

VISUAL EXPLORATION FOR SOCIAL RECOMMENDATIONS

Junichi Tatemura

Institute of Industrial Science, University of Tokyo

7-22-1 Roppongi, Minato-ku Tokyo, 106-8558 JAPAN

tatemura@iis.u-tokyo.ac.jp

Abstract We propose a visual exploration technique, “visual exploration by example,” for recommendations in a social filtering system. A dynamic semantic map of items, rearranged dynamically for each query or filtering result, gives explanations of the result with contextual information and helps the user’s composition of a new query. We also have introduced visual interaction techniques to the map to examine more detailed relationship of filtering results: coloring by rating and highlighting by similarity. We demonstrate an example of the user’s exploration using a prototype interface for a movie database.

Keywords: Informaion visualization, visual query, social filtering, recommendation, information exploration

1. INTRODUCTION

Social filtering is an information retrieval technique that utilizes knowledge from other users (U. Shardanand and P. Maes, 1995) (P. Resnick et al., 1994). It can deal with a user’s subjective “taste” for items such as movies and music by computing similarity of the users based on their rating patterns; the filtering system provides the user with similar users (neighbors) and items they recommend.

However, even if the filter successfully provides items the user will like, it will not automate all of the user’s tasks to obtain information. A simple social filter lacks the following functions to help users.

- **Explanation of filtering results.** A user often likes various kind of items with different tastes. Because of this diversity of a user’s tastes, filtering results (i.e., items and reviewers) also have diversity in tastes. The user needs to understand how filtering results relate to tastes or interests.

- **Dealing with specific needs.** A user needs a query interface to access items actively. A 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.
- **Avoiding Over-fitting.** If a user relies too much on the filter to get information and gives ratings only for filtering results, the filter, learning only from this feedback, tends to *overfit* to a portion of the user's interests, and the user might miss potentially useful information. As long as the user gives ratings only for the filtering results, the filter will continue to provide items according to this portion and fail to cover all the tastes. Although the filter can be trained by specifying missing items, the user's needs is often too vague to specify. Sometimes, the user is not aware of the needs before encountering unexpected but interesting information, which is called "serendipitous information."

The user needs to explore the information space to find items that meets specific needs and items missing from the filtering results. We claim that the system should provide contextual information of filtering results to explain how some items are included in the filtering results and others are excluded. The system should also provide querying function to let the user access items actively.

In this paper, we propose a visual exploration technique, "visual exploration by example," for recommendations in a social filtering system. A dynamic semantic map of items, rearranged for each query or filter result, gives explanations of the result with contextual information and helps the user to compose a new query.

We have developed a movie database on the World-Wide Web and collected ratings and comments on movies. Our prototype interface has been developed to visualize data from this movie database. At the time this paper is written, the database has 6,740 movies, 530 users, 70,416 ratings, and 6,346 reviews.

The next section shows an outline of our approach. Section 3 describes a query model that enables the user to query items and other users by example. Section 4 presents the algorithm to arrange a semantic map dynamically. Section 5, introduces interactive visualization functions that helps the user to browse a map. Section 6 demonstrates an example of the user's exploration. Sections 7 and 8 discuss related work and future work, respectively. Section 9 concludes the paper.

2. VISUAL EXPLORATION BY EXAMPLE

2.1. EXPLORATION BY EXAMPLE

To explore a information space for her or his needs, the user requires means by which the user understands filtering results and specifies the needs to refine the results. Keyword-based filtering can provide a set of keywords related to filtering results so that the user can explore the information space by selecting some of them. However, social information has no such concrete feature as keywords. Moreover, social filtering often deals with subjective taste, which is hard to describe as keywords.

Our method, exploration by example, realizes querying and explanation of recommendation by means of examples instead of specific features. It utilizes similarities between items, or users, derived from rating patterns. Exploration is realized as repetition of the user's querying by example and the system's explanation by example (Figure 1).

First, the user composes a query with items or other users; giving items as a query yields items similar to them and users who like them (an item query by item and a user query by item), and giving users as a query yields users similar to them and items they like (a user query by user and an item query by user). The user can select items and users from query results to compose a new query.

Then, the system provides a query result that includes not only items relevant to the query but also less relevant items. The system explains items by showing their relationship to other items and users. This contextual information helps the user to understand filtering results and find candidates for the next query.

A query model for exploration by example is described in Section 3.

2.2. DYNAMIC SEMANTIC MAPS

To visualize filtering (or query) results with contextual information, we have applied a semantic map that automatically composes overall information structure of data by laying them out so that geographic distances approximate semantic distances (i.e., dissimilarity).

Although semantic maps are originally meant to provide a static layout of large data, our map dynamically visualizes filtering results (i.e., recommendation) and query results. The map is rearranged for each query or filtering result. The algorithm to create a dynamic semantic map is described in Section 4.

