Recommender Systems Review: Types, Techniques and Applications

George A. Sielis
University of Cyprus, Cyprus
Aimilia Tzanavari
University of Nicosia and Cyprus University of Technology, Cyprus
George A. Papadopoulos
University of Cyprus, Cyprus

ABSTRACT
Recommender or recommendation systems are software tools that make useful suggestions to users, by taking into account their profile, preferences and/or actions during interaction with an application or website. They are usually personalized and can refer to items to buy, people to connect to or books/ articles to read. Recommender Systems (RS) aim at helping users with their interaction by bringing to surface the information that is relevant to them, their needs, or their tasks. This chapter's objective is to present a review of the different types of RS, the techniques and methods used for building such systems, the algorithms used to generate the recommendations and how these systems can be evaluated. Finally, a number of topics are discussed as envisioned future research directions.

INTRODUCTION
Recommender or Recommendation Systems (RS) are software tools in applications or websites that suggest information (e.g. items, people, news articles) that might be of interest to the end user, taking into account various types of knowledge and data, such as the user's preferences, actions, tasks and contextual information. In most cases these systems use computational methods to analyze users' past actions and decisions, along with other user-related or task-related information, to offer useful, usually personalized recommendations. The motivation behind this is to alleviate the information overload problem, by bringing to surface what is most relevant, interesting to the user. Examples can be seen in many well-known e-commerce websites such as Amazon.com, which promotes products that were last examined or purchased by a user, or products that have been rated or reviewed by other users. In addition to that, there are cases where the recommendations are the result of a combination of factors that are difficult to accurately determine. In such cases, a variety of alternative methods are employed to generate recommendations.

RS belong to the information filtering systems family and therefore seek to predict the rating or the preference that a user would give to an item. Thus the common methodology that RS follow is to find the correlation between three types of models, user, rating and item, in order to produce recommendations. All recommendation algorithms and their variations follow this model for the computation of recommendations. The present chapter includes a review of the different types of RS, their typical architecture and the algorithms used for generating recommendations. Finally the challenging topic of evaluation of RS is discussed, outlining the possible approaches to assess the accuracy, usefulness and user satisfaction from recommendations.

BACKGROUND

The Internet is a source of information where we can find almost everything. Nevertheless, this information is not structured and well organized and following the web's expansion information filtering has become more complex. This complexity is due to more information-related factors being introduced, e.g. social, psychological, and behavioral and other factors related to the users who receive or created the information. RS are mechanisms that are used for filtering and removing the irrelevant information based on how each user perceives the information. In other words, RS take into account the preferences of a particular user, compare it to what other users with similar preferences liked or disliked and try to predict the information that would satisfy the user the most. Based on this logic several recommendation algorithms have been implemented and used in commercial, as well as research recommendation tools.

RECOMMENDATION FILTERING TECHNIQUES/ ALGORITHMS

In general RS refer to the production of recommendations to be presented to a user, where these recommendations are useful to the user for the accomplishment of a task. This task might be related to navigation to web pages that interest the user, finding items to buy, explore learning resources or find people to socialize with or collaborate with. The types of recommendations are usually domain and task dependent and consequently context dependent. Specifically, recommendations systems can be identified as content-based, collaborative, or hybrid, depending on the basis of the filtering technique they use. In this section we describe the high level RS architecture and describe the several recommendation techniques that can be used for the development of a RS.

Functional Architecture of Recommender Systems

A recommender system consists of cyclic functioning procedures that are divided in the following four steps: *Data collection, data filtering, Rank the recommended items* and *Presentation of data.*
By the execution of the above-mentioned procedural steps a recommender system aims at two tasks. Firstly, the production of recommendations and secondly to use the users feedback after the delivery of the recommendations to them, so the process can be repeated and produce new recommendations as it is depicted in Figure 1.

