Recommendation Centre: inspecting and controlling recommendations with radial layouts

Gianni Fenu, Lucio Davide Spano
Dipartimento di Matematica e Informatica - Università di Cagliari
Via Ospedale 72, 09124, Cagliari, Italy
{fenu, davide.spano}@unica.it

ABSTRACT
In this paper we propose to use radial layouts for representing the matching between the user's interest and particular objects and/or categories. The technique supports the visualization of different data: we discuss here the relationships on social networks, the related videos on YouTube and topics in Wikipedia. The user can change the position of the object in the representation, which can be used in recommender systems for providing a fine-grained control over its internal preference representation.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous;

Author Keywords
Human Computer Interaction, Recommendation Systems, Visual Interfaces, Radial Layout, Inspection, Control

INTRODUCTION
End users often see recommender systems as black boxes, which suggest them objects, people or concepts while they are trying to find something inside a huge amount of data. On the contrary, recommender systems have difficulties in collecting the user's opinion on the suggested contents, since they mostly rely on explicitly expressed preferences, which are known to be biased [1]. Explicit preferences express love or hate, without helping much for intermediate values. In addition, how to collect the information (e.g. through rating scales) has an impact on the overall recommendation performances [2].

Our position with respect to this problem is that advanced techniques coming from the Human Computer Interaction field may help both the system and the users. A possible solution is to make the two communication endpoints more transparent to each other. If the user would be able to understand, through a simplified representation, how the recommender system is currently "reasoning" while providing suggestions, she would be encouraged in fixing possible prediction errors. On the other hand, the fixing action can be exploited by the system not only for changing a parameter related to a single user, but also for updating future predictions, either for the same or for similar users.

We developed a visualization technique for displaying a summary of the social network interactions through a radial layout [3]. The user can both inspect and control the representation, and the content filtering is updated accordingly. In this paper, we discuss how a similar approach may be applied to recommender systems, in order to support the end-users in understanding their internal state. In addition, the users should be able to modify the position of the object in view. The system should update its internal model accordingly. We describe two early application prototypes and we define the direction of future work.

VISUALIZATION
In this section we discuss the visualization technique, which exploits a radial layout [8] for showing the relationships between object and/or users. It positions a set of nodes, each one representing an object, inside a set of concentric circles. The main node is positioned in the layout centre: it represents the person, object or concept the user is currently focusing-on. The different concentric circles give immediately a feeling of the distance between the main node and the other ones.

Currently, the visualization displays only the nodes that are directly connected with each other. This means that, differently from the original version in [8], the circles are not related to the graph depth, but it represent a weight associated to the edge.

More in detail, the position of a node inside the visualization depends on two factors. The first one is related to the "distance" we want to represent, e.g. how many times we interact with a social network friend, how close a topic is related to another in Wikipedia, etc. We can define different ways for calculating such distances and consequently assign a value graph edges, according to the considered domain. Such definition would position continuously the different objects in the radial layout.

In addition, we included a discretization step in order to help the user in identifying the different levels of relationship, while keeping the visualization tidy. Therefore, the object position depends on discrete distance levels, whose number is estab-

CHI'14, April 26–May 1, 2014, Toronto, Canada.

DOI string from ACM form confirmation

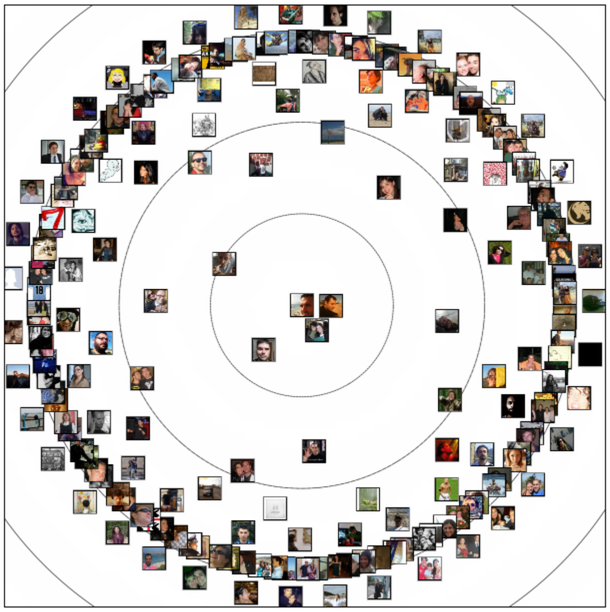

Figure 1. Social Network radial layout visualization

lished according to the application domain. Both the distance and the levels are defined through two functions that control the visualization layout.

