CRUISE on Quantum Computing for Feature Selection in Recommender Systems

Notebook for the QuantumCLEF Lab at CLEF 2024

Jiayang Niu, Jie Li, Ke Deng and Yongli Ren

School of Computing Technologies, RMIT University, Melbourne, Victoria 3000

Abstract

Using Quantum Computers to solve problems in Recommender Systems that classical computers cannot address is a worthwhile research topic. In this paper, we use Quantum Annealers to address the feature selection problem in recommendation algorithms. This feature selection problem is a Quadratic Unconstrained Binary Optimization (QUBO) problem. By incorporating Counterfactual Analysis, we significantly improve the performance of the item-based KNN recommendation algorithm compared to using pure Mutual Information. Extensive experiments have demonstrated that the use of Counterfactual Analysis holds great promise for addressing such problems.

Keywords

Quantum Computers, Recommender Systems, Counterfactual Analysis, Feature Selection

1. Introduction

Collaborative filtering technology [1, 2], which predicts potential user-item interactions based on the patterns of user behavior and item characteristics, is widely applied in recommendation algorithms, Some well-known techniques in this field include matrix factorization methods [3], neighborhood-based methods [4], deep learning approaches [5, 6], graph-based techniques [7, 8], factorization machines [9], hybrid methods [10], Bayesian methods [11], and large language models (LLMs) [12]. However, collaborative filtering technology [1] heavily relies on the quality of data. For instance, using user profiles, item features, reviews, images, and other information can significantly improve the performance of recommendation algorithms, but in some cases, it can also decrease their performance. Therefore, it's critical to distinguish what information are useful for recommendations so as to help the the construction of efficient systems and reduction of energy consumption [13, 14, 15, 16]. Quantum computers, with its use of qubits and quantum effects like superposition, entanglement, and quantum tunneling, is an effective tool for identifying useful information from redundant data [17]. It significantly enhances the processing speed of search problems and large integer factorization [18]. Therefore, in this paper, we aim to find useful features for recommendations by leveraging quantum computing techniques. Our goal is to improve the efficiency and accuracy of recommendation systems by identifying and utilizing relevant data, thereby reducing computational requirements and energy consumption [18, 19, 20].

In QuantumCLEF 2024, we focus on Task 1B, where 150 and 500 features are provided for each item, respectively[21, 22]. We will analyze these features to extract the most relevant ones for recommender systems. The task requires participants to use Quantum Annealing and Simulated Annealing to select appropriate features from the given data for an Item-Based KNN recommendation algorithm (**Item-KNN**). The organizers provided an example of feature selection by using Mutual Information [18]. However, our preliminary experiments showed that using only Mutual Information for feature selection resulted in limited improvement in the performance of Item-KNN compared to using all features without any selection. This is because Mutual Information only reflects the mutual relationship between two variables and is not associated with the final goal of the recommendation algorithm. Therefore, to

CLEF 2024: Conference and Labs of the Evaluation Forum, September 9–12, 2024, Grenoble, France

s4068570@student.rmit.edu.au (J. Niu); hey.jieli@gmail.com (J. Li); ke.deng@rmit.edu.au (K. Deng); yongli.ren@rmit.edu.au (Y. Ren)

achieve better performance, we propose taking the impact of features on recommendation quality into consideration when performing feature selection.

One approach to achieve this is through Counterfactual Analysis [23], which is a causal research tool to examine the impact of a factor on the final result by hypothesizing the absence or alteration of that factor. This approach mainly considers three aspects: Which factors need to be evaluated? What metrics are used to assess the impact of these factors on the model's outcomes? And what models are used to derive the values of these metrics? In this work, due to the limited time for this task, we aim to measure and explore the impact of item features by Counterfactually Analyzing their effect on nDCG [24] performance of recommendation lists and we chose the KNN-based recommendation algorithm, a commonly used method in collaborative filtering, to perform these measurements. Specifically, we used Item-KNN to derive the change in nDCG values after removing a specific item feature. Since Mutual Information can reflect the relationship between two features, which may positively affects the final results, we did not discard it. Instead, we integrated the results of Counterfactual Analysis into Mutual Information using a temperature coefficient, which is used to control the influence of Counterfactual Analysis on the final results. Given the current limitations on the number of qubits in Quantum Computers, directly performing Quantum Annealing on 500 variables remains a challenging task. Therefore, in this task, we first partitioned the 500 features into subsets manageable by the Quantum Computer, and then combined the results.