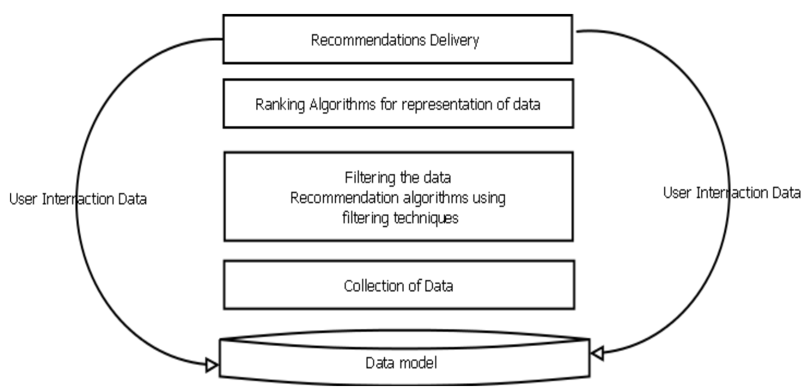

Figure 1. Functional RS Architecture

Collection of data - The collection of data is directly correlated to the data model that is used within a software application. The data is usually defined based on the overall design of a software application and based on the contextual information that a software application collects and processed into further computations. The data model usually depends on the domain that a recommender system is built for and the methods of storage and representation of the data (e.g. relational databases or semantic web representation methods).

Recommendation filtering techniques - As mentioned above the recommendation filtering techniques depend on the type of data that a RS is processing and the type of recommendations it plans to produce. The recommendation filtering techniques are divided in three categories according to the type of data they use for the computation of recommendations as well as the computational algorithm methods they use. The three categories are the Content-based Filtering (CBF), the Collaborative Filtering (CF) and the Hybrid Filtering (HF) techniques.

Ranking Algorithms and Representation of recommendations - The collection of data and its parsing through the recommendation filtering techniques generate a set of recommendations that fulfill the rules that a recommender system has to take into account during the computations. The next and final step that a RS has to complete is to present the generated set of recommendations to the final recipients, the users. Recommendations must be presented to the users ranked. The ranking of the recommendations is based on the preferences of each user and their personal interests. A recommendation is considered successful when the highest in priority recommendation offered to a user is closer to her interests and when this recommendation is actually accepted by the user.

Recommendation filtering techniques

TF-IDF - One of the most known and most used CBF techniques is the TF-IDF (Term Frequency – Inverse Document Frequency) measure (Salton, 1989). TF-IDF measure is defined as follows: For total number of documents N that can be recommended to user's keyword k_i appears in n_i of them. If $f_{i,j}$ is the number of times that k_j appears in document d_j then $TF_{i,j}$ is the term frequency of keyword k_i in document d_j and it is defined as

$$TF_{ij} = \frac{f_{i,j}}{\max_z f_{z,j}}$$

Keywords may appear in many documents, so with TF, keywords are not useful in distinguishing the relevance between documents. For that reason the inverse document frequency (IDF_i) is used in combination with the simple term frequency ($TF_{i,j}$). The IDF_i is defined as

$$IDF_i = \log\left(\frac{N}{n_i}\right)$$

Finally the TD-IDF weight for each keyword k_j in each particular document d_j is defined as

$$w_{i,j} = TF_{i,j} \times IDF_i$$

and the content for each document dj is defined as

$$Content(d_j) = (w_{1j}, \dots \dots . w_{kj})$$

Naïve Bayes - Naïve Bayes algorithm is a machine learning probabilistic algorithm and belongs to the general class of Bayesian Classifiers. Bayesian Classifiers construct their models based on previous observations used as training data. The Bayesian method estimates the a posteriori probability of document d belonging in class c, which is the probability P(c|d). Priori probability P(c) is the probability of observing a document in class c, probability P(d|c) is the probability of observing the document d given the class c and probability P(d) is the probability of observing the instance d. The Bayesian theorem is expressed as:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

and a document is classified using the class with the highest probability

$$c = argmax_{c_i} \frac{P(c_j)P(d|c_j)}{P(d)}$$

Rocchio's algorithm - Rocchio's Algorithm applies the relevance feedback technique, a technique that helps users incrementally refine queries based on previous search results. Rocchio's algorithm represents documents as vectors, so that documents with similar content have similar vectors. Each component of such a vector corresponds to a term in the document, typically a word. The weight of each component is computed using the TF-IDF term weighting scheme. Learning is achieved by combining document vectors (of positive and negative examples) into a prototype vector for each class in the set of classes C. To classify a new document d, the similarity between the prototype vectors and the corresponding document vector representing d are calculated for each class (for example by using the cosine similarity measure), then d is assigned to the class whose document vector has the highest similarity value.

