Chapter 6
Recommender Systems in E-Learning Environments

Abstract Recommender system can be defined as a platform for providing recommendations to users based on their personal likes and dislikes. These systems use a specific type of information filtering technique that attempt to recommend information items (movies, music, books, news, Web pages, learning objects, and so on.) to the user. Recommender systems strongly depend on the context or domain they operate in, and it is often not possible to take a recommendation strategy from one context and transfer it to another context or domain. Personalized recommendation can help learners to overcome the information overload problem, by recommending learning resources according to learners' habits and level of knowledge. The first challenge for designing a recommender component for e-learning systems is to define the learners and the purpose of the specific context or domain in a proper way. This chapter provides an overview of techniques for recommender systems, folksonomy and tag-based recommendation to assist the reader in understanding the material which follows in subsequent chapters.

Recomender systems (RS) strongly depend on the context or domain they operate in, and it is often not possible to take a recommendation strategy (Drachsler et al. 2009) from one context and transfer it to another context or domain. The first challenge for designing a RS is to define the learners and the purpose of the specific context or domain in a proper way (McNee et al. 2006). Learning process includes three components: learners, teachers/instructors, and learning materials. From a teacher's point of view, teaching is an activity to deliver information and skill to learners with some goals to be achieved. From the learners' point of view, learning is an activity to acquire information from teacher to achieve goals set by the teacher. Learners, with their prior knowledge, acquire new information from the teacher. Here, social constructivism paradigm can help learners learn collaborative and sharing knowledge with each other. Basically, knowledge which is needed to be achieved according to the course, mainly does not influence how many Learning

A. Klašnja-Milićević et al., *E-Learning Systems*,
Intelligent Systems Reference Library 112, DOI 10.1007/978-3-319-41163-7_6

Objects (LOs) the learners have read, but how relevant are LOs that have retrieved and learned. A learner who has high prior knowledge according to the course is different from other learners who have low prior knowledge.

In a virtual classroom, teachers provide resources such as text, multimedia and simulations, and moderate and animate discussions. Remote learners are encouraged to peruse the resources and participate in activities. However, it is very difficult and time consuming for educators to thoroughly track and assess all the activities performed by all learners on all tools. Moreover, it is hard to evaluate the structure of the course content and its effectiveness on the learning process. Resource providers do their best to structure the content assuming its efficacy (Zaïane and Luo 2001). When instructors put together an on-line course, they may compile interactive course notes, simulations, demos, exercises, quizzes, asynchronous forums, chat tools, Web resources, etc. This amalgam of on-line hyperlinked material could form a complex structure that is difficult to navigate. Hence, personalization features are needed, which adaptively facilitate learner in monitoring their learning progress and provide any resources or learning material, that's suitable to what they need.

6.1 Recommendations and Recommender Systems

The information on the Web is increasing far more quickly than people can cope with. Learners are forced to review a number of choices before they discover what they need. This is often time consuming and frustrating. Given today's fast paced lifestyle, a slow and careful search for the elusive item of choice is surely not a sustainable option. People would rather look at items that are customized to their interests and preferences. Personalized recommendation (Resnick and Varian 1997) can help people to overcome the information overload problem, by recommending items according to users' interests.

Recommender systems can be defined as a platform for providing recommendations to users based on their personal likes and dislikes. These systems use a specific type of information filtering (IF) technique that attempt to recommend information items (movies, music, books, news, Web pages, learning objects, etc.) to the user (Ricci et al. 2011). To handle this the user's profile is compared to some reference characteristics. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach) (Adomavicius and Tuzhilin 2005a).

Typically, RSs apply personalization techniques, considering that different users have different preferences and different information needs (Herlocker et al. 2004). In order to generate personalized recommendations that are tailored to the user's specific needs, recommender systems must collect personal preference information, e.g., the user's history of purchase, click-stream data, demographic information, and so forth. Traditionally, expressions of preference of users for products are generally called ratings. Two different types of ratings are distinguished.

1. *Explicit ratings.* Users are required to explicitly specify their preference for any particular item, usually by indicating their extent of appreciation on 5 or 7-point Likert scales (Ziegler 2013).
2. *Implicit ratings.* Explicit ratings require additional efforts on users. Consequently, users often tend to avoid the burden of explicitly stating their preferences and either leave the system or rely upon "free-riding" (Avery and Zeckhauser 1997). Alternatively, gathering preference information from observations of user behaviour is less intrusive (David 1997).

A ratings database consists of pairs of users and items rated by them, along with additional information such as timestamps, etc. Given such a ratings database and a set of preliminary ratings by a new user, the basic requirement of a recommendation system is to recommend the largest and the most significant set of recommendations to the new user (Miller 2003). The largest set of recommendations refers to the maximum number of items in the item database that could be recommended to the user. The significant items are the ones that the new user will be more likely to rate in the future.

Based on the nature of reference characteristics, two broad categories of information filtering for computing recommendations have emerged: content-based filtering, and collaborative filtering (Goldberg et al. 1992).

Content-based recommender systems recommend items "by comparing representations of content contained in an item to representations of content that interests the user" (Malone et al. 1987). In most cases, a keyword profile is created. Apparently, this works well in text domains, but not in domains where there is not much content associated with the items or where a computer cannot easily analyse this content. Relying on rich descriptions, content-based recommender systems need significant knowledge engineering efforts to create substantial metadata for the items. Content-based systems "form profiles for each user independently" (Basilico and Hofmann 2004). Even if two items were in two neighbour categories that people normally like both of them, the item from category A would never be recommended to the user if (s)he only rated items from category B. This problem is often addressed by introducing some unpredictability. Also, the user has to rate a suitable number of items before a content-based recommender system can really comprehend the user's preferences and present the user with trustworthy recommendations. Therefore, a new user, having very few ratings, would not be able to get accurate recommendations.

Collaborative filtering systems compute profile similarity between the target user and the other users "by comparing users' opinions of items" (Balabanović and Shoham 1997). Profile similarity is usually computed by comparing rating-vectors with various distance metrics, e.g. Pearson correlation or cosine similarity. They supply the user with the items (s)he will most likely be interested in, either one single item or a “ranked list of items”—usually referred as *top-N-item* (Cosley et al. 2002; McLaughlin and Herlocker 2004). In contrast to content-based systems, recommender systems based on collaborative filtering can provide the user with unexpected but fitting recommendations that do not have anything in common with afore rated items. Collaborative filtering is a very successful methodology in almost every domain—especially “where multi-value ratings are available” (McLaughlin and Herlocker 2004). However, they suffer from two key problems: sparsity and first-rater problem. As most users only rate a small portion of all items, it is highly difficult to find users with “significantly similar ratings.” Furthermore an item cannot be recommended before one user has rated it. This can be the case if the item has newly been introduced to the system (Melville et al. 2002).

A number recommendation systems use a *hybrid approach* by combining collaborative and content-based methods, which helps to avoid certain limitations of content and collaborative-based systems (Balabanović and Shoham 1997; Basu et al. 1998; Claypool et al. 1999; Pennock and Horvitz 1999). Different ways to combine collaborative and content-based methods into a hybrid recommender system can be classified as follows (Adomavicius and Tuzhilin 2005a):

1. implementing collaborative and content-based methods separately and combining their predictions,
2. incorporating some content-based characteristics into a collaborative approach,
3. incorporating some collaborative characteristics into a content-based approach, and
4. constructing a general unifying model that incorporates both content-based and collaborative characteristics.

