FlickOh: Personalized Movie Recommendation and Rating System

INFO 290 Project Report

Natth Bejraburnin (arvizon@gmail.com)
Naehee Kim (tyche92@gmail.com)
Seongtaek Lim (stlim@berkeley.edu)
Mentor: Brian Guarraci (@emiscc)

1. Team Members and Roles

We have assigned roles depending on each member's strength and interest. Below are our tentative role assignments:

Natth - designs a metric that computes movie ratings, sentiment analysis of the tweets, evaluates different sentimental statistics, and helps with interest graph analysis.
Naehee - processes movie-relevant tweets, generates rating scores, and helps develop web UI and visualization.
Seongtaek - works on personalized recommendation, generates rating scores, and develops web UI and visualization.

2. Project Mentor

Brian Guarraci from Twitter

3. Project Goals and Achievement Summary

We aim to build a movie recommendation and rating system that utilizes Twitter data. The project is comprised of 3 core components, which are

1. General rating of movies — we will be collecting tweets from the streaming API and see what and how much people talk about the movie in question. The rating will roughly depend on the volume and the sentiment polarity of tweets.
2. Movie reviews on Twitter — extract tweets that are relevant to movies and present them as movies' reviews, which can be an alternative online resource for movie-lovers.
3. We will do the first two parts combined but the results will be personalized for *the user*. We will use only tweets that are published by the user's friends, and friends of friends within 2 degree of separation, and see how his/her circle talks about that movie. We will use collaborative filtering approaches to make a list of recommended movies for the user.

As the project has concluded, we have achieved every goal described above. In the past month, we have collected over 100 million raw tweets from the Streaming API and use them to rate movies based on their popularity on Twitter. We also have developed algorithms to make personalized movie recommendation for certain users, based on their Twitter interest graph. All the results can be found at http://people.ischool.berkeley.edu/~stlim/flickoh/. However, there were some ideas that we wish to have accomplished but were not able to by the deadline:

1. Find a way to verify our results (i.e. movie recommendations), which could be by conducting a survey on a sampled group of users.
2. Enable our system to respond to the user's request for movie recommendations in real time. This is quite a challenge largely due to the API limit. Based on our experiment, to come up with a list of recommended movie for a single user took about 15 hours on average and most of the time was spent on getting data from the Twitter API.
3. Fine-tune parameters or try different models and compare the results.
4. Take into account the timestamps of tweets when computing movies' ratings. The weights of tweets' polarity should be diminishing with time. Also integrate the number of retweets into the formula somehow.

4. Project Milestones and Timeline

We have defined 5 milestones for our project, which are the following:

1. Choosing data sources and collecting data. Having working codes to retrieve relevant data from various APIs, and to extract movie-related tweets from the data.
2. Classifying the sentiment polarity of movie-related as positive, neutral, or negative.
3. Designing and implementing a framework for analyzing interest graphs of users.
4. Designing an algorithm that takes as input the information from the above milestones and puts together personalized movie recommendations.
5. Developing web UI for these rating features.

Milestones	October		November				December
1							
2							
3							
4							
5							

5. Literature Reviews

5.1 Prediction Power of Twitter

Asur et al.(2010) found out that the tweet-rate (the number of tweets referring to a particular movie per hour) accurately predicts the revenue of a movie. Their method works best for prediction of first weekend Box-office revenues. This shows that the public attention on Twitter can be important indicators of future outcomes. However, sentiments of movies are not as important as tweet-rate though sentiments improve the accuracy of predictions.

Despite many studies that support the predictive power of Twitter data, it is still a controversial topic in the research community. In contrast to Asur et al.(2010), Wong et al (2012) find out that the movie reviews on Twitter, IMDB and Rotten Tomatoes cannot be used to predict the box office results. Also, there was no strong correlation between Twitter trends (on movies) and the movies' ratings from the users of the two popular movie-review sites (IMBb and Rotten Tomatoes). This suggests that the population on Twitter is different from that on the other two sites, which is generally considered more representative of the movie-audience community. Reproducibility seems to be one of the big issues here, along with others listed by Gayo-Avello (2012). In his survey paper, he points out that many of the papers that publish positive results about Twitter's predictive power have flaws in many ways, some of which are:

- they assumed that all tweets are trustworthy;
- they neglected demographics information, and therefore could not validate that the population on Twitter is representative of the true population;
- they did not address the issue of self-selection of the samples (those who voluntarily tweet about their movie experiences/opinions), which could be a source of bias;
- they did the analysis after learning the results (of elections, movie box office), which could potentially interfere with their methodology design.

These issues indeed play down the legitimacy of the previously published results and need be properly addressed if one decides to conduct similar research about Twitter.

References:

Asur, Sitaram, and Bernardo A. Huberman. "Predicting the future with social media." arXiv preprint arXiv:1003.5699 (2010).

