

Personalized Learning in K-12 Education: Exploring Weak-Labels for a Random Forest-based Collaborative Filtering Approach

Pedro Ilídio^{1,2,*}, Alireza Gharahighehi^{1,2}, Felipe Kenji Nakano^{1,2} and Celine Vens^{1,2}

¹*KU Leuven, Campus Kulak, Department of Public Health and Primary Care, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium*

²*Itec, imec research group at KU Leuven, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium*

Abstract

Education, a cornerstone of human development, increasingly leverages digital learning tools, generating valuable data from student interactions. This data can enhance learning efficiency through adaptive and personalized systems, moving beyond the traditional "one-size-fits-all" model. Recommendation systems learn user profiles to suggest relevant items and personalize students' learning experiences. In the context of implicit feedback, binary interactions are used in weak-label learning, where negative label annotations are unreliable. This paper proposes a weak-label learning method for recommending learning materials and trajectories, combining local and global Random Forests in a multi-step collaborative filtering process. The proposed approach is named PentaForest, and outperforms other popular collaborative filtering methods in terms of NDCG and recall.

Keywords

Random Forest, weak-label learning, k-12 education, educational recommendation, collaborative filtering

1. Introduction

One of the most important aspects of human development and sustainable development goals is education. In the age of digitization, the use of digital learning tools has become more popular among students, either as a primary means of completing learning assignments or as an auxiliary learning environment. This facilitates the creation of data based on learners' behaviors and interactions with these learning platforms. Such data can be used to provide data-driven interventions, making learning more efficient and effective. There is a growing interest in adaptive and personalized learning systems due to their presumed benefits on cognitive and non-cognitive learning outcomes [2, 3]. Education is continuously moving from the classic "one-size-fits-all" model to more adaptive and personalized learning approaches. As learners have different needs and preferences, they can be served accordingly.

Recommendation systems are machine learning methods that learn users' profiles based on their previous interactions and recommend items that best fit these profiles. These systems

ALL'24: Workshop on Adaptive Lifelong Learning, July 08–12, 2024, Recife, Brazil [1]

*Corresponding author.

✉ pedro.ilidio@kuleuven.be (P. Ilídio); alireza.gharahighehi@kuleuven.be (A. Gharahighehi); felipekenji.nakano@kuleuven.be (F. K. Nakano); celine.vens@kuleuven.be (C. Vens)

🆔 0000-0002-5750-0204 (P. Ilídio); 0000-0003-1453-1155 (A. Gharahighehi); 0000-0002-4884-9420 (F. K. Nakano); 0000-0003-0983-256X (C. Vens)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

are generally categorized into two main types: content-based filtering and collaborative filtering. The former aims to match item features with user profiles, recommending items whose features best match the profiles, while the latter models user preferences and needs based on collaborative information between users and items—i.e., their interactions—to recommend items. User feedback on items is usually implicit, meaning that, in most of the cases, we do not have explicit ratings from users. In such cases, binary feedback (user interaction with an item) is the only available signal to learn user profiles. This setting is known as one-class collaborative filtering and is also referred to as positive-unlabeled (PU) learning or weak-label learning in machine learning. In this context, the given labels are all positive and missing labels are either from the negative class (i.e., the user observed the item but deliberately did not interact with it) or are missing positive labels (i.e., the user did not observe the item, otherwise (s)he would have interacted with it). Furthermore, because the given labels are based solely on simple interactions between users and items, they may be unreliable and weak positive labels. In this paper, we propose a weak-label learning method to recommend learning materials and learning trajectories to students based on their past interactions within the system. We combine local and global Random Forests in a multi-step procedure for collaborative filtering, utilizing self-learned label probabilities to address label unreliability. The resulting procedure is named PentaForest.

The following sections are organized as follows. Section 2 provides an overview of related studies. Section 3 then defines the specific learning problem being approached, and Section 4 presents our method proposal. We present our experimental setup in Section 5, and experimental results are provided in Section 6. Finally, Section 7 concludes our work and proposes future research directions.

