Internal Placement: Job Recommender Systems with Social Regularization

Simon De Vos[1] [0000-0002-3032-8678], Johannes De Smedt[1] [0000-0003-0389-0275], Marijke Verbruggen[1] [0000-0002-7375-7771], and Wouter Verbeke[1] [0000-0002-8438-0535]

[1] KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium
{simon.devos,johannes.desmedt,marijke.verbruggen,wouter.verbeke}@kuleuven.be

Keywords: HR Analytics, Internal Placement, Matching, Recommender Systems.

1 Introduction

With a vast amount of available data, companies in almost every sector are devoted to exploiting this data for gaining a competitive advantage [1]. In recent years, many organizations focus more and more on data-driven insights to support daily decision-making [2]. The field of human resources (HR) is strongly anticipating the transformative potential of big data, analytics, and machine learning [3]. Hence, albeit with a certain delay [4, 5], HR departments are actively looking for ways to adopt analytical techniques to execute activities where current evidence-based techniques are supplemented with data-driven insights. Consequently, HR analytics as a discipline and research field is strongly gaining importance. However, despite the notable rising enthusiasm for HR analytics, the adoption rate of insights from the scientific literature to practice in businesses remains limited [6].

As organizations are moving away from rigid structures and are delayering hierarchy, career ladders that steer internal flows of employees become more and more blurry [7, 8]. Therefore, the traditional internal market organization theories, which mainly focus on economic and institutional factors [9], are being challenged by more data-driven mechanisms with the help of new electronic human resource management (e-HRM) regimes [10, 11]. More and more, organizations are looking into data-driven decision support on how employees can be mobilized within an organization to help plot a path for changing careers or identifying how to move forward in their current career path [12].

Given the longitudinal structure of HR data that spans multiple years, a series of subsequent activities can be defined per employee. We regard the career of an employee within an organization as a sequence of activities as is done in the domain of process analytics [13]. Hence, the HR data can be transformed into the format of an event log. Events refer to the execution of such activities, resulting in start, completion, and/or cancellation recordings in an event log.The process perspective, resulting in a comprehensive end-to-end view, offers a dynamic approach that corresponds well with the increasing complexity of career paths. Appendix A displays an employee journey map, derived from an event log.

We focus on the career paths of employees within an organization by predicting a performance score between an employee and a job position with the aid of recommender systems (RSs), where matches are rated with a score y (see Section 4). Translated to the context of HR, RSs can be deployed to propose fitting jobs to employees and vice versa. We start with a collaborative filtering (CF) approach, which recommends matches based on the performance similarity with other employees. To integrate personal employee information like education and field of study into the RS, we introduce social regularization.

2 Motivation, related work, and contributions

Motivation. Consider the example in Figure 1, representing a simplified employee journey map where three employees can *visit* eight possible functions, resulting in three separate career paths. A more realistic employee journey map can be found in Appendix A. Each career path is characterized by the jobs an employee has held and currently holds within an organization, supplemented with individual, employee-specific data. For each job-employee combination, we observe a performance score y (see Section 4). Employee 1 covers the sequential trace of functions 1, 3, and 5 with respective observed performance scores of 0.4, 0.8, and 0.5. Our recommendation system proposes a sensible next step in this employee's career path by ranking all possible job-employee matches and selecting the one with the highest predicted score $\hat{y}$.

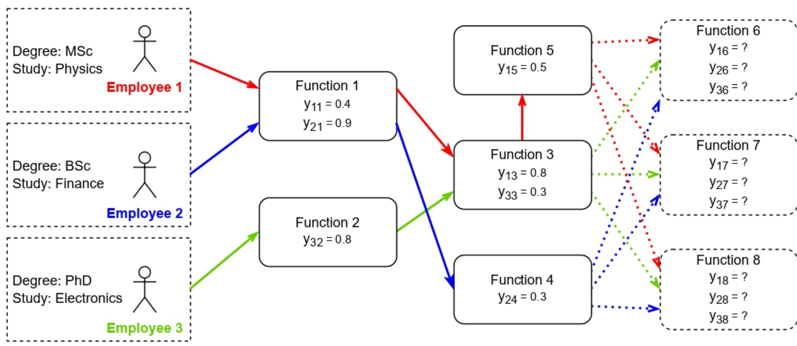

Fig. 1: A simplified example of three careers from a process point of view.

