Optimizing Conversations in Chatous's Random Chat Network

Alex Eckert (aeckert)
Kasey Le (kaseyle)
Group 57

December 11, 2013

Introduction

Social networks have introduced a completely new medium for communication and the incredible amounts of resulting data make it possible to understand more about human interaction than previously possible. While these datasets are exciting to social scientists and computer scientists alike, online interactions are quite unnatural compared to how people interact face-to-face. On Facebook, information about each user is publically available, resulting in "Facebook Stalking" and little back and forth communication. On Twitter, users have followers, an inherently directed relationship that results in a natural imbalance between users. Since real communcation is not a one-way interaction, our group wanted to use a dataset that better mirrors how humans naturally interact with one-another and develop relationships. The Chatous dataset is a great opportunity to achieve just that.

Chatous is a company, started last year in CS 224W, that randomly initiates chats between pairs of users. Very little information is provided about each user so information discovery and relationship development happens organically. Within this context, we hope to predict whether a random pair of users will have a positve or negative conversation using only the graph of chat histories and the limited information we have on users.

Prior Work

We began our research studying signed networks because they provide a useful framework to structure and understand relationships between users within social networks with less robust and intricate interactions. We read "Predicting Positive and Negative Links in Online Social Networks" by Leskovec, Huttenlocher, and Kleinberg and "Low Rank Modeling of Signed Networks" by Hsieh, Chiang and Dhillon. Both papers discuss strategies for signed edge prediction, as well as the social-science theory behind the models.

Leskovec et. al

The Leskovec study analyzed three social networks with clear positive and negative relationships directly reported by the users. Edges within the network were designated as positive, negative, or neutral and were directed. Features were determined locally for each unknown edge. To predict the sign of an edge(u, v), two classes of features were used. The first class of features primarily looks at the individual properties of each node separately with only one feature considering any relationship between the two nodes. The second class of features analyzes the makeup of the triads containing both nodes.

The study sought to accurately predict positive and negative relationships between nodes and determine whether the social-psychological theories of balance and status explained these relationships.

Hsieh et. al

The second paper talks about another way to model signed networks in order to predict missing edges, as well as find clusters in the graph. They build upon the idea of "weak balance" in signed networks and show the effectiveness of modeling the graph as a low-rank matrix.

It defines a complete "weakly-balanced" network as:

> A complete signed network is weakly balanced if all edges are positive, or the vertices can be divided into several groups such that within-group edges are positive and between-group edges are negative.

and a k-weakly balanced network is when there are exactly k such groups.

Finally, the paper shows that the adjacency matrix of a k-weakly balanced network has low-rank, so modeling a graph as a low-rank matrix is essentially modeling it as a k-weakly balanced network.

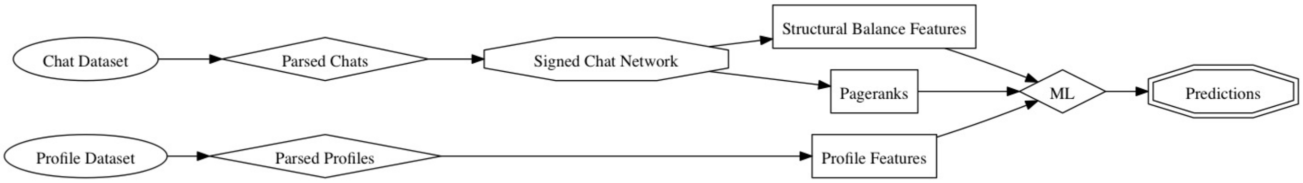
The paper argues that the most effective, and scalable, way of predicting signed edges is to create a k-weakly balanced complete graph while minimizing k. The paper tests this approach against a number of real-world datasets and it performs extremely well, even better than the Leskovec study from two years prior.

Objective

Our ultimate goal is to develop a prediction algorithm that takes in any two users and predicts if they will have a positive or negative conversation based on their prior chat histories and user profiles. This algorithm can be used by the Chatous matching algorithm to increase the number of positive chat experiences.

Framework

To accomplish this we created a data-pipeline that allowed us to easily iterate on our algorithm as we incorporated more advanced techniques. Here is a visualization of the pipeline:



We chose this particular organization because it isolates the development of our algorithm in one stage (marked as ML above), which allowed us to easily iterate on our prediction algorithm and utilize more advanced techniques.