Decision Trees - A decision tree is a collection of nodes arranged as a binary tree. The arcs of the tree represent decisions and the nodes contain the classified objects. The decision arcs can take values such as true or false. For each decision value the tree expands containing new children nodes containing new decisions until a leaf is reached. For the case of a true–false decision tree, if the decision referring to the root node is true then follow the path to the right child whereas if it is false follow the path to the left child. This is repeated for each child node until a leaf is reached, where a leaf is the last child node of the path and it contains the classification value of the item.

TYPES OF RECOMMENDER SYSTEMS

The recommendation filtering techniques and the method that each recommender system employs in the several domains of application, can divide RS into types. In the following section the several types of RS are presented, along with examples of applications in which they are used.

Content-Based Recommendation Systems

The usual methodology used for implementing CBF recommendations is the combination of CBF techniques and CF techniques. Pazzani (1996) separates the CBF recommendations into a two-phase methodology; first is the recommendation of items to a user based on the description analysis of the item in relation to the interests of a particular user and second, the implementation of strategies for the representation of items, the creation of user profiles that contain description of the types of items that the user likes or dislikes and finally strategies for the comparison of user profile to reference characteristics.

Collaborative Recommendation Systems

Terveen (2005) defines CF as the exploration of techniques for matching people with similar interests. This approach premises the participation of many people, of an easy way to represent their interests and finally the adoption of algorithms that are able to match people with similar interests (Adomavicious, 2005). CF systems are usually applied in software systems addressing communities of users, who express preferences for items and they are produced based on the users' preferences similarities.

CF algorithms are classified in two categories based on the recommendation technique in which each algorithm is applied: Memory-based and Model-based recommendation algorithms. Memory-based algorithms process the entire set of data of a space S for the production of recommendations, while on the contrary, Model-based algorithms use a subset of an entire set of data. The majority of Memory-based algorithms, are related to CF in combination with CBF techniques and Model-based algorithms are commonly used in Rule-based or Hybrid filtering RS (Balabanovic, 1997; Basu, 1998; Schafer, 2007).

Knowledge-based Recommendation Systems

Knowledge-Based RS (KBRS) offer recommendations that already exist in databases or knowledge bases that are therefore not dynamically influenced by ratings or recent preferences. KBRS are divided into two subcategories:

- **Case-Based Systems** - CBR are based on case-based reasoning, which relies on the similarity between a current case and the solutions that already exist in a database. The interaction with a CBR System consists of four steps cycle: Retrieve, Reuse, Revise and Retention (Mantaras, 2005).
- **Constrained-Based Systems** - Based on a given set of preferences, Constraint Based Systems provide a set of possible solutions including explanations as to why these solutions were selected. A Constrained Satisfaction Problem is defined in (Jannach, 2012) Constraints set can be of three types: Compatibility Constraints, Filter conditions and Product constraints (Jannach, 2012).

Trust-based Recommendation Systems

Trust-based recommendations (TR) are an enhancement of the classical recommendation techniques that aimed to improve the accuracy of the recommendation results taken from the well-known CF techniques. The logic behind TR is the use of graphs representing the relation between users and items based on their connection on particular attributes. TRs are commonly used in social networks where a huge number of users are connected within the network and usually the users are connected because of a reason or an attribute. The enhancement of trust-based techniques in relation to the traditional collaborative techniques considers the replacement of the similarity measure used in the formulas, with the trust factor between pairs of users. Trust – Based weighted mean:

$$r(u,i) = \frac{\sum_{v}^{v \in R^T} t_{u,v} r_{v,i}}{\sum_{v}^{v \in R^T} t_{u,v}}$$

The Trust – Based weighted mean present the trust between users' u and v as a weight value in place of the similarity measure between users' u and v. Trust – based CF:

$$r(u,i) = mean(r_u) + \frac{\sum_{v}^{v \in R^T} t_{u,v} (r_{v,i-mean(r_u)})}{\sum_{v}^{v \in R^T} t_{u,v}}$$

This formula is a refinement of the Resnick's formula where the Pearson's Correlation Coefficient is replaced be the trust factor t as a ratings weight factor. Trust filtered mean:

$$r(u,i) = \frac{i}{|R^T|} \sum_{v}^{v \in R^T} r_{v,i}$$

In the existing literature more techniques and trust based enhancement can be found such as "Ensemble Trust" (Victor, 2011) and Trust-filtered CF (Smyth, 2005).