Gayo-Avello, Daniel. ""I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper" A Balanced Survey on Election Prediction using Twitter Data." arXiv preprint arXiv:1204.6441 (2012).

Wong, Felix Ming Fai, Soumya Sen, and Mung Chiang. "Why watching movie tweets won't tell the whole story?." *Proceedings of the 2012 ACM workshop on Workshop on online social networks*. ACM, 2012.

5.2 Movie Recommendation

Recommender systems have been an active area of research for years both in the industry and the academia. Adomavicius et al. (2005) writes a comprehensive survey on modern approaches to the recommendation problems, which can be categorized to 3 types: content-based, collaborative, and hybrid.

Since Twitter does not provide much direct information on the movies' contents, the collaborative approach seems to be most suitable to our project. This approach basically assumes that 'similar' users are likely to favor same or similar items. It thus predicts if a user likes a particular items based on the preferences of those who are similar to the user. The metric used to measure similarity varies from model to model and so does the set of features or information about users and items. The paper points out to the system Ringo (1995) that successfully uses the collaborative approach to find recommendations of music albums and artists. But their framework can be applied to any type of database.

One traditional method for movie recommendation system is using data of users who are interested in a specific movie. Basu et al. (1998) suggested a hybrid method that used content features of movies (actors, actresses, directors, genres, and so on) and social features (users' preference) to make recommendation based on data of combination of content features and social features, such as users who liked genre of drama. Bogers (2010) suggested recommendation model using probability matrix, which is basically constructed upon similarity between movies' features. This work presented an algorithm for calculating probability of path on a graph structure with vertices of users and movies. These two utilized movies' factual features, or tags, as key criteria to recommend them to users. On the other hand, Shi et al. (2010)'s work employed contextual information of movies to make recommendations. Specifically, the algorithm used mood-specific movie similarity as an input to generate relevance between a user and a movie. For example, if each movie has scores for four kinds of mood

including sad, anxious, upset, and scared, mood-specific similarity between movies can be obtained and the system recommends the most relevant movie in terms of mood as a result.

References:

Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." Knowledge and Data Engineering, *IEEE Transactions* on 17.6 (2005): 734-749.

Basu, C., H. Hirsh, W. Cohen. 1998. "Recommendation as classification: Using social and content-based information in recommendation." *Proceedings of the American Association for Artificial Intelligence*.

Bogers, T. 2010. "Movie recommendation using random walks over the contextual graph." *Proceedings of the 2nd Workshop on Context-Aware Recommender Systems*.

Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating "word of mouth"." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995.

Shi, Y., M. Larson, A. Hanjalic. 2010. "Mining mood-specific movie similarity with matrix factorization for context-aware recommendation." *Proceedings of the Workshop on Context-Aware Movie Recommendation*.

5.3 Sentiment Analysis on Twitter

Bermingham et al.(2010) found that the accuracy of classifying sentiment in microblogs and microreview is not less than that in blogs and reviews. The sparsity of information in the short documents does not negatively affect the accuracy of sentiment analysis. This result gives us confidence in efficacy of applying sentiment analysis in order to classify feelings of movies. Though Bermingham et al. (2010) shed the light on the equivalent accuracy of sentiment analysis in short document, there have been approaches to enhance Twitter sentiment classification. Davidov et al. (2010) presented a framework which allows an automatic identification and classification using 50 hashtags (e.g. #sad, #bored, #fun) and 15 smileys (e.g.  , ). Their experiment shows that sentiment analysis on Twitter data can be improved by additional information such as hashtags and smileys. Jiang et al. (2011) proposed target-

dependent, and context-aware approaches. By identifying the extended targets (target related nouns) and relations of words, they try to distinguish the expressions describing the target from other expressions. For the short and ambiguous tweets, they found that context information – tweets by the same person, tweets replying to or replied to the tweet to be classified – can enhance the accuracy of sentiment classification.

References:

Bermingham, Adam, and Alan F. Smeaton. "Classifying sentiment in microblogs: is brevity an advantage?." *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport. "Enhanced sentiment learning using twitter hashtags and smileys." *Proceedings of the 23rd International Conference on Computational Linguistics:* Posters. Association for Computational Linguistics, 2010

Jiang, Long, et al. "Target-dependent twitter sentiment classification." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. 2011.

5.4 Similar Project

Twitflicks.com provides similar information/service as we plan to accomplish for a part of our project. The website basically assigns (general) ratings to movies based on sample public tweets and their sentiment polarity and the strength thereof. They have been fine-tuning their algorithms for a year and claim to be "the best and most unbiased user-based reviews on the web". We will basically deliver similar information but due to time limitation, we will probably use a less complicated algorithm. However, we distinguish ourselves from them by providing personalized rating and recommendations for the user.

