

ReCon: Reducing Congestion in Job Recommendation using Optimal Transport

Yoosof Mashayekhi

yoosof.mashayekhi@ugent.be

IDLAB - Department of Electronics and Information
Systems, Ghent University
Ghent, Belgium

Jefrey Lijffijt

jefrey.lijffijt@ugent.be

IDLAB - Department of Electronics and Information
Systems, Ghent University
Ghent, Belgium

Bo Kang

bo.kang@ugent.be

IDLAB - Department of Electronics and Information
Systems, Ghent University
Gent, Belgium

Tijl De Bie

tijl.debie@ugent.be

IDLAB - Department of Electronics and Information
Systems, Ghent University
Gent, Belgium

ABSTRACT

Recommender systems may suffer from *congestion*, meaning that there is an unequal distribution of the items in how often they are recommended. Some items may be recommended much more than others. Recommenders are increasingly used in domains where items have limited availability, such as the job market, where congestion is especially problematic: Recommending a vacancy—for which typically only one person will be hired—to a large number of job seekers may lead to frustration for job seekers, as they may be applying for jobs where they are not hired. This may also leave vacancies unfilled and result in job market inefficiency.

We propose a novel approach to job recommendation called ReCon, accounting for the congestion problem. Our approach is to use an optimal transport component to ensure a more equal spread of vacancies over job seekers, combined with a job recommendation model in a multi-objective optimization problem. We evaluated our approach on two real-world job market datasets. The evaluation results show that ReCon has good performance on both congestion-related (e.g., Congestion) and desirability (e.g., NDCG) measures.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Job recommendation, Congestion-avoiding recommendation

1 INTRODUCTION

Recommendation systems are widely used in online platforms [1]. They are designed to enhance the user experience by providing personalized recommendations [7, 15] to facilitate access to the most relevant content. Recommendation systems have been applied to various types of content, such as movies [16, 18], online shopping [8, 17], and jobs [14].

The aim of job recommendation is to recommend jobs (items) that are suitable for job seekers (users) and, if applied correctly, can have a positive impact on their career paths. In a job market, there is competition between job seekers for jobs and also between jobs (or companies) for job seekers [10]. This competition may be aggravated by the presence of so-called *congestion*, meaning

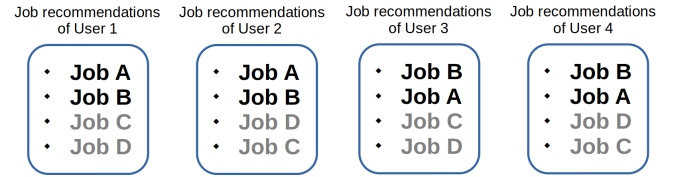


Figure 1: An illustration of congestion in job recommendation. Four jobs are recommended to four users, while the top-2 recommendations only include jobs A and B for all users (e.g., because of the popularity bias or their inherent higher relevancy score). Hence, if users only apply for their top-2 recommendations, jobs A and B receive four applications, while jobs C and D receive none.

that some items are more visible and desirable than others [5]. The congestion problem is illustrated with an example in Figure 1. The use of job recommender systems risks to create or amplify congestion, as desirable items are often recommended to many users. This risks frustrating job seekers, as they will be unlikely to be successful when all applying for the same jobs. It is therefore important that job recommender systems explicitly account for congestion and try to limit or reduce it. The aim of this paper is to address that need.

PROBLEM 1. *The problem we address in this paper is to reduce congestion in job recommendation, i.e., to improve congestion-related measures such as Congestion [2], Coverage, and Gini index, while recommending desirable recommendations, i.e., to keep good performance of desirability measures such as Recall, NDCG, and Hit Rate.*

Our approach is to spread jobs over job seekers more equally using the optimal transport theory. We optimize the spread of jobs over job seekers with a given job recommendation model as a multi-objective task. Since optimal transport aims to match two distributions, we minimize the optimal transport between job seekers and jobs, to enforce a high matching degree between each job and a few job seekers and vice versa. Hence, we expect that the congestion in the generated recommendations by the model is reduced.

The joint optimization of the base recommendation model and the optimal transport component leads to having a trained model that offers advantages over post-processing approaches in real-life scenarios that require incremental updates of the model.

