



Hybrid Course Recommendation System Design for a Real-Time Student Automation Application

Alper Arık^{1*}, Savaş Okyay², Nihat Adar³

^{1*} Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Department of Computer Engineering, Eskişehir, Turkey (ORCID: 0000-0002-6542-6286), alperarik26@gmail.com

² Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Department of Computer Engineering, Eskişehir, Turkey (ORCID: 0000-0003-3955-6324), osavas@ogu.edu.tr

³ Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Department of Computer Engineering, Eskişehir, Turkey (ORCID: 0000-0002-0555-0701), nadar@ogu.edu.tr

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(DOI: 10.31590/ejosat.944596)

ATIF/REFERENCE: Arık, A., Okyay, S., & Adar, N. (2021). Hybrid Course Recommendation System Design for a Real-Time Student Automation Application. *European Journal of Science and Technology*, (26), 85-90.

Abstract

Recommender systems provide personalized suggestions by processing user and item information and interactions. Personalized product recommendations make it easier for users to access products that interest them. Course recommendation systems, on the other hand, aim to guide students to fields of interest in which they can succeed. On e-learning sites, there are many courses and students from different fields. Also, students can select courses from other than the fields they are studying. However, students in educational institutions must follow a curriculum. Since each educational institution has distinct constraints on course selection, a specific approach to the problem is required to develop a course recommender system. Due to the restrictive nature of the problem, developing a recommendation system for institutions is considered challenging. Therefore, students consult a faculty member when selecting a course for enrollment. In this study, a hybrid recommender system is proposed using student and course information with collaborative filtering and content-based filtering models. The proposed system provides consistent recommendations by using explicit and implicit data, without predefined association rules. The collaborative filtering algorithms use grades as rating values. The content-based filtering algorithms utilize text-based information about students and courses by converting them into feature vectors using natural language processing methods. In the combination phase of the hybrid recommender system, only one of the collaborative filtering and one of the content-based filtering models are used with different ensembling methods. It is found that the suggested hybrid recommender system can achieve outperforming results for all evaluation metrics. The results show the values of the rank-aware metrics Precision@N, AP@N, mAP@N, and NDCG@N for the individual models and the hybrid models with different combinations. In particular, for content-based filtering with Bayesian personalized ranking, the hybrid model performs better than any algorithm in practice.

Keywords: Course Recommendation System, Hybrid Recommender System, Collaborative Filtering, Content-Based Filtering.

Gerçek Zamanlı Öğrenci Otomasyon Uygulaması için Hibrit Ders Öneri Sistemi Tasarımı

Öz

Öneri sistemleri, kişi ve öğe bilgilerini kullanarak ve birbirleriyle olan etkileşimlerini işleyerek kullanıcılara göre özelleştirilmiş öneriler sunmaktadır. Kişiselleştirilmiş ürün önerileri, kullanıcıların ilgilerini çeken ürünlere erişmelerini kolaylaştırmaktadır. Ders öneri sistemleri ise öğrencileri ilgilendikleri ve başarılı olabilecekleri alanlara yönlendirmeyi amaçlamaktadır. E-öğrenme sitelerinde farklı disiplinlerden çok sayıda kurs ve öğrenci bulunmaktadır. Bu durumun yanı sıra, öğrenciler eğitim aldıkları disiplinler dışındaki diğer alanlardan ders alabilmektedir. Buna karşın, eğitim kurumlarındaki öğrenciler ise önceden belirlenmiş bir müfredatı takip etmek zorundadır. Her eğitim kurumu, ders seçimi için farklı kısıtlara sahip olduğundan, ders öneri sistemi geliştirme problemine özel bir