Related work. In this work, we focus on data-driven decision support for internal employee placement. We look at predictive job-employee matching in a post-hire setting where we consider careers from a process perspective. To the best of our knowledge, this has not been studied before.

Human Resource Analytics (HRA) as a term is relatively new, only appearing in the HR literature in 2003-2004 [14]. HR analytics as a whole cover a broad variety of subjects, including descriptive, predictive, and prescriptive analyses. A taxonomy of research topics is given by [14] and [15].

In the field of HRA, analytics are often used for the prediction of employee performance, employee turnover, the design of retention strategies, job-employee matching, and simulating recruiters' decisions. Research on performance prediction in the setting of recruitment and selection has been done by [16–25]. Several data mining and machine learning approaches to predict employee turnover have been proposed by [19, 26–29]. As an extension, [16] also come up with data-driven retention policies. A literature review is provided by [30]. [31] combine the prediction of performance and turnover simultaneously.

A good overview of job-employee matching can be found in reference works such as [31–34]. However, most work focuses on pre-hire matching in the context of recruitment and selection. Research on job-employee matching through social network information and based on resumes is done by [24, 35–39]. Research on job recommender systems has been can be found in the work of [32–34, 40–42].

Contributions. We develop an internal recommendation system for organization-specific internal employee placement.To provide decision support, we propose a job-employee matching method to manage the internal placement of employees and guide career path decisions. We look at careers from a longitudinal perspective, starting from data in the format of an event log, supplemented with personal employee information and a performance score for each observed job-employee match. We start with a collaborative filtering approach, which recommends job-employee matches based on the similarity of performance with other employees. To integrate information in the recommender system captured by personal employee data like the education and field of study, we introduce a social regularization term.

3 Methodology

We denote job-employee matches with their corresponding outcome in the format of an event log, where a case represents an employee, an activity a function, and a trace an employee journey within an organization. Formally, we have access to $\mathcal{D}=\{(u_k,t_k^s,t_k^e,v_k,\mathbf{x}_k,y_k,):k=1,...,K\}$. An example of $\mathcal{D}$ can be found in Table 1. Each tuple $(u_k,t_k^s,t_k^e,v_k,\mathbf{x}_k,y_k)$ represents a job-employee match with employee u_k holding job v_k from time t_k^s to time t_k^e with an observed outcome y_k. Employee u_k has features $\mathbf{x}_k$ which include amongst others, dependent on the dataset, the branch of study, degree, age, and gender. In total, we observe K job-employee matches multiple tuples can refer to the same employee.

Personalized RSs provide suggestions to a user based on their profile. In the setting of this research, an RS provides a relevant next step in a career to an employee, based on their profile. An employee profile $\mathcal{D}_i \subseteq \mathcal{D}$ of person u_i consists of the combination of H_i tuples where H_i is the number of jobs this employee has occupied within this organization. Hence, H_i tuples $\{(u_h,t_h^s,t_h^e,v_h,\mathbf{x}_h,y_h,):h=1,...,H_i\}$ contribute to one employee profile. Additionally, we assume that each unique job v can be executed at most once by each employee u_i. Consequently, an employee profile $\mathcal{D}_i$ consists of (i) the *visited* jobs with their corresponding performance score y and (ii) personal information $\mathbf{x}_h$.

u	t^s	t^e	v	x_1	x_2	x_3	x_4	x_5	y	
1	10/2014	06/2016	Function 1	MSc	Physics	F	1975	1	0.4	
1	07/2016	02/2019	Function 3	MSc	Physics	F	1975	1	0.8	$\mathcal{D}_1$
1	03/2019	07/2022	Function 5	MSc	Physics	F	1975	1	0.5	
2	09/2009	02/2016	Function 1	BSc	Finance	M	1981	0.8	0.9	$\mathcal{D}_2$
2	03/2016	07/2022	Function 4	BSc	Finance	M	1981	0.8	0.3	
3	06/2016	03/2019	Function 2	PhD	Electronics	M	1977	1	0.8	$\mathcal{D}_3$
3	04/2019	07/2022	Function 3	PhD	Electronics	M	1977	1	0.3	
4	...	...	...	...	...	...	...	...	...	