The main **contributions** of this paper are:

- We introduce a novel approach, ReCon, to reduce congestion in a given job recommendation model. We jointly optimize the job recommendation model objective function and optimal transport cost between job seekers and jobs as a multi-objective task. (Section 2)
- We show how to employ the so-called entropic optimal transport using the Sinkhorn algorithm [6] to facilitate the optimization in ReCon. (Section 2.3)
- We evaluate ReCon on two job recommendation datasets and compare it with the baselines in terms of three desirability measures (e.g., NDCG), and also three congestion-related measures (e.g., Congestion). (Section 3)

Related work is summarized in Section 4 and final conclusions are presented in Section 5.

2 PROPOSED METHOD

In this section, we introduce our approach, ReCon, to address Problem 1, reducing congestion in job recommendation. We first give a brief introduction to optimal transport theory and its efficient optimization using the Sinkhorn algorithm in Section 2.1. Next, we explain the multi-objective approach in Section 2.2. We then introduce the requirements for the efficient optimization of ReCon in Section 2.3. Finally, we suggest a matching cost function and a similarity function for computing optimal transport between job seekers and jobs in Section 2.4 that meet those requirements.

2.1 Optimal Transport

Optimal transport refers to the problem of finding the most efficient way to transport one given distribution of resources (e.g., masses, probabilities) to another. Mathematically, this can be represented as finding a transportation plan that minimizes the cost of transporting the resources from one place to another, subject to certain constraints. The cost of transporting the resources is often represented using a distance function. In the context of the present paper, we are interested in the optimal transport between users U (job seekers) and items I (jobs), where $D = [d_{ui}]$ denotes a cost of matching users with items. Hence, the optimal transport plan $F = [f_{ui}]$ is given as:

$$\min_F \sum_{u \in U} \sum_{i \in I} f_{ui} d_{ui}, \quad \text{s.t.} \quad f_{ui} \geq 0 \quad \forall (u, i) \in U \times I, \\ \sum_{i \in I} f_{ui} = w_u \quad \forall u \in U, \quad \sum_{u \in U} f_{ui} = w_i \quad \forall i \in I. \quad (1)$$

A tractable relaxation of the above optimization problem is proposed in [6], by regularization with an entropy term:

$$\min_F \sum_{u \in U} \sum_{i \in I} f_{ui} (d_{ui} + \epsilon \log(f_{ui})), \quad \text{s.t.} \quad \sum_{i \in I} f_{ui} = w_u \quad \forall u \in U, \\ \sum_{u \in U} f_{ui} = w_i \quad \forall i \in I, \quad (2)$$

with ϵ the regularization weight. This relaxed optimization problem can be solved with the Sinkhorn algorithm [6], where its solution is differentiable w.r.t. the cost values D , and suitable for various optimization tasks.

For simplicity, we set $w_u = \frac{1}{|U|}$ for all users $u \in U$ and $w_i = \frac{1}{|I|}$ for all items $i \in I$ in Eq. (1) and Eq. (2), so that the total mass for users and items is the same and all users, respectively items, are treated equally.

2.2 ReCon

Our approach is to optimize the desirability objective function of a recommendation model and an optimal transport cost between job seekers and jobs as a multi-objective task. The costs of matching individuals to jobs in the optimal transport are based on the recommendation scores, such that it uses the preferences of each individual, but it goes beyond the objective of the recommender, as the optimal transport problem accounts also for the preferences of other individuals. The optimal transport problem distributes job seekers more equally over all jobs and vice versa. Hence, we expect that the recommendations from ReCon have lower congestion.

For a given recommendation model M with an objective function O_M and a set of parameters $P = [p_{ui}]$, where p_{ui} represents the matching score between user u and item i , the multi-objective task is as follows:

$$O_{\text{ReCon}} = O_M + \lambda O_C, \quad (3)$$

where O_C is the objective function of the optimal transport between job seekers and jobs that is defined using the recommendation scores P (defined in Section 2.4). The goal of O_C is to reduce the congestion in the recommendations. The weight of O_C in the multi-objective task is also indicated by the hyper-parameter λ .

