# TALENT RECRUITMENT ANALYTICS IN THE ERA OF BIG DATA

by

QINGXIN MENG

A dissertation submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

written under the direction of

Dr. Hui Xiong

and approved by

_____

_____

_____

_____

Newark, New Jersey

May 2020

ABSTRACT OF THE DISSERTATION

Talent Recruitment Analytics in the Era of Big Data

By QINGXIN MENG

Dissertation Director: Dr. Hui Xiong

This dissertation aims at developing effective and efficient data mining techniques to solve varied talent recruitment issues, reforming the overall process with respect to talent sourcing, screening, matching, and assessment. Intelligent talent recruitment has gained increasing attention due to the critical talent competitions and intensive talent mobilities over the years. Previous studies mainly focus on discovering conceptual and theoretical topics, while applications for supporting organizational decision making are still under-explored.

To this end, we propose several approaches purposed to not only help the people to make intelligent talent-related decisions, but also obtain domain understandings through a multifaceted data-driven perspective. In particular, we first present a hierarchical career-path-aware neural network to study individuals' job mobilities. In this work, two problems are predicted all together on the basis of one's historical career paths: 1) who will the individual's next employer? 2) How long will the individual stay with his/her next employer? Several job mobility patterns regarding working duration, firm types, and *etc.* are discovered simultaneously. Also, we propose an intelligent matrix factorization based framework to address job salary benchmarking tasks. In this work, we consider multiple contextual factors to improve the prediction accuracy, such as job responsibility, company features, work location, and the time the

job wanted. Furthermore, we put forward a Non-parametric Dirichlet Process-based graphical model to address the "cold-start" problem for salary benchmarking, which also has superior interpretability associated with job responsibility and company.

ACKNOWLEDGEMENTS

This work is completed with many years of efforts, and can not be finished without many people's support and assistance. Being grateful, I would first like to thank my mentor and advisor, Professor Hui Xiong. Without any reservation, he has provided visionary advice for my research direction, given his best support for my Ph.D. study, intellectually and mentally. His personality, values of life, the persistent pursuit of breakthroughs in career, the genuine caring and support to his family, students and friends deeply and consistently have an influence on me, motivating me to be a better myself.

I also wish to express my deepest gratitude to my co-advisor, Dr. Hengshu Zhu. He spent much time and effort assisting me in completing the researches. His intelligence, optimism, and friendship encouraged me to overcome the difficulties during the research.

I would like to show my special thanks and love to my husband Chengmi He. Without his support and love, I can not make this work possible. I also need to thank my parents, Xianting Meng and Jinlan Li, who raised me to be a positive person, fulfilling with confidence and courage to accomplish challenges.

I am indebted to Professor Xiao Fang, Professor Xiaodong Lin, and Professor Periklis Papakonstantinou, for taking their time to serve as my dissertation committee

and providing invaluable advice on my thesis. I would like to recognize the valuable assistance from Professor Keli Xiao, Dazhong Sheng, and Le Zhang for my research. It is a pleasure to work with them. I would also like to extend my gratitude to Ken Chen, Professor Junming Liu, Dr. Yanchi Liu, Professor Chuanren Liu, Professor Xiaolin Li, Professor Hongting Niu, Professor Meng Qu, Farid Razzak, Mingfei Teng, Professor Jiangyuan Yang, Dr. Hao Zhong, for their help, friendship, and valuable suggestions. It is frustrating that I am not able to name them all.

Last but not least, I would like to acknowledge the Department of Management Science & Information Systems (MSIS), Rutger - the State University of New Jersey, and Baidu Inc. for providing me a comfortable working environment and facilitating devices, making the experiments successfully conducted.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Research Background and Motivation

Talent Recruitment (TR) has become a challenging issue in today's human resource management due to three main reasons (Farndale, Scullion, & Sparrow, 2010). 1) There has been increasing global competition for talents, especially highly-skilled talents (Grant, 2008). 2) The job mobilities happen more frequently during one's career life than what had happened decades before. The factors that influence individuals in the job search decision-making process are more complicated and multifaceted (Sullivan & Al Ariss, 2019), such as work-life balance, high-growth environment, and personal visions. 3) In order to cope with the dynamic and competitive business environment, the organizations need to train their employees a longer time, and may bear a big loss if they decide to leave (Lawler, 2017).

Although there exists extensive research underlying the important issues behind TR in a varied range, most of them focus on theoretical and conceptual development (Collings, Wood, & Szamosi, 2018; Muriithi & Makau, 2017; Alic et al., 2016; Stone & Rosopa, 2017); thus organizations cannot make use of them directly. In this dissertation, we aim to develop data-driven methods to address TR related issues, which can benefit the human resource department in the decision-making process, such as talent

selecting, assessment and *etc.* Moreover, our solutions can provide certain domain interpretability simultaneously.

## 1.2 Research Summary

Our research involves two TR tasks–*job mobility prediction* and *Job Salary Benchmarking* (JSB)– with three different methodologies. We first outline the major results of the research and then summarize the overall contributions.

One major effort of my research is to predict job mobility at the individual level. We propose a hierarchical career-path-aware neural network method to understand the talents' job mobilities by learning their historical job-hopping behaviors. In this work, two main issues regarding job mobility are addressed: 1) who will be the individual's next employer; 2) how long will the individual stay with his or her next employer. Three different levels of information are considered, including personal-related information, company-related information, and position-related information. As we know, during one's career path, one may experience several internal transfers within one company, as well as several external transfers among companies. A neural network model is designed to capture and understand those internal transfers and external transfers hierarchically; moreover, a delicate attention mechanism is implemented to obtain model interpretability. This model can effectively identify both environmental and individual historical patterns that may influence the decision-making process of talents.

Another major effort of my research involves JSB. JSB is a process by which organizations acquire and analyze labor market data to determine appropriate compensa-

tion for their actual and prospective employees. The traditional salary benchmarking methods are largely based on limited survey data and statistical methods, and they suffer from un-inferable problems when the data are deficient. However, the lack of data will be the most common situation in the real world. Another problem is that most of the previous studies are based on job category, which is too general to meet the particular requirement of the Compensation and Benefit (C&B) department. To this end, we propose a fine-grained, automatic JSB system for organizations, where we construct an expanded salary matrix, and then transform the problem into a Matrix Factorization (MF) task. Four domain-related assumptions associated with job responsibility, company, work location and time, are first tested then integrated into the framework to improve the estimation efficiency. Based on the four observations, we design four corresponding regularizers to optimize the learning process of the basic MF model.

While the MF-based intelligent salary benchmarking model can effectively estimate the job salary with confounding factors, two main issues remain unsolved. It suffers from the "cold start" issue for a new company or job position, and limited model interpretability. Along this line, we design a Nonparametric Dirichlet-Process-based Latent Factor Model for JSB, namely NDP-JSB, which can jointly model the latent representations of both company and job position, and then predict job salaries for each company and job position combination. Moreover, as a probabilistic graphical model, it can address the "cold-start" issue well, as well as provide deeper interpretation on the estimations, such as the components of skill sets the job emphasizes on, and the similar companies the model refers to.

## 1.3 Contributions

The contributions of our research can be summarized in four aspects. First of all, the applications provide a unique data-driven perspective for the organizations in the process of talent management. In particular, for talent job mobility, the understanding of it can benefit talent management operations in a number of ways, such as talent recruitment, development, and retention. Knowing the potential career paths of an employee would help executives and department managers in internal promotion decisions to motivate key talents and reduce turnover rates. For example, when hiring people, recruiters want to know who has the greatest opportunity to accept the offer, and if there is a high chance of successful hiring, who will stay long. One of the most interesting problems for job seekers is what is the best route to join their dream company and positions. Some job seekers may also want to know, based on their previous working experiences, what's their next possible move. Also, as a fundamental tool for attracting, retaining, and motivating employees, salary benchmarking plays an important role in support of the success of a company's human resource management (e.g., maximizing the productivity of the company; minimizing the cost of human capital in a long-run view). Our framework provides an effective and efficient solution for studying the overall market data and organizations' special factors together, and then offer useful job salary advice. Moreover, the solutions have been demonstrated effectiveness with extensive experiments on massive real-world datasets.

Secondly, analytical findings related to TR are deserved to be referred to for future relevant research. We have discovered several interesting patterns related to

the individual's job transitions based on data-driven methods. For instance, the longer an individual stays with an employer, the higher attention (importance) the employer has; a job appearing in a later position in one's career path has higher attention and *etc.* Also, in the MF-based JSB solution, four domain-related assumptions are first tested then integrated into the framework to improve the estimation efficiency: 1) If jobs have similar job responsibilities, their salaries should be closer. 2) If jobs are opened by two similar companies, their salaries should be closer. 3) If the jobs are released within a near period, their salaries should be closer. 4) If the work locations for the two jobs have similar economic conditions, their salaries should be closer. In the NDP-JSB solution, the model learned five classes for jobs and their corresponding key skillsets, namely, promotion, front-end, back-end, testing, and administration. Also, it can identify similar companies from the job market.

Thirdly, the contributions have also stemmed from the degrees of difficulty and the novelty of the problems. For difficulty, data mining has become popular a decade ago before it enters the TR area. One of the reasons is the high uncertainty nature of TR problems. For example, given a picture, what's in the picture is determined, and the uncertainty is low. On the contrary, given the previous working experience, who will be the individual's next employer is hard to predict, as the uncertainty is high. The nature of high uncertainty makes the TR problems hard to address. For novelty, to the extent of my knowledge, among the existing work on individual-level job mobility prediction, we are the first to conduct dual highly specific tasks of predicting people's next employer and the eventual duration. Also, we are the first to transform the JSB into a matrix completion task, in order to address the sparsity

problem in existing salary observations. Our research can enlighten other scholars who want to apply data mining techniques to managerial issues: an effective framework must be developed based on the well understanding of domain knowledge and a successful problem abstraction, especially for those problems with high uncertainty.

Last but not least, our methodology is easy to be generalized into other application areas. For example, the NDP-JSB model adopted an NDP structure to group companies with categorical features, and a Latent Dirichlet Allocation structure to learn the latent factors for job descriptions in the form of texts. In this way, NPD-JSB can take full advantage of correlations among companies and job descriptions in order to assign a suitable salary to a (job, company) combination. There exist many problems with a similar abstract problem structure. For example, to predict the buying behaviors of users, the information gathered from users is usually categorical, such as age and gender, while the items are described by texts. The NPD-JSB framework is an appropriate solution for user-item consumption predictions.

## 1.4 Overview

Chapter 2 introduces a hierarchical career-path-aware neural network for job mobility prediction. We will give a detailed explanation of the design art of the hierarchical sequential structure, and how we formulate the problem of predicting an individual's next employer and stay duration. We will describe how we process and select the features in the model, as well as the details of experimental steps and results.

Chapter 3 presents an MF-based framework to estimate job salaries. Given a combination of job and company, and multiple contextual information, such as job

description, company features, work location, and job opening time, we will describe how we can give lower and upper bound of the salary. The correlations among those factors are first examined and then integrated into the framework to improve the performance.

Chapter 4 proposes an NDP-JSB model for the salary benchmarking problem. We will clarify the mechanisms of the NDP and LDA structures, which are designed to learn latent factors for company and positions, respectively. As a probabilistic graphical model, we describe how it can solve the "cold-start" problems with the corresponding experimental results. Furthermore, we will present several interpretations regarding profiling a job salary.

CHAPTER 2

# A HIERARCHICAL CAREER-PATH-AWARE NEURAL NETWORK FOR JOB MOBILITY PREDICTION

## 2.1 Introduction

In this chapter, we focus on providing data-driven solutions to the job mobility prediction problem. The importance of job mobility has been widely documented as a key element of human behaviors by researchers from different areas. For instance, (Topel & Ward, 1992) claimed that work experience accumulation is mainly attributable to job changing activities for locating good job matches, especially for young employees. It has also been found that people have renewed interests in job movements by which job mobility occurs and results in different career paths (Rosenfeld, 1992). In addition, evidence has been provided to support the significant relationship between individual's decision of migration and job mobility (Bartel, 1979), the connection between social ties and job changes (Wegener, 1991), the wage effect of cumulative job mobility (Keith & McWilliams, 1995), and the like.

