Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 1 (2010) 2877–2882
Procedia Computer Science
www.elsevier.com/locate/procedia

1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)

Meta-Mender: A meta-rule based recommendation system for educational applications

Vicente Arturo Romero Zaldivar[a,*], Daniel Burgos[a,b]

[a]Atos Origin SAE, Albarracin 25, Madrid 28027, Spain
[b]International University of La Rioja, Gran Via Rey Juan Carlos I 41 26002, Logroño, La Rioja, Spain

Abstract

Recommenders are central in current applications to help the user find useful information spread in large amounts of data. Most Recommenders are more effective when huge amounts of user data are available in order to calculate user similarities. In general, educational applications are not popular enough in order to generate large amount of data. In this context, rule-based Recommenders are a better solution. Meta-rules can generalize a rule-set, providing bases for adaptation. The authors present a meta-rule based Recommender as an effective solution to provide a personalized recommendation to the learner, which is a new approach in rule-based Recommender Systems.

Keywords: rule-based recommendation systems; personalization; adaptation; meta-rule; rule generation

1. Introduction

Nowadays, when the amount of information is becoming over-exceeding, Recommendation Systems emerge as the solution to find the small piece of gold in mountains of garbage. In electronic commerce, knowledge management systems, social networks, and other fields and markets, they help users to find useful products, lessons or contributions. There are many inputs which can be used as information sources like i.e. similarities between users, user profile, and preferences. All these inputs provide the system with valuable data to suggest the user the best way to follow or the most appropriate choice.

People ratings [1] are another important source of information for Recommendation Systems. With this information, the system tries to find pages matching high rated pages and to do not match the poorly rated ones. With enough information these systems are able to categorize and retrieve new pages with great accuracy. They are called ***content-based Recommendation Systems***.

Another option is to recommend based on the ratings of other people who has been interested in a given resource or product before. Systems which use this information are called ***collaborative Recommendation Systems***. These

* Corresponding author.
E-mail addresses: vicente.romero@atosresearch.eu, daniel.burgos@atosresearch.eu.

doi:10.1016/j.procs.2010.08.014

systems usually ask people to explicitly rate pages or products and later use this information to recommend to similar users.

Other sources of information are i.e. user interests, goals, and objectives, all of them more useful for educational applications. However, educational applications lack of enough amounts of data to establish user similarities in a precise way. In this case, recommendations are based on information stored in a user model which is extended explicitly or implicitly.

For educational applications ***rule-based Recommendation Systems*** have proved as more useful than other systems [2]. In general acceptable recommendations can be obtained with a small amount of information. However, when the system achieves a better knowledge of the user, recommendations increase precision since rules evolve in parallel or new ones are included to the rule-set. The present work builds on the success of rule-based Recommendations Systems for applications with little usage data, in particular educational applications. Furthermore, we also introduce a new abstraction level, meta-rules, which provide foundations for effective adaptive and personalized processes, such as in, e.g. learning.

2. Background and Related Work

Collaborative Recommendation Systems make use of user ratings to build clusters of similar users. If a given user belongs to a cluster then items selected by other people in the cluster are strong candidates to be recommended. There are several good examples of collaborative Recommendation Systems like i.e. PHOAKS [3] and Group Lens [4], which are quite remarkable.

On the other hand, content-based Recommendation Systems suggest items which are similar in content to others that the user has bought or has been interested in previously. As an example, Fab [5] is mentioned, which recommends web pages. Fab is able to create a training set with few ratings from each user. Another example is ELFI [6] which recommends funding information from a database. In both cases users are required to operate the system for a certain period of time in order to allow the system to retrieve enough information and provide useful, accurate recommendations. Also of relevance, there are systems such as CiteSeer [7], which uses content-based similarity matching to help searching for interesting research papers within a digital library.

Regarding the field of educational recommendations there are approaches like [8] which use a modified page ranking algorithm for the generation of recommendations. This work is valuable for us because we propose the generation of rules based on user's activity and the input is as in the referenced paper comes from actors, activities and resources. The main difference is that in our approach, a set of meta-rules is obtained instead of a directed graph.

In the field of rule-based recommenders there are few related reports in the literature [2], for example, describes a rule based Recommendation System for online discussion forums. Actually the system is able to call several encapsulated recommenders, collaborative filtering or content-based recommenders, and the rules decide according to the amount and type of user data which recommender should be called. In doing so, the rules define a meta-recommender which is very interesting but tangential to this work..