To recommend top- k items to user u , we have to compare the recommendation scores between u and all items. To account for this competition between different items for the same user, it is important not only to penalize the cost of matching user-item pairs that are part of the optimal transport solution in the optimal transport objective function, but also to penalize the similarity of user-item pairs that are not part of the optimal transport solution (more intuition about this approach is provided in Section 2.4). Hence, the optimal transport optimization problem in ReCon is:

$$\min_F \sum_{u \in U} \sum_{i \in I} f_{ui} c(p_{ui}) + (1 - f_{ui}) s(p_{ui}), \\ \text{s.t.} \quad f_{ui} \geq 0 \quad \forall (u, i) \in U \times I, \\ \sum_{i \in I} f_{ui} = w_u \quad \forall u \in U, \quad \sum_{u \in U} f_{ui} = w_i \quad \forall i \in I, \quad (4)$$

where c is a function of P to indicate the cost of matching a user and an item and s is a function of P to indicate the similarity of a user and an item. We can write the objective function in Eq. (4) as $\sum_{u \in U} \sum_{i \in I} s(p_{ui}) + \sum_{u \in U} \sum_{i \in I} f_{ui} (c(p_{ui}) - s(p_{ui}))$. Since the first double summation does not depend on F , it can be optimized outside the linear program in Eq. (4). Thus, the objective function in the linear program can be written as $\sum_{u \in U} \sum_{i \in I} f_{ui} (c(p_{ui}) - s(p_{ui}))$, which can be optimized efficiently using the Sinkhorn algorithm with Eq. (2), using $d_{ui} = c(p_{ui}) - s(p_{ui})$. We denote this objective function by

O_{COT} . Hence, the multi-objective task can be written as:

$$O_{\text{ReCon}} = O_M + \lambda O_{\text{COT}} + \lambda \sum_{u \in U} \sum_{i \in I} s(p_{ui}). \quad (5)$$

2.3 Efficient optimization

For efficient optimization, all terms in Eq. (5) have to be differentiable w.r.t. P . Since O_{COT} computed by Eq. (1) is not differentiable w.r.t. functions c and s , it may also not be differentiable w.r.t. P because the functions c and s depend on the recommendation scores P . If we compute O_{COT} by Eq. (2), the Sinkhorn algorithm could be used for the optimization, which is differentiable w.r.t. P when c and s are differentiable w.r.t. P . We also require the objective function of the recommendation model O_M to be differentiable w.r.t. P . Hence, the objective function O_{ReCon} in Eq. (5) would be differentiable w.r.t. P and optimization algorithms such as gradient descent may be employed for the optimization.

In summary, the requirements for efficient optimization are:

- (1) The objective function of the recommendation model O_M must be differentiable w.r.t. P .
- (2) Functions c and s must be differentiable w.r.t. P .

We now propose a matching cost function c and a similarity function s that meet the second requirement.

2.4 The matching cost and similarity functions

High recommendation score p_{ui} indicates that user u and item i may match well. Hence, the cost of matching them in optimal transport should be low and their similarity should be high. If $p_{ui} > 0$, we suggest using the following functions c and s as the cost of matching user u with item i and the similarity of user u and item i respectively:

$$c(p_{ui}) = -\ln(p_{ui}), \quad s(p_{ui}) = -\ln(1 - p_{ui}). \quad (6)$$

Hence, O_C in Eq. (3) would be $-\sum_{u \in U} \sum_{i \in I} f_{ui} \ln(p_{ui}) + (1 - f_{ui}) \ln(1 - p_{ui})$.

$$p_{ui} = -\sum_{u \in U} \sum_{i \in I} \ln(p_{ui}^{f_{ij}}) + \ln((1 - p_{ui})^{1-f_{ij}}) = -\ln \prod_{u \in U} \prod_{i \in I} p_{ui}^{f_{ij}} (1 - p_{ui})^{1-f_{ij}}.$$

The intuition behind choosing this term is as follows. In the case where $|U| = |I|$ the optimal transport has an optimum where each user is mapped to exactly one item and vice versa, i.e., $f_{ij} \in \{0, 1\}$ in $F = [f_{ij}]$. For this case, if P represents the matching probabilities, it is easy to see that computing the optimal transport would be the same as computing a matching between users and items with the highest probability accounting for the user-item pairs that are not part of the matching as well.

Functions c and s in Eq. (6) are differentiable w.r.t. the recommendation scores P (the set of parameters of model M). Hence, functions c and s in Eq. (6) meet the second requirement in Section 2.3, allowing for efficient optimization.

3 EXPERIMENTAL EVALUATION

In this section, we describe the experimental evaluation of ReCon. We investigate two research questions **Q1**: How does ReCon perform in congestion-related measures and desirability measures compared to the base recommendation model and the baselines? **Q2**: How

does ReCon perform in terms of execution time compared to other methods?