From the perspective of human resource manager, it is important to understand the job mobility in the organization level as well as the individual level. The main purpose of related studies is to support the decision-making process regarding talent management. Understanding the potential career paths of an employee would help

executives and department managers in internal promotion decisions to motivate key talents and reduce the turnover rate. Also, during the recruiting process, employers may be interested in knowing the probability for candidates to accept the job offers. Meanwhile, if there is a high chance of hiring, how long will they stay? On the other hand, from an employee's viewpoint, people also concern about their career development and growth for achieving professional success, and a question that may keep bothering them is: what is the best and fastest career path leading to the success in professional life?

However, job mobility prediction is not an easy task. Traditional studies of job mobility were largely based on limited survey data and focused on the empirical analysis of key factors affecting people's career paths (Miller, 2011; Vance, 2005). The rapid development of information technology and the emergence of professional social networks enable us to collect and analyze large-scale career path data from the real word. For example, as one of the earliest works in the individual-level job mobility prediction topics, (H. Xu, Yu, Xiong, Guo, & Zhu, 2015) developed a framework to predict whether there is a large chance of job change in the next six months for individuals. (Liu, Zhang, Nie, Yan, & Rosenblum, 2016) proposed a multi-view multi-task learning approach to predict the promotion in one's career path. These works considered work experience and daily activity data in their models, but the target problems were somehow general and had limited practical applications. Thus, in this chapter, we propose to address the problem of job mobility prediction by answering two specific questions: (1) "Who is your next employer?" and (2)"How long will you work for your next employer?" The first question is regarding the position prediction,

and the second one tells the eventual duration of your new job.

The main challenges of the proposed prediction tasks are twofold. (1) We need to handle the dynamic hierarchical nature of career paths for employees, such as internal job mobilities and external job mobilities. For example, one person may experience several internal job transfers within a company before he/she hops to a new company. Both the internal transfers and external job hoppings may influence the direction of the future of the career path at different levels. Moreover, the data are complex with heterogeneous forms, including the personal-specific, company-specific, and position-specific data. For example, the personal self-introductions and company descriptions are freely structured, some features are categorized, while others are numerical. (2) The other challenge is to jointly consider the influence between environmental factors and individual historical patterns. The closeness between companies is one of the environmental factors. For example, one person working in a bank may have a high chance to hop to another bank. Meanwhile, tracing back the whole history of one's career path, which company or position takes the main role in the decision-making process, is another important factor we need to figure out.

We provide our solutions to the aforementioned issues and contribute to the literature in four ways as follows:

- To the best of our knowledge, among the existing work on individual-level job mobility prediction, we are the first to conduct dual highly specific tasks to predict people's next employer and the eventual duration.

- We propose a hierarchical career-path-aware neural network approach to inte-

grate three levels of information, including personal-specific, company-specific and position-specific knowledge. The model embeds survival analysis and attention mechanism, which lead to a certain level of interpretability of results.

- For both proposed forecasting tasks, we demonstrate evidence showing the superiority of our model in comparison to several well-known benchmarks.

- Our model offers a new way to show data-driven evidence in support of the connection between specific factors and job mobility. As case studies, new evidence has been presented to show the impacts of various factors (*e.g.*, job duration, firm type, *etc.*) on the job mobility prediction performance.

### 2.1.1 Data Description

The data were collected from a famous online professional social platform, where users can build professional profiles introducing their education and work experience, like a public online curriculum vitae. We summarize the collected features into three categories, including personal-specific, company-specific, and position-specific data. Personal-specific information is static and includes freely structured self-description texts and the number of social connections. Company-specific features (*e.g.*, company name, type, size, etc.) and position-specific features (*e.g.*, position type, service duration, etc.) were collected as sequential data to describe the work timeline of the users. Our data contain both internal and external job transitions in their professional life. Figure 2.1 shows an example of the hierarchical structure of our data.

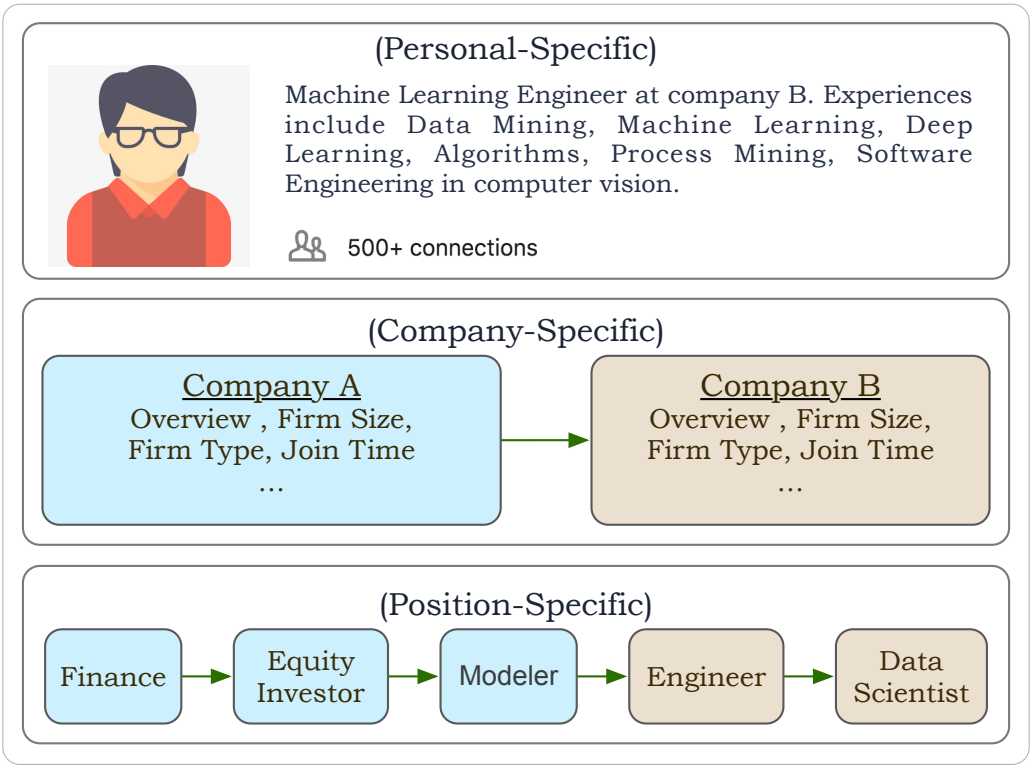



Figure 2.1. An example of the three-layer structure of our job mobility data.

## 2.2 Preliminary Analysis

How to design an effective framework to model such a hierarchical structure becomes a key challenge in this job mobility prediction task. To handle the complex data structure, we formatted those three kinds of features into different levels. The static personal information were transformed into a vector as one level, while company-specific and position-specific features were transformed into a sequence of vectors respectively as the second and the third levels. Each level in the structure contains useful information that we do not want to mass them up in a simple machine learning model. Thus, we propose to construct a neural network model to handle the three level of inputs hierarchically. We will provide detailed discussion in section 2.3.

The second challenge and the motivation of our model are problem-specific. We believe the job mobility prediction is a sequential problem and even a long-distance dependent sequential problem. The decisions in people's career paths rely on two groups of factors. The first one refers to the work environmental factors, which describe the natural connections among company-specific characteristics, such as firm types. For instance, employees in a bank are highly prone to hop to another bank, rather than other manufactories. Although we do not have direct features to represent the similarities between companies, such information will be obtained by learning from people's job-hopping patterns. The second group of factors we should consider are the individual historical factors. In one's historical career path, she/he might have served several employers and been occupied in different positions. An effective model should be able to understand which experiences during the career path play the most important roles for future decisions. Such information should be captured during the model training process.

In particular, here we introduce a motivating example of the prediction problem in Figure 2.2, which is a real case in our sample. Specifically, the person has worked for three employers, namely "Fannie Mae", "CGI", and "BearingPoint", one after another before he hopped to "Freddi Mac". If we use the Markov Chain Model, which only considers the environmental factors to predict the next employer, the result is "Accenture". The model considers the last employer "BearingPoint" as an important reference in the prediction process, given that "Accenture" and "BearingPoint" are both consulting companies. The result is reasonable as it only considers the environmental factors. On the other hand, our model intelligently gave higher attention

for his first employer "Fanni Mae" than "CGI" and "BearingPoint", which might be due to their associated duration. Therefore, it can successfully predict the right next employer "Freddi Mac", which is closely tied up with the person's first employer "Fannie Mae". Indeed, "Fannie Mae" and "Freddi Mac" are two large house mortgage companies. To get the correct predictions, we need to jointly consider the environmental factors as well as the individual historical factors, which can be discovered from the detailed information of people's career paths, such as the job duration of each position.

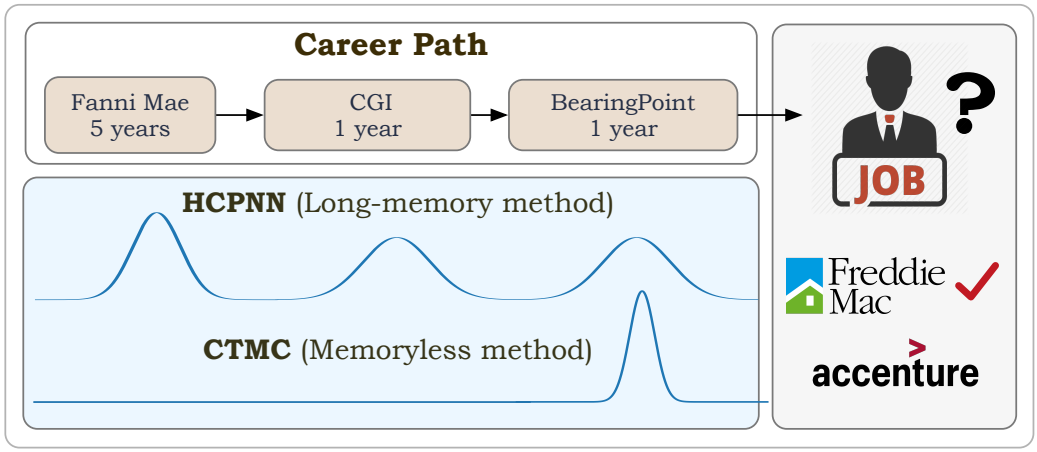
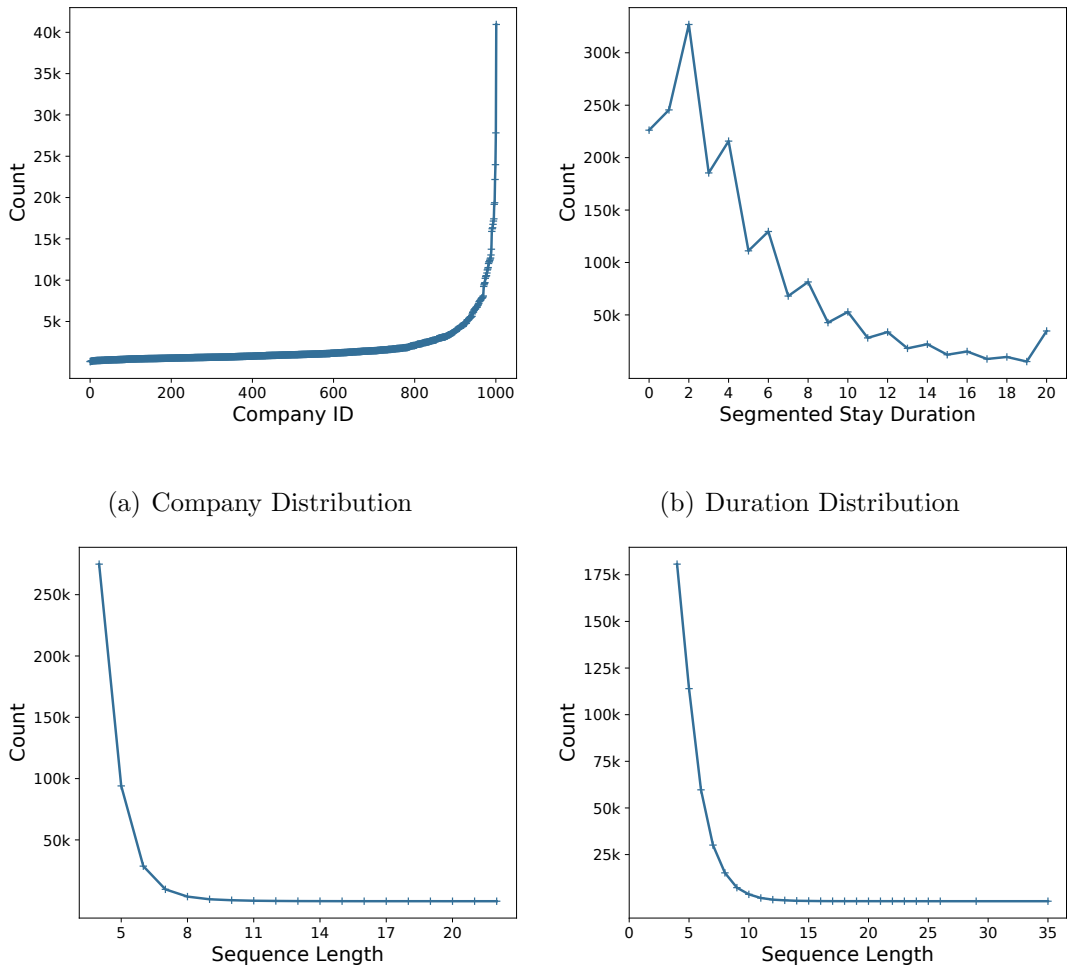



Figure 2.2. The prediction of a real case.