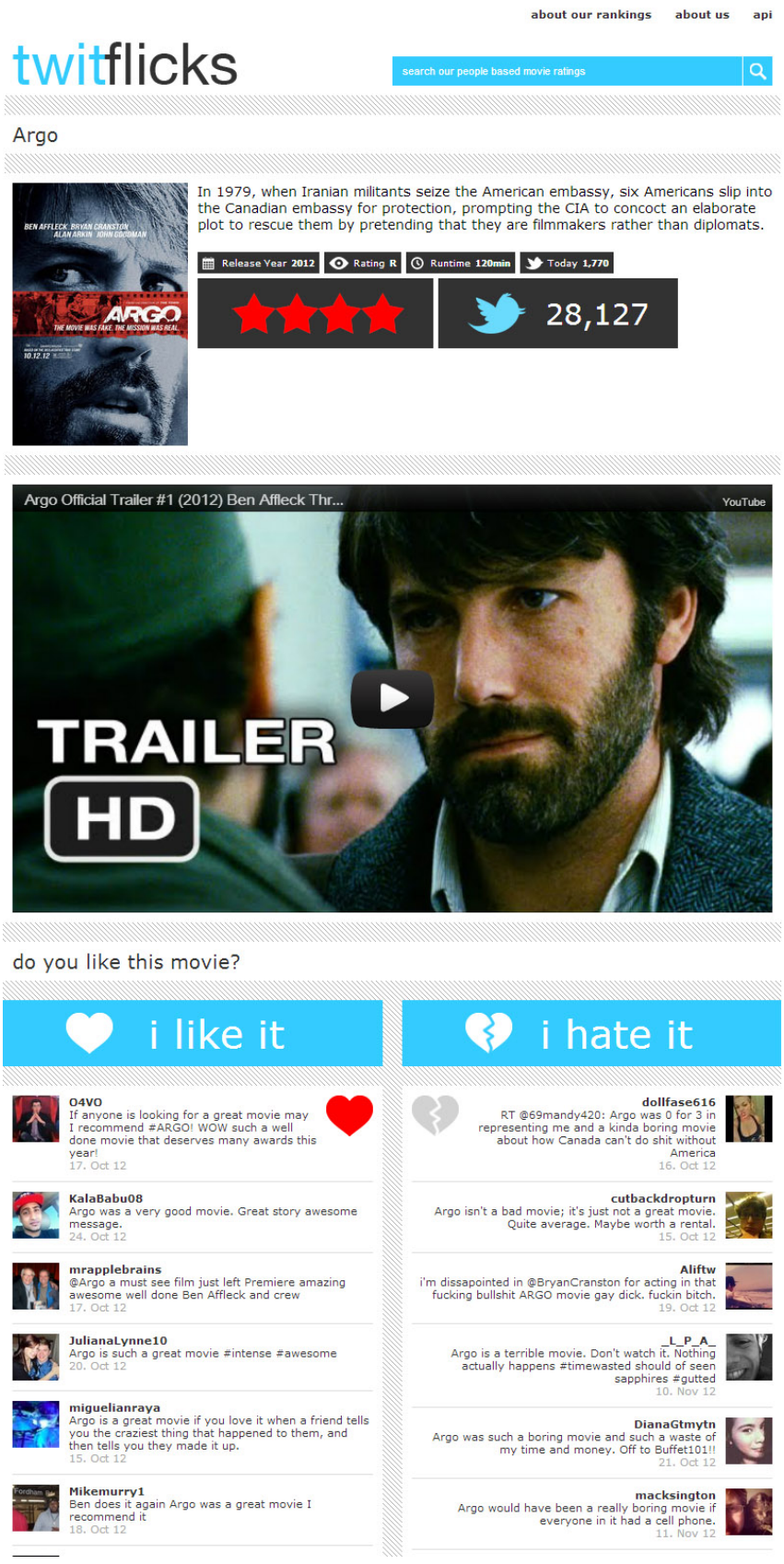

Figure 1 : A screenshot of a Twitflicks wegpage

6. Project Strategy and Algorithms

6.1 General Movie Ratings and Reviews

We had continually collected tweets from the streaming API from Oct 26, 2012 to Dec 2, 2012. We also manually picked a list of 86 movie titles that were released in October and November. Around 132,598,524 tweets were collected and we filtered out those which did not contain any movie title in the list. As a result, 127,815 of them were retained for use in sentiment analysis and computing the rating.

