Career Path Recommendations for Long-term Income Maximization: A Reinforcement Learning Approach

Spyros Avlonitis[1,†], Dor Lavi[2], Masoud Mansoury[3,4] and David Graus[5]

[1] *KLM Royal Dutch Airlines, Amstelveen, The Netherlands*
[2] *Meta, Amsterdam, The Netherlands*
[3] *University of Amsterdam, Amsterdam, The Netherlands*
[4] *Discovery Lab, Elsevier, Amsterdam, The Netherlands*
[5] *Randstad, Diemen, The Netherlands*

Abstract

This study explores the potential of reinforcement learning algorithms to enhance career planning processes. Leveraging data from Randstad The Netherlands, the study simulates the Dutch job market and develops strategies to optimize employees' long-term income. By formulating career planning as a Markov Decision Process (MDP) and utilizing machine learning algorithms such as Sarsa, Q-Learning, and A2C, we learn optimal policies that recommend career paths with high-income occupations and industries. The results demonstrate significant improvements in employees' income trajectories, with RL models, particularly Q-Learning and Sarsa, achieving an average increase of 5% compared to observed career paths. The study acknowledges limitations, including narrow job filtering, simplifications in the environment formulation, and assumptions regarding employment continuity and zero application costs. Future research can explore additional objectives beyond income optimization and address these limitations to further enhance career planning processes.

Keywords

Career Path Recommendation, Machine Learning, Career Planning, Income Optimization, Employee Development, Markov Decision Process (MDP), Reinforcement Learning (RL)

1. Introduction

The importance of career planning in shaping an individual's professional journey cannot be overstated. It involves strategic decision-making related to one's career goals, which may be as diverse as the individuals themselves. However, despite varied ambitions, a proactive approach towards career planning universally benefits all, allowing individuals to align their career trajectory with their objectives, such as maximizing lifetime income. Recognizing that reality often presents multifaceted goals and constraints, this paper aims to simplify the career planning process using the power of artificial intelligence.

The efficacy of career planning significantly depends on the insight one has into potential career paths and their expected rewards. This study leverages AI to provide such insights to employees. Collaborating with Randstad, a global leader in the HR services industry, this paper harnesses a vast array of data encompassing anonymized employee profiles, job applications, and salary information. Using machine learning, we simulate the Dutch job market and employ reinforcement learning to strategize for maximizing employees' long-term income. Although income is not the sole objective for everyone, this research assumes it as the sole optimizing objective for simplicity. However, the proposed framework is flexible and can accommodate other objectives, such as job satisfaction or a mix of objectives, given the availability of relevant data.

The primary goal is to design a system that uses an employee's work experience as input to recommend a career path, a series of occupations and industries, that on average, delivers the highest income over ten years. A ten-year timescale is selected under the assumption that job market dynamics remain unpredictable beyond this period, making further recommendations potentially unreliable. It is important to note that the suggested career paths should be practical, implying a high likelihood of hiring should employees opt to pursue them.

2. Background

In this section, we briefly describe general reinforcement learning architecture and review the literature on career path recommendations.

RecSys in HR'23: The 3rd Workshop on Recommender Systems for Human Resources, in conjunction with the 17th ACM Conference on Recommender Systems, September 18–22, 2023, Singapore, Singapore.

[†] Work done while on internship at Randstad Groep Nederland.

spyrosavl@gmail.com (S. Avlonitis); dor.la.vie@gmail.com (D. Lavi); m.mansoury@uva.nl (M. Mansoury); david.graus@randstadgroep.nl (D. Graus)

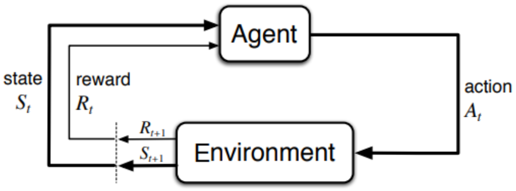

Figure 1: The agent-environment interaction in a Markov decision process.

2.1. Reinforcement Learning

Reinforcement Learning (RL), as characterized by Sutton and Barto [1], is a decision-making paradigm adept at handling tasks with potential delayed outcomes, such as career planning. Unlike other learning strategies, RL operates on trial-and-error, aiming to optimize a specific metric without any direct instruction. It involves an agent navigating an environment to maximize a cumulative reward over time. The RL system incorporates six primary elements: the **Agent** that interacts with the environment based on its policy, the **Environment** providing feedback, the **State** and **Action** representing the environment, and the choices available, the **Reward** as a numerical feedback, and the **Policy** directing the agent's actions.

