Spring 17, CSE 6242

A Novel Movie Recommender: Surprise Me!

Prepared by Team 18 (Surprise Me!):

Presented by:

Chia-Lin Cheng

Hongyi Jiang

Hyunjee Jin

Kuan-Yu Li

Po-Nien Lin

Tong-Hong Lin

Georgia Institute of Technology

Introduction

Today, availability of online resources and information technology has gigantic grown along with the growth of computer networks. Users can find massive information, but not always supported with tools when making decisions because the potential issues of usefulness and relevancy for the information. The purpose of recommendation system is aiming to find a more efficient way to solve such problems and to provide meaningful recommendations (movies, music, books etc.) on users based platforms.

Movie recommendation is the most widely used and pragmatic implemented with online multimedia platforms. Current movie recommenders are comprised by content-based or collaborative filtering algorithm. However, limitations exist: wrong results with poor scalability, cold start problem, or too predictable results.

To solve the problems, a novel recommendation system "Surprise Me!" is proposed. It can broader the horizons of users by satisfying their interests, but providing a new area that users haven't discovered yet. The tools and technologies of our team are listed:

- Amazon Web service: Codes are stored in Amazon S3. It could host the website, allowing us to access website through URL provided by Amazon Route 53 (Figure 1.).

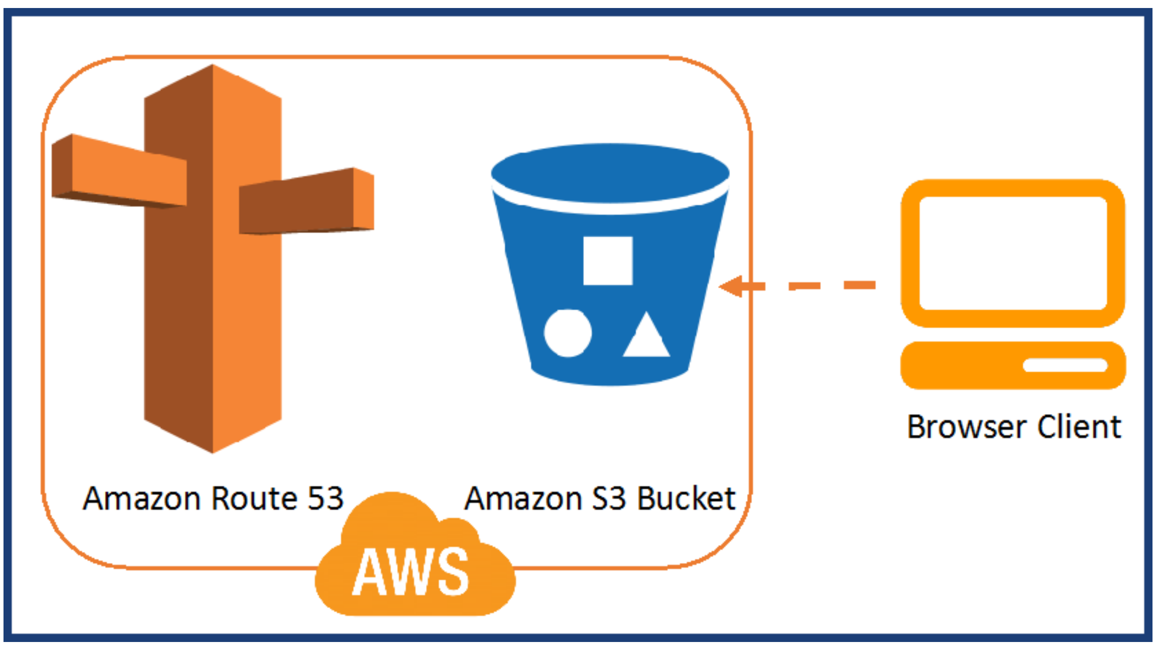

Figure 1. Amazon S3 website hosting

- Github: Use as the version-control tool and the cloud storage for codes by several people
- Languages
    - HTML: Construct the whole website.
    - CSS: Improve the appearance of our website.

- Javascript: Retrieve data, manipulate data, and implement our recommendation algorithms. Use XMLHttpRequest to send http request to TMDB and retrieve data.

- Data sources:
  - TMDB: A lot of useful data could be retrieved from TMDB by sending requests through APIs. It is a useful movie database because the API documentation is complete, and the interface of API is friendly and easy to use.
  - MovieLens: TMDB could provide us with only the data of users who logs in. Therefore, we need another dataset to get a lot of past data of other users. It is downloaded from GroupLens Research. It represents five-star rating and free-text tagging activity from MovieLens. The data were created by the selected users randomly who should rate at least 20 movies.

