Profiled Search Methods for e-Learning Systems

Tomasz Orzechowski, Sebastian Ernst, Andrzej Dziech

AGH University of Science and Technology
{tomeko, ernst}@agh.edu.pl, dziech@kt.agh.edu.pl

Abstract. Most currently used e-Learning Systems do not often offer search functionality. Even if methods are provided to search for Learning Objects (LOs), they don't usually utilize information about users' interests stored in their profiles.
Moreover, most of the search engines are only use query conformity to order the result list. User profiling methods are usually absent, as behaviour analysis methods are difficult to implement in specialised e-Learning systems. In this paper, a new approach for profiled search, which enables better adjustment of the order of results for end-users' expectations is proposed. It is related to the situation when both LOs and users profile descriptions are standardized.

Keywords: e-learning, profiled search, collaborative filtering, recommender systems

Introduction

Various e-Learning systems, including open source based, are very popular nowadays. These systems are used by numerous commercial content vendors as well as by government organisations. All these systems share a common purpose: they provide access to on-line resources for education.

These resources, are used by teachers/lecturers to create courses in a given subject or topic. Wide availability of resources lets course designers choose the elements which are most suitable for a given application.

As browsing such repositories is inconvenient, inefficient and sometimes impossible, various search engines have been developed to help users retrieve specific data. However, often it is not possible to formulate the search queries alone so that they can pick out the elements which are of value to a user.

The problem of ordering search results by predicted level of interest for the user pertains both to general-purpose Internet search engines and to browsing assistants in specialised repositories. It gets even more severe if elements are very similar as far as their keywords or descriptions are concerned.

Acknowledgment

The work presented in this paper is partially supported by the European Commission under the Information Society Technologies (IST) program of the 6th FP for RTD as part of the CALIBRATE project, contract IST-28025.

The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Commission, and the European Commission is not responsible for any use that might be made of data appearing therein.

Motivation for the approach

Most search methods used in Internet repositories match words in the search query and keywords in the known available elements to produce a list of results. As query conformity often isn't enough to order the search results properly, various schemes are used to prioritise the results of most predicted value for the user. However, people evaluate things on a very subjective basis. Thus, marks on a ranking scale provided by different users may have completely separate meanings. It is impossible to provide a uniform ranking scale that's universal – applicable for all users of a system. The user will always face the question: "Do the ranks assigned to that element reflect my preferences?"

One approach to grouping similar items is to derive proprietary classification schemes. However, experts in particular topics – music, movies or a given scientific field – would need to be employed to assign items to those groups and to provide evaluations within them. Even in cases this approach seems feasible – usually in closed, very narrow-subject systems – it still does not reflect the taste of individual users. Moreover, users are often unable to narrow their search to a specific field within a classification scheme – e.g., most people can't define their favourite movie or music genres.

The profiled search advantage

As we already mentioned, proprietary classification schemes do not reflect users' preferences, and universal ranking schemes try to blend all ranks into one general value.

However, as everyone's preferences are different, most people have their peers taste-wise – their preferences are usually similar to preferences of other users within the system. Therefore, if the system was able to group users according to their preferences, it could sort the result list using ranks assigned by people who have – with a large degree of probability – similar taste.

We shall refer to such approach as profiled search. Profiled search – just like classic search methods – uses various data to assess the value of particular elements for a user, but unlike the classic methods, it tries to take the user's preferences into account.

Collaborative filtering

A method commonly used to assess user preferences with regard to a set of items is called collaborative filtering.

According to paper [1]: "Automated collaborative filtering (ACF) systems predict a person's affinity for items or information by connecting that person's recorded interests with the recorded interests of a community of people and sharing ratings between likeminded persons."

The quoted definition is general enough to encompass all applications of the aforementioned technique.

As already mentioned, the basic concept for collaborative filtering is similarity between users' tastes. Data in a collaborative filtering system is usually represented as a m by n user/item matrix, referred to as R, where m is the number of users and n is the number of items rated by at least one user. Depending on the way a user's interest in an item is represented, elements of $R - r_{ij}$ are either logical (true/false), discrete (integer values) or continuous (floating-point numbers). [3]

One of the largest problems in collaborative filtering systems is data sparsity. This may make the filtering results irrelevant, especially in early stages of operation. Even in large systems which have been in operation for a long period of time, ratings from a single user usually encompass less than 1% of all items. Data sparsity also induces the lack of transitivity – if preferences of user A correlate with those of user B, and if preferences of user B correlate with those of user C, the system still cannot assume that users A and C share a common taste.