Context-Aware Recommendation Systems

Context Awareness in RS involves the use of data that characterizes an entity in order to use them as contextual information for the computation of recommendations, wherever this is needed. Through the general overview in RS it became noticeable that the major effort in building RS is focused on the well-modeled information that the RS use for their computations. Using the latter as a principle for the development of RS, Context Awareness is used as a mean for the collection of the valuable information that characterizes entities within a system. Adomavicius (2005) correlates the context with topics that are related with RS, such as Data Mining, e-Commerce Personalization, Ubiquitous and mobile Context Aware RS and Information Retrieval aiming to demonstrate the need of Context within RS.

EVALUATION OF RECOMMENDER SYSTEMS

Shani (2011), separate the RS evaluation in three experimental levels. The separation of the evaluation into levels facilitates the evaluation of RS by comparing them to each other. The evaluation levels depend on attributes and characteristics that each system has. Moreover, the types of evaluation that can be applied on RS depend on the data model that a recommender system is using, the domain that the RS refers to and the type of the expected results. Thus Shani et al. separate the evaluation of RS into the following three levels: offline experiments, user studies and online evaluation. In the same work it is highlighted that for each RS evaluation experiment setup it is important to follow the following basic guidelines:

- Before running an experiment a hypothesis must be formed
- When comparing candidate algorithms on certain hypothesis, all variables that are not tested will stay fixed
- When drawing conclusions from experiments, the extracted conclusions must be able to generalize beyond the immediate context of the experiments.

The evaluation of RS is the proof that the developed system and its recommendation algorithm produce the desirable recommendations. Thus the evaluation process must be taken into account from the design phase of a RS to its final development. The awareness of how a Recommender System will be applied, aids to design the appropriate evaluation experiment that corresponds to what the RS is focused on.

Evaluation Metrics for Recommender Systems

RS aim to offer alternative selections related to the context of the working environment or application that is the RS is applied on, as well as based on the user preferences and actions. A successful recommendation is the recommendation that got selected by the user but at the same time satisfied her/him. The satisfaction of the user can vary according to what the user want achieve. In this section we examine the most well-known evaluation metrics.

Prediction based metrics – Accuracy metrics

Parra (2013) define the prediction metrics as the metrics used to compare which Recommender System makes fewer mistakes when inferring how a user will evaluate a proposed recommendation.
Based on (Hernandez del Olmo, 2008) "*accuracy metrics measure the quality of nearness to the truth or the true value achieved by a system*" and usually accuracy is measured using the following expression.

$$accuracy = \frac{number\ of\ successful\ recommendations}{number\ of\ recommendations}$$

A more flexible metric used as a prediction metric is the Mean Absolute Error (MAE). MAE measures the average absolute standard deviation between each predicted rating and each user's real selections. For better understanding we depict the MAE equations as they are used in (Parra, 2013).

$$MAE = \frac{\sum_{i=1}^{N}|p_i - r_i|}{N}$$

Where pi is the predicted rating ri is the actual rating and N is the total number of predictions. For cases with larger deviances from the actual ratings the Means Squared Error MSE is used instead of MAE.