### 2.2.1 Data Analysis

We also analyzed the characteristics regarding the distribution of our samples. Figure 2.3 (a) demonstrates the distribution of the occurrence number of the firms in our real-world dataset. As can be seen, 20% of the firms cover about 60% samples in the data. Figure 2.3 (b) shows the distribution of the job duration, which was split into 21 windows as 0.5 years, 1 year, 1.5 years, ..., 10 years, and more than 10 years. We

can see that most people stay in one position for 1-3 years, and there is a decreasing pattern for the longer duration. Interestingly, there is also a "sawtooth" pattern in the job duration distribution, which may indicate that people try to avoid leaving a job in the odd number of half years. Also, both the length of company and position sequence have long tail shape too, as illustrated in Figure 2.3 (c) - (d). Our model needs to handle the imbalanced distribution for better predictions.




(a) Company Distribution

(b) Duration Distribution

(c) Company Transition

(d) Position Transition

Figure 2.3. The data distribution of different aspects.

## 2.3 Problem and Methodology

In this section, we formulate the job mobility prediction problem based on the data availability and then discuss the new method we proposed for addressing the problem.

### 2.3.1 Problem Statement

Let $u \in U$ denote a person, $c \in C$ denote a company, $p \in P$ denote a position, where $U$, $C$ and $P$ represent the full set of people, companies, and positions respectively. Given the company sequence $\overrightarrow{Q(u)}$, the position sequence $\overrightarrow{B(u)}$, and the personal-specific information $\Omega(u)$, we represent $u$'s three-layer career path as $S(u) = \left\{\overrightarrow{Q(u)}, \overrightarrow{B(u)}, \Omega(u)\right\}$. The company sequence $\overrightarrow{Q(u)}$ can be written as:

$$\overrightarrow{Q(u)} = \{(c_1, c_2, ..., c_g)|u\}, \tag{2.1}$$

and position sequence $\overrightarrow{B(u)}$ can be written as:

$$\overrightarrow{B(u)} = \{(p_{11}, p_{12}, ...), ..., (p_{g1}, ..., p_{gh})|u\}, \tag{2.2}$$

where $p_{gh}$ describes the $h$-th position for his $g$-th employer $c_g$. For example, $p_{24}$ represents the fourth position in the second company $c_2$ where a person worked.

Then, we formulate our problem as follows:

**Problem 1** *Given a person's three-layer career path,*

$$S(u) = \left\{\overrightarrow{Q(u)}, \overrightarrow{B(u)}, \Omega(u)\right\}, \tag{2.3}$$

*where* $\overrightarrow{Q(u)}$ *stands for company sequence,* $\overrightarrow{B(u)}$ *stands for position sequence, and* $\Omega(u)$ *stands for personal information, we want to predict person* $u$*'s next employer* $c_{g+1}$ *and the duration* $d_{g+1}$ *at* $c_{g+1}$.

### 2.3.2 An Overview of the Model

Now we introduce the methodology we proposed for addressing the job mobility prediction problem. The design of our model is rooted in the hierarchical data structure, and we name it as the *hierarchical career-path-aware neural network* (HCPNN). The model includes three main components, namely *Internal Job Mobility Representation*, *External Job Mobility Representation*, and *Prediction*.

Figure 2.4 illustrates the framework of our HCPNN. Specifically, for the component of *Internal Job Mobility Representation*, we embed the sequential position features as the inputs to a long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) layer. Then, we apply a local attention mechanism for obtaining the internal job mobility representation. For the component of *External Job Mobility Representation*, we first concatenate sequential company feature embeddings with the internal job mobility representation, then we feed them into another LSTM layer for training the external job mobility representation. Meanwhile, we conduct the embedding process for the personal-specific features, and then we apply the global attention mechanism to both external job mobility representation and static personal representation. Following that, we form a hierarchical job mobility neural network, which has the ability to learn the influences of internal and external job mobility on their next job decisions. Finally, for different learning tasks, the output from the HCPNN will be fed into different *prediction* widgets. As emphasized in the problem statement, we aim at predicting the next employer as well as the job duration with the next employer for every person.

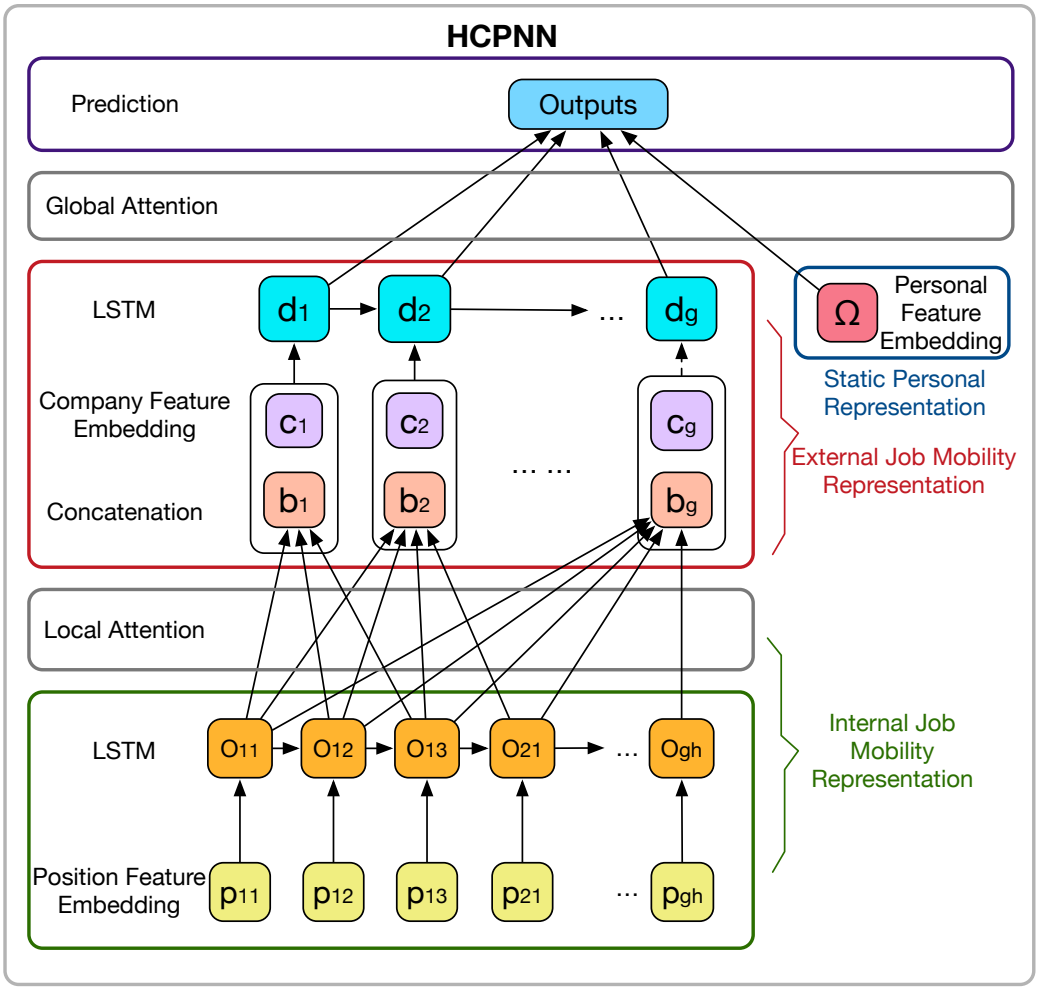



Figure 2.4. The graphical representation of the HCPNN model.

### 2.3.3 Technical Details

Here, we introduce the details of *Internal Job Mobility Representation*, *External Job Mobility Representation*, and *Prediction* components mentioned above.

**Internal Job Mobility Representation**

The inputs of internal job mobility representation layer are position-specific features $\overrightarrow{B(u)}$. After the embedding, we feed them into an LSTM layer to learn the hidden representation of the position-specific sequential features. We choose LSTM to handle this task due to its predictive power, as well as the ability to alleviate the gradient vanishing problem in long-distance dependent sequential problems. In our framework, we refer the output of LSTM of this layer as $o_{11}, o_{12}, ..., o_{gh}$, and then we apply a local-attention mechanism with these outputs to get the final representation for internal job mobility. In particular, we propose to add an attention mechanism for obtaining the model interpretability, on which we rely for result analysis. Also, we use this mechanism to align the internal job mobility representation $b_1, b_2, ..., b_g$ and company sequence embeddings $c_1, c_2, ..., c_g$ with the same length.

The attention mechanism tries to capture the degree of attention for representing the importance of inputs in the learning process. Based on our proposed tasks and the data structure, we apply the attention technique as follows. For each attention output $b_i$, we assign attention based on the company sequence inputs before and include $c_i$. For instance, suppose position sequence $p_{11}, p_{12}, p_{13}$ is associated with company $c_1$, and $p_{21}$ is associated with $c_2$, then we assign the attention value on $o_{11}, o_{12}, o_{13}$ to get the attention output $b_1$, and assign attention value on $o_{11}, o_{12}, o_{13}, o_{21}$ to obtain the

output $b_2$. In this way, we prevent to use future information to predict the future. Also, the internal job mobility layer is aligned with the external job mobility layer. The local attention mechanism can be formulated as follows.

$$
\begin{aligned}
v_{ij} &= tanh(W_a o_{ij} + b_a), \\
\alpha_{ij} &= \frac{\exp(v_{ij}^T u_a)}{\sum_{i=1}^{g}\sum_{j=1}^{h}\exp(v_{ij}^T u_a)}, \\
b_g &= \sum_{i=1}^{g}\sum_{j=1}^{h}\alpha_{ij}(W_a o_{ij}),
\end{aligned}
\tag{2.4}
$$

where $W_a$, $b_a$ and $u_a$ are training parameters, $o_{ij}$ means the $i$-th company $j$-th position's hidden states learned from the first LSTM layer, and $b_g$ is the output vector for $g$-th internal job mobility representation.

**External Job Mobility Representation**

Similar to the internal job mobility representation, we utilize an LSTM layer and attention mechanism to model the external job mobility information. First, we concatenate the aligned sequential company embedding data $c_1, c_2, ..., c_g$ with the internal mobility representation $b_1, b_2, ..., b_g$, then we feed them into another LSTM layer and obtain the output $d_1, d_2, ..., d_g$. Personal features $\Omega$ is further embedded. Then a global attention is computed based on both $d_1, d_2, ..., d_g$ and $\Omega$ for getting a final output. The attention technique implemented here not only integrates the personal information into our framework, but also improves the result interpretability of our model.

**The Prediction Module**

Our job prediction problem contains two major tasks: the next company and the duration at the next company. For the first one, we formulate it as a classification task as below. We first feed the output vector learned from the HCPNN model into a fully-connected layer where the output dimension matches our total company numbers. Then, we use a *softmax* activation function to normalize the probabilities $P(c)$ of each possible company. The process is demonstrated in Figure 2.5.