The paper is organized as follows: Section 2 introduces related works; Section 3 describes the QUBO formulation, how Mutual Information is applied to QUBO for feature selection, and our proposed method of using Counterfactual Analysis for feature selection in QUBO; Section 4 explains our experimental setup and experimental result; Section 5 discusses our main findings; finally, Section 6 draws some conclusions and outlooks for future work.

2. Related Work

2.1. Quantum Computers

In recent years, the rapid development of Quantum Computers has demonstrated their tremendous potential in solving problems that Classical Computer cannot address, such as NP and NP-hard problems [25]. Based on their functionality and application scenarios, Quantum Computers can be categorized into Universal Quantum Computers, Quantum Annealers, Quantum Machine Learning Accelerators, and others [26]. Recent studies have utilized Quantum Annealers for feature selection to enhance the performance of recommendation systems or retrieval systems [27, 28, 18]. Nembrini et al. [27] attempted to apply Quantum Computers to recommendation systems by using Quantum Annealing to solve a hybrid feature selection approach. Their work demonstrates that current Quantum Computers are already capable of addressing real-world recommendation system problems. Nikitin et.al.[28] reproduced Nembrini's work and employed Tensor Train-based Optimization (TTOpt) as an optimizer for the cold start problem in recommendation systems. MIQUBO [18] discussed the problem of feature selection using Quantum Computers and formalizes it as a Quadratic Unconstrained Binary Optimization (QUBO) problem. It demonstrates the potential of Quantum Computers to solve ranking and classification problems more efficiently.

2.2. Counterfactual Analysis

Existing deep learning models have complex decision-making processes that are difficult for people to understand, often functioning as black-box models, Counterfactual Analysis is a highly effective method for helping people understand these complex models and robust them [29]. For example, $\mathbf{CF}^2$ [30] used Counterfactual Analysis to explore the explanations of Graph Neural Networks. In recommender systems, Counterfactual Analysis is primarily used for explainability and to combat data sparsity. ACCENT [31] was the first to apply Counterfactual Analysis to neural network-based recommendation algorithms. CountER [32] utilizes Counterfactual Analysis to construct a low-complexity, high-strength model for explaining recommendation systems. It also highlights that using Counterfactual Analysis contributes to the interpretability and evaluation of recommendation systems. Zhang et al [33] designed a CauseRec framework that utilizes Counterfactual to enhance representations in the data distribution, aiming to mitigate data sparsity.

In summary, Counterfactual Analysis can help people understand complex deep learning decision systems and has the potential to analyze how various factors interact in recommendation systems. Given the current advancements in Quantum Computers, utilizing Counterfactual Analysis combined with the ability of Quantum Computers to handle NP problems presents a promising direction.

3. Methodology

3.1. Preliminary

3.1.1. QUBO Formulation

In this work, we follow the approach described in [18], which utilizes Quantum Annealing for feature selection. To apply these methods, the feature selection problem is formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem. The QUBO formulation can be used to solve certain NP and NP-hard optimization problems and is defined as follows [18]:

$$\min Y = x^T Q x, \tag{1}$$

where x is a binary vector of length m, with each element of the vector being either 0 or 1. Q is a symmetric matrix, where each element represents the relationship between the elements of x. m denotes the number of features to be selected. In other words, the elements of vector x indicate whether the corresponding features are selected, and the elements in Q influence the search direction of the function, determining feature selection.

3.1.2. Feature Selection Based on Mutual Information

Following [18], Mutual Information QUBO (MIQUBO) is a quadratic feature selection model based on Mutual Information. MIQUBO aims to maximize the Mutual Information, which measures the dependency between two variables, and the Conditional Mutual Information, which measures the dependency between two variables given a target variable, of the selected features. In this context, the matrix Q in Equation 1 is defined as:

$$Q_{ij} = \begin{cases} -\mathrm{CMI}(f_i; y \mid f_j) & \text{if } i \neq j \\ -\mathrm{MI}(f_i; y) & \text{if } i = j, \end{cases} \tag{2}$$

where $\mathrm{MI}(f_i; y)$ is the Mutual Information between feature f_i and target feature y, and $\mathrm{CMI}(f_i; y \mid f_j)$ is the Conditional Mutual Information between feature f_i and target feature y given feature f_j. Since QUBO formulation is used to find the minimum state, a negative sign is required before MI and CMI.