$$MSE = \frac{\sum_{i=1}^{N}(p_i - r_i)^2}{N}$$

Information Retrieval Related Metrics

Information Retrieval refers to the recommendation of items based on their content. Such recommendations might be documents that are recommended with the use of tag-based filtering methods. The metrics used for the evaluation of such recommendations do not rely on the user's preference through ratings but mostly on the usefulness of the recommendation or the user's satisfaction. The most commonly used metrics are the recall and precision or the Discounted Cumulative Gain (DCG). According to (Manning, 2008) precision is the fraction of recommended items that are relevant and it is expressed as

$$Precision = \frac{|relevant\ items\ recommended|}{|items\ in\ the\ list|}$$

Recall is defined in (Manning. 2008) as the fraction of relevant recommendations that are presented to the user and it is expressed as

$$Recall = \frac{|relevant\ items\ recommended|}{|relevant\ items|}$$

The necessity of knowing the relevant items in order to be able to compute the recall metric led (Herlocker, 2004) to characterize recall as useless for the evaluation of a Recommender System. Overcoming the latter characterization, recall is useful in cases where the evaluation of RS is done with the use of datasets for which the recommended items and the user preferences already exist.
Discounted Cumulative Gain (DCG) (J¨arvelin, 2000; J¨arvelin, 2002) - DCG is used for measuring the effectiveness of recommendation items based on their order of appearance.
DCG is defined as:

$$DCG = \sum_{i}^{p} \frac{2^{rel_i - 1}}{\log_2(1+i)}$$

Where *p* is the position of the item in the ranked list and rel_i is the graded relevance of the item at the position *i*.
Maximum Matrix Factorization (MMMF) - MMMF is an effective method used for the estimation of the rating functions by taking advantage of the collaborative effects such as rating patterns from other users which are used to estimate ratings for the current user (Weimer, 2007). The exact definition of the MMMF method is given by (Rish, 2008).

Diversity, Novelty and Coverage

A user will consider a recommendation list as successful if it contains results that she is not aware of but at the same time are satisfying her tastes (McNee, 2006). Beyond accuracy it is also very important for a RS to produce useful and satisfying recommendations. For the differentiation between accuracy and satisfaction or the accuracy and usefulness, Diversity, Novelty and Coverage metrics have been introduced. The novelty of a piece of information generally refers to how different is the piece with respect to "what has been previously seen", by a specific user, or by a community as a whole (Castells, 2011). Diversity generally applies to a set of items, and is related to how different the items are with respect to each other. Coverage refers to the percentage of items, part of the problem domain, that a recommender system can produce for a user or a group of users. (Ziegler, 2005) propose the Intra-List Similarity for measuring the diversity of a recommendation list and defined as:

$$ILS(P_{w_i}) = \frac{\sum_{b_k \in \partial P_{w_i}} \sum_{b_e \in \partial P_{w_i}, b_k \neq b_e} c_0(b_k, b_e)}{2}$$

Where c_0 (b_k, b_e) is the similarity between the two items b_k and be and Pw_i is the intra-list similarity of a list.

FUTURE RESEARCH DIRECTIONS

Recommender Systems research is usually combined with other research topics, depending on the particular domain of application. In the existing literature we can find several research works on RS in e-commerce, e-learning, social networking, ubiquitous computing and others.

A new emerging challenge in the area of RS is that of multi-criteria RS. The usefulness and the satisfaction resulting from recommendations in social networks is a result of a combination of factors and attributes that in many cases are hard to predict. Human behaviors, social activities, personal interests or psychological factors are some of the criteria that the new RS have to take into account in their computations. In that manner the similarity measures are not computed as linear equations, but they are computed as multi-criteria complex equations. Ubiquitous computing, another domain in which RS are applicable is related to mobile technologies and particularly to smart technologies. The definition of the context and modeling the context for its usage in Context Aware Personalized recommendations is one of the emerging challenges in ubiquitous computing.

Another area in which RS are being applied in many ways is e-Learning. RS in e-learning mainly targeted in the recommendation of CBF on knowledge bases. The aim of the usage of RS in learning environments was always to produce recommendations about the learning material that was related to a learning topic. Modern e-learning systems use RS for more than that. Users of e-learning environments expect from the system to guide them (learning strategies), being adapted to them (learning styles, adapted interfaces) or track their needs and offer to them new learning material which is part of their interests.

Finally we will refer to the use of RS for fostering and supporting creativity. Creativity is by definition multi-dimensional and can depend on several factors like for example prior knowledge, social background etc. A creativity process is usually related to novelty and problem solving and is carried out by individual users or groups of users. During an individual or a collaborative creative process several recommendation types can be used in order to guide and enhance the process by offering supporting recommendations. .

