# Design and Development of Curious Worth Putrefaction Basedne-Absorbing Manuscript Recommender Procedure

Dr. S. Ravichandran[1], Dr. N. Jeyakumar[2]
[1](Computer Science Department, Annai Fathima College of Arts & Science, Madurai, India)
[2](Commerce Department, MKU Constituent College, Sattur, India)

***Abstract***

*Research on E-learning process has developed day today for the web based educational system. To make the system even more beneficial and adaptive to the student needs, many interoperable and information retrieval services are in the existing system. Especially in E-learning process the data are aligned from different domain representation. The proposed technology Singular Value Decomposition (SVD) Recommender System, summarize the complex matrix to find the singular value of Eigen values and Eigen Vectors. The SVD has U, V and S matrices and they are known as the Orthogonal matrix (U & V) and Diagonal matrix(S) respectively to decompose into single values of Eigen values and Eigen vectors. The diagonal matrix consists of r, where r is the rank of the matrix and it has non-zero entries. This system is a factorization technique for producing low rank of a input matrix to find the singular value. So the SVD enhances the real life classroom teaching which can increase the learning effectiveness to answer the various drawbacks of web based education system.*

**Keywords –** *SVD, E-learning process, TEL, neural network, fuzzy system, and NFPR*

## I. INTRODUCTION

In general, recommender systems are used for recommending some items that might be of interest to the users. Recommendations are typically given based on information such as user profiles, item properties (content based recommenders), and users preferences (collaborative filtering) expressed explicitly (e.g., by user ratings and ‘likes’) and/or implicitly (e.g., by the frequency of visits/downloads) (Jameson, Konstan, & Riedl, 2002). By combining this information with a set of recommendation rules, a recommender system tries to predict which items will be of interest to the user, so that he/she can achieve some predetermined goals. Important questions to be addressed when designing recommender systems include (but are not limited to):

1. What are the most effective techniques for recommendation in specific domain?
2. What information about the users is needed; how to collect and represent it?
3. What information about the items is needed; how to collect and represent it?
4. How to evolve and adapt recommendations in order to make them continuously effective (i.e., to sustain their effectiveness despite the changes e.g., in users preferences or any other requirements?

In addition to these challenges, information gathered about the users is often incomplete or unreliable, and that makes the generation of useful recommendations even more difficult. In the context of Technology Enhanced Learning (TEL), recommender systems are used for suggesting learning activities, materials and/or topics to students in order to assist them in achieving their desired learning goals – in general, to increase their level of knowledge on some subject (Tang & McCalla, 2003). In this case, the recommendation problem can be defined as the student’s request to the system: ‘‘given a representation of my current knowledge and preferences, recommend me the next topic/content/activity in order to help me learn the given subject’’ (Basu, Hirsh, Cohen, & evillManning, 2001). To address this request, the system generates recommendations based on the student model (i.e., its internal representation of the students’ knowledge and preferences), and the teaching model (i.e. the chosen pedagogical strategy usually defined by the teacher).Student model typically contains information about the student’s knowledge, preferences, learning style and the accomplished learning activities. This information is often extracted from history of interaction between the student and the learning environment. Teaching model defines a pedagogical strategy typically as a set of rules that determines the optimal way for learning some topic for certain type of students.

The challenging issues in educational recommender systems are an equivalent as those recognized in other recommender systems:

1. The way to collect and represent relevant information about students, and the way to structure the student’s model?
2. The way to use the student-related data (stored within the student model) to get useful recommendations, i.e., the way to define and evolve pedagogical recommendation rules?

3. From the sensible point of view, the way to implement the recommendation rules within the most effective way?

The first two questions are often addressed by leveraging the research work and therefore the results achieved by other researchers in the field. The last question is particularly challenging considering that at the instant there's a scarcity of open implementations of general pedagogical recommenders that would be reused in different domains and TEL systems. Getting to address this technical challenge, we've developed an open and adaptive software component, named Neuro Fuzzy Pedagogical Recommender (NFPR), for creating pedagogical recommenders in learning environments. NFPR is that the central topic of this paper provides a wizard-style interface and an easy-touse API, which makes it suitable for straightforward integration with various learning environments.

The paper is organized as follows: related work which includes recommender and Singular value Decomposition for TEL is given in Section 2; Section 3 outlines current challenges in the field of pedagogical recommender systems and clearly states the matter his work aims to address; the general architecture of the proposed software and therefore the algorithms it is based upon are given in Section 4; and a sample application (Section 5); usability and pedagogical evaluation is given in Section 6, whereas Section 7 outlines conclusions of this research.