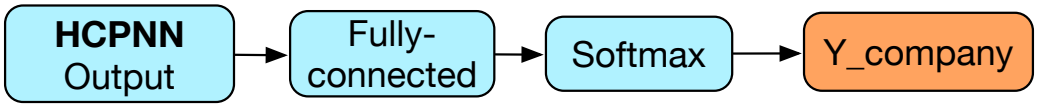



Figure 2.5. The process of predicting next employer.

Based on the maximum likelihood estimation, we optimize the loss function for predicting the next employer, which is formulated as follows. Given a person $u$, we maximize

$$L^u_{company} = \sum_{i=2}^{g} \log \left( P(c = c_i) | S(u) \right). \tag{2.5}$$

In the optimization process, we can not predict the first employer $c_1$, as $\overrightarrow{B(u)}$ and $\overrightarrow{Q(u)}$ are empty before $c_1$. So we summarize the log-likelihood of company sequence, start from the second index.

For the second task of predicting job duration, we integrate survival analysis into our framework. We regard the event a person joins a company as her start life in this company. And the event of leaving the company as a death event.

Survival analysis has been widely used for estimating the occurrence time of an

event with censored observations. We denote the probability of an event does not happen before time $t$ as $P\left(T_{survival} \geq t\right)$, and the instantaneous rate of the occurrence of the target event at time $t$ as $\lambda(t)$, so we have

$$\frac{d}{dt}P\left(T_{survival} \geq t\right) = -\lambda(t)P\left(T_{survival} \geq t\right). \tag{2.6}$$

To solve the Equation 2.6, we can get

$$P\left(T_{survival} \geq t\right) = exp\left(-\int_0^t \lambda(\tau)\, d\tau\right). \tag{2.7}$$

Sometimes, we can only observe the survival event within a period time $t$, after time $t$ we cannot continue the observations. This is called right-censored data. And the probability can be computed by Equation 2.7. And if a target event occurred at the exact time $t'$, the probability is computed as

$$P\left(T_{survival} = t'\right) = \lambda(t')exp\left(-\int_0^{t'} \lambda(t)\, dt\right), \tag{2.8}$$

where the meaning of the function can be explained as the joint probability of the event happening at the exact time $t'$ and the event does not happen before time $t'$.

In our problem, we first use a fully-connected layer to transform the output learned from the HCPNN into $k+1$ dimension, where the first $k$ dimensions can represent the individual turnover probability for the segmented time period $\left[(0, \frac{1}{k}T), [\frac{1}{k}T, \frac{2}{k}T), ..., [\frac{k-1}{k}T, T)\right]$. The last dimension denotes the turnover probability after $T$. Note that $T$ is the longest observation time in our problem. The larger of $k$, the higher precision of the simulation. In this way, we can transform the task of predicting the next duration to a survival analysis problem. Figure 2.6 illustrates the process of job duration prediction.

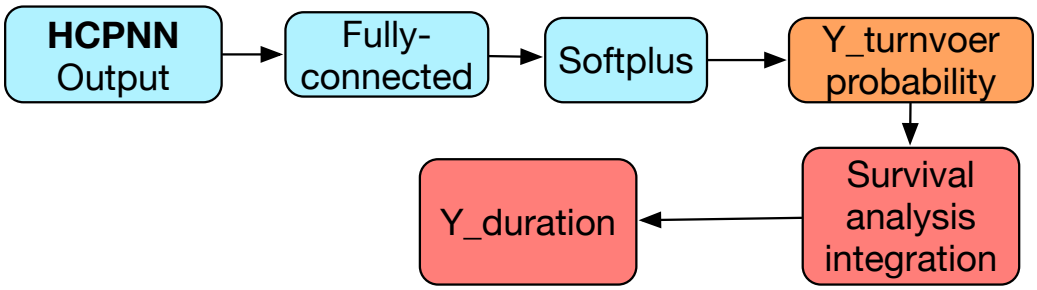



Figure 2.6. The process of predicting job duration.

The technical details can be summarized as follows. Let $\lambda(\tau)|S(u)$ denote the individual turnover probability for the next employer under the condition of previous career path $S(u)$, where $\tau \in (0, +\infty)$. We will use $\lambda(\tau)$ for short in the following. We have $\lambda(\tau) > 0$ by definition, so we use the *softplus* function to constrain the value to be positive. So the log-likelihood of predicting the duration at the next company can be computed as:

$$\begin{aligned} L^u_{duration} &= \log\left(\prod_{i=2}^{g} P\big(d = d_i|S(u)\big)\right) \\ &= \sum_{i=2}^{g} log\left(\lambda(d_i)\exp\left(-\int_0^{d_i}\lambda(\tau)\,d\tau\right)\right) \\ &= \sum_{i=2}^{g} log\left(\lambda(d_i)\right) - \sum_{i=2}^{g}\int_0^{d_i}\lambda(\tau)\,d\tau. \end{aligned} \tag{2.9}$$

The reason why the summation starts from index 2 is the same with that we explained for computing $L^u_{company}$. If we split the observation time into two parts, $(0, T)$ and $(T, +\infty)$, the job-hopping events occurred after time $T$ will be treated as right-censored data points. The Equation 2.9 can be rewritten as:

$$\begin{aligned} L^u_{duration} = \sum_{i=2,d_i<T}^{g} log\left(\lambda(d_i)\right) &- \sum_{i=2,d_i<T}^{g}\int_0^{d_i}\lambda(\tau)\,d\tau \\ &- \sum_{i=2,d_i\geq T}^{g}\int_0^{T}\lambda(\tau)\,d\tau. \end{aligned} \tag{2.10}$$

By summarizing the loss function for predicting the next employer and job duration, we get our final loss function as below:

$$Loss = -\sum_{u \in U} \left( \alpha L^u_{company} + (1-\alpha) L^u_{duration} \right), \tag{2.11}$$

where the $\alpha$ is the tuning parameter of these two types of loss functions. Given the probabilities of $P(c_{g+1}|S(u))$ and $\lambda(\tau)|S(u)$, it is easy to deduce the most possible next hopping company to be $argmax\{P(c_{g+1}|S(u)\}$. As for the next job duration, we need to calculate the integration of $P(d = \tau|S(u))$ over time $\tau \in (0, +\infty)$, the formulation will be

$$\begin{aligned} duration &= \int_0^{+\infty} \tau \cdot P(d = \tau|S(u))\, d\tau \\ &= \int_0^{+\infty} \tau \cdot \lambda(\tau) \exp\left( -\int_0^{\tau} \lambda(s)\, ds \right) d\tau. \end{aligned} \tag{2.12}$$

Since the Equation 2.12 is non-linear, and there is no analytic solution, we can use simulation to solve the integration problem. We segment the time window $t \in (0, +\infty)$ into $z$ intervals, then use the function above to calculate the integration.

## 2.4 Experiments

In this section, we introduce the details of experiments conducted on a real-world dataset for validating our HCPNN.

### 2.4.1 Experimental Setup

The data were collected from a well-known online professional social platform. We filtered out the samples with the number of the external job transitions less than four. And we selected those companies having the highest occurrence frequency as our research targets. The major statistics of the data are summarized in Table 2.1.

Table 2.1. The statistics of experimental data.

| **Name** | **value** |
|---|---|
| Number of samples | 414,266 |
| Number of companies | 1,002 |
| Number of normalized position types | 26 |
| Max/min/mean company sequence length | 22/4/4.52 |
| Max/min/mean position sequence length | 35/4/5.14 |
| Observed time periods | 1988.1-2018.11 |

As described before, our data have three levels, personal-specific features, company-specific features, and position-specific features. To handle the rich forms of data (free text, numerical and categorical features), we preprocessed the data with the following methods. For the free text feature, such as company description and personal self-introduction, we used the *wold2vec* (Řehůřek & Sojka, 2010) embedding method to transform a word into a vector. Then we computed the mean value of the embedding for every dimension respectively, in this way we got a fixed length of the vector for the free text of varying length. For the categorical features, the number of types of which less than ten, we used one hot encoding; for those with the number of types more than ten, such as company ID, we used a fully-connected layer for the embedding process. To process the job duration at companies and positions, we first segmented the time less than ten years by every half year into 20 small windows, and the time larger than 10 years was set into one category. In this way, the job duration was segmented

into 21 categories, namely 0.5 years, 1 year, 1.5 years,..., 10 years, and more than 10 years. We segmented job duration in this way because it is hard to predict the exact leaving time when an employee stays service for more than 10 years. We counted and normalized the personnel flow in/out/transfer number of every company for every three years. Thus, given a specific company $c$ and a timestamp $m$, we can draw the corresponding flow in/out/transfer value of the company $c$ at the time period $m-1$. We treated them as company-specific features. The features used in our model are summarized in Table 2.2. After preprocessing of the data, we set up the configuration of our HCPNN based on our preliminary experiments. The key dimensions and value settings of the model are reported in Table 2.3.

Table 2.2. The features used in HCPNN.

| | |
|---|---|
| **Personal Specific Feature** | |
| Number of social connections | Numerical |
| Self-introduction | Free text |
| **Company Specific Feature** | |
| Job duration at the company | Numerical |
| Company personnel flow (in/out/transfer) | Numerical |
| Company description | Free text |
| Company id, Size ,Type, Location, Age | Categorical |
| **Position Specific Feature** | |
| Duration in the position | Numerical |
| Position type | Categorical |

Table 2.3. The network configuration of HCPNN.

| **Name** | **Dimension/value** |
|---|---|
| duration embedding | 10 |
| company id embedding | 50 |
| position id embedding | 5 |
| company description words embedding | 50 |
| personal description words embedding | 50 |
| hidden states of company LSTM layer | 150 |
| hidden states of position LSTM layer | 20 |
| output dimension of local attention layer | 20 |
| output dimension of global attention layer | 80 |
| dropout probability | 0.9 |
| number of samples in a batch | 64 |
| the loss tuning parameter $\alpha$ | 0.5 |
| segmentation to compute the integration of survival analysis $z$ | 21 |

### 2.4.2 Baselines

We compared our model with state-of-the-art techniques, which are listed as follows: non-sequential models (*e.g.*, Logistic Regression (LR), Random Forest (RF), and Decision Tree (DT)), sequential models (*e.g.*, Conditional Random Field (CRF) (Lafferty, McCallum, & Pereira, 2001), Continuous Time Markov Chain (CTMC) (Anderson, 2012) [1] ) , and the stochastic time series models (*e.g.*, Poisson Process (PP) (Karr, 2017), Multi-variable Hawkes Process (MHP) (Mei & Eisner, 2017)). Also, we tested two modified versions of our method HCPNN named HCPOP and HCPOS. HCPOP model does not contain internal transition representation layer, while HCPOS does not contain the survival analysis technique, the job duration prediction was treated as a classification problem. We modified the CRF and MHP to fit our problems, the technique details will be introduced in Appendix A.1.3.

### 2.4.3 Evaluation Metrics

For predicting the next employer, we use *Accuracy@k* (*Acc@k*) and *mean reciprocal rank* (*MRR*) to evaluate the results, where $Acc@k = \frac{1}{N}\sum_{i=1}^{N} I(rank(i) \leq k)$, and $MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank(i)}$, where the $N$ is the total number of predictions, and $rank(i)$ represents the real label rank in the predicting ranking list. If $rank(i) \leq k$, then $I(rank(i) \leq k)$ equals one, else equals zero. In this experiment, we set $k = 1, 15, 30$ respectively. The higher value of *Acc@k* and *MRR*, the better performance. For predicting the job duration, we use *mean absolute error* $MAE = \frac{1}{N}\sum_{i=1}^{N} |p_i - r_i|$ and *Root Mean Square Error* $RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(p_i - r_i)^2}$ to evaluate the performances.

[1] https://github.com/kmedian/ctmc

$p_i$ and $r_i$ are the predicted job duration and the real job duration.

### 2.4.4 The Overall Performance

To validate the effectiveness of our model, we first randomly split the samples by (0.8/0.1/0.1) as the training/validation/test datasets. And the overall performance including predicting next employer and job duration respectively. The results of predicting next employer are reported in Table 2.4. We calculated the improvements of our model against all the other baselines. We can observe that the tree-based model are not able to effectively handle this predicting task, and the performances of the sequential models are better than that of non-sequential models. Our model has the best performances with significant improvements. For example, we achieved improvements of 231.8%, 160.3%, 121.4%, and 609.1%, in terms of *Acc@1*, *Acc@15*, *Acc@30*, and *MRR*, against the DT. Comparing to the best baseline, CTMC, our model also resulted in a consistent superior. To validate the improvement of HCPNN over HCPOP is statistically significant, we randomly split the data by (0.8/0.2) ten times, and conducted a standard student t-test. As the results, the p-value is very small for both employer and duration predictions, demonstrating a statistically significant improvement, and validating the importance of internal job mobility representation layer in our model. More detailed results about t-test are reported in Appendix A.1.4 Table A.3.