* Corresponding Author: alperarik26@gmail.com

yaklaşım gerekmektedir. Problemin sınırlayıcı doğası gereği, eğitim kurumları için ders öneri sistemi geliştirilmesi zorlu bir alan olarak kabul edilmektedir. Bu nedenle, öğrenciler kayıt için ders seçerken bir öğretim üyesine danışmaktadırlar. Bu çalışmada, öğrenci ve ders bilgileri ile işbirlikçi filtreleme ve içerik tabanlı filtreleme modelleri kullanan hibrit öneri sistemi önerilmiştir. Sistem, önceden tanımlanmış ilişkilendirme kuralları olmadan, belirgin ve dolaylı verileri kullanarak tutarlı öneriler sunmaktadır. İşbirlikçi filtreleme algoritması, öğrencilerin notlarını değerlendirme skoru olarak kullanmaktadır. İçerik tabanlı filtreleme algoritması ise öğrenciler ve dersler hakkındaki metin formatında bulunan bilgileri, doğal dil işleme yöntemleri ile özellik vektörlerine dönüştürerek kullanmaktadır. Hibrit öneri sistemini oluşturma işleminde, işbirlikçi filtreleme ve içerik tabanlı filtreleme modellerinden birer tane seçilmiş ve farklı birleştirme yöntemleri uygulanmıştır. Deneysel sonuçlarda ise, sunulan hibrit öneri sisteminin kendisini oluşturan algoritmalarından, tüm değerlendirme metriklerinde, daha başarılı sonuçlar elde edebildiği görülmüştür. Sonuç bölümünde, farklı kombinasyonlar ile oluşturulmuş hibrit modeller için Precision@N, AP@N, mAP@N ve NDCG@N sıralamaya duyarlı metrik değerleri gösterilmektedir. Özellikle, içerik tabanlı filtreleme ve Bayes kişiselleştirilmiş sıralamasından oluşan hibrit model, diğer tüm tekil modellerden daha iyi performans göstermiştir.

Anahtar Kelimeler: Ders Öneri Sistemi, Hibrit Öneri Sistemi, İşbirlikçi Filtreleme, İçerik Tabanlı Filtreleme.

1. Introduction

After the spread of the *Big Data* concept, a large amount of unrelated data has accumulated, and search engines struggle to provide personalized results for users (Grechanik et al., 2010). Recommendation systems (RS) are designed to find relationships between users and items and filter personalized items according to users' previous preferences (Lu, Wu, Mao, Wang, & Zhang, 2015). Recommendations that attract users' attention make it easier for users to find the item they are looking for. RS is mainly used in e-commerce and provides successful results (Schafer, Konstan, & Riedl, 2001). Course recommendation, on the other hand, has not been widely researched and is considered challenging because each student takes a different set of courses in different semesters. Course recommendation systems (CRS) can help students to find appropriate courses and reduce the time to discover interesting fields. Since there are a variety of elective courses in each semester, students may spend too much time exploring, and they might overlook some courses that match their interests (Bhumichitr, Channarukul, Saejiem, Jiamthaphaksin, & Nongpong, 2017). In addition to helping students, CRS ensures that course quotas are balanced by indirectly discovering courses that do not have many students.

In RS, there is not much feedback data that belongs to users in general. In most problems, the number of users and items is too large, but users directly interact with a small number of items compared to the total number, and this problem is called *data sparsity* (Guo, 2012). In another variation of this problem, users who have just started using the application that uses an RS have no interaction with the items. Therefore, user preferences are uncertain, and this situation is called the cold start problem. In cases where the *cold start* problem occurs, users have no rating scores, and collaborative filtering algorithms that generate recommendations based on rating scores become non-functional (Hernando, Bobadilla, Ortega, & Gutiérrez, 2017). In such cases, content-based filtering models that do not use rating scores are used. Structures formed by combining different types of RS methods to avoid sparsity and cold start problems are called hybrid recommender systems (Burke, 2002).

In the literature, there are many studies on CRS for e-learning platforms using data mining methods and RS. Castro et al. (Castro, Vellido, Nebot, & Mugica, 2007) develop a system that uses data mining methods to classify students according to their course success, navigation information on the course selection website, and interactions with courses. Carmona et al.