Survey

One of the recommendation algorithm, collaborative filtering, the approaches are based on user information such as gender, location, or preference [1]. Also, some collaborative method, like Facebook-based recommendation, can do the recommendation through information of close or trusted friends on social networks [2]. On the other hand, hybrid method use the elements of both content-based and collaborative-filtering methods to improve the computational performance [3], [4]. However, the hybrid algorithm will be too complex to run if it is not properly developed, causing long runtime of algorithm. Moreover, a paper proposed another approach based on the inference of users' emotion and input from other users [5]. Last, one more approach we have surveyed was the "Context-Walk" algorithm, which generating contextual graph and modeling user browsing process by implementing the random walk algorithm over the graph [6]. We can develop the normal mode of our movie recommender system based on these algorithms mentioned above. However, these approaches did not provide users "surprising" experience as we proposed.

Proposed Method

Intuition

Figure 2. shows the intuition of our proposed method. The user A is interested in Comedy movie, and then the conventional algorithm recommends the comedy movies to the user A. (The recommended items from the conventional algorithm are shown in blue dotted box.) However, the novel movie recommender "Surprise Me" finds a connectivity between the different groups and recommends new items beyond the interested group of user. (The recommended items from "Surprise Me" are shown in red dotted box.) In other words, "Surprise Me" considers the connectivity to other groups to broaden users' horizon and shows the recommended items which are not too predictable. Thus, it finds area that the user has not yet discovered.

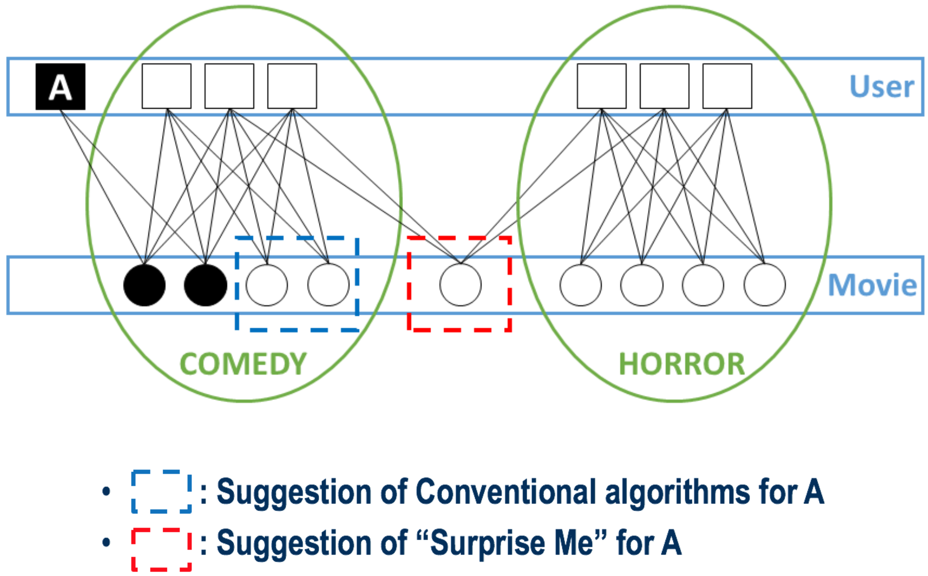

Figure 2. Visualization of “Surprise Me” algorithm

Flow and Architecture of Algorithm

The figure 3. shows how our website works. Due to the limitation of computing resources, we could not use complete data set from database. For example, the complexity of the naive matrix multiplication is $O(n^3)$, which is too much for PC. Javascript is not a language with good performance. Thus, we carefully chose the size of the data set used and modified the algorithm from the references.

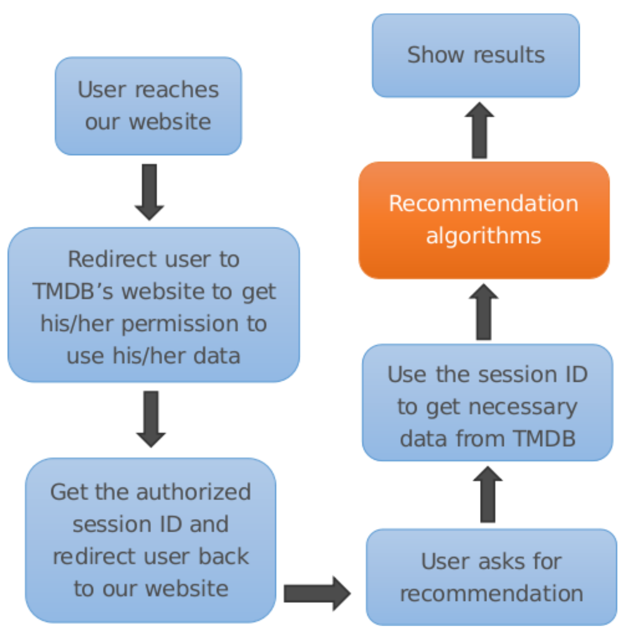

Figure 3. Flow architecture of overall algorithm

Recommendation Algorithm – TANGENT [6]