We first discuss the datasets, recommendation model, baselines, and settings. Next, we present the result of each experiment. The source code for repeating the experiments, the preprocessed public datasets, details of the selection of hyper-parameters for each dataset, more results for baseline comparison, and hyper-parameter sensitivity analysis for λ are available in the online supplementary material¹.

3.1 Datasets

We evaluated the methods using two datasets:

VDAB²: VDAB is the employment service of Flanders in Belgium. The dataset contains a suitably anonymized sample of the applications made by job seekers to available job vacancies in the last ten days of 2018.

CareerBuilder³: CareerBuilder is an e-recruitment platform, which matches job seekers with jobs. We use the applications made by job seekers to available job vacancies in the last ten days of its public dataset⁴ for the evaluation.

To better use collaborative filtering signals, we only keep job seekers and jobs with at least four interactions in both datasets. Table 1 shows the main statistics of the datasets.

3.2 Recommendation model

Although ReCon can be applied to any recommendation model satisfying the requirements in Section 2.3, we used a collaborative filtering method for the evaluation. The reason for choosing a CF method is that the performance of content-based methods depends on the quality of the contextual features whereas obtaining those features requires different feature engineering pipelines for different datasets. Since the focus of this paper is not improving the recommendation performance with different feature engineering techniques, we evaluated ReCon using a CF method.

We use Conditional Network Embedding (CNE; [9]) as the base recommendation model. CNE is a network embedding method that proposes a probability distribution for the network conditional on the embedding and finds the optimal embedding by maximum likelihood estimation. CNE satisfies the requirements in Section 2.3 for efficient optimization.

3.3 Baselines

We compare ReCon with the following baselines that both receive the recommendation scores from a trained model in their first step. Hence, both baselines have post-processing approaches.

CAROT [2]: CAROT employs optimal transport to redistribute jobs among job seekers. The final recommendation is according to the optimal transport solution. Since both methods presented in [2] and [11] have a similar approach, we only compare our proposed method with CAROT [2].

FairRec [12]: FairRec is a greedy method that ensures Envy-Free job recommendations for job seekers and guarantees a minimum

¹<https://github.com/aida-ugent/ReCon/>

²<https://www.vdab.be/>

³<https://www.careerbuilder.com/>

⁴<https://www.kaggle.com/c/job-recommendation>

Table 1: Main statistics of the datasets used for evaluation.

Dataset	Job seekers	Jobs	Train Interactions	Validation Interactions	Test Interactions
VDAB	1693	2931	9766	1428	2950
CareerBuilder	3876	4337	24316	1071	4557

exposure for job vacancies by converting the fair recommendation problem into a fair allocation problem. Since [12] and [3] have a similar approach and the modification in [3] is not focused on the congestion problem, we only compare our proposed method with FairRec [12]. As FairRec requires $|I| \leq k \cdot |U|$, we did not evaluate it for $k = 1$.

3.4 Settings

Both datasets include the interactions in a period of ten days. For each dataset, we use the interactions in the first six days for training, the seventh day for validation, and the last three days for the test set. We also use the source code provided by the authors for the baselines.

For Carot, we select ϵ from $\{1, 10, 100\}$. We also use four different functions to obtain the matching costs in the second step of the method (that solves the optimal transport problem) including linear (Id+), exponential (Exp+), rank-based (Rank), or NDCG-like (NDCG) functions as mentioned in CAROT paper [2]. For FairRec, we select α (the coefficient for the producer-side guarantee in their method) from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. Since FairRec generates a specific list of k recommendations for a given k that is not guaranteed to contain the same order of recommendations for e.g., $k - 1$, we run the method for each value k in the top- k recommendation for evaluation. For ReCon, we set ϵ to 10, the number of Sinkhorn iterations to 10, and we select λ from $\{1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6\}$.

We evaluate the methods on the following desirability measures:

- NDCG: Normalized discounted cumulative gain, which is computed by summing the true scores ranked in the order induced by the predicted scores after applying a logarithmic discount, divided by the best possible score.
- Recall: The proportion of relevant items found in the top- k recommendations.
- Hit Rate: The fraction of users for which the correct answer is included in the top- k recommendations.