(Carmona, Castillo, & Millán, 2007) attempt to learn personal course preferences from student actions, such as taking the course and completing homework assignments using machine learning methods. Considering the studies conducted for e-learning platforms, the studies on CRS for educational institutions are very few. Booker (Booker, 2009) develops a CRS, which takes the keywords of the users' interests and their current GPA scores as input. The system uses a content-based filtering model and does not evaluate information about students' grades. Bydžovská (Bydžovská, 2016) develops data mining algorithms to find other students who have taken similar courses and grades. After finding similar students, the system recommends the courses that one student has taken and the other student has not. Another situation that occurs in the development of RS for educational institutions is that the features in the open-source datasets provided by e-learning sites do not match the characteristics of educational institution data, such as a number of users who have dropped out of the course, users do not have occupation/department information, users do not have a common curriculum, and exam evaluations may differ.

In this study, a hybrid CRS is designed for a real-time student automation application with student and course information in the ESOGU course management system. The proposed system provides stable and bias-free recommendations by using both explicit and implicit data coherently, without predefined association rules. We solved this design problem from a recommendation perspective and created a hybrid RS, which consists of collaborative filtering and content-based filtering models. Instead of using memory-based methods in collaborative filtering models, model-based matrix factorization methods are used so that the prediction success is high and scalable. In the content-based filtering model, TF-IDF feature vectors are computed using the information of users and items in text form with natural language processing (NLP) methods. In the hybrid RS creation step, models are combined with the use of majority voting and weighted average methods. The system uses grades as ratings with student and course information to prevent data sparsity and cold start, which are common RS problems.

2. Material and Method

2.1. Dataset and Preprocessing

The dataset used in this study consists of 966 students and 187 courses extracted from ESOGU Computer Engineering Department. Fifty of them are elective courses. The dataset consists of course characteristics, student characteristics, and grades for each course taken by the students. The dataset

comprises the grades of student-course pairs, includes data between the years 2000-2021 and has a sparsity of 80.74%. Data between 2000-2019 is used as the training set, and data from 2020 is used as the testing set. Data from students who started their studies in 2020 and have no counterpart in the training set are ignored. Similarly, courses that offered for the first time in 2020 are not included in the test set. Considering the time-series nature of the data and the size of the training set, hold-out cross-validation is applied to the test set.

Grades earned by students in courses are used as ratings. Table 1 shows the preset numerical values of letter grades between 0-4 and scaled values between 1-5. Grades for courses that students do not take are assigned a value of 0. Courses with DZ letter grades that indicated students do not pass due to absence and non-credit courses are not included in the train set. Students may make interactions such as failing a course or retaking the course to improve the letter grade. In this situation, there is more than one letter grade between the student and the course. To solve this problem, we aggregate all the grades that the student has achieved in a course with a weighted sum.

Table 1. The Grading Scale with Numerical Values

Letter Grade	Numerical Value	Scaled Value
AA	4.0	5.0
BA	3.5	4.5
BB	3.0	4.0
CB	2.5	3.5
CC	2.0	3.0
DC	1.5	2.5
DD	1.0	2.0
FF	0.0	1.0

2.2. Recommender System Models

2.2.1. Content Based Filtering

Content-based filtering depends only on the prior information of the user and the items, which makes this model robust and avoids the cold start problem. For textual properties of items, raw text is used to create item profiles and user profiles. The TF-IDF method from information retrieval converts unstructured text into a vector structure, where each word is represented by a position in the vector, and the value measures how relevant a particular word (unigram/bigram) is to a text (Ramos, 2003). Since all items are represented in the same vector space model, it is used to calculate the similarity between texts (Bagga & Baldwin, 1998).