Data Sets

Chatous provides two datasets — **chats** and **profiles**. The schemas are as follows:

Chats

chat id	user1 id	user2 id	user1 profile id	user2 profile id
start time	end time	who disconnected	reported id	reason for report
user1 # lines	user2 # lines	user1 word map	user2 word map	friendship

Profiles

id	location	location flag	age	gender	subscription date	about	screenname

Generating a Signed Network

After parsing the data, we created a signed network by assigning each chat a value from -1 to 1. We calculated this score using several criteria. First, if a friendship was established during the conversation, the chat recieved a very positive score of 1. If either user was reported during the conversation, the chat got a very negative score of -1. If these clear signals were not present, then we relied on the conversation duration and length. The duration is the time elapsed in seconds from when the chat was initialzed until a user disconnected. The length is the total number of messages sent during the conversation.

In past analysis of the Chatous dataset, conversation length was used as the key metric to determine the quality of a chat. However, we wanted to also incorporate clear signals (friendship/reporting) and a more robust usage of length and duration in our calculation.

We normalized the chat length and durations between -1 and 1 using summary statistics for chat length and duration. We used the whole dataset to determine the mean, the standard deviation, and the max values for chat length and for chat duration. We computed the following values.

	Mean	Standard Deviation	Max
Chat Length	2.968	12.865	143.0
Chat Duration	1156.277	8076.967	84950.0

Finally, we computed the final sign of each edge by adding up the individual chat scores for each edge and taking the sign of the total. Using this method, we labeled roughly 4% of all edges as positive and the remaining 96% as negative. We then added a third bucket of neutral conversations. We decided to count chat lengths of zero as indifferent. This change resulted in total edges signs of 4% positive, 20% negative, and 76% neutral.

Logistic Regression

We began by using Logistic Regression as our main prediction algorithm. To use a binary classifier, we reframed the problem as predicting positive chats so a positive chat was a 1 and a negative chat was a 0. Most of our work involved generating a robust feature set based on the chat network. In our model we used three main groups of features. The first used the structure of the signed network as was done in the Leskovec et. al study. Then, we applied the PageRank algorithm separately to the positive, negative, and neutral networks of chats to produce our second set of features. The final feature set takes in account profile information.

Signed Network

Following the model used in Leskovec et. al, the first feature set actually has two groups of features. The first concerns the signs of the known edges. For a node u and v where we wish to determine the sign of the edge (u, v), the first four features are the number of positive and negative edges for each node: $d^+(u)$, $d^-(u)$, $d^+(v)$, and $d^-(v)$. The fifth feature is $C(u, v)$, the number of common neighbors shared by u and v. Finally, the last two feautres are the total degree of each node: $d(u)$ and $d(v)$. The second feature set has four features. This set considers all the traids that contain both u and v. Suppose the third node in a triad is node x. The edge (u, x) can be positive or negative and the edge (v, x) can be positive or negative. This results in four different possibilities of triads. Thus, the second feature set is a vector of four values which are the counts of each of these types of traids that u and v are a part of. Combining the two groups, we have a total of 11 features.

PageRank

We also wanted to incorporate a notion of good and bad users into our learning algorithm. One option would be to simply plug in a feature like the percent of a user's chats that are positive, but that does not encapsulate the cooperative nature of chats. If a chat goes badly, the negative chat is not always a reflection on both users. If one of the users is particularly malicious, for instance, it is not the other user's fault and he or she should not be penalized (as much) for this chat. To that end, we explored using a modified version of PageRank, which computes a node centrality measure, to represent how good and bad users are. PageRank outputs an ordering of nodes by importance, gathered from the structure of the network. The question then became, how do we repurpose PageRank's notion of importance to instead represent user quality in the Chatous dataset.