Table 1: Synthetic example data $\mathcal{D}$ in the format of an event log. The visual representation of these career paths is depicted in Figure 1.

Collaborative filtering. Typically, two types of collaborative filtering (CF) algorithms are commonly used: neighborhood-based and model-based. Neighborhood-based approaches focus on the similarity between either users or items, whereas model-based approaches start from the user-item rating matrix R. The latter category includes the latent factor approach which we apply and further extend in this work.

CF starts from a matrix $R \in \mathbb{R}^{m\times n}$ describing the outcome of m employees on n jobs which can be directly derived from $\mathcal{D}$. An example is given in Tables 1 and 2. We observe K job-employee matches, which results in matrix R with a sparsity of $(1-\frac{K}{m\times n})$. The latent factor approach aims to factorize this matrix R by two matrices $U \in \mathbb{R}^{l\times m}$ and $V \in \mathbb{R}^{l\times n}$ with $l<\min(m,n)$.

u \ v	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
1	0.4		0.8		0.5	...
2	0.9			0.3		...
3		0.8		0.3		...
4	...	...	...	...	...	...

Table 2: Synthetic example data in the format of observed job-employee rating matrix $R^{m\times n}$, derived from $\mathcal{D}$.

Equation 1 represents the loss function $\mathcal{L}$ which is minimized by using gradient descent to obtain two matrices U and V [43, 44]. Ultimately, with these two matrices, the matrix $\hat{R}=U^TV$ with a predicted outcome $\hat{y}_{ij}$ for each combination of u_i and v_j is calculated.

$$\min_{U,V}\mathcal{L}(R,U,V)=\underbrace{\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}\left(R_{ij}-U_i^TV_j\right)^2}_{\textbf{(i)}}+\underbrace{\frac{\lambda_1}{2}\|U\|_F^2+\frac{\lambda_2}{2}\|V\|_F^2}_{\textbf{(ii)}} \tag{1}$$

Term (i) expresses the difference between the observed outcomes in matrix R and the reconstructed outcomes U^TV. I_{ij} is the indicator function that takes value 1 if employee u_i held position v_j in the past and takes value 0 otherwise. Term (ii) consists of two regularization terms with hyperparameters λ_1 and λ_2 to avoid overfitting. $\|\cdot\|_F^2$ denotes the Frobenius form. The complete algorithm of default CF and its implementation can be found in Appendix B.

Collaborative filtering with social regularization. The basic version of CF does, however, have its shortcomings. The algorithm only learns similarity through the unordered set of job positions that employees held in the past. Consequently, the basic version of CF does not take into account personal information $\mathbf{x}_i$. To overcome this shortcoming, we extend the basic methodology of collaborative filtering through matrix factorization with social regularization as introduced by [45]. Social regularization stems from the idea that users of a system attach more importance to the recommended items of friends they trust and are close to in a social network. In this work, we make an abstraction of trust, friends, or even proximity in a social network and replace a list of friends with a list of similar employees based on $\mathbf{x}$.

For each employee u_i, personal data is stored in $\mathbf{x}_i$. Dependent on the dataset, this can include the branch of study, degree, date of birth, full-time equivalent, type of contract, location of employment, and marital status. Some features are numerical, others are categorical. With this set of personal information, similarities between different employees to identify resembling peers are calculated.

The more similar two employees u_i and u_p are, e.g. by having the same field of study or degree, the more similar their latent representations U_i and U_p should be as it is expected that the more similar two employees are, the more similar their performance will be in a certain position. We enforce this by introducing a social regularization term that compares one employee u_i and their $m-1$ peers u_p individually, given by **(iii)** in Equation 2 where $\beta>0$.

$Sim(i,p)$ is some similarity metric that takes as input the personal information of two employees and handles the mix of numerical and categorical components by combining and weighting a numerical and categorical metric: $\mathrm{Sim}(\mathbf{x},\mathbf{y})=\gamma\cdot\mathrm{NumSim}(\mathbf{x}^{num},\mathbf{y}^{num})+(1-\gamma)\cdot\mathrm{CatSim}(\mathbf{x}^{cat},\mathbf{y}^{cat})$ where $\mathbf{x}^{num}$ are the numerical and $\mathbf{x}^{cat}$ the categorical variables of vector $\mathbf{x}$. For NumSim we use the cosine similarity. For CatSim we use the overlap measure between two categorical vectors, which will be directly proportional to the number of attributes in which they have an equal value [46]. γ is set to the fraction of numerical features. A large value of $Sim(i,p)$ indicates that the distance between feature vectors U_i and U_p should be low and vice versa.