In the following sections we discuss the application of the radial layout to different case studies.

EXAMPLE APPLICATIONS

Social Networks

We show the first example in figure 1, where we represented a user's ego network on a social network according to an interaction distance between the main user (show in the centre) and his/her friends.

We represented each friend using a square icon including the profile image. Each icon belong to a different circle according to the distance function value. The continuous distance was defined counting the following events:

1. The friend comments one of the user's posts on her wall
2. The user comments one of the friend's posts on her wall
3. The friend likes one of the user's post on her wall
4. The user likes one of the friend's post on her wall

After this counting step, we normalized the distance value by the maximum value of interactions with a single friend. Such sum gives us a value between 0 and 1 that is higher for friends that communicate with our user very often, and lower for the others.

The visualization confirms the results in [4]: a user communicates often with a small set of friends, while with most of them has less than one interaction per year. In figure 1 most people is contained into the last circles, while in the inner ones are less crowded.

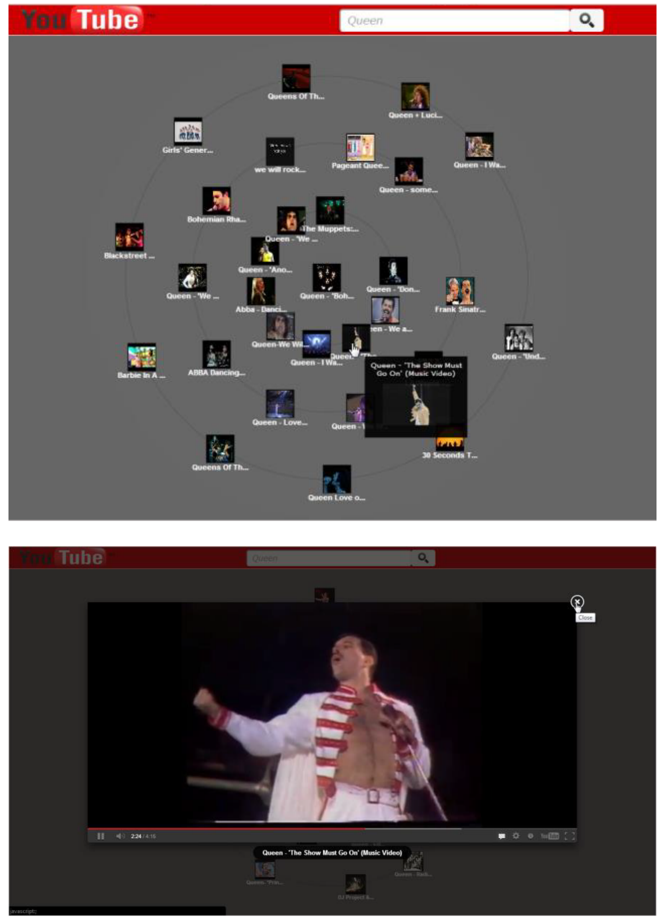

Figure 2. YouTube videos related to the "Queen" keyword

YouTube Videos

In this example, we allow the user to visualize the results of keyword search on YouTube. The resulting visualization is shown in figure 2. The icons are video key-frames, hovering the mouse on top of each video result, the tool shows more information on the selected video, magnifying the key-frame and showing the full title (the bigger icon in figure 2, top part). Clicking on an icon, the tool plays the video, showing it on a modal window.

In this case, we defined the distance function according to three different parameters, which we obtain invoking services from the YouTube Data API v3 [5]:

1. *Relevance*: match between the query and the result.
2. *View count*: number of times the video has been watched by any user.
3. *Date*: publication date.

The three parameters are considered hierarchically in order to establish the visualization distance. This means that we first consider the semantic matching, secondly a crowd-based ranking of the different videos and then we consider the content age.