2. Literature Review

We propose using ensembles of randomized decision trees, called *Random Forests* [4], to perform collaborative filtering in weakly-supervised settings. Decision tree-based methods for weakly-supervised tasks have not received much attention in recent years [5]. Two general approaches are usually proposed: i) impute new labels in a self-supervised manner; or ii) consider the structure of the feature space when growing each tree. Here, we focus on the first approach, where the label matrix is either completed to yield a dense representation or new positive annotations are added before training the final estimator. In this context, Tanha et al. [6] proposed an iterative approach, where the most confident predictions are imputed before building the next tree. Wang et al. [7] focused specifically on the problem of weak-labels, where only negative annotations are unreliable. Their proposal is based on a deep forest model, in which multiple decision forests are trained sequentially and the predicted probabilities at each layer are appended as new features for the next one. The authors adapted this procedure to weak-labels by performing label imputation after each layer of the deep forest. We extended this idea and addressed limitations of the original proposal in a recent work [8]. In the context of PU interaction prediction, Pliakos and Vens [9] used Neighborhood-Regularized Logistic Matrix Factorization to complete the label annotations, converting the binary label matrix to a dense representation. This representation was then used to build a global multi-output forest to serve as the final estimator. *Global* estimators consider both item-related *and* user-related

information during training [9]. In contrast, *local* models take either users *or* items as input instances, predicting its interactions as multiple outputs [9]. Gharahighehi et al. [10] proposed a two-step approach to address the cold-start problem in a PU learning context. First, the interaction matrix between users and warm items is reconstructed using SLIM [11]. Then, an inductive multi-target regressor is trained on this reconstructed interaction matrix to predict interactions for new items that enter the system. In the context of Massive Open Online Course (MOOC) recommendations, Gharahighehi et al. [12] considered censored time-to-event data (time to dropout from MOOCs) as weak-labels and extended Bayesian Personalized Ranking [13], which is a learning-to-rank collaborative filtering approach, to incorporate these weak-labels in training the model.

In the field of recommendation systems, Random Forests were employed by Li et al. [14] as a dimensionality reduction strategy, preprocessing the data before employing similarity-based collaborative filtering. Panagiotakis et al. [15], on the other hand, completed the label matrix with Synthetic Coordinate Recommendations and then trained a local Random Forest as an item recommender given a user as input. However, in both cases, Random Forests are employed to explore side-information or context-information on the problem, in contrast to the current scenario where only interactions are utilized.

We combine multi-output local forests to complete the label matrix, and then leverage the completed annotations to train a global single output forest. We now define the learning problem under study (Section 3), and a detailed description of our method is presented in Section 4.

3. Problem Definition

Let \mathcal{U} be a set of users $\{u_1, u_2, \dots, u_n\}$ and \mathcal{I} be a set of items $\{i_1, i_2, \dots, i_m\}$. The $n \times m$ binary matrix $Y = (Y_{lp})$ then represents known interactions between the user u_l and item i_p . In this matrix, 1 is assumed to be a confirmed interaction. 0-valued entries, however, can be either non-occurring relationships or unannotated positive values, which characterizes the weak-label scenario. Each item is considered a different label and each user is considered a different sample.

Our goal is to indicate k new items for each user u_l in \mathcal{U} , representing the annotations that are most-likely to be missing or interactions that are most-likely to occur in the future. The set of k indicated items for u_l is called the *recommendations* for the user u_l . To generate recommendations, we only receive the label matrix Y , no side-information is assumed. Having only Y characterizes *collaborative filtering* techniques based on implicit feedback.

In the present setting, users represent students in an online learning platform. For items, two scenarios are separately explored:

- **Learning materials:** items are learning activities available at the platform;
- **Learning trajectories:** items are sets of learning materials. Each trajectory represents a path of materials manually defined by a teacher to be followed by the students.

We say a user interacted with a learning material if the material was accessed by the user, independently of the activity being concluded or not, and independently of the number of accesses. For learning trajectories, a positive value indicates the trajectory was concluded.

Weak-labels arise from the user not knowing the item, or having not interacted with it yet at the time that the dataset was built.

4. Proposed Method: PentaForest

The proposed procedure¹ employs five Random Forest (RF) [4] models to perform collaborative filtering. Usually, training each RF requires a feature matrix describing the input samples and a label matrix to be modeled. In this case, however, we use Y as both the feature and the label matrices. The algorithm is divided into three main steps:

1. Train primary item and user recommenders

- **Forest 1:** In the first step, a RF is trained to predict the probabilities of a given user to interact with each of the items. We call this RF an "item recommender".
- **Forest 2:** Also in this step, another RF is built to solve the transposed problem: it predicts the probabilities of a given item to interact with each of the users. It is called an "user recommender".