To control the number of selected features, a penalty term is added to Equation 1, which is then transformed to:

$$\min Y = x^T Q x + \left(\sum_{i=1}^{N} x_i - k \right)^2 . \tag{3}$$

This formula will be minimized when selecting k features, this also following the descriptions in [18].

3.2. Counterfactual Analysis

To better identify features directly associated with recommendation performance, we integrate a widely used recommendation ranking metric into Mutual Information through Counterfactual Analysis.

3.2.1. Counterfactual Analysis for Feature Selection

Counterfactual Analysis [23] is usually used to examine the causal relationship between conditions, decisions, and outcomes by hypothesizing how the results of observed events would change if the conditions and decisions were altered. In the field of Recommender System, Counterfactual Analysis is often used for the interpretability of recommendation models, helping researchers enhance algorithm performance [32, 33]. Inspired by existing works [32, 33], the impact of item features can be explored by excluding the corresponding feature and analyzing the difference in recommendation performance between the recommendation lists generated by the model with and without the corresponding feature.

In this work, we use the widely used Item-KNN recommendation algorithm, termed as model G, and employ the recommendation performance metric Normalized Discounted Cumulative Gain (nDCG) [24] for Counterfactual Analysis. nDCG is defined as:

$$\mathrm{E}_i = \mathrm{nDCG}_{G(\mathrm{F})} - \mathrm{nDCG}_{G(\mathrm{F}\backslash f_i)}, \tag{4}$$

where E_i represents the change in the nDCG result of the recommendation model G after removing the feature f_i. $\mathrm{nDCG}_{G(\mathrm{F})}$ represents the nDCG@10 value obtained by the G using all item features set F, while $\mathrm{nDCG}_{G(\mathrm{F}\backslash f_i)}$ represents the nDCG@10 value obtained by the G using features set which is set F removing feature i. It is important to note that E_i ultimately reflects the impact of feature i on the result. Since the final outcome is influenced by the interactions between all features, simply removing features with positive E_i values does not yield the optimal feature selection solution.

When $E_i \geq 0$, it indicates that the algorithm's performance decreases after removing the feature i. The extent of this decrease reflects the positive impact of this feature on the algorithm. Conversely, an increase in the value reflects the negative impact of this feature on the algorithm. We hypothesize that if the selected set of features is $set(F^*)$, the maximization the sum of E_i ($i \in set(F^*)$), the maximization the performance improvement of the baseline algorithm. Since the QUBO problem is a minimization optimization problem, we redefine Q as follows:

$$Q_{ij} = \begin{cases} -CMI(f_i; y \mid f_j) & \text{if } i \neq j \\ -MI(f_i; y) - \lambda \mathrm{E}_i & \text{if } i = j \end{cases} \tag{5}$$

where λ is a coefficient used to control the influence of E on the search results. The larger the value of λ, the greater the influence of E on the final results. The overall process of the above algorithm, which we refer to as Counterfactual Analysis QUBO (CAQUBO), is as follows in Algorithm 1.

3.3. Handling Large Feature Set

Although Quantum Computers are developing rapidly, the limitation in the number of qubits restricts them to handling only a limited number of feature selection problems. For selecting from 500 features, we partition them into several subsets and use Quantum Annealing (QA) or Simulated Annealing (SA) to perform feature selection on these subsets individually, then combine the results.

First, partition the 500 features into n subsets by order, $S_1, S_2, \cdots, S_i, \cdots, S_n$, where S_i is the i-th subset of features, and n is the number of subsets.

$$S_1, S_2, \cdots, S_i, \cdots, S_n = \text{divide(F)} \tag{6}$$

Then, use Quantum Annealing (QA) or Simulated Annealing (SA) to perform feature selection on each subset, and combine the results:

$$\tilde{S} = \bigcup_{i=1}^{n} \text{QA/SA}(S_i), \tag{7}$$

where $\tilde{S}$ is the final selected features set, represents each partitioned subset of features, and QA/SA (S_i) represents the selected features from subset S_i using QA and SA. The final feature set is obtained by merging the selected features from all subsets.

