Appl Intell
DOI 10.1007/s10489-017-1051-8

Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques

Aleksandra Klašnja-Milićević[1] · Mirjana Ivanović[1] · Boban Vesin[2] · Zoran Budimac[1]

Abstract Personalization of the e-learning systems according to the learner's needs and knowledge level presents the key element in a learning process. E-learning systems with personalized recommendations should adapt the learning experience according to the goals of the individual learner. Aiming to facilitate personalization of a learning content, various kinds of techniques can be applied. Collaborative and social tagging techniques could be useful for enhancing recommendation of learning resources. In this paper, we analyze the suitability of different techniques for applying tag-based recommendations in e-learning environments. The most appropriate model ranking, based on tensor factorization technique, has been modified to gain the most efficient recommendation results. We propose reducing tag space with clustering technique based on learning style model, in order to improve execution time and decrease memory requirements, while preserving the quality of the recommendations. Such reduced model for providing tag-based recommendations has been used and evaluated in a programming tutoring system.

✉ Boban Vesin
boban.vesin@ntnu.no

Aleksandra Klašnja-Milićević
akm@dmi.uns.ac.rs

Mirjana Ivanović
mira@dmi.uns.ac.rs

Zoran Budimac
zjb@dmi.uns.ac.rs

[1] Faculty of Sciences, Department of Mathematics and Informatics, University of Novi Sad, Novi Sad, Serbia

[2] Department of Computer & Information Science, Norwegian University of Science and Technology, Trondheim, Norway

Keywords E-learning · Personalization · Recommender systems · Collaborative tagging

1 Introduction

"Web 2.0" has made new significant concepts for education on the Internet. Collaborative tagging, as a fragment of Web 2.0, is an important tool for classification dynamic content [1]. Progressive user experiences and collective intelligence impact the conception of authority in educational systems. Based on the current trend, present in the recent period, we investigate the process of integration recommender systems and collaborative tagging techniques into e-learning systems.

The collaborative tagging process creates a set of tags, usually named folksonomies, useful for describing a resource [2]. In order to define relevancy, folksonomies can add semantics to resources through a social tagging process. The folksonomies allow adding of semantics without the support of individual manual indexers or automated keyword generators [3].

The tagging process permits the user to utilize his own words or concepts of great significance. In e-learning environments, learners can take advantage from creating tags in several essential manners: first, tagging has proved as an adequate meta-cognitive strategy that engages learners more successfully in the learning process. By highlighting the most important part of a text, learners can remember it better. Furthermore, tagging activities could inspire learners to more thoroughly engage themselves in a learning process and could help them understand learning content [4]. Gathering of learners' opinions about specific resources, could give more comprehensible recommendations about the resources for other learners.

Published online: 23 August 2017

An increasing number of users, exposing their preferences for specific content via the tag assignments, initiated the appearance of tag-based profiling approaches. As a result, tags have become exciting and valuable information to improve recommender system's algorithms. The key improvement of tag-based recommender systems [5] can be expressed with examination of tags, determination of user preferences and interests and make available recommendations of which resources could be interesting for the user.

This paper analyzes possible integration of recommender systems and collaborative tagging in an intelligent web-based programming tutoring system, named Protus (PRogramming TUtoring System) that has possibilities to take into account pedagogical aspects of the learner. The research presented in this paper focuses on the proper selection of collaborative tagging techniques that could increase motivation of learners and their comprehension of the learning content. We incorporate tag-based recommendation into a learning approach implemented in Protus. Such approach allows the system to quickly identify the most suitable supplementary material for learners and present it to them. Tagged parts of lessons provide information that will be used for further recommendation of adequate material to a learner. Therefore, it can be expected that personalized recommendations in the system will be consistent with the learner's interests and previously acquired knowledge. This approach should also simplify collaboration and interaction between learners.

The most appropriate model ranking based on tensor factorization technique has been calculated, used and evaluated in our tutoring system to provide the most suitable recommendation of learning resources to learners The clustering technique based on learning style model has been implemented to reduce tag space for providing recommendations. We argue that improved collaborative tagging technique will shorten the execution time and decrease memory requirements, while at the same time the quality of the provided recommendations will be guaranteed.

We performed several experiments using our proposed approach to prove that learners' motivation, as an important concept in learning process, can be enhanced with generated recommendations. The performance of the system has also been evaluated from the view-points of both teachers and learners. In addition, we investigated collaborative tagging processes and how position of a tag correlates with its expressiveness. We performed a semantic analysis of tags to better understand their different utilization.

The paper is organized as follows. Section 2 summarizes the results of contemporary research about using collaborative tagging for learning process and knowledge acquisition. Section 3 provides descriptions of tag-based recommender systems suitable for applications in e-learning environments and our proposed approach used for personalization process in programming domain. Experimental evaluation of integrated recommender system based on collaborative tagging techniques into our tutoring system is presented in Section 4. The concluding remarks are given in Section 5.

2 Brief literature review

Delivering personalized learning resources often includes tasks in which a system recommends a learning material to an active learner [6]. The anticipated benefits of tag-based recommendation within e-learning systems were presented in several studies along with the potential of social tagging for increasing metadata descriptions [7]. Other benefits include improvement of search and data mining of educational resources.

Recommendations provide learners with personal learning environments and connect them to learner networks where they can collaborate by using the available tools [8]. Author in [9] examined different recommender strategies and their applicability in personal learning environment settings and presented a possible strategy for generating recommendations of educational content.

Several authors investigated the possibility of using social tagging in education as a support for learning practices. The expected benefits of tagging for describing educational resources were presented in [10–14]. These methodologies can be regarded as appropriate applications of social tagging for improvement of e-learning systems.

The authors in [15] presented an approach for estimating the value and usability of an existing learning material. They presented a methodology to investigate the learners' individual and social behavior while interacting with learning objects. The authors used social tagging to utilize context data on the way to make set of learning objects more useful, and thus improve their reuse.

Efficiency of social tagging depends heavily on motivation of learners to express their opinions about educational content in tag form. The authors in [16] investigated the relation between users' motivation for tagging and the resulting enlarged metadata descriptions. The authors proposed a methodology that aims to evaluate whether motivation of users' tagging can influence the extensions of educational resources and the resulted folksonomy.

Collaborative tagging could enlarge metadata of learning material but more in depth studies are needed regarding the use of tagging within specialized intelligent tutoring systems [17]. In this paper, we will present intelligent tutoring system (Protus) that is used for a programming course and we will present how recommendation of learning resources could be achieved by using social tagging techniques.

Intention of obtained experimental results is to indicate how graph-based and tensor based approaches for tag recommendation can be applied to e-learning environments. The goal was not to increase attention level of learners, or their entertainment, but to improve the teaching process and make the learning more efficient.

Novelties are introduced in recommendation method and possibilities for its application in the learning system are presented. Main novelties are: reducing tag space with clustering based on identified learning styles, speeding up of tensor based recommenders with such reduced tag space and improving the quality of recommendations with ranking tensor factorization.

3 Applying tag-based recommender systems to e-learning environments

By tradition, e-learning systems should provide custom instructions for learners on finding the appropriate learning materials, personalized and adapted to the needs, knowledge, talents and learning style of learners [18]. Their main limitations are:

- there is a need to develop specific and different types of e-learning systems in accordance with the domain for which they are designed and
- it is not possible to embed into the systems all the possibilities that cover the specific requirements and necessities of each learner in each part of the learning course.

For that reason, a dynamic support is desirable in order to recommend learners appropriate activities, necessary for achieving their learning goals. Furthermore, these systems should have the possibility to discover applicable content on the Internet, which would adapt, customize and personalize within learners and professionals' domains.

Our aim in this chapter is to examine the appropriateness of integration recommender systems based on tags into an innovative framework for e-learning purposes. The improvement of such integration considering the e-learning systems is mostly evident in their ability to recognize the performance of individual learners and in their capability to encourage learners in their own learning path according to the most appropriate tags and learning items [19].

Writing tags can be very useful for learners in several significant ways [20–22]:

1. Tagging, as a metacognitive strategy, has the potential for engaging learners in the learning process.
2. For both educators and administrators, the information delivered by tags makes accessible insights on learner's knowledge and activities.
3. Tagging represent a reflective practice that can give learners an opportunity to summarize their ideas, while receiving efficient support through viewing other learners' tags/tag recommendations.
4. Collaborative tags in a tutoring system could be used by content authors to recognize learners' progress on different levels. Tags could be used for analysis of learners' comprehension or misconception of a learning material (at the individual level), and to identify the overall progress of the class (at the group level).
5. E-learning systems currently lack efficient support for self-organization and annotation of teaching material [20]. However, learners in the campuses can be seen engaged in a number of annotation activities, such as writing notes, highlighting text, or bookmarking pages. Tagging encourages learners' engagement as it allows them to add comments, corrections, links, or join discussions.

In the remainder of this section, we present ranking based on factorization model (RTF - *Ranking with tensor factorization* algorithm) for providing tag-based recommendations with emphasis on their application in e-learning environments. This model was chosen as it performed the best in the comparative study presented in the Section 4. Details of the remaining techniques that were used in the study could be found in [2, 23–31].