2. Train secondary item and user recommenders

- **Forest 3:** In the second step, the probabilities predicted by the item recommender are used as targets for a second user recommender.
- **Forest 4:** We also train a second item recommender using the probabilities predicted by the user recommender.

3. Train single output predictor

- In the third step, we average the probabilities predicted by the secondary user recommender and the secondary item recommender.
- **Forest 5:** The averaged probabilities are used as targets for a fifth and final RF. This RF considers concatenated user-item interaction profiles as input, and predicts the single probability of interaction between the given pair. Its predictions are the final probabilities we output.

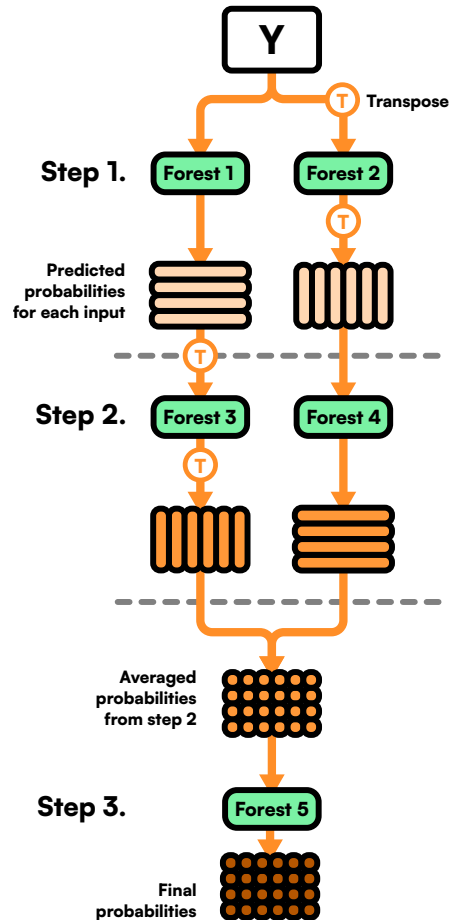


Figure 1: The PentaForest procedure.

¹The implementation of the PentaForest algorithm is available at <https://itec.kuleuven-kulak.be/supporting-material/>.

Algorithm 1: PentaForest: Random Forests for Collaborative Filtering

Input: User-item interaction matrix Y

Output: Completed matrix \tilde{Y}

Step 1: Train primary recommenders

$\tilde{Y}_{\text{items}, 1} \leftarrow \text{LocalRandomForest.fit_predict_proba}(Y, Y)$

$\tilde{Y}_{\text{users}, 1}^T \leftarrow \text{LocalRandomForest.fit_predict_proba}(Y^T, Y^T)$

Step 2: Train secondary recommenders

$\tilde{Y}_{\text{users}, 2}^T \leftarrow \text{LocalRandomForest.fit_predict_proba}(Y^T, Y_{\text{items}, 1}^T)$

$\tilde{Y}_{\text{items}, 2} \leftarrow \text{LocalRandomForest.fit_predict_proba}(Y, Y_{\text{users}, 1})$

Step 3: Final prediction

$\tilde{Y}_{\text{avg}} \leftarrow \text{average}(\tilde{Y}_{\text{users}, 2}, \tilde{Y}_{\text{items}, 2})$

$\tilde{Y}_{\text{final}} \leftarrow \text{GlobalRandomForest.fit_predict_proba}(Y, Y^T, \tilde{Y}_{\text{avg}})$

Return \tilde{Y}_{final}

We note that the concatenations in the third step are not performed as such. Instead, the Bipartite Global Single Output procedure [16] is used to generate the forests more efficiently from both user and item "feature matrices" directly. In each step, the user feature matrix is always the original binary label matrix itself (Y), and the transposed label matrix (Y^T) is taken as the item feature matrix. We call *local* the forests that utilize either users *or* items as input samples, predicting multiple outputs for each of them [9]. The *global* forest, on the other hand, considers both at the same time [9]. The whole procedure is summarized in algorithm 1 and illustrated in Figure 1.