The data in the text format of the courses are course information forms whose contents are as follows: *course name*, *purpose*, *content*, *outcomes*, *resources*, and *weekly schedule*. There are variants of these properties of the courses in Turkish and English. These Turkish and English properties of the courses are preprocessed separately using NLP methods, and their features are extracted. In preprocessing, the text data is converted to lowercase and tokenized, and later punctuation and stopwords are removed. Finally, the lemmas of the words are obtained using part-of-speech tags (POS), and the TF-IDF values are calculated based on the obtained features. User profiles are created from the TF-IDF vectors belonging to the students' previous course information. In the prediction stage, the cosine similarity between the user profiles was calculated. The models are combined into a hybrid model in the final step, and the similarity values of each model are scaled between 0-1 and

called the recommendation strength. The cosine similarity value in the content-based filtering model is between 0-1, so it is used as the recommendation strength without any scaling operation. Previous courses taken by the student are filtered, and courses belonging to similar students are presented as recommendations. Depending on the available student data, different approaches can be developed. For example, students with similar GPAs and students with similar demographic characteristics, etc. can be used to find similarities between students.

2.2.2. Collaborative Filtering

Collaborative filtering models can be divided into two main categories: memory-based methods and model-based methods. The memory-based method uses previous user interactions to compute similarities between users based on the items they have interacted with (user-based approach) or similarities between computer items based on the users who have interacted with them (item-based approach). The memory-based method is easy to implement, but usually does not scale well for many users (Xue et al., 2005). Model-based matrix factorization models compress the user-item matrix into a low-dimensional representation in terms of latent factors. An advantage of this approach is that instead of dealing with a high-dimensional matrix containing a large number of missing values, it deals with a much smaller matrix in low-dimensional space. This feature handles the sparsity of the original matrix better than memory-based models. Also, the comparison of similarity on the resulting matrix is much more scalable, especially for large sparse datasets. Matrix factorizations are widely used and have repeatedly shown better accuracy than other methods, such as nearest-neighbor methods, which belong to memory-based models (Bell & Koren, 2007).

In this study, matrix factorization methods, alternating least squares (ALS), and Bayesian personalized ranking (BPR) are used in the collaborative filtering model. In the ALS algorithm, first, the item matrix is fixed, and the user matrix is solved. Then for each iteration, the user matrix is fixed, and the item matrix is solved. At each step, the cost function can either decrease or remain unchanged but never increase. Alternating between the two steps guarantees that the cost function will decrease until convergence. Similar to gradient descent optimization, it is guaranteed to converge only to a local minimum and ultimately depends on the initial values for the user matrix or the item matrix. Unlike gradient descent, ALS does not need to loop over each training case, which would be impractical, and can use parallelization (Koren, Bell, & Volinsky, 2009). ALS does not optimize its model parameters directly for ranking. Instead, it optimizes to predict whether an item will be selected by a user or not. However, the BPR optimization criterion involves pairs of items to create a personalized ranking function for each user. For this process, BPR approximates the area under curve (AUC) metric. Instead of calculating item scores, the user's log probabilities of selecting items are calculated, and a logistic function is used (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012).

The numerical equivalent of students' letter grades is evaluated as implicit feedback and used as the rating value in the model. The courses that students do not take are expressed as 0, and the range of numerical letter grades is shifted from 0-4 to 1-5. The prediction scores are scaled to the range 0-1 to make the collaborative filtering model compatible with other models and referred to as recommendation strength as in the content-based

filtering model. During the training phase of the matrix factorization models, the hyperparameters such as latent feature factor, regularization and learning rate for BPR are tuned in different iterations. Root mean squared error (RMSE) is used for ALS, and AUC is used for BPR as the loss function. Item recommendations are generated using the dot product of the user-latent and item-latent matrices obtained as a result of training.