An advantage of rule-based recommenders against other approaches is how easy it is to generate explanations for these systems. In many cases, it is almost impossible to explain to the user how a Recommendation System has derived a conclusion. This problem is usual in automated collaborative filtering systems, see [9], where the lack of explanations decreases the system acceptance and affects user trust.

Regarding meta-rules we have not found anything similar to our proposal. The closest concept is rule templates, followed by Open Rules [10] and the Object Oriented RuleML approach of handling rules as data [11], thus generating entire rules from its component parts. These approaches are very valuable but our approach of producing rules using imperative programming comprises both approaches and at the same time is more powerful. For example our meta-rules can generate meta-rules effortless. For other approaches this can be a tricky and painful task.

3. Meta-Mender: concepts and implementation

First the concept of meta-rule will be defined. In short, a meta-rule can be seen, using the mathematical logic language, as a duple M(C, A): where C belongs to the infinite set of conditions and A belongs to the infinite set of actions. A condition is expressed logically as conjunctions of logical predicates. The action is executed only if the condition evaluates to true. And finally to remark its character of meta-rule: the action can generate only a set of rules. An empty set or a set with only one element are also valid.

This definition is abstract and of little use in practice. A more useful definition, and the one used in Meta-Mender is the following. A meta-rule is a statement of the form: if condition then action, where the condition is a conjunction of logical predicates and the action is a call to a function which generates none, one or several rules. Rules can be assigned a priority in order to force an execution order.

It can also be added, but this is not part of the concept, that predicates are evaluated on facts extracted from the user model. The user model in general is stored in a database, in most cases ontological. In order for these facts to be consumed by the meta-rule they must be introduced as Java Beans. So in a way a set of meta-rules can be seen as an ASP page that after been interpreted generates a HTML page. The differential element in a meta-rule is its action, the power of generating a new rule thanks to code execution. This code must execute file output statements in order to generate an output file with the rules.

With regards to implementation, the Meta-Mender Recommendation System uses DROOLS [12] as the rule engine. This rules engine works as follows: first it is required to feed the engine with a set of meta-rules, these meta-rules are defined by the professor or by the technical team using the application requirements. Later some facts should be added to the engine. As the facts are inserted meta-rules conditions are checked for completeness and once the engine is started, meta-rules that fulfill its conditions are fired and the corresponding actions are executed.

```
package service

//java imports
import java.util.ArrayList;
import java.util.List;

//bean classes
import domain.LRNClassData;
import domain.LRNStudentData;
import domain.LRNClassPerStudent;

//global variables definitions
global java.lang.String filePath;

//rules
rule Header
//empty antecedent, executes always
  when
  then
    WriteHeader(filePath);
end

rule CourseSequenceRule
  salience -1 //priority
  when //antecedent
  //classId is an object field
    LRNClassData($classId : classId, $className : className)
    LRNStudentData($studentId : studentId, $studentName : studentName)
    LRNClassPerStudent(classId == $classId, studentId == $studentId)
  then
    WriteClassMembershipRule(filePath, $studentName, $className);
end

//function implementation
function void WriteHeader(String filePath) {...}

function void WriteClassMembershipRule(String filePath, String studentName,
  String className) {...}
```

Fig. 1. A meta-rules file

The output, a set of rules, is used for a second iteration from which the final recommendations are obtained. See Fig 1 for an example of a meta-rules file, this file can generate rules for recommending the next course to follow in a

.LRN educational application. We must notice that at the action of the meta-rules a function is called, this function receives some information that allows it to write the adapted rules. A function that produces a rule is called a rule generator function. In general a rule generator function contains logic to analyze input data in order to decide which rules to generate and the amount of them. Finally file output primitives are used to produce the rules in its final form. Not all functions appearing at the action part of a meta-rule are rule generator functions. There are always one or several functions which accomplish other tasks, for example rule file header generation. These functions are called supporting generator functions.