The reason for using multiple steps of reconstruction is to generate several diverse models. We then encourage the different types of learned information to be exchanged between the forests, controlling overfitting and improving their ability to generalize. Further discussion is presented by Section 6. We also provide four reasons for utilizing the original label matrix as features instead of its completed version:

1. the original (confirmed) positive annotations are prioritized, being used to define the decision boundaries;
2. the lower cardinality of the binary labels reduces the tendency of one forest to overfit the predictions of the previous;
3. the lower cardinality also allows for faster training of the forests;
4. relying on the original annotations makes the model more easily interpretable, allowing us to look only at the final estimator's structure to gain insights on the learning task.

5. Dataset and Experimental Setup

We used two datasets from an educational K-12 platform in Belgium. The first dataset contains students' interactions with learning materials, while the second dataset includes students'

Table 1
Datasets Description

	Learning material dataset		Learning trajectory dataset	
	Original	preprocessed	Original	preprocessed
# users	8573	1171	9211	1183
# Items	3306	1057	1234	577
Density	0.2%	1.8%	0.2%	1.5%

interactions with learning trajectories, each comprising a series of learning materials defined by teachers. We excluded students and materials with fewer than 10 interactions from the first dataset and students and tracks with fewer than 5 interactions from the second dataset. This is done to ensure that a reasonable amount of information remains on the training set after masking the interactions to be used for scoring. Table 1 provides a description of the two datasets after preprocessing. To form our training, validation, and test sets, we kept one interaction per user for the test set, one interaction per user for the validation set, and all remaining interactions for the training set. This way of splitting ensures that the same set of users and items appears in the training, test, and validation sets.

For evaluation, we applied two measures: normalized discounted cumulative gain (NDCG) (Eq. 1) and recall (Eq. 2).

$$DCG_u = \sum_{i=1}^{|R_u|} \frac{2^{rel_i} - 1}{\log_2(i + 1)},$$

$$IDCG_u = \sum_{i=1}^{|R_{idl_u}|} \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (1)$$

$$NDCG = \frac{1}{|U|} \sum_{u \in U} \frac{DCG_u}{IDCG_u}.$$

$$Recall = \frac{1}{|U|} \sum_{u \in U} \frac{|T_u \cap R_u|}{|T_u|}. \quad (2)$$

where U is the set of users, T_u is the test items for user u , R_u is the recommendation list for user u , rel_i is the real rating value of the i^{th} item in R_u , $IDCG$ is the ideal DCG value, and R_{idl_u} is the ideal recommendation list for user u , that one can create based on the ground truth. Therefore, the DCG value for each user is normalized with the ideal value ($IDCG$) to get the $NDCG$ value for that user.

As competing methods, we included five approaches:

- **UKNN**: user-based (UKNN) collaborative filtering (CF) [17] is a memory-based CF methods that impute missing interactions between users and items based on the interactions of neighbor users.
- **IKNN**: item-based (UKNN) collaborative filtering (CF) [18] is a memory-based CF methods that impute missing interactions between users and items based on the interactions of neighbor items.

- **WRMF**: weighted regularized matrix factorization (WRMF) [19] is a model-based CF method that utilizes the alternating-least-squares optimization algorithm to learn its parameters.
- **EASE**: Embarrassingly Shallow Autoencoders (EASE) [20] is a linear collaborative filtering model for implicit feedback datasets based on shallow autoencoders [21].
- **MVAE**: multi-variational autoencoders (MVAE) [22] is a CF-based recommender system for implicit feedback, based on variational autoencoders, with the main assumption that user interactions follow a multinomial distribution.

We selected these methods because they demonstrated high performance in collaborative filtering tasks, as reported in the award-winning paper by Dacrema et al. [23], which showed that memory-based approaches (UKNN and IKNN) and MVAE outperform recent complex deep neural network-based approaches .

As for the hyperparameters of our method, each forest was composed of 1000 trees. The forests employed bootstrapping to resample the set of input instances, and were grown until their maximum depth. The remaining hyperparameters were left to their default values in the scikit-learn package. For instance, the objective criterion was set to the mean squared error, and no feature sampling was performed.

6. Results and Discussion

The results of applying the proposed method and the competing approaches are reported in Tables 2 and 3 for the learning material and trajectory recommendation tasks, respectively. The proposed PentaForest method is shown to clearly outperform the other competing approaches in both tasks. In terms of recall, PentaForest is especially superior when fewer recommendations are selected, which represents the most difficult tasks under study (see top 3 in Table 2 and Table 3). Regarding NDCG, the superiority of our method is evident and consistent across all scenarios. Again, this suggests that the proposed technique is especially proficient for fewer recommendations, for the reason that NDCG prioritizes the recommendations predicted to be most likely. As such, NDCG is less sensitive to the number of recommendations we select for evaluation.