We also evaluate the methods on the following congestion-related measures:

- Congestion: Congestion is defined as the negative entropy of item market shares, where item market share $MS_I(i)$ of item i is defined as the fraction of users such that item i appears among their top- k recommendations [2]. As suggested by Bied et al. [2], Congestion is divided by $\log(|I|)$ (I represents the set of items) to map it to $[-1, 0]$. Congestion is minimized if the entropy of the market shares is minimized. Hence, -1 is the optimal value.
- Coverage: The fraction of items involved in top- k recommendation of at least one user [2].

- Gini Index: The Gini index is a statistical measure of the inequality of a distribution [13]. In the context of job recommendation, the Gini index is computed for the market share of items.

3.5 Results

In this section, we present the results of the experiments to investigate the research questions **Q1** and **Q2**.

3.5.1 Baseline comparison. Here we compare the methods in terms of the desirability measures and congestion-related measures (**Q1**). Figure 2 shows desirability measures against congestion-related measures for all datasets. ReCon mostly improves congestion-related measures, while keeping good performance (or improving over CNE in some cases) of desirability measures for some selections of hyper-parameters. We can also observe that ReCon is Pareto optimal for some selections of hyper-parameters. While achieving almost the same or better desirability measure performance compared to CNE, ReCon can result in better congestion-related measures compared to plain CNE and the baselines.

The same conclusion could generally be established for all the desirability measures against all congestion-related measures, where for some selections of hyper-parameters, ReCon usually finds a good trade-off between both measures. More figures for different combinations of measures are available in the online supplementary material⁵.

Compared to the baselines, ReCon has an additional advantage in that it trains a recommendation model, whereas the baselines simply generate a recommendation list by post-processing the original recommendations from a model. This makes ReCon particularly advantageous in real-world applications where the model requires incremental updates to account for new interactions.

3.5.2 Execution time comparison. In this experiment, we compare the methods in terms of the execution time (**Q2**).

Figure 3 shows the execution time in hours for both datasets. The execution time of each baseline is averaged over different selections of hyper-parameters. The training time of CNE is added to the execution time of CAROT and FairRec, as they receive the matching scores from the base recommendation model. ReCon is slower than CNE and CAROT since it optimizes the optimal transport cost and the base recommendation model jointly. However, ReCon is faster than FairRec.

4 RELATED WORK

One important aspect of job recommendation models is that there is competition between job seekers/jobs for the same jobs/job seekers