Flowchart of TANGENT:

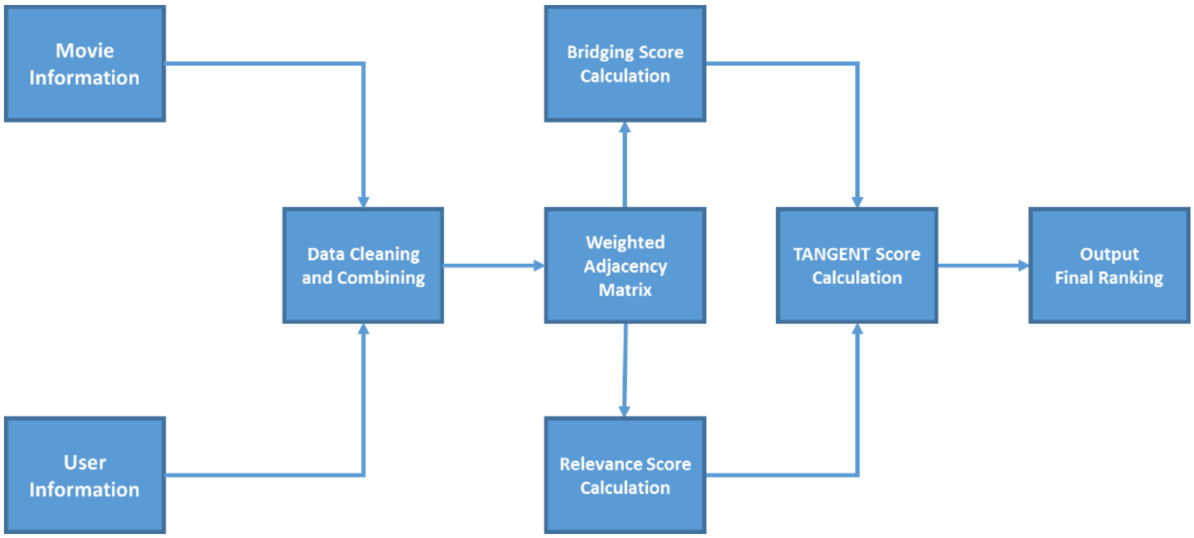

Figure 4. TANGENT flowchart

1. Since we gather data from the TMDB and MovieLens, the first step is to clean and combine the data. Because two data base uses different movie ID for the same movie, we map all movie ID in MovieLens to respective IDs in the TMDB.

2. Each movie and user is considered as a node. The weighted adjacency matrix (**A**) is the relation between all nodes. For example, if node 1 (user 1) rated node 5 (movie 3) as 5 points, then the element $a_{15} = a_{51} = 5$. If node 1 (user 1) didn't rate node 5 (movie 3), then the element $a_{15} = a_{51} = 0$.

3. If the user likes movie q_i, then we can calculate the relevance score of all other movies as the following:

   (a). $\overrightarrow{r_{q_i}} = 0.5(\mathbf{I} - 0.5\overline{\mathbf{A}})^{-1}\overrightarrow{e_{q_i}}$ where $\mathbf{I}$ is the identity matrix, $\overline{\mathbf{A}}$ is the normalized weighted adjacency matrix, $\overrightarrow{e_{q_i}}$ is a vector with all elements 0 except the q_i row is 1. The first row of the relevance matrix $r_{q_i,1}$ is the relevance score between movie 1 and loved movie q_i

   (b). If there is multiple loved movies q_i (ex: q_1, q_2, q_3), the relevance score between all loved movie and movie j will be $r_{Q,j} = 1 - \prod_{i=1}^{k}\left(1 - r_{q_i,j}\right)$

4. The bridge score b_i is used to evaluate if user or movie i can be served as a good bridge between multiple clusters. It is easier to use an example to explain the procedure:

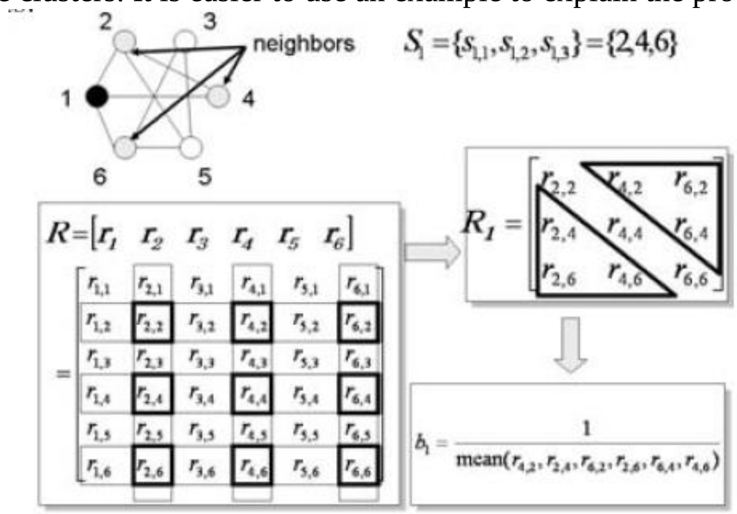

Figure 5. Example of TANGENT algorithm matrix

(a). The first step is to construct a set **S** which contains all nodes which are linked to user or movie i. If the total elements in the set S are 0 or 1, then the bridge score is 0

(b). Then we need to construct a matrix R which contains each pair of relevance score we calculated in step 3 in the set **S** as shown in the figure

(c). The bridge score is 1 divided by the average of all non-diagonal elements in matrix R

5. The tangent score is multiplication of the relevance score and the bridge score

6. By input the loved movies user chooses, we can calculate tangent scores of all other movies in the database and recommend user movies with high tangent scores

The following are the pseudocodes of TANGENT algorithm, please see the actual one attached with this report in CODE folder.

Pseudocode to calculate relevance score:

```
Load the relevance score matrix R generated using matlab
If there are n nodes and p loved movie

Create an initial relevance vector RQ = 0
for j = 1 to n do
{
  temp = 1;
  for i = 1 to p do
  {
    temp*=(1-R(i,j));
  }
  RQ(j,1) = 1-temp;
}
```

Pseudocode to calculate bridge score:

```
Load the relevance score matrix R generated using matlab
for i=1 to n do
{
 Pickup node i and let set S be the set of nodes which have links to i
Using set S to pickup wanted relevance score from the  relevance score matrix R to form a new matrix
  if (# elements = 0) OR (# elements = 0)
    bi = 0;
  else
  {
    Take average of all non-diagonal elements as avg_value
    bi = 1/ avg_value;
  }
}
```

Data Processing – MATLAB

The key to both relevant score and the bridge score is the relevance matrix. If we have the relevance matrix, both relevant score and the bridge score can be calculated by choosing specific elements in the matrix. However, to obtain the relevance matrix, both matrix multiplication and inversion are involved. These matrix process are very slow for our 2671 times 2671 matrix. Thus, it is impossible to use real-time calculation with javascript.

Our solution is to use matlab to calculate the relevant matrix and export them as a csv file. The website only needs to import the csv file and use simple calculation to obtain relevant and bridge scores. There are two advantages for using pre-calculation with matlab. First, matlab has specific algorithm to speed up the matrix calculation. Second, once we have the matrix, we don't need to calculate the matrix when user press the "recommend" button. This will make the whole process much faster.