The complete loss function, Equation 2, consists of three terms. The social regularization term is labeled by **(iii)**. The algorithm of CF with SR can be found in Appendix C.

$$
\begin{aligned}
\min_{U,V} \ \mathcal{L}(R,U,V) = & \underbrace{\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}\left(R_{ij}-U_i^T V_j\right)^2}_{\textbf{(i)}} \\
& + \underbrace{\frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|V\|_F^2}_{\textbf{(ii)}} + \underbrace{\frac{\beta}{2}\sum_{i=1}^{m}\sum_{p=1}^{m}\mathrm{Sim}(i,p)\|U_i - U_p\|_F^2}_{\textbf{(iii)}}
\end{aligned}
\tag{2}
$$

4 Experimental evaluation

Data. We have access to three real-life datasets in the format of an event log. Formally, each tuple $(u_k, t_k^s, t_k^e, v_k, \mathbf{x}_k, y_k)$ represents a job-employee match. The observed performance score $y \in [0,1]$ is a weighted score that combines multiple criteria. The criterion as a function of lead time is sigmoid shaped where an ideal score is reached when the employee stays in a position for at least three years to account for onboarding costs. Other criteria include the personal fit in a team. How this observed y is calculated, e.g. the details on the ideal lead time, is the result of industry feedback by the providers of the datasets.

We have access to three private datasets extracted from an HRIS. An concise overview is given by Table 3. Dependent on the dataset, available features in $\mathbf{x}$ contain information on degree, branch of study, contract type, location of employment, and full-time equivalent (FTE). The values for these features change over time only for a small fraction of observations. In these cases, we take into account the last observed values, since the dataset is split in an out-of-time manner for validation and testing. Available features that are not taken into account are nationality, marital status, residence zip code, and gender.

	Sector	#Employees	#Jobs	#Matches	Timeframe	$R^{m\times n}$ sparsity
Dataset 1	High-tech R&D	±3000	±250	5062	2009-2021	99.3%
Dataset 2	IT Services	±4500	±200	11327	2012-2022	98.7%
Dataset 3	HR Services	±1500	±500	3792	2012-2021	99.5%

Table 3: Overview of datasets

Experiment setup. The dataset is split into a training, validation, and test set with proportions of 0.5/0.25/0.25 in an out-of-time way. Then we create a job-employee rating matrix $R^{m\times n}$, based on the performance scores of the observed matches. Table 2 shows a simplified example. Next, the following hyperparameters are tuned on an out-of-time validation set: learning rate α, regularization parameters λ_1 and λ_2, learning steps n, social regularization parameter β, and the dimensionality L of the latent factor representation. The tuning of these hyperparameters is done separately for each dataset.

We consider metrics for the estimated performances $\hat{y}$ (MAE and RMSE) and rank-correlation measures to assess the relative ranking of proposed job-employee matches (Spearman and Kendall). The logic for the latter is that the expected performance of matches can be compared relative to other proposed matches, rather than looking at the predicted scores themselves. To be able to make a ranking, only observations withs more than two matches are taken into account. A higher score is better.

Results. We test the performance of collaborative filtering (CF) without and with social regularization ($CF+SR$) on three real-life datasets. We compare this to the case where predictions of y are drawn randomly from a uniform distribution $\hat{y} \in [0,1]$ and to the case where all predictions of y are set to 0.5. Table 4 summarizes the results.