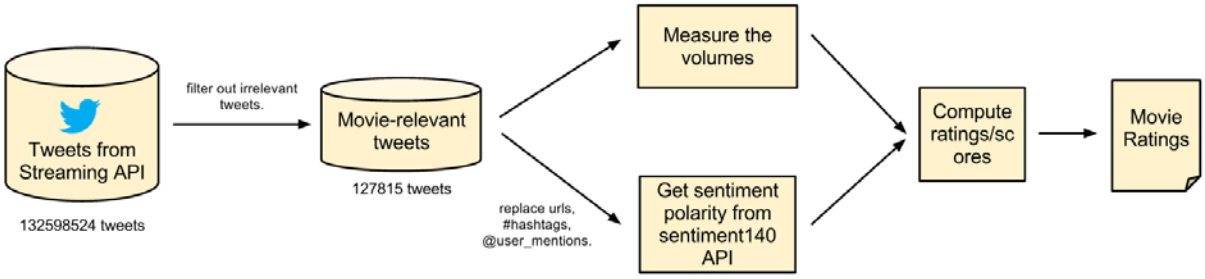

Figure 2: Our algorithm for general movie rating

For sentiment analysis, we solely rely on the external API from sentiment140.com, which will determine if a tweet is positive (4), neutral (2) or negative (0). We noticed quite a number of false negatives, especially positive or negative tweets classified as neutral. But improvement on this part is beyond the scope of our project.

We compute the general scores using the following formula

$$\frac{\#PT}{\#PT + \#NT + 1} \times (total\ number\ of\ tweets),$$

where PT means positive tweets, NT means negative tweets. The +1 in the denominator prevents division by zero. We multiply the ratio by the total number of tweets to take into account the movie's popularity. The results can be found at http://people.ischool.berkeley.edu/~stlim/flickoh/index.html.

For each of the movies on the top ten list, we also provide a page containing its factual information (credits to IMDb) as well as reviews from recently-published tweets.

6.2 Personalized Movie Recommendations

The results from the previous part are based on the public's opinions and thus might not be suitable for a particular user. So we aim to also provide movie recommendations personalized to *the user* of our system, based on the preferences of people on his/her Twitter interest graphs. Below is the flowchart of our algorithm.

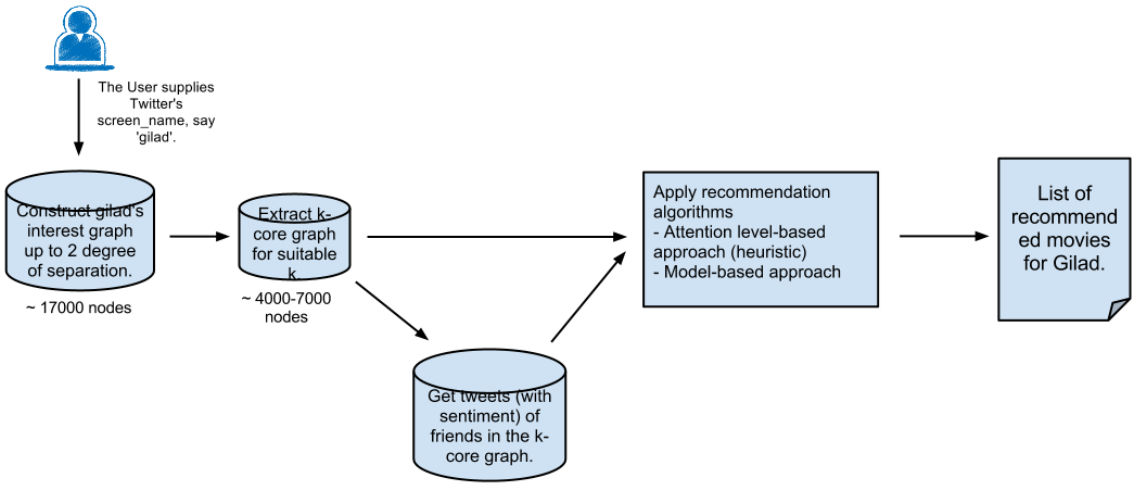

Figure 3 : Our algorithm for personalized movie recommendations

The process starts with retrieving the lists of the user's friends, friends of friends, as well as their statuses published in the past month. Then we construct the interest graph and extract its k-core for some suitable k that would reduce the size of the graph to about 5000 - 7000 nodes. This is necessary because retrieving their statuses from the API is quite expensive in terms of the number of queries. Then we supply this data into the recommendation algorithms, as described below.

6.2.1 Attention Level-based Approach

In this project, overall attention level in an interest graph is computed in order to score a movie for a user. This scoring method is computing levels of average preferences and attention of a user's friends with whom he share interest. This assumes that a user and his Twitter friends share interests and he is likely to have interest in what the friends are paying attention to. Step-by-step operations for the scoring are below.