It is clear that meta-rules can generate rules according to current system conditions and properties; this approach saves time and effort as long as increases adaptation. See Fig 2 for an example of a rules file obtained from the meta-rules file of Fig 1. In this figure it can be noted that the recommendation is created as rules consequences are executed. The function `AddRecommendedCourse()` builds a recommendation list and the parameter `(10 – $courseLevel)` allows giving more priority to those courses with a lesser level of complexity. This criterion can be personalized to every student. It is clear that similar rules can be effective for recommending products, services and items of any kind not only courses or lessons.

Meta-rules can handle changing conditions. Different rules can be generated depending on user data, system properties, etc.; also rules priority can be different which implies different recommendations. Also multiple optimizations can be considered in order to reduce response time. For example rules obtained for a given user can be cached avoiding continued generation for current users. This can be done because user interests will not change, in general, during the same session. On the contrary, if the user modifies his profile during the session manually, new rules must be derived. In this sense, meta-rules can increase the system performance. If rules are adapted to user conditions, an optimized set of rules can be generated for every user. Furthermore, this strategy will decrease the response time.

```
package service

import domain.UserData;
import domain.TotalCoursePercentage;
import domain.Course;

global domain.RulesOutData outData;

rule MathI
  when
    UserData($userName : name, $userId : id)
    Course($courseLevel : courseLevel, $courseName : name, id == 1)
    not (TotalCoursePercentage(userId == $userId, courseId == 1))
  then
    outData.AddRecommendedCourse($courseName, 10 - $courseLevel,
     $userName);
end

rule AlgebraI
  when
    UserData($userName : name, $userId : id)
    Course($courseLevel : courseLevel, $courseName : name, id == 2)
    not (TotalCoursePercentage(userId == $userId, courseId == 2))
    exists (TotalCoursePercentage(userId == $userId, courseId == 1))
  then
    outData.AddRecommendedCourse($courseName, 10 - $courseLevel,
      $userName);
end
```

Fig. 2. A rules file obtained from the meta-rules file of Fig 1

4. Practical implementations of the Meta-Mender recommendation system

The Meta-Mender recommender is been used at present as part of two projects. The first one, TELMA (www.ines.org.es/telma), is focused on the application of Communications and Information Technologies in lifelong training of surgeons of Minimally Invasive Surgery (MIS).

TELMA develops an online learning environment which manages the content and knowledge generated by users in an efficient way. TELMA creates a new training strategy based on knowledge management, cooperative work and communications and information technologies aiming to the improvement of the formation process of the surgeons of MIS.

In order to cover all these needs and achieve its goals, TELMA develops a cooperative and adaptable learning platform. This platform consists of a training system which integrates a tool for the authoring of didactic multimedia content, a Recommendation System and a social network. The Recommendation System, Meta-Mender, manages the knowledge generated by the users creating the foundations for adaptive learning. The learning platform allows the construction of a complete knowledge base thanks to the reuse and sharing of the knowledge generated for the professionals who access the learning system.

The second project, GAME·TEL (www.ines.org.es/gametel), is focused on the creation of a system for the design, development, execution and evaluation of educational games and simulations, adapted to student preferences, educational goals, profile, etc [13].

The games and simulations are conversational adventures which are both usable and understandable. These characteristics benefits to the content creator, the professor in most cases, as long as to the final user, usually the student [14, 15].

The software system is composed of several interconnected modules which allow the integration and intercommunication between games and simulations and several tools widely used for communities of professors. Initially these tools are the learning management systems Moodle and .LRN, and the authoring and learning units execution system LAMS [16, 17]. In this context the Meta-Mender Recommendation System will suggest to users the best learning path according with their preferences, goals and objectives and will help with the game adaptation problem.

As an example, the following meta-rule allows the generation of rules for the forums that the user has access to, see Fig 3. This kind of rules allows for adaptation in case of the addition of a new forum to the application, a fact that can happen at any moment.

```
rule ForumRecommendationRule
salience -1
when
  TelmaUser($userId : userId, $userName : userName, $mainInterest:
    mainInterest)
  TelmaForum($forumId : forumId, $mainTopic : mainTopic)
  TelmaForumAccessPerStudent(forumId == $forumId, userId == $userId)
then
  WriteForumRecommendationRules(filePath, $userId, $userName, $forumId,
    $mainTopic);
end
```

Fig. 3. A meta-rule from TELMA application

5. Conclusions and future work

In this paper a meta-rule based Recommendation System, Meta-Mender, has been presented. This Recommendation System is oriented to educational applications where few data is available to make useful recommendations. User models implicit and explicit information are fundamental in order to provide clues that allow the user to achieve a goal, objective or interest.