The results of predicting job duration are summarized in Table 2.5. Similar to Table 2.4, we computed the performance improvement of HCPNN against all the other baselines. We can observe that the stochastic time series models and the variant of our

model HCPOS achieved relative better performances, indicating that the task should be considered as a time series problem. Our model achieved the best performance with obvious advantages, while HCPOS, which uses the same structure but without the survival analysis, resulted in worse performance, even comparing it to MHP and PP. These results confirm the importance of our framework as well as the survival analysis on the duration prediction task.

Table 2.4. The overall performance (next employer prediction).

| **Model** | **Acc@1** | **Improvement** | **Acc@15** | **Improvement** | **Acc@30** | **Improvement** | **MRR** | **Improvement** |
|---|---|---|---|---|---|---|---|---|
| DT | 0.022 | 231.8% | 0.156 | 160.3% | 0.243 | 121.4% | 0.022 | 609.1% |
| RF | 0.021 | 247.6% | 0.157 | 158.6% | 0.250 | 115.2% | 0.021 | 642.9% |
| LR | 0.054 | 35.2% | 0.313 | 29.7% | 0.420 | 28.1% | 0.120 | 30.0% |
| CRF | 0.053 | 36.5% | 0.320 | 26.9% | 0.433 | 24.2% | 0.120 | 30.0% |
| CTMC | 0.060 | 21.7% | 0.336 | 20.7% | 0.457 | 17.6% | 0.089 | 75.4% |
| HCPOP | 0.071 | 2.8% | 0.402 | 1.0% | 0.534 | 0.7% | 0.154 | 1.3% |
| **HCPNN** | **0.073** | - | **0.406** | - | **0.538** | - | **0.156** | - |

[1] The improvement of our HCPNN over HCPOP is statistically significant with a p-value consistently less than 0.01.

Table 2.5. The overall performance (duration prediction).

| **Model** | **MAE** | **Improvement** | **RMSE** | **Improvement** |
|---|---|---|---|---|
| DT | 3.839 | 28.8% | 5.608 | 31.3% |
| RF | 4.070 | 32.8% | 5.782 | 33.4% |
| LR | 3.096 | 11.7% | 4.857 | 20.7% |
| CTMC | 4.128 | 33.8% | 5.872 | 34.4% |
| PP | 3.143 | 13.0% | 4.228 | 8.9% |
| MHP | 3.029 | 9.7% | 4.214 | 8.6% |
| HCPOS | 3.095 | 11.7% | 4.898 | 21.4% |
| HCPOP | 2.739 | 0.2% | 3.880 | 0.7% |
| **HCPNN** | **2.734** | - | **3.852** | - |

The improvement of our HCPNN over HCPOP is statistically significant with a p-value consistently less than 0.01.

### 2.4.5 Robustness Analysis

We also conducted additional experiments to confirm the robustness of our method. We first randomly split the dataset by samples with different training proportion settings (i.e., 90%, 80%, 70%, 60%, and 50%), the results of which are reported in Table 2.6. We can observe that with the training proportion increasing, the performance is improving as well. Furthermore, we split the dataset by years as well, for instance, if we set the splitting year as 2005, the whole sample sequences will be truncated by the year 2005, the points in a sequence before the year 2005 will be used for training,

and the points in the sequence after 2005 will be used for predicting. The results are shown in Table 2.7. We can observe that with the splitting year approaching recent, the performance improves. The results of two different splitting settings are stable, demonstrating the robustness of our model HCPNN.

Table 2.6. The performance on randomly split samples.

| **Ratio** | **Acc@1** | **Acc@15** | **Acc@30** | **MRR** | **MAE** | **RMSE** |
|---|---|---|---|---|---|---|
| 0.9 | 0.074 | 0.405 | 0.538 | 0.157 | 2.729 | 3.855 |
| 0.8 | 0.072 | 0.403 | 0.534 | 0.155 | 2.732 | 3.892 |
| 0.7 | 0.071 | 0.401 | 0.532 | 0.154 | 2.722 | 3.912 |
| 0.6 | 0.070 | 0.398 | 0.528 | 0.152 | 2.746 | 3.884 |
| 0.5 | 0.068 | 0.393 | 0.524 | 0.149 | 2.724 | 3.919 |

Table 2.7. The performance on splitting data by years.

| **Year** | **Acc@1** | **Acc@15** | **Acc@30** | **MRR** | **MAE** | **RMSE** |
|---|---|---|---|---|---|---|
| 2005 | 0.042 | 0.297 | 0.419 | 0.106 | 2.692 | 3.517 |
| 2006 | 0.041 | 0.313 | 0.440 | 0.109 | 2.556 | 3.366 |
| 2007 | 0.043 | 0.313 | 0.437 | 0.109 | 2.566 | 3.271 |
| 2008 | 0.045 | 0.328 | 0.455 | 0.115 | 2.651 | 3.241 |
| 2009 | 0.046 | 0.331 | 0.460 | 0.116 | 2.466 | 2.999 |
| 2010 | 0.048 | 0.340 | 0.470 | 0.120 | 2.277 | 2.796 |

### 2.4.6 Attention Analysis

With the attention mechanism, our HCPNN model offers new opportunities to investigate the importance of considered factors and related patterns in the job-mobility prediction tasks. Here, we show some examples in which we study the characteristics of three job-mobility factors, including the job duration, the firm type, the time index of career paths.

In Figure 2.7 (a) - (b), each column represents a time index which is set to be the position distance prior to the last job. For example, the last job has a time index of zero; the one before the last job has a time index of 1, and so forth. Each row represents the duration of a job, and the color of each grid shows the mean value of attention. The brighter of the color, the higher attention. The grids in white are missing values (no observation). Two interesting patterns can be found: (1) The longer stay with an employer, the higher attention (importance) it has; (2) A job appearing in a later position in one's career path has higher attention. Specifically, we find that 76.8% of the people in our sample have the highest attention weights for their last jobs.

On the other hand, the firm type also matters. As demonstrated in Figure 2.7 (c) - (d), an interesting pattern can be found. In general, with the job duration increases, the importance of an employer increases as well. However, this pattern is reversed for government-based organizations. That is, the longer people stay in the government, the less attention it has for the job mobility.

### 2.4.7 Individual Effect and Firm Effect

We also find the evidence of the existence of individual effect and organization effect in the predictions. We showcase the importance of number of social connections in the job mobility prediction in Figure 2.8. As can be seen, with the number of social connections increases, the attention increases as well. Moreover, the HCPNN will pay more attention to personal information when predicting next employer than predicting job duration. These findings are consistent with (Wegener, 1991) regarding the relationship between social ties and job mobility. We also evaluate the mean attention grouped by companies and plot the sorted attention in Figure 2.9, which appears as a sinh curve. We report the top-10 companies with the highest attention and compare them to the top-10 companies with highest occurrence number. The result is listed in Table 2.8. None of those companies overlapped. As can be seen, most of the top-10 companies with highest attention are emerging high-tech companies, while the most of the top-10 companies with highest occurrence frequency are relative old famous companies.

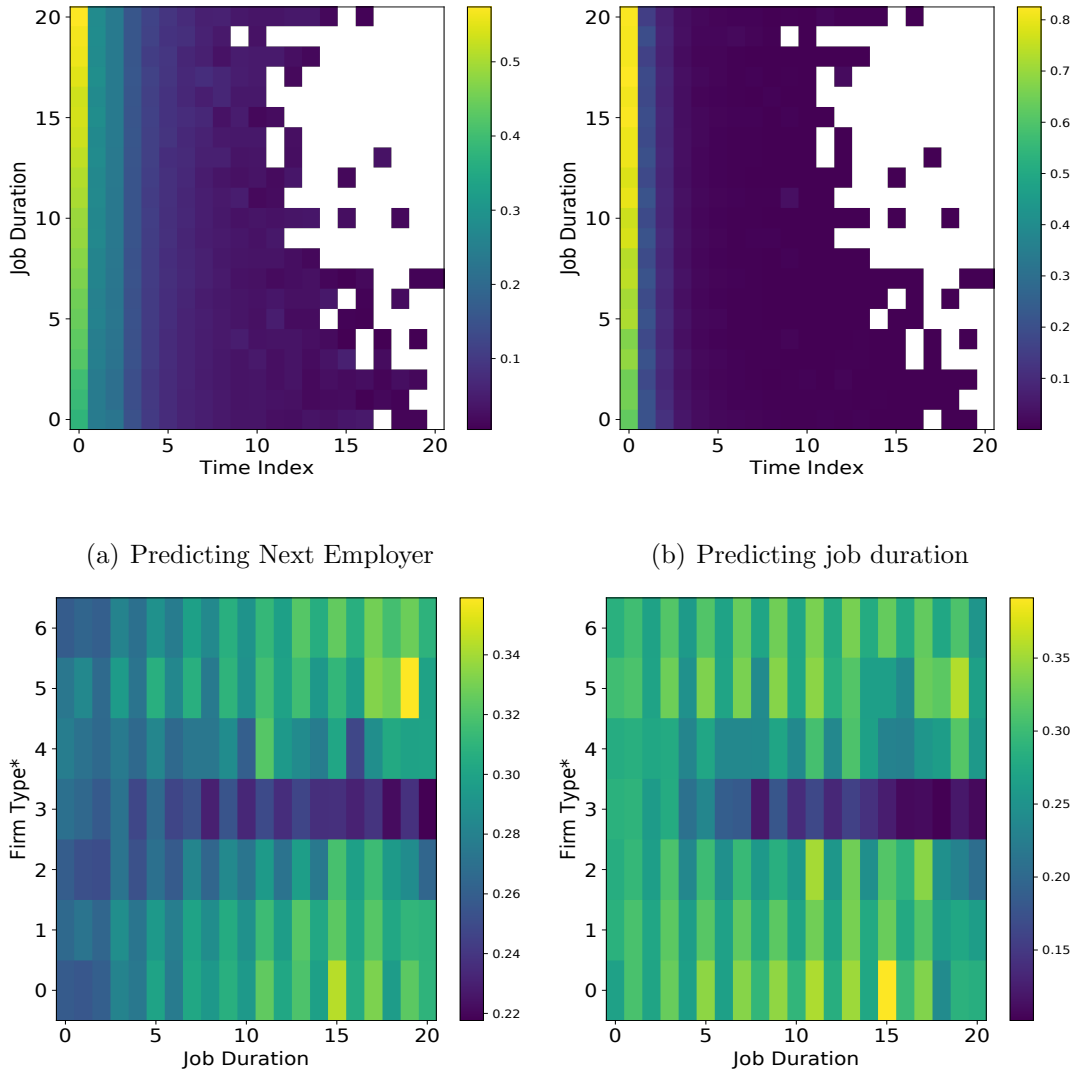
### 2.4.8 Individual-level Turnover Analysis

To analyze the patterns of turnover probability for individuals, we gathered the individual-level turnover probability for all samples and plot them in Figure 2.10. We can observe that with the working years increasing, the instantaneous turnover probability steady increases too. We also found an interesting phenomenon, which shows the individual turnover probability follows a "sawtooth" shape. This is consistent with our findings regarding the job duration distribution, as shown in Figure

2.3 (b). The pattern indicates that people tend to stay with an employer for integer years rather than odd number of half years. Our model learned this pattern without any pre-defined constraints.

