

# Virtual Reviewers for Collaborative Exploration of Movie Reviews

**Junichi Tatemura**

Institute of Industrial Science, University of Tokyo  
7-22-1 Roppongi Minato-ku Tokyo, 106-8558 JAPAN  
+81 3 3402 6231  
tatemura@iis.u-tokyo.ac.jp

## **ABSTRACT**

We propose a collaborative exploration system that helps users to explore recommendations from various viewpoints. Given ratings and reviews on movies from reviewers, the system provides “virtual reviewers” that represent particular viewpoints. Each virtual reviewer navigates the user by recommending and characterizing both movies and reviewers according to its viewpoint. We have developed a browsing method with virtual reviewers and visual interfaces.

## **Keywords**

Collaborative filtering, browsing, navigation, agents, information visualization, recommender systems

## **INTRODUCTION**

Collaborative filtering is an information retrieval technique that utilizes knowledge from other users [5] [6]. It can deal with a user’s subjective “taste” for items such as movies and music based on users’ ratings. However, the filter will not automate all of the user’s tasks to obtain information from other users: the user needs a query interface to access items actively. The user sometimes has more specific (and often temporary) needs than her or his general interests on which filtering results are based. In such cases, the user will need to explore for items that meet the specific needs. The user will also need to explore for less predictably but potentially interesting items the filter might exclude, that is, serendipitous information.

We propose a collaborative exploration system that generalizes the automatic recommendation technique of collaborative filtering in order to help users to explore recommendations from various viewpoints. Information about a certain kind of items can be obtained by consulting people who know those items well. Collaborative information exploration virtualizes this process by using rating data.

We have developed a movie database that realizes users’ collaborative exploration of movie reviews and ratings given by a number of reviewers on the Internet. Based on the rating data, the system provides a “virtual reviewer” that has a particular viewpoint. A viewpoint is represented as a set of movies, and a virtual reviewer simulates a reviewer who likes these movies. A virtual reviewer navigates the user by recommending and characterizing both movies and reviewers according to its viewpoint.

In this paper, we describe a virtual reviewer and its functions. We then propose a browsing method that uses virtual reviewers and an automatic clustering technique. We also introduce a visual embodiment of a virtual reviewer to realize visual explanation and querying of movies and reviewers from various viewpoints.

## **COLLABORATIVE FILTERING AND BROWSING**

The Tapestry system [2], which coined the term “collaborative filtering,” is a mail system that filters mail or news articles based on annotations given by other users. The system supports a query language TQL that enables the user to find articles that meet specific needs. The user, for instance, can obtain articles recommended by a specified person. The system, however, is not suitable for a large community in which users don’t necessarily know each other: it is meant to support a work group in which the user can specify appropriate actual users to get information.

Recent popular collaborative filtering systems [5] [6] automate selection of users who share interests with the user. By computing similarity of the users based on their rating patterns, the system provides the user with similar users (neighbors) and items they recommend. However, as described in the previous section, the user needs tools for exploration in addition to automatic filters.

Collaborative browsing has been studied to help multiple users to collaborate in browsing synchronously [7] or asynchronously [8][9]. These systems provide a group of users with tools to share processes or histories of browsing. For a large community, however, the systems will need additional ways to organize diverse information from users with various interests or tastes.

Let's Browse [3] helps a group of users to find items of common interest even if they don't know each other. The system, designed for real-time browsing sessions, provides appropriate topics for specific participants but does not search appropriate reviewers for specific interests.

Our system is designed for exploration of various users (reviewers) in a large community as well as exploration of items (movies). The user can automatically or manually create various virtual reviewers, each of which consists of a set of movies and a set of users who like those movies. While a filter uses a single durable profile to handle the user's general interests, multiple virtual reviewers provide various viewpoints to help the user to explore both reviewers and movies.

## VIRTUAL REVIEWERS

### Ratings and Reviews

For each movie  $m$  in the database, a user  $u$  can give a rating  $rating(m,u)$  and a review  $review(m,u)$ . To distinguish users' rolls in collaborative exploration, we call a user that has given reviews a reviewer and a user that browses reviews a reader. A user can be both of a reviewer and a reader.