The map provides overall contextual information and explains each item by visualizing relationship to other items. The user can compose and issue a new query by selecting items on the map.

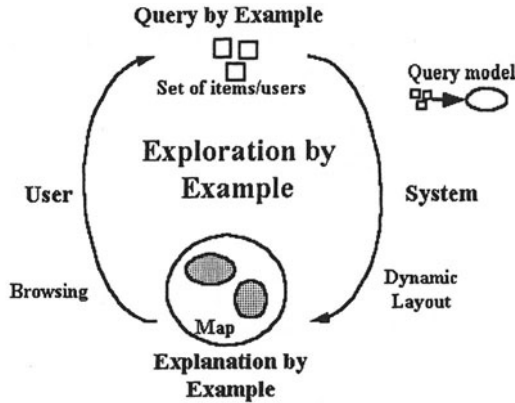


Figure 1 Exploration by Example

A query result is visualized with less similar (relevant) items. Although less relevant items in a query result are generally regarded as noise, they can contribute to compose the contextual information on the map.

2.3. VISUALIZING CONTEXTUAL INFORMATION

The map provides overall contextual information by coloring each item to indicate its rating given by the user. The color pattern of the map shows the following information:

- The scattered pattern of items with high rating clarifies the user's implicit information needs and gives landmarks useful to understand the layout.
- Recommended items can be explained with their spatial relationship to items with rating.
- Outlying items with high rating suggest the user's potential information needs.

Figure 2 is a simplified illustration of a map, which contains two major clusters of nodes with high rating, A and B, and an outlying node P. The clusters indicate that most of the user's interests fall into two categories. If the user is interested in one of these categories more than the other, he or she can browse the recommended items around the cluster. If the user wants something different from the recommendation,

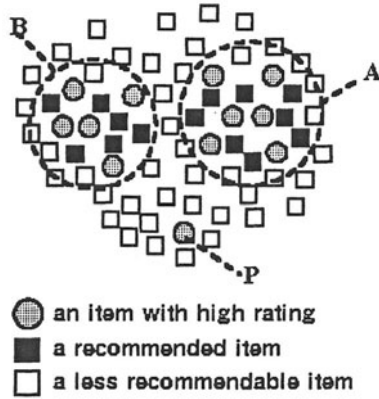


Figure 2 Color pattern of map

the area around the node P can be a starting point for exploration. In a statistical viewpoint, P is regarded as an outlier to be excluded. In the user’s viewpoint, however, the fact that this node lies far from the clusters can be a clue to a new interesting category.

To enable the map to explain more detailed relationship of filtering results, we also have introduced visual interaction techniques: coloring by rating of selected users and highlighting by similarity to selected items as described in Section 5.

3. A QUERY MODEL

Ratings. 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 social exploration, we refer to a user that has given reviews as a “reviewer.” 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.

Similarity Measure. We have applied a simple vector space model to compute similarities based on rating patterns. The feature vectors M_i of the movie i ($i = 1, \dots, n$) and U_j of the user j ($j = 1, \dots, m$) are represented as:

$$M_i = \frac{\langle r_{i1}, \dots, r_{im} \rangle}{\sqrt{r_{i1}^2 + \dots + r_{im}^2}}, \quad U_j = \frac{\langle r_{1j}, \dots, r_{nj} \rangle}{\sqrt{r_{1j}^2 + \dots + r_{nj}^2}}$$

where $r_{ij} = rating(i, j) - 2.5$. Note that, for the movie i that has not been rated by the user j , r_{ij} is considered to be 0. The similarity between two movie vectors or two user vectors is defined as their cosine coefficient (i.e., inner product).

The feature vector to represent multiple movies or users can be defined simply as their summation normalized by the length:

$$and(M_i, M_j) = \frac{M_i + M_j}{|M_i + M_j|}$$

Based on the similarity measure, movies can be retrieved by a query that consists of multiple movies and a certain threshold.

Recommendation. The system can recommend items based on one of common algorithms of social filtering (U. Shardanand and P. Maes, 1995). Users whose similarity to the user i is greater than a certain threshold t are called neighbors of i ($N(i)$). For each item k which is not rated yet by the user i , a predicted rating $p(k, i)$ is computed as the mean of neighbors' ratings weighted by their similarities:

$$p(k, i) = 2.5 + \frac{\sum_{j \in N(i)} sim(U_i, U_j) \cdot r_{kj}}{\sum_{j \in N(i)} |sim(U_i, U_j)|}$$

Focused Recommendation. Given a similarity measure between a movie and a user (reviewer), the system can provide more focused recommendations than those of social filtering results.

The similarity between a movie and a user can be introduced by regarding a user as a vector on the movie space:

$$\langle u_1, \dots, u_m \rangle (u_j = 1, u_k = 0(k \neq j)).$$

Based on this similarity, the user can search movies by users (to obtain recommendation from specific users) or users by movies (to obtain fans of specific movies).