The experiments have been performed to analyze the suitability of RTF technique for applying tag-based recommendations in e-learning environments as well as comparing graph-based and tensor based approaches for tag recommendation applied to e-learning environments. As it will be presented in the Section 4, the advantage of selected technique is that it shortened the execution time and decreased memory requirements, while in the same time the quality of the provided recommendations have been preserved.

3.1 Personalized recommendations in a programming tutoring system

Programming tutoring system named Protus has been implemented and used at our Department. Protus allow learners to learn with personalized learning material prepared within appropriate course and to test the acquired knowledge. The first fully implemented and tested version of the system contained introductory Java programming course [32]. Java is chosen because it is a representative example of an object-oriented language and is therefore suitable for the teaching object-oriented concepts. The course is designed for learning programming basics for learners with no object-oriented programming experience. Architecture of Protus, presented in Fig. 1, contains

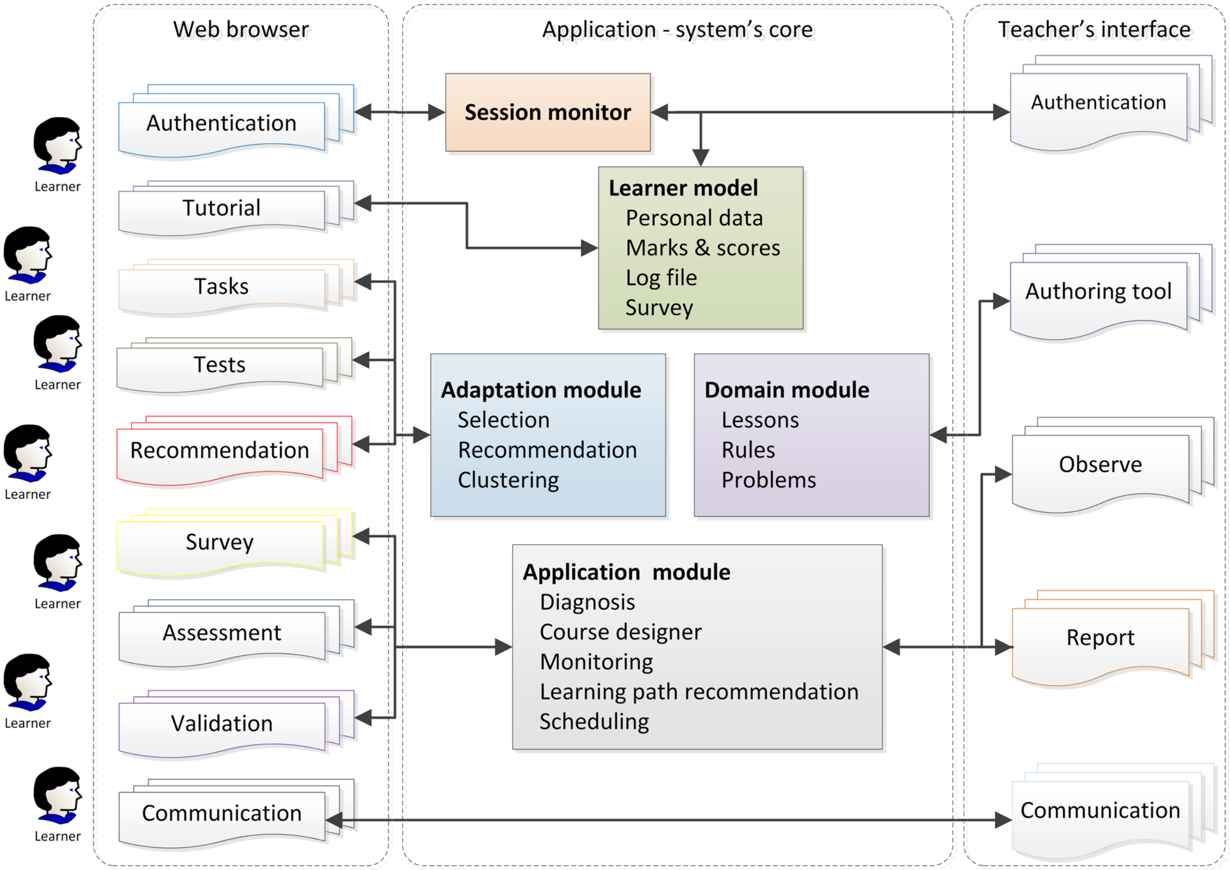


Fig. 1 Protus system architecture

five practical components: learner model, session monitor, domain module, application module and adaptation module.

Learner model is used for collecting useful information about an individual learner. Data from learner model are arranged in three categories according to shared qualities and characteristics [39]:

- Personal information - data about learner supplied during the registration process, like: personal data, preferences, previous knowledge, etc. The learner can edit this data during the registration to the system.
- Performance information - data monitored and recorded by the system, like: cognitive style, information about learner's progress and current level of knowledge, his/her misconceptions and the overall performance.
- Learning history information – data about lessons and tests learner has studied, learner's interaction with the system, the assessments for particular lesson etc. The data is automatically retrieved from the Protus system.

In order to retain track of the learner's activities and progress, **session monitor** gradually updates the learner model during sessions. It is responsible for detection and correction learner's errors and redirection of the session consequently.

Domain module represents storage of learning concepts and objects from the domain, teaching material and tests. The structure of information content is defined by the domain module. Java programming course in Protus is divided into six units, each containing several lectures [32]. Each lecture (out of eighteen) contains several learning objects (LOs): explanations (tutorials), illustrative examples that explain theoretical parts and illustrates key concepts. Additional tests were implemented for inspection of learners' knowledge.

Application module, as essential part of an adaptive and personalized system, is designed to adapt the learning content to the individual learners. It supports different techniques and strategies aimed to select or compute particular navigation patterns through the learning resources based on the learner model.

Adaptation module generates personalization based on recommender system [33]. It consists of three components for generating personalized recommendations (Fig. 2):

A. **A learner-system interaction module** collects and prepares data about learner's activities: visited pages, sequential patterns, grades earned, test results etc.

B. **An off-line module** applies learner model and data analysis strategy to identify the learner's objectives and content profiles. The first step is determination of learning style for each user based on the initial questionnaire. Learning content can be filtered based on learning styles, current status of the course and learner's affiliation.

C. **A recommendation engine** creates a recommendation list for the groups of learners, as even for learners with similar learning interests, their capability to resolve a task can vary due to the variety of their knowledge level. As a first step to cluster learners, we carry out a data clustering technique based on their learning styles. Then, a recommendation list is created based on:

- The learners' tags, or
- Recognized learning patterns, through mining the frequent sequences by the AprioriAll algorithm. Personalized recommendations of a teaching material are generated based on the ratings of frequent sequences and with use of collaborative filtering method, provided by the Protus system.

The task of tag-based personalized recommendation is to offer a learner a personalized ranked tags list for a specific item.

3.2 Tag-based personalized recommendation using ranking with tensor factorization technique and learning style clustering

The recommendation component of the Protus system has been implemented in order to recommend most popular tags and experimentally compare this tag-list with the list from the previous version of the system [34]. Based on comprehensive comparisons of techniques that can be used to recommend tags, in the rest of this section, we will show tag-based personalized recommendation using ranking with tensor factorization (RTF) technique. The recommendation process consists of three phases:

- Creating initial tensor
- Calculating tensor factorization
- Generating a list of recommended items

