In the Mood4: Recommendation by Examples

Rubi Boim and Tova Milo
School of Computer Science,
Tel Aviv University
boim@post.tau.ac.il, milo@cs.tau.ac.il

ABSTRACT

Traditional recommender systems generate personalized recommendations based on a profile that they create for each user. We argue here that such profiles are often too coarse to capture the current user's state of mind and desire. For example, a serious user that usually prefers documentary features may, at the end of a long and tiring conference, be in the mood for a lighter entertaining movie, not captured by her usual profile. As communicating one's state of mind to a system in (key)words may be difficult, we present in this demo Mood4 - a novel plug-in for recommender systems, which allows users to describe their current desire/mood through examples. Mood4 utilizes the user's examples to refine the recommendations generated by a given recommender system, considering several, possibly competing, desired properties of the recommended items set (rating, diversity, coverage). The system uses a novel algorithm, based on a simple geometric representation of the items, which allows for efficient processing and the generation of suitable recommendations even in the absence of semantic information.

1. INTRODUCTION

Recommendations are an integral ingredient of almost any everyday task; websites provide suggestions for related pages at the bottom of the page, app stores provide users with application suggestions, movies are recommended to potential viewers, etc. Providing good and relevant recommendations is clearly an important challenge.

Recommender systems aim to provide *personalized* recommendations by capturing the user's *taste*. Solutions range from ones using semantic properties of users and items (e.g. age, genre, etc.) to semantic-less ones such as Collaborative Filtering that are based only on users scores for items (e.g. "people who liked this set of items also liked...")[8]. While many recent works attempted to improve the accuracy of such systems, we argue that the granularity of "user's taste" that they capture is too coarse. Indeed, none of these methods, sophisticated as they might be, captures the user's *current "state of mind"*, or her *current "mood"*. For instance, a user might usually prefer documentary features, but before a date with her boyfriend she might be in a lighter romantic mood and prefer recommendations for a romantic comedy. Alternatively, the same user may be sharing her laptop with her daughter and thus get recommendations for the new "Harry Potter" book, rather than ones that interest her, because her daughter often uses her laptop and thus the profile is more affected by her choices.

A key difficulty in providing suitable recommendations to users in such a scenario is that it is not always easy for a user to describe her current mood/desire to the system in (key)words. Indeed, recent works suggested the use of an *example*, instead of verbal description [1]. For instance, the *Pandora Internet Radio* [1] asks users for an example of a song they would want to hear, then attempts to generate a playlist of similar songs. (We will discuss what "similar" means later).

But is a single example indeed enough to describe the user's current state of mind/mood? We argue in this work that the answer is *No*. A user, for instance, might have had a long day at work and is interested in watching a "light" movie (that is, an enjoyable movie which does not require the full attention of its viewers). Capturing such a desire with a single movie example is hard, as there are different kinds of light movies, e.g. of different genres. If she gives, for instance "American Pie" - a comedy - as a (single) example, or alternatively "The Rock" - an action movie, the two recommendation lists would be very different: In the first case it is likely to consist only of comedy-like movies, whereas in the second case of only action-like features. While each individual list indeed contains relevant items, it clearly does not cover the relevant spectrum. The overspecialization may further yield improper recommendations, e.g. "heavy" action movies like "No Country for Old Men". Indeed, what is desirable here is to use jointly the two examples above to capture the user desire more accurately, recommending action-comedy features like "Lethal Weapon" or "Bad Boys". (We refer below to such items as *joint representatives*).

In this demo we present Mood4 - a novel plug-in for recommender systems, which allows capturing the user's current desire (mood) through multiple examples. Mood4 utilizes the user's examples to refine the recommendations generated by a given recommender system, by considering several, possibly competing, properties of the proposed items (to be further discussed in the sequel); (a) the *similarity* to the given individual examples, (b) the *joint similarity* to subsets of the examples, as illustrated above, (c) the (possibly personalized) items scores - often called *rating* - as computed by the given recommender system (items with higher rating are more relevant), (d) the *diversity* of the recommended items, and (e) their *coverage* of the examples.