2.1.1. Markov Decision Processes

The application of RL to career planning necessitates formulating the problem as a Markov Decision Process (MDP), as suggested by Puterman [2]. This enables us to leverage established RL research and precise theoretical results. MDPs formalize sequential decision-making where actions influence not only immediate rewards but also future states, and by extension, future rewards. The inherent Markov property in an MDP posits that the transition probabilities to a new state depend solely on the current state and action.

2.2. Recommender Systems in Human Resources

Historically, research into workforce mobility and career development has utilized traditional data sources such as surveys and censuses, as noted by Topel and Ward [3] and Long and Ferrie [4]. However, the rise of Online Professional Networks (OPN) has allowed for the employment of data-driven machine learning methods. The focus has increasingly shifted towards modelling career paths to predict mobility and aid in career development. This has proved valuable for both employers and employees, facilitating strategic decision-making in hiring and career progression.

Several studies have taken varied approaches to this issue. For instance, Paparrizos et al. [5] employed a naive Bayes model to predict job transitions, while Wang et al. [6] used a proportional hazards model to estimate when employees might decide to change jobs. Further, Liu et al. [7] explored career path prediction using social network data, while Li et al. [8] introduced the NEMO model for predicting future company and job titles using Long Short-Term Memory (LSTM) networks.

The advent of more complex models has also been witnessed. Meng et al. [9] used a hierarchical neural network with an embedded attention mechanism, and Xu et al. [10] performed a talent flow analysis for predicting the increments in a dynamic job transition network. Other models, like the one proposed by Liu and Tan [11], utilized logistic regression to predict career choices, while Al-Dossari et al. [12] proposed a recommendation system for IT graduates based on skill similarity.

A separate line of research rejects the notion that frequently observed paths are necessarily the most beneficial. Lou et al. [13] recommended the shortest career path using a Markov Chain model, whereas Oentaryo et al. [14] focused on achieving the best payoff trade-off in career path planning. Shahbazi et al. [15] optimized towards the career development of employees rather than productivity. Other approaches have included the use of skill graphs for transition pathway recommendations as demonstrated by Gugnani et al. [16] and Dawson et al. [17], and the use of reinforcement learning for dynamic career path recommendations as presented by Kokkodis and Ipeirotis [18]. Most recently, Guo et al. [19] proposed a reinforcement learning variant for optimizing career paths.

The research presented in this paper is similar to previous work such as that of Oentaryo et al. [14], Kokkodis and Ipeirotis [18], and Guo et al. [19]. Unlike Kokkodis and Ipeirotis [18], which studied online freelancers and projects, this paper focuses on long-term employment relationships. In contrast to the work of Oentaryo et al. [14] and Guo et al. [19], which do not incorporate monetary rewards, the focus here is to chart the optimal path for the highest long-term income. Also, where Guo et al. [19] posits any transition between jobs as possible, this study takes a more realistic approach and models transitions as a stochastic process learned from the data. Oentaryo et al. [14] also model transitions as a stochastic process but assume it to be memoryless, making a person's next job dependent only on their current job. In contrast, this study introduces two settings: a *naive setting* that makes the same assumption, and a *standard setting* that leverages employees' past experiences to predict their next career move.

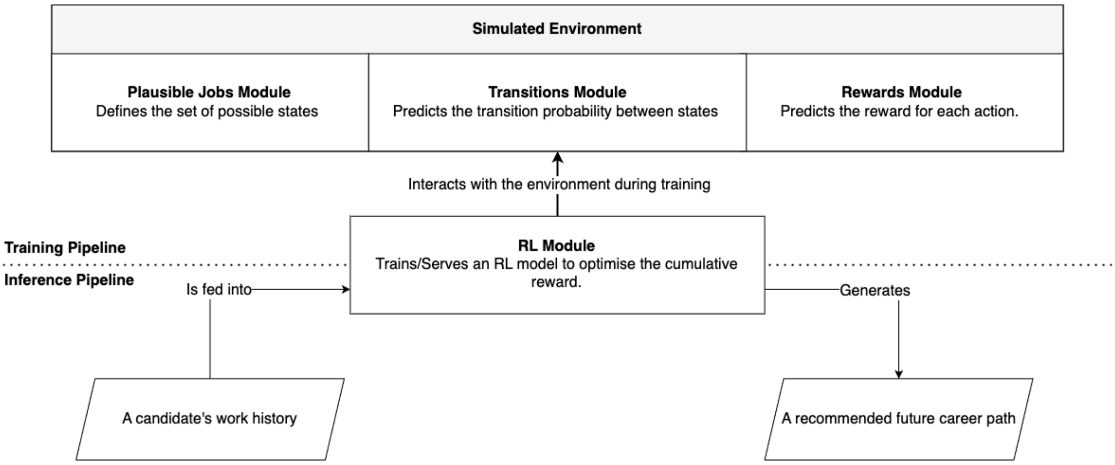
Simulated Environment
Plausible Jobs Module
Defines the set of possible states
Transitions Module
Predicts the transition probability between states
Rewards Module
Predicts the reward for each action.
Interacts with the environment during training
Training Pipeline
Inference Pipeline
RL Module
Trains/Serves an RL model to optimise the cumulative reward.
Is fed into
Generates
A candidate's work history
A recommended future career path