According to (Breese et al. 1998), algorithms for collaborative recommendations can be grouped into two general classes: memory-based (or heuristic-based) and model-based.

Memory-based algorithms (Breese et al. 1998; Resnick et al. 1994; Shardanand and Maes 1995) essentially are heuristics that utilize the entire database of user preferences when computing recommendations. These algorithms tend to be simple to implement and require little training cost. They can also easily take new preference data into account. However, their online performance tends to be slow as the size of the user and item sets grow, which makes these algorithms as stated in the literature unsuitable in large systems. One workaround is to only consider a subset of the preference data in the calculation, but doing this can reduce both recommendation quality and the number of items that can be recommended due to data being omitted from the calculation. Another solution is to perform as much of the computation as possible in an offline setting. However, this may make it difficult to add new users to the system on a real-time basis, which is a basic necessity of most online systems. Furthermore, the storage requirements for the pre-computed data could be high.

Model-based algorithms (Billsus and Pazzani 1998; Goldberg et al. 2001; Hofmann 2003) use the collection of ratings to learn a model, which is then used to make rating predictions. Often, the model building process is time-consuming and is only used periodically. The model is compact and can generate recommendations very quickly. The disadvantage of model-based algorithms is adding new users, items, or preferences, which can be the same as re-computing the entire model.

The most important difference between collaborative model-based techniques and heuristic-based approaches is that the model-based techniques calculate utility predictions based not on some ad hoc heuristic rules, but, rather, based on a model learned from the underlying data using statistical and machine learning techniques (Adomavicius and Tuzhilin 2005a). A method combining memory-based and model-based approaches was proposed in Pennock et al. (2000). It was empirically confirmed that the use of this approach can afford better recommendations than pure memory-based and model-based collaborative approaches.

Over the past several years there has been much research done on recommendation technologies which use a variety of statistical, machine learning, information retrieval, and other techniques that have significantly advanced early recommender systems, collaborative and content-based heuristics. As was discussed above, recommender systems can be classified as (1) content-based, collaborative, or hybrid (based on the recommendation approach used), and (2) heuristic-based or model-based (based on the types of recommendation techniques used for the rating estimation). These two orthogonal dimensions are used to classify the recommender systems research in the 2×3 matrix, presented in Table 6.1 (Adomavicius and Tuzhilin 2005a).

The recommendation techniques explained in this chapter have performed well in several applications, including the ones for recommending books, CDs, news articles or movies (Marlin 2003; Rosset et al. 2002) and some of these methods are used in the "industrial-strength" recommender systems, such as the ones developed at Amazon,[1] MovieLens,[2] and Last.fm.[3] However, both collaborative and content-based methods have certain limitations. Recommender systems can be extended in several ways that include improving the understanding of users and items, incorporating the contextual information into the recommendation process, sustaining multicriteria ratings, and providing more flexible and less disturbing types of recommendations (Adomavicius and Tuzhilin 2005a).

[1] http://www.amazon.com.

[2] http://www.movielens.umn.edu.

[3] http://www.last.fm.

Table 6.1 Classification of RS research (Adomavicius and Tuzhilin 2005b)

Recommendation Approach	Recommendation technique	
	Heuristic based	Model based
Content - based	Commonly used techniques:	Commonly used techniques:
	TF-IDF (information retrieval)	Bayesian classifiers
	Clustering	Clustering
	Representative research examples:	Decision trees
	Lang 1995	Representative research examples:
	Balabanović, Shoham 1997	Pazzani and Billsus 1997
	Pazzani and Billsus 1997	Mooney et al. 1998
		Mooney and Roy 1999
		Billsus and Pazzani 1998, 1999
		Zhang et al. 2002
Collaborative	Commonly used techniques:	Commonly used techniques:
	Nearest neighbour	Bayesian networks
	(cosine, correlation)	Clustering
	Clustering	Artificial neural networks
	Graph theory	Linear regression
	Representative research examples:	Probabilistic models
	Resnick et al. 1994	Representative research examples:
	Hill et al. 1994	Billsus and Pazzani 1998
	Shardannand and Maes 1995	Breese et al. 1998
	Breese et al. 1998	Goldberg et al. 2001
	Nakamura and Abe 1998	Ungar and Foster 1998
	Aggarwal et al. 1999	Chien and George 1999
	Delgado and Ishii 1999	Getoor and Sahami 1999
	Pennock and Horwitz 1999	Pennock and Horwitz 1999
	Sarwar et al. 2001	Pavlov and Pennock 2002
		Shani et al. 2002
		Hofmman 2003, 2004
Hybrid	Combining content—based and collaborative components by:	Combining content—based and collaborative components by:
	Linear combination of predicted ratings	Incorporating one component as a part of the model for the other
	Various voting schemes	Building one unifying model
	Incorporating one component as a part of the heuristic for the other	Representative research examples:
		Soboroff and Nicholas 1999
	Representative research examples:	Basu et al. 1998
	Balabanović and Shoham 1997	Condiff et al. 1999
	Pazzani 1999	Popescul et al. 2007
	Billsus and Pazzani 1998	Schein et al. 2002
	Claypool et al. 1999	Ansari et al. 2000
	Good et al. 1999	
	Train and Cohen 2000	

6.2 The Most Important Requirements and Challenges for Designing a Recommender System in E-Learning Environments

A RS in e-learning environments utilizes information about learners and learning activities (LA) and recommend items such as papers, Web pages, courses, lessons and other learning resources which meet the pedagogical characteristics and interests of learners (Drachsler et al. 2008). Such a RS could provide recommendations to online learning materials or shortcuts. Those recommendations are based on previous learners' activities or on the learning styles of the learners that are discovered from their navigation patterns. To design an effective RS in e-learning environments, it is important to understand specific learners' characteristics (Drachsler et al. 2008; García et al. 2009):

1. Learner's goal or learner's task is a feature related with the context of a learner's activities in educational system rather than with the learner as an individual. Depending on the kind of system, it can be the goal of the activity (in application systems), a search goal (in information retrieval systems), and a problem solving or learning goal (in educational (e-learning) systems). In all of these cases the goal is an answer to the question "Why is the learner using the system and what does the learner actually want to achieve?" Learner's goal is the most unpredictable learner feature: almost always it changes from session to session and often can change several times within one session. In some systems it is reasonable to distinguish local or low-level goals which can change quite often and general or high level goals and tasks which are more stable. For example, in educational systems the knowledge acquisition is a high-level goal, while the problem-solving goal is a low-level goal which changes from one educational problem to another several times within a session.
2. Prior learner's knowledge of the subject represents one of the most important feature of the learner for adaptive educational systems. Almost all adaptive presentation techniques rely on the learner's knowledge as a source of adaptation. Related to the first point, we also need to know if the learners already have any prior knowledge about what they want to learn. The proficiency level of the learning activity should fit the proficiency level of the learner (prior knowledge). The learners may want to reach the learning goals on specific competence levels, like beginner, advanced or expert levels. Learner's knowledge is a variable for a particular learner. This means that an adaptive educational system which relies on learner's knowledge has to recognize the changes in the learner's knowledge state and update the learner model accordingly.
3. Background and experience are two features of the learner which are similar to user's knowledge of the subject but functionally differ from it. Learner's background can be presented as all the information related to the learner's previous experience outside the subject of the educational system, which is relevant enough to be considered. This includes the learner's profession,

experience of activities in related areas, as well as the learner's point of view and perspective. According to learner's experience it could be identified how familiar is the learner with the structure of the similar learning environments and how easy can the learner navigate in it. This is not the same as learner's knowledge of the subject (Vassileva 1998). Sometimes, the learner who is generally quite familiar with the subject itself is not familiar at all with the structure of educational system. Vice versa, the learner can be quite familiar with the structure of the educational system without deep knowledge of the subject. One more reason to differentiate experience from knowledge level is the existence of an adaptive navigation technique (Pérez et al. 1995; Vassileva 1998) which relies on this feature of the learner.