Mood4 integrates several existing technologies in a new algorithm to produce a user mood-specific recommendations list. Given the set of user examples, Mood4 takes a Collaborative Filtering

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

(CF) approach [8] to evaluate the similarity between the given examples and the items in the database, without requiring any semantic information. (We explain briefly in the sequel how CF works). Mood4 then creates a geometric representation of the items space, in order to find the joint representatives mentioned above (also to be discussed in the sequel) which play a key role in the algorithm. The geometric representation is then updated and blended with the (personalized) rating of each item, as given by the underlying recommender system, in preparation for the final weighting mechanism. Finally, Mood4 uses a novel technique from [5], based on priority-medoids, to diversify the recommendations presented to the user.

Figure 1 depicts a screenshot of Mood4 when used in the context of movie recommendations, following the example of the "light-viewing" mood discussed above. "American Pie" (comedy), "The Rock" (action) and "Independence Day" (action) are the examples chosen by the user to describe her current desire/state of mind (on the left part of the screen). The recommendation generated by Mood4 are presented on the main screen, and indeed include joint representatives such as "Bad boys" and "Lethal Weapon" (both are action-comedy) . Interestingly, Mood4 also captures a different kind of relationship - one that is not only based on the genre, but also on the casting: "Hitch" is a comedy movie, and its leading actor is Will Smith, which stars in "Independence Day". This is a especially interesting as it is achieved in spite of the fact that no semantic information on movies is used by Mood4 , and is due to the power of CF. Finally, Mood4 also provides two useful features to its users: a visual explanation for each recommendation (that is, why Mood4 believes this recommendation to be relevant), and a "zoom-in" facility attached to each recommended item, allowing users to further explore similar recommended items.

Outline of the demonstration. We demonstrate the operation of Mood4 in the context of movie recommendations. We use a real data set by Netflix [3] which contains more than 100 million ratings of users to movies. Mood4 is implemented as a smartphone/tablet web application and the first part of the demonstration will be held *on the attendees' own smartphones or tablets* (we will provide a few devices for the attendees that do not have such devices). After a quick setup (no installation is required) and a brief description of Mood4 , we ask each of the attendees to select 2-4 movies that capture their current mood and view the recommendations generated by Mood4 . The application also allows to compare the generated list to what would have been otherwise proposed by the underlying (Mood4 -less) recommender system, as well as to recommendations generated by focusing only on individual examples and to a less sophisticated variation of our algorithm that does not exploit the joint representatives.

The second part of the demonstration will then show (on a laptop and a larger screen) what happens "behind-the-scenes". We will pick one volunteer from the attendees and illustrate (and analyze) the different steps of the algorithm, for her individual set of examples, explaining how together they yield a desired mood-aware recommendations list.

2. TECHNICAL BACKGROUND

Let I be a finite domain of items and let $S \subseteq I$ be the set of example provided by the user. We refer to the items in S as *seeds*. Our goal is to find a subset $I_k \subset I$, of size k, of items to recommend the user, matching best the user's mood as reflected by the provided seeds.

We briefly overview below the algorithm used by Mood4 ; we start with a brief description of CF and some notation. We then describe the geometric representation being used and explain how

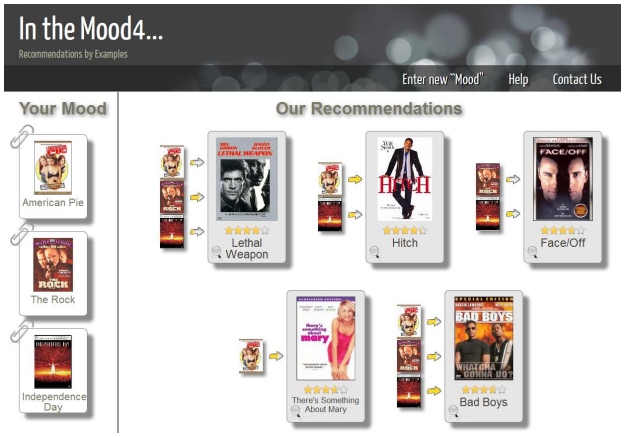


Figure 1: Mood4 's main screen

joint representatives are selected. Finally we discuss the employed items weighting mechanism and the diversification element.

Collaborative Filtering. Collaborative Filtering (CF) is a method of making automatic predictions, and thereby recommendations, for how much a user would like a given item, based on the ratings that other (similar) users gave to (similar) item. Unlike semantic-based systems, where recommendations are made by analyzing the (semantic) properties of each item (such as color, genre, size, etc.), CF utilizes only raw user ratings (such as 1 to 5 stars). Intuitively, it is based on the assumption that users who agreed in the past on items ratings are likely to agree again in the future. Recommendations are made first by the estimation of unknown ratings, that is the items the user have yet rated, and then by the selection of the items with the highest estimated rating. (Intuitively, these items are the ones the system believes the user would like best).