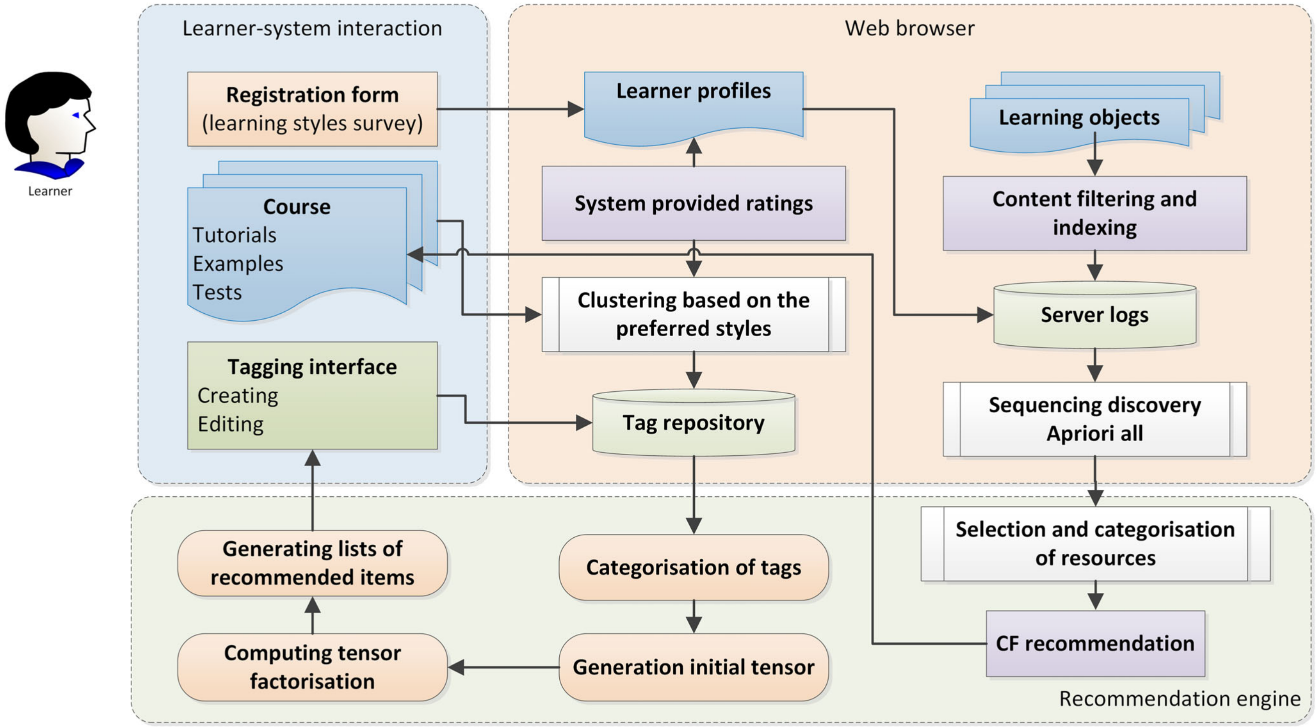

Learner-system interaction
Web browser
Learner
Registration form (learning styles survey)
Course
Tutorials
Examples
Tests
Tagging interface
Creating
Editing
Learner profiles
System provided ratings
Clustering based on the preferred styles
Tag repository
Learning objects
Content filtering and indexing
Server logs
Sequencing discovery Apriori all
Selection and categorisation of resources
CF recommendation
Generating lists of recommended items
Categorisation of tags
Computing tensor factorisation
Generation initial tensor
Recommendation engine

Fig. 2 The recommendation component of Protus system

3.2.1 Creating initial tensor

We have been used 3-dimensional data of learners, items (learning objects) and tags in order to generate the initial tensor. The third order tensor A$\in$ $R^{|L|\times|I|\times|T|}$ represents this data where $|L|$, $|I|$ and $|T|$ are the dimensions of the data of learners, items and tags, respectively. A value $(\mathrm{A})_{litk} = a_{lit}$ can represent, for example, how many times learner l tagged an item i with a tag t. In this phase following steps can be recognized:

- Creating a learner set based on learning styles.
- Determining a set of tags used by the learners.
- Resolving the item set, tagged with the tags by the learners.
- Iterating through the learners, items and tags. Determination if a current item is tagged by the current learner and current tag and mark the obtainable association in the tensor.
- If a learner does not tag an item with a tag keeping the empty relations of a current learner.

Our innovative approach can be observed as a process of creating a learner set, based on the learning styles, which other similar algorithms in e-learning environments haven't applied. Namely, essential point is that the clusters for the classification of the data set were based on learners' learning styles. It is obvious that different learners have different preferences, needs and approaches to learning. Therefore, it is very important to accommodate the system for the different styles of learners through learning environments that they prefer and find more efficient. Learning styles can be defined as unique manners in which learners begin to concentrate on, process, absorb, and retain new and difficult information. They are distinctive individual patterns of learning, which vary from person to person. After the clusters are formed based on learning styles categories, tagging recommendations have been validated to discover useful knowledge for the learner in terms of his/her behaviour in respect to different psychological categories.

We selected the Felder-Silverman learning style model (FSLSM) to be applied in Protus. The learning style determination is done using the Index of Learning Styles (ILS), a data collection tool for investigating learning styles [35]. The ILS is a 44-question, freely available, multiple-choice learning styles instrument, which assesses variations in individual learning style preferences across four dimensions: *Information Processing*, *Information Perception*, *Information Reception* and *Information Understanding.* The results gathered by this questionnaire are used to define appropriate clusters, which determine groups of learners with the similar learning style preferences.

Algorithm 1 Generating initial tensor

```
Input:
    List of triples <learners, tags, items>;
    Dimensions of learners, tags, and items: $|L|$ , $|T|$ , and $|I|$;
Output:
Initial Tensor A
Determine a set of learners;
for $l \leftarrow 0$ to $|L|$do
    get learner $l$, add it to the learners set learners
end
Get a tagset used by the learners;
Get an item set tagged by the tags and by the learners;
initialTensor =new double[$|L|$] [$|T|$][$|I|$];
int learnerIndex = 0;
for each learner $l \in$ learners and learnerIndex < $|L|$do
    for tagIndex ←0 to $|T|$do
        for itemIndex ←0 to $|I|$do
            if a current learner tagged a current item
            with a current tag
            then
                initialTensor[tagIndex][learnerIndex]
                [itemIndex]++;
            end
        end
end
    if current learner == $l$ then
        StoreEmptyRelations (initialTensor);
    end
    learnerIndex++;
End
```

3.2.2 Calculating tensor factorization

The initial tensor has to be defined in order to calculate a tensor factorization. Then the following steps should be applied:

- First, the initial tensor needs to be divided in the 3 mode matrices.
- Second, the dimensions are reduced for each mode matrix. A core tensor is computed according to these reduced matrices.
- As a final point, the matrices are transformed and multiplications are applied. The factorized tensor is computed.

Algorithm 2 Computing tensor factorization

```
Input:
    Initial tensor A, dimensions of learners, tags, and items: c1, c2, c3
Output:
    Define factorized tensor $\hat{A}$
    Split initial tensor A into the 1-mode $A_{(1)}$, 2-mode $A_{(2)}$, and 3-mode $A_{(3)}$ matrices, and compute SVD for each mode;
    Compute a core tensor: reduce dimensions and execute multiplication of matrices;
    Create the modification matrix $C_i = A_{(i)} A_{(i)}^T$ for each mode $1 \leq i \leq 3$
    Calculate $U^{(i)}$ by diagonalizing $C_i$ , $1 \leq i \leq 3$
    Taking its eigenvectors: $\hat{U}^{(1)}$, $\hat{U}^{(2)}$and$\hat{U}^{(3)}$
    Estimate the approximation of the core tensor: $\hat{S} = A \times_1 \left(\hat{U}^{(1)}\right)^T \times_2 \left(\hat{U}^{(2)}\right)^T \times_3 \left(\hat{U}^{(3)}\right)^T$
    Compute factorized tensor by: $\hat{A} = \hat{S} \times_1 \hat{U}^{(1)} \times_2 \hat{U}^{(2)} \times_3 \hat{U}^{(3)}$
```

3.2.3 Generating a list of recommended items

The process of tag recommendation is useful in prediction which tags a learner prefers to use for tagging an item. It means that system for tag recommendation must predict the numerical values of the factorized tensor specifying how much the learner favors a particular tag for an item. The system provides a learner with a personalized list of the best N tags for a specific item, instead of predicting a particular element.

3.3 Tagging interface in Protus system

Specialized learning resources have been created by teachers, using the authoring tools of Protus. Along presentation of resources, user interface of Protus also provides options for tags creation and review for each resource (Fig. 3).

The menu for tagging in Protus (Fig. 4) allows learners to participate in annotation activities: adding comments, links, and corrections or join shared discussion. The learner creates a tag by selecting learning object and entering keywords in the appropriate textfield. Protus allows participants to enter as many tags as they wish. The tags could be separated by commas or spaces, therefore not restricted to a single word.

As opposed to many popular tagging systems, which only permit single word as tags, Protus allows the use of multi-word tags. Every time a learner returns to particular learning resource, the list of tags previously made will reappear (Fig. 5). Every click on individual tag gives the learner one of two options: *Edit* or *Remove*.

The *Other's tags* list, representing the most popular tags of other learners, appears under *Others' Tags*. Learners can add choose tags from the *Others' tags* and add it to *My tags* list. An example of these functionalities is shown in Fig. 6.

Based on the findings from performed comparative analysis of tag-based recommender algorithms, the "recommended tags" list is generated according to the learners' and based on Ranking with Tensor Factorization model (Fig. 7).

4 Experimental research

The goal of the experiments was to analyze the suitability of RTF technique for applying tag-based recommendations in e-learning as well as comparing graph-based and tensor based approaches for tag recommendation applied to e-learning environments. Defined model for providing tag-based recommendations has been used and evaluated in our programming tutoring system.