The system provides a focused recommendation as follows. The user issues a user query with movies that interests him or her and obtains reviewers similar to these movies, that is, fans of these movies. By composing a movie query with these reviewers, the user obtains new movies as recommendations from them.

For example, suppose that the user is interested in a movie he or she has seen recently but does not know much about such kind of movies. The user can search fans of the movie by issuing a user query by movie and get recommendation, that is, what else they like and how they comment on them.

4. DYNAMIC SEMANTIC MAPS

4.1. FORCE-DIRECTED LAYOUT

Since the similarity of movies is defined on the user space where the number of dimension is as large as the number of users, the relationship of these similarities can not be perfectly projected on a 2 or 3 dimensional display. To represent the original relationship on the display as complete as possible, several optimization techniques have been developed (X. Lin et al. 1991) (M. Chalmers and P. Chitson, 1992) (J. A. Wise et al., 1995). We have taken one of these techniques, a force-directed layout approach.

There are some studies of heuristic algorithms for drawing undirected graphs based on force-directed models (e.g., (T. M. J. Fruchterman and E. M. Reingold, 1991)). In a typical model, a couple of vertices are linked with a spring whose length represents the ideal distance between them. The vertices are placed in some initial layout and the spring forces on the vertices move the system into a minimal energy state.

We apply this spring model by defining the length of a spring between movies or users as: $len(a, b) = 1 - sim(a, b)$.

The placement of the nodes is optimized by n-body simulation. The simulator moves each node iteratively on the two dimensional display according to the force between nodes.

The force-directed layout approach has the following features that are useful to create a map optimized for a query result dynamically.

- Various constraints and forces can be added to the model. For example, we can fix the place of a particular object to control the layout.
- The iterative process of optimization can be shown as animation.

4.2. SMOOTH TRANSITION OF MAPS

While other applications of a semantic map visualize overall information structure of the entire database, our map is designed to explain query or filter results. Instead of visualizing all the items, a portion of the items is selected from the database to show focused information with contextual information for each query result.

To visualize each query result, the layout is rearranged for the new set of nodes. Note that we have taken an optimization algorithm to lay out the nodes; this algorithm does not always compute a unique layout for a given semantic space and the meaning of spatial position on the map is therefore variable. If each layout is optimized independently, the user cannot understand the new map soon. To enable the user to shift

the viewpoint from the old map to the new map smoothly, the new map should be as consistent with the old map as possible.

The new map usually has some nodes that have already been on the old map. To help the user to understand the new map, the continuity of these nodes is kept as follows: (1) If a node has been shown already on the old map, the system assigns the coordinates on the old map to the initial coordinates of the node. (2) The system visualizes the optimization process of the layout as an animation.

Figure 3 shows an example of shifting a viewpoint by querying. The map on the left is an old map presented before submitting a query to the database. The visualized movies are the result of the previous query “Back to the Future.” The movie “Dial M for Murder” is selected and highlighted as a bigger node on the map. The map on the right in Figure 3 has been obtained by issuing the movie as a query. A dark gray node is an old node that has been shown on the previous map and a light gray node is a new node. The change of the popularity of the movies have pushed the selected node to the center of the map. The area that includes “Back to the Future” on the old map has been shrunk into peripheral area on the new map.

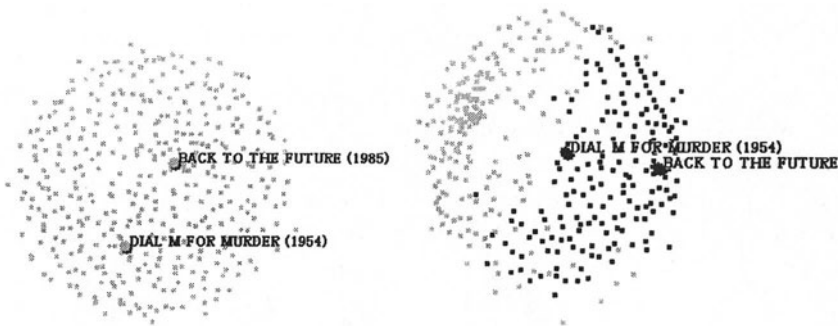


Figure 3 Shifting the viewpoint: (left) before the query; (right) after the query

5. BROWSING A MAP

5.1. VISUALIZING FUNCTIONS

We have introduced two visualizing functions that help the user to browse relationship between items: coloring by rating and highlighting by similarity.

Coloring by Rating. As described in Section 2.3, the map colors the items according to the user’s ratings in order to help the user to understand the layout. The map can also display color patterns according to other users’ ratings. The user can get a list of reviewers as the result of a user query by movie. When some reviewers are selected from the list, the color of each node on the map turns according to their ratings. Given color patterns on the map, the user can understand their tastes of movies and browse movies from their viewpoints.