First, we performed k-core analysis of a user's Twitter friends' graph to get his interest graph. This helped us identify a user group with similar interests and also eliminate insignificant nodes and edges in the graph. Second, the latest 60 tweets from each user within the graph were

collected. Assumingly, those recent tweets could cover the user's tweets for one or two months, thus, it is regarded reasonable amount of tweets to analyze scores for newly released movies. Third, using given database (e.g., IMDb), tweets with valid movie names were matched and stored. Some movie names are normal vocabulary so that we couldn't decide if it means a movie or not. For example, the word flight frequently appears within tweets, even though it means actual flight but not the movie title. In this project, this issue was set aside and all the matched tweets are used. Fourth, sentiment analysis was conducted to examine each tweet's polarity, which reflects the user's preferences on the movie, using sentiment 140 API. The results were from 0 (negative) to 4 (positive). Using this data, the formula for the personalized score was defined like this:

$$\sum\left(\frac{S_i R_i D_i}{L_i}\right)$$

Figure 4: Attention Level-Based Score of a Movie for a User, where S = result of sentiment analysis (polarity), R = reference of a movie title, D = degree of the friend node, and L = level of the friend node.

This implies the likelihood that the user prefers a movie that his friends do is proportional to the polarity, the number of references, degree of a friend, and inverse of the distance between the user and a friend. In other words, if a very influential and close friend in one's interest graph positively mentioned about a movie, there is bigger chance for the user to like the movie, either.

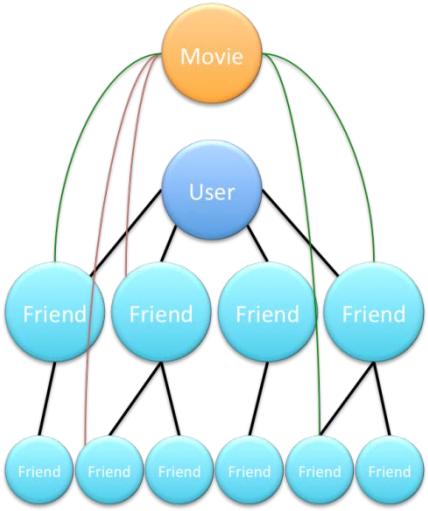

Figure 5: Virtual Edges to Illustrate the Formula: The green edges represent positive references of the movie and the red ones are negative ones.

6.2.2 Model-based Approach

As an alternative, we implement a recommendation system based on a probabilistic graphical model. Precisely, we use collaborative filtering with naive Bayes classifier [Miyahara et al. 2000], which aims to determine whether the User will 'like' a movie or not. The algorithm requires as input the rating matrix whose entries are people's ratings (features) on the movies (items). Since the matrix is usually sparse as people do not usually tweet about their movie preferences in our context, the paper suggests using the *boolean rating matrix*, whose entries are either 1 or 0, depending on whether the feature (positive/negative rating) of an item is present or not. More details and a sample matrix will be given below

However, there is still a problem with this framework when *the user* has only few or none of tweets indicating his/her movie preferences, as this is a supervised learning method. This is similar to the cold-start problem in any typical recommendation system when it cannot provide a personalized recommendation to a *new* user.

But Twitter data is advantageous as it not only provides information about users' movie preferences but also some insight on the relationships among users. We can utilize this type of information as a substitute for the missing one. Precisely, we divide the group of people in the user's interest graph as direct friends (DFs) and indirect friends (IDFs). If we regard the direct friend group (DF) as 'common interests' of the user and the indirect friend group (IDF), then we can reasonably treat the direct friends and movies as items alike in our model. With this assumption, we can train our model on the input IDF-DF matrix and compute the prediction (or recommendation) on the IDF-MV matrix.

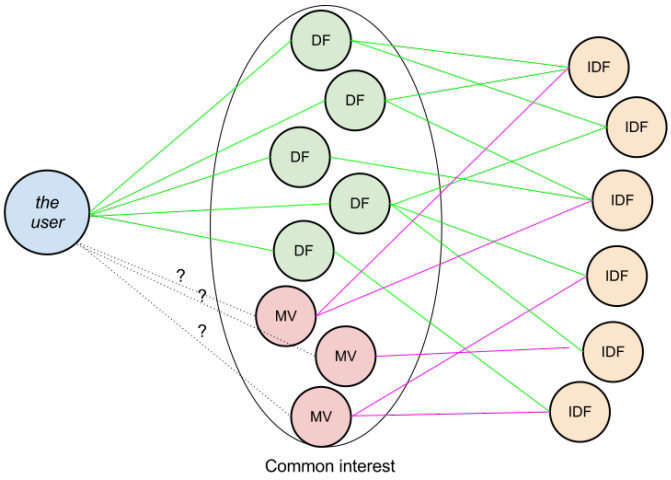

Figure 6: A sample graph showing relationships between users and items

	Training Data						Prediction		
	DF 1	DF 2	DF 3	DF 4	...	DF N	MV 1	MV 2	...
IDF 1 like	1	0	0	1		1	0	1	
IDF 1 dislike	0	1	1	0		0	0	0	
IDF 2 like	1	0	1	0		0	1	1	
IDF 2 dislike	0	1	0	1		1	0	1	
...									
the user	1	0	0	1		1	?	?	...