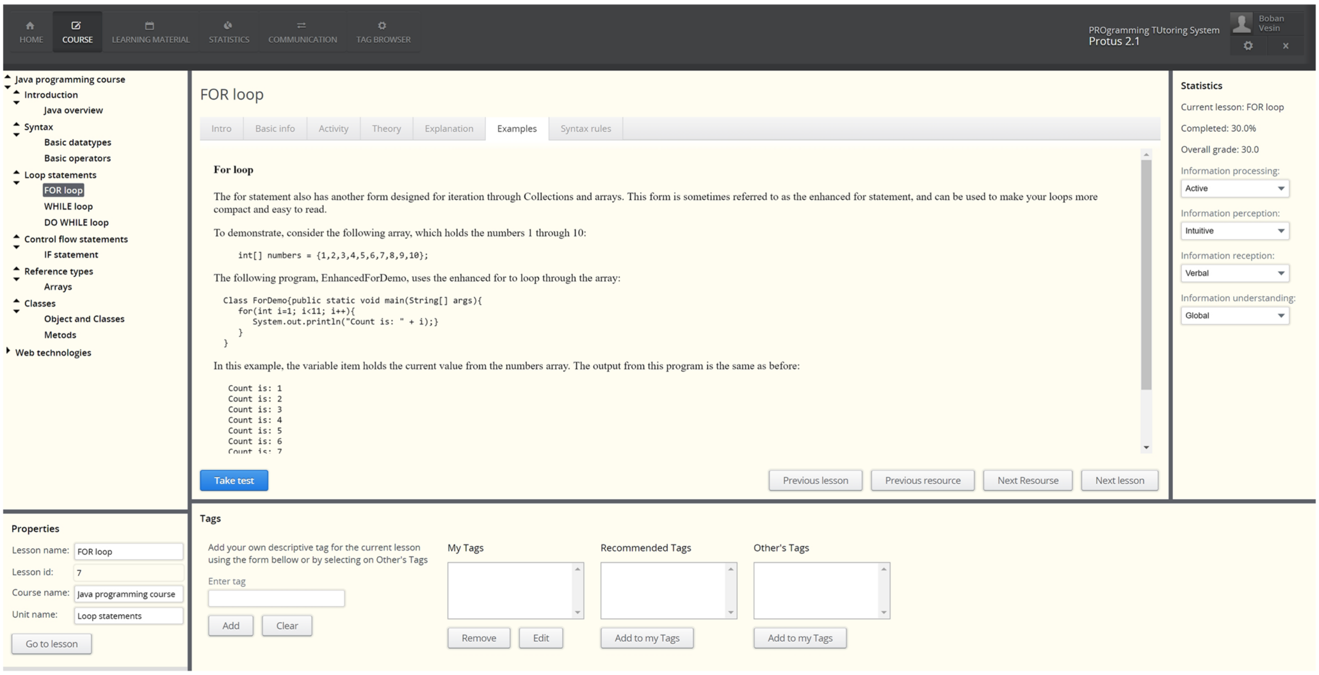
In addition, we have carried out some experiments to evaluate the performance of the system from the points of view of both teachers and learners. The goal was to:

- investigate the performance of RTF complexity runtime compared to the FolkRank (Section 4.5.),
- investigate collaborative tagging processes and how position of a tag correlates with its expressiveness (Section 4.6),
- perform a semantic analysis of tags to better understand different utilization of tags (Section 4.7) and
- perform expert validity study to systematically verify the relationship between learning comprehension and learner data tagging (Section 4.8)

The experiments were performed on an educational dataset, consisting of 120 learners, high school students, participants of the Center for Young Talents project, supported by the Schneider Electric DMS NS company, Novi Sad. The experiment took a month, in September 2015. Involved students were programming beginners that successfully passed the basic computer literacy courses at the High school for Electrical engineering, Novi Sad.

4.1 Data clustering

Aiming to cluster the learners into a sub-classis, we encouraged them to fill the Felder-Soloman ILS (Index of Learning Styles) Questionnaire [36]. This questionnaire contains a



FOR loop

For loop

The for statement also has another form designed for iteration through Collections and arrays. This form is sometimes referred to as the enhanced for statement, and can be used to make your loops more compact and easy to read.

To demonstrate, consider the following array, which holds the numbers 1 through 10:

```
    int[] numbers = {1,2,3,4,5,6,7,8,9,10};
```

The following program, EnhancedForDemo, uses the enhanced for to loop through the array:

```
 Class ForDemo{public static void main(String[] args){
    for(int i=1; i<11; i++){
       System.out.println("Count is: " + i);}
    }
 }
```

In this example, the variable item holds the current value from the numbers array. The output from this program is the same as before:

```
  Count is: 1
  Count is: 2
  Count is: 3
  Count is: 4
  Count is: 5
  Count is: 6
  Count is: 7
```

Statistics

Current lesson: FOR loop

Completed: 30.0%

Overall grade: 30.0

Information processing: Active

Information perception: Intuitive

Information reception: Verbal

Information understanding: Global

Properties

Lesson name: FOR loop

Lesson id: 7

Course name: Java programming course

Unit name: Loop statements

Tags

Add your own descriptive tag for the current lesson using the form bellow or by selecting on Other's Tags

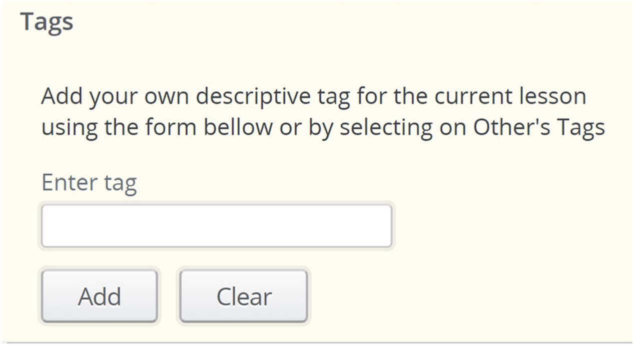
Enter tag

My Tags

Recommended Tags

Other's Tags

Fig. 3 Interface of Protus

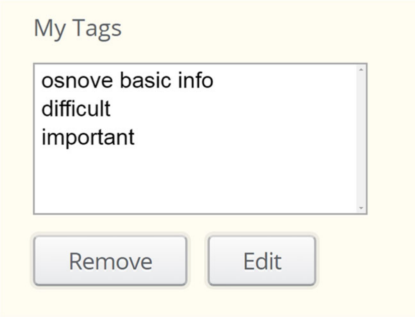


Tags

Add your own descriptive tag for the current lesson using the form bellow or by selecting on Other's Tags

Enter tag

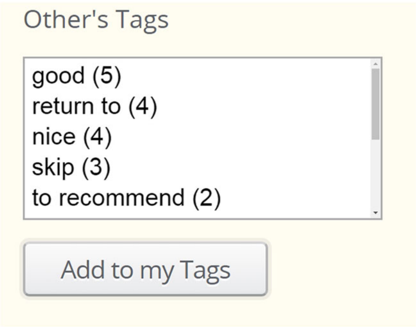
Fig. 4 Interface for creating tags



My Tags

osnove basic info
difficult
important

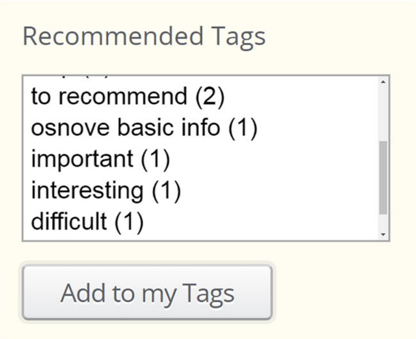
Fig. 5 *My tags* list



Other's Tags

good (5)
return to (4)
nice (4)
skip (3)
to recommend (2)

Fig. 6 List of Other's tags



Recommended Tags

to recommend (2)
osnove basic info (1)
important (1)
interesting (1)
difficult (1)

Fig. 7 List of recommended tags

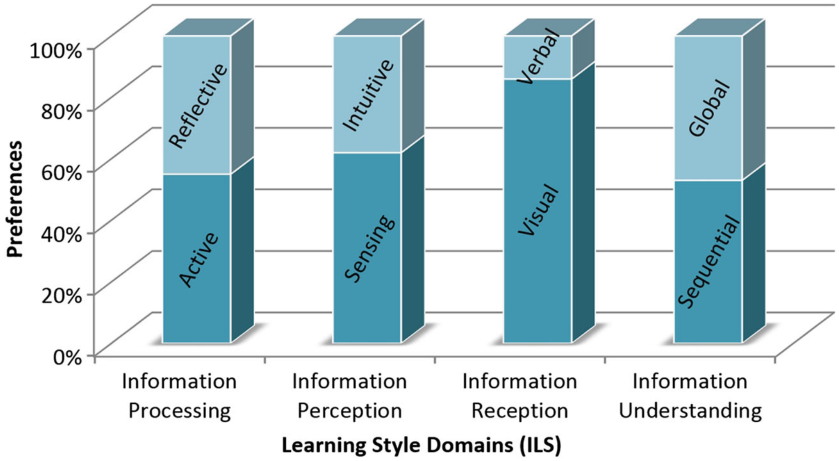


Fig. 8 Learning styles results

set of 44 questions over 4 categories on behalf of learning preferences and styles: *Sensing* versus *Intuitive*, *Active* versus *Reflective*, *Visual* versus *Verbal*, and *Sequential* versus *Global*. Figure 8 shows the association of learners identified preferences according to learning styles through all 4 categories.

It was possible to define appropriate clusters based on the results of the questionnaires, which determined learner profiles for 120 learners. Clusters were formed for different combinations of learning styles within the three categories (Table 1).

Information processing Category was omitted in order to increase the number of learners in separate clusters and to obtain more relevant recommendation data. The usage data for each cluster is presented in Table 2. Number of learners, number of learning objects and their tags, average number of tags for separate learner and average number of tags per learning object were presented.