2.2.3. Hybrid Recommendation System

Hybrid recommendation systems combine two or more recommender models with different ensembling methods to overcome two traditional and top problems; cold-start and data sparsity (Çano & Morisio, 2017). In the hybrid modeling phase, the results of the recommender models are combined. Thanks to the hybrid model, the goal is to use the power of the community instead of using an individual model. In the popular Netflix competition, the winner used an ensembling method to implement a powerful collaborative filtering algorithm (Bell & Koren, 2007). In this study, ensembling is performed using majority voting and weighted average method. Majority voting is used for classification problems and averaging is used for regression problems in machine learning (Caruana, Niculescu-Mizil, Crew, & Ksikes, 2004). In RS, the recommendation strength can be considered as a continuous value as in regression and the hit rate as a discrete value as in classification. In majority voting, each model makes a prediction for each item in the test set, and the final prediction is the one that receives more than half of the votes, which is the model's recommendation. If none of the predictions receives more than half of the votes, the ensemble method cannot make a stable prediction for any item. In such a situation, the prediction with the most votes, even if it receives less than half of the votes, becomes the final prediction. This method is called plurality voting (Randhawa, Loo, Seera, Lim, & Nandi, 2018). In simple averaging, the average predictions for each item in the test set are calculated based on the strength of the recommendation. This method reduces overfitting and creates a smoother model. The weighted average is a modified version of simple averaging in which the recommendation strength of each model is multiplied by the coefficients alpha, beta, and gamma etc., by a sum of 1, and then their average is calculated (Sewell, 2008).

The proposed system uses students' grades as explicit data and course information as implicit data. These data types are used in collaborative filtering and content-based filtering,

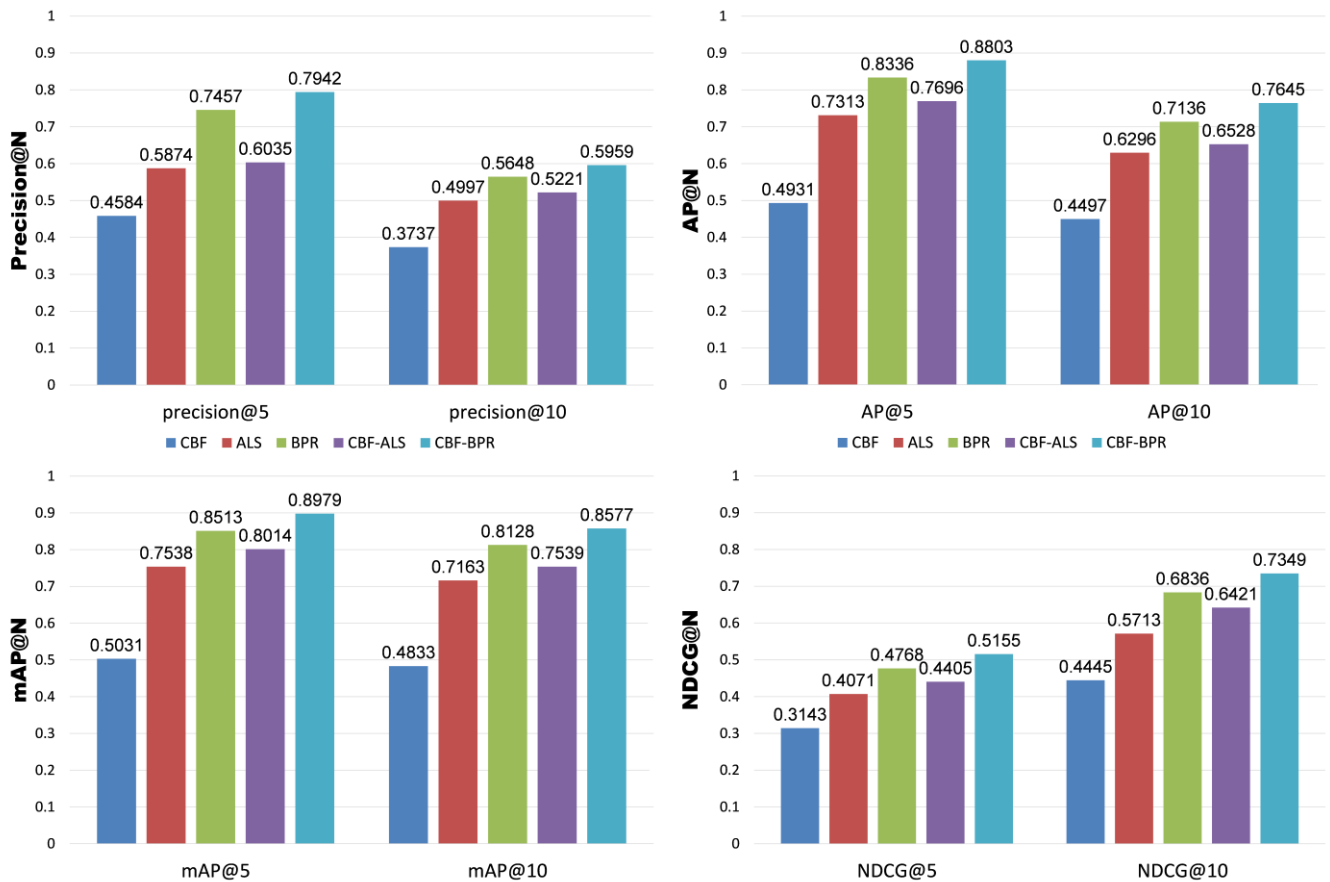
respectively. The utilization of both data types enables the proposed system to recommend courses more stable. Moreover, the system does not use predefined association rules and becomes free from general patterns such as trend data.