Figure 7: A sample matrix for training the model

In the IDF-DF table, the entry (IDF i like, DF j) is 1 if there is an edge between the two nodes, and is 0 otherwise. The entry (IDF i dislike, DF j) is simply the opposite of (IDF i like, DF j). The entries in the last row, however, are determined differently. The entry (the user, DF j) is 1 if the node DF j has more than or equal to T edges to the user's indirect friends, where T is some threshold and it is a model parameter. In other word, each entry in the last row will be 1 if there are many 1's in the cells 'IDF i like' in its respective column. This corresponds well to the IDF-MV matrix because the user should like the movie, if there are a lot of his/her indirect friends who posts positive tweets about the movie.

In the IDF-MV table, the entry (IDF i like, MV j) is 1 if the indirect friend i has at least one positive tweet about the movie j, and 0 otherwise. The entry (IDF i dislike, MV j) if he/she has at least one negative tweet about the movie j. Note that it is possible for both of them to be 1. And if both of them are 0, it means the indirect friend never tweeted about the movie.

The Algorithm: Naive Bayes Classifier

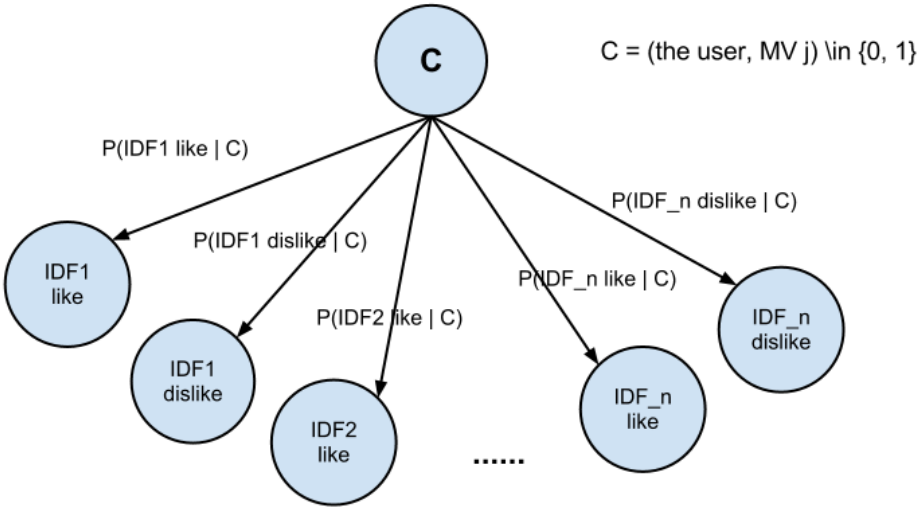

Figure 8: The graphical model underlying the naive Bayes classifier

Under this model, we basically want to estimate the probability that the user will like (and dislike) a movie given information about the features, i.e. IDF 1 like, IDF 1 dislike, IDF 2 like, and etc. Equivalently, these are the same as the probability that the entry (the user, MV j) will be 1 (and 0) given the information in the column MV j. The naive Bayes' model assumes that the features are pairwise independent given the class value C. Thus by the Bayes' rule, we have

$$P(class_j \mid IDF1\ like, IDF1\ dislike, \dots) \propto P(class_j)P(IDF1\ like \mid class_j)P(IDF1\ dislike \mid class_j)\dots$$

where $class_j$ represents the entry (the user, MV j), which could be classified as 'like' (1) or 'dislike' (0). As all the probabilities on the right hand side can be approximated from the training data, we can easily estimate the desired probability (on the left). And then we compare the two probabilities—of the user liking and disliking the movie—to determine whether to recommend the movie or not. In actual computation, we also use Laplace smoothing to avoid assigning a probability to be 0.

The recommended movie lists for 4 of our mentors (Gilad, Shai, Matthew, Alex) can be found on the following pages:
http://people.ischool.berkeley.edu/~stlim/flickoh/gilad.html
http://people.ischool.berkeley.edu/~stlim/flickoh/shai.html

http://people.ischool.berkeley.edu/~stlim/flickoh/mbilotti.html
http://people.ischool.berkeley.edu/~stlim/flickoh/alex.html

References:

Wong, Felix Ming Fai, Soumya Sen, and Mung Chiang. "Why watching movie tweets won't tell the whole story?." *Proceedings of the 2012 ACM workshop on Workshop on online social networks*. ACM, 2012. (http://arxiv.org/pdf/1203.4642v1.pdf)

Asur, Sitaram, and Bernardo A. Huberman. "Predicting the future with social media." *arXiv preprint arXiv:1003.5699* (2010). (http://arxiv.org/pdf/1003.5699.pdf)

Miyahara, Koji, and Michael Pazzani. "Collaborative filtering with the simple Bayesian classifier." *PRICAI 2000 Topics in Artificial Intelligence* (2000): 679-689.