4.2 Experimental protocol and evaluation metrics

The proposed models are evaluated by maintaining the part of the data set specified as the test set, and constructing prediction models from the remains data, specified as the training set. The data set was randomly divided into an experiment and a test set with respective sizes 80 and 20 percent of the observed set. Classical metrics of precision and recall were chosen as performance measures for the item and tag recommendations [26]. Precision and recall are typical metrics used for evaluation of information recovery systems in different educational settings. Mapping in a recommender system manner of speaking, metrics of precision and recall are defined concerning the estimation of the top N recommendations.

Recall and precision can be calculated for a test user u, the randomly picked item i and the set of recommended tags $\hat{T}(u,i)$, as:

$$precision\ (\hat{T}(u,i)) = \frac{1}{|U|}\sum_{u\in U}\frac{\left|tags\ (u,i)\cap\hat{T}(u,i)\right|}{\left|\hat{T}(u,i)\right|} \tag{1}$$

$$recall(\hat{T}(u,i)) = \frac{1}{|U|}\sum_{u\in U}\frac{\left|tags(u,i)\cap\hat{T}(u,i)\right|}{|tags(u,i)|} \tag{2}$$

Precision represents the relation of the recommended relevant tags number or items from the top N list entries to N (formula 1).

Recall represents the relation of recommended relevant tags or items number from the top-N list of the total number

Table 1 Identification of clusters based on different learning styles

Cluster1	Cluster2	Cluster 3	Cluster 4	.....	Cluster 8
Sensing	Sensing	Sensing	Sensing		Intuitive
Visual	Visual	Verbal	Verbal	Visual	
Sequential	Global	Sequential	Global	Global	
16 learners	20 learners	14 learners	10 learners	.....	18 learners

Table 2 Characteristics of the data sets per cluster

	Clust1	Clust2	Clust3	Clust4	Clust5	Clust6	Clust7	Clust8
Num. of learners	16	20	14	10	16	11	15	18
Num. of LO	72	72	72	72	72	72	72	72
Num. of tags	1328	1804	1100	850	1380	898	1373	1458
Avg. Num. of tags per learners	18.4	25.1	15.3	11.8	19.2	12.5	19.1	20.3
Avg. Num. of tags per LO	83.0	90.2	78.6	85.0	86.3	81.6	91.5	81.0

of items or tags set by a given user from the estimation part of a data set (formula 2).

4.3 Experimental settings

In our experiments, the classical measures precision and recall were chosen in order to evaluate several suitable recommendation techniques in e-learning environments. At the beginning, we will describe the settings used to execute algorithms being evaluated. Later, we will present and analyze the results of performed experiments and evaluations. In advance to start the complete experimental evaluation of selected algorithms (Adapted Page-Rank, FolkRank, Collaborative based on Tags, Higher Order Singular Value Decomposition - HOSVD, Ranking with Tensor Factorization - RTF) we will first determine the sensitivity of the important parameters for each algorithm. For the remaining experiments optimal values of these parameters were fixed and used. The analysis included all eight clusters.

Adapted PageRank algorithm We stopped calculation at the moment when the distance between two consecutive weight vectors was less than 10^{-6}. The parameter d was 0.7. In $\vec{p}$, we gave higher weights to the user and the item from the post which was chosen. While each user, tag and item got a preference weight of 1, the user and item from that particular post got a preference weight of 1 + |U|and 1 + |I|, respectively.

FolkRank algorithm The equal parameters and preferences were selected as in the Adapted PageRank.

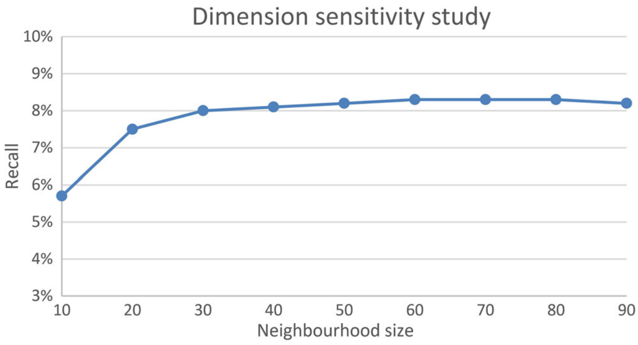


Fig. 9 Recall of collaborative tag-based CF according to the variation of neighborhood size

Collaborative Filtering (CF) based on Tags algorithm Based on the user-tag matrix, $\pi_{UT}Y$ the neighborhood has been computed. The number k of best neighbors represents the parameter that needs to be tuned in the CF based algorithms [33]. We investigated the effect of the recalls variation, according to the neighborhood size k which is carefully associated with generation of tag preferences. A graph of changes the neighborhood size from 10 to 90 is shown in Fig. 9. When the neighborhood size increases from 10 to 30 recommender quality improves, appropriately. Nevertheless, an increase of the k value did not perform statistically important enhancements after the neighborhood of size 30. Once the number of nearest neighbors' k (NNk), is acceptably large, the recommendation quality for each user is not changed by any further increment of the k parameter. According to this trend, we selected 30 as our optimum choice of the neighborhood size.

Higher Order Singular Value Decomposition (HOSVD) algorithm. Since there is no straightforward way to find the optimal values for *c1*, *c2* and *c3*, we follow the technique presented by [30] that a 70% of the original diagonals of $X(1)$, $X(2)$ and $X(3)$ matrices can give good approximations. Thus, *c1*, *c2* and *c3* are set to be the numbers of singular values by preserving 70% of the original diagonals of $X(1)$, $X(2)$ and $X(3)$ respectively in each run.

RTF. We ran RTF with $(k_u, k_i, k_t) \in \{(8, 8, 8); (16, 16, 16); (32, 32, 32)\}$ dimensions, as it was proposed by [28]. The corresponding model is called "RTF 8", "RTF 16", and "RTF32". The other hyper parameters are: learning rate$\alpha = 0.5$, regularization $\gamma = \gamma_c = 10^{-5}$, iterations $iter = 500$. Parameters of the model $\hat{\theta}$ has been initialized with small random values calculated from the normal distribution $N(0,\ 0.1)$.

4.4 Results

We analyzed the prediction quality of User-based Collaborative Filtering, (*Collaborative Filtering based on Tags)*, graph-based approaches (*Adapted PageRank* and F*olkRank* algorithms), and tensor based approaches (*HOSVD* and *RTF*).

Some data transformations need to be performed in order to apply typical CF-based algorithms to folksonomies. These transformations cause the information loss, which can decrease the recommendation quality. Likewise, evaluation of large projection matrices need to be kept in memory, which can be time or space consuming and thus compromise real time recommendations of CF-based methods. Also, the CF algorithm could be changed for each different mode to be recommended, challenging additional effort for proposing multimode recommendations.

FolkRank algorithm is developed on PageRank algorithm. It is evident that FolkRank gives importantly better recommendations of tags than CF. The main reason is ability of FolkRank algorithm to utilize the information that is appropriate to the specific user together with input from other users via the integrating structure of the underlying hypergraph. Regarding the prediction quality of CF, Adapted PageRank and FolkRank approaches (Fig. 10) it could be noticed that FolkRank outperforms both of them. In addition to generally relevant tags, FolkRank algorithm is capable to predict the most appropriate tags for the user in contrary to CF. This is a consequence of the fact that FolkRank technique takes into consideration the vocabulary of the particular user, through the hyper-graph structure, which CF by definition can't achieve. Furthermore, FolkRank method permits mode changing with no additional modification in the algorithm. Additionally, FolkRank, as well as CF-based algorithms, is robust concerning online updates because it does not have to be trained whenever a new user, tag or item appears in the system. On the other hand, FolkRank algorithm is more suitable for systems where real-time recommendations are not demanded, because it is computationally expensive and not trivially scalable. Tensor factorization method, as well as FolkRank, works over the ternary relation of the folksonomy. Even though the phase of tensor reconstruction is expensive, it can be realized offline. When the lower dimensional tensor is computed, the recommendations can be generated faster, which makes this algorithm appropriate for real time recommendations.

In practice, RTF model was much faster in prediction than HOSVD model, mainly due to its need for less dimensions than HOSVD for attaining better quality.

A crucial problem with HOSVD is sensitivity to the number of dimensions and the relations between the user, tag and item dimensions. Selecting the identical dimension for all three proportions causes poor results.

On the other hand, for RTF method it can be chosen the same number of dimensions for user, tag and item. Besides this theoretical analysis, in Fig. 10, the prediction quality of RTF is convincingly better in comparison with the HOSVD. Also, Fig. 10 shows that RTF can achieve similar results as HOSVD even with a small number of 8 dimensions. When the dimensions of RTF are increased to 16 it already performs better than HOSVD in quality. Furthermore whenever the dimensions for RTF have been increasing better results for this method have been obtained. Concerning the prediction quality of FolkRank and RTF algorithms it can be noticed that RTF with 8 and 16 dimensions achieves comparable results whereas 32 dimensions outperform FolkRank on quality.