Figure 2: The architecture of the proposed career path recommendation system.

3. The Proposed Career Path Recommendation Model

We consider the problem of recommending a sequence of jobs — a career path — to the candidates that, if followed, would maximize their earnings during their foreseeable future.

Formally, given $C = \{c_1, ..., c_n\}$ as n candidates and $J = \{j_1, ..., j_m\}$ as m jobs, we define R_c as the recommended career path generated for candidate c. We denote $W_c = \{w_{c,1}, ..., w_{c,k}\}$ as the work experience of candidate c, k different jobs that c worked in the past with $w_{c,1}$ being her first job and $w_{c,k}$ being her last (current) job. Each work experience contains information about the period (start date and end date) and the area (or role) that the candidate worked on in that job. The work area represents a high-level categorization for the jobs. In our experiments, we define it as a combination of an occupation and an industry. Examples are a Data Science role in Insurance or a Data Science role in Banking. Refer to Section 4.1 for more details. We also denote $App_c = \{app_{c,j_1}, ..., app_{c,j_l}\}$ as l jobs that candidate c applied for in the past in which their outcomes are either hired or rejected. Finally, we denote V as all the vacancies posted on the market.

Given these three input data (work experience W, job applications App, and vacancies V), our career path recommendation model comprises four distinct modules, as depicted in Figure 2.

The first three modules—**Plausible Jobs**, **Transitions**, and **Rewards**—simulate the job market environment. The fourth module employs Reinforcement Learning (**RL**) to learn an optimal strategy for navigating these environments.

Plausible Jobs Module An employee's state at any given time is characterized by his current job, which is defined as a combination of occupation and industry, along with their work history. This concept forms the basis of the state space defined by this module, which comprises the set of available jobs and industries the agent can occupy. However, due to the constraints imposed by the dataset and to ensure a computationally feasible environment, our experiments are restricted to the 142 most prevalent jobs present in our dataset.

Transition Module Job applications do not always have deterministic outcomes; similarly, the actions taken by the agent should not always have deterministic outcomes. When the agent applies for a job and succeeds in being hired, it transitions from the current state $\mathbf{s}$ to a new state $\mathbf{s'}$. If unsuccessful, it remains in the current state $\mathbf{s}$. This transition occurs with probability $P(\mathbf{s'}|\mathbf{a}, \mathbf{s})$. A Random Forest binary classifier is trained on the Job Application data and is used to predict the aforementioned probabilities. In other words, this module computes the transition probability between different jobs within the environment.

We consider the following approaches for computing the transition probabilities:

- **Last Job State Representation:** We assume that state $\mathbf{s'}$ only contains information about the last job of a person, implying that the probability of being hired depends solely on their latest job.
- **Full History State Representation:** Conversely, in the alternative approach, we assume that the state contains information about a person's entire work history. This second approach

is closer to reality, but it also greatly increases the size and complexity of the state space, which could make learning more challenging and could potentially suffer from a lack of data.

Reward Module After each transition, this module is used to compute the reward earned from that transition. We define the reward in the form of the estimated salary that the individual earns after the transition. We use a Random Forest regressor trained on V (i.e. all vacancies in the market) to predict the salary corresponding to each job. Given each $v \in V$ consists of the textual job description, and annual salary information, for our experiments we perform this prediction on a yearly basis for each pair of job role and industry.

Reinforcement Learning Module Lastly, the RL module uses RL algorithms to learn policies that can yield optimal rewards. After training, these models can be used to recommend high-income-yielding career paths to employees. We experiment with and compare multiple algorithms during the training of the RL module. The details of the algorithms are described in section 4.3.