Meta-rules are a novel approach in the field of rule-based recommendation because they impose a new abstraction level. This abstraction level is highly valuable for adaptation (e.g. personalized learning). In this context, rules can be different for different users or for the same user at different periods of time.

At present, the Meta-Mender Recommendation System is a component of two educational projects, TELMA and GAME·TEL, both under development. The evaluation phase must yet be done to proof its benefits for the user´s experience. Indeed, the concept which the Meta-Mender is based on (e.g. meta-rules for adaptation) starts a new branch in the recommendation field and it broadens the scope for new solutions in the field.

Acknowledgements

The research presented in this paper has been partially supported by the following projects of the Plan Avanza, a Spanish, nationally funded R&D programme: FLEXO (www.ines.org.es/flexo, TSI-020301-2009-9), GAMETEL (www.ines.org.es/gametel, TSI-020110-2009-170), TELMA (www.ines.org.es/telma, TSI-020110-2009-85).

References

1. Rocchio, J.J.: Relevance feedback in information retrieval, in the SMART Retrieval System -- Experiments in Automatic Document Processing, Englewood Cliffs, NJ. Prentice Hall, Inc. (1971), pp. 313-323.
2. Abel, F., Bittencourt, I.I., Henze, N., Krause, D., Vassileva, J.: A Rule-Based Recommender System for Online Discussion Forums. Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany (2008).
3. Terveen, L., Hill, W., Amento, B.,McDonald, D., Creter, J.: PHOAKS: a system for sharing recommendations, Communications of the ACM Volume 40, No. 3 (1997).
4. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news, Communications of the ACM Volume 40, No. 3 (1997).
5. Balabanovi, M., Shoham, Y.: Fab: content-based, collaborative recommendation, Communications of the ACM Volume 40, No. 3 (1997)
6. Schwab, I., Pohl, W., Koychev, I.: Learning to Recommend from Positive Evidence, Proceedings of Intelligent User Interfaces, ACM Press (2000), pp. 241-247.
7. Bollacker, K.D., Lawrence, S., Giles, C.L.: CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications, Proceedings of the Second International Conference on Autonomous Agents, Minneapolis MN, USA, (1998).
8. El Helou, S., Salzmann, C., Sire, S., Gillet, D: The 3A contextual ranking system: simultaneously recommending actors, assets, and group activities. In Proceedings of the Third ACM Conference on Recommender Systems (New York, New York, USA). RecSys '09. ACM, New York, NY (2009), pp. 373-376.
9. Herlocker, J.L.: Position Statement - Explanations in Recommender Systems. Proceedings of the CHI' 99 Workshop, Pittsburgh, USA (1999).
10. OpenRules. http://openrules.com/index.htm.
11. Object Oriented RuleML. http://ruleml.org/indoo/indoo.html.
12. DROOLS. The Business Logic Integration Platform, http://www.jboss.org/drools.
13. Burgos, D., Tattersall, C., Koper, R.: How to represent adaptation in eLearning with IMS Learning Design. Interactive Learning Environments, 15(2) (2007), pp. 161-170.
14. Moreno-Ger, P., Burgos, D., Martínez-Ortiz, I., Sierra, J., Fernández-Manjón, B.: Educational Game Design for Online Education. Computers in Human Behavior 24(6) (2008) pp. 2530–2540
15. Torrente, J., Moreno-Ger, P., Fernández-Manjón, B.: Learning Models for the Integration of Adaptive Educational Games in Virtual Learning Environments. Proceedings of the 3rd International Conference on E-learning and Games, Nanjing, China. Lecture Notes in Computer Science 5093 (2008), pp. 463–474.
16. Burgos, D., Tattersall, C., Koper, R.: Re-purposing existing generic games and simulations for e-learning. Special issue on Education and pedagogy with Learning objects and Learning designs. Computers in Human Behavior (2006).
17. Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J., Fernández-Manjón, B.: A Descriptive Markup Approach to Facilitate the Production of e Learning Contents. Proceedings of 6th International Conference on Advanced Learning Technologies (ICALT 2006), 19-21, Kerkrade, The Netherlands. (IEEE Computer Society) (2006).