4.5 Runtime complexity comparison

The general way to predict a top-n list of tags t for a specific user u and item i is based on formula:

$$Top\,(u, i, N) \quad := \quad \underset{t \in T}{\operatorname{arg}\overset{N}{\max}}\, \hat{y}_{u,i,t} \tag{3}$$

where the superscript N represents the number of tags to return. One of the benefits from a factorization model like RTF or HOSVD is that after a model is built, predictions only rely on the model. This leads to faster prediction runtime than with models like FolkRank. We compare the runtime complexity of RTF method to the state-of-the-art tag recommendation method FolkRank [27].

For predicting $\hat{y}_{u,i,t}$ with a factorization model, formula is used:

$$\hat{y}_{u,i,t} = \sum_{\tilde{u}} \sum_{\tilde{i}} \sum_{\tilde{t}} \hat{c}_{\tilde{u},\tilde{i},\tilde{t}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{i}_{i,\tilde{i}} \cdot \hat{t}_{t,\tilde{t}} \tag{4}$$

and the runtime complexity of RTF method is $O(k_U \cdot k_I \cdot k_T)$. Thus, the trivial upper bound of the runtime for predicting a top-n list is $O(|T| \cdot k_U \cdot k_I \cdot k_T)$. The authors in [37] have proven that the runtime complexity for top-n predictions with FolkRank is $O(iter \cdot (|S| + |U| + |I| + |T|) + |T| \cdot N)$ where *iter* is the number of iterations. That means for predicting a personalized top-n list, FolkRank has to pass several times through the whole database of observations. When we compare this to RTF it is obvious that RTF models have a much better runtime complexity as they only depend on the small dimensions of the factorization and on the number of tags. The only disadvantage of RTF to FolkRank, with regarding to runtime, is that it requires a training phase. But as training is usually done offline, this does not affect the applicability of RTF for fast large-scale tag recommendation.

The small Protus dataset and the dataset of the world's largest social music service Last.fm [38] have been used in order to make quality runtime comparison of the selected methods. FolkRank is compared to RTF with an increasing number of dimensions. An empirical runtime comparison for predicting a ranked list of tags for a post can be found in Fig. 11. The runtime of the RTF model is dominated by the dimension of the factorization and is independent of the size of the dataset. The runtime on the Protus dataset and larger Last.fm dataset are almost the same – e.g. for RTF64 7.5 ms for Protus and 9.3 ms for Last.fm. With smaller factorization, the number of tags has a larger impact on the runtime – e.g. for RTF16 it is 0.24 ms vs. 1.08 ms. For the

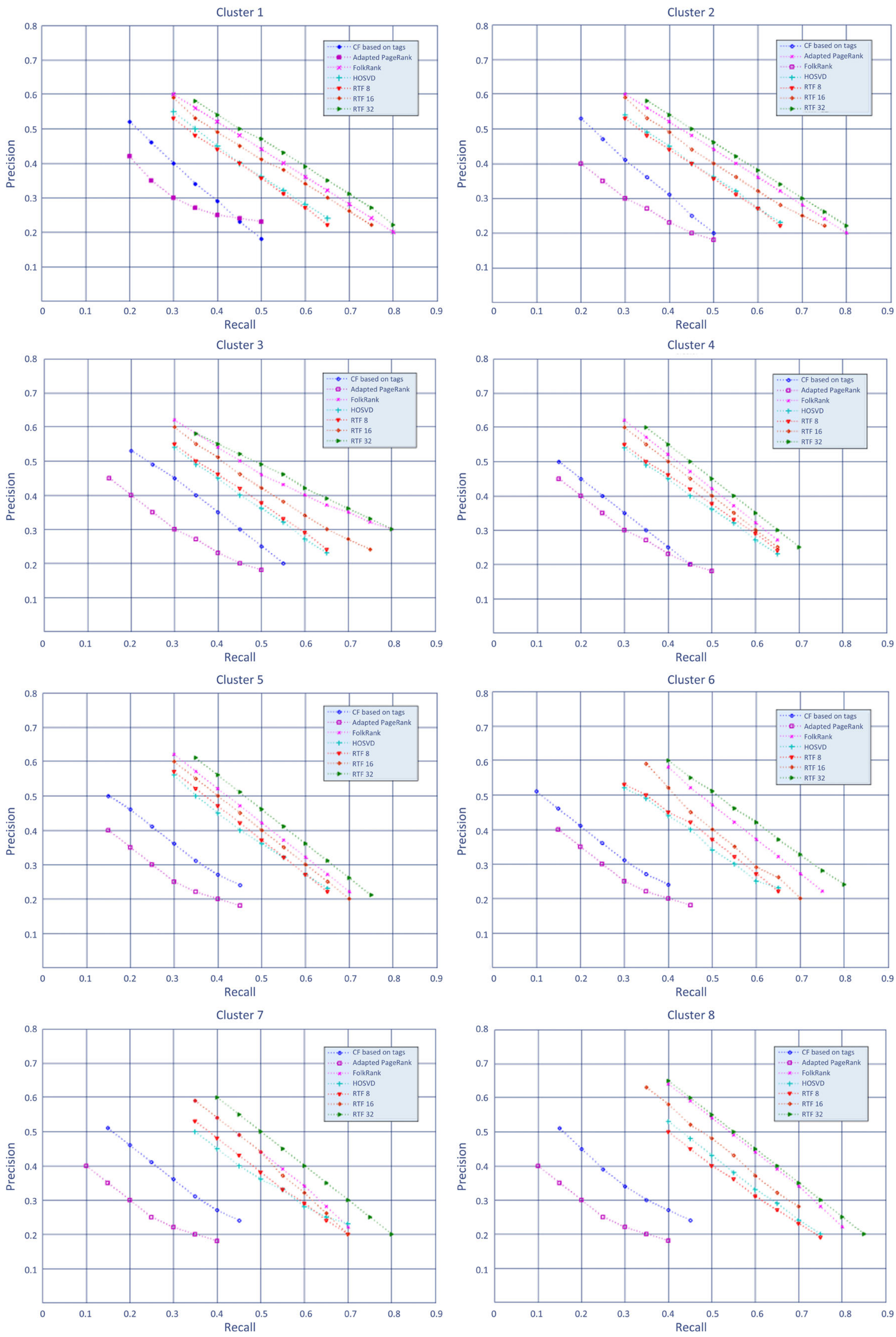


Fig. 10 Results for *Recall* and *Precision* for graph-based and tensor-based methods

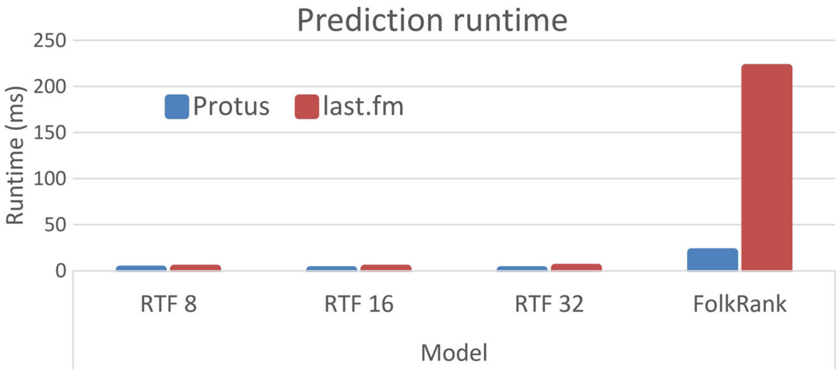


Fig. 11 Runtime comparison for predicting one ranked list of tags

large dataset the runtime of RTF is 220 ms. The reason is that the runtime of FolkRank depends on the size of the dataset and on the very small Protus dataset that leads to a reasonable runtime but already for the larger Last.fm dataset the runtime of FolkRank gets unfeasible any more for real-time predictions. In contrast to this RTF only depends on the factorization dimensions and not on the size of the dataset.

4.6 Tag entropy over time

Research on collaborative tagging processes has found that the position of a tag correlates with its expressiveness [39]. Tag entropy is a measurement of specificity where more general tags should have higher entropies because they might appear in different topics, whereas seldom tags are often more specific to a topic, thus have lower entropies. The entropy of a tag is defined as:

$$H(T) = -\sum_{t \in T} p(t) \log_2 p(t) \tag{5}$$

Here, T is the set of tags in the profile of learner, p(t) is the probability that the tag t was utilized by learner and $\log_2 p(t)$ is called self-information. Using base 2 for the computation of the logarithm allows for measuring self-information as well as entropy in bits. Figure 12. shows the strong correlation between the position and the informativeness of a tag.

It appears that learners tend to assign common tags at the beginning of the tagging process and more specific tags later. There exist at least three potential explanations for this effect [33]:

1. The affinity to label from general to specific could be a universal behavioural pattern of humans that persists in other domains.
2. The effect could also consequence from users' intention to classify new content into a set of relatively constant categories. Adding frequent category tags in the beginning and content specific tags later would result in an increase in entropy as the one observed in Fig. 12.
3. Finally, the perceived association between tag position and entropy could be initiated by the tag recommendation functionality.