CONCLUSION

By reviewing the RS types and filtering techniques we can see that research in the topic of RS is challenging and at the same time very interesting. The plurality of algorithms and recommendation filtering methods can be used for different types of recommendations based on the domain they are applied for. The selection of particular methods to apply in a RS is based on the desired recommendation results. Thus the development of a recommendation algorithm must be accompanied by the corresponding evaluation metrics. This chapter gave readers an introduction to the area, through the overview of the types of RS, the frameworks they can use for development and the metrics that can be used for recommendations evaluation. Specifically, it addressed the following main topics: presentation of the functional architecture of a RS; the types of RS that exist; the filtering and similarity methods used by RS and also the most important evaluation methods and metrics that can be used for evaluating the RS. We concluded the chapter with a subjective selection of topics considered as emerging challenges, elaborating on possible future directions of RS research.

REFERENCES

Adomavicius, G. Tuzhilin, A. (2005). *Toward the next generation of recommender systems: a survey of the state-of-theart and possible extensions.* IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749.

Asnicar, F., Tasso, C. (1997). *ifWeb: a Prototype of User Model-based Intelligent Agent for DocumentationFiltering and Navigation in the Word Wide Web.* C. Tasso, A. Jameson, C.L.

Balabanović, M., Shoham, Y. (1997). *Fab: content-based, collaborative recommendation*. Commun. ACM 40, 3 (March 1997), 66-72. DOI=10.1145/245108.245124 http://doi.acm.org/10.1145/245108.245124
Basu, C., Hirsh, H., Cohen, W. (1998). *Recommendation as classification: using social and content-based information in recommendation*, in Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98) (Madison, WI), American Association for Artificial Intelligence, pp. 714–720.
Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T. (2004). *Evaluating collaborative filtering recommender systems*. ACM Trans. Inf. Syst. 22, 5–53
Hernandez del Olmo. F., Gaudioso, E. (2008). *Evaluation of recommender systems: A new approach*. Expert Syst. Appl. 35, 3 (October 2008), 790-804. DOI=10.1016/j.eswa.2007.07.047 http://dx.doi.org/10.1016/j.eswa.2007.07.047
Jannach, D., Zanker, M., Felfernig,A., Friedrich, G. (2012). *Recommender Systems: An Introduction.* Robin Burke, International Journal of Human-Computer Interaction, Vol. 28, Iss. 1
Järvelin, K., Kekäläinen J. (2002). *Cumulated gain-based evaluation of ir techniques*. ACM Trans. Inf. Syst. 20, 422–446
Järvelin, K., Kekäläinen J. (2000) *IR evaluation methods for retrieving highly relevant documents.* Published in: Belkin, N.J., Ingwersen, P. and Leong, M.-K. (eds.) Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY: ACM, pp. 41–48.
Lieberman, H. (1995). *Letizia: an Agent that Assists Web Browsing*. Proceedings of the International Joint Conference on Artificial Intelligence, pp. 924–929. Morgan Kaufmann.
Manning, C.D., Raghavan, P., Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge
Mantaras, R., Mcsherry, D., Bridge,D. (2005). *Retrieval, reuse, revision and retention in case-based reasoning*. The Knowledge Engineering Review, Vol. 20, No. 03. (2005), pp. 215-240, doi:10.1017/S0269888906000646.
McNee, S. M. Riedl, J., Konstan, J. A. (2006) *Being accurate is not enough: how accuracy metrics have hurt recommender systems*. In ACM CHI EA.
Marmanis, H., & Babenko, D. (2009). *Algorithms of the intelligent web*. Greenwich: Manning.
Mladenic, D. (1999). *Machine learning used by PersonalWebWatcher*. Proceedings of ACAI-99 Modeling on the World Wide Web, Sixth International Conference on User Modeling, pp. 3–12. Chia Laguna, Sardinia, Italy
Moukas, A. (1997). *Amalthaea Information Discovery and Filtering Using a Multiagent Evolving Ecosystem*. Applied Artificial Intelligence 11(5), 437–457
Pazzani, M., Billsus, D. (1997). *Learning and Revising User Profiles: The Identification of Interesting*
Pazzani, M., Muramatsu J., Billsus, D. (1996). *Syskill&Webert: Identifying interesting web sites*. Proceedings of the National Conference on Artificial Intelligence (pp. 54–61). Portland, OR.
Rish I., Tesauro G. (2008).*Active collaborative prediction with maximum margin matrix factorization.* International Symposium on Artificial Intelligence and Mathematics.
Salton, G.(1989). *Automatic Text Processing*, Addison-Wesley
Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007). *Collaborative filtering recommender systems.* In The adaptive web, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Lecture Notes In Computer Science, Vol. 4321. Springer-Verlag, Berlin, Heidelberg 291-324. Science 69 (2000), no. 32, 180–200.