4. Learner preferences. For different reasons the learner can prefer some links more than others and some parts of a page more than others. These preferences can be absolute or relative, i.e., dependent from the current link, goal and current context in general. Learner's preferences differ from other learner model components in several aspects. Unlike other components, the preferences cannot be deduced by the system. The learner has to inform the system directly or indirectly (by a simple feedback) about such preferences. It looks more close to adaptability then to adaptivity.
5. Learner group models accumulate preferences of a specific group of learners (such as a research laboratory). A group model is a nice starting model for a new member of the group. Group models are important also for collaborative activities. It is very hard to collaborate when collaborators use individual learner models and thus have different adapted views on the same subject.
6. Rated learning activities (LAs). The aggregated ratings of the learning activities as awarded by other learners can provide valuable information (the rated learning activities). Learners with the same learning goal or similar study time per week could benefit from the ratings received from more advanced learners. Nearly all potential learning activities are unknown to the learners. Learners are (by definition) not able to rate learning activities in advance, because if they already knew them, they would no longer be potential learning activities. Moreover, the learners will at least have to read through a learning activity before they are able to rate it. Many people are able to rate movies because they have heard or read about it, or have already seen the movie. In the domain of learning, however, it is unlikely that a learner will already be familiar with certain learning activities. Consequently, it is less of a problem for 'movie lovers' to rate movies in advance to specify a profile than it is for learners to rate learning activities in advance. Requiring learners to rate an initial set of learning activities, as in movielens.org, does not, therefore, seem feasible. Other mechanisms to specify a learner profile have to be devised. Even for the learners with the same interests, we may need to recommend different learning activities, depending on the individual proficiency levels, learning goals and context. For instance, the learners with no prior knowledge in a specific domain should be advised to study basic learning activities first, while more advanced learners should be advised to continue with more specific learning activities.

7. Learning paths. Beginning learners could benefit from historical information about the successful study behaviour of the more advanced learners in the same learning network, in the same learning paths. From the learning activities which are frequently positively rated and their sequence, the most popular learning paths will emerge. The most successful learning paths with regard to efficiency and effectiveness could be recommended.
8. Learning strategies. RS in e-learning would benefit if we apply the learning strategies derived from educational psychology research (Koper and Olivier 2004). Such strategies could use pedagogical rules as guiding principles for recommendation, like 'go from simple to more complex tasks' or 'gradually decrease the amount of contact and direct guidance'. This entails taking into account the metadata about specific learning activities, but not the actual design of the specific learning activities themselves.

E-learning systems should be able to recognize and exploit these learners' characteristics serve as guidelines for framework design and platform implementation for a good RS for e-learning (Angehrn et al. 2001; Savidis et al. 2007; Zaïane and Luo 2001).

- *A good RS should be highly personalized.* Relevant learning materials should be chosen and presented to learners or researchers based on learner's learning style, interests, preferences, current activities, etc.
- *A good RS should recommend materials at the appropriate time and location*. A good RS should deliver relevant learning materials to learner at the most appropriate time and locations to facilitate learners' acquisition of knowledge and skills.
- *A good RS should support non-disruptive view of experience*. Non-disruptive means that learners have the option to either follow or discount relevant materials based on their learning needs.
- *A good RS should be socially situated*. A good RS should be able to recognize and exploit the learners' social networks, role models, levels of trust and influence, etc. RS should also help the learners to recognize their knowledge acquisition process in the context of the group.
- *A good RS should include the adoption phase*. A good RS should be able to monitor, understand and model the different phases of adoption of the knowledge by the learner. In particular it includes the phases in which the new concepts are experimented with, evaluated, internalized and finally applied.
- *A good RS should support the continuous learning process*. A good RS should support just-in-time learning, by better analysing their current and future activities. Also it should provide motivational support and stimulation.
- *A good RS should provide high level of interactivity*. A good RS should provide very active, cognitive and diverse mode of interaction with the learner in the form of a rich choice of interaction strategies.
- *A good RS should provide appropriate course materials according to learners' learning style*. Each person learns differently and needs to develop his/her own

learning skills in his/her own way. Learners have different backgrounds, strengths and weaknesses, interests, ambitions, senses of responsibility, levels of motivation, and approaches to studying and learning. For example, different learners prefer different presentation forms: some prefer multimedia contents (simulations, presentations, graphical material and hypertext documents); while others prefer traditional Web pages (questionnaires, exercises, research studies).

6.3 Recommendation Techniques for RS in E-Learning Environments—A Survey of the State-of-the-Art

Personalized recommendation approaches are first proposed in e-commerce area for product purchase (Balabanović and Shoham 1997; Resnick and Varian 1997), which help consumers to find products they would like to purchase by creating a list of recommended products for each given consumer (Cheung et al 2003; Schafer et al. 2001). Literature review shows that there are also many researchers who have attempted to adopt recommender systems to e-learning environments. For example, (Shen and Shen 2005) described a mechanism focused on how to organize the learning materials based on domain ontology which can guide the learning resources recommendation according to learning status. A multi-attribute assessment method is proposed in Lu (2004) to justify a learner's need and deployed a fuzzy matching method to find suitable learning contents to best perform each learner need. Research paper (Luo et al. 2002) presented a method to organize components and courseware using the hierarchy and association rules of the concepts, which can recommend the relative contents to learners and also can help them to control the learning schedule. However, most of these methods missing one important issue in e-learning RS, that is, the natural learning behaviour is not lonely but interactive which relying on friends, classmates, lecturers, and other sources to make the choices for learning.

Designers and instructors, when devising the on-line structure of the course and course material, have a navigation pattern in mind and assume all on-line learners would follow a consistent path; the path put out in the design and materialized by some hyperlinks. Learners, however could follow different paths generating a variety of sequences of learning activities. Often some sequences are not the optimum sequences, and probably not the sequence intended by the designer. Instructors are in desperate need for non-intrusive and automatic ways to get objective feedback from learners in order to better follow the learning process and appraise the on-line course structure effectiveness. On the learner's side, it would be very useful if the system could automatically guide the learner's activities and intelligently recommend on-line activities or resources that would favour and improve the learning. The automatic recommendation could be based on the teacher's intended sequence of navigation in the course material, or, more interestingly, based on navigation patterns of other successful learners. For example, during the learning process, a learner read a useful material, summarized what (s)he has learned or got the answer of a typical question, some learners with similar learning status will likely need these resources.