Table 2.8. Attention on companies.

| **Top 10 companies with highest attention** |
|---|
| Facebook, LinkedIn, SapientNitro, GE Oil & Gas |
| Amazon Web Services, BBVA Compass bank, inVentiv Health |
| IndusInd Bank, Societe Generale Corporate and Investment Banking |
| Everything Everywhere (EE) |
| **Top 10 companies with highest occurrence number** |
| PricewaterhouseCoopers, Deloitte, Microsoft |
| Oracle, JPMorgan Chase, Bank of America, Citibank |
| Accenture, Hewlett Packard Enterprise, IBM |




(a) Predicting Next Employer

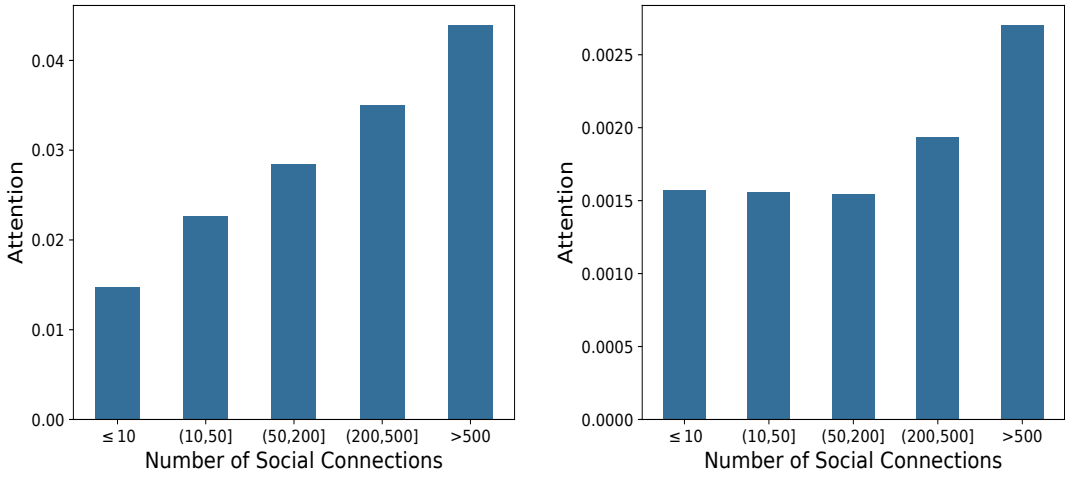
(b) Predicting job duration

(c) Predicting Next Employer

(d) Predicting job duration

Figure 2.7. The attention analyses of job mobility patterns.

(* 0: Sole Proprietorship, 1: Privately Held, 2: Joint Venture, 3: Government, 4: Educational Institution, 5: Non-Profit Organization, 6: Public.)




(a) Predicting Next Employer (b) Predicting job duration

Figure 2.8. The attention on number of social connections.

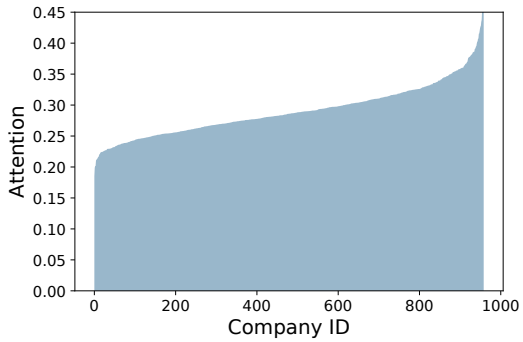



Figure 2.9. The sorted attention value on companies.

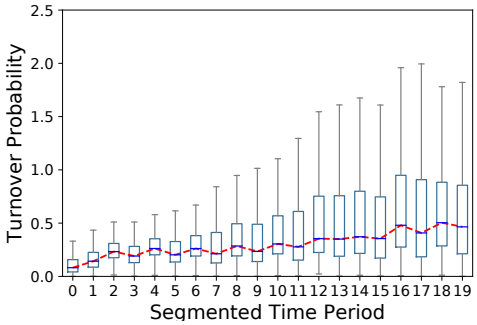



Figure 2.10. The turnover probability over time before 10 years.

## 2.5 Related Work

Career path analysis is a hot topic in management and psychology fields due to its significant values for guiding the decision-making process of organizations as well as individuals. Those works were largely based on limited survey data and gave qualitative analyses of key factors that would influence one's career path (Miller, 2011; Vance, 2005). Recent years, AI technology has enhanced the development and re-designed the paradigm of human resource management in many aspects (Meng, Zhu, Xiao, & Xiong, 2018; Qin et al., 2018; C. Zhu, Zhu, Xiong, Ding, & Xie, 2016; Shen et al., 2018; H. Xu, Yu, Yang, Xiong, & Zhu, 2016), of the area, career path analysis is one hot target problem. For example, (L. Li et al., 2017) designed a neural network framework to predict the next employer and positions together. (H. Li, Ge, Zhu, Xiong, & Zhao, 2017) proposed a survival analysis to model the promotion and turnover within one company, which is different from our trans-company analysis. (H. Xu, Yu, Yang, Xiong, & Zhu, 2018) analyzed the talent flow into and out of the target organizations, regions, or industries.

The technologies used in our model are associated with recurrent neural networks, as well as sequential event data analysis. Various recurrent neural network approaches have been developed to address the time series problem, such as LSTM (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Unite (GRU)(Cho, Van Merriënboer, Gulcehre, et al., 2014). These techniques have been widely used due to their strong performance as well as the ability to capture long-term temporal dependencies, especially in the text mining and image recognition areas (L. Zhang et al., 2018; Cho,

Van Merriënboer, Bahdanau, & Bengio, 2014). After that, attention-based model are introduced to improve the prediction power of RNNs further (Bahdanau, Cho, & Bengio, 2014; Luong, Sutskever, Le, Vinyals, & Zaremba, 2015). Recent years, sequential event data and survival analysis models have been developed to solve various problems (Ye et al., 2018; Du et al., 2016; Mei & Eisner, 2017). (Jing & Smola, 2017) applied RNN to model the user return pattern of a musician application. (Ren et al., 2019) proposed a deep learning model to analyze both censored and uncensored data. Our research is different from the above works in two aspects. First, we use a hierarchical LSTM and attention mechanism to model a hierarchical sequence data. Second, we do not suppose any preliminary assumptions on the form of hazard rate, as the preliminary assumptions may be against the true nature of the real values.

## 2.6 Concluding Remarks

In this chapter, we focused on understanding job mobility at an individual level. Specifically, the goal is to predict the next potential employer of an individual and how long he/she will stay in the new position. Along with this line, we proposed a *hierarchical career-path-aware neural network* for answering these two questions. Our approach was designed to provide a certain level of interpretability by embedding the attention mechanism. As shown in our experimental results, our method provided much better accuracy for both prediction tasks. Finally, based on the assigned attention, we also provided data-driven evidence to show the importance of various factors (*e.g.*, job duration, firm type, *etc.* ) for job mobility prediction.

CHAPTER 3

# INTELLIGENT SALARY BENCHMARKING FOR TALENT RECRUITMENT: A HOLISTIC MATRIX FACTORIZATION APPROACH

## 3.1 Introduction

Compensation and Benefits (C&B), one of the most important sub-disciplines of human resources, plays an indispensable role in attracting, motivating and retaining talents. A major part of C&B planning is salary benchmarking, which has a goal of identifying the market pay scales of employees with respect to different job positions. Indeed, comprehensive and accurate salary benchmarking can help companies to keep and strengthen their core competitiveness in the market.

Traditional approaches for salary benchmarking rely heavily on the experience from domain experts and market surveys provided by third-party consulting companies and governmental organizations (Johnson, Riggs, & Downey, 1987; Schau & Heyward, 1987; Porter, Cordon, & Barber, 2004), such as *OECD*[1] . However, the rapidly evolving technology and industrial structure result in the variation of positions and job requirements, leading to the difficulties in timely salary benchmarking under a dynamic scenario. For example, it is nontrivial for traditional approaches to timely benchmark salaries in the scenarios where there are millions of job-company

[1] http://www.oecd.org/

combinations with respect to many possible work locations and time periods.

Recently, the prevalence of emerging online recruitment services, such as Glassdoor, Indeed and Lagou, provide opportunities to accumulate massive job related-data from a wide range of companies, and thus enable a new paradigm for salary benchmarking in a data-driven way. To this end, in this chapter, we propose a method for intelligent salary benchmarking based on large-scale fine-grained online recruitment data. Specifically, we first construct an *expanded salary matrix* based on the recruitment data, in which time-specific job positions and location-specific companies are represented as rows and columns. In this way, the problem of salary benchmarking can be naturally formalized as a matrix completion task. Along this line, we develop a Holistic Salary Benchmarking Matrix Factorization (HSBMF) model for predicting the missing salary information in the salary matrix. Also, by integrating multiple confounding factors, such as company similarity, job similarity, and spatial-temporal similarity, the HSBMF model can provide a holistic and dynamic view of salary benchmarking. Indeed, with the help of HSBMF, we can obtain fine-grained salary benchmark with respect to different companies, job positions, time periods and locations. At last, we conduct extensive experiments based on large-scale real-world recruitment data to validate the effectiveness of our approach in terms of accurately identifying the market rates for job positions in various contexts.

To be specific, the contributions of this work can be summarized as follows:

- We propose a novel approach HSBMF for large-scale fine-grained job salary benchmarking based on the massive online recruitment data.

- We propose and validate four domain assumptions with respect to the recruitment market, and integrate them as confounding constraints into HSBMF, which can provide a holistic view of salary benchmarking.

- We evaluate the proposed approach with extensive experiments on a large-scale real-world dataset. The results clearly validate the effectiveness of our approach.

## 3.2 Preliminary Analysis

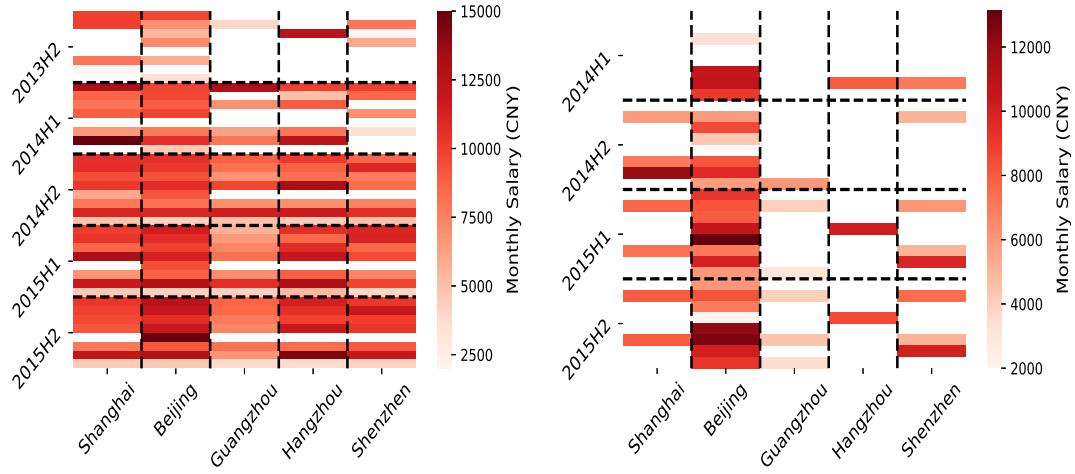
In this section, we briefly introduce the recruitment data used in our study and formalize the problem of fine-grained salary benchmarking. Also, we discuss the numerical characteristics of the data related to the design of our model.

### 3.2.1 Data Description

In this chapter, we aim to develop an effective method for salary benchmarking based on massive online recruitment data. Our data were collected from a major online recruitment website in China, which consist of more than 700,000 job postings from more than 50,000 high-tech companies during a three-year time interval. The information of each job posting contains posting time, job details (e.g., job title, work location and job description), company details (e.g., company name, industry category, company size, and financial stage), and a scale of expected monthly salary (e.g., lower bound and upper bound). More details of the data will be discussed in Section 3.4. Indeed, the information similar to our recruitment data is generally available worldwide. Therefore, the method developed in this chapter should be able to easily applied to a broader job market.

One of the most important jobs for C&B is salary benchmarking, which aims at identifying the appropriate market pay scale for each job position. One intuitive solution is to predict salary scales with respected to specific job requirements. However, based on real-world cases, it can be commonly found that companies offer different pay levels to similar job positions. Even for the same job-company combinations, salaries vary a lot at the different time and work locations. For example, one corporation may offer quite different salaries to two software developers, of which one works at New York while the other works at Nashville, even though their work duties are similar. Thus, we believe it is necessary to develop a more delicate salary benchmarking method to support the decision making process for C&B. An effective approach for salary benchmarking should be able to handle job positions of different companies under different contexts, such as work locations and posting time.