The main component in CF system is the similarity estimation between two items (users). Intuitively, each item is viewed as a vector of ratings in a multi-dimensional space, where each dimension represents the rating of the user corresponding to that dimension. (The similarity between users is analogously defined). Similarity between items is then evaluated by measuring the distance between different vectors, by some distance measure such as cosine or Pearson's correlation coefficient [7]. (The latter is the preferred choice of most recommender systems today, and the one used in our system). Rating predictions are traditionally computed in two steps: First, we search for a neighborhood of items, similar to the given one, that have already been rated. Then the predicted rating is computed by aggregating (e.g. averaging) the known ratings of the neighborhood members. In the reminder of this paper we denote by $rate(u, i)$ the rating of an item i by a user u. For simplicity, we assume the current user is known from the context and simply use $rate(i)$. We denote the similarity between two items i and j by $sim(i, j)$. W.l.o.g. we assume below that distance values are in the range of $[0, 1]$. (When this is not the case one may naturally map the values to this range). The larger the value, the more similar are the items.

Geometric Representation. Our algorithm starts by computing, for each seed item $s \in S$, a neighborhood $N(s)$ consisting of the m items most similar to it, for some constant value m (m is the only global constant used in our algorithm, and is set to 50 in this demonstration). We then look at the geometrical representation of the seeds and their neighborhoods. Note that, in general, sim is not always a metric function. But to simplify the presentation we

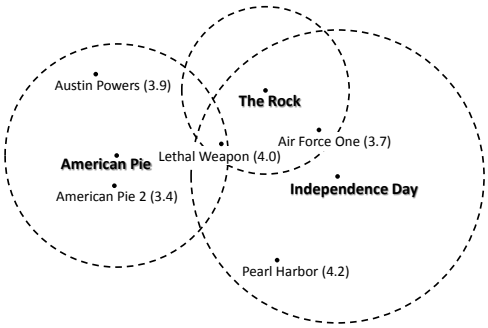
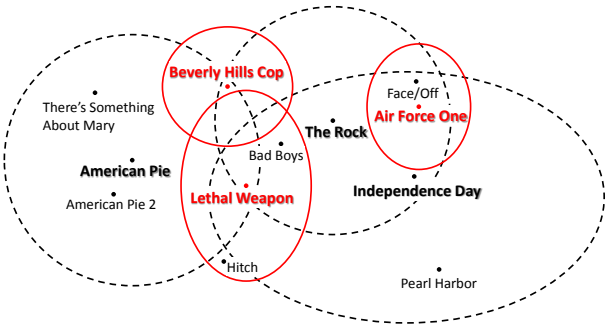


Figure 2: (a) Geometric Representation (partial)



(b) Geometric Representation with Joint Representatives (partial)

will assume below that it is, and display the geometric representation on a simple 2-d plan, where the geometric distance between items represents the sim function (items with a higher similarity measure will appear closer). Figure 2 (a) depict the geometric representation of the example discussed in the Introduction. The rating of each item (as evaluated by the underlying recommender system) is given next to the item's name. The neighborhood of each seed item defines a *circle* where the corresponding seed is the center and the radius is defined by the least similar item in its corresponding neighborhood. (The radius may be different for different circles).

Joint Representatives. To identify joint representatives, Mood4 next searches for items that capture the mood reflected by multiple seeds. In the geometric representation, these are the items located in the intersection of multiple circles. Each such area corresponds to a subset of seeds that are the center of the overlapping circles. Intuitively, the items in each such area are similar (only) to all seeds of the corresponding subset. For example, in Figure 2 (a) American Pie 2 is indeed similar only to the seed American Pie, but Air Force One is similar to both The Rock and Independence Day.

For each subset of seed items, corresponding to the overlapping areas discussed above, we wish to find the item that "represents" it the most. Formally, for a subset of seed $S' \subseteq S$ we define the joint representative item to be the one that maximizes the value of the following equation:

$$JointRep(S') = \max_{i \in N(s) | s \in S'} [\min_{s \in S'} [\frac{sim(i,s) * rate(i)}{avg_{j \in N(s)}(rate(j))}]]$$

The formula captures the following intuition: Intuitively, the joint representative is the item which resides in the center of the geometric area, with respect to the subset of seeds. To find this item, we search the candidate items (the ones in the corresponding neighborhoods) for the item whose minimal similarity among the subset of seeds is the maximal. In order to also take item ratings into consideration, we combine the similarity measure with the rating of each item and divide it by the mean rating of the neighborhood of the corresponding seed.