4.7 Semantic analysis of tags

A semantic analysis of tags was performed to better understand different utilization of tags. Tags were classified according to [40] that is also based on the categories of [25], which are:

1. **Factual tags** - tags may be used to identify the topic of an object using nouns and proper nouns (e.g. operators, loop, arrays) or to classify the type of object (e.g. tutorial, task, example, basic info, explanation, definition),
2. **Subjective tags** - tags may be used to denote the qualities and characteristics of the item (e.g. useful, interesting, difficult, easy, understandable, blurry) and
3. **Personal tags** - item ownership, self-reference, tasks organization - a subset of tags often used by individuals to organize their own learning objects. Much like self-referencing tags, some tags are used by individuals for task organization (e.g. to read, to practice, to print).

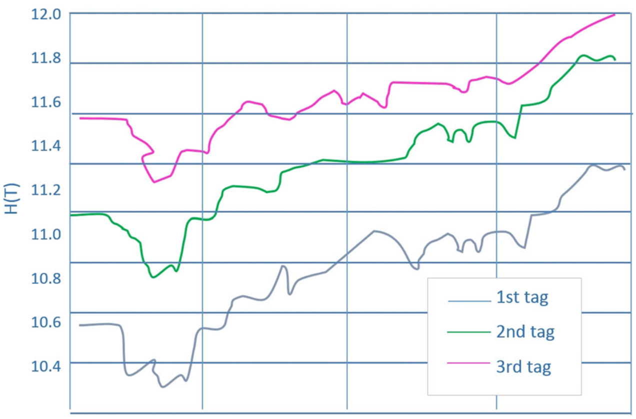


Fig. 12 Tag entropy H(T) over time

When we analyzed how these tags were used and re-used among learners, we found the vast majority of the tags were of the personal (44% of tags) and subjective type (40% of tags). The rest of the tags (16% of tags) were factual in their nature and could be used to identify the topic of a learning object. The obtained distribution indicates the fact that learners adapt learning objects themselves and organize them for easily managing. In a MovieLens study [40], for comparison, the distribution was 63% factual, 29% subjective, 3% personal and 5% other.

4.8 Expert validity survey

In order to systematically verify the relationship between learning comprehension and learner data tagging, and in order to help teachers evaluate learner knowledge, an experimental test was created with the expert tag set. The collection of learner tags was compared with the tags given independently by 4 experts in the field. The expert tag set was comprised of 165 tags of which 100 were different.

The collection of learner tags was compared with the tags given independently by 4 experts in the field. The expert tag set was comprised of 165 tags of which 100 were different.

Within expert tag set, we elaborated two research questions:

1. Which learning objects can be found by a simulated query with the expert tags on the complete set of learner tags and which relevance (number of matching tags) does it have? With respect to this question, we found that the ratio of matches was in average 45% of the expert tags also assigned to a learning object by the learners.
2. How many keywords assigned as tags are already present as text in the LO? This question addresses if the tags given to the learning items stay close to the original item. The results were that experts tend to tag more abstractly and conceptually than learners. According to [40] categories, the distribution was 73% factual, 16% subjective, 4% personal and 7% other.

Given that 55% of the expert tags were tags not within the body of tags used by learners, we question the benefits of providing these tags to learners at all. The lack of expert time and willingness to fill in metadata has been cited [41] as a significant hurdle to deploying learning objects. If expert tags provide limited value to learners, it may be more appropriate to bootstrap data sets with automatic tagging features and reduce the load on those who are creating content. We note the potential pedagogical benefits of collaborative tagging as suggested by [42]: that the tags themselves represent the expertise of the users. This proposes that at a collaborative level, a tag set can be observed as the course is being given by the experts to improvement an insight into the topics and concepts that learners are filtering from the online material.

Beyond the issue of expert time is the issue of control in the classroom. Unlike the open web, where individual success is evaluated by the individual, success in e-learning systems is typically dictated through a series of educator prepared exams. It has observed that educators are hesitant to change their teaching to adopt new methods in the classroom (virtual or otherwise), because of a loss of control. By engaging educators actively in the process of creating tags, it may reduce their fears of these new technologies. However, our results showed only 45% of the expert tags were represented in the tags of the learners. Also [43] suggested that unlike open a web system, the educator in the classroom is not merely a peer, and their tags may be more relevant to the examinations, which may be useful to learners. Therefore, end-use of tags in an educational context is of significant interest to us.

4.9 Discussion

The experiments were performed to evaluate the performance of the system from the view-points of both teachers and learners. The results of these experiments demonstrated the potential of collaborative tagging as a mean for gathering expertise from the instructional designers. At a collaborative level, a tag set can be observed as the course is being given by the experts to improve an insight into the topics and concepts that learners are filtering from the online material. The general opinion of experts has been very positive. They have demonstrated a high degree of motivation and have especially liked the novelty of using learners' data to improve e-learning courses, to be able to apply modifications to courses directly from the system and having the possibility of working and sharing information with other teachers and educational experts.

We studied how many LO were tagged on average by each learner in the system and found that even 65% learners perform high activity, tagging between 50 and 72 LOs. We examined tag characteristics of learners in Protus in order to comprehend the individualities of learner tags and learner's tagging behavior.

It has been noticed: the frequency at which a tag is selected tends to become fixed if the number of learners increased enough and a tagged object receives enough tags. This concept of convergence or stabilization has important consequences for the collective usefulness of individual tagging behavior. Similarly, this stabilization can perform for the period of sharing knowledge, as well as when learners repeat the tag selection of other learners.

We analyzed the prediction quality of graph-based approaches, Adapted PageRank and FolkRank, and tensor based approaches, HOSVD and RTF in our specific educational environment. FolkRank algorithm is constructed on PageRank and proved as significantly better tag recommendation technique than CF, because FolkRank has ability to utilize the information appropriate to the particular learner analyzing inputs from other learners passing through the underlying hypergraph.

Regarding the prediction quality of CF, Adapted PageRank and FolkRank it could be seen that FolkRank outperforms the other two methods. FolkRank algorithm also permits mode converting with no modification in the algorithm. FolkRank is robust in the terms of online updates since it doesn't need to be reapplied every time a new learner, item or tag is introduced to the system. However, FolkRank is computationally more expensive and not trivially scalable, making it more suitable for systems where real-time recommendations are not a requirement. Similarly to FolkRank, tensor factorization methods also work directly over the ternary relation of the folksonomy. Although the tensor reconstruction phase can be expensive, it could be performed offline. After the lower dimensional tensor is computed, the recommendations can be done quickly, making these algorithms appropriate for real-time recommendations.

The property about mode changing, which can only be completed by minimizing the same error function, is examined as potential drawback of tensor factorization methods. For example, HOSVD algorithm uses the reconstructed tensor for multi-mode recommendations with simple mode changing. However, although it could be the case that the problem is wrong solved: HOSVD reduces a least-square error function while social tagging RS is more associated to ranking. We selected only the best representatives of the considered techniques and concluded that the best results can be achieved with RTF followed by FolkRank then HOSVD.

5 Conclusions and future directions

Over the last two decades, recommender systems have made significant progress. Numerous approaches, such as collaborative, hybrid and content-based methods were applied along with several "industrial-strength" systems. Regardless of these improvements, the contemporary recommender systems still require additional enhancements in order to improve recommendation methods within a range of applications. The growing popularity of the collaborative tagging systems, presented in this paper, encourages the practice of tags and could provide enhancement to recommender systems in e-learning. Tags can help user in a process of organization private collections. In addition, a tag can be regarded as a personal opinion of a user, whereas tagging includes implicit voting or rating of the items. Therefore, it is expected that the tagging information can be useful for recommendations generation.

Collaborative tagging can be considered as important means for classification of dynamic content for sharing or searching. In this paper, we have studied the prospective of collaborative tagging systems, specifying analysis of personalized and subjective user preferences, classification of learning content for applying collaborative tagging techniques into Java Tutoring system. The action of carefully choosing collaborative tagging techniques can encourage growing motivation in learning process and better understanding of the learning content.

The most significant part of our research focuses on suitable selection of collaborative tagging techniques which could lead to improved and personalized recommendations, that are in accordance with the learner's interests, his/her learning style, demographic characteristics and previously acquired knowledge. Learners' motivation, as an important concept in learning process, can be enhanced with recommendation for learning, because the system can intelligently recognize how well the learner handles the matter to get instructions what to read next.

When we were preparing recommendations by applying selected techniques in the field of e-learning we had to consider the semantics of tagging, which varies according to different applications. A semantic analysis of tags was performed to better understand different utilization of tags. When we analyzed how these tags were used and re-used among learners, we found the clear majority of the tags were of the personal and subjective type. The obtained distribution is different from other well-known data sets, indicates the fact that learners adapt learning objects themselves and organize them for easier managing. Consequently, intelligent data extraction from the tag data set in terms of knowledge needs to be different and innovative. A system should generate recommendations list like a professor in the classroom, during the face-to-face learning process.

Our future studies could focus more specifically on measuring the impact of prior learner experience with computers and the Internet and interest in the knowledge domain on the effect of creating tags. Additionally, future studies could investigate whether are there more factors which also have an influence on the choice of tags. Possible factor candidates could be a mood or stress level of learners.