4. Methodology

4.1. Datasets

We conducted our experiments on anonymized data provided by Randstad as follows:

Work Experience Dataset This tabular dataset consists of work experience items that employees may submit to Randstad either online or through consultants, or are directly taken from the administration of job placements made through Randstad. Relevant attributes for this research include: 1) Employee ID, 2) Job start and end dates, 3) ISCO code[1] (occupation identifier), and 4) SBI code[2] (industry identifier).

Almost all the work experience items (99.99%) pertain to Randstad placements, as these are jobs employees secured through Randstad. Most of the previous experiences (before using Randstad's services) are missing essential attributes.

Vacancies Dataset This dataset includes salary ranges for about six million vacancies, including their ISCO and SBI codes, posted on various Dutch websites. We use this dataset to estimate expected salaries for each occupation.

Job Applications Dataset This dataset contains information on job applications made by candidates to Randstad's vacancies, with the outcome of each application (hired or rejected) also available.

4.2. Data Preprocessing

During preprocessing, we filtered out employees with missing data, jobs with durations less than a week, and employees with more than fifty work experience items from the work experience dataset, yielding 200K employees with 400K work experience items.

In line with Randstad's business model, most placements are short-term or temporary jobs common in staffing, resulting in a mean job duration of 161 days and a median duration of 95 days.

The average annual salary in the vacancies dataset is approximately 42K euros, with a median salary of 38K euros.

4.3. Reinforcement Learning Algorithms

Our experiments employ various Reinforcement Learning (RL) algorithms, which are primarily categorized into tabular methods and approximate RL methods.

4.3.1. Tabular Methods

Tabular methods are a class of RL algorithms that work well with a discrete, small state-action space. They maintain a table of values, with each entry in the table representing the value of each possible state-action pair.

State–action–reward–state–action (Sarsa) Introduced by Rummery et al. [20], Sarsa is an on-policy, tabular, temporal difference (TD) method. TD learning, which is a hybrid of Monte Carlo and dynamic-programming ideas, can learn directly from raw experience without a model of the environment's dynamics. Like dynamic programming, TD methods update estimates based in part on other learned estimates, without waiting for a final outcome (they bootstrap). The *Sarsa* algorithm aims to learn an action-value function $q_\pi(s, a)$, providing the expected reward starting from state s, taking action a, and following the policy π.

Q-Learning Introduced by Watkins and Dayan [21], Q-Learning is another tabular TD method. However, Q-Learning is an off-policy method, where the learned action-value function, Q, directly approximates q_*, the optimal action-value function, regardless of the policy followed (behavior policy).

[1] ISCO Wikipedia page: https://en.wikipedia.org/wiki/International_Standard_Classification_of_Occupations

[2] SBI official website: https://www.kvk.nl/overzicht-standaard-bedrijfsindeling/

4.3.2. Approximate RL Methods

While tabular methods perform well in environments with a small number of state-action pairs, they face challenges when the state-action space becomes large or continuous. They are not able to efficiently store the value of every possible state-action pair, nor can they generalize the value of unvisited state-action pairs effectively. This is where approximate RL methods come in to help with the Full History State Representation. These methods use function approximation, typically employing neural networks, to estimate the value of state-action pairs, allowing them to handle environments with larger or more complex state spaces more effectively.

Deep Q-Learning (DQN) DQN is an off-policy approach introduced by Mnih et al. [22]. DQN is the first successful deep-learning model to learn control policies directly from high-dimensional sensory input using reinforcement learning. It utilizes a convolutional neural network trained with a variant of Q-learning, taking raw pixels as input and estimating future rewards through a value function.

Advantage Actor-Critic (A2C) A2C is an approximate solution RL method that utilizes deep reinforcement learning for function approximation. Unlike DQN, A2C is an on-policy method. Introduced by Mnih et al. [23], A2C is an actor-critic method, where the policy function is represented independently of the value function. The "critic" model estimates the value function and the "actor" learns the target policy. Both the Critic and Actor functions are parameterized with neural networks. As explained by Mnih et al. [23], the main advantage of A2C over DQN is its faster training speed.

4.4. Baselines

Besides the above RL algorithms, we also perform experiments using two naive action selection approaches as baselines:

Greedy Most Common Transition In this approach, the agent always applies for the job with the highest transition probability, that is the most likely job to be hired. In the case of multiple jobs with the same ranking, a random selection is made.

Greedy Highest Expected Reward In this strategy, the agent applies for the job with the maximum expected reward, defined as the product of the transition probability and the immediate salary after the transition. In reality, this signifies the job with the highest likelihood of both being attained and yielding the highest immediate income. As before, in the case of multiple top-ranking jobs, a random selection is made.