A rating value is an integer that ranges from 1 (awful) to 5 (excellent) and the users are supposed to assume the margin between positive and negative ratings at 2.5. If  $u$  has not given rating on  $m$ ,  $rating(m,u)$  has the default value 2.5.

### Virtual Reviewer

The user can represent a *viewpoint* by combining multiple movies. Let  $M_v$  be a set of movies that represents a viewpoint  $v$ . A set of users  $U_v$  who like movies in  $M_v$  can be defined as  $U_v = Fans(M_v)$ :

$$Fans(M) = \{u \mid liking(u, M) \geq t\}.$$

The function  $liking(u, M)$  returns the user  $u$ 's overall liking of the movies in a set of movie  $M$ .

We define a *virtual reviewer* with respect to a viewpoint  $v$  as  $\langle M_v, U_v \rangle$ . A virtual user can characterize movies and reviewers by using rating data from  $U_v$  and  $M_v$ , respectively. For example, liking values on a user  $u$  and a movie  $m$  can be defined as  $liking(u, M_v)$  and  $liking(U_v, m)$ , respectively.

As a liking value, we currently adopt the mean of ratings:

$$liking(U, M) = \frac{1}{|U| \cdot |M|} \sum_{u \in U} \sum_{m \in M} rating(m, u).$$

### Recommendation

The set of movies that appeal to a set of user  $U$  can be defined as:  $Favorites(U) = \{m \mid liking(m, U) \geq t\}$ .

Then, recommendations  $R_v$ , and reviews  $Reviews_v$  from the virtual user with respect to a viewpoint  $v$  can be defined as:

$$R_v = R(M_v) = Favorites(U_v) - M_v.$$

$$Reviews_v = \{review(m, u) \mid m \in R_v, u \in U_v\}$$

By selecting some of the movies on which the reader has rated, she or he can obtain more focused recommendations than those of collaborative filtering results.

### Delivery of News

By storing a virtual reviewer, the reader can use it as a filter that delivers new information according to its viewpoint. The virtual reviewer with respect to a viewpoint  $v$  is stored in the user profiles as the set of movies  $M_v$ . Note that the set of user  $U_v = Fans(M_v)$  varies its members as new reviewers give reviews and ratings. The virtual reviewer thus provides the reader with new members in  $U_v$ ,  $R_v$ , and  $Reviews_v$ .

### Composing Virtual Reviewers

The reader can represent a viewpoint with an arbitrary set of movies, such as movies directed by a particular director, movies that the reader likes, and movies that the reader wants to see. As described in the next section, an automatic clustering function helps the reader to select movies from the database.

## EXPLORATION BY CLUSTERING

To enable the reader to explore movies and reviews according to various viewpoints, we have developed a browsing method that involves recommendation by virtual reviewers and an automatic clustering technique. Applying recommendation and clustering repeatedly, the reader can refine a set of movies to represent a viewpoint or traverse sets of movies to explore various viewpoints.

A clustering procedure divides a set of movies  $M$  into multiple sets of movies based on similarities between movies:  $Cluster(M, n) = \{M_i \mid \bigcup_{i=1}^n M_i = M\}$ . We have applied a simple vector space model to compute the similarity from the rating patterns.

Since members in  $M_i$  are similar to each other in a certain sense,  $M_i$  is considered to represent a certain viewpoint, from which the reader can obtain recommendations  $R(M_i)$ .

By applying the clustering procedure, the reader can refine movie clusters to represent more desirable viewpoints. This iterative refinement is based on a document browsing method, called Scatter/Gather [1], which applies iterative document clustering to browse large document collections. Given an initial set of items  $M$ , our browsing method repeats the following steps:

1. The system shows a list of clusters  $L$  given by one of the following equations (the user chooses one).

$$L = Cluster(M, n) \tag{1}$$

$$L = Cluster(R(M), n) \cup \{M\} \tag{2}$$

2. The reader selects a subset of  $L$ :  $S \subseteq L$ .

3. The system merges clusters in  $S$  into a new cluster  $M$ .

We have extended the original Scatter/Gather interaction by introducing recommendations from a virtual user, i.e. the equation (2).