Algorithm 1 Counterfactual Analysis QUBO

Initialize variable set E, set F, $n \leftarrow len$(F), k, Q, λ
procedure CALCULATE E$_i$
  for f_i in F **do**
    F$^{'}$ $\leftarrow$ F
    F$^{'}$.pop(f_i)
    E$_i$ $\leftarrow$G(F) - G(F$^{'}$)
  end for
  return E
end procedure
procedure FEATURE SELECTION
  Calculate MI and CMI
  for f_i in F **do**
    $Q_{ii} = -\text{MI}(f_i; y) - \lambda \text{E}_i$
  end for
  for f_i in F **do**
    for f_j in F **do**
      $Q_{ij} = -\text{CMI}(f_i; y \mid f_j)$
    end for
  end for
  set F* $\leftarrow$ QA or SA $\leftarrow$ Q and λ # Input parameters Q and λ into the Quantum Annealer.
  return set F* # Selected Feature Set
end procedure

4. Experimental Setup

Datasets: In this work, two tasks are undertaken: the first involves selecting appropriate features from a set of 150 item features for training G, and the second involves selecting features from a set of 500 item features. Three data sets are provided for these tasks: 150_ICM, 500_ICM, and URM. The 150_ICM and 500_ICM contain item features, while the URM includes interaction data between 1,890 users and 18,022 interacted items.
Experimental parameter setting: We used a self-implemented Item-KNN recommendation model based on the problem statement to calculate E. The interaction data was split into training and test sets in an 80:20 ratio. It is worth noting that calculating E is very time-consuming, so we only used a subset of items for the calculations. In the use of Quantum Annealing (QA) and Simulated Annealing(SA), the coefficient λ significantly affects the features selected by QA and SA. Due to the limited usage time of the Quantum Annealer (QA), it is necessary to use Simulated Annealing (SA) to explore the effectiveness of the selected features under different parameters λ and k before using QA. In preliminary experiment, we attempt **[λ: 0, 1e1, 1e3, 1e5, 1e7]**, **[k: 50, 100, 130, 140, 145] in Feature 150** and **[λ: 0, 1e1, 1e3, 1e5, 1e7]**, **[k: 300, 350, 400, 450, 470] in Feature 500**. For the selection of 500 features, **n** (is mentioned in Section 3.3) is set to 5. The preliminary experiment results can be found in Table 1.
Repeated Calculations: Due to the heuristic nature of Simulated Annealing (SA) and Quantum Annealing (QA), the final results may vary even with fixed parameters. To mitigate this effect, we perform multiple iterations of QA and SA under the same parameters and select the final feature set via voting. For example, we repeated the experiment five times. f_i was not included in F^* in any of the five experiments, while f_j was included in F^* in four out of the five experiments. Therefore, the final submitted feature set F^* does not include f_i but includes f_j.

Table 1
nDCG@10 for Feature 150 and Feature 500 datasets individually using SA-based feature selection, with different numbers of selected features k and different coefficients λ.

k	50	100	130	140	145	300	350	400	450	470
λ	Feature 150 nDCG@10					Feature 500 nDCG@10				
0	0.0602	0.0870	0.0968	0.1033	0.1018	0.1078	0.0894	0.0971	0.0969	0.0991
1	0.0870	0.0974	0.0999	0.1009	0.1029	0.1066	0.1108	0.1195	0.1291	0.1197
1e3	0.0755	0.1051	0.1151	0.1119	0.1152	0.1206	0.1249	0.1257	0.1305	0.1302
1e5	**0.0878**	**0.1160**	**0.1232**	0.1256	**0.1180**	0.1224	**0.1238**	**0.1303**	0.1290	**0.1307**
1e7	0.0795	0.1155	0.1221	**0.1264**	**0.1180**	**0.1235**	0.1218	0.1298	**0.1306**	0.1293
	150 Feature nDCG 0.1028					**500 Feature nDCG 0.0988**				

Table 2
This table contains the final data submitted to the organizers, with data sourced from the organizers' website[1]. Due to the fact that when $\boldsymbol{\lambda}$ is too large, the values of elements in $\mathbf{Q}$ become excessively large, which is detrimental to the performance of QA and SA, a coefficient $\boldsymbol{\mu}$ is applied to all elements in $\mathbf{Q}$. An asterisk (*) after the sub_ID indicates that the selected features are the result of repeated calculations. Those submissions was repeated five times to determine the final feature set.