Determination of users' preferences

Methods for determining users' preferences can be divided into two groups: explicit (active) and implicit (passive).

Active methods involve action from the user – usually explicit assignment of a rank to an item. This approach has many advantages – primarily, it is capable of producing ranks assigned to items by persons with knowledge of the subject, which improves the reliability of collected opinions. The main disadvantage is that active methods require user input, which can exacerbate the problem of data sparsity. Also, there is no way of avoiding biased opinions.

Passive methods, on the other hand, require no user input and analyse the users' actions to "guess" which elements are useful to a user. In applications such as online stores, the obvious action to record is purchase of an item. In such case, the amount of false information is quite low, as people tend to be rather careful when their money is involved. In other systems, data collected using implicit methods can be less dependable. Many profiled search engines collected the events of users clicking on particular results within the result list. Some go as far as measure how much time a user spends viewing a particular result. This may be misleading, as the user may click a possibly interesting item, examine it and either decide it is useless or decide it is of good value. The system usually cannot distinguish between those two situations.

Implementation of CF-based systems

Although it is possible to create a system using collaborative filtering which operates on original data – the R matrix, it is usually unfeasible due to the size of input data and the required computational complexity.

Usually, the input data for a CF-based system is viewed as n-dimensional, where n is the number of items for which implicit or explicit preference data has been recorded.

Therefore, various methods are used to reduce the original data, while maintaining its characteristics. A well-tested approach is to use feature extraction methods – a subset of dimensionality reduction algorithms aimed at maintaining as much characteristics of the original data as possible. The resulting data puts the original

users in an *n'*-dimensional space. One of the most commonly used algorithm is PCA (Principal Component Analysis), described in [2] and [4].

Computation of per-user preferences is inefficient and can impact the scalability of the entire system. Therefore, users are grouped into clusters according to their perceived preferences. Various clustering algorithms can be used, but the tendency is to use partitional clustering rather than hierarchical clustering, as reduced CF data usually has an uniform distribution within the *n'*-dimensional space. A general overview of CF system architecture is shown in Fig. 1.

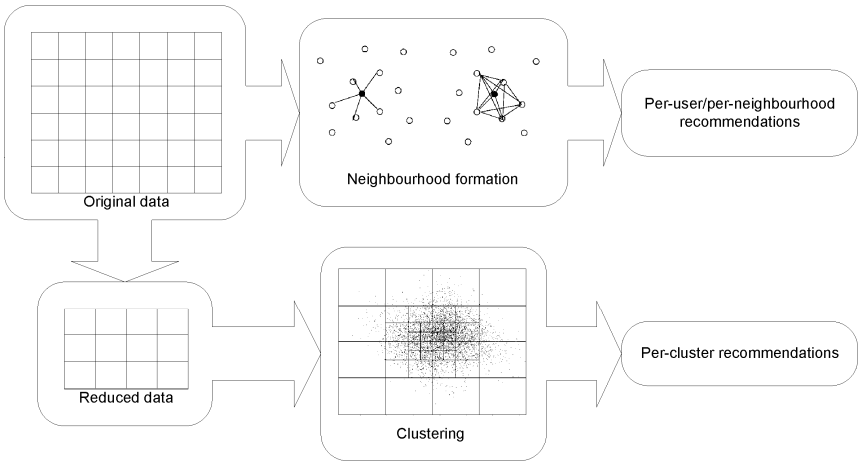

Fig. 1. Overview of CF system architecture [2, 3, 4]

Ranking vs. recommendation systems

User preference data computed as described in the preceding sections can utilized by online systems in two ways. One – as mentioned before – is to order the results within a result list to prioritise those of higher predicted value for a given user.