## II. RELATED WORK

Related work for this research includes recommender systems generally, recommender systems in TEL and neurofuzzy systems in TEL. Accordingly, this section gives a quick overview of those three research areas. Currently, a widely used state of the art approach in recommender systems is Matrix Factorization (Bell, Koren, & Volinsky, 2009) which belongs to collaborative filtering family of recommender systems (Bobadilla, Hernando, Ortega, & Bernal, 2011). The most general data representation technique applied during this sort of recommender systems may be a matrix of n users and m items, where each matrix cell corresponds to the rating given to item I by the user u (Melville & Sindhwani, 2010, chap. 00338). The Matrix Factorization algorithm is employed to predict which item will have the highest rating for a few user, supported the ratings of other items by that user and ratings of other users. the most issue with this approach is that the so called 'cold start' problem, which suggests that within the beginning there's not enough data (ratings) to form good recommendations, and it's impossible to offer recommendations for brand spanking new users (before they supply some ratings) and new items (before they get some ratings). In practice these issues are resolved using simple average ratings, by creating hybrid recommenders together with content filtering techniques (Hummel et al.,2007), or using some more sophisticated methods (Gantner, Drumond, Freudenthaler, Rendle, & Schmidt-Thieme, 2010; Preisach, Marinho, & Schmidt-Thieme, 2010). a good range of other techniques, including statistics and machine learning based techniques, are also utilized in order to research data and provides recommendations (Melville & Sindhwani, 2010, chap. 00338). within the area of recommender systems for Technology Enhanced Learning (Manouselis, Drachsler, Vuorikari, Hummel, & Koper, 2010), research is concentrated on the development of recommender systems for recommendation of learning resources (materials or peers to provide help) or learning activities to the learners (Ghauth & Abdullah, 2010; Manouselis et al., 2010). Recommender systems for educational purposes are challenging research direction (Drachsler, Hummel, & Koper, 2009) since preferred learning activities of scholars might pedagogically not be the foremost adequate (Tang & McCalla, 2004) and proposals in elearning should be guided by educational objectives, and not only by the user's preferences (Santos & Boticario, 2010). Also, there are varieties of specific features that need to be taken under consideration, such as (Drachsler et al., 2009):

– The importance of context (which is not taken into account in common recommender systems);

– The inherent novelty of most learning activities;

– The need for a learning strategy;

– the need to take changes and learning processes into account.

There are many various approaches for recommenders in TEL, from collaborative and content filtering to hybrid approaches and every of them has some advantages and disadvantages counting on the context during which they need been used and the way they're evaluated (Manouselis et al., 2010). For example, the above mentioned Matrix Factorization technique which has already proved to be very successful in e-commerce and movie recommendation domains (Melville & Sindhwani, 2010, chap. 00338), is promising for educational domain, as well (Thai-Nghe et al., 2011), but it lacks one important feature – the ability to adapt to teacher's pedagogical strategy.

SVD has a crucial property that creates it interesting for recommender systems. SVD provides the simplest low-rank linear approximation of the first matrix and therefore the low-rank approximation of the first matrix is best than the first matrix itself.

Filtering out of the tiny singular values are often introduced as removing ―noise‖ data within the matrix. SVD-based approaches produce results better than traditional collaborative filtering algorithms most of the time. Be that as it may, SVD requires computationally over the top expensive network estimations and this makes SVD-based recommender frameworks less appropriate for large-scale frameworks. For this reason, most of the researches on SVD based recommendation specialize in scalability problem while protecting the top quality recommendations of the tactic.

In this thesis, SVD-based recommendation techniques are compared with experiments and a few new approaches are introduced to the present technique. the primary contribution we've proposed is that the categorization of things and users. Our experiments showed that, item and user categorization increases both the advice quality and speed performance of the SVD technique. Moreover, we adopted the tags to the normal 2-Dimensional SVD approach. By this manner, we've the prospect to analyze the effect of dimension (tags) to the SVD recommendation performance. Our experiments illustrated that, tags also increase the performance to some extent.

## III. PROBLEM STATEMENT

The work presented during this paper addresses three challenges related to pedagogical recommender systems:

A pedagogical recommender should support any number of criteria for recommendation.

A pedagogical recommender system should be adaptable, in order to support different pedagogical approaches and different domains. This might be the key for a wider adoption of such recommenders. However, to the simplest of our knowledge, most (if not all) implementations of pedagogical recommenders at the moment are very domain- and problem-specific, and that they cannot be reused in several environments. This also means the development of each pedagogical recommender starts almost from scratch

Last but not the smallest amount, a pedagogical recommender system should be intuitive and easy-to-use for end users (pedagogical experts). Once a pedagogical strategy is implemented with the recommender, it should be easy for a user to increase, modify it to suit the changes in his/her teaching practice.