6.3 Interest graph visualization

We visualized the social networks of sample Twitter users for their top recommended movies using D3 JavaScript. D3 is a web visualization library and provides graphic layouts. To represent social network, we used Force Graph layout which assigns the positions of nodes and links arbitrarily according to the global setting for the entire graph such as gravity and charge. This force graph is rendered by reading JSON file which contains nodes, links, and additional information.

Below is the flowchart of our visualization. Using pre-processed K-core graph of individual user and tweets with sentiment polarity, we reconstructed all the partial interest graphs for each movie for each target user. The partial interest graph contains only nodes which are either target user's direct and indirect friends talking about the movie or target user's direct friends who have friends talking about the movie. This graph reconstruction is necessary for better graph presentation and user interactions. For example, the number of nodes and links of gilad's interest graph for Skyfall movie were decreased from 3,690 and 17,368 to 191 and 1,686 respectively. By reconstructing interest graphs, we deleted unnecessary nodes for the specific movie.

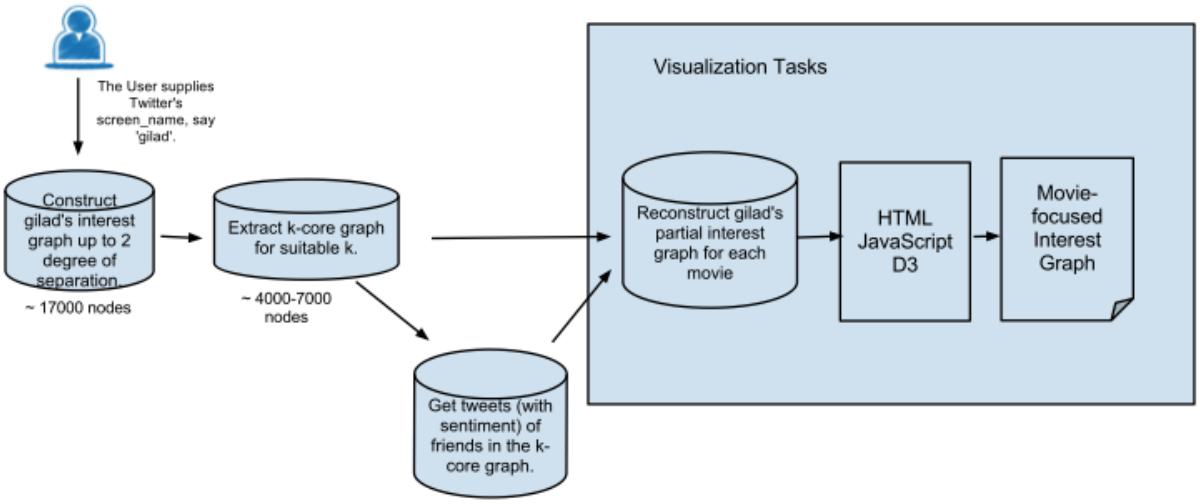

Figure 9: Flow chart for the visualization task

Below is a Skyfall-relevant interest graph for Gilad, one of our target users. Gilad node is a big purple circle centered in the graph. Nodes have three different sizes: the big size means Gilad himself, the medium size denotes Gilad's direct friends, and the small size represents Gilad's indirect fiends.

The color of each node represents the average sentiment polarity of Skyfall of the node. Blue - Orange color spectrum is applied to colors. Blue color represents positive sentiment; brown color shows neutral mood; orange color represents negative sentiment. When the node doesn't post about Skyfall, the color is light purple.

This visualization supports user interactions. When a user put his mouse over a node, the movie relevant tweets from the node show up.