The other way the data can be utilised is to provide recommendations of new items for users. This can be done based on a set of items, or the user's preferences. While recommendations based on user's preferences can be provided by the ranking system using already computed user preference similarity data, recommendations of new items to match a pre-selected set of items require modifications in the system architecture.

Offline computation

As shown above, collaborative filtering systems may require many calculations to be performed. Online applications require a short response time, Therefore, the CF system is divided into two modules – one for online operation, and one for off-line computation.

The offline module is launched periodically to perform all necessary calculations (user/cluster assignment, evaluation computation) – usually once a day. Calculations

are performed using a current snapshot of input data, and all results are spooled until all procedures are finished. The previously computed results are left intact at this stage. Upon completion, the old results are overwritten with the newly calculated data.

This allows the online module, used to retrieve pre-computed ranking and recommendation data from the database, to be very quick, with computational complexity kept at minimum. That is because all that needs to be done online is to look up the user in the users/clusters assignment database and retrieve ranks and recommendations for that given cluster. [2]

Profile Analysis in E-Learning Systems

The remarks in the previous chapter pertain to collaborative filtering-based systems in general. As mentioned above, e-Learning systems possess some characteristics which require adaptation of these general methods in order to improve efficiency in those specialised systems.

Types of collected event data

The first step for creation of a CF-based system is selection of data used to predict users' preferences. Analysis of e-Learning object repositories and search systems resulted in establishment of a set of explicit and implicit data collection methods.

The explicit methods include:

- ranking: when the user applies a rank within a given scale to an object,
- labelling: when the user attaches a label to an object.

The implicit methods are:

- selection: when the user clicks a link on the result list; a page with an extended item description is displayed,
- downloading: when the user clicks a link on an item description page; this either displays the Learning Object itself in the browser or downloads the object to the user's computer.

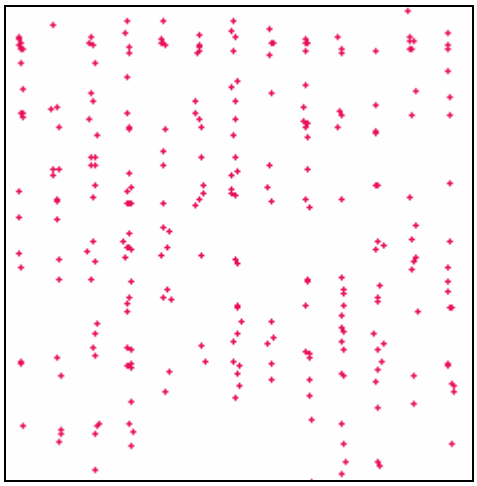

Fig. 2. Distribution of example user profiles in a 2-dimensional space.

Behaviour vs. profile-based user clustering

Most collaborative filtering systems rely on user behaviour and user actions to determine user preference similarity. The underlying idea is that contents of the objects and characteristics of user profiles are not taken into account.

However, characteristics of e-Learning applications require a slightly different approach, for a few reasons. They are connected to the characteristics of e-Learning systems themselves as well as the target audience. Firstly, the most dependable user behaviour data collected by the system – the item ranks – are extremely sparse. Secondly, the implicitly collected data has low dependability, as the user may decide at any point that the item is not what they are looking for after all. Because some users of the system are young children (users of type *learner*), ranks may not be reliable, and their action records are prone to be chaotic.

Therefore, we have decided to take a step aside from the classic collaborative filtering guidelines and develop a new approach to user similarity computation.

The approach features a new way of grouping users – based on their profiles rather than their behaviour. The following profile elements are taken into account: *age*, *gender*, *country of education* and *points of interest*.

The procedure is similar to the procedure used in "classic" collaborative filtering. A matrix of size *m* times *n* is created. However, columns of the matrix are composed of all possible values for the four profile fields, instead of object ranks/events. In the current implementation, enumerated fields get a '1' if a given value applies, and a '0' if it does nit. For other fields, the value is numeric.

To provide balance between the four sections, weights have to be defined for each one of them. This allows the administrator to fine-tune the distribution of users after feature extraction in the *n'*-dimensional space. Figure 2 shows a 2-D distribution of example user profiles.

Types of user clusters

There is a strict division of users in e-Learning systems into two separate groups: teachers and students (called learners). Therefore, the system should maintain separate clustering schemes for these types of users.