Exploration by clustering goes as follows. By applying iterative clustering, the reader composes a set of movies  $M_1$  from  $M_0$ . The reader then obtains recommendations  $R(M_1)$  according to the viewpoint represented by  $M_1$ . From  $R(M_1)$  and  $M_1$ , the reader can compose a new set of movies  $M_2$  to represent a new viewpoint.

The original Scatter/Gather interaction starts with a large set of documents such as search results or the whole data set. Our exploration of viewpoint can start with various sizes of movie sets that interest the reader; even a single movie can be an initial set of movies. Typical initial sets include the set of movies on which the reader has given ratings and any meaningful lists of movies the database will display such as a list of the most popular movies and a search result queried by person, year, or country.

### VISUAL EXPLANATION

Making use of a virtual reviewer that can evaluate both movies and reviewers, we introduce a visual embodiment of a virtual reviewer to visualize characters of movies and reviewers. We have developed a visual query interface that visualizes the relationship between virtual reviewers and their recommendations.

#### Visual Virtual Reviewers

A virtual reviewer is visualized as a rectangle icon that displays a bar chart showing the component ratio of ratings to characterize movies and reviewers. When a movie or reviewer is pointed with a mouse cursor, the virtual reviewer displays a bar chart based on ratings on the pointed movie given by the reviewers  $U_v$  or ratings by the pointed user on the movies  $M_v$ , as can be seen in Figure 1.

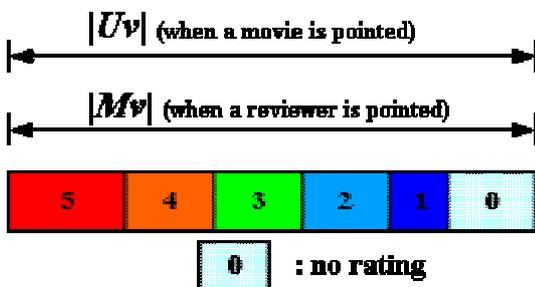


Figure 1 Bar charts that characterize movies and reviewers.

While *liking* values give overall relevance of movies and reviewers to the viewpoint, the bar chart provides detailed relationships of them to the viewpoint. For example, bar charts can differentiate the following movies (or reviewers) that can have the same *liking* value: (1) a movie for which some reviewers in  $U_v$  give high ratings but others give low ratings, (2) a movie for which most reviewers in  $U_v$  give moderate ratings, and (3) a movie for which a small group of reviewers in  $U_v$  give high ratings but the others give no rating.

### Recommendation

Figure 2 shows a result of recommendation from a virtual reviewer. As movie titles or reviewer names are pointed with a mouse cursor, the virtual reviewer shows corresponding bar charts. The bar charts visualize the relationship of individual movies and reviews in the recommendation to the viewpoint.



Figure 2 A bar chart characterizing recommendations. The bar chart at the top of the figure visualizes the character of the movie (titled “Les Amants du ...”) pointed by the mouse cursor. Sentences following the title are reviews (in Japanese) from reviewers in  $U_v$ . The bar chart visualizes the character of the reviewer if the cursor points the name at the end of a review.

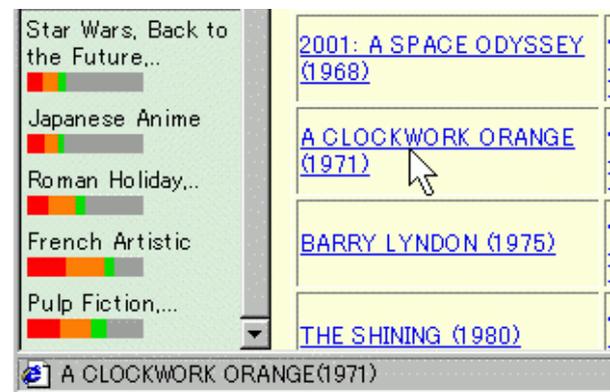


Figure 3 Guides for browsing movies and reviews. The five bar charts on the left frame of the window are virtual reviewers the reader has stored. The words attached to the bar charts are names of the virtual reviewers given by the reader. Each bar chart characterizes the pointed movie “A Clockwork Orange.”