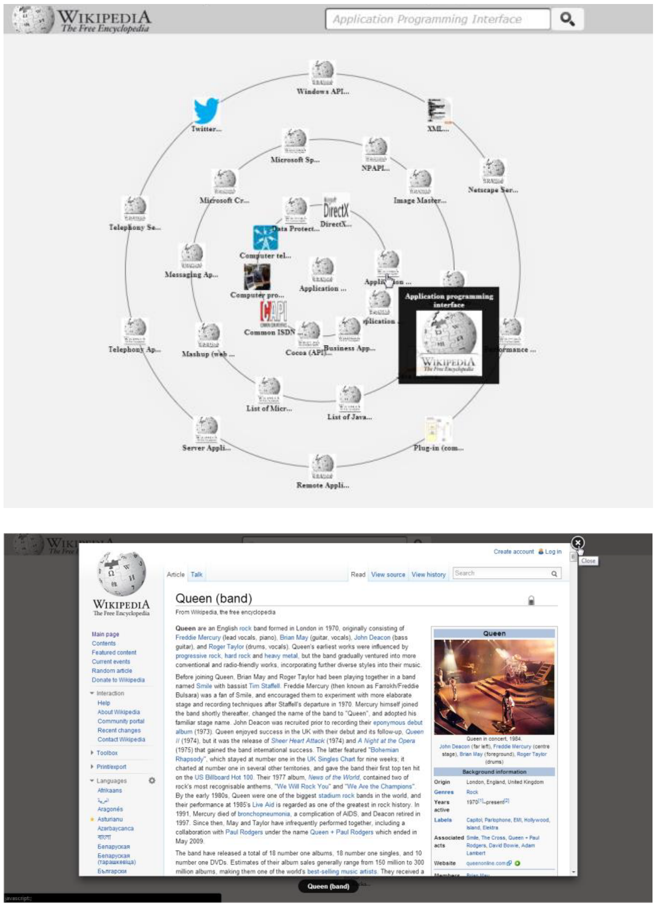

Figure 3. Wikipedia keyword search

Wikipedia

We considered to apply the visualization to the Wikipedia content, in order to apply it in showing the semantic distance between concepts. In this case we used the Wikipedia API [7] for accessing the data.

Similarly to the previous example, we focused on the visualization of a keyword search result. We considered the following properties in order to define the distance function:

1. *Query matching*: the similarity between the Wikipedia page and the keyword
2. *Last page update*.

With respect to the usual result list visualization of the search results, the layout in figure 3 provides the user with the possibility to understand how distant the results are from each other. Indeed, the search result page shows the ranking, but the matching-distance between the results is not uniform. For instance, it is possible that the distance between the 10th and the 11th is smaller than the distance between the first and the second.

The graph nodes are represented through both the Wikipedia article title and its thumbnail image. Since not all articles have an associated image, we used the first image included in the article if any, otherwise we used a default image, i.e. the Wikipedia logo.

As in the YouTube application, the tool shows a small preview when the user overs the mouse on a node, showing the full article title and a bigger thumbnail image (figure 3, top part). In addition, if the user clicks on a node, the tool shows the corresponding article (figure 3, bottom part).

CONTROL FEATURES

The possibility to visualize a distance between friends or objects according to the internal system representation is useful for the user, since it helps her to understand what the recommendation support has learned from the data analysis. However, this is not enough: users may want to change the system internal representation when she is not satisfied with it.

This would have a twofold positive effect on the interaction. On the one hand, the system would gain an explicit feedback, and this would be useful for both creating a more precise user's model. In addition, the same feedback can be propagated to similar users. On the other hand, the user will be more satisfied with a system that allows her to change the representation of her interests, which would result in more relevant recommendations.

Considering this, we inserted in the visualization tool the possibility for the user to change the node position. We show an example of this manipulation in figure 4. The user selects one of the nodes in the visualization, and then she can change the position inside the distance levels either dragging the node or changing the slidebar values.

Such action has an effect not only in the visualization, but it can be exploited also by the recommender system for updating its internal representation. Indeed, the system may invert the distance function and let the user to specify directly the matching value, without the need for her to understand how the system internally calculates it.

In figure 4 example, we show a sample case for such manipulation. The user, inspecting his social network, sees that one of his best friends is quite distant from him. They do communicate few times through the social network, but they see each other at least once or twice a day. So the user decides to change his friend's position. The system updates his internal representation consequently.

This has an impact for instance on the content that the social network application shows on the user's news feed: the content published by the considered friend should be visualised immediately in the first positions, even if from the collected data the interaction between the two users is weak.

CONCLUSION AND FUTURE WORK

In this paper we discussed a simple example application of Human Computer Interaction techniques for increasing the user's understanding of a recommender system. In our opinion, providing simple yet effective visualization of the their internal state to the user may have different advantages.