4.5. Evaluation Metrics

We assess the effectiveness of our methods based on the income difference between *observed career paths (factuals)* and *recommended career paths (counterfactuals).*

Observed Career Paths Using the Work Experience dataset, we generate a list of observed career paths and their corresponding income. Given that workers can hold multiple jobs simultaneously or have periods of unemployment, the dataset requires processing to align with the requirements of our simplified environment. Our models assume people only have one job at a time and there is no unemployment. Therefore, in cases of simultaneous employment, we estimate each job's monthly salary and assume the worker earned the mean salary. For periods of unemployment, we consider the salary from the worker's last job to be ongoing.

Counterfactual Career Paths After training each RL method, we sample observed career paths to generate their counterfactuals. These are the paths each model recommends, starting from the observed path's initial job, and lasting the same duration.

Reported metrics For each model under consideration, we report two primary quantities - the *Mean Factual* and *Mean Counterfactual* accumulated rewards. These metrics represent the mean income accumulated by employees in reality versus the projected income they would have earned in a counterfactual scenario respectively.

For an employee e their factual income denoted as FI, over their career of M months is calculated as

$$FI(e) = \sum_{m=1}^{M} I(J_e, m) \tag{1}$$

where $I(J_e, m)$ is a function that returns the salary that the employee e had earned by performing job J during the month m. Similarly, the counterfactual income is calculated as

$$CFI(e) = \sum_{m=1}^{M} I(J'_e, m) \tag{2}$$

where J' is the job that employee e would have performed if she had followed the recommendations of our system. Finally, the mean of these quantities is calculated over a sample of 20,000 observed (factual) and generated (counterfactual) career paths.

Following this, we present the *Change %*, illustrating the percentage change between the factual and counterfactual means. To determine the statistical significance

Model	Mean *FI* €	Mean *CFI* €	Change %	p-value	Gainers %	Mean Gain %	Losers %	Mean Loss %
Baseline: Most Common	90,283.42	89,644.81	-0.7	0.69	8.85	8.17	11.62	-8.40
Baseline: Highest Exp. Reward	90,283.42	89,434.75	-0.94	0.59	8.39	7.50	12.52	-8.48
Q-Learning	90,283.42	95,077.13	**5.3**	0.01	27.53	13.81	12.56	-7.63
Sarsa	90,283.42	94,836.08	**5.04**	0.01	32.84	11.50	10.95	-7.46

Table 1
Last Job State Representation: Factual vs Counterfactual career paths. Metrics described in Section 4.5.

Model	Mean *FI* €	Mean *CFI* €	Change %	p-value	Gainers %	Mean Gain %	Losers %	Mean Loss %
Baseline: Most Common	90,386.16	95,871.78	**6.18**	0.02	71.07	17.27	25.15	-13.39
Baseline: Highest Exp. Reward	90,386.16	161,774.45	**79.18**	0.00	96.01	80.37	0.04	-11.44
Deep Q-Learning	90,283.42	94,547.94	**4.7**	0.01	67.91	16.95	27.87	-13.71
A2C	90,283.42	95,616.29	**5.9**	0.00	70.82	17.22	25.35	-13.64

Table 2
Full History State Representation: Factual vs Counterfactual career paths. Metrics described in Section 4.5.

of the observed difference, we calculate a *p-value* using a two-sided permutation test with an alpha level of 0.05.

Furthermore, we detail the proportion of employees experiencing an income rise in the counterfactual world, referred to as *Gainers*, along with the average magnitude of their income change. Similarly, we present data for those experiencing a decline, termed as *Losers*, including the mean change in their income.

4.6. Experimental results

This subsection presents a detailed analysis of the experiment results. We juxtapose our factual and counterfactual career paths in terms of the mean income they generate. Additionally, we assess the effectiveness of our models by examining the percentage of gainers and losers as well as the magnitude of their respective income changes.

Table 1 presents the results for the Last Job State Representation, where the job seekers' state depends only on the last held job. From this table, we can observe that baselines do not perform significantly differently than the factual career paths, with differences under 1% (at -0.7% and -0.94% for Most Common and Highest Expected Reward baselines, respectively). However, Q-Learning and Sarsa models perform well with a notable percentage of income gainers (27.53% and 32.84% respectively) and a reasonable mean gain percentage of 13.81% and 11.5% respectively.