#### Browsing Guides

The reader can store virtual reviewers she or he composed and retrieve them later as browsing guides. Whenever the reader browses movies and reviews in the database with a

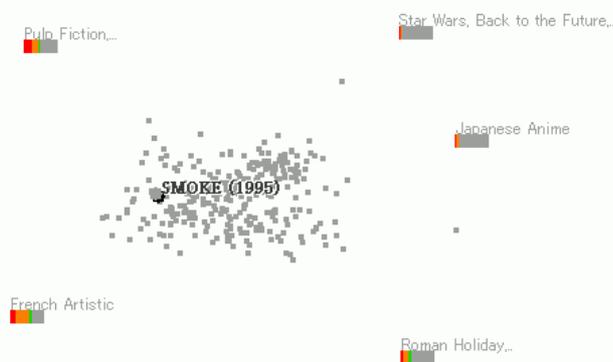
web browser, the system displays a list of virtual reviewers aside in the same window. If any item on the display is pointed, the list visualizes its character from multiple viewpoints. In Figure 3 the list of virtual reviewers characterizes the movie pointed with the cursor. The list of virtual reviewers also helps the reader to examine recommendations from a virtual reviewer by giving opinions from other viewpoints.

### VISUAL QUERYING

The reader can use multiple virtual reviewers as a query to retrieve movies and reviewers. Applying the document visualization technique of the VIBE system [4], our system provides a visual interface to explore the information space of movie reviews.

The VIBE system visualizes a document collection according to a user's "points of interest" defined by a number of keywords and their display positions. Each document is automatically located at a position representing the ratios of its relevance to the points of interest.

To represent the information space of reviews, our points of interest are represented by virtual reviewers instead of keywords. Each item is placed according to its liking values given by virtual reviewers. While the original VIBE lays out documents, virtual reviewers can visualize both movies and reviewers on the same plane.



**Figure 4 Visual querying of movies and reviewers.**

Figure 4 shows a result of a visual query with virtual reviewers. Retrieved movies are visualized as small squares surrounded by rectangles that represent the virtual reviewers. Pointing a movie exposes its title and bar charts of the virtual reviewers to characterize the movie. As a virtual reviewer is dragged, each movie follows the move in proportion to its relevance to the virtual reviewer.

When a movie is selected by clicking, reviewers who have given ratings for the movie are plotted on the display and colored by the ratings they gave. While the color patterns of bar charts gives summaries of ratings based on the multiple

viewpoints, the color pattern of plotted actual reviewers shows the relationship of the individual ratings to the movie. The locations of the reviewers also help the reader to understand reviews in the context of the visualized viewpoints. When a review is selected, the virtual reviewers show bar charts corresponding to the reviewer.

### CONCLUSION

We proposed a collaborative exploration system that helps users to explore movie reviews from various viewpoints. A virtual reviewer navigates the user by recommending and characterizing both movies and reviewers according to its viewpoint represented by a set of movies. We have developed a browsing method with virtual reviewers and iterative clustering of movies. We also introduced visual interfaces with virtual reviewers that retrieve and explain items visually from various viewpoints. Future work includes user testing to evaluate the effectiveness of the browsing method and empirical studies through uses in actual communities.

### REFERENCES

1. Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of ACM SIGIR'92*, 1992
2. Goldberg, D., et al. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, vol. 35, No. 12, 1992, 61-70.
3. Lieberman, H., Van Dyke, N., and Vivacqua, A., Let's Browse: A Collaborative Web Browsing Agent. *Proceedings of ACM IUI'99*, 1999, 65-68.
4. Olsen, K. A. and Korfhage, R. R., Desktop Visualization. *Proceedings of IEEE Symposium on Visual Languages*, 1994, 239-244.
5. Resnick, P., et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of ACM CSCW'94*, 1994, 175-186.
6. Shardanand, U. and Maes, P. Social Information Filtering: Algorithms for Automating "Word of Mouth." *Proceedings of ACM CHI'95*, 1995, 210-217.
7. Sidler, G., Scott, A. and Wolf, H. Collaborative Browsing in the World Wide Web, *Proceedings of 8th Joint European Networking Conference*, 1997.
8. Twidale, M. and Nichols, D. Collaborative Browsing and Visualization of the Search Process, *Proceedings of ELVIRA-96*, 1996
9. Wittenburg, K., Das, D., Hill, W., and Stead, L., Group Asynchronous Browsing on the World Wide Web. Fourth International World Wide Web Conference, 1995.