The color is printed in gray scale in this paper (darker gray indicates higher rating), although we use warm color (red) for high rating and cold color (blue) for low rating on the actual system.

Highlighting by Similarity. The layout of nodes gives only approximated information on the actual semantic distances. To represent more detailed relationship between these nodes, we have introduced a dynamic filter that highlights nodes similar to specified nodes. The user can browse details of the information space by selecting nodes and controlling the threshold of similarity. As the threshold is lowered, the area of highlighted nodes spreads out gradually. Figure 4 shows an example of a map with highlighted nodes. The round, biggest node with a shadow is the selected node and the rectangles with shadows are the highlighted neighbor nodes.

5.2. DYNAMIC QUERYING AND EXPLANATION

Visualizing detailed relationship between nodes as scattered patterns of colored nodes and highlighted nodes, the system helps the user to select known items (i.e., items with the user’s rating) as a query and to understand unknown items (i.e., items without the user’s rating).

Dynamic query by known items. By selecting a known item, the user can dynamically extract similar items from the map. The area of highlight shows which item is near in the original semantic space and eliminates the occlusion of the other irrelevant nodes.

Whenever any combination of the nodes are selected, the area of highlight changes dynamically in order to represent the neighbor of the new selection. The selection of the nodes is therefore regarded as “dynamic query” for the visualized subset of the database.

The highlighted nodes (i.e., dynamic query results) help the user to find items to select next.

- When one of the highlighted node is selected, and the original selections are unselected, the area of highlight shifts to show the neighbor nodes of the new selection. By selecting one highlighted node after another, the user can traverse the map according to the similarity.
- When one of the highlighted node is selected and added to the original selection, the area of highlight shows the result of this more specific query composed with multiple movies. By modifying combination of selected items, the user can refine dynamic query results for a particular need.

After exploring combinations of items, the user can issue a new query with the combination to get a new set of items.

Dynamic explanation for unknown items. By selecting an unknown item, the user can obtain scattered pattern that explains the feature of the item. Since the similar items are scattered over the map according to their similarities to the other items, their positions indicate what kind of movies are similar to the selected movie.

Figure 4 and 5 show snapshots of highlighted patterns of a map that visualizes movies based on user ratings in our movie database. In these snapshots, two different movies, “Back to the Future” and “Star Wars,” are selected respectively. These two sci-fi movies, rated similarly by the users, are laid out close to each other. The highlighted patterns, however, reveals the difference in their details. Although these two patterns share some nodes, they consist of different sets of nodes spreading differently: “Back to the Future” has similar movies in the northeast such as “Top Gun” and “Ghost,” and “Star Wars” has similar movies in the south such as “Blade Runner” and “2001: a Space Odyssey.” In fact, the former is more popular among general fans, while the latter is welcomed by sci-fi specialists.

The explanation given by scattered patterns could not be obtained neither from an ordered list of the similar items nor from an semantic map that includes only the nearest neighbors. While the highlighted nodes provide the focused information, the other, less relevant nodes compose the overall arrangement of the map and provide contextual information.

5.3. OTHER FEATURES

We have implemented a prototype interface based on our method. This interface has the following features in addition to the above two visualizing functions.

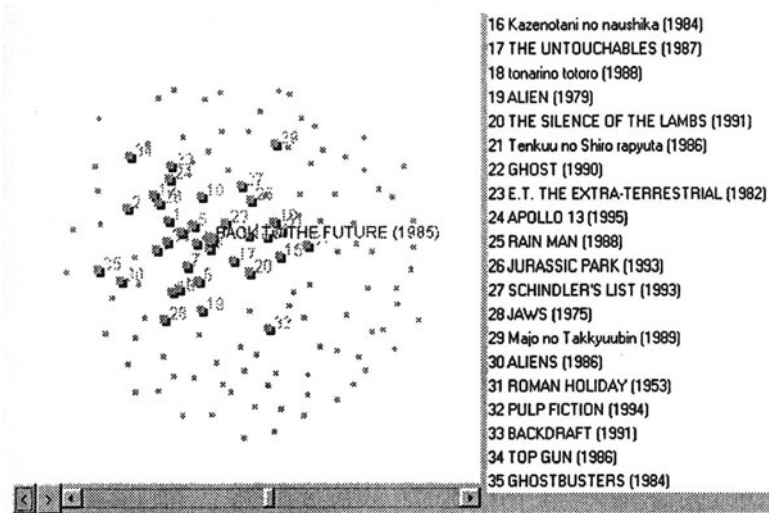


Figure 4 Highlighted items similar to “Back to the Future”