Shani, G., Gunawardana, A. (2011). *Evaluating recommendation systems*. In: Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, Berlin, pp 257-298.
Smyth, B., O'Donovan, J. (2005). *Trust in recommender systems.* 10th international conference on Intelligent user interfaces, pages 167–174.
Terveen, L., McDonald, D.W.(2005). *Social matching: A framework and research agenda*. ACM Trans. Comput.-Hum. Interact. 12, 3 (September 2005), 401-434. DOI=10.1145/1096737.1096740 http://doi.acm.org/10.1145/1096737.1096740
University Press, New York
Victor, P. Cornelis, C. Cock, M. D., Teredesai. A. M. (2011). *Trust- and distrust-based recommendations for controversial reviews*. IEEE Intelligent Systems, 26:48–55, *Web Sites.*
Machine Learning 27(3), 313–331 Workshop on Machine Learning and Intelligent Agents
Weimer, M. Karatzoglou, A., Le, Q., Smola, A. (2007). *COFI RANK - Maximum Margin Matrix Factorization for Collaborative Ranking*. In Advances in Neural Information Processing Systems 20, pp. 1593-1600
Yu, K., Schwaighofer A., Tresp,V., Xu , X., Kriegel, H. P. (2004) *Probabilistic memory-based collaborative filtering*. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 1, pp. 56–69, 2004
Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G. (2005). *Improving recommendation lists through topic diversification*. Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 22–32. ACM, New York

ADDITIONAL READING SECTION

Adomavicius, G., Kwon, Y. O. (2007). *New recommendation techniques for multicriteria rating systems*, Intelligent Systems, IEEE 22 (2007), no. 3, 48–55.
Basu, C., Hirsh, H. Cohen, W. (1998). *Recommendation as classification: using social and content-based information in recommendation*. in Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98) (Madison, WI), American Association for Artificial Intelligence, pp. 714–720.
Beenen, G., Ling, K., Wang, X., Chang, K., Frankowski, D., Resnick, P. Kraut, R.(2004). *Using social psychology to motivate contributions to online communities*. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04) (Chicago), pp. 212–221.
Bridge, D., G¨oker, M., McGinty, L., Smyth, B. (2005).*Case-based recommender systems*. Knowledge Engineering Review 20 (2005), no. 3, 315–320.
Burke, R.(2002). *Hybrid recommender systems: survey and experiments.* User Modelling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370.
Chen, S. Y., Huang, P. R., Shih, Y. C., & Chang, L. P. (2013). *Investigation of multiple human factors in personalized learning*. Interactive Learning Environments, (ahead-of-print), 1-23.
Codina, V., & Ceccaroni, L. (2011, October). *Extending Recommendation Systems with Semantics and Context-Awareness: Pre-Filtering Algorithms*. InCCIA (pp. 81-90).
Felfernig, A., Gula, B. (2006). *An empirical study on consumer behavior in the interaction with knowledge-based recommender applications.* E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on (pp. 37-37). IEEE.
Felfernig, A., Teppan, E., Gula, B. (2007). *Knowledge-based recommender technologies for marketing and sales*. International Journal of Pattern Recognition and Artificial Intelligence, 21(02), 333-354.
Guo, Y., Cheng, X., Dong, D., Luo, C., & Wang, R. (2010, August). *An Improved Collaborative Filtering Algorithm Based on Trust in E-Commerce Recommendation Systems*. In Management and Service Science (MASS), 2010 International Conference on (pp. 1-4). IEEE.