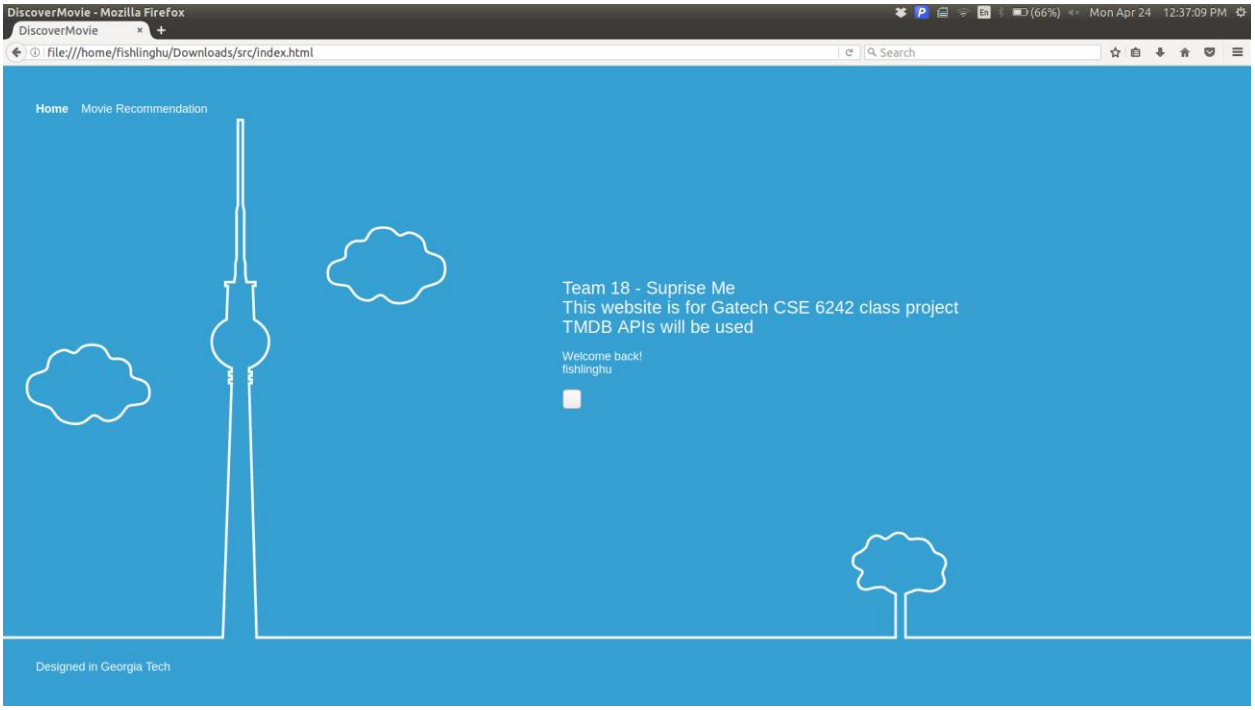

Figure 6. Index of our website

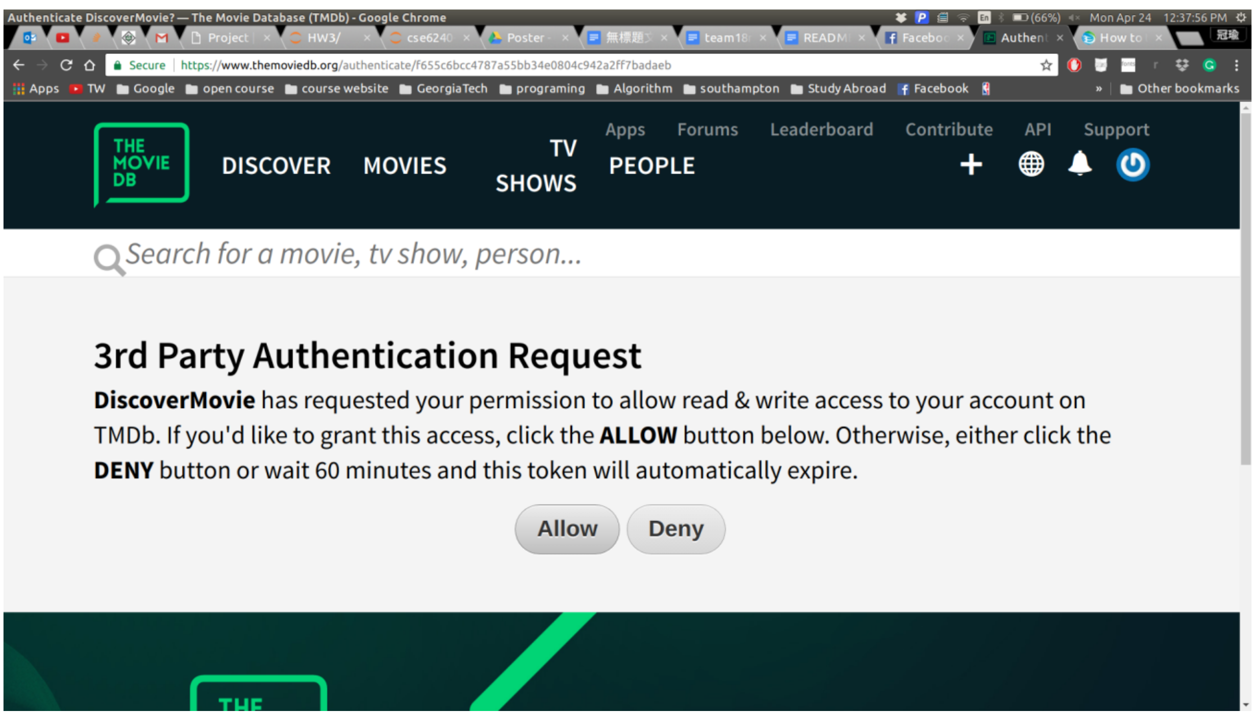

Figure 7. Request of permission to access users' TMDB preference

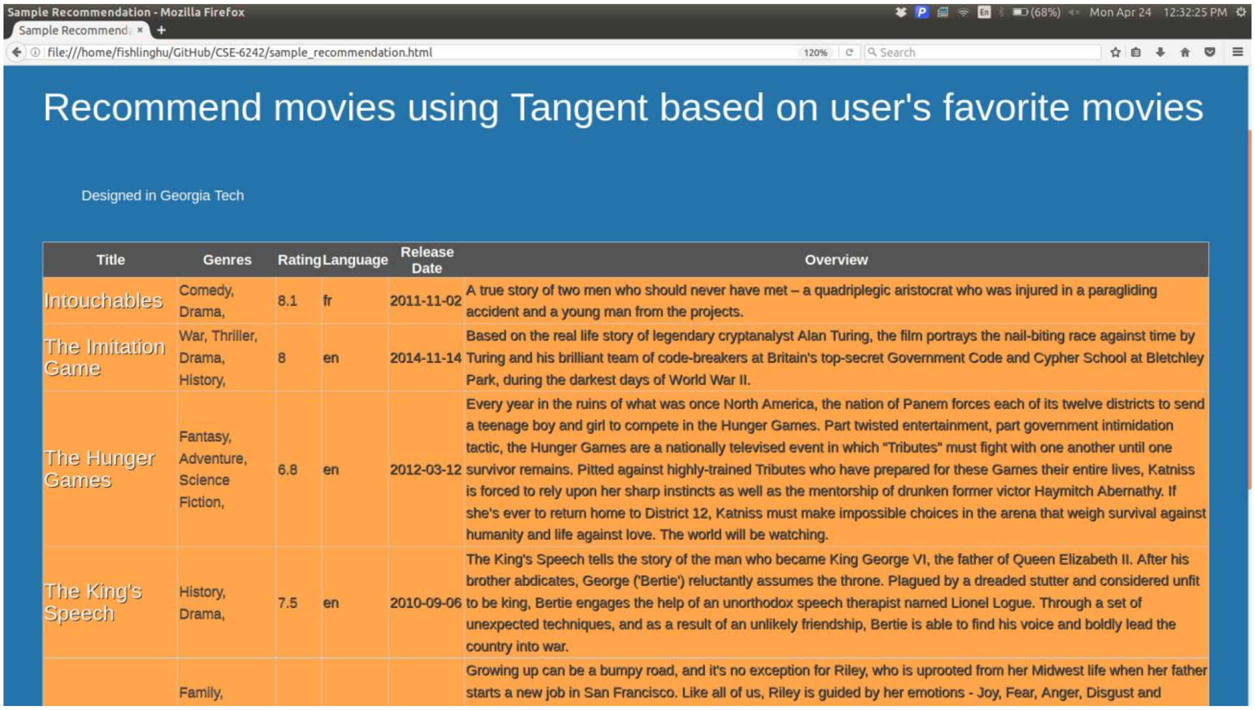

Figure 8. Our website automatically provides movie recommendations based on user' favorite movie lists.

Experiment and Evaluation

Inputs

We selected two users randomly in our team. They picked the favorites up to testing our approach. The below tables show the lists picked by two users respectively.

Table 1. 1st user's favorite movies

Movie Title	Movie Genre
Deadpool	Action, Adventure, Comedy, Romance
About Time	Comedy, Drama, Science, Fiction
The Best Offer	Drama, Romance, Crime, Mystery
One Day	Drama, Romance
Inception	Action, Thriller, Science Fiction, Mystery, Adventure

500 Days of Summer	Comedy, Drama, Romance
Twelve Monkeys	Science, Fiction, Thriller, Mystery
Schindler's List	Drama, History, War
Dumbo	Animation, Family

Table 2. 2nd user's favorite movies

Movie Title	Movie Genre
X-Men: Apocalypse	Action, Adventure, Fantasy, Science Fiction
Now You See Me	Thriller, Crime
The Dark Knight	Drama, Action, Crime, Thriller

Usage Scenarios

1. Make sure you have an account for TMDB and a list of favorite movies
2. Go to the index page of our website
3. When asked to give permission to our website, click on "allow"
4. Switch to the recommendation page
5. Our website will automatically gather information from you favorite movie list and provide movie recommendation

Outputs

Based on the usage Scenarios, finally we yield the recommendation list for the users respectively. The below tables show the recommended lists based on the user's' preference.