Weighting. We now augment the seeds set by the selected joint representatives[1] and rebuild the geometric representation for the extended seeds set. To take the items rating into consideration, we combine (multiply) as above the similarity and the rating of the items, then normalize the results not only by the mean rating, but also by the mean similarity of the neighborhood of the corresponding seed item. This refined similarity measure between an item i and a seed s is given by:

$$Sim'(i,s) = \frac{sim(i,s) * rate(i)}{avg_{j \in N(s)}(sim(j,s)) * avg_{j \in N(s)}(rate(j))}$$

It will (i) better help our algorithm merge the neighborhoods on the next step, and (ii) will prevent biased towards highly rated items.

Figure 2 (b) depicts the updated plot which includes the joint representatives and the updated similarity measure. The joint representatives added by Mood4 are in red, and their neighborhoods are surrounded by solid circles (as opposed to the dashed ones surrounding the seeds). Note that these neighborhoods may include new items - that is, ones which were not members in any of the original neighborhoods. "Hitch" in this current scenario is such an example. Also note that the circle symmetry shown before in Figure 2 (a) is not preserved as the similarity measure is now blended with the rating. The rating next to each item, shown in (a), is now redundant and thus omitted.

Diversification. The center of the circles (the original seeds and the joint representatives) describe best the users's current mood. Intuitively, we would like to recommend a set of items which are close to all centers. As the size of the screen is limited, especially on mobile devices, this set is relatively small (usually $k = 5$). To choose the right items we first create a sorted list of the neighborhood items, giving priority to items that appear in more neighborhoods, and within items that appear in the same number of neighborhoods, we sort the items by their distance to the seed closest to them.

Note that simply choosing the top-k items may lead to an unattractive overly homogenous set of recommendations; for example, in the movie domain a set all consisting of sequel movies, like all the Star Wars saga. Clearly the user would prefer to be presented with a wider and more diverse subset of the highly rated items (possibly with an option to view more sequels via a click on a "more of that" button if she desires). Mood4 archives this by employing the diversification mechanism that we developed in [5].

Related work. Much of the recent academic research in this area focuses on improving the rating estimation of recommender systems [8, 6]. Most relevant to our work is [2] where the authors consider the problem of group-recommendations, i.e. identifying items suitable for a group of people. While one may view each group member as an "example", the algorithms developed in [2] are not suitable here as they focus on reducing the group disagreement (e.g. maximizing the similarity to all the seeds) not incorporating the other properties discussed above. From the industry, Pandora [1] is a good example for a system that generates recommendations by asking the user to enter a sample song (or an artist) she likes, generating in response a playlist of similar tunes. Unlike Mood4 ,

[1]For space constraints we omit several steps of the algorithm that avoid considering joint representatives which are too close to the original seeds

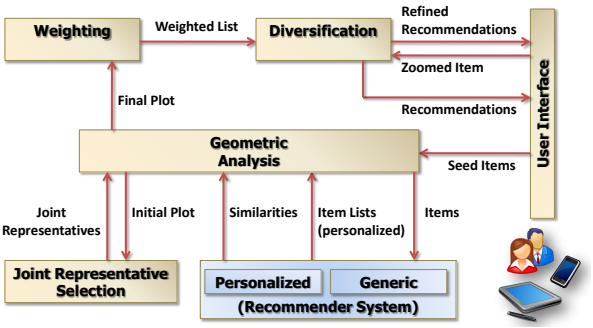


Figure 3: System Diagram

Pandora only focuses on a single example at a time (even with its new "variety" option), and does not try to find overlapping features as Mood4 .

3. SYSTEM OVERVIEW

Mood4 is implemented in a client-server architecture and is designed for *large-scale* usage. Its core algorithms are written in Java and the user interface (UI) is accessible via HTML and iOS. We stress that Mood4 is not intended to replace a fully working recommender system, but rather to enrich it with the option to capture the current user's state of mind/mood. We next provide an overview of the main system components. These components, and their connections, are depicted in Figure 3.

Geometric Analysis. This component is responsible for normalizing the similarity/rating measures and building the geometric representations discussed on the previous section.