Table 2 exhibits the outcomes for the Full History State Representation, where the state of a job seeker contains their full work history. The Highest Expected Reward baseline model stands out with a significant mean income change (79.18%) and a large percentage of gainers (96.01%). As we will discuss later, this is caused by Transitions module biases. Deep Q-Learning and A2C also show promising results but fail to outperform the baselines.

5. Results and Discussion

In this chapter, we delve into the outcomes derived from our experimental setup featuring two unique versions of the transitions module - the Last Job State Representation and Full History State Representation. We start our discussion with findings from the two baseline methods outlined in Section 4.6. Subsequently, we elaborate on the results achieved through our efforts to learn an efficient policy.

5.1. Implications of the Findings

Last Job State Representation In the Last Job State Representation, our RL approaches learned policies that resulted in career paths with higher incomes than the observed career paths. In both the last job state representation and full history state representation, the learned policies improved the mean accumulated income by around 5%. While not a drastic increase, this change is significant over longer time scales, such as an individual's career. Notably, these improvements surpassed those of the baseline models. However, there was also a significant amount, approximately 12%, of agents for which the recommended paths performed worse than the observed.

Full History State Representation However, the Full History State Representation demonstrated a different pattern. While the DQN and A2C models also found policies improving counterfactual incomes, the baselines showed significantly larger improvements, particularly

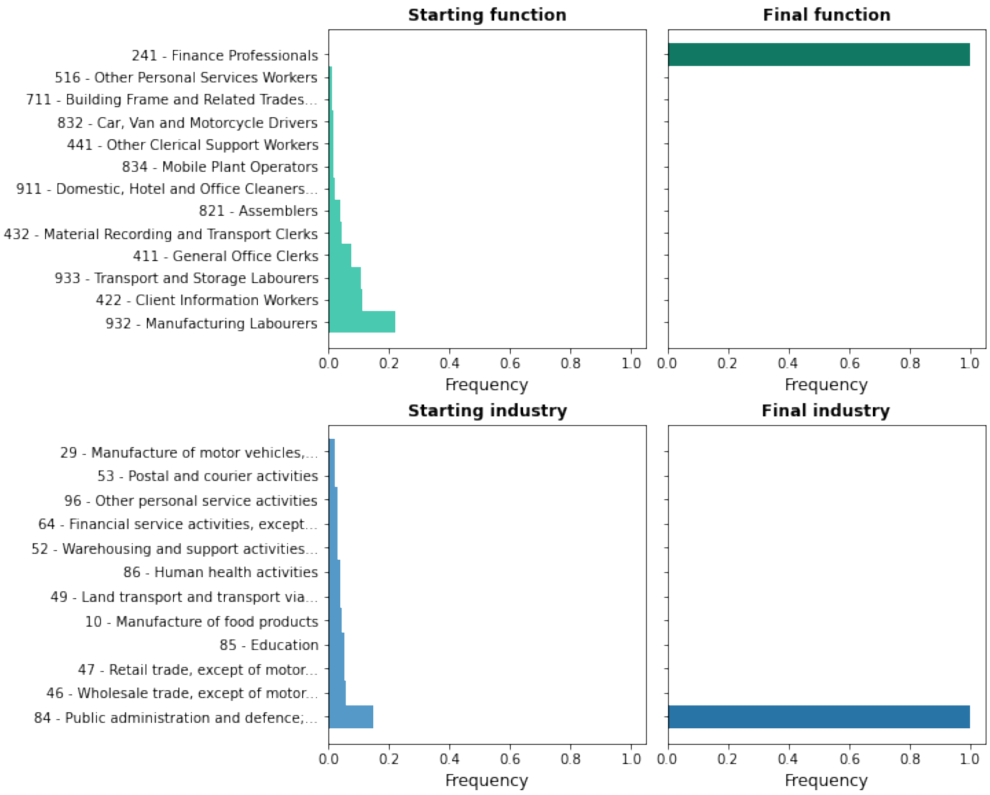

Figure 3: Full History State Representation - Greedy Highest Expected Reward Baseline: Starting and final distributions for the 12 most common functions and industries. The data were generated by running 1000 episodes of 40 time steps (10 years) each.

the *Highest Expected Reward* baseline. This raises questions about the validity of the environment. After careful investigation, we found out that this environment can be easily exploited by the Highest Expected Reward baseline due to the small-but-substantial transition probabilities predicted by the Transitions module. As we can see in Figure 3, regardless of the agent's starting state, it always applies and eventually succeeds to be hired for the highest-paying job of our dataset. By looking deeper into the classifier trained to predict the transition probabilities of the Full History State Representation, we see that regardless of the agents' prior experience there is always a small but significant probability of employment. After analysis of the training data, we believe that this is due to missing data in the prior experiences of employees hired in senior positions. Therefore, the training dataset is incomplete and depicts a world where someone can be hired, for example, as a Senior Finance professional with no experience in Finance.