150 Feature submissions	**All Feature nDCG 0.0810**				
Parameters set	**nDCG@10**	**Annealing Time**	**Type**	**n° features**	**sub_id**
k=140 λ=1e7 μ=1e-5	0.0805	536250	Q	138	1
k=140 λ=1e7 μ=1e-3	0.0826	528844	Q	136	2
k=140 λ=1e7 μ=1e-3	0.0690	530804	Q	132	3
k=140 λ=0 μ=1	0.0763	558321	Q	133	4
k=140 λ=1e7 μ=1e-2	0.1003	1375068	Q	144	5*
k=140 λ=1e7 μ=1e-5	0.0998	1745487	S	140	1
k=140 λ=1e7 μ=1e-3	0.0993	17357899	S	140	2
k=140 λ=1e7 μ=1e-3	0.1001	1760252	S	140	3
k=140 λ=0 μ=1	0.0793	17387227	S	140	4
k=140 λ=1e7 μ=1e-2	0.1003	88395437	S	144	5*
500 Feature submissions	**All Feature nDCG 0.0827**				
k=450 λ=1e7 μ=1e-2	0.0757	2287019	Q	407	1
k=450 λ=1e1 μ=1	0.0839	2122701	Q	397	2
k=450 λ=1e7 μ=1e-2	0.1196	43339285	S	450	1
k=450 λ=1e1 μ=1	0.1198	42776695	S	450	2

[1] **https://qclef.dei.unipd.it/clef2024-results.html**

5. Results

Table 1 describes the performance in nDCG@10 of G using features selected by QA and SA under different parameters λ and k. When $\lambda = 0$, QA and SA select features based solely on Mutual Information (MI) and Conditional Mutual Information (CMI). Across different values of parameter k, the performance of selected features in G rarely surpasses the performance in Counterfactual Analysis QUBO. As the parameter λ increases, the performance of the features selected by QA and SA in the item-KNN shows significant improvement compared to using all features. The effectiveness of feature selection shows no significant improvement when $\lambda > 1e5$. This may be because as the value of λ increases, the impact of MI and CMI on feature selection diminishes, causing QA and SA to rely entirely on E for feature selection.

Table 2 reflects the same situation: feature selection relying solely on MI and CMI does not surpass the performance in Counterfactual Analysis QUBO. After incorporating the counterfactual analysis-derived E into Q, the features selected by QA and SA show a significant performance improvement in item-KNN compared to using all features. An unusual observation is that, under the same parameters, the features selected by QA generally do not perform as well as those selected by SA in item-KNN, and sometimes do not even surpass the performance of using all features. During the experiments, we noticed that this is due to QA often returning results before finding the optimal solution.

6. Conclusions and Future Work

In this paper, we present the explorations conducted by our team and the details of our final submission for the QuantumCLEF 2024 activities. We used Counterfactual Analysis of individual item features to select appropriate features for item-KNN using Quantum Annealing. Our preliminary experiments and the results returned by QuantumCLEF 2024 demonstrated that our use of Counterfactual Analysis significantly improved the performance of item-KNN.

Within the limited time of QuantumCLEF, we attempted Counterfactual Analysis of individual features. However, because the performance of collaborative filtering is actually the result of feature interactions, Counterfactual Analysis of individual features has significant limitations. Additionally, since Quantum Annealing cannot directly handle the selection of 500 features, we adopted a sequential partitioning and merging approach. As negative features are not uniformly distributed by their indices among all features, this sequential partitioning and merging method still requires improvement.

References

[1] X. Su, T. M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in artificial intelligence 2009 (2009).

[2] J. Lee, M. Sun, G. Lebanon, A comparative study of collaborative filtering algorithms, arXiv preprint arXiv:1205.3193 (2012).

[3] N. Koenigstein, P. Ram, Y. Shavitt, Efficient retrieval of recommendations in a matrix factorization framework, in: Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 535–544.