Recommender System. This is the underlying CF-based recommender system which Mood4 enriches. It is assumed to provide a similarity function to compare two given items and a personalized score (rating) to each user-item pair. When no information is available about a given user (e.g. users that just started using the system), the recommender system generates a generic score for each item, which Mood4 later caches to improve performance.

Joint Representative Selection. This component receives the basic geometric representation and, based on it, identifies the joint representatives. They are then passed back to the Geometric Analysis component for the evaluation of the final geometric plot.

Weighting. This component receives the refined geometric representation and evaluates for each item its final weight based on it's location in the space. It then computes the sorted items list.

Diversification. This component finally decides which items will be indeed presented on the user's screen in order to provide a diversified set of recommendations and supports a "zoom-in" mechanism, allowing users to view similar recommended items.

User Interface. This component provides the connection between the core program written in Java (the server) and the end users (the clients) in both HTML and iOS interfaces. The UI provides an intuitive presentation of the recommendations, their justifications, and the "zoom-in" facility (Figure 1).

4. DEMONSTRATION

We start with a brief description of the settings used in the demonstration, then describe the interactive demonstration scenario.

Settings. The algorithms used by Mood4 do not assume any semantic structure of the data, and thus, Mood4 could be deployed on any domain desired. For demonstration purposes, we chose to use the familiar cinema domain, that allows the attendees to better judge the resulting recommendations. We use the Netflix data set [3] which is considered the industry standard in such scenarios. The data set consists of over 100 millions unique ratings given by more than 500,000 users to approximately 18,000 movies. This data set provides only raw user ratings (such as 1 to 5 "stars") and does not hold any semantic information besides the movie names.

We attach Mood4 to a CF-based recommender system that we developed [4]. The distance measure used in our implementation for estimating items similarity is Pearson's correlation coefficient [7]. CF-based systems typically require new users to provide a few ratings (at least 15 [6]) for bootstrapping, in order to produce decent recommendations. As this would take too long for a conference demo, we set the underlying recommender to use, for new users, a default rating value for each item (its average rating). While this may seem disadvantageous at first, it helps demonstrate the true power of Mood4 in capturing the attendees' current mood.

Interactive Demonstration. As mentioned, Mood4 is implemented as a smartphone/tablet web application. The first part of the demonstration will be held *on the attendees' own smartphones or tablets* (we will provide a few devices for the attendees that do not have such devices). We will give a brief description of Mood4 and ask each of the attendees to enter 2-4 example movies indicating their own current mood. Mood4 will generate in response, for each attendee, a list or recommendations best capturing these examples. In addition, Mood4 , set in demonstration mode, will also generate alternative recommendations lists computed by three other algorithms; (a) the mood-insensitive underlying recommender system, (b) a naive algorithm which considers only a single example, and (c) a simplified variant of Mood4 that does not employ the joint representatives. The attendees will then be asked to compare the recommendations generated by Mood4 to those of the other algorithms. We will verify that indeed the majority of the attendees likes and prefers the recommendations generated by Mood4 .

We then proceed to the next part of the demonstration by selecting a random participant and asking her to re-enter her movie examples on our main workstation. This time, Mood4 is set to "under the hood" mode which allows to present and analyze each step of the algorithm. We start by showing the geometric representation derived from the attendee's selection, and then compute the corresponding joint representatives. Then we show the resulting refined geometric representation and discuss the weighting mechanism used to evaluate the final weights. Finally we present the diversification mechanism used to generate the final set of recommendations. To conclude the presentation, we will tell the attendees that they can continue using Mood4 (with their own devices) throughout the conference week (for instance, to search for a movie to watch before bedtime). We further encourage them to present Mood4 to all their colleagues who have not yet seen our demonstration, serving as Mood4 's "honorary ambassadors".

5. REFERENCES

[1] Pandora internet radio. *http://www.pandora.com/*.

[2] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2009.

[3] J. Bennet and S. Lanning. The netflix prize. *KDD Cup*, 2007.

[4] R. Boim, H. Kaplan, T. Milo, and R. Rubinfeld. Improved recommendations via (more) collaboration. *WebDB*, 2010.

[5] R. Boim, T. Milo, and S. Novgorodov. Diversification and refinement in collaborative filtering recommender. *CIKM*, 2011.

[6] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. *WSDM*, 2011.

[7] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 1988.

[8] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.