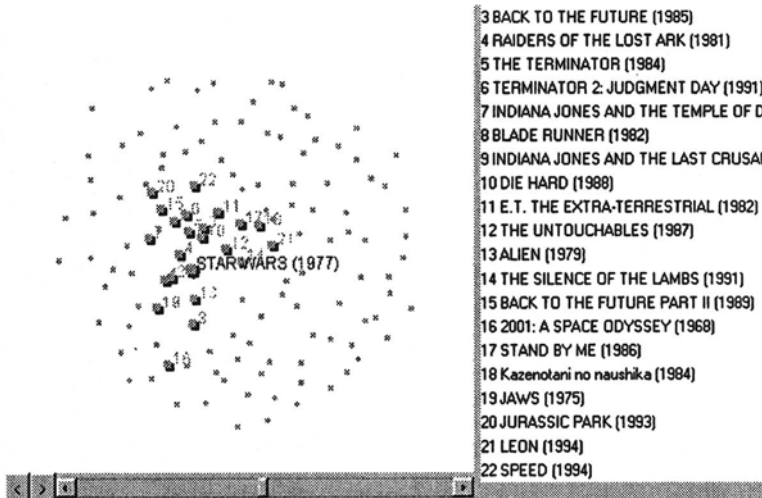


Figure 5 Highlighted items similar to “Star Wars”

Fisheye View. The map uses one of fisheye view techniques that integrate local detail and global context seamlessly on a limited display surface. We adopt a hyperbolic fisheye view (V. Hovestadt, 1995) (J. Lamping et al., 1995) that transforms infinite data planes into a finite circle. Figure 7 demonstrates an example of fisheye views.

Labeling. Each item is visualized as a small rectangle without any label. When the user points an item with the mouse cursor, the system displays the title of the item.

Listing Reviewers. When the user selects items on the map, the system dynamically lists reviewers similar to these items (i.e., fans of these movies). Selecting some of the reviewers colors the map according to their ratings.

Listing Items. The system dynamically lists the titles of highlighted items. Although the user can get the title of an item by pointing the node with the mouse cursor, the list of the titles shows the contents of the area of highlight more concisely.

6. EXAMPLE

In this section, we demonstrate how the dynamic semantic map helps the user to understand filtering results, find items missing from the results, explore for specific needs, and get a focused recommendation.

Visualizing the user's tastes. Figure 6 shows a map visualizing the user's tastes as a color pattern. Two maps in Figure 6 are identical but pointed at different items by the mouse cursor. The map consists of the 600 most popular movies in the database colored by the user's ratings. As can be seen, the map includes some clusters of black nodes (i.e., items with high ratings) indicating areas the user like. At the center of the map, there is a major cluster that includes several dozen of movies the user like. The left map is pointed at one of the movies in the major cluster and displays the title of the movie, "Back to the Future." Far from the major cluster, there are several minor clusters that have only a few movies. On the right map, the user points a movie, "Les Amants du Pont-Neuf," in one of such clusters. This color pattern shows diversity of the user's tastes in movies.

Visualizing recommendation. Figure 7 shows the same map in Figure 6 highlighting the 50 most probably recommendable movies. The recommendations are located around the major cluster of movies the user

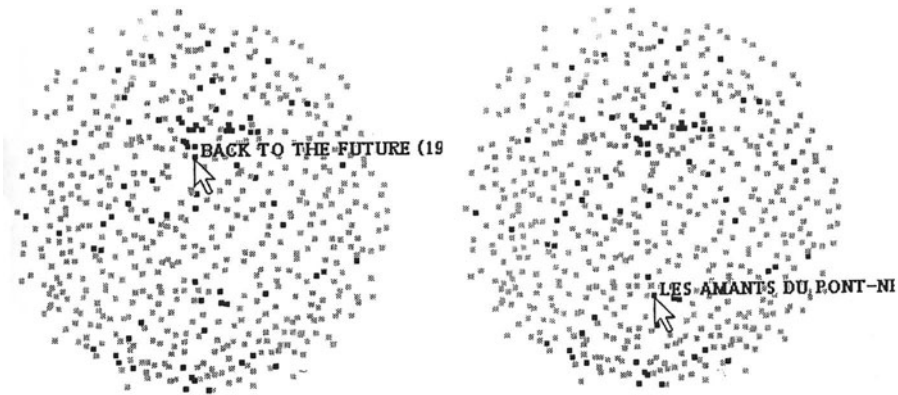


Figure 6 Visualized the user's tastes

likes. The fisheye view can zoom in on the area of the recommendations as can be seen in the right map in Figure 7.

The user can browse each recommendation with contextual information. The location of a recommended movie roughly indicates the relationship to the movies in the major cluster. To examine detailed relationship, the user can select the recommended movie to highlight similar movies.

Revealing overfitted results. Although the filter shows the most probably recommendable items, this recommendation does not cover all the areas of movie the user will like. The area of highlighted movies covers the major cluster of movies with high ratings but miss other small clusters. There are too few rated movies for the filter to recommend movies around there with confidence. Even if the user like movies the filter recommends, he or she might not be satisfied with the result missing some kind of movies the user likes. Visualizing filtering results with contextual information as in Figure 7, the system lets the user aware of this situation.