2.3. Evaluation

RS has a specific and primary concern; relevant items must rank high in the recommendation list. For this reason, rank-aware metrics are chosen. Rank-aware metrics consist of binary relevance-based metrics, which care about whether an item is good or not in a binary sense, and utility-based metrics, which care about the relative goodness of an item. Rank-aware metrics evaluate the entire list of recommended items up to a certain threshold N . The decision support metric Precision@ N computes the precision value for a subset of the N recommendations. The drawback of this metric is that it does not consider the recommendation list as an ordered list. Precision@ N considers the entire list as a set of items and treats all errors in the recommendation list equally. The goal is to weigh the errors at the top of the list heavily and then gradually decrease the importance of the errors as while moving to the lower items in a list. The metric of average precision (AP@ N) attempts to approximate this sliding weighting scale. It uses Precision@ N in successive sublists. Finally, the mean of the average precision for all users is calculated, and the mean average precision (mAP@ N) is obtained (McFee & Lanckriet, 2010).

The cumulative gain (CG) provides a basic measure of the accumulation of graded relevance and does not take into account the position of the items in the recommendation list. Discounted cumulative gain (DCG) adds a logarithmic reduction factor to penalize relevance scores proportional to the item's position. Users of RS are recommended a variable number of relevant items. Consequently, the DCG measure is not comparable between different systems and needs to be normalized. To overcome this problem, the ideal ranking for a user is determined and then used as the ideal discounted cumulative gain (IDCG) for normalization (Busa-Fekete, Szarvas, Elteto, & Kégl, 2012).

Figure 1. Model Results



3. Results and Discussion

In this study, the proposed method first takes user-item pairs for each user and for each item that the user interacted with in the test set. Then, M randomly selected items that the user never interacted with are taken from the training set. Items that have not been interacted with are assumed to be irrelevant to the user, but this is not necessarily true since the user may not be aware of these non-interacted items. Therefore, the recommendation model generates a ranked list of recommendations consisting of a set of one interacted item and M non-interacted items. Finally, the evaluation metrics for the ranked list are computed, and the results are aggregated for each user and interacted item in the test set pairs. Rank-aware metrics; Precision@N, AP@N, mAP@N, and NDCG@N results of all individual models and all hybrid models in different combinations are shown in Figure 1. Individual models consist of content-based filtering model and collaborative filtering model as ALS and BPR respectively. Hybrid models are CBF-ALS and CBF-BPR combinations that use majority voting as the ensembling method. In all rank-aware metrics, hybrid models are more successful than their individual models. The CBF-BPR hybrid model is more successful than the CBF-ALS hybrid model in terms of all evaluation metrics and becomes the best model overall. According to the difference between the top-5 and top-10 results, there is a decrease in Precision@N, AP@N, and mAP@N metrics and an increase in NDCG@N metric. Precision-based metrics are probably not relevant as more items are evaluated and scores have decreased. On the other hand, scores increase due to the CG, when the number of evaluated items in the NDCG@N metric increases.