Figure 3.1 demonstrates a snippet of salary distribution in our real-world dataset. We randomly selected eight job positions and companies and plot their salary heatmap at different locations and time periods. As can be seen, salaries at different time intervals and locations vary a lot. Unfortunately, due to a large number of job-company-context combinations, it is impossible to directly obtain all of their salary observations, even for the massive online recruitment data, as the blank areas presented in Figure 3.1. Therefore, in this chapter, we propose a novel approach for fine-grained salary benchmarking to effectively predict expected salaries for unobserved job-company-context combinations.




(a) Position (b) Company

Figure 3.1. A snippet of salary distribution in our data. Here, each grid represents a specific job position or company.

### 3.2.2 Fine-Grained Salary Benchmarking

Traditionally, the problem of salary benchmarking is to estimate the expected salary level (e.g., the lower/upper bound of salary) of each job position offered by a specific company. The classical method is straightforward and a common procedure is as follows. It firstly constructs a job-company salary matrix, where each entry indicates the corresponding salary. Then, it formalizes the problem as a matrix completion task. However, an important issue is that the traditional method is usually too general to satisfy various special needs of C&B professionals, because only the job-company matrix is considered. To this end, we propose to address the salary benchmarking problem in a fine-grained manner by considering more contextual information, such as work locations and posting time. To be specific, we define the problem of fine-grained salary benchmarking as follows.

***Problem Statement*** *(Fine-Grained Salary Benchmarking)*: Given a specific combination of companies, work locations, and posting time, the objective is to estimate the expected salary level of each job position (e.g., estimating the lower/upper bound of the salary for a software engineer of a company located in NYC in the first half year of 2017).

To address the problem, we propose an *expanded salary matrix* by expanding original job-company salary matrix with locations and time information. For example, Figure 3.2 shows the structure of our expanded salary matrix, where the company and job dimensions are expanded with work locations and posting time respectively. One motivation for the matrix expanding process is that each company usually has multiple work sites for talent recruitment, while the salary of each job position drifts along time. A more sophisticated explanation to the design of the salary matrix is highly related to the data characteristics, and we will provide more detailed discussions in Section 3.2.3. Along this line, the problem of fine-grained salary benchmarking is naturally equivalent to the task of estimating missing values in the expanded salary matrix.

### 3.2.3 Numerical Characteristics of the Data

Before introducing the technical details of our approach to job salary benchmarking, here we discuss some important numerical characteristics, which may significantly affect job salaries and motivate the design of our HSBMF model.

First, we check the relationship between job similarity and salary. Intuitively, positions with similar job descriptions should have similar salary scales. Therefore, the

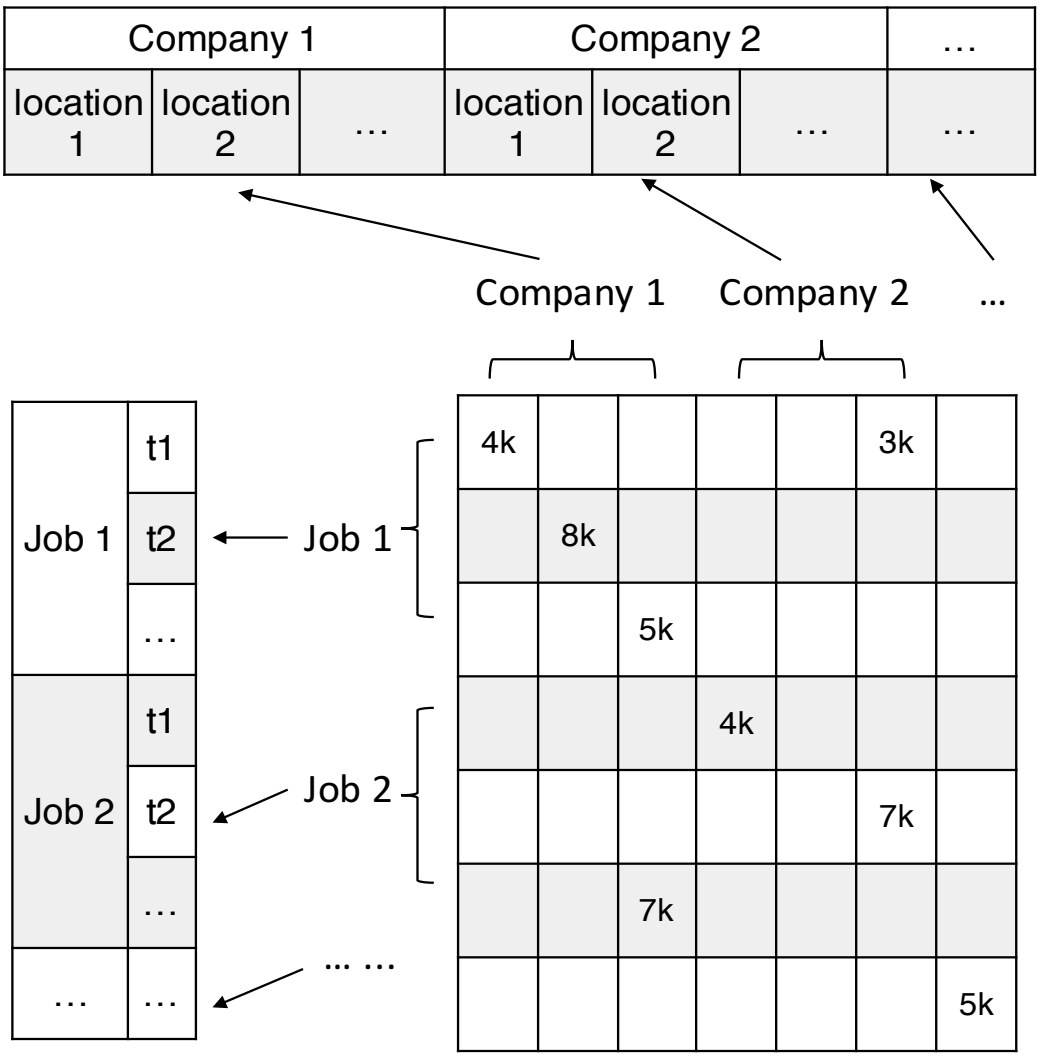
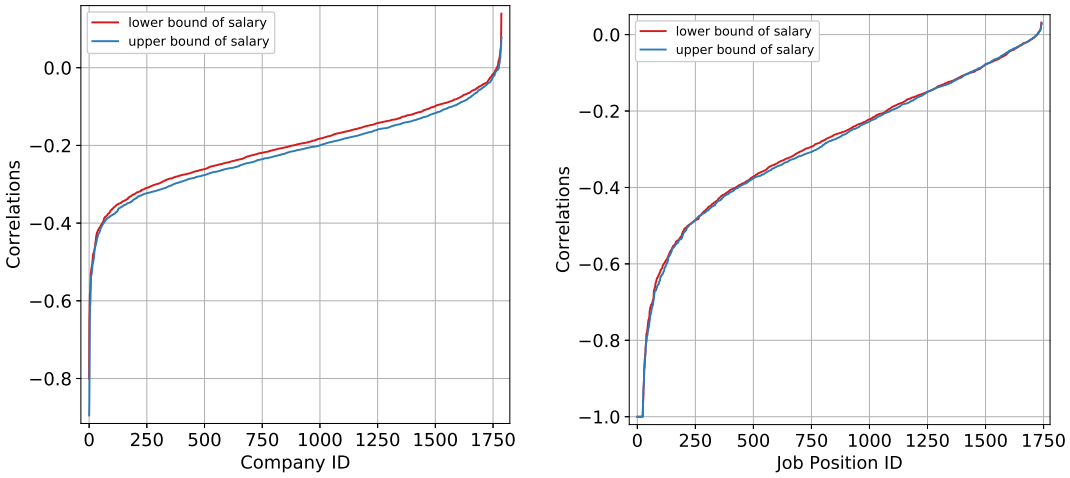



Figure 3.2. The structure of the expanded salary matrix.

similarities between job positions should be negatively correlated to related salary differences. Following that, we calculate the pair-wise similarities between its job descriptions and corresponding salary differences, and then compute their Pearson correlation coefficient (The details of how to calculate the pair-wise similarities will be introduced in Section 3.3.2.). Figure 3.3 (a) shows the sorted "job similarity-salary difference" correlations grouped by companies. As can be seen, most of the correlations fall into the negative range, which is consistent with our domain assumption.




(a) (b)

Figure 3.3. The correlation between job/company similarity and salary difference.

Second, we study the relationship between company similarity and job salary. Intuitively, companies in the same business sector and with comparable sizes should provide positions with similar rate scales. Thus, the similarities between companies should have a negative correlation with their salary differences. We follow the similar approach as discussed before to calculate the "company similarity-salary difference" correlation for every job position. The result is plotted in Figure 3.3 (b), and we find it is consistent with our assumption as well.

Third, we investigate the relationship between posting time and salary. We group the data in two ways for calculating the correlations. Intuitively, the differences of job salary should have the positive correlation with their posting time intervals. Thus, we calculate the "time interval-salary differences" Pearson correlation coefficient for every job position and company respectively and report the results in Table 3.1. We can observe that the correlations are positive for both grouping methods. Moreover, it can be found that the correlation grouped by job positions is higher than that grouped by

companies, indicating a stronger "time interval-salary difference" relationship when grouping the data by job positions.

Last, we investigate the relationship between work location and job salary. We also group the data by job positions and companies respectively. Intuitively, the differences of job salary should hold a positive correlation with the average revenues of their work locations. To this end, we calculate the Pearson correlation between the government-released average revenues and corresponding average job salaries in different locations. The results are report in Table 3.1. The positive values clearly support our domain assumption. Moreover, the correlations grouped by companies are higher than that grouped by job positions, suggesting a stronger "location similarity-salary difference" relationship when grouping the data by companies.

Table 3.1. The Pearson correlation between posting time/work location similarity and salary difference.

| Grouping Method | Lower Bound | | | | Upper Bound | | | |
|---|---|---|---|---|---|---|---|---|
| | Time-Salary | | Location-Salary | | Time-Salary | | Location-Salary | |
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| Job Position | **0.341** | **0.802** | 0.248 | 0.528 | **0.281** | **0.701** | 0.208 | 0.465 |
| Company | 0.244 | 0.734 | **0.328** | **0.738** | 0.222 | 0.697 | **0.354** | **0.751** |

Following the above results, we design the expended salary matrix as demonstrated in Fig 3.2 (i.e., time-specific job positions and location-specific firms are represented as rows and columns). In summarize, we identify four confounding factors, including

job similarity, company similarity, and time-spatial similarities, which have significant impacts on salary benchmarking. In Section 3.3.2, we will provide technical details regarding how we calculate those similarities and integrate them into HSBMF model for higher performance.

## 3.3 Matrix Factorization for Salary Benchmarking

In this section, we introduce the technical details of our HSBMF model for fine-grained salary benchmarking. Important mathematical notations used throughout this chapter are summarized in Table 3.2.

### 3.3.1 A Basic Model

Matrix Factorization (MF) is among the most widely-used methods for recommendation systems. It aims to factorize an incomplete user-item rating matrix into two lower rank latent matrices, and use their dot product for estimating the possible ratings of the missing entries. In this work, we follow the idea of biased SVD (bSVD) for salary benchmarking as suggested by (Koren, 2008; Paterek, 2007). Specifically, given an entry $S(j, c)$ in expanded salary matrix $S$, the predictor is equal to

$$\hat{S}(j, c) \approx \mu + B_j(j) + B_c(c) + J(j, :)C(c, :)^T, \tag{3.1}$$

where $\mu$, $B_j$, $B_c$ denote the global mean of $S$, the bias vector of job position, and the bias vector of company, respectively. Furthermore, by adding Frobenius norm regularization terms for avoiding the ill-posed problem (Luo, Zhou, Xia, & Zhu, 2014; Koren & Bell, 2015), we can formulate the preliminary loss function for salary benchmarking

as

$$
\begin{aligned}
min : \mathcal{F} = & \sum_{j=1}^{M}\sum_{c=1}^{N}(I_s(j,c) \circ (S(j,c) - \hat{S}(j,c)))^2 \\
& + \lambda_J ||J||_F^2 + \lambda_C ||C||_F^2 + \lambda_{B_j} ||B_j||_F^2 + \lambda_{B_c} ||B_c||_F^2,
\end{aligned} \tag{3.2}
$$

where $\circ$ means element-wise multiplication of two matrices, and $I_S$ is the indicator matrix of $S$, which is defined as

$$
I_S(j,c) = \begin{cases} 1, & S(j,c) \text{ exists}, \\ 0, & \text{else}. \end{cases} \tag{3.3}
$$

### 3.3.2 HSBMF with Holistic Constraints

To further refine the performance of salary benchmarking, we integrate more confounding factors as constraints into Equation 3.2, including the company similarity, job similarity, and spatial-temporal similarity.