Moreover, as presented earlier, the suggested architecture of a ranking system for e-Learning consists of two separate approaches towards user similarity assessment: behaviour-based (classic) and profile-based.

Therefore, six different types of user classification are present in the current implementation:

1. all users / behaviour-based
2. learners / behaviour-based
3. teachers / behaviour-based
4. all users / profile-based
5. learners / profile-based
6. teachers / profile-based

Calculations are performed independently and separately for each of the cluster types presented above.

Positioning of results using search criteria and profile conformance data

In the preceding sections, we've presented usage examples and advantages offered by our profiling system, designed specifically for profiled search within e-Learning resource repositories.

While developing the methods for discovering users' preferences, we have been investigating another problem as well. It concerns methods of presenting search results using three different indicators computed by our system (see below) within a single set of results. Output from each of the classification schemes is a set of percentage values, representing the three aspects conformity of given LO (search result):

1. *Search query conformity* specifies how the given LO (described in compliance with the IEEE LOM standard) matches a given search method. Our system delivers separate sets of results for each of the used search methods.
2. *User profile conformity* specifies how the given LO matches the user's profile, provided upon registration. The system applies values to LOs found previously by the search methods.
3. *User cluster conformity* delivers separate sets of results representing the predicted attractiveness of given LOs within clusters to which the user belongs. Detailed information regarding cluster types and clustering methods have been described in previous sections.

The simplified architecture of the entire result generation system is shown on Fig. 3.

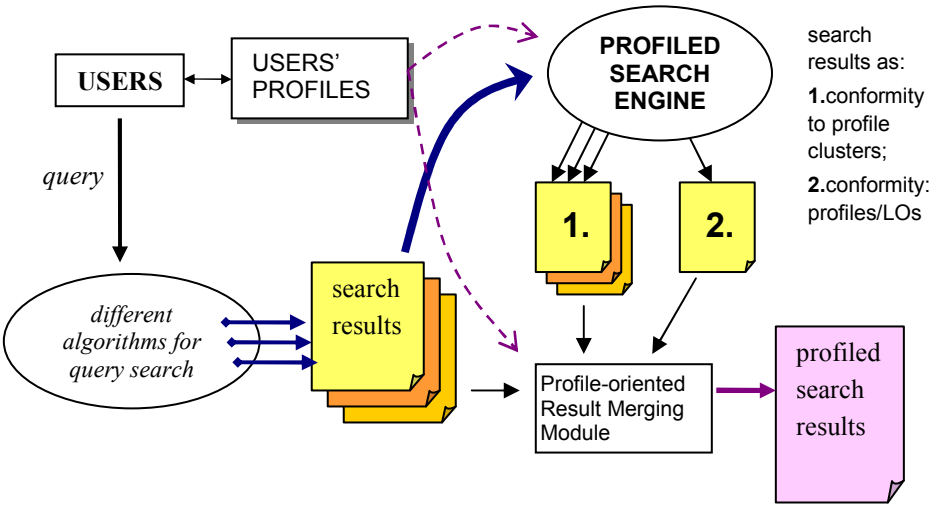

Fig. 3. Simplified schema of merging search results process.

Merging results

A Web GUI was implemented to allow users of the system to choose their preferred way of merging results. Furthermore, an API based on the Web Services technology was implemented to enable utilisation of results generated in our system by external systems.

As each of the criteria is separate from others, we let users to choose which features are used to create the end list of results. These features were divided into two separate parts: *search methods relevance* (query conformity) and *profile relevance* (user profile/cluster conformity).

Even if the user decides to utilize all of the offered features, the system prepares two separate lists of results for each part.

Search method/query conformity

We suggest two different solutions:

- *weighted average* – end-user can choose the influence of each search method on the end results' list;
- *maximum* – the maximum value of LO conformity is taken;

Popularity and conformity to profiles and clusters

We divided this part into the following elements:

- conformity to user profiles,
- conformity to clusters determined by user behaviour similarity,
- conformity to clusters determined by multi-dimensional user profiled analysis.
- general popularity (not taking individual users' preferences into account).