		CF	$CF+SR$	*Random* $\hat{y}$	$\hat{y}=0.5$
Dataset 1	MAE	0.2346	0.2083	0.3221	**0.2058**
	$RMSE$	0.2909	**0.2517**	0.3892	0.2543
	Spearman	0.2327	**0.3897**	-0.0432	0
	Kendall	0.1621	**0.3197**	-0.0288	0
Dataset 2	MAE	0.2242	**0.2099**	0.368	0.286
	$RMSE$	0.2846	**0.2577**	0.4416	0.3241
	Spearman	0.0310	**0.1646**	-0.0936	0
	Kendall	0.0458	**0.1582**	-0.1191	0
Dataset 3	MAE	0.1971	**0.1720**	0.3552	0.2775
	$RMSE$	0.2464	**0.2264**	0.4338	0.3188
	Spearman	0.2022	**0.2405**	0.0007	0
	Kendall	0.1908	**0.1949**	0.0281	0

Table 4: Summary of results. for MAE and $RMSE$, a lower score is better. For *Spearman* and *Kendall*, a higher score is better.

For every dataset, adding an SR term improves the performance of collaborative filtering. It is important to check for multiple metrics. For dataset 1, a fixed prediction of $\hat{y}=0.5$ performs surprisingly well in terms of MAE and $RMSE$, but fails to predict the ranking of the matches. For dataset 2, the addition of an SR term barely improves MAE and $RMSE$, but strongly corrects the weak performance of CF's ability to rank.

5 Conclusion

Analytics have great transformative potential in HR decision support. To provide this support, we approach careers from a process perspective which offers dynamic modeling techniques in correspondence with the increasing non-linearity of career paths. In this article, we present a recommender system to manage internal staffing by proposing job-employee matches. We use collaborative filtering as a baseline method, which we then extend with social regularization. This allows for the inclusion of other sources of data than the set of jobs an employee held by calculating similarities between different employees based on personal information. Adding the social regularization term to the loss function results in improved performance in comparison with the default collaborative filtering approach.

Future work will focus on mitigating the effect of selection bias, i.e., controlling for the current HR policies that result in the non-random selection of the next functions in a career path. This issue could be solved with causal methods.

6 Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This project was funded by the Flemish Government, through Flanders Innovation Entrepreneurship (VLAIO, project HBC.2021.0833).

References

1. Foster Provost and Tom Fawcett. Data science and its relationship to big data and Data-Driven decision making. *Big Data*, 1(1):51–59, March 2013.
2. Clyde Holsapple, Anita Lee-Post, and Ram Pakath. A unified foundation for business analytics. *Decis. Support Syst.*, 64:130–141, August 2014.
3. David Angrave, Andy Charlwood, Ian Kirkpatrick, Mark Lawrence, and Mark Stuart. HR and analytics: why HR is set to fail the big data challenge. *Hum. Resour. Manag. J.*, 26(1):1–11, January 2016.
4. Vicenc Fernandez and Eva Gallardo-Gallardo. Tackling the HR digitalization challenge: key factors and barriers to HR analytics adoption. *Copmetitiveness Rev. J.*, 31(1):162–187, July 2020.
5. John Boudreau and Wayne Cascio. Human capital analytics: why are we not there? *Journal of Organizational Effectiveness: People and Performance*, 4(2):119–126, January 2017.
6. Thomas Rasmussen and Dave Ulrich. Learning from practice: how HR analytics avoids being a management fad. *Organ. Dyn.*, 44(3):236–242, July 2015.
7. David G Collings, Kamel Mellahi, and Wayne F Cascio. *The Oxford Handbook of Talent Management*. Oxford University Press, 2017.
8. Peter Cappelli. Talent on demand – managing talent in an age of uncertainty. 25, 02 2009.
9. Paul Osterman. Choice of employment systems in internal labor markets. *Industrial Relations: A Journal of Economy and Society*, 26(1):46–67, 1987.
10. Patrick Gavan O'Shea and Kerrin E Puente. How is technology changing talent management? In David G Collings, Kamel Mellahi, and Wayne F Cascio, editors, *The Oxford Handbook of Talent Management*. Oxford University Press, September 2017.
11. Philip Rogiers, Stijn Viaene, and Jan Leysen. The digital future of internal staffing: A vision for transformational electronic human resource management. *Intell. Syst. Account. Finance Manag.*, 27(4):182–196, October 2020.
12. Nsf career compass challenge. https://www.challenge.gov/?challenge=nsf-career-compass-challenge. Accessed: 2022-06-16.
13. Wil Van Der Aalst. Process mining. *Commun. ACM*, 55(8):76–83, August 2012.
14. Janet H Marler and John W Boudreau. An evidence-based review of HR analytics. *int. j. hum. resour. manag.*, 28(1):3–26, January 2017.
15. José Garcia-Arroyo and Amparo Osca. Big data contributions to human resource management: a systematic review. *The International Journal of Human Resource Management*, 32(20):4337–4362, November 2021.
16. Edouard Ribes, Karim Touahri, and Benoît Perthame. Employee turnover prediction and retention policies design: a case study. July 2017.
17. Kuo-Yan Wang and Hsien-Yu Shun. Applying back propagation neural networks in the prediction of management associate work retention for small and medium enterprises. *Univers. J. Manag.*, 4(5):223–227, May 2016.
18. Liu S., Yang Y., and Liu S., editors. *A decision tree applied to the grass-roots staffs' turnover problem - Take C-R Group as an example*, volume 2015-October. IEEE Computer Society, 2015.
19. Rohit Punnoose and Pankaj Ajit. Prediction of employee turnover in organizations using machine learning algorithms. *Int. j. adv. res. artif. intell.*, 5(9), 2016.
20. E Sikaroudi, A Mohammad, R Ghousi, and A Sikaroudi. A data mining approach to employee turnover prediction (case study: Arak automotive parts manufacturing). *Int. J. Ind. Syst. Eng.*, 8(4):106–121, 2015.
21. Xude Gui, Zhongyi Hu, Jinlong Zhang, and Yukun Bao. Assessing personal performance with M-SVMs. In *2014 Seventh International Joint Conference on Computational Sciences and Optimization*, pages 598–601, July 2014.
22. Chin-Yuan Fan, Pei-Shu Fan, Te-Yi Chan, and Shu-Hao Chang. Using hybrid data mining and machine learning clustering analysis to predict the turnover rate for technology professionals. *Expert Syst. Appl.*, 39(10):8844–8851, August 2012.