Exploration for something else. In search for possible interesting movies that have been excluded from the recommendations, the user points a movie, "Smoke," in one of small clusters. The map highlights similar movies: some of the movies have been rated and others are unknown movies. Zooming in on the area around "Smoke," the user can examine relationship between unknown movies and known movies.

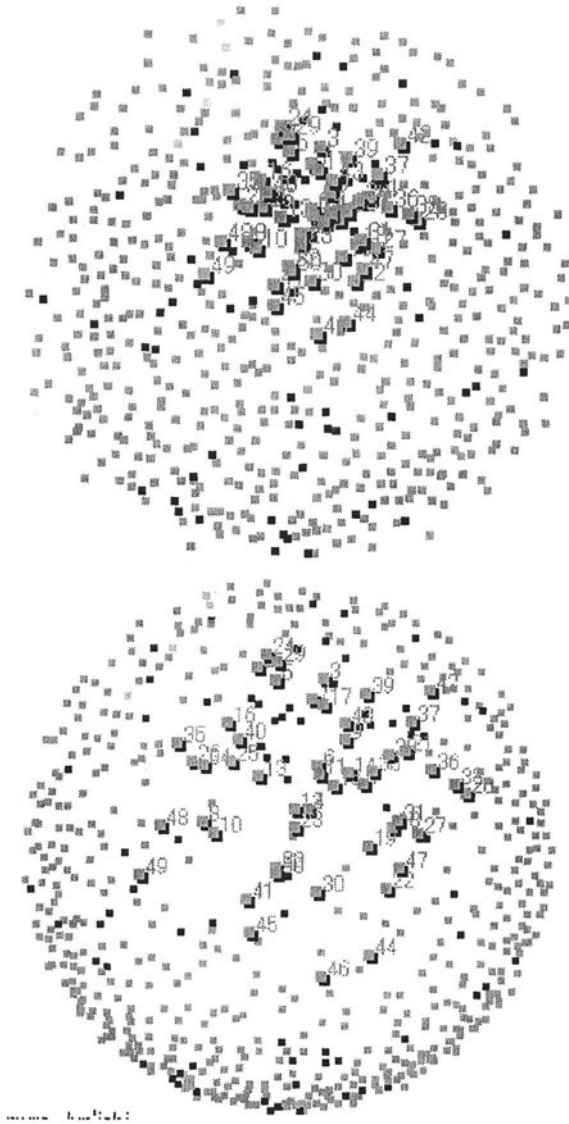


Figure 7 Visualized recommendation: normal view (left), fisheye view (right)

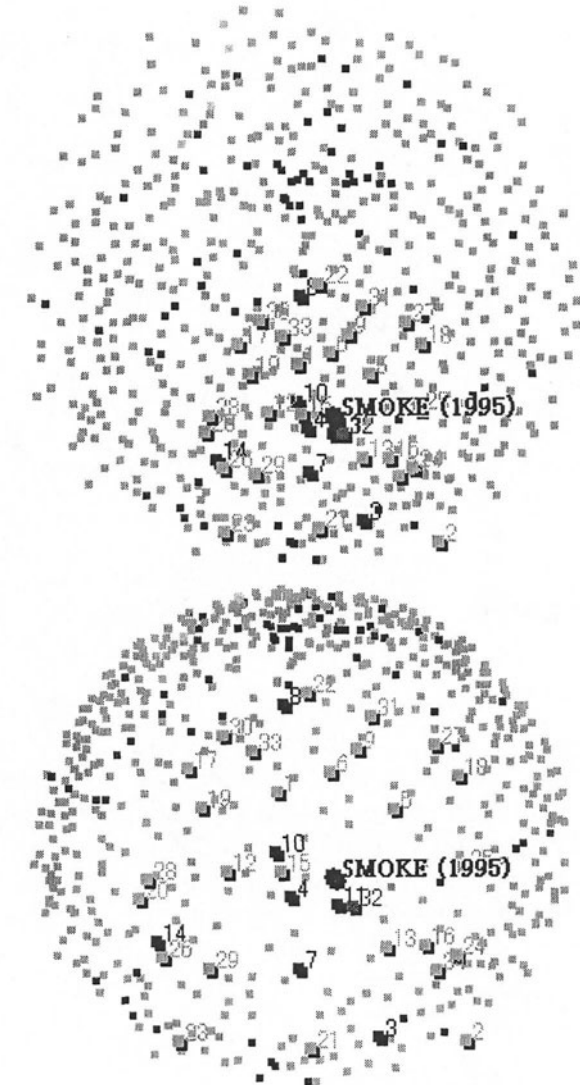


Figure 8 Selecting an item in search for missing items

To refine highlighted movies, that is, dynamic query results, the user select one of the highlighted movies, “Bagdad Cafe,” and obtain a list of movies similar to the selected movies (Figure 9).