E-learning system uses different recommendation techniques in order to suggest online learning activities to learners, based on their preferences, knowledge and the browsing history of other learners with similar characteristics. RSs assist the natural process of relying on friends, classmates, lecturers, and other sources for making the choices of learning (Lu 2004). In the educational setting, these recommendation systems can be classified according to their field of application or focus (Romero et al. 2007):

1. learner-centered (Gaudioso et al 2003; Zaíane 2002), in order to suggest good learning experiences for the learners in accordance to their preferences, needs and level of knowledge; and
2. teacher-centered, with the aim of helping the teachers and/or authors of the e-learning systems to improve the functionalities or performances of these systems based on learner information (W Chen and Wasson 2003; Romero et al. 2003). Some other examples of educational applications of these systems are: obtaining more feedback about teaching; finding out more about how learners learn on the Web; evaluating learners in terms of their browsing patterns; classifying learners into groups; or restructuring the contents of the website in order to personalize the course.

Each recommendation strategy has its own strengths and weaknesses. According to set of the most important requirements for a good RS in e-learning environment, have been explored and defined in the previous section, in the remainder of this section we present a survey of the state-of-the-art in RSs for e-learning systems. We identify challenges and various limitations for each traditional recommendation method, then consider some tag-based profiling approaches for extending their capabilities.

6.3.1 Collaborative Filtering Approach

Collaborative systems track past actions of a group of learners to make a recommendation for individual members of the group (Tan et al. 2008). Based on the assumption that learners with similar past behaviours (rating, browsing, or learning path) have similar interests, a collaborative filtering system recommends learning objects the neighbours of the given learner have liked.

This approach relies on a history record of all learner interests such as can be inferred from their ratings of the items (learning objects/learning actions) on a website. Rating can be explicit (explicit ratings or customer satisfaction questionnaires) or implicit (from the studying patterns or click-stream behaviour of the learners). The proportion of actual studying hours to the total hours of the course is recorded as the implicit rating scores, and transformed to corresponding explicit rating scores, from 1 to 5. The learners' rating scores can be given in a m * n

Table 6.2 Learner's rating matrix

	O_1	…	O_k	…	O_n
I_1	$R_{1,\ 1}$	…	$R_{1,\ k}$	…	$R_{1,\ n}$
…					
I_1	$R_{j,\ 1}$	…	$R_{j,\ k}$	…	$R_{j,\ \mathrm{n}}$
…					
I_1	$\mathrm{R}_{m,\ 1}$	…	$\mathrm{R}_{m,\ k}$	…	$\mathrm{R}_{m,\ n}$

matrix, as it is shown in Table 6.2, where $L = \{I_1,I_2,\ldots,I_m\}$ is a list of m learners, $O = \{o_1,o_2,\ldots,o_n\}$ is the list of n learning objects, and $R_{j,k}$ gives the rating of object O_k, given by learner j. Also, it can be rating of object o_k given by intelligent tutoring system for learner j. There exists a distinguished learner $I_a \in L$ called the active learner for whom the task of collaborative filtering algorithm is to find learning object likeliness.

The neighbourhood formation scheme usually uses Pearson correlation or cosine similarity as a measure of proximity (Resnick et al. 1994; Shardanand and Maes 1995).

An exploratory study of a recommender system, using collaborative filtering to support (virtual) learners in a learning network, has been reported in Koper (2005). The author simulated rules for increasing/decreasing motivation and some other disturbance factors in learning networks, using the Netlogo tool. Closely related to this study is an experiment reported in Janssen et al. (2007). The authors offered to learners a similar recommendation system. The recommendations did not take personal characteristics of learners (or possible 'matching errors') into account. Another system implemented by Soonthornphisaj et al. (2006) allows all learners to collaborate their expertise in order to predict the most suitable learning materials to each learner. This smart e-learning system applies the collaborative filtering approach that has an ability to predict the most suitable documents to the learner. All learners have the chance to introduce new material by uploading the documents to the server or pointing out the Web link from the Internet and rate the currently available materials.

One of the first attempts to develop a collaborative filtering system for learning resources has been the Altered Vista (AV) system (Recker et al. 2003; Recker and Walker 2003; Walker et al. 2004). The AV system (Walker et al. 2004) uses a database in which a learner evaluations of learning resources are stored. Learners can browse the reviews of others and can get personalized learning resource recommendations from the system. AV does not aim to support learners directly by giving them feedback on their work. Instead, AV provides an indirect learning support in which suitable learning tools are recommended. The team working on AV explored several relevant issues, such as the development of non-authoritative metadata to store learner-provided evaluations (Recker and Walker 2003), the design of the system and the review scheme it uses (Walker et al. 2004), as well as results from pilot and empirical studies from using the system to recommend to the members of a community both interesting resources and people with similar tastes and beliefs. A survey-based evaluation of AV showed a predominant positive feedback, but also identified issues with the system's incentive and with regard to privacy (Walker et al. 2004).

Another system of the educational collaborative filtering applications is the Web-based PeerGrader (PG) (Gehringer 2001; Lynch et al. 2006). The purpose of this tool is to help learners improve their skills by reviewing and evaluating solutions of their fellow learners blindly. PG works in the following way:

- the learners get a task list and each learner chooses a task,
- the learners submit their solutions to the system, where they are read by another learner who then provides feedback in form of textual comments,
- the authors modify their solutions based on the comments they have received, and re-submit their modified solutions again to the system, where they will be reviewed by other learners, then the solutions' authors grade each review with respect to whether it was helpful or not.
- finally, the system calculates grades for all learner solutions.

One of PG's strengths is to provide learners with high-quality feedback also in ill-defined homework tasks that do not have clear-cut gold standard solutions (such as design problems). This kind of feedback could not be generated automatically. A disadvantage is the time required for the system to work effectively: due to the complexity of the reviewing process and the textual comments, the evaluation of a single learner answer is very time consuming. This may cause learner drop-outs and deadline problems (Lynch et al. 2006). Also, studies with PG revealed problems with getting feedback of high quality. An evaluation of subjective usefulness showed that the system was appreciated by its users (Lynch et al. 2006), yet a systematic comparison of PG scores to expert grades has not been conducted.

A newer Web-based collaborative filtering system, the Scaffolded Writing and Rewriting in the Discipline (SWoRD) system (Cho et al. 2006; Cho and Schunn 2007) addresses the problem of writing homework in the form of a long text, which cannot be reviewed in detail by a teacher for time reasons. Because of this, learners do often not receive any detailed feedback on their solutions at all. Having such feedback, it would be beneficial for learners, since they could use it to improve their future work. To address this problem, SWoRD relies on peer reviews and implements an algorithm that follows the typical journal publication and reviewing process. An evaluation showed that the participants benefitted from multi-peers' feedback more than from single-peer's or single expert's feedback (Cho and Schunn 2007).

A different approach is used by the LARGO system (Pinkwart et al. 2006), where learners create graphs of US Supreme Court oral arguments. Within LARGO, collaborative scoring is employed to assess the quality of a "decision rule" that a learner has included in his diagram. Since this assessment involves interpretation of legal argument in textual form, it cannot be automated reasonably. While the overall LARGO system has been tested in law schools and shown to help lower-aptitude learners (Pinkwart et al. 2007), empirical studies to test the educational effectiveness of the specific collaborative scoring components have not been conducted.

Rule-Applying Collaborative Filtering (RACOFI) Composer system (Anderson et al. 2003; Lemire et al. 2005; Lemire 2005) combines two recommendation approaches by integrating a collaborative filtering engine, that works with ratings that learners provide for learning resources, with an inference rule engine that is mining association rules between the learning resources and using them for recommendation. RACOFI studies have not yet assessed the pedagogical value of the recommender, nor do they report some evaluation of the system by learners.