Comparison Between Environments It's important to note that the results from the Last Job State Representation and Full History State Representation cannot be directly compared. Each model learns a policy to exploit the unique dynamics of the environment it is trained on, therefore, the ground truth differs for each environment. As such, results should be compared within the specific environment they originated from.

5.2. Limitations of the Study

Job Filtering The experiments relied on a narrowed field of 142 most common jobs. The decision to do so was driven by the challenges posed by the vast state space and unreliable transition probability prediction for less common jobs.

The Cost of an Action The research assumes no monetary cost for applying to a job, which is typically not the case in reality, where applications cost both time in interviewing or preparing, and perhaps other forms of preparation (e.g., studying). Considering the real-world costs in future studies could bring the environment formulation closer to reality.

Continuous Employment The assumption that individuals are always working doesn't account for potential breaks in employment. These breaks could result from various factors, including vacations, relocation, or further education, and should be considered in a more realistic formulation of the job market as an MDP.

Rivalrous market Another notable limitation is that the real-world job market is inherently rivalrous — a job offered to one applicant becomes unavailable to others. Recommending the most highly paying jobs to all users could potentially lead to an overwhelming influx of applications for those positions, resulting in many disappointed users due to the increased competition. Our approach focuses on income optimization, but it's crucial to recognize that a well-balanced approach considering job availability, individual preferences, and market dynamics is necessary to avoid an undue concentration of applications in specific roles. Future research should delve into strategies that account for these challenges while still aiming for income optimization to create a more comprehensive and realistic career planning model.

State Space and the Markov Property The models used in this research made different assumptions about state space and respected the Markov Property in different ways. The Last Job State Representation model simplified the state to be a job, assuming an employee's future depends solely on their last job. On the other hand, the Full History State Representation model considered employees' whole work experience as part of the state. The latter approach is closer to reality but can create states with too many dimensions, slowing policy learning. An option suggested in the literature for similar challenges is learning low-dimensional embeddings and reducing the state space size.

6. Conclusion

In conclusion, this research explored the use of artificial intelligence, specifically reinforcement learning, in the field of career planning. By harnessing data on employee work experience and job applications, the research aimed to recommend career paths that maximize long-term income for individuals.

The findings of this study showed promising results in both the Last Job State Representation and Full History State Representation approaches. The reinforcement learning models, particularly Q-Learning and Sarsa, were able to learn policies that improved the counterfactual incomes of individuals. In the Last Job State Representation, the mean accumulated income increased by around 5%, surpassing the performance of the baseline models.

However, it is important to acknowledge the limitations and failures of the Full History State Representation. The baseline models exhibited greater improvements in counterfactual incomes compared to the reinforcement learning models. This discrepancy was due to inaccuracies in the transition probability predictions, which allowed the Highest Expected Reward baseline to exploit the system.

These limitations indicate the need for further research to refine and improve the Full History State Representation. Future studies could explore alternative methods for estimating transition probabilities and address the issue of missing prior experience data. By addressing these challenges, future research can work towards creating more robust and accurate environments that better reflect the complexities of real-world career planning.

References

[1] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, 2 ed., 2018.

[2] M. L. Puterman, Markov decision processes, Handbooks in operations research and management science 2 (1990) 331–434.

[3] R. H. Topel, M. P. Ward, Job Mobility and the Careers of Young Men, The Quarterly Journal of Economics 107 (1992) 439–479. URL: https://academic.oup.com/qje/article/107/2/439/1838303. doi:10.2307/2118478.

[4] J. Long, J. Ferrie, "Labour Mobility" Oxford Encyclopedia of Economic History (2006).

[5] I. Paparrizos, B. B. Cambazoglu, A. Gionis, Machine learned job recommendation, RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems (2011) 325–328. doi:10.1145/2043932.2043994.

[6] J. Wang, Y. Zhang, C. Posse, A. Bhasin, Is it time for a career switch?, in: Proceedings of the 22nd international conference on World Wide Web - WWW '13, ACM Press, Rio de Janeiro, Brazil, 2013, pp. 1377–1388. URL: http://dl.acm.org/citation.cfm?doid=2488388.2488509. doi:10.1145/2488388.2488509.