The promising results corroborate the hypothesized benefits of weak-label techniques in recommendation contexts. Note that using the label matrix as features for the Random Forests means that each tree will cluster instances with similar labels. When we use self-learned label probabilities to train a new forest, we are instructing this forest to cluster labels that are *expected* to be similar, even if not similar from the original label annotations. This is what mitigates the effect of uncertain annotations.

Furthermore, we argue that our model proficiency is in great part due to the diversity of the forests employed. This diversity might not be apparent at a first glance, since all the component models are based on the same Random Forest algorithm. Notwithstanding, notice that the secondary user recommender is trained on the outputs of the primary item recommender, and vice-versa. This compels the secondary estimators to learn a representation of the problem that is different from the primary models. That is, the predictions made based on the user-wise information now need to be explained from the item information alone, and vice-versa. Similarly,

Table 2

Results for learning material recommendation

	top 3		top 5		top 10	
	NDCG	Recall	NDCG	Recall	NDCG	Recall
UKNN	0.709	0.746	0.751	0.837	0.782	0.924
IKNN	0.717	0.756	0.760	0.846	0.790	0.933
WRMF	0.668	0.726	0.729	0.811	0.750	0.873
EASE	0.710	0.746	0.752	0.836	0.783	0.923
MVAE	0.792	0.830	0.826	0.915	0.840	0.977
PentaForest	0.885	0.900	0.909	0.952	0.921	0.987

Table 3

Results for learning trajectory recommendation

	top 3		top 5		top 10	
	NDCG	Recall	NDCG	Recall	NDCG	Recall
UKNN	0.814	0.854	0.847	0.922	0.862	0.963
IKNN	0.819	0.860	0.852	0.927	0.866	0.967
WRMF	0.582	0.609	0.616	0.680	0.634	0.731
EASE	0.812	0.850	0.847	0.923	0.861	0.963
MVAE	0.840	0.881	0.882	0.968	0.889	0.997
PentaForest	0.931	0.952	0.954	0.979	0.950	0.995

the representation learned at the final step by the global Random Forest is also different. This forest will induce a biclustering of the interaction matrix, grouping interactions that are similar both in terms of user and item profiles. This differs from the local forests used in the first two steps, that only consider either user-user or item-item label similarity.

Finally, using a global forest that receives the original binary labels yields a crucial advantage for the deployment of these systems: it enables acknowledging new interactions, that were added to the dataset after training the models. This is due to the fact that the final estimator only needs the user and item interaction profiles to estimate new interaction probabilities. As such, updated interaction profiles could be provided to obtain updated recommendations without rebuilding the models. Even interactions between unseen items and unseen users can be inferred, based on their interactions with the users and items in the training set.

The global forest also allows estimating probabilities only for a subset of items and users. This can be used, for example, to provide recommendations within a user-specified category of activities, without generating probabilities for all activities in the training set.

An educational system could also benefit from the known transparency of forest estimators. It could, for instance, prioritize items with higher importance when making recommendations, instead of only relying on the predicted probabilities. This could enable the system to discover new interests of a user, making recommendations that are not necessarily similar to the user's previous interactions, but are useful for profiling the user's preference.

In summary, the trained final forest of a PentaForest model can be employed to create highly adaptive recommendation systems, for a dynamic and personalized learning experience.

7. Conclusion

In this paper we proposed PentaForest, a new collaborative filtering approach for recommending learning material and learning trajectories to students in a web-based educational platform. Our method combines local and global Random Forests in a way to acknowledge the weakly-supervised nature of our problem. Our results suggest state-of-the-art performance in our current learning settings, prompting future studies to further investigate decision forests for recommendation tasks in general. In subsequent work, we would like to explore semi-supervised split evaluation metrics, since they have been shown to be useful for other PU-learning scenarios [24, 16]. We would also like to evaluate the potential of deeper ensembles of decision forests as recommender systems, adapting deep forests presented by Wang et al. [7], Ilidio et al. [8]. In contrast, shallower versions of our estimators might also be an interesting topic of study in a broader variety of problems, performing more detailed ablation analysis to elucidate the effectiveness of the reconstruction steps. Furthermore, exploring the explainability of decision trees could be a valuable method for selecting the most informative activities to recommend when profiling new users. Finally, the use of side-features is also a topic for further exploration, as a way of mitigating the cold start problem or possibly improving transductive predictions.