First of all, the user would be able to inspect the recommender system state and to fix possible prediction errors that cause

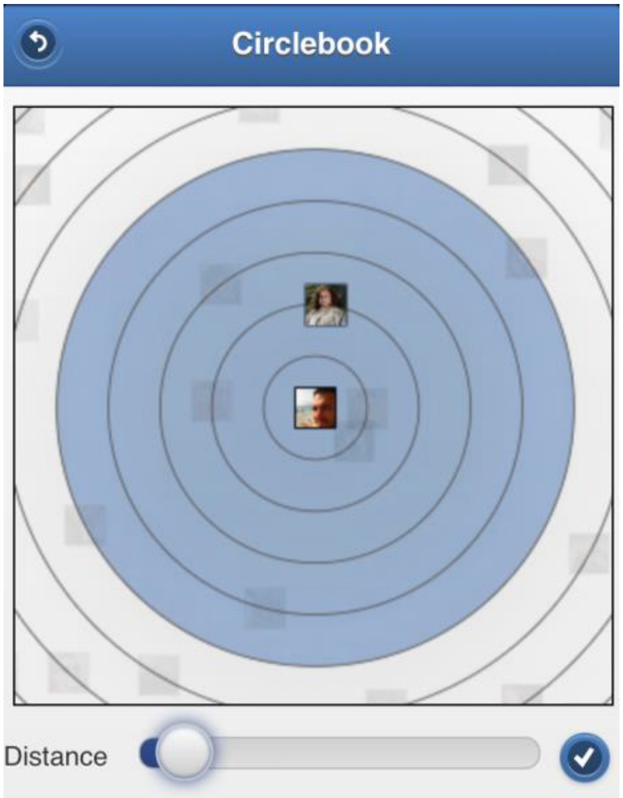

Figure 4. Distance control functionality

incorrect suggestions. While the user would receive better content, the recommender system would learn from the user's feedback and use it also for similar users. In addition, the user would trust more a system that explains how it suggests a content, with respect to other ones where she cannot find out if it is relevant for her or it is simply advertised.

We described an early application of a radial visualization from the distances between users in the same social network to contents such as videos and Wikipedia articles. In addition, we discussed how control techniques on such visualization may have impact on the recommender system data.

In future work, we plan to study more in detail the End User Development techniques [6] that may be used for defining other internal aspects, such as recommendation algorithms and data collection. In this case the user would not develop new algorithms or directly manipulate the data, but it would be useful for graphically describing how the system work. This would guide further user's control actions on the recommendation interface.

REFERENCES

1. Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. 2009. *User Modeling, Adaptation, and Personalization: 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems, 247–258. DOI: http://dx.doi.org/10.1007/978-3-642-02247-0_24

2. Paolo Cremonesi, Franca Garzottto, and Roberto Turrin. 2012. User Effort vs. Accuracy in Rating-based Elicitation. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, New York, NY, USA, 27–34. DOI: http://dx.doi.org/10.1145/2365952.2365963

3. Gianni Fenu and Lucio Davide Spano. 2013. *Mobile Web Information Systems: 10th International Conference, MobiWIS 2013, Paphos, Cyprus, August 26-29, 2013. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter Circlebook: Visual Display of Friend Proximity, 129–142. DOI: http://dx.doi.org/10.1007/978-3-642-40276-0_11

4. Scott A. Golder, Dennis M. Wilkinson, and Bernardo A. Huberman. 2007. *Communities and Technologies 2007: Proceedings of the Third Communities and Technologies Conference, Michigan State University 2007*. Springer London, London, Chapter Rhythms of Social Interaction: Messaging Within a Massive Online Network, 41–66. DOI:http://dx.doi.org/10.1007/978-1-84628-905-7_3

5. Google. 2014. YouTube API v3. (2014). https://developers.google.com/youtube/v3/ accessed 6 April 2016.

6. Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. 2006. *End User Development*. Springer Netherlands, Dordrecht, Chapter End-User Development: An Emerging Paradigm, 1–8. DOI: http://dx.doi.org/10.1007/1-4020-5386-X_1

7. MediaWiki. 2016. Wikipedia API. (2016). https://www.mediawiki.org/wiki/API:Main_page/it accessed 6 April 2016.

8. Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst. 2001. Animated exploration of dynamic graphs with radial layout. In *InfoVis*. IEEE, 43.