Hammar, M., Karlsson, R., & Nilsson, B. J. (2013, October). *Using maximum coverage to optimize recommendation systems in e-commerce*. In Proceedings of the 7th ACM conference on Recommender systems (pp. 265-272). ACM.
Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). *An algorithmic framework for performing collaborative filtering*. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237). ACM.
Karypis, G. (2000). *Evaluation of item-based top-N recommendation algorithms*. Technical Report 00-046, University of Minnesota, Department of Computer Science.
Keikha, F., Fathian, M., Gholamian, M. R. (2013). *Comparison and Evaluation of Recommendation Systems on Social Networks*.
Konstan, J. A., & Riedl, J. (2012). *Recommender systems: from algorithms to user experience.* User Modeling and User-Adapted Interaction, 22(1-2), 101-123.
Lakiotaki, K., Matsatsinis, N. F., & Tsoukiàs, A. (2011). *Multicriteria user modeling in recommender systems*. Intelligent Systems, IEEE, 26(2), 64-76.
Lops, P., de Gemmis, M., Semeraro, G. (2011*). Content-based recommender systems: State of the art and trends*. In Recommender Systems Handbook (pp. 73-105). Springer US.
Mehta, B., Hofmann, T., Nejdl, W.(2007). *Robust collaborative filtering.* In: RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems, pp. 49–56. ACM, New York, NY, USA. DOI http://doi.acm.org/10.1145/1297231.1297240
Parameswaran, A., Venetis, P., & Garcia-Molina, H. (2011). *Recommendation systems with complex constraints: A courserank perspective.* Transactions on Information Systems (TOIS)
Parra, D.,Sahebi, S., (2013). *Recommender Systems: Sources of Knowledge and Evaluation Metrics,* Advanced Techniques in Web Intelligence – 2,pp. 149-175
Pham, X. H., Luong, T. N., & Jung, J. J. (2013). *An Black-Box Testing Approach on User Modeling in Practical Movie Recommendation Systems*. Computational Collective Intelligence. Technologies and Applications (pp. 72-79). Springer Berlin Heidelberg.
Pham. ,X.H., Jason,J.,J.(2013).P*reference-based user rating correction process for interactive recommendation systems*. Multimedia Tools and Applications, http://dx.doi.org/10.1007/s11042-012-1119-8,10.1007/s11042-012-1119-8,Springer US, pp. 119-132
Roitman, H., Carmel, D., Mass, Y., Eiron. I. (2013). *Modeling the uniqueness of the user preferences for recommendation systems*. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13). ACM, New York, NY, USA, 777-780. DOI=10.1145/2484028.2484102 http://doi.acm.org/10.1145/2484028.2484102
Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2000). *Application of dimensionality reduction in recommender system - A case study*. ACM WebKDD 2000 Web Mining for E-Commerce Workshop, Boston, MA.
Ricci, F., Rokach, L., Shapira, B. & Kantor, P.B. (Eds.) (2011). *Recommender Systems Handbook.* Springer 2011 ISBN 978-0-387-85819-7

KEY TERMS & DEFINITIONS

Recommender Systems – Software systems designed to filter data information and based on this predict the ratings or preferences for particular users in relation to items.
Recommendation Filtering Techniques- Techniques that are used to filter the data in order to make the data compatible to the standard RS model which includes the three main concepts user-items-ratings. Collaborative Filtering, Content Based Filtering and Hybrid filtering are the most known techniques. The selection of a filtering technique depends on the type of data which will be filtered.
Recommendation Frameworks – Set of layered functions that are available to software developers as tools for the development of recommendation systems.

Recommendation Systems Evaluation Metrics – Metrics used for the evaluation of recommendation systems algorithms. Metrics can be defined based on pure mathematic formulas but also based on objective opinions of end users.
Similarity Distance – The distance between preference attributes within a geometric vector based space.
Multi Criteria Recommender Systems - Recommender Systems that incorporate preference information upon multiple criteria.