For all four elements presented above, weighted average is used to create a single list of results. Additionally, as ranks for each element can be based on data collected using explicit or implicit methods, the user can choose how these methods influence the computed value for the relevant elements., also using weighted average

Merging profiles and search methods

The methods presented above generate two lists of results – one sorted by search method conformity, and one sorted by profile conformity. Determination of methods for merging these two lists is a significant problem. While profiling methods are very important in search engines, they should not diminish the importance of conformity returned by the search methods.

While research is being performed, the problem was temporarily solved by using geometric mean. Ultimate conclusions in this aspect will require extensive real-life testing and fine-tuning of the production system.

Future work

Each innovation described in this article brings the possibility of further development, improvement and refinement. This section describes several ways of improving the current implementation.

The current system is designed to keep and include historic data in the calculations. This, of course, could become very time-consuming, and the daily recalculation routine could take up most of the available CPU time.

Therefore, we are experimenting with different methods of limiting the amount of processing that needs to be done. One of the most obvious of these is incremental processing – in this case, only "new" events are processed every time, and the previously computed data, instead of being overwritten, is merged with the new data to include the changes.

As pointed out in several of the preceding sections, ranking and recommendation systems require a lot of "fine-tuning" – usually by modifying various weights used to multiply data in various stages of computation.

In order to view the results of parameter modifications, we are working on a user-friendly GUI for visualising the distribution of elements in appropriate sections of the system. This will improve accuracy and will make the system more usable in real-life applications, especially when the amount of data increases.

Currently, individual values of enumerative elements in user profiles used for profile-based user similarity assessment are treated separately.

Therefore, if a user likes physics, the value in the appropriate cell of the matrix used for collaborative filtering will be set to '1', without affecting other cells. However, expert knowledge can be used to define similarity levels between individual values – for instance, if the user likes physics, but says nothing about maths or chemistry, we may assume that the user likes maths with the level of 0.7 and chemistry with the level of 0.5. This should allow the user profiles to blend more, especially considering the low age of a significant part of users in the system.

Although the primary goal is to implement an integral element of the e-Learning search system, capable of providing user-specific ranks and recommendations, we are also investigating the capabilities of independent software packages capable of performing those tasks.

The advantages are two-fold. On one hand, external software could be used as a back-up for the existing system. On the other, even with the existing system functional and on-line, other software packages can be used to test new concepts and verify the current implementation.

One of the software solutions taken into account is Microsoft SQL Server 2005, which includes numerous data mining techniques, including functionality to implement a CF system for e-Learning. Functions particularly useful for the discussed application include: [5]

- clustering: a set of iterative routines to group processed data,
- association: used to analyse market baskets; can be useful to implement of user similarity criteria – users who added similar Learning Objects to their "bookmarks"
- sequence clustering: used to analyse sequences of events with attributes; can be used to view a search engine user behaviour as a sequence of actions, instead of just analysing individual events.

Conclusion

The presented profiled search system offers new, important features not available in currently-used e-Learning Systems. Simultaneous theoretical research and practical development work open the unique opportunity of testing how new ideas perform in real-life and lets us acquire valuable experimental data.

Moreover, analysis of the history of the users' activities, made possible by our system, in connection to analysis of LOM fields, is a significant step towards creation of an intelligent Learning Object search engine, which will present search results in a way as close to the user's expectations as possible.

References

1. Herlocker, J., Konstan, J., and Riedl, J., *Explaining Collaborative Filtering Recommendations*. Proc. of the ACM 2000 Conference on Computer Supported Cooperative Work , December 2-6, 2000.
2. Ernst, S., Pacewicz, D., and Klimek, R., *Recommendation systems: prediction of web site user preferences*. Proc. of Computer Methods and Systems 2005, November 14-16, 2005.
3. Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., *Analysis of recommendation algorithms for ecommerce,* Proc. of the ACM Conference on Electronic Commerce, 2000.
4. Goldberg, K., Roeder, T., Gupta, D. and Perkins, C., *Eigentaste: A constant time collaborative filtering algorithm*, Information Retrieval, 2001.
5. Skurniak, T., *Business Intelligence w SQL Server 2005*, Microsoft Technology Summit 2006.