We first split up the chats into three subgraphs, where each subgraph was comprised of only edges (chats) of a certain type. Thus, there was one graph comprised of only positive chats, one comprised solely of negative chats and one comprised of solely neutral chats. The stats about these graphs are as follows:

Graph	Nodes	Edges	Max Deg	Avg. Deg	Cluster Coeff
Positive	56901	2.8×10^5	499	9.83	4×10^{-3}
Negative	82463	1.4×10^6	3012	34.61	6.5×10^{-2}
Neutral	88070	5.3×10^6	4678	122.2	0.31

We then ran the PageRank algorithm on each of the three graphs with the following results:

Graph	Max PageRank	Avg. PageRank
Positive	6.6×10^{-4}	1.8×10^{-5}
Negative	8×10^{-4}	1.2×10^{-5}
Neutral	3.6×10^{-4}	1.1×10^{-5}

For each node, we now have three values that represent how central the node is to the graph of positive chats, the graph of negative chats and the graph of neutral chats, respectively. We can use all of these values in our learning algorithm as features, so for a given chat we now have six additional features comprised of the three PageRank scores for the first user and the three PageRank scores for the second user. These features turned out to be very important towards the success of our learning algorithms.

Profiles

Any matchmaking algorithm must also take into account features of the users, so, given user u and user v, we extracted a number of features for use in our learning algorithm.

[1] Both users female
[2] Both users male
[3] Users are opposite genders
[4] Age difference between users
[5] Age of u

[6] Age of v
[7] Age differerence between u and v
[8] How long has u been a member
[9] How long has v been a member
[10] Cosine similarity between the about sections for u and v
[11] Cosine similarity between the screen names for u and v

For features 10 and 11, we ignored all stop words, which were given to us along with the Chatous dataset.

Incorporating Neutral Conversations

In the first run of our algorithm, we decided to ignore neutral conversations and removed all neutral edges from the network. Since neutral conversations had a length and duration of zero, we decided to treat neutrals as if there had been no conversation at all. However, we were throwing away a lot of data this way since 76% of conversations were neutral. Thus, we decided to try putting them back in and marking them as negative conversations. This makes sense because a user still had to actively end the conversation. In fact, because they ended the conversation immediately, that may be an even stronger indication of a negative conversation. This means that the user chose to end the conversation based only on the profile of the other user so these actions provide insight into the user's chat preferences. In order to reincorporate all these neutral conversations, we had to better optimize our code since we were now working with fives times the number edges.

Feature Selection

After building out a robust feature set, we decided to perform a backwards feature selection on our model to identify the most predictive feature set. We did this by building a wrapper around our model that would delete one feature at a time, run the algorithm on the reduced feature set, and then remove the worst performing feature if any of the reduced feature sets performed better than the original. We decided to use Matthews Correlation Coefficient as our measure of performance since we were performing binary classifications. Additionally, given that our data is skewed with a small fraction of positive examples, we wanted to put less emphasis on accuracy and instead use a balanced measure of precision and recall. The coefficient returns a value between -1 and +1 with +1 indicating a perfect prediction, 0 indicating a random prediction, and -1 indicating a completely opposite prediction.

Feature selection removed both the neutral PageRank features as well as the age of the second user. Since we applied feature selection before incorporating the neutral edges described above, it makes sense that these PageRanks only confused the model. This result actually motivated us to reincorporate the neutral edges. After running feature selection, we realized that including the second age was redundant since we also included the age difference as a feature.

Naive Bayes

After having only mild success using logistic regression, we decided to try other prediction algorithms. We used an SVM with a polynomial kernal with degrees from 1 through 4 but they all performed worse than logsitc regression and were computationally expensive since our features matrix is fairly dense. We then used Naive Bayes. To do this, we had to discretize our feature matrix. For our PageRank features, we decided to order all the PageRank values and use the quintile as the feature instead of the actual PageRank. For the cosine similaries of profiles, we noticed that most similarities were zero, so we decided to make that feature binary and used 1 if the cosine similarity was greater than 0 and used 0 otherwise. Because Naive Bayes has stricter requirements for features, we also had to clean the data a bit and take care to remove all negative values.

Feature Selection

After a good first run, we decided to apply feature selection on our Naive Bayes model as well. It ended up removing several features. It removed three of the four features that counted the number of different triads which suggests that the theories of structural balance do not hold in the Chatous network. This may be because relationships between users are not visible to other users since the chats are random. This is different from the networks in the Leskovec et. al study where positive and negative ratings could be observed by all users.

As before, running feature selection also removed redundant features like the second age and the third category for gender (since only one of the gender features could be positive anyways).