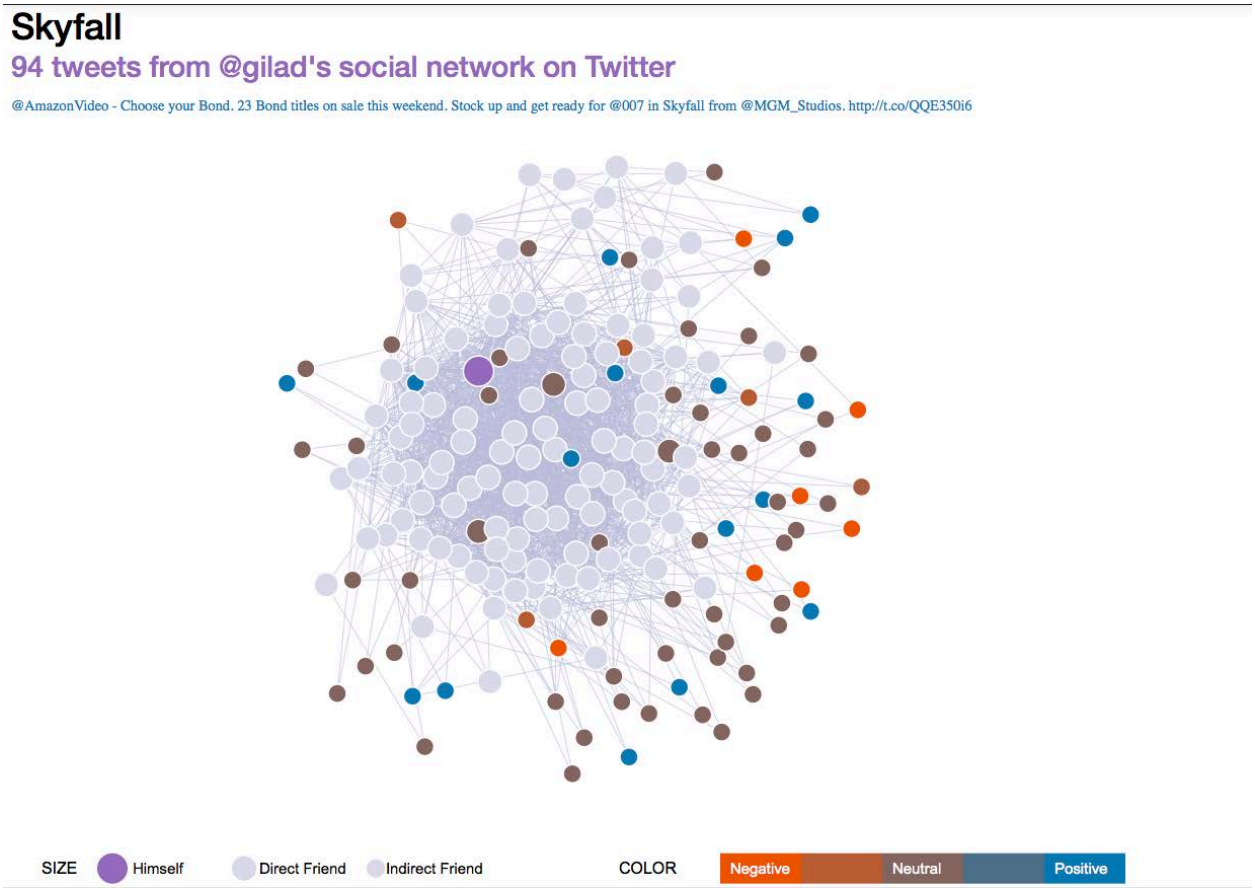

Figure 10: A screen shot of an interest graph visualization

7. Code Repositories and Data

Programs/Data	Repositories/Storage
Filtering movie-related tweets, graph visualization.	https://github.com/tyche7/flickoh https://github.com/tyche7/flickoh/wiki/FlickOh-Code-Documentation
Sentiment analysis, model-based recommendation, tweet-collecting both from Streaming and REST APIs.	https://github.com/arvizon/FlickOh-project https://github.com/arvizon/FlickOh-project/wiki/FlickOh-Code-Documentation https://github.com/stlim0730/TwitterDataCollector
Attention level-based recommendation	https://github.com/stlim0730/flickoh---personalized-scoring
Interest graph data for @gilad, @eismcc, @alsmola, @mbilotti, @shai.	https://github.com/arvizon/FlickOh-project

All movie-related tweets with sentiment polarity.	https://www.dropbox.com/s/hlrcacnb116i9t0/tweets_with_sentiment.zip
Tweets separated by their corresponding movie.	https://dl.dropbox.com/u/10429304/tweets_by_movies.zip
Sample raw tweets from the Streaming API**	http://dl.dropbox.com/u/10429304/1028122315.zip

** We cannot post all the data online as its total size is over 50 GB in zipped files.

8. Work Percentages

	Naehee	Natth	Seongtaek
Collecting streaming data - choose a list of movies - write code for getting tweets from Streaming API - collect tweets and curate the data	 100% 0% 0%	 0% 0% 100%	 0% 100% 0%
Extracting Movie-related Tweets - write code for extracting movie-related tweets - run code against the collected data	 100% 0%	 0% 100%	 0% 0%
Sentiment Analysis	0%	100%	0%
General Rating Algorithm - brainstorm ideas - apply algorithm to the collected data	 20% 0%	 60% 100%	 20% 0%
Personalized Recommendation Algorithm - brainstorm ideas - retrieve necessary data for analysis, including interest graphs of sample users, statuses of people in their graphs. - Attention level-based approach - Model-based approach	 5% 0% 0% 0%	 35% 80% 0% 100%	 60% 20% 100% 0%
User interface - brainstorm ideas - generate d3 graphs - create and maintain the website	 40% 100% 10%	 20% 0% 0%	 40% 0% 90%
Presentation and project report	30%	40%	30%