Manouselis and (Manouselis and Costopoulou 2007) tried a typical, neighborhood-based set of collaborative filtering algorithms in order to support learning object recommendation. The examined algorithms have been multi-attribute ones, allowing the recommendation service to consider multi-dimensional ratings that learners provide on learning resources. The performance of the same algorithms is changing, depending on the context where testing takes place. The results from the comparative study of the same algorithms in an e-commerce and a e-learning setting (Manouselis et al. 2011) have led to the selection of different algorithms from the same set of candidate ones.

In summary, the relatively few educational systems with collaborative filtering components have an underlying algorithm to determine solution quality based on collaborative scoring. Yet, existing systems are often specialized for a particular application area such as legal argumentation (LARGO), writing skills training (SWoRD), or educational resource recommendation (AV), or they involve a rather complicated and long-term review process (SWoRD, PG).

The collaborative filtering (CF) based techniques, in general, suffer from several limitations. Two serious limitations with quality evaluation are: the sparsity problem and the "cold-start" problem (Lu 2004). The sparsity problem occurs when available data is insufficient for identifying similar learners or items (neighbours) due to an immense amount of learners and items (Sarwar et al. 2001). It is difficult for collaborative filtering based recommender systems to precisely compute the neighbourhood and identify the learning objects to be recommended even though learners are very active, each individual has only expressed a rating on a very small portion of the items (Linden et al. 2003). Also, a severe problem is the cold start problem (first-rater), which occurs when a new learner/learner object is introduced and thus has no previous ratings information available (Massa and Avesani 2004). With this situation, the system is generally unable to make high quality recommendations.

The CF-based techniques rely heavily on explicit learner input (e.g., previous customers' rating/ranking of products), which is either unavailable or considered intrusive. With sparsity of such learner input, the recommendation precision and quality drop significantly. This is because without good and trusted ratings entered by the learners, recommendations become useless and untrustworthy. To recommend learning activities or learning objects it is better to use real past activities (history logs) by learners as input for their profiles. Also, in the case of intelligent tutoring system, collaborative filtering approach can be carried out according to ratings (grades) for learners' knowledge level, provided by the tutoring system.

6.3.2 Content-Based Techniques

Content-based techniques recommend items (learning objects/learning actions) similar to the ones the learners preferred in the past. They base their recommendations on individual information and ignore contributions from other learners (Billsus and Pazzani 1998). In content-based systems, items are described by a common set of attributes. Learner's preferences are predicted by considering the association between the item ratings and the corresponding item attributes. Therefore, learner can receive proper recommendations without help from other learners. Content-based techniques can be classified into two different categories (Aguzzoli et al 2002; Schmitt and Bergmann 1999; Wilson et al. 2003):

1. Case based reasoning (CBR) techniques and
2. Attribute—based techniques.

Case based reasoning techniques recommend items with the highest correlation to items the learner liked before. Case-based reasoning is useful to keep the learner informed about aimed learning goals. These techniques are domain-independent, do not require content analysis and the quality of the recommendation improves over time when the learners have rated more items. The disadvantage of the new learner problem also states to case-based reasoning techniques. Nevertheless, specific disadvantages of case-based reasoning are overspecialization and sparsity, because only items that are highly correlated with the learner profile or interest can be recommended. Through case-based reasoning the learner is limited to a set of items that are similar to the items (s)he already knows (Adomavicius and Tuzhilin 2005a).

Recent research papers present different facets of CBR in teaching or learning process. Pixed (Project Integrating eXperience in Distance Learning), which is an adaptive hypermedia ontology-based system implements case based reasoning method (Heraud et al. 2004). The Pixed approach assumes positions of a learner as a kind of expert of her/his own learning skills, or at least as a real practitioner of his own practices. The learner builds her/his knowledge by interacting with the learning environment, trying to benefit as much as possible from the available educational activities. Learning is considered as a problem-solving task. The goal is to learn a specific concept proposed in the domain knowledge ontology. The way to reach this goal is one particular path among the different available educational activities linked to that ontology. (Sørmo and Aamodt 2002) propose building "a cognitive model of how humans solve problems in the domain and use this model in attempting to solve the problem, both from the point of view of the current learner (using the learner model) and of an expert (represented by an expert model)". The case-based reasoner has to evaluate the learner's solution and to explain why s/he does or does not fit the observed features of the problem. (Funk and Conlan 2003) make research more closely related to Pixed. Their goal is the same: to use learner feedback in order to adapt the learning environment. The learner feedback can be exploited in two ways: direct feedback exploitation during the learning process, in the form of learners' comments, and feedback exploitation by authors and tutors after the learning process in order to integrate it into the proposed courses, by comparing the learners' result with the result of other cases. The authors associate CBR with filtering techniques by attempting to create learner profiles taking into account different feedbacks. (Elorriaga and Fernández-Castro 2000) propose to use CBR to deploy an instructional planner which adapts the sequences observed in logs in order to create instructional sequences for a complete course. In Heraud et al. (2004), a case—based reasoning system was developed to offer navigational guidance to the learner. It is based on past user's interaction logs and it includes a model describing learning sessions.

Attribute–based techniques recommend items based on the matching of their attributes to the learner profile. Attributes could be weighted for their importance to learner. Adding new LAa or learners to the network will not cause any problem. Attribute-based techniques are sensitive to changes in the profiles of the learners (Drachsler et al. 2008). They can always control the personalized RS by changing their profile or the relative weight of the attributes. A description of needs in their profile is mapped directly to available LA. A serious disadvantage is that an attribute-based recommendation is static and not able to learn from the network behaviour. That is the reason why highly personalized recommendation cannot be achieved. Attribute-based techniques work only with information that can be described in categories. Media types, like audio and video, first need to be classified to the topics in the profile of the learner. This requires category modelling and maintenance which could raise serious limitations for learning environments. Also the overspecialization can be a problem, especially if learners do not change their profile. Attribute-based recommendations are useful to handle the 'cold-start' problem because no behaviour data about the learners is needed. Attribute-based techniques can directly map characteristics of learners (like learning goal, prior knowledge, and available study time) to characteristics of LA (Drachsler et al. 2007). There are several applications that tackle attribute—based techniques problems such as prediction and visualization. Attribute–based Ant Colony System (AACS) (Yang and Wu 2009) uses a method of finding learning objects that would be suitable for a learner based on the most frequent learning trails followed by the previous learners. The system updates the trails pheromones from different knowledge levels and different styles of learners to create a powerful and dynamic learning object search mechanism. There are three prerequisites for achieving this:

1. the adaptive learning portal knows the learner's attributes which include the learner's knowledge level and learning style
2. the learner's attributes and learning object's attributes which have been annotated by teacher or content providers
3. matching the relationships between learners and learning object.

6.3.3 Association Rule Mining

Association rule mining techniques (Agrawal and Srikant 1995) are one of the most popular ways of representing discovered knowledge and describe a close correlation between frequent items in a database. An association rule consists of an antecedent (left-hand side) and a consequent (right-hand side). The intersection between the antecedent and the consequent is empty. An:

$$X \Rightarrow Y$$

type association rule expresses a close correlation between items (attribute-value) in a database (Zheng et al. 2001). Most association rule mining algorithms require the user to set at least two thresholds, one of minimum support and the other of minimum confidence. The support S of a rule is defined as the probability that an entry satisfies both X and Y. Confidence is defined as the probability an entry has satisfies Y when it satisfies X. Therefore the aim is to find all the association rules that satisfy certain minimum support and confidence restrictions, with parameters specified by the user. Therefore, the user must have a certain amount of expertise in order to find the right support and confidence settings to achieve the best rules.