References

1. Ley T, Seitlinger P (2015) Dynamics of human categorization in a collaborative tagging system: how social processes of semantic stabilization shape individual sensemaking. Comput Human Behav 51:140–151

2. Hotho A, Jäschke R, Schmitz C, Stumme G (2006) Information retrieval in folksonomies: search and ranking. In: European Semantic Web Conference, pp 411–426
3. Gruber T (2007) Ontology of folksonomy: a mash-up of apples and oranges. Int J Semant Web Inf Syst (IJSWIS) 3(1):1–11
4. Cao Y, Kovachev D, Klamma R, Jarke M, Lau RW (2015) Tagging diversity in personal learning environments. J Comput Educ 2(1):93–121
5. Milicevic AK, Nanopoulos A, Ivanovic M (2010) Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. Artif Intell Rev 33(3):187–209
6. Paiva JC, Leal JP, Queirós R (2016) Gamification of learning activities with the Odin service. Computer Science and Information Systems 13(3):25–25
7. Salehi M (2013) Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation. Data Knowl Eng 87:130–145
8. Zhenzhen X, Jiang H, Kong X, Kang J, Wang W, Xia F (2016) Cross-domain item recommendation based on user similarity. Comput Sci Inf Syst 13(2):359–373
9. Möditscher F (2010) Towards a recommender strategy for personal learning environments. Procedia Comput Sci 1(2):2775–2782
10. Anjorin M, Rensing C, Steinmetz R (2011) Towards ranking in folksonomies for personalized recommender systems in e-learning. In: SPIM, pp 22–25
11. Cernea DA, Del Moral E, Gayo JEL (2008) SOAF: Semantic indexing system based on collaborative tagging. Interdisciplinary Journal of E-Learning and Learning Objects 4(4):137–150
12. Chen J-M, Chen M-C, Sun YS (2014) A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension. Comput Educ 70:256–268
13. Lavoué É (2012) Towards social learning games. In: International conference on web-based learning. Springer, pp 170–179
14. Verpoorten D, Chatti M, Westera W, Specht M (2010) Personal learning environment on a procrustean bed–using PLEM in a secondary-school lesson. In: Proceedings of the London international conference on education (LICE-2010). LICE, pp 197–203
15. Kurilovas E, Serikoviene S, Vuorikari R (2014) Expert centred vs learner centred approach for evaluating quality and reusability of learning objects. Comput Human Behav 30:526–534
16. Zervas P, Sampson DG (2014) The effect of users' tagging motivation on the enlargement of digital educational resources metadata. Comput Human Behav 32:292–300
17. Chen W, Persen R (2012) Recommending collaboratively generated knowledge. Comput Sci Inf Syst 9(2):871–892
18. Santos OC, Boticario JG, Baldiris S, Moreno G, Huerva D, Fabregat R (2008) Recommender systems for lifelong learning inclusive scenarios. In: Workshop on recommender systems, p 45
19. Manouselis N, Drachsler H, Vuorikari R, Hummel H, Koper R (2011) Recommender systems in technology enhanced learning. In: Recommender systems handbook. Springer, pp 387–415
20. Bateman S, Brooks C, Mccalla G, Brusilovsky P (2007) Applying collaborative tagging to e-learning. In: Proceedings of the 16th international world wide web conference (WWW2007)
21. Dahl D, Vossen G (2008) Evolution of learning folksonomies: social tagging in e-learning repositories. Int J Technol Enhanc Learn 1(1–2):35–46
22. Doush IA, Alkhateeb F, Maghayreh EA, Alsmadi I, Samarah S (2012) Annotations, Collaborative Tagging, and Searching Mathematics in E-Learning. arXiv:12111780
23. Brin S, Page L (2012) Reprint of: the anatomy of a large-scale hypertextual web search engine. Comput Netw 56(18):3825–3833
24. De Lathauwer L, De Moor B, Vandewalle J (2000) A multilinear singular value decomposition. SIAM J Matrix Anal Appl 21(4):1253–1278
25. Golder SA, Huberman BA (2006) Usage patterns of collaborative tagging systems. J Inf Sci 32(2):198–208
26. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst (TOIS) 22(1):5–53
27. Marinho LB, Nanopoulos A, Schmidt-Thieme L, Jäschke R, Hotho A, Stumme G, Symeonidis P (2011) Social tagging recommender systems. In: Recommender systems handbook. Springer, pp 615–644
28. Rendle S, Balby Marinho L, Nanopoulos A, Schmidt-Thieme L (2009) Learning optimal ranking with tensor factorization for tag recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 727–736
29. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating "word of mouth". In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., pp 210–217
30. Symeonidis P, Nanopoulos A, Manolopoulos Y (2010) A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. IEEE Trans Knowl Data Eng 22(2):179–192
31. Tan H, Guo J, Li Y (2008) E-learning recommendation system. In: 2008 international conference on science computer and software engineering. IEEE, pp 430-433
32. Vesin B, Ivanovic M, Klasnja-Milicevic A, Budimac Z (2011) Rule-based reasoning for altering pattern navigation in programming tutoring system. Springer
33. Klasnja-Milicevic A, Vesin B, Ivanovic M, Budimac Z, Jain LC (2016) E-learning systems: intelligent techniques for personalization
34. Ivanović M, Mitrović D, Budimac Z, Vesin B, Jerinić L (2012) Different roles of agents in personalized programming learning environment. Springer, pp 161–170
35. Felder RM, Soloman B (1996) Index of learning styles questionnaire
36. Soloman BA, Felder RM (2005) Index of learning styles questionnaire. NC State University Available online at: http://www.engr.ncsu.edu/learningstyles/ilsweb.html (last visited on 1405 2010)
37. Jäschke R, Marinho L, Hotho A, Schmidt-Thieme L, Stumme G (2007) Tag recommendations in folksonomies. In: European conference on principles of data mining and knowledge discovery. Springer, pp 506–514
38. Henning V, Reichelt J Mendeley - A Last.fm for research? In, 2008. pp 327–328. https://doi.org/10.1109/eScience.2008.128
39. Klašnja-Milićević A, Vesin B, Ivanović M, Budimac Z, Jain LC (2017) Folksonomy and tag-based recommender systems in e-learning environments. In: Springer International Publishing, pp 77–112
40. Sen S, Lam SK, Rashid AM, Cosley D, Frankowski D, Osterhouse J, Harper FM, Riedl J (2006) Tagging, communities, vocabulary, evolution. In: Proceedings of the 2006 20th anniversary conference on computer supported cooperative work. ACM, pp 181–190
41. Ims (2006) IMS meta-data best practice guide for IEEE 1484.12.1-2002 standard for learning object metadata
42. Jones AK, Hoare RR, Dontharaju SR, Tung S, Sprang R, Fazekas J, Cain JT, Mickle MH (2006) An automated, reconfigurable, low-power RFID tag. In: Proceedings of the 43rd annual design automation conference. ACM, pp 131–136
43. Halpin H, Robu V, Shepherd H (2007) The complex dynamics of collaborative tagging. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp 211–220

Aleksandra Klašnja-Milićević is an assistant professor at Faculty of Sciences, University of Novi Sad, Serbia. She joined the graduate program in Computer Sciences at Faculty of Sciences, Department of Mathematics and Informatics, University of Novi Sad in 2003, where she received her M.Sc. degree (2007) and PhD degree (2013). Her research interests include e-learning and personalization, Intelligent Tutoring Systems, information retrieval, Internet technologies and recommender systems. She actively participates in several international projects. She has also served as Program Committee member of several international conferences. She co-authored one university textbook and one monograph. She has published over 30 scientific papers in proceedings of international conferences and journals.

Mirjana Ivanović holds the position of Full Professor at Faculty of Sciences, University of Novi Sad. She is a member of the University Council for Informatics. She is author or co-author of 14 textbooks, several monographs and more than 350 research papers on multi-agent systems, e-learning, and intelligent techniques, most of which are published in international journals and conferences. She is/was a member of Program Committees of more than 230 international conferences, participant of numerous international research projects and principal investigator of more than 15 projects. Mirjana Ivanovic delivered several keynote speeches at international conferences, and visited numerous academic institutions all over the world as visiting researcher and teacher. Currently she is Editor-in-Chief of the Computer Science and Information Systems journal.

Boban Vesin is a Postdoctoral fellow at the Norwegian University of Science and Technology, Trondheim, Norway. He received a M.Sc. (in Computer science) from the University of Novi Sad in 2007, and Ph.D. (Computer science) from the same University in 2014. Currently, his research is focused on engineering of e-learning systems and use of learning analytics for assisting teachers to understand and improve teaching processes. Boban Vesin has developed and experimented with learning environments since 2004, and since that time has published over 35 scientific papers in proceedings of journals and international conferences in the field of software engineering, e-learning, and recommender systems.

Zoran Budimac holds the position of full professor at Faculty of Sciences, University of Novi Sad, Serbia. Currently, he is head of Computing Laboratory and Chair of Computer Science. His fields of research interests involve: Software Engineering, Programming Languages and Tools, Educational Technologies, Agents and WF-MS, Case-Based Reasoning. He was principal investigator of more than 20 projects. He is author of 13 textbooks and more than 250 research papers most of which are published in international journals and international conferences. He is/was a member of Program Committees of more than 100 international Conferences and is member of Editorial Board of Computer Science and Information Systems Journal.