Mangaki: an Anime/Manga Recommender System with Fast Preference Elicitation

Jill-Jênn Vie
LRI – Bât. 650 Ada Lovelace
Université Paris-Sud
91405 Orsay, France
jj@mangaki.fr

Camille Laïly
Mines-ParisTech
60 boulevard Saint-Michel
75005 Paris, France
camille@mangaki.fr

Solène Pichereau
Éditions Soleil Manga
8 rue Léon Jouhaux
75012 Paris, France
solene@mangaki.fr

ABSTRACT

Rapid profiling of new users by a recommender system is a known issue. In this demo, we present Mangaki, a manga and anime recommender system that implements a preference elicitation process based on welcome decks. We describe its design and interface.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

Keywords

Preference elicitation interface, recommender system.

1. INTRODUCTION

A large community is crucial for a healthy recommender system, thus a good strategy for preference elicitation is to ensure that a new user won't be annoyed by a long signup process and that he won't lose confidence in the system due to early recommendations of low quality.

To tackle this so-called *user cold start problem*, one can resort to decision trees to bootstrap the system [2]. Many other criteria have been tested, such as minimizing the entropy or information gain through clustered neighbors [3]. In our case, we also want to provide a selection item rule that is fast enough to compute so as not to annoy the user with loading issues.

The community we address here is: anime and manga fans. As an indication, 10,664 anime and 35,623 manga items are listed on the website myAnimeList, the most popular anime (*Death Note*) being rated by 625,366 members and the most popular manga (*Naruto*) being rated by 156,149 members. To these ends, we developed *Mangaki*, one of the first websites in the field of manga and anime recommendation.

The key feature in our system is a simple preference elicitation process that aims to harvest a maximum of information while guaranteeing an enjoyable first-user experience.

2. SYSTEM ARCHITECTURE

When the user enters the system for the first time, they are presented with a selection of items to rate that we will call *welcome decks*. While they rate items, other items appear. Once they can't rate anything, they are shipped to a page that provides recommendations. This simple architecture can be adapted to other types of items.

3. DESIGN

3.1 Tag-Based Rating System

Users can rate any item, anime or manga, using the following tags: Like, Neutral, Dislike if they know the item, Willsee if they intend to watch it, Wontsee if they do not.

We chose not to allow users to rate on a scale from 0 to 10 because people have various notation scales and normalization can misshape the data. For example, if a user rates anime from 8 to 10 over 10, it does not necessarily mean that they dislike anime rated 8.

In order to compute similarity between users, tags are mapped into numeric values as we will show hereafter.

3.2 Content Filters

A user can browse the item list using a few content filters: most popular items, most controversial, most liked and a few *random jewels*: items that got no Dislike rating and at least 3 Like ratings. Most liked items contains only items that got at most 5 Dislike ratings.

For the controversy filter, if L denotes the number of Like and D the number of Dislike of an item, the controversy score also used in the open-source platform Reddit [1] is given by the following expression:

$$\mathit{controversy} = \begin{cases} (L+D)^{\min(L/D, D/L)} & \text{if } L, D \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

4. WELCOME DECK INTERFACE

When a user starts interacting with Mangaki, they are presented with a set of four decks of items, selected on the base of the community ratings presented above: "most popular", "most controversial", "most liked", and "random jewels". For example in Figure 1 corresponding to a new user entering the system, *Death Note* is the most popular anime with 243 ratings, *Naruto* is the most controversial with 30 Like and 27 Dislike, *Spirited Away* is the most liked item (83 / 3) and *Castle in the Sky* is a random jewel (50 / 0). Every time an item is rated, it is removed from the deck and the next one appears, like in a set of cards. The system is guaranteed not to show an item that has already been rated, and there is no duplicate among all four decks.

After this first shot, the user can refine they profile to add more items by going back to the Welcome Deck, browsing the database, or searching directly for a name. On the Mangaki version, from a great user request, we implemented a

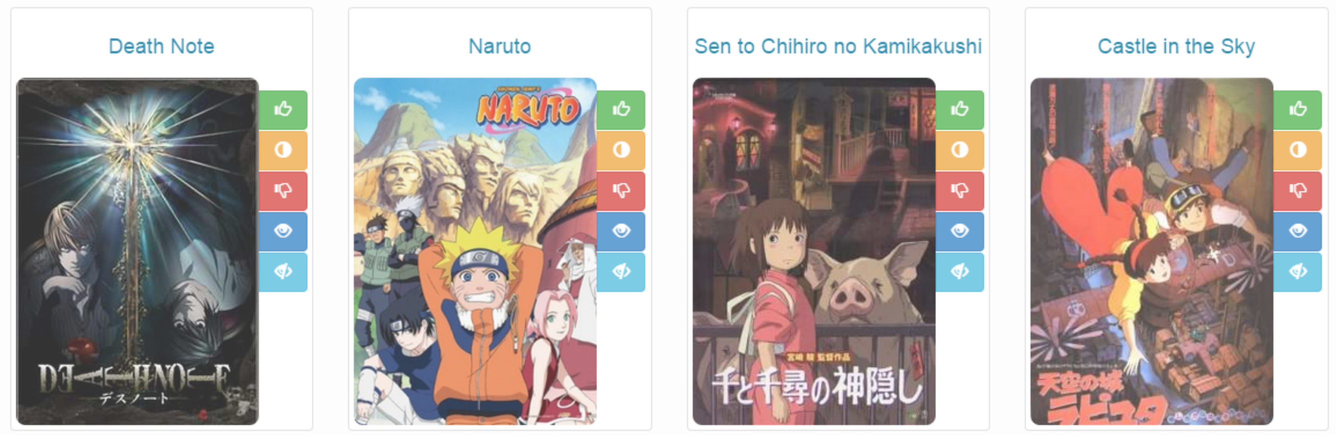

Figure 1: Screenshot of the welcome decks.

possible aspiration of a user profile on MyAnimeList, a famous anime database site.

The most informative deck is the controversial one. But we make the assumption that if the user is not allowed to rate popular or liked works, its reward will be lower. From another side, the last deck browses mostly-liked uncommon works, which will be appreciated by aesthetes.

5. RECOMMENDATION ALGORITHM

Ratings are mapped to numbers the following way:

Like	Dislike	Neutral	Willsee	Wontsee
2	-2	0.1	0.5	-0.5

Thus, the similarity between users is computed via dot product and we can use a variant of the k-nearest neighbor algorithm for item recommendation. Willsee and Wontsee ratings are not used for recommendation but for similarity.

6. CONCLUSION

We presented Mangaki, a website that recommends anime and manga works after a fast preference elicitation phase. A demo of the system is available at `http://mangaki.fr/static/demo.mp4`.

As of April 30, 2015, Mangaki hosts 41,602 ratings by 333 users. We plan to study user behavior in order to estimate how many items they rate during the welcome deck phase and if they are satisfied with the recommendations provided right after. We also want to know how many items they rate per column and what it says about their tastes.

7. REFERENCES

[1] Reddit's Cython code for controversy sort. `https://github.com/reddit/reddit/blob/2620100c37e4f12e04612072abc544d84e7e7b82/r2/r2/lib/db/_sorts.pyx#L59`. Accessed: 2015-04-30.

[2] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 595–604. ACM, 2011.

[3] A. M. Rashid, G. Karypis, and J. Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *ACM SIGKDD Explorations Newsletter*, 10(2):90–100, 2008.