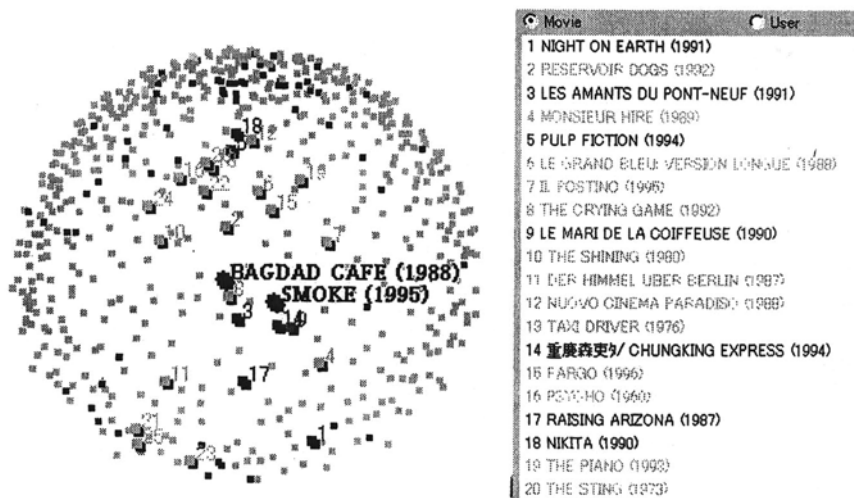


Figure 9 Refining dynamic query results: highlighted items (left), list of the titles (right)

Retrieving reviewers. Whenever the user selects movies on the map, the system dynamically lists reviewers who like these movies. In Figure 10, the system lists reviewers who like both of the selected movies in Figure 9. By selecting some of the reviewers in the list, the user can have the map colored by their ratings (Figure 11). This color pattern represents their tastes of movies. After understanding their tastes, the user can get their recommendations if needed: the user issues a movie query by users and obtains a new map that includes their recommendations.

7. RELATED WORK

Social Query Models. The Tapestry system (D. Goldberg, et al., 1992), 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 acquire articles recommended by a specified person. The system,

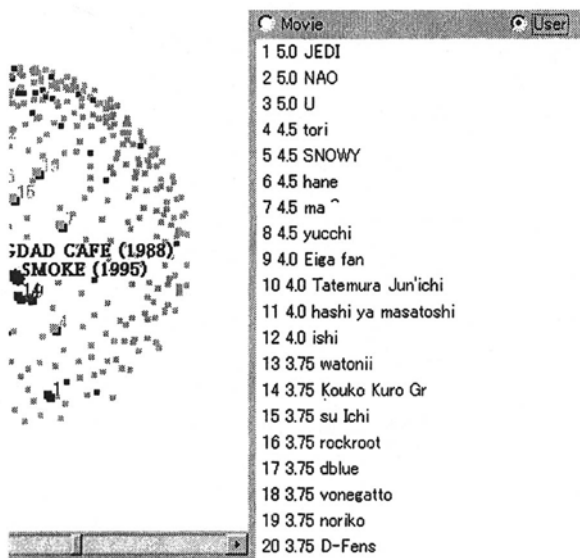


Figure 10 List of reviewers who like the movies of interest

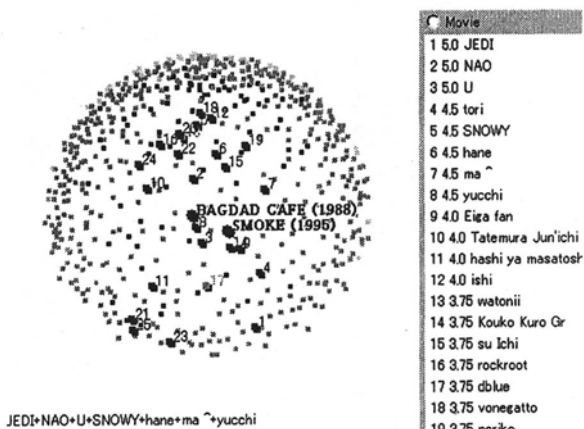


Figure 11 Movies colored by the reviewers' ratings

however, is meant to support a work group among which the user can specify actual users to get information. In a larger and open community on the Internet, the user does not always know appropriate people to ask for recommendation. Our system enables the user to find appropriate reviewers out of unknown reviewers: the user can retrieve reviewers by items of interest and examine their interests or tastes as color patterns.