Table 3. Recommendation items for 1st user

Movie Title	Movie Genre
Intouchables	Comedy, Drama
Interstellar	Science Fiction, Drama, Adventure

The King's Speech	History, Drama
The Imitation Game	War, Thriller, Drama, History,
The Wolf of Wall Street	Comedy, Drama, Crime
Django Unchained	Western, Drama
The Hunger Games	Fantasy, Adventure, Science Fiction
Gone Girl	Drama, Thriller, Mystery
Inside Out	Family, Animation, Comedy
Mad Max: Fury Road	Thriller, Science Fiction, Adventure, Action

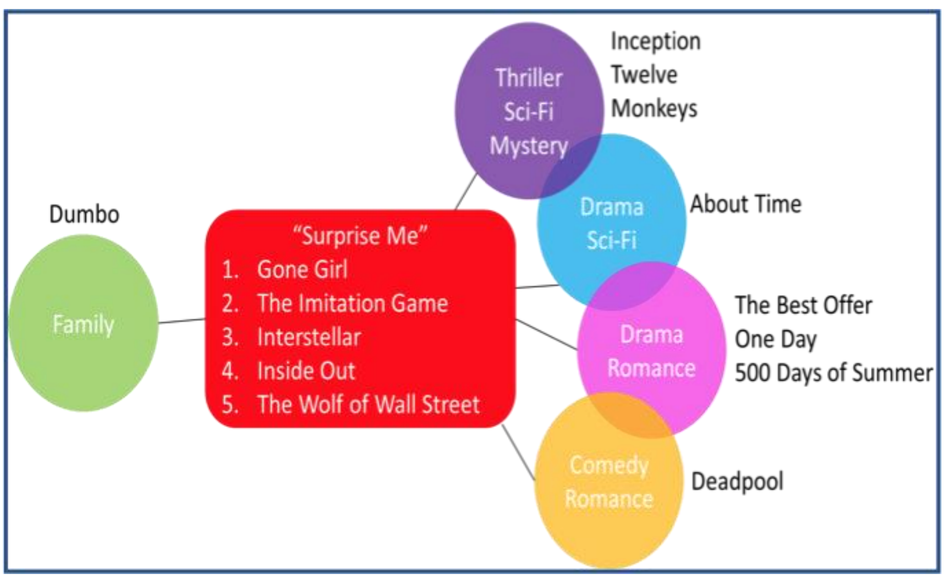

Figure 9. Visualization of recommendation results for 1st user

For 1st. User's usage scenario, the user has provided nine favorite movies as input and the recommendation system recommends ten movies as output (only top five movies are shown in the visualization figure). As shown in Figure 9, our recommendation system categorizes 1st. user's nine favorite movies into five clusters: (1) Thriller+Sci-Fi+Mystery, (2) Drama+Sci-Fi, (3) Drama+Romance, (4) Comedy Romance, and (5) Family. Our recommendation engine then calculates the TANGENT scores based on the mentioned five clusters' relevant scores and bridge scores. Finally, the top ten movies with the highest TANGENT scores are recommended to the users. Our recommendation system provides user “surprising effects” by connecting those tangential clusters with similar but not the same movie genres. Take the top one recommended movie Gone Girl for example, from the user’s favorite movie list, there are separate movies falling into the categories of Drama and Thriller, respectively. However, there is no any movie covering both Drama and Thriller genres. Thus, our recommendation engine recommends Gone Girl, which covers both Drama and Thriller genres, to provide a new movie category that the user has not discovered yet.

Table 4. Recommendation items for 2nd user

Movie Title	**Movie Genre**
The Hunger Games	Fantasy, Adventure, Science Fiction
L'écume des jours	Fantasy, Drama
About Time	Science Fiction, Drama, Comedy
Enemy	Mystery, Thriller
The Hobbit: An Unexpected Journey	Action, Fantasy, Adventure
Safety Not Guaranteed	Drama, Science Fiction, Romance, Comedy
Django Unchained	Western, Drama
The Wolf of Wall Street	Comedy, Drama, Crime
Guardians of the Galaxy	Adventure, Science Fiction, Action
Star Wars: The Force Awakens	Fantasy, Science Fiction, Adventure, Action