23. Chen-Fu Chien and Li-Fei Chen. Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Syst. Appl.*, 34(1):280–290, January 2008.
24. Yung-Ming Li, Cheng-Yang Lai, and Chien-Pang Kao. Incorporate personality trait with support vector machine to acquire quality matching of personnel recruitment. In *4th international conference on business and information*, pages 1–11. iebi.nctu.edu.tw, 2008.
25. Sameep Mehta, Rakesh Pimplikar, Amit Singh, Lav R Varshney, and Karthik Visweswariah. Efficient multifaceted screening of job applicants. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 661–671, New York, NY, USA, March 2013. Association for Computing Machinery.
26. Sepideh Hassankhani Dolatabadi and Farshid Keynia. Designing of customer and employee churn prediction model based on data mining method and neural predictor. In *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*. IEEE, July 2017.
27. Evy Rombaut and Marie-Anne Guerry. Predicting voluntary turnover through human resources database analysis. *Manag. Res. Rev.*, 41(1):96–112, January 2018.
28. V Vijaya Saradhi and Girish Keshav Palshikar. Employee churn prediction. *Expert Syst. Appl.*, 38(3):1999–2006, March 2011.
29. Yue Zhao, Maciej K Hryniewicki, Francesca Cheng, Boyang Fu, and Xiaoyu Zhu. Employee turnover prediction with machine learning: A reliable approach. In *Advances in Intelligent Systems and Computing*, Advances in intelligent systems and computing, pages 737–758. Springer International Publishing, Cham, 2019.
30. Ardhianiswari D Ekawati. PREDICTIVE ANALYTICS IN EMPLOYEE CHURN: A SYSTEMATIC LITERATURE REVIEW. *Journal of Management Information and Decision Sciences; Weaverville*, 22(4):387–397, 2019.
31. Dana Pessach, Gonen Singer, Dan Avrahami, Hila Chalutz Ben-Gal, Erez Shmueli, and Irad Ben-Gal. Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming. *Decis. Support Syst.*, 134:113290, July 2020.
32. Shaha T Al-Otaibi and Mourad Ykhlef. A survey of job recommender systems. *IJPS*, 7(29):5127–5142, July 2012.
33. Zheng Siting, Hong Wenxing, Zhang Ning, and Yang Fan. Job recommender systems: A survey. In *2012 7th International Conference on Computer Science & Education (ICCSE)*, pages 920–924, July 2012.
34. Mauricio Noris Freire and Leandro Nunes de Castro. e-recruitment recommender systems: a systematic review. *Knowl. Inf. Syst.*, 63(1):1–20, January 2021.
35. Andrzej Janusz, Sebastian Stawicki, Michał Drewniak, Krzysztof Ciebiera, Dominik Ślezak, and Krzysztof Stencel. How to match jobs and candidates - a recruitment support system based on feature engineering and advanced analytics. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, pages 503–514. Springer International Publishing, 2018.
36. Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, and Rocky Bhatia. A machine learning approach for automation of resume recommendation system. *Procedia Comput. Sci.*, 167:2318–2327, January 2020.
37. Nimet Tülümen, Gökhan Akgün, Ali Nohutçu, Günnur Sevgi Aktoros Genç, and Serdar Genç. Hybrid job and resume matcher. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 163–168, September 2021.
38. Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. An enhanced neural network approach to Person-Job fit in talent recruitment. *ACM Trans. Inf. Syst. Secur.*, 38(2):1–33, February 2020.
39. Le Zhang, Ding Zhou, Hengshu Zhu, Tong Xu, Rui Zha, Enhong Chen, and Hui Xiong. Attentive heterogeneous graph embedding for job mobility prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 2192–2201, New York, NY, USA, August 2021. Association for Computing Machinery.