⁵<https://github.com/aida-ugent/ReCon/>

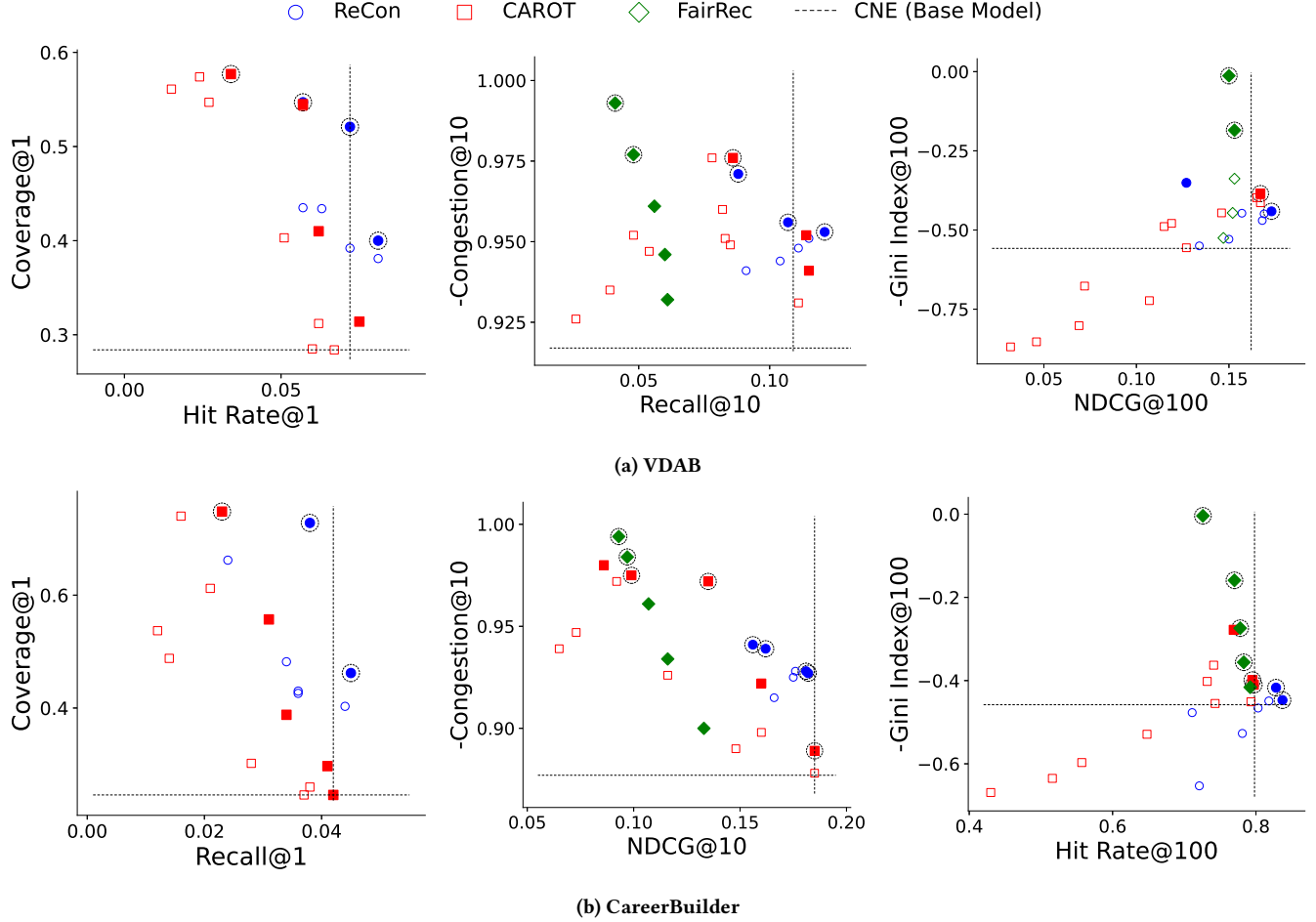


Figure 2: Desirability versus congestion-related measures (higher values are better). Points represent different hyper-parameter combinations. Pareto optimal points per method are solid (filled). Pareto optimal points across methods have a circle around.

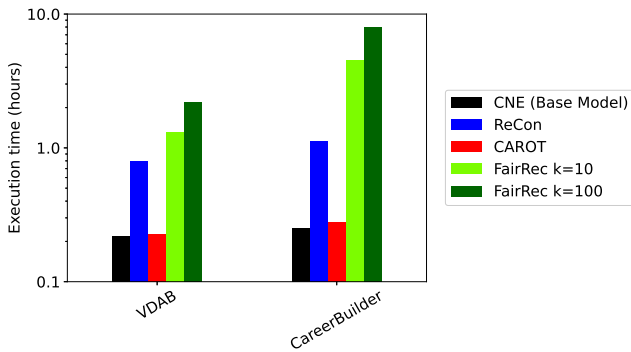


Figure 3: Execution time (log-scale) for both datasets. The execution time of each baseline is averaged over different selections of hyper-parameters.

[10]. This competition amplifies the problem of congestion with recommendations in the job market.

In [4], a job application redistribution system was proposed for LinkedIn to prevent job postings from receiving an excessive or insufficient number of applications. To achieve this goal, the job recommendation scores were adjusted using a dynamic forecasting model that penalized or boosted scores based on the predicted number of applications. Chen et al. [5] use a decentralized economic model to learn the scoring functions and use an optimal transport approach for the final recommendations to reduce congestion in the recommendations.

To reduce congestion in job recommendation, two-step approaches (including CAROT) are proposed in [2, 11]. The first step uses a model to predict the recommendation scores, and the second step employs optimal transport to better distribute jobs between job seekers and generate the final recommendations. In contrast, ReCon integrates the congestion reduction with recommendation into a single multi-objective task.

From a fairness perspective, two recent approaches (including FairRec) [3, 12] develop greedy algorithms to ensure Envy-Free

job recommendations for job seekers and to guarantee a minimum exposure for job vacancies.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel approach, ReCon, for reducing congestion in job recommendation systems using optimal transport theory. Our approach aimed to ensure a more equal spread of jobs over job seekers by combining optimal transport and a given job recommendation model in a multi-objective optimization problem. The evaluation results showed that our approach had good performance on both congestion-related and desirability measures. The proposed method can be a promising solution to the problem of congestion in job recommendation systems, and can potentially lead to more efficient and effective job matching.