Visual Querying. VIBE (K. A. Olsen and R. R. Korfhage, 1994) and GUIDO (A. Nuchprayoon and R. R. Korfhage, 1994) are tools to visualize a document collection according to the relationship to multiple keywords (or reference points). VIBE visualizes documents according to a user's "points of interest" defined by keywords and their display positions. Each document are placed at the center of gravity of the keywords which are weighted by the factors derived from a term's frequency and documents are thus clustered geographically according to the positions of selected keywords. GUIDO visualizes the two-dimensional distance space where documents are plotted according to distances from two reference points. These tools can explain the query result by their relationship to the keywords and enable the user to explore the document space by selecting multiple keywords.

One of the principal differences of our system from them is that any of the visualized items can be selected as a reference point with direct manipulation. The movie and user can be both of a reference point and a subject of retrieval. The user can compose reference points by selecting items out of the query results visualized on the map. The result of the selection (i.e., highlighted nodes) even suggests another reference point to select.

This feature of our system is especially important to explore a similarity space when any explicit keyword is not available.

8. FUTURE WORK

Evaluation by user testing. To evaluate the effectiveness of our technique, we need to test the system with users performing tasks. Tasks will include understanding users' tastes visualized on the map, getting recommendation from reviewers with some specific tastes, and serendipitous discovery of movies. Defining performance measures for such tasks is also challenging future work.

Smart labeling of items. One of the problems of a semantic map is a lack of comprehensibility at first sight. Once the user becomes familiar with the overall arrangement of the map, it helps the user to browse items. However, unlike a scatterplot of 2 dimensional data with x-y

axes, a semantic map shows an emerging structure of multi-dimensional data which the user does not know in advance. Landmarks of the map are thus important to help the user to understand the arrangement. Since the current map shows only labels of the item pointed and items selected, the user has to move the cursor around the map to understand the arrangement. To give the user a comprehensive "at-a-glance" map of items, we have to develop a method to select appropriate items for landmarks.

Combination with content-based navigation. The current system uses only social similarity to navigate users. However, rating data is not the only information that represents features of items. In case of our movie database, a movie has attributes such as title, year, countries, genres, directors, and casts. Whereas social information enables the user to explore items and reviews according to subjective tastes, these attributes lets the user explore by specific features. Our next research issue includes developing navigation techniques that use both of social information and content information.

9. CONCLUSION

We proposed a visual exploration technique, "visual exploration by example," for recommendations in a social filtering system. The user's querying by example and the system's explanation by example are repeated on a dynamic semantic map. Rearranged dynamically for each query or filtering result, the map gives explanations of the result with contextual information and helps the user to composition a new query.

We also introduced visual interaction techniques to the map to explain more detailed relationship of filtering results: coloring by rating and highlighting by similarity. Color patterns on the map according to reviewers' ratings help the user to understand their tastes of movies and browse movies from their viewpoints. Highlighted patterns of similar items visualize relationship of selected items to the others.

We demonstrated an example of the user's exploration using a prototype interface for a movie database.

References

- Chalmers, M. and Chitson, P., Bead: Explorations in Information Visualization. *Proc. of ACM SIGIR'92* (1992), pp. 330-337.
- Fruchterman, T. M. J. and Reingold, E. M., Graph Drawing by Force-directed Placement. *Software - Practice and Experience*, vol. 21, No.11 (1991), pp. 1129-1164.

- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, vol. 35, No. 12, pp.61-70, (1992).
- Hill, W., Stead, L. Rosenstein, M. and Furnas, G., Recommending and Evaluating Choices in a Virtual Community of Use, *Proc. of ACM CHI'95*, pp. 194-201 (1995).
- V. Hovestadt, O. Gramberg, and O. Deussen. Hyperbolic User Interfaces for Computer Aided Architectural Design. *ACM CHI'95 Conference Companion*, pp. 304-305 (1995).
- Lin, X., Soergel, D. and Marchionini, G., A Self-organizing Semantic Map for Information Retrieval. *Proc. of ACM SIGIR'91*, pp. 262-269 (1991).
- J. Lamping, R. Rao, and P. Pirolli. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proc. of ACM CHI'95*, 1995.
- Nuchprayoon, A. and Korfhage, R. R., GUIDO, a Visual Tool for retrieving documents. *Proc. of IEEE Symposium on Visual Languages*, pp. 64-71 (1994).
- Olsen, K. A. and Korfhage, R. R., Desktop visualization. *Proc. of IEEE Symposium on Visual Languages*, pp. 239-244 (1994).
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. of ACM CSCW'94*, pp. 175-186 (1994).
- Shardanand, U. and Maes, P., Social Information Filtering: Algorithms for Automating "Word of Mouth", *Proc. of ACM CHI'95*, pp. 210-217 (1995).
- Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A. and Crow, V., Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents. *Proc. of IEEE Information Visualization '95*, pp. 51-58, (1995).

Biographies

Junichi Tatemura is a Lecturer at Institute of Industrial Science, University of Tokyo. He received his Ph.D. degree in Information Engineering from University of Tokyo in 1994. His research interests include user interfaces, information visualization, multimedia information retrieval, social media, and communitywares.