40. Jacob Bollinger, David Hardtke, and Ben Martin. Using social data for resume job matching. In *Proceedings of the 2012 workshop on Data-driven user behavioral modelling and mining from social media*, DUBMMSM '12, pages 27–30, New York, NY, USA, October 2012. Association for Computing Machinery.
41. Corné de Ruijt and Sandjai Bhulai. Job recommender systems: A review. November 2021.
42. Anna Giabelli, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Andrea Seveso. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Appl. Soft Comput.*, 101:107049, March 2021.
43. Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering, 2007. Accessed: 2022-6-15.
44. Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Major components of the gravity recommendation system. *SIGKDD Explor. Newsl.*, 9(2):80–83, December 2007.
45. Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 287–296, New York, NY, USA, February 2011. Association for Computing Machinery.
46. Craig Stanfill and David Waltz. Toward memory-based reasoning. *Commun. ACM*, 29(12):1213–1228, December 1986.

Appendix A Employee journey map

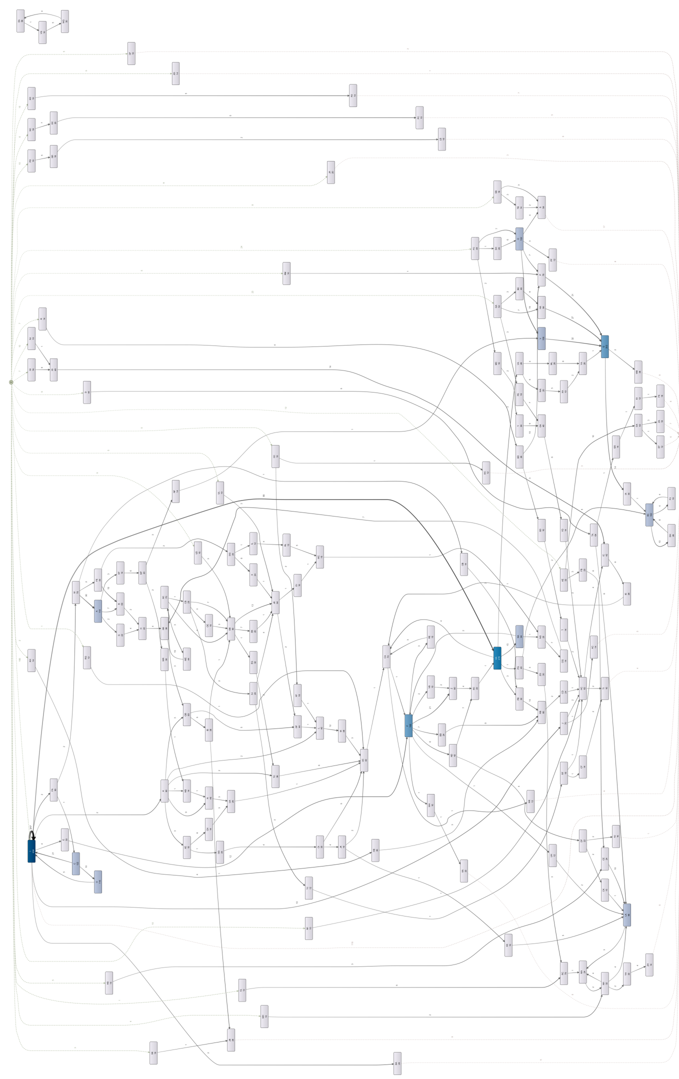

Fig. 2: An employee journey map of dataset 1 displayed as a directly-follows graph. Each rectangle corresponds to a function and each arc to a possible transition between functions. This representation is simplified, as only 30% of functions and 10% most frequent transitions are displayed. The full employee journey map has a higher complexity.

Appendix B Algorithm Collaborative Filtering

Algorithm 1: Collaborative Filtering

Input : observed ratings R, empty matrices U and V, learning rate α, regularization parameters λ_1 and λ_2, learning steps n, stopping threshold t

Output : matrix $\hat{R}=U^TV$ with estimated ratings

for *steps*$=1,2,...,n$ **do**

  for *each element* $R_{i,j}$ **do**

    if $R_{i,j}>0$ **then**

      $e_{i,j}:=R_{i,j}-\hat{R}_{i,j}$; ▷ calculate error

      for *each user* i **do**