Acknowledgments

The work received funding from the Flemish Government (AI Research Program). The authors would like to also thank FWO, 1235924N.

References

- [1] A. Gharahighehi, R. Van Schoors, P. Topali, J. Ooge, Adaptive lifelong learning (all), in: *International Conference on Artificial Intelligence in Education*, Springer, 2024, pp. 452–459.
- [2] X. Zhai, X. Chu, C. S. Chai, M. S. Y. Jong, A. Istenic, M. Spector, J.-B. Liu, J. Yuan, Y. Li, A review of artificial intelligence (ai) in education from 2010 to 2020, *Complexity* 2021 (2021) 1–18.
- [3] K. Zhang, A. B. Aslan, Ai technologies for education: Recent research & future directions, *Computers and Education: Artificial Intelligence* 2 (2021) 100025.
- [4] L. Breiman, Random forests, *Machine Learning* 45 (2001) 5–32. Publisher: Springer.
- [5] V. G. Costa, C. E. Pedreira, Recent advances in decision trees: an updated survey, *Artificial Intelligence Review* 56 (2023) 4765–4800.
- [6] J. Tanha, M. van Someren, H. Afsarmanesh, Semi-supervised self-training for decision tree classifiers, *International Journal of Machine Learning and Cybernetics* 8 (2017) 355–370.
- [7] Q.-W. Wang, L. Yang, Y.-F. Li, Learning from weak-label data: a deep forest expedition, in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, 2020, pp. 6251–6258.
- [8] P. Ilidio, C. Vens, R. Cerri, F. K. Nakano, Deep forests with tree-embeddings and label impu-

- tation for weak-label learning, in: Proceedings of the 2024 International joint conference on neural networks, IJCNN, 2024.
- [9] K. Pliakos, C. Vens, Drug-target interaction prediction with tree-ensemble learning and output space reconstruction, *BMC Bioinformatics* 21 (2020) 1–11. Publisher: Springer.
- [10] A. Gharahighehi, K. Pliakos, C. Vens, Addressing the cold-start problem in collaborative filtering through positive-unlabeled learning and multi-target prediction, *Ieee Access* 10 (2022) 117189–117198.
- [11] X. Ning, G. Karypis, Slim: Sparse linear methods for top-n recommender systems, in: 2011 IEEE 11th international conference on data mining, IEEE, 2011, pp. 497–506.
- [12] A. Gharahighehi, M. Venturini, A. Ghinis, F. Cornillie, C. Vens, Extending bayesian personalized ranking with survival analysis for mooc recommendation, in: Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization, 2023, pp. 56–59.
- [13] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618* (2012).
- [14] X. Li, Z. Wang, L. Wang, R. Hu, Q. Zhu, A Multi-Dimensional Context-Aware Recommendation Approach Based on Improved Random Forest Algorithm, *IEEE Access* 6 (2018) 45071–45085.
- [15] C. Panagiotakis, H. Papadakis, P. Fragopoulou, A dual hybrid recommender system based on SCoR and the random forest, *Computer Science and Information Systems* 18 (2021) 115–128.
- [16] P. Ildio, A. Alves, R. Cerri, Fast Bipartite Forests for Semi-supervised Interaction Prediction, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 979–986.
- [17] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th international conference on World Wide Web, 2001, pp. 285–295.
- [18] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, *Recommender systems handbook* (2011) 73–105.
- [19] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 502–511.
- [20] H. Steck, Embarrassingly shallow autoencoders for sparse data, in: The World Wide Web Conference, 2019, pp. 3251–3257.
- [21] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al., Wide & deep learning for recommender systems, in: Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 7–10.
- [22] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 world wide web conference, 2018, pp. 689–698.
- [23] M. F. Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: Proceedings of the 13th ACM conference on recommender systems, 2019, pp. 101–109.

- [24] A. Alves, P. Ilidio, R. Cerri, Semi-supervised hybrid predictive bi-clustering trees for drug-target interaction prediction, in: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, ACM/SIGAPP, Tallinn, 2023, pp. 1163–1170.