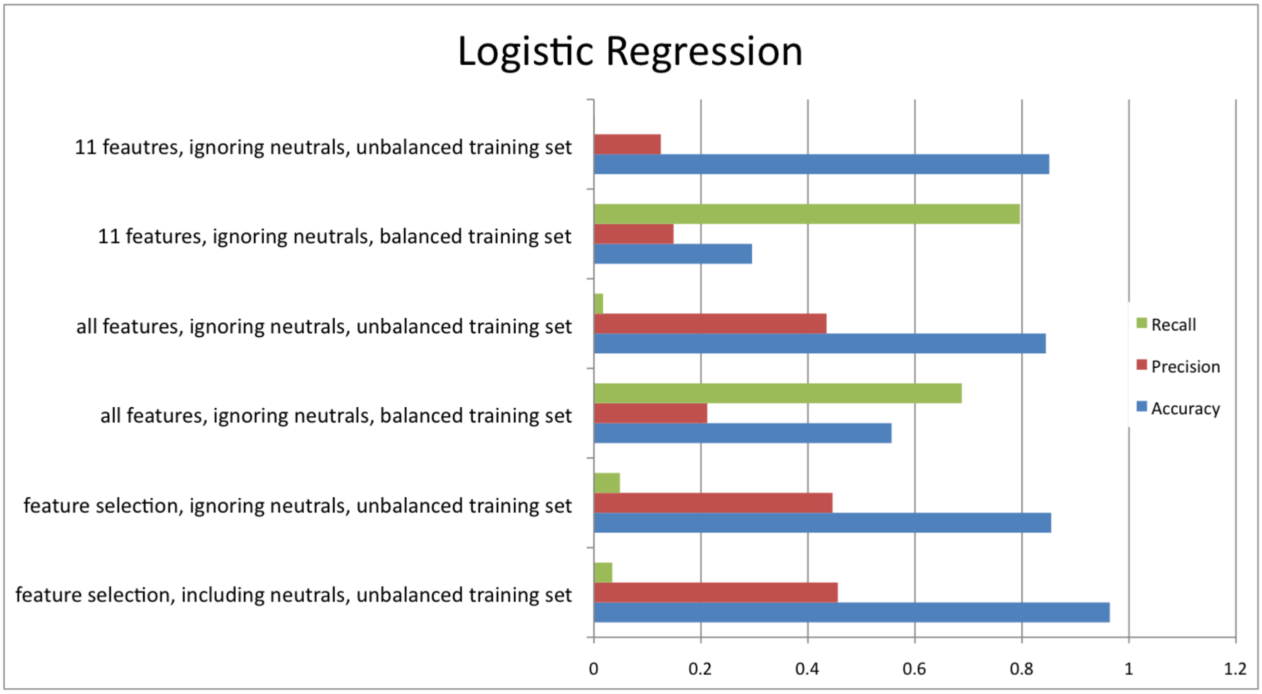
Results

Logistic Regression

We tested our Logistic Regression model using hold-out cross validation in two ways. First, we used all the data, training on 90% of our data and testing on the remaining 10%. Then, we wanted to train on a balanced set of negative and positive conversation as was done in the Leskovec et. al study since we have a significantly greater number of negative edges than positive. We constructed this balaned dataset by uniformly selecting and including one negative edge for each positive edge we have.

We used the accuracy, precision, and recall of our predictions as a measure of success. Since our dataset is highly skewed, we wanted to put less of an emphasis on accuracy.

The following graph dispays the results of our iterations on our Logisitric Regression model.