## IV. PROJECTED ELUCIDATION

An important feature of NFPR is its flexibility: it can be used on custom input and output data sets (which correspond to, e.g., the student model and the recommended learning content, respectively), and allows for the creation of personalized recommendation rules. Fuzzy set theory is used to transform high-level pedagogical rules into a computational model, whereas a neural network is used to provide adaptively to the teacher's preferences. Thanks to the wizard-style user interface, using the system does not require in-depth knowledge of fuzzy sets and neural networks. NFPR is available as an open source implementation that can be easily integrated with almost any TEL system.

It illustrates how NFPR can be used with the student model (comprising the student's knowledge and learning style) as the input, and the recommended learning content as the output. In what follows we present the main building blocks of NFPR and their role in this recommender system.

### A. Domain Paradigm

NFPR's domain model contains structured knowledge of the domain in the form of a topic map (Amruth, 2006). Topics are related to one another with the prerequisite relation who means that one topic is a prerequisite for learning another topic. This type of domain model is chosen since it is intuitive to the end users (i.e., teachers). Other, more complex techniques for domain modeling (such as ontologies), do offer advantages, but have one significant disadvantage: they are difficult to accept by the end users and thus pose problems to wider adoption in educational practices (Hatala, Gasevic, Siadaty, Jovanovic, & Torniai, 2009). This model is the base for creating pedagogical recommendation rules.

### B. Student Paradigm

Student model stores information about student's current state of knowledge and personal characteristics (Stathacopoulou, Magoulas, & Grigoriadou, 1999). The student model used in NFPR is the overlay student model which represents student's knowledge as a subset of expert/system's knowledge of domain (Kass, 1989). The most important information it contains is a list of topics to learn and corresponding test results for those topics which represent the current state of student's knowledge.

### C. Responses

The responses for NFPR are extracted from the student model. They can be, for example, the student's knowledge of some topics and the preferred learning style. The student's knowledge can be evaluated with tests, while the learning style can be elicited through an appropriate questionnaire (Kinshuk et al., 2001).

### D. Productions

The production of NFPR is the recommended learning content that corresponds to some domain concept (i.e., a concept from the domain model). Possible outputs are identified by relating the available learning content to the appropriate concepts from the domain model.

### E. Sanction procedures

. Sanction procedures define the mapping of the inputs to the outputs of a recommender system and are based on the following set of high-level pedagogical assumptions:

IF (Student has good knowledge of Topic1) THEN Student should learn Topic2 Topic1 and Topic2 are topics (concepts) defined in the Domain model and they are related through the prerequisite relationship. Student's knowledge of these topics is stored in the Student model.

If Topic2 is related to some learning content, e.g., LearningContent2, then the above rule gets the following form:

IF (Student has good knowledge of Topic1) THEN Student should study LearningContent2 The conditional part of the rule may be more complex and include several conditions, like:

IF (Student has good knowledge of Topic1) AND (Student has excellent knowledge of Topic2) THEN Student should learn Topic3 Or, if Topic3 is related to some learning content (e.g., LearningContent3):

IF (Student has good knowledge of Topic1) AND (Student has excellent knowledge of Topic2) THEN Student should study LearningContent3 If the student's learning style is also considered, then the rule takes the following form:

IF (Student has good knowledge of Topic1) AND (Student has excellent knowledge of Topic2) AND (Student learning style is SomeLearningStyle1) THEN Student should learn LearningContent4.

These are high-level rules, typically used and understood by teachers. In NFPR such rules are converted to a computational model using the fuzzy set theory. Students' knowledge and learning style are considered to be linguistic variables, which can take values of the corresponding fuzzy sets.It is also assumed that each learning topic has the corresponding learning content.

In the current implementation, three fuzzy sets are used to express the student's knowledge of some domain topic:

POOR – insufficient knowledge of the topic

GOOD – basic understanding of the topic

EXCELLENT – advanced understanding of the topic

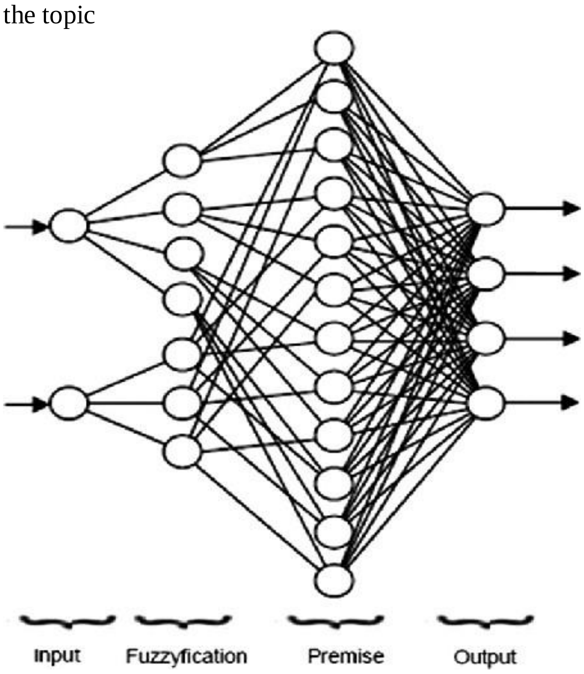


Fig1. Architecture of NEPR Neural Network

### F. Benefits of projected elucidation

The proposed model of pedagogical recommender system is very flexible as it allows for various customizations in order to support individual pedagogical strategy. It supports intuitive, non-formal pedagogical models that can be created by teachers based on their teaching experience, and can also be adapted to the teacher's preferences. Verbal pedagogical model can be easily translated to the corresponding fuzzy model by using fuzzy sets and rules. Furthermore, a neural network can be automatically generated and trained thanks to its straightforward architecture and learning rule. This means that different pedagogical recommenders can be automatically generated without a need to change low level implementation details. Accordingly, it is possible to create very sophisticated tools with intuitive and easy-to-use user interface that can produce ready to use neuro-fuzzy pedagogical recommenders. Possible customizations include:

1. Any number of inputs (theoretically), which means that it can support any number of criteria for recommendation (e.g., customized umber of learning topics, learning styles, and even other criteria constituting learning context);

2. Each recommendation criterion can have a customized set of corresponding fuzzy sets, so the translation from the verbal pedagogical model to fuzzy computational model can also be customized (for example, instead of POOR, GOOD and EXCELLENT, some may want to have five levels of grading with different naming);

3. Any number of outputs, which means customized number of learning topics for recommendation;

4. Adaptation of recommendation rules to some specific preferences; this is possible due to the fact that high level pedagogical rules transformed to fuzzy domain are automatically learned by the neural network. An additional benefit lies in the fact that regardless of all the above mentioned customizations, the internal operations of the proposed pedagogical recommender remain the same. This further means that the same implementation can be applied to a wide range of learning domains. Possible constraints could be faced when working with large number of inputs and fuzzy sets, which can cause rule layer to grow fast, so it may require more memory than usual (than standard configurations provide). However, having in mind the amounts of memory that modern systems provide, this can be easily resolved through appropriate system configuration (e. g., by assigning more heap memory to Java Virtual Machine).

## V. APPLICATIONS

The NFPR's wizard-style user interface for creating recommendation rules and neural network that implements those rules neural network is created with Neuroph, an open source Java framework for neural network development. 2 Neuroph provides simple Java API for using neural networks within Java applications, and a tool, called easy Neurons, which offers rich and intuitive GUI (Graphical User Interface) for creating and training neural networks. NFPR is created as an application sample within easy Neurons tool. how to create and test NFPR using two step wizards. Step 1. Define recommendation rules. In this step user (teacher) loads all domain topics, prerequisites and possible recommendations, from QTI files, and system generates recommendation matrix QTI (Question and Test Interoperability specification) defines a typical format for the representation of assessment content and results.3 Generated recommendation matrix contains all possible combinations of prerequisite relationships between domain topics, and the teacher selects recommendation for each combination (Fig. 5), thus creating a recommendation rule. Each row in the recommendation matrix represents one recommendation rule. Each field contains the name of a domain topic appearing in the corresponding rule, whereas its color indicates the knowledge level of the topic (expressed as a fuzzy set): green for EXCELLENT, yellow for GOOD and red for POOR. Once rules are defined, user clicks the Next button, and neural network and training set are automatically created. Step 2. Train and test NFPR In this step, the user (teacher)just has to click the Train button to train the neural network with the training set (created at the end of the previous step), and that is how the neural network learns the rules. When network is trained, the user can load some student's test results from a QTI file with test results and see the recommendations. The trained neural network can be serialized as a Java object and used as a Java component in any TEL application. It provides a simple API with only two methods for setting the input and getting the output (i.e., recommendation) from the network. The following sample code illustrates how easy it is to use the created neural network with an external, e.g., TEL application:

If end users (teachers) wish to change the pedagogical strategy, the network needs to be retrained. Existing rules can be modified, new domain topics and learning styles can be added, and even new pedagogical criteria can be introduced. To accomplish this teacher just needs to re-run the NFPR wizard.

