of zero-length conversations and 79% of non-zero-length conversations, none of our 4 basic models does well in predicting longer conversations.

However, we have seen that a recommender system holds promise with respect to a baseline gender model, giving better recall with only a slight decrease in precision on the bipartite graph, translating into better recall and equal precision on the original graph.

VI. Further Work

In future work, we would like to use other properties of the graph. Although we took advantage of the bipartite structure of the graph, there may be promise in other aspects of the graph structure if we can work past its sparseness. One way of making the graph less sparse may be to focus on a subset of users and all of their conversations, rather than only examining conversations within a time period. We chose to look at a single time period to best reflect the original dataset; however, density may be increased by creating a subset of conversations based on users, at the possible expense of the subgraph's similarity to the original.

To further explore recommender systems, we may also wish to explore other similarity mechanisms, such as previous conversation content. We would also like to refine a method of determining weights in the similarity score,

Finally, to be able to deploy such a matching algorithm in real time, we may wish to determine how this male-female matching relates to the general matching problem. Although we showed that success on the bipartite graph translated into even better success on the original graph, we may be able to perform even better. For example, some users are equally compatible with same-sex users as with opposite-sex users. It may be useful to detect how likely a user is to wish to speak only with opposite-gender users, to avoid over-prediction based on gender.

References

[1] J. Leskovec, D. Huttenlocher, J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. In Proc. WWW, 2010.

[2] D. Liben-Nowell, J. Kleinberg. The Link Prediction Problem for Social Networks. Proc. CIKM, 2003.

[3] K. Guo, P. Bhakta, S. Narayen, Z. Loke. Predicting Human Compatibility in Online Chat Networks 2012

[4] A. Rajaraman, J. Ullman. Mining of Massive Datasets 2013

[5] J. Leskovec. Recommender Systems: Content-based Systems & Collaborative Filtering cs246.stanford.edu. 2013.