Association rule mining has been applied to e-learning systems in order to intelligently recommend on-line learning activities to learners, based on the actions of previous learners which can improve course content navigation as well as to assist the on-line learning process (Arenas-García et al. 2007).

Count the learners' browsing records, learning path and testing grades and finding out the connection between learning objects, association rule can be used to calculate the learning profiles of the new learners and perform the following tasks:

- building recommender agents for on-line learning activities or shortcuts (Zaíane 2002),
- automatically leading the learner's activities and intelligently recommend on-line learning activities or shortcuts in the course Web site to the learners (Lu 2004),
- identifying attributes of performance inconsistency between various groups of learners (Minaei-Bidgoli et al. 2004),
- discovering interesting learner's usage information in order to provide feedback to course author (Romero et al. 2004),
- finding out the relation among the learning materials from a large amount of educational material data (Lu et al. 2003),
- finding learners' mistakes that are often occur together (Merceron and Yacef 2004),
- optimizing the content of an e-learning portal by determining the content of most interest to the learner (Ramli 2005),
- deriving useful patterns to help educators and instructors evaluating and interpreting on-line course activities (Zaiane 2002), and

- personalizing e-learning based on comprehensive usage profiles and a domain ontology (Markellou et al. 2005).

Most of the subjective approaches involve learner participation in order to express, in accordance to his or her previous knowledge, which rules are of interest. Hence, subjective measures are becoming increasingly important (Silberschatz and Tuzhilin 1996). Some suggested subjective measures (Liu et al. 2000) are:

- *Unexpectedness*: Rules are interesting if they are unknown to the learner or contradict the learner's knowledge.
- *Actionability*: Rules are interesting if learners can do something with them.

There are several specific research papers about the application of association rule mining and recommender systems in e-learning systems. Association rules for classification applied to e-learning (Castro et al. 2007), have been investigated in the areas of learning recommendation systems (Chu et al. 2003; Zaíane 2002), learning material organization (Tsai et al. 2001), learner learning assessments (Hwang et al. 2003; Kumar 2005; Okamoto and Matsui 2003; Silva and Vieira 2001), course adaptation to the learners' behaviour (Hsu et al. 2003; Markellou et al. 2005), and evaluation of educational Web sites (Machado and Becker 2003).

(Wang et al. 2002) developed a portfolio analysis tool based on associative material clusters and sequences among them. This knowledge allows teachers to study the dynamic browsing structure and to identify interesting or unexpected learning patterns. (Minaei-Bidgoli et al. 2004) propose mining interesting contrast rules for Web-based education systems. Contrast rules help one to identify attributes characterizing patterns of performance difference between various groups of learners. (Markellou et al. 2005) propose an ontology-based framework and discover association rules, using the Apriori algorithm. The role of the ontology is to determine which learning materials are more suitable to be recommended to the learner. (Jia Li and Zaïane 2004) use recommender agents for recommending online learning activities or shortcuts in a course Web site based on a learner's access history. (Romero et al. 2004) propose to use grammar-based genetic programming with multi-objective optimization techniques for discovering useful association rules from learner's usage information. (Merceron and Yacef 2004) use association rules and symbolic data analysis, as well as traditional SQL queries to mining learner data captured from a Web-based tutoring tool. Their goal is to find mistakes that often occur together. (Freyberger et al. 2004) use association rules to determine what operation to perform on the transfer model that predicts a learner's success.

Apriori algorithm (Agrawal et al. 1993) is a prominent algorithm for mining frequent itemsets for Boolean association rules. In Apriori algorithm, it is time-consuming that the database has been scanned for many times. Therefore, many algorithms, like the DIC algorithm (Brin et al. 1997), DHP algorithm (Park et al. 1995) and AprioriTid algorithm (Agrawal et al. 1993), etc., are proposed successively to improve the performance.

Association rule mining and frequent pattern mining were applied in (Zaíane 2002) to extract useful patterns that might help teacher, educational managers, and

Web masters to evaluate and understand on-line course activities. A similar approach can be found in Minaei-Bidgoli et al. (2004), where distinguish rules, defined as sets of conjunctive rules describing patterns of performance difference between groups of learners, were used. A computer-assisted approach to diagnosing learner learning problems in science courses and offer learners advice was presented in Hwang et al. (2003), based on the concept effect relationship (CER) model, a specification of the association rules technique.

A hypermedia learning environment with a tutorial component was described in (Costabile De Marsico et al. 2005). It is called Logiocando and targets children of the fourth level of primary school (9–10 years old). It includes a tutor module, based on if-then rules, that emulates the teacher by providing suggestions on how and what to study. In Okamoto and Matsui (2003) it can be found the description of a learning process assessment method that resorts to association rules, and the well-known ID3 DT learning method. A framework that employ Web usage mining to support the validation of learning site designs was defined in Machado and Becker (2003), applying association and sequence techniques (Srivastava et al. 2000).

In Markellou et al. (2005), a framework for personalized e-learning based on aggregate usage profiles and domain ontology were presented, and a combination of Semantic Web and Web mining methods was used. The Apriori algorithm for association rules was applied to capture relations among URL references based on the navigational patterns of learners. A test result feedback (TRF) model that analyses the relationships between learner learning time and the corresponding test results was introduced in Hsu et al. (2003). The objective was twofold: on the one hand, developing a tool for supporting the tutor in reorganizing the course material; on the other, a personalization of the course tailored to the individual learner needs. The approach was based on association rules mining. A rule-based mechanism for the adaptive generation of problems in Intelligent Tutoring System (ITS) in the context of Web-based programming tutors was proposed in Kumar (2005). In Hwang et al. (2003), a Web-based course recommendation system, used to provide learners with suggestions when having trouble in choosing courses, was described. The approach integrates the Apriori algorithm with graph theory.

Some of the main drawbacks of association rule algorithms are (García et al. 2007):

- association rule mining algorithms normally discover a huge quantity of rules and do not guarantee that all the rules found are relevant,
- the used algorithms have too many parameters for somebody non expert in data mining and
- the obtained rules are far too many, most of them non-interesting and with low comprehensibility.

In order to provide better recommendations, and to be able to use recommender systems in more complex types of e-learning environments, most of the methods reviewed in this subsection would need significant extensions. Therefore, we consider some tag-based profiling approaches for extending their capabilities.

References

Adomavicius, G., & Tuzhilin, A. (2005a). Personalization technologies. *Communications of the ACM*. http://doi.org/10.1145/1089107.1089109.

Adomavicius, G., & Tuzhilin, A. (2005b). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Database mining: A performance perspective. *Knowledge and Data Engineering, IEEE Transactions on, 5*(6), 914–925.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on* (pp. 3–14).

Aguzzoli, S., Avesani, P., & Massa, P. (2002). Collaborative case-based recommender systems. In *Advances in Case-Based Reasoning* (pp. 460–474). Springer.

Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., McGrath, S., & Lemire, D. (2003). Racofi: A rule-applying collaborative filtering system.

Angehrn, A., Nabeth, T., Razmerita, L., & Roda, C. (2001). K-InCA: using artificial agents to help people learn and adopt new behaviours. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on* (pp. 225–226).

Arenas-García, J., Meng, A., Petersen, K. B., Lehn-Schioler, T., Hansen, L. K., & Larsen, J. (2007). Unveiling music structure via plsa similarity fusion. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on* (pp. 419–424).

Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*. http://doi.org/10.1145/245108.245124.

Basilico, J., & Hofmann, T. (2004). Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning* (p. 9).

Basu, C., Hirsh, H., Cohen, W., & others. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI* (pp. 714–720).

Billsus, D., & Pazzani, M. J. (1998). Learning Collaborative Information Filters. In *ICML* (Vol. 98, pp. 46–54).

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43–52).

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record* (Vol. 26, pp. 255–264).

Castro, F., Vellido, A., Nebot, À., & Mugica, F. (2007). Applying data mining techniques to e-learning problems. In *Evolution of teaching and learning paradigms in intelligent environment* (pp. 183–221). Springer.

Chen, W., & Wasson, B. (2003). Coordinating collaborative knowledge building. *International Journal of Computers and Applications, 25*(1), 1–10.

Cheung, K. W., Kwok, J. T., Law, M. H., & Tsui, K. C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems*, *35*(2), 231–243. http://doi.org/10.1016/S0167-9236(02)00108-2.

Cho, K., & Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education, 48*(3), 409–426.

Cho, K., Schunn, C. D., & Wilson, R. W. (2006). Validity and reliability of scaffolded peer assessment of writing from instructor and student perspectives. *Journal of Educational Psychology, 98*(4), 891.

Chu, K.-K., Chang, M., & Hsia, Y.-T. (2003). Designing a course recommendation system on web based on the students' course selection records. In *World conference on educational multimedia, hypermedia and telecommunications* (Vol. 2003, pp. 14–21).

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems* (Vol. 60).

Cosley, D., Lawrence, S., & Pennock, D. M. (2002). REFEREE: An open framework for practical testing of recommender systems using ResearchIndex. In *Proceedings of the 28th international conference on Very Large Data Bases* (pp. 35–46).

Costabile, M. F., De Marsico, M., Lanzilotti, R., Plantamura, V. L., & Roselli, T. (2005). On the usability evaluation of e-learning applications. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on* (p. 6b–6b).

David, N. (1997). Implicit Rating and Filtering. In *5th DELOS Workshop on Filtering and Collaborative Filtering (ERCIM), Budapest, Hungary* (pp. 31–36).

Drachsler, H., Hummel, H., & Koper, R. (2007). Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning.

Drachsler, H., Hummel, H. G. K., & Koper, R. (2008). Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. *International Journal of Learning Technology, 3*(4), 404–423.

Drachsler, H., Hummel, H. G. K., & Koper, R. (2009). Identifying the goal, user model and conditions of recommender systems for formal and informal learning. *Journal of Digital Information*, *10*(2).

Elorriaga, J. A., & Fernández-Castro, I. (2000). Using case-based reasoning in instructional planning. towards a hybrid self-improving instructional planner. *International Journal of Artificial Intelligence in Education (IJAIED), 11*, 416–449.

Freyberger, J., Heffernan, N., & Ruiz, C. (2004). Using association rules to guide a search for best fitting transfer models of student learning. In *Workshop on analyzing student-tutor interactions logs to improve educational outcomes at ITS conference* (pp. 1–10).

Funk, P., & Conlan, O. (2003). Using case-based reasoning to support authors of adaptive hypermedia systems. In *AH2003: workshop on adaptive hypermedia and adaptive web-based systems* (pp. 113–120).

García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education, 49*(3), 794–808.

García, E., Romero, C., Ventura, S., & De Castro, C. (2009). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction, 19*(1–2), 99–132.

Gaudioso, E., Santos, O. C., Rodríguez, A., & Boticario, J. G. (2003). A proposal for modeling a collaborative task in a web-based collaborative learning environment. In *International Conference on User Modeling* (pp. 70–79).

Gehringer, E. F. (2001). Electronic peer review and peer grading in computer-science courses. *ACM SIGCSE Bulletin, 33*(1), 139–143.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61–70.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*(2), 133–151.

Heraud, J.-M., France, L., & Mille, A. (2004). Pixed: An ITS that guides students with the help of learners' interaction log. In *International conference on intelligent tutoring systems, workshop analyzing student tutor interaction logs to improve educational outcomes. Maceio, Brazil* (pp. 57–64).

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*. http://doi.org/10.1145/963770.963772.

Hofmann, T. (2003). Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 259–266).

Hsu, H.-H., Chen, C.-J., & Tai, W.-P. (2003). Towards error-free and personalized Web-based courses. In *Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on* (pp. 99–104).

Hwang, G.-J., Hsiao, C.-L., & Tseng, J. C. R. (2003). A computer-assisted approach to diagnosing student learning problems in science courses. *J. Inf. Sci. Eng., 19*(2), 229–248.

Janssen, J., den Berg, B., Tattersall, C., Hummel, H., & Koper, R. (2007). Navigational support in lifelong learning: enhancing effectiveness through indirect social navigation. *Interactive Learning Environments, 15*(2), 127–136.

Koper, R. (2005). Increasing learner retention in a simulated learning network using indirect social interaction. *Journal of Artificial Societies and Social Simulation*, *8*(2).

Koper, R., & Olivier, B. (2004). Representing the learning design of units of learning. *Educational Technology and Society*.

Kumar, A. (2005). Rule-based adaptive problem generation in programming tutors and its evaluation. In *The 12th international conference on artificial intelligence in education. July* (pp. 18–22).

Lemire, D. (2005). Scale and translation invariant collaborative filtering systems. *Information Retrieval, 8*(1), 129–150.

Lemire, D., Boley, H., McGrath, S., & Ball, M. (2005). Collaborative filtering and inference rules for context-aware learning object recommendation. *Interactive Technology and Smart Education, 2*(3), 179–188.

Li, J., & Zaïane, O. R. (2004). Combining usage, content, and structure data to improve web site recommendation. In *E-Commerce and Web Technologies* (pp. 305–315). Springer.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, *7*(1), 76–80.

Liu, B., Hsu, W., Chen, S., & Ma, Y. (2000). Analyzing the subjective interestingness of association rules. *Intelligent Systems and Their Applications, IEEE, 15*(5), 47–55.

Lu, J. (2004). Personalized e-learning material recommender system. In *International conference on information technology for application* (pp. 374–379).

Lu, J., Yu, C. S., & Liu, C. (2003). Learning style, learning patterns, and learning performance in a WebCT-based MIS course. *Information and Management*, *40*(6), 497–507. http://doi.org/10.1016/S0378-7206(02)00064-2.

Luo, S., Sha, S., Shen, D., & Jia, W. (2002). Conceptual network based courseware navigation and web presentation mechanisms. In *Advances in Web-Based Learning* (pp. 81–91). Springer.

Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). Defining ill-defined domains; a literature survey. In *Proceedings of the workshop on intelligent tutoring systems for ill-defined domains at the 8th international conference on intelligent tutoring systems* (pp. 1–10).

Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., & Cohen, M. D. (1987). Intelligent information-sharing systems. *Communications of the ACM, 30*(5), 390–402.

Machado, L. dos S., & Becker, K. (2003). Distance education: A web usage mining case study for the evaluation of learning sites. In *Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference on* (pp. 360–361).

Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, *10*(4), 415–441. http://doi.org/10.1007/s11280-007-0019-8.

Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., & Koper, R. (2011). Recommender Systems in Technology Enhanced Learning. In *Recommender Systems Handbook* (pp. 387–415). http://doi.org/10.1007/978-0-387-85820-3.