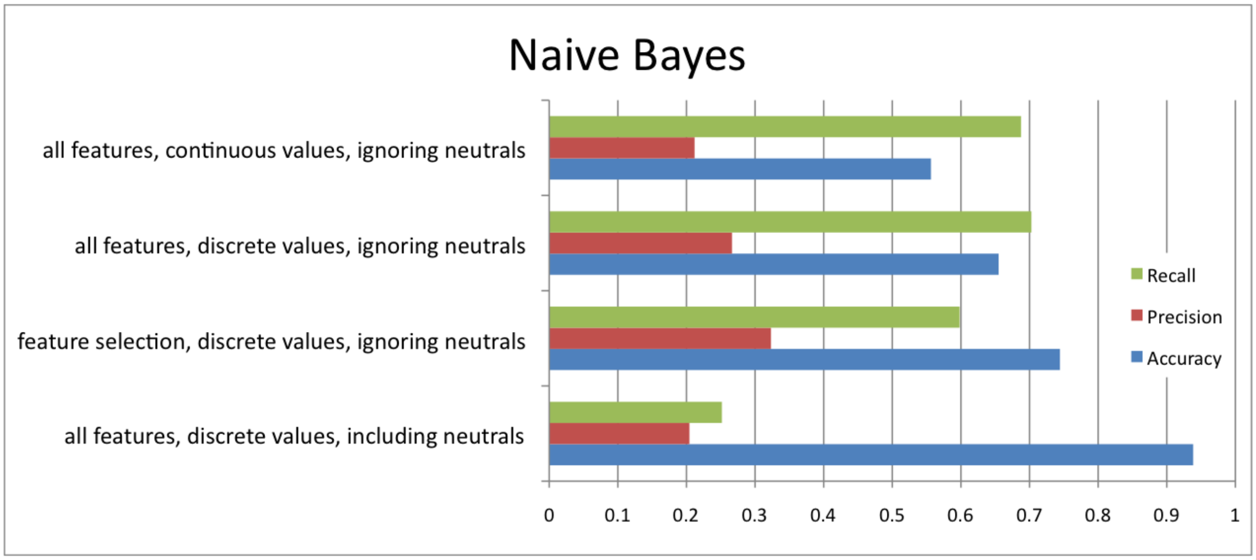


For the first two runs, we used a basic model with only the 11 features generated using the structure of our network. The next two runs use all of our features. We had significant gains in precision when we incorporated our PageRank and profile features. This means that with a better sense of the quality of a user, we are able to better predict good conversations. In our final two runs, we decided to only train using all the data since our results were poor using a balanced set. Adding in feature selection increased all our metrics. Finally, incorporating all the neutral edges increased our accuracy but that does not mean we improved our ability to predict good conversations since precision and recall remained constant. Our increased accuracy only means that we were very successful in identifying neutral conversations as negative. You can clearly see the tension between recall and accuracy in the above figure.

Naive Bayes Results

We tested our Naive Bayers model using hold-out cross valdiation as before, but this time we never tried training on a balanced set since we always had better performance training on the whole set.

The follwing graph displays the results of our iterations on our Naive Bayes model.



In the first run, we used our old feature set that had not be discretized with all 27 features. In the second run, we binned our PageRank and cosine similarity values. The boost in performance again confirms the predictive power of incorporating the quality of users into our network using PageRank. Next, we applied feature selection which signifcanly improved our predictions. Finally, we added back in all the neutral edges and again applied feature selection in our last run. Understandably, adding in all the neutrals decreases precision and recall because the number of conversations we predict to be good decreases since our training data now includes significantly more negative values.

Conclusion

Our best prediction algorithm ended up being Naive Bayes using discrete features with feature selection and ignoring neutral conversations. Thus, we found that neutral conversations in which one user immediately ends the chat do not help in predicting future conversations. Therefore, we recommend that Chatous should not heavily weight these conversations in their algorithms or calculations.

Additionally, we found that incorporating a notion of good and bad users into the algorithm using PageRank was very predictive. Although it may be computationally expensive to compute the PageRank of each user before a matching, we believe it would greatly benefit matching to occasionally compute the quality of each user and use this measure in predicting future conversations.

One of the most interesting revelations from training our models was that the features derived from the theory of structural balance did not add much information. Unlike the three graphs analyzed in Leskovec et al., the Chatous graph does not seem to follow with the triad-based theory of structural balance. One possible explanation for this is the anonymous/hidden nature of chats on Chatous. User1 never sees that one of his or her friends had a negative chat experience with User2. Thus, when User1 and User2 are matched up for the first time, there is no way for that negative chat to inform User1's opinion of User2. It would be interesting to further explore how anonymity and a lack of transparency in social networks affect the theory of structural balance.

All in all, the Chatous network provides an interesting context to observe interactions between people and determine what features are most important in user compatability. We were able to define a number of such features, but because the Chatous founders specfically wanted to initiate organic conversations by limiting user profile information, there is little data to use when making predictions. Hopefully, as their matching algorithm improves and the percentage of good conversations increases, they will be able to continually improve their models to predict more and more good matches.