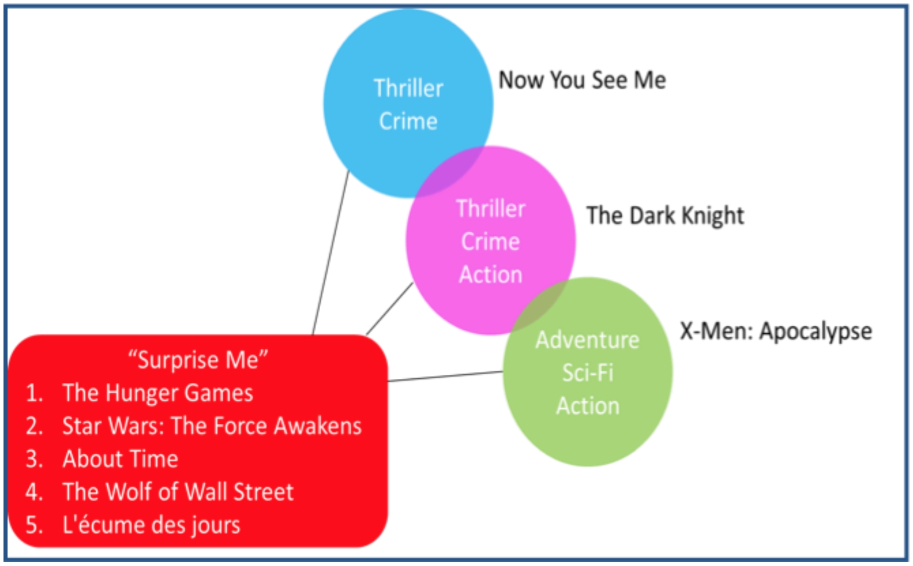

Figure 10. Visualization of recommendation results for 2nd user

For 2nd. User's usage scenario, the user has provided three favorite movies as input and the recommendation system recommends ten movies as output (only top five movies are shown in the visualization figure). As shown in Figure 10, our recommendation system categorizes 2nd. User's three favorite movies into three clusters: (1) Thriller+Crime, (2) Thriller+Crime+Action, and (3) Adventure+Sci-Fi+Action. Similarly, our recommendation engine then calculates the TANGENT scores based on the above three clusters' relevant scores and bridge scores and recommends users the top ten movies with the highest TANGENT scores. Take the top one recommended movie The Hunger Games (genres: Fantasy, Adventure, Science Fiction) for example, from the user's favorite movie list, there are a few movies including both Adventure and Science Fiction genres but not any movie including fantasy, which is identified by our recommendation system as relevant but not the same as genres Adventure and Science Fiction. Thus, our recommendation system recommends The Hunger Games that covers an additional movie genre of Fantasy, to provide "surprising effects" to users.

Conclusion and Future Work

Advantages and Summary of our result

- We can see that our system would recommend movies which are related to user's favorite movies not the same genres.
- The movie with high popularity has high probability of being recommended, since the recommendation algorithm is based on users' reviews to movies.

Future Work

From this report, we learned the power of TANGENT algorithm can really help users of movie recommendation system to explore their horizon, increasing the capabilities of users to enjoy their choice of movies with much better experience. Next, we are looking forward to have more user try on our algorithm and user interface (UI) then we can have more user data to help us improve, modify, and adjust this whole structure into a better application program for future users. Moreover, a better UI will be needed to help users to understand the process and to interpret the recommendation results more easily. Also, the final goal of our project is not just providing a platform for people who are looking for a good movie, we believe this TANGENT, or Surprise Me feature, can be implemented on different type of recommendation system as well, such as online shopping, book recommendation system in library, travel planning platform, etc. Finally, to help people exploring the world in much more different way or to initiate a whole new venture in your life are the goal of our project.

Work Distribution (After Progress Report)

Work Distribution (After Progress Report)	
Chia-Lin Cheng	Results Analysis and Writing/ Poster Making / Assisting the UI Coding
Hyunjee Jin	Introduction Writing / Poster Making / Result Analysis and Writing
Tong-Hong Lin	TANGENT Algorithm Coding / Data Preprocess Implementation / Algorithm Part in Report
Hongyi Jiang	TANGENT Algorithm Coding / Algorithm Part in Report
Kuan-Yu Li	User Interface Coding and Implementation / Algorithm Implementation / README Writing
Po-Nien Lin	Survey Writing and Plan of Activities / Result Analysis and Writing

References

1. Sang-Min Choi, "A movie recommendation algorithm based on genre correlations", *Expert Systems with Applications* 39 (2012) 8079–8085.
2. Golbeck, J. "Generating predictive movie recommendations from trust in social networks," in *Proceedings of Fourth International Conference on Trust Management*, Pisa, Italy, May 2006.
3. Gong S. "A collaborative filtering recommendation algorithm based on user clustering and item clustering," in *Journal of Software*, 2010, 5(7): 745-752.
4. Guanwen Yao and Lifeng Cai, "User-Based and Item-Based Collaborative Filtering Recommendation Algorithms Design," 2010 *Third International Conference on Knowledge Discovery and Data Mining*.
5. Ai Thanh Ho, Ilusca LL Menezes, and Yousra Tagmouti. "E-mrs: Emotion-based movie recommender system," In *Proceedings of IADIS e-Commerce Conference. USA:*

University of Washington Both-ell, pages 1–8, 2006.
6. K. Onuma, H. Tong, C. Faloutsos, Tangent: a novel, ”surprise me”, recommendation algorithm, in: *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, ACM, 2009, pp. 657–666