For future work, we intend to evaluate ReCon using various recommendation models. Moreover, we aim to scale ReCon to be applicable to large-scale job market datasets by optimizing optimal transport in each batch of the training data, instead of the whole dataset. Additionally, this could reduce the training time as well.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the European Union’s Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Flemish Government under the “Onderzoeksprogramma Artificial Intelligence (AI) Vlaanderen” programme, and from the FWO (project no. G091017N, G0F9816N, 3G042220). Part of the experiments were conducted on pseudonimized HR data generously provided by VDAB.

REFERENCES

- [1] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. 2019. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review* 52, 1 (2019), 1–37.
- [2] Guillaume Bied, Elia Perennes, Victor Naya, Philippe Caillou, Bruno Crépon, Christophe Gaillac, and Michele Sebag. 2021. Congestion-Avoiding Job Recommendation with Optimal Transport. In *FEAST workshop ECML-PKDD 2021*.
- [3] Arpita Biswas, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2021. Toward Fair Recommendation in Two-sided Platforms. *ACM Transactions on the Web (TWEB)* 16, 2 (2021), 1–34.
- [4] Fedor Borisjuk, Liang Zhang, and Krishnamurthy K. 2017. LiJAR: A System for Job Application Redistribution towards Efficient Career Marketplace. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Halifax, NS, Canada) (KDD ’17)*. Association for Computing Machinery, New York, NY, USA, 1397–1406.
- [5] Kuan-Ming Chen, Yu-Wei Hsieh, and Ming-Jen Lin. 2019. Prediction and congestion in two-sided markets: Economist versus machine matchmakers. *SSRN Electronic Journal* (2019).
- [6] Marco Cuturi. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc.
- [7] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2022. A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2022), 3549–3568.
- [8] Jiayu Han, Lei Zheng, Yuanbo Xu, Bangzuo Zhang, Fuzhen Zhuang, S Yu Philip, and Wanli Zuo. 2019. Adaptive deep modeling of users and items using side information for recommendation. *IEEE Transactions on Neural Networks and Learning Systems* 31, 3 (2019), 737–748.
- [9] Bo Kang, Jeffrey Lijffijt, and Tijl De Bie. 2019. Conditional network embeddings. In *7th International Conference on Learning Representations, ICLR 2019*.
- [10] Yoosof Mashayekhi, Nan Li, Bo Kang, Jeffrey Lijffijt, and Tijl De Bie. 2022. A challenge-based survey of e-recruitment recommendation systems. *arXiv preprint arXiv:2209.05112* (2022).
- [11] Victor Naya, Guillaume Bied, Philippe Caillou, Bruno Crépon, Christophe Gaillac, Elia Pérennes, and Michèle Sebag. 2021. Designing labor market recommender systems: the importance of job seeker preferences and competition. In *4. IDSC of IZA Workshop: Matching Workers and Jobs Online-New Developments and Opportunities for Social Science and Practice*.
- [12] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *Proceedings of The Web Conference 2020 (Taipei, Taiwan) (WWW ’20)*. Association for Computing Machinery, New York, NY, USA, 1194–1204.
- [13] Thomas Piketty. 2015. *The economics of inequality*. Harvard University Press.
- [14] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. 2020. An enhanced neural network approach to person-job fit in talent recruitment. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.
- [15] J Ben Schafer, Joseph A Konstan, and John Riedl. 2001. E-commerce recommendation applications. *Data Mining and Knowledge Discovery* 5, 1 (2001), 115–153.
- [16] Chao Wang, Hengshu Zhu, Chen Zhu, Chuan Qin, and Hui Xiong. 2020. Setrank: A setwise bayesian approach for collaborative ranking from implicit feedback. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 34. 6127–6136.
- [17] Yuanbo Xu, Yongjian Yang, Jiayu Han, En Wang, Fuzhen Zhuang, Jingyuan Yang, and Hui Xiong. 2019. NeuO: Exploiting the sentimental bias between ratings and reviews with neural networks. *Neural Networks* 111 (2019), 77–88.
- [18] Fuzhen Zhuang, Zhiqiang Zhang, Mingda Qian, Chuan Shi, Xing Xie, and Qing He. 2017. Representation learning via dual-autoencoder for recommendation. *Neural Networks* 90 (2017), 83–89.