## VI. EVALUATION

Despite the increasing number of systems proposed for recommending learning resources, a closer look at the current status of their development and evaluation reveals a lack of systematic evaluation studies in the context of real-life applications. As indicated in (Manouselis et al., 2010), more than a half of the analyzed systems, namely 12 out of 20 the authors considered, were still in the design or prototyping stage of development, while only 10 systems were reported as being evaluated through trials that involved human users. Another observation is that, very often, experimental investigation of the recommendation algorithms does not take place, although it is a common evaluation practice in recommender systems examined for other domains (e.g., Breese, Heckerman, & Kadie, 1998). Deshpande and Karypis (2004), Papagelis, Plexousakis, and Kutsuras (2005), and Herlocker, Konstan, Terveen, and Riedl (2004), indicate that careful testing and parameterization has got to be administered before a recommender system is finally deployed in a real-world setting. One of the main reasons is that the performance of recommendation algorithms seems to depend on the particularities of the application context. Hence it is advised to experimentally analyze recommender system before its actual deployment. Following this advice, NFPR was evaluated at the University of Belgrade with a group of 24 teachers. The group has been introduced to the idea of pedagogical rules based on test results as well as the steps needed to create pedagogical rules and save them as expert knowledge. Then the group was introduced to the NFPR tool and its features, and asked to create a set of pedagogical rules that reflects their pedagogical strategy. Participants were then asked to assess the recommendations given by NFPR, by using the previously prepared set of test results. These test results which correspond to typical student's knowledge levels, were created by the teachers based on their teaching experience.

## VII. CONCLUSION

Recommender systems are rapidly becoming an important tool especially on the Web. Recommender system developers have encountered some problems which are currently attractive research areas in the data mining and information retrieval topics for the researchers. The first challenge is to improve the accuracy of the recommendations for the customers. Another challenge is to improve the scalability of the recommendation algorithms. SVD proposes better results than traditional collective separating calculations more often than not, be that as it may, it incorporates computationally very expensive matrix calculations and this makes SVD-based recommender systems less suitable for large-scale systems. In this thesis study, SVD-based recommendation techniques are compared with experiments and some new approaches are introduced to this technique. The first contribution we have proposed is the categorization of items and users. Our experiments indicated that, thing and client order increments both the suggestion quality and speed execution of the SVD method. Moreover, we adopted the tags to the traditional 2-Dimensional SVD approach. By this way, we've chance to research the effect of dimension (tags) to the SVD recommendation performance. Our experiments illustrated that, tags also increase the performance to some extent.

.


## ACKNOWLEDGEMENTS

The authors are thankful to Barry, Gupta D Goldberg.K for providing the necessary facilities for the preparation of the paper. Also thanks to IJRES Journal staffs to publish this paper.


## REFERENCES


[1] J. Ben Schafer, Nathaniel Good, and Joseph Konstan et. al. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the 1999 National Conference of the American Association of Artificial Intelligence, pages 439–436, 1999.

[2] Berry, M. W., Dumais, S. T., and O'Brian, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review, 37(4)*.

[3] Gupta, D., and Goldberg, K. (1999). Jester 2.0: A Linear Time Collaborative Filtering Algorithm Applied to Jokes. In *Proc. of the ACM SIGIR '99*.

[4] Hill, W., Stead, L., Rosenstein, M. and Furnas, G.:1995, ‗Recommending and evaluating choices in a virtual community of use'. In: *CHI '95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 194- 201.

[5] Shardanand, U. and Maes, P.: 1995, ‗Social Information Filtering: Algorithms for Automating "Word of Mouth"'. In: *CHI '95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 210-217.

[6] Terveen, L. and Hill, W: 2001, ‗Human-Computer Collaboration in Recommender Systems'. In: J. Carroll (ed.): *Human Computer Interaction in the New Millenium*. New York: Addison-Wesley.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In Proc. of ACM Conference on Computer Supported Cooperative Work, 1994.

[8] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. In *Proc. of the ACM EC'00 Conference*. Minneapolis, MN, pp. 158-167.

[9] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Application of Dimensionality Reduction in Recommender System—A Case Study. In *ACM WebKDD'00 (Web-mining for Ecommerce Workshop)*.

[10] Badrul Sarwar, George Karypis, Joseph Konstan, John Reidl: Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems

[11] Schafer, J. B., Konstan, J. and Riedl, J.: 1999, ‗Recommender Systems in E-Commerce'. In: *EC '99: Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, pp. 158-166