[7] Y. Liu, L. Zhang, L. Nie, Y. Yan, D. S. Rosenblum, Fortune teller: Predicting your career path, 30th AAAI Conference on Artificial Intelligence, AAAI 2016 (2016) 201–207.

[8] L. Li, H. Jing, H. Tong, J. Yang, Q. He, B.-C. Chen, NEMO: Next Career Move Prediction with Contextual Embedding, in: Proceedings of the 26th International Conference on World Wide Web Com-

panion - WWW '17 Companion, ACM Press, New York, New York, USA, 2017, pp. 505–513. doi:10.1145/3041021.3054200.

[9] Q. Meng, H. Zhu, K. Xiao, L. Zhang, H. Xiong, A hierarchical career-path-aware neural network for job mobility prediction, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019) 14–24. doi:10.1145/3292500.3330969.

[10] H. Xu, Z. Yu, J. Yang, H. Xiong, H. Zhu, Dynamic Talent Flow Analysis with Deep Sequence Prediction Modeling, IEEE Transactions on Knowledge and Data Engineering 31 (2019) 1926–1939. URL: https://ieeexplore.ieee.org/document/8478343/. doi:10.1109/TKDE.2018.2873341.

[11] R. Liu, A. Tan, Towards interpretable automated machine learning for STEM career prediction, Journal of Educational Data Mining 12 (2020) 19–32. doi:10.5281/zenodo.4008073.

[12] H. Al-Dossari, F. A. Nughaymish, Z. Al-Qahtani, M. Alkahlifah, A. Alqahtani, CareerRec: A Machine Learning Approach to Career Path Choice for Information Technology Graduates, Engineering, Technology & Applied Science Research 10 (2020) 6589–6596. doi:10.48084/etasr.3821.

[13] Y. Lou, R. Ren, Y. Zhao, A Machine Learning Approach for Future Career Planning (2010) 1 – 4. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.375.3061&rep=rep1&type=pdf.

[14] R. J. Oentaryo, R. J. Oentaryo, X. Jayaraj, S. Ashok, E.-p. Lim, P. Kokoh, JobComposer : Career Path Optimization via Multicriteria Utility Learning (2018).

[15] B. Shahbazi, A. Akbarnezhad, D. Rey, A. Ahmadian Fard Fini, M. Loosemore, Optimization of Job Allocation in Construction Organizations to Maximize Workers' Career Development Opportunities, Journal of Construction Engineering and Management 145 (2019) 04019036. doi:10.1061/(asce)co.1943-7862.0001652.

[16] A. Gugnani, V. K. Reddy Kasireddy, K. Ponnalagu, Generating unified candidate skill graph for career path recommendation, IEEE International Conference on Data Mining Workshops, ICDMW 2018-Novem (2019) 328–333. doi:10.1109/ICDMW.2018.00054.

[17] N. Dawson, M. A. Williams, M. A. Rizoiu, Skill-driven recommendations for job transition pathways, PLoS ONE 16 (2021) 1–20. URL: http://dx.doi.org/10.1371/journal.pone.0254722. doi:10.1371/journal.pone.0254722.

[18] M. Kokkodis, P. G. Ipeirotis, Demand-aware career path recommendations: A reinforcement learning approach, Management Science 67 (2021) 4362–4383. doi:10.1287/mnsc.2020.3727.

[19] P. Guo, K. Xiao, Z. Ye, H. Zhu, W. Zhu, Intelligent career planning via stochastic subsampling reinforcement learning, Scientific Reports 2022 12:1 12 (2022) 1–16. URL: https://www.nature.com/articles/s41598-022-11872-8. doi:10.1038/s41598-022-11872-8.

[20] G. Rummery, M. Niranjan, U. of Cambridge. Engineering Department, On-line Q-learning Using Connectionist Systems, CUED/F-INFENG/TR, University of Cambridge, Department of Engineering, 1994. URL: https://books.google.de/books?id=JdyRPgAACAAJ.

[21] C. J. C. H. Watkins, P. Dayan, Q-learning, Machine Learning 1992 8:3 8 (1992) 279–292. URL: https://link.springer.com/article/10.1007/BF00992698. doi:10.1007/BF00992698.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning (2013). URL: https://arxiv.org/abs/1312.5602v1. doi:10.48550/arxiv.1312.5602.

[23] V. Mnih, A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, K. Kavukcuoglu, Asynchronous Methods for Deep Reinforcement Learning, 33rd International Conference on Machine Learning, ICML 2016 4 (2016) 2850–2869. URL: https://arxiv.org/abs/1602.01783v2. doi:10.48550/arxiv.1602.01783.