        $U_i:=U_i+\alpha(\underbrace{e_{i,j}V_j}_{(i)}+\underbrace{\lambda_1 U_i}_{\text{(ii)}})$; ▷ update vector U_i

      for *each job* j **do**

        $V_j:=V_j+\alpha(\underbrace{e_{i,j}U_i}_{\text{(i)}}+\underbrace{\lambda_2 V_j}_{\text{(ii)}})$; ▷ update vector V_j

  $$e:=\underbrace{\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}\left(R_{ij}-U_i^TV_j\right)^2}_{\text{(i)}}+\underbrace{\frac{\lambda_1}{2}\|U\|_F^2+\frac{\lambda_2}{2}\|V\|_F^2}_{\text{(ii)}} ;$$

  if *error* $e<t$ **then**

    break ; ▷ stop if error falls below threshold t

Return : $\hat{R}=U^TV$

Appendix C Collaborative Filtering with Social Regularization

Algorithm 2: Collaborative Filtering with Social Regularization

Input : observed ratings R, empty matrices U and V, learning rate α, regularization parameters λ_1 and λ_2, learning steps n, stopping threshold t, social regularization parameter β, similarity metric Sim

Output : matrix $\hat{R}=U^T V$ with estimated ratings

for *steps* $=1,2,...,n$ **do**

  for *each element* $R_{i,j}$ **do**

    if $R_{i,j} > 0$ **then**

      $e_{i,j} := R_{i,j} - \hat{R}_{i,j}$; `> calculate error`

      for *each user* i **do**

        $\gamma_i = \sum_{p=1}^{m} \mathrm{Sim}(i,p)(U_i - U_p)$; `> social regularization`

        $U_i := U_i + \alpha(\underbrace{e_{i,j} V_j}_{\text{(i)}} + \underbrace{\lambda_1 U_i}_{\text{(ii)}} + \underbrace{\beta \gamma_i}_{\text{(iii)}})$; `> update vector` U_i

      for *each job* j **do**

        $V_j := V_j + \alpha(\underbrace{e_{i,j} U_i}_{\text{(i)}} + \underbrace{\lambda_2 V_j}_{\text{(ii)}})$; `> update vector` V_j

  $$e := \underbrace{\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \left(R_{ij} - U_i^T V_j\right)^2}_{\text{(i)}} + \underbrace{\frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2}_{\text{(ii)}} + \underbrace{\frac{\beta}{2} \sum_{i=1}^{m} \sum_{p=1}^{m} \mathrm{Sim}(i,p) \|U_i - U_p\|_F^2}_{\text{(iii)}}$$

  if *error* $e < t$ **then**

    break ; `> stop if error falls below threshold` t

Return : $\hat{R}=U^T V$