Finally, for the precision-based metrics from Precision@N to mAP@N, an increase in top-5 and top-10 scores is observed, as expected, because the scores are evaluated cumulatively.

4. Conclusion

In this study, we designed a solution from the recommendation perspective for the course selection problem. The proposed system evaluates the implicit data of the students and integrates the attributes of the items into the system. We created collaborative filtering and content-based filtering models. Majority voting and weighted average are used as ensembling methods to combine individual models into a hybrid model. Suggested hybrid RS uses students' grades as explicit data and course information as implicit data. Through this feature, the hybrid RS recommends courses without bias. Furthermore, the system does not use generalized behavioral patterns and becomes free from common information such as course popularity. Finally, we observed that the succession rate of the hybrid models are increased. RMSE metric is used in training the ALS model and AUC metric is used as loss function in training the BPR model. In general, we performed the evaluation of all models with the test set using the rank-aware metrics Precision@N, AP@N, mAP@N, and NDCG@N. The CBF-BPR hybrid model, which uses majority voting as the ensembling method, gives the highest result in each rank-aware metric as the most successful model. As a result, we built a hybrid recommender system that can overcome sparsity, cold start, and scalability problems.

For future work, more detailed models using course evaluation questionnaires, various grade scaling methods, and course success thresholds can be used. Finally, the hybrid RS model can be combined with different ensembling methods, such as algorithm prioritization with cascading and dedication of certain items in the output of the recommender with business rules.

References

- Bagga, A., & Baldwin, B. (1998). *Entity-Based Cross-Document Coreferencing Using the Vector Space Model*. Paper presented at the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1.
- Bell, R. M., & Koren, Y. (2007). Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2), 75-79.
- Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthaphaksin, R., & Nongpong, K. (2017, 12-14 July 2017). *Recommender Systems for university elective course recommendation*. Paper presented at the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE).
- Booker, Q. E. (2009). A student program recommendation system prototype. *Issues in Information Systems*, 544-551.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331-370.
- Busa-Fekete, R., Szarvas, G., Elteto, T., & Kégl, B. (2012). *An apple-to-apple comparison of learning-to-rank algorithms in terms of normalized discounted cumulative gain*. Paper presented at the ECAI 2012-20th European Conference on Artificial Intelligence: Preference Learning: Problems and Applications in AI Workshop.
- Bydžovská, H. (2016). Course Enrollment Recommender System. *International Educational Data Mining Society*.
- Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), 1487-1524.
- Carmona, C., Castillo, G., & Millán, E. (2007). *Discovering student preferences in e-learning*. Paper presented at the Proceedings of the international workshop on applying data mining in e-learning.
- Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). *Ensemble selection from libraries of models*. Paper presented at the Proceedings of the twenty-first international conference on Machine learning.
- Castro, F., Vellido, A., Nebot, A., & 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.
- Grechanik, M., Fu, C., Xie, Q., McMillan, C., Poshyvanyk, D., & Cumby, C. (2010). *A search engine for finding highly relevant applications*. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1.
- Guo, G. (2012). *Resolving data sparsity and cold start in recommender systems*. Paper presented at the International Conference on User Modeling, Adaptation, and Personalization.
- Hernando, A., Bobadilla, J., Ortega, F., & Gutiérrez, A. (2017). A probabilistic model for recommending to new cold-start non-registered users. *Information Sciences*, 376, 216-232.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.
- Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32.
- McFee, B., & Lanckriet, G. R. (2010). *Metric learning to rank*. Paper presented at the ICML.
- Ramos, J. (2003). *Using tf-idf to determine word relevance in document queries*. Paper presented at the Proceedings of the first instructional conference on machine learning.
- Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access*, 6, 14277-14284.
- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.
- Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery*, 5(1-2), 115-153.
- Sewell, M. (2008). Ensemble learning. *RN*, 11(02), 1-34.
- Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., & Chen, Z. (2005). *Scalable collaborative filtering using cluster-based smoothing*. Paper presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.