[4] D. A. Adeniyi, Z. Wei, Y. Yongquan, Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method, Applied Computing and Informatics 12 (2016) 90–108.

[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939 (2015).

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[7] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019, pp. 165–174.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[9] F. Yuan, G. Guo, J. M. Jose, L. Chen, H. Yu, W. Zhang, Lambdafm: Learning optimal ranking with factorization machines using lambda surrogates, in: Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 227–236.

[10] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE transactions on knowledge and data engineering 17 (2005) 734–749.

[11] R. Lopes, R. Assunção, R. L. Santos, Efficient bayesian methods for graph-based recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 333–340.

[12] Y. Yang, K. S. Kim, M. Kim, J. Park, Gram: Fast fine-tuning of pre-trained language models for content-based collaborative filtering, arXiv preprint arXiv:2204.04179 (2022).

[13] S. Marchesin, A. Purpura, G. Silvello, Focal elements of neural information retrieval models. an outlook through a reproducibility study, Information Processing & Management 57 (2020) 102109.
[14] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for deep learning in nlp, arXiv preprint arXiv:1906.02243 (2019).
[15] Y. Himeur, A. Alsalemi, A. Al-Kababji, F. Bensaali, A. Amira, C. Sardianos, G. Dimitrakopoulos, I. Varlamis, A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects, Information Fusion 72 (2021) 1–21.
[16] G. Adomavicius, J. Zhang, Impact of data characteristics on recommender systems performance, ACM Transactions on Management Information Systems (TMIS) 3 (2012) 1–17.
[17] Y. Lu, A. Sigov, L. Ratkin, L. A. Ivanov, M. Zuo, Quantum computing and industrial information integration: A review, Journal of Industrial Information Integration (2023) 100511.
[18] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, P. Cremonesi, Towards feature selection for ranking and classification exploiting quantum annealers, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2814–2824.
[19] F. Glover, G. Kochenberger, Y. Du, Quantum bridge analytics i: a tutorial on formulating and using qubo models, 4or 17 (2019) 335–371.
[20] G. Pilato, F. Vella, A survey on quantum computing for recommendation systems, Information 14 (2022) 20.
[21] A. Pasin, M. Ferrari Dacrema, P. Cremonesi, N. Ferro, QuantumCLEF 2024: Overview of the Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, September 9th to 12th, 2024, 2024.
[22] A. Pasin, M. Ferrari Dacrema, P. Cremonesi, N. Ferro, Overview of QuantumCLEF 2024: The Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9-12, 2024, Proceedings, 2024.
[23] J. Pearl, M. Glymour, N. P. Jewell, Causal inference in statistics: A primer, John Wiley & Sons, 2016.
[24] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (2002) 422–446.
[25] L. Bittel, M. Kliesch, Training variational quantum algorithms is np-hard, Physical review letters 127 (2021) 120502.
[26] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, R. Buyya, Quantum computing: A taxonomy, systematic review and future directions, Software: Practice and Experience 52 (2022) 66–114.
[27] R. Nembrini, M. Ferrari Dacrema, P. Cremonesi, Feature selection for recommender systems with quantum computing, Entropy 23 (2021) 970.
[28] A. Nikitin, A. Chertkov, R. Ballester-Ripoll, I. Oseledets, E. Frolov, Are quantum computers practical yet? a case for feature selection in recommender systems using tensor networks, arXiv preprint arXiv:2205.04490 (2022).
[29] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, C. Shah, Counterfactual explanations and algorithmic recourses for machine learning: A review, arXiv preprint arXiv:2010.10596 (2020).
[30] M. L. Olson, R. Khanna, L. Neal, F. Li, W.-K. Wong, Counterfactual state explanations for reinforcement learning agents via generative deep learning, Artificial Intelligence 295 (2021) 103455.
[31] K. H. Tran, A. Ghazimatin, R. Saha Roy, Counterfactual explanations for neural recommenders, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1627–1631.
[32] J. Tan, S. Xu, Y. Ge, Y. Li, X. Chen, Y. Zhang, Counterfactual explainable recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1784–1793.

[33] S. Zhang, D. Yao, Z. Zhao, T.-S. Chua, F. Wu, Causerec: Counterfactual user sequence synthesis for sequential recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 367–377.