Markellou, P., Mousourouli, I., Spiros, S., & Tsakalidis, A. (2005). Using semantic web mining technologies for personalized e-learning experiences. *Proceedings of the Web-Based Education*, 461–826.

Marlin, B. M. (2003). Modeling User Rating Profiles For Collaborative Filtering. In *NIPS* (pp. 627–634).

Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE* (pp. 492–508). Springer.

McLaughlin, M. R., & Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 329–336).

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Making recommendations better: An analytic model for human-recommender interaction. In *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1103–1108).

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI* (pp. 187–192).

Merceron, A., & Yacef, K. (2004). Mining student data captured from a web-based tutoring tool: Initial exploration and results. *Journal of Interactive Learning Research, 15*(4), 319.

Miller, B. (2003). *Toward a personalized recommender system*. PhD thesis, University of Minnesota–Twin Cities.

Minaei-Bidgoli, B., Tan, P.-N., & Punch, W. (2004). Mining interesting contrast rules for a web-based educational system. In *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on* (pp. 320–327).

Okamoto, T., & Matsui, T. (2003). Knowledge Discovery from Learning History Data and and its Effective Use for Learning Process Assessment under the e-learning Environment. In *Society for Information Technology & Teacher Education International Conference* (Vol. 2003, pp. 3141–3144).

Park, J. S., Chen, M.-S., & Yu, P. S. (1995). *An effective hash-based algorithm for mining association rules* (Vol. 24). ACM.

Pennock, D. M., & Horvitz, E. (1999). Analysis of the axiomatic foundations of collaborative filtering. *Ann Arbor, 1001*, 42110–48109.

Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* (pp. 473–480).

Pérez, T., Lopistéguy, P., Gutiérrez, J., & Usandizaga, I. (1995). HyperTutor: From hypermedia to intelligent adaptive hypermedia. In *ED-MEDIA'95, World conference on educational multimedia and hypermedia* (pp. 529–534).

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006). Toward legal argument instruction with graph grammars and collaborative filtering techniques. In *Intelligent Tutoring Systems* (pp. 227–236).

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2007). Evaluating legal argument instruction with graphical representations using largo. *Frontiers In Artificial Intelligence And Applications, 158*, 101.

Ramli, A. A. (2005). Web usage mining using apriori algorithm: UUM learning care portal case. In *International conference on knowledge management, Malaysia* (pp. 1–19).

Recker, M. M., & Walker, A. (2003). Supporting" word-of-mouth" social networks through collaborative information filtering. *Journal of Interactive Learning Research, 14*(1), 79.

Recker, M. M., Walker, A., & Lawless, K. (2003). What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education. *Instructional Science, 31*(4–5), 299–316.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175–186).

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*. http://doi.org/10.1145/245108.245121.

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook. Recommender Systems Handbook*. http://doi.org/10.1007/978-0-387-85820-3_1.

Romero, C., Ventura, S., & De Bra, P. (2004). Knowledge discovery with genetic programming for providing feedback to courseware authors. *User Modelling and User-Adapted Interaction*, *14*(5), 425–464. http://doi.org/10.1007/s11257-004-7961-2.

Romero, C., Ventura, S., De Bra, P., & De Castro, C. (2003). Discovering prediction rules in AHA! courses. In *User Modeling 2003* (pp. 25–34). Springer.

Romero, C., Ventura, S., Delgado, J. A., & De Bra, P. (2007). Personalized links recommendation based on data mining in adaptive educational hypermedia systems. In *Creating New Learning Experiences on a Global Scale* (pp. 292–306). Springer.

Rosset, S., Neumann, E., Eick, U., Vatnik, N., & Idan, Y. (2002). Customer lifetime value modeling and its use for customer retention planning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 332–340).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th …*, *1*, 285–295. http://doi.org/10.1145/371920.372071.

Savidis, A., Grammenos, D., & Stephanidis, C. (2007). Developing inclusive e-learning and e-entertainment to effectively accommodate learning difficulties. *Universal Access in the Information Society, 5*(4), 401–419.

Schafer, J. Ben, Konstan, J., & Riedl, J. (2001). E-commerce recommendation applications. *Applications of Data Mining to Electronic …*, 115–153. http://doi.org/10.1007/978-1-4615-1627-9_6.

Schmitt, S., & Bergmann, R. (1999). Applying case-based reasoning technology for product selection and customization in electronic commerce environments. In *12th Bled Electronic Commerce Conference* (Vol. 273).

Shardanand, U., & Maes, P. (1995). Social Information Filtering: Algorithms for Automating "Word of Mouth." In *ACM Conference on Human Factors in Computing Systems (CHI)* (Vol. 1, pp. 210–217). http://doi.org/10.1145/223904.223931.

Shen, L., & Shen, R. (2005). Ontology-based learning content recommendation. *International Journal of Continuing Engineering Education and Life Long Learning, 15*(3–6), 308–317.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *Knowledge and Data Engineering, IEEE Transactions on, 8*(6), 970–974.

Silva, D. R., & Vieira, M. T. P. (2001). An Ongoing assessment model in distance learning.

Soonthornphisaj, N., Rojsattarat, E., & Yim-Ngam, S. (2006). Smart e-learning using recommender system. In *Computational intelligence* (pp. 518–523). Springer.

Sørmo, F., & Aamodt, A. (2002). Knowledge communication and CBR. In *ECCBR Workshops* (pp. 47–60).

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter, 1*(2), 12–23.

Tan, H., Guo, J., & Li, Y. (2008). E-learning recommendation system. In *Computer Science and Software Engineering, 2008 International Conference on* (Vol. 5, pp. 430–433).

Tsai, C.-J., Tseng, S.-S., & Lin, C.-Y. (2001). A two-phase fuzzy mining and learning algorithm for adaptive learning environment. In *Computational Science-ICCS 2001* (pp. 429–438). Springer.

Vassileva, J. (1998). DCG + GTE: Dynamic Courseware Generation with Teaching Expertise. *Instructional Science, 26*(3–4), 317–332.

Walker, A., Recker, M. M., Lawless, K., & Wiley, D. (2004). Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence in Education, 14*(1), 3–28.

Wang, D., Bao, Y., Yu, G., & Wang, G. (2002). Using page classification and association rule mining for personalized recommendation in distance learning. In *Advances in Web-Based Learning* (pp. 363–374). Springer.

Wilson, D. C., Smyth, B., & Sullivan, D. O. (2003). Sparsity reduction in collaborative recommendation: A case-based approach. *International Journal of Pattern Recognition and Artificial Intelligence, 17*(05), 863–884.

Yang, Y. J., & Wu, C. (2009). An attribute-based ant colony system for adaptive learning object recommendation. *Expert Systems with Applications, 36*(2), 3034–3047.

Zaiane, O. (2002). Building a Recommender Agent for e learning Systems Computers in Education. *Citeulike.org*.

Zaíane, O. R. (2002). Building a recommender agent for e-learning systems. In *Computers in Education, 2002. Proceedings. International Conference on* (pp. 55–59).

Zaïane, O. R., & Luo, J. (2001). Towards evaluating learners' behaviour in a web-based distance learning environment. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on* (pp. 357–360).

Zheng, Z., Kohavi, R., & Mason, L. (2001). Real world performance of association rule algorithms. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 401–406).

Ziegler, C.-N. (2013). *Social Web Artifacts for Boosting Recommenders*. Springer.