The first constraint is to reveal the relationship between job similarity and salary. Intuitively, job positions with similar job descriptions tend to have similar salary scales. Thus, we formulate the ***Job Similarity Regularizer*** as

$$
\begin{aligned}
R_J &= \frac{1}{2}\sum_{j=1}^{M}\sum_{j'=1}^{M} S_j(j,j') ||J(j,:) - J(j',:)||_F^2 \\
&= \sum_{j=1}^{M}\sum_{j'=1}^{M}\sum_{k=1}^{K} S_j(j,j') J(j,k)^2 - \sum_{j=1}^{M}\sum_{j'=1}^{M}\sum_{k=1}^{K} S_j(j,j') J(j,k) J(j',k) \\
&= \sum_{k=1}^{K} J(:,k)^T (D_{S_j} - S_j) J(:,k) \\
&= tr(J^T (D_{S_j} - S_j) J).
\end{aligned} \tag{3.4}
$$

where $tr(\cdot)$ represents the matrix trace, and $S_j(j,j')$ is the similarity between two job positions $j$ and $j'$, which is estimated by the Cosine similarity between the TF-IDF

vectors of corresponding job descriptions. $D_{S_j}$ is the degree matrix of $S_j$, which is defined as

$$D_{S_j}(u,v) = \begin{cases} \sum_{v=1}^{M} S_j(u,v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{3.5}$$

Here, we use the job similarity matrix $S_j$ to regularize the learning process of job position latent matrix $J$, which guarantees that the components of $J$ will be similar if their corresponding job descriptions are similar.

Second, we propose another ***Company Similarity Regularizer***, which guarantees that similar companies should offer jobs with similar salary levels. Specifically, the regularizer is formulated as

$$\begin{aligned} R_C &= \frac{1}{2}\sum_{c=1}^{N}\sum_{c'=1}^{N} S_c(c,c')||C(c,:) - C(c',:)||_F^2 \\ &= tr(C^T(D_{S_c} - S_c)C), \end{aligned} \tag{3.6}$$

where $S_c(c,c')$ is the similarity between two companies $c$ and $c'$, which is estimated by the Jacquard similarities between the basic information of companies, such as company size, industry category, and financial stage. Similarly, $D_{S_c}$ is the degree matrix of $S_c$, which is defined as

$$D_{S_c}(u,v) = \begin{cases} \sum_{v=1}^{N} S_c(u,v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{3.7}$$

In addition to the above constraints, we also propose to explore spatial-temporal related regularizers. Specifically, we propose a ***Time-Aware Regularizer*** to evaluate the relationship between posting time and salary. Intuitively, the differences of salaries should have the positive correlation with their posting time intervals. To this end, inspired by (Yao et al., 2017; Gao, Tang, Hu, & Liu, 2013), we assume that

the salary of a job at the current time is influenced by its historical salaries, and the degree of influences is affected by corresponding time spans. Therefore, we define the temporal correlation $\rho(j, j')$ between job $j$ and $j'$ as

$$\rho(j, j') = exp(-\alpha|\tau_j - \tau_{j'}|), \tag{3.8}$$

where $\alpha$ is a positive parameter that controls the temporal evolutionary process, and $\tau_j$ is the posting time of job position $j$ (note that, in the expanded salary matrix, every job position is associated with a posting time). Moreover, if $\alpha = 0$, all job salaries have equal correlations without considering corresponding time spans. On the contrary, if $\alpha \to +\infty$, salaries of jobs will not have any temporal relationships. Furthermore, the time-aware regularizer can be defined as

$$\begin{aligned} R_T &= \frac{1}{2}\sum_{j=1}^{M}\sum_{j'=1}^{M} T(j, j')||J(j,:) - J(j',:)||_F^2 \\ &= Tr(J^T(D_T - T)J), \end{aligned} \tag{3.9}$$

$$D_T(u, v) = \begin{cases} \sum_{v=1}^{M} T(u, v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{3.10}$$

$T$ is a temporal transition matrix, which is defined as

$$T = \begin{bmatrix} 1 & \rho(1,2) & \cdots & \rho(1,M) \\ \rho(2,1) & 1 & \cdots & \rho(2,M) \\ \vdots & \vdots & \vdots & \vdots \\ \rho(M,1) & \rho(M,2) & \vdots & 1 \end{bmatrix}_{MM}. \tag{3.11}$$

Finally, we introduce the ***Location-Aware Regularizer*** to evaluate the relationship between work locations and salary. Indeed, the salaries of job positions have positive correlations with the average income levels of their work locations. Thus,

we define a location awareness matrix $L$ to depict the relationship between two jobs positions in different work locations, of where $\varphi(c, c')$ denotes the entry, which can be computed as follows:

$$\varphi_(c, c') = 1 - \frac{|AS_c - AS_{c'}|}{max(AS_c, AS_{c'})}, \tag{3.12}$$

where $AS_c$ is the average salary of company $c$'s location (note that, in the expanded salary matrix, every company is associated with a specific location). Furthermore, we define the location-aware regularizer as

$$\begin{aligned} R_L &= \frac{1}{2}\sum_{c=1}^{N}\sum_{c'=1}^{N} L(c, c')||C(c, :) - C(c', :)||_F^2 \\ &= Tr(C^T(D_L - L)C), \end{aligned} \tag{3.13}$$

$$D_L(u, v) = \begin{cases} \sum_{v=1}^{N} L(u, v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{3.14}$$

With above holistic constraints, we can obtain the final loss function of our HSBMF model by integrating Equation 3.2 with all regularizers. That is,

$$\begin{aligned} min : \mathcal{F} = \frac{1}{2}\Big( &\sum_{j=1}^{M}\sum_{c=1}^{N}(I_s(j, c) \circ (S(j, c) - \hat{S}(j, c)))^2 \\ &+ \lambda_J||J||_F^2 + \lambda_C||C||_F^2 + \lambda_{B_j}||B_j||_F^2 + \lambda_{B_c}||B_c||_F^2 \\ &+ \lambda_{S_j} tr(J^T(D_{S_j} - S_j)J) + \lambda_{S_c} tr(C^T(D_{S_c} - S_c)C) \\ &+ \lambda_T tr(J^T(D_T - T)J) + \lambda_L tr(C^T(D_L - L)C)\Big). \end{aligned} \tag{3.15}$$

In summary, Figure 3.4 shows the graphical representation of the HSBMF model.

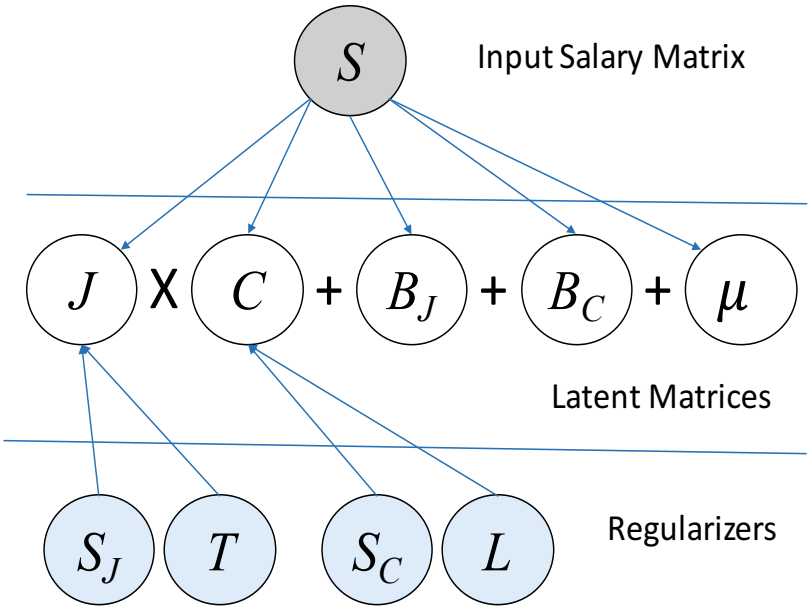



Figure 3.4. The graphical representation of our HSBMF model.

### 3.3.3 Algorithm Optimization

Here, we introduce how to use the gradient descent approach to learn our HSBMF model. The goal is to learn the parameters $J$, $C$, $B_j$ and $B_c$. Specifically, with the partial derivatives of $\mathcal{F}$ in (3.15), we have

$$\frac{\partial \mathcal{F}}{\partial J(j,k)} = -\sum_{c \in I_J(j)} (S(j,c) - \hat{S}(j,c))C(c,k) + |I_J(j)|$$
$$\times \Big(\lambda_{S_j}(D_{S_j} - S_j)J(j,k) + \lambda_T(D_T - T)J(j,k) + \lambda_j J(j,k)\Big),$$

$$\frac{\partial \mathcal{F}}{\partial C(c,k)} = -\sum_{j \in I_C(c)} (S(j,c) - \hat{S}(j,c))J(j,k) + |I_C(c)|$$
$$\times \Big(\lambda_{S_c}(D_{S_c} - S_c)C(c,k) + \lambda_L(D_L - L)C(c,k) + \lambda_c C(c,k)\Big),$$

$$\frac{\partial \mathcal{F}}{\partial B_j(j)} = -\sum_{c \in I_J(j)} ((S(j,c) - \hat{S}(j,c)) + |I_J(j)|\lambda_{B_j} B_j(j),$$

$$\frac{\partial \mathcal{F}}{\partial B_c(c)} = -\sum_{j \in I_C(c)} ((S(j,c) - \hat{S}(j,c)) + |I_C(c)| \lambda_{B_c} B_c(c),$$

where $I_J(j)$ denotes the set of companies at where $I_s(j,:)$ existing values, while $I_C(c)$ denotes the set of job positions at where $I_s(:,c)$ existing values.

Denoting the learning rate by $\gamma$, we get the updating rules of HSBMF as follows:

$$\begin{aligned} J(j,k) \leftarrow J(j,k) + \gamma \Big( & \sum_{c \in I_J(j)} \big(S(j,c) - \hat{S}(j,c)\big) C(c,k) \\ & - |I_J(j)| \times \big(\lambda_{S_j}(D_{S_j} - S_j) J(j,k) + \lambda_T (D_T - T) J(j,k) + \lambda_j J(j,k)\big) \Big), \end{aligned} \tag{3.16}$$

$$\begin{aligned} C(c,k) \leftarrow C(c,k) + \gamma \Big( & \sum_{j \in I_C(c)} \big(S(j,c) - \hat{S}(j,c)\big) J(c,k) \\ & - |I_C(c)| \times \big(\lambda_{S_c}(D_{S_c} - S_c) C(c,k) + \lambda_L (D_L - L) C(c,k) + \lambda_c C(c,k)\big) \Big), \end{aligned} \tag{3.17}$$

$$B_j(j) \leftarrow B_j(j) + \gamma \Big( \sum_{c \in I_J(j)} \big(S(j,c) - \hat{S}(j,c)\big) - |I_J(j)| \lambda_{B_j} B_j(j) \Big), \tag{3.18}$$

$$B_c(c) \leftarrow B_c(c) + \gamma \Big( \sum_{j \in I_C(c)} \big(S(j,c) - \hat{S}(j,c)\big) - |I_C(c)| \lambda_{B_c} B_c(c) \Big). \tag{3.19}$$

Here, we summarize the steps of optimization. First, we extract raw data from our dataset and construct the expanded salary matrix $S$, and calculate global mean $\mu$. Second, we calculate four auxiliary matrices, i.e.,$S_j$, $S_c$, $T$, and $L$, with corresponding degree matrices, i.e., $D_{S_j}$, $D_{S_c}$, $D_T$, and $D_L$. At last, the matrices $J$, $C$, $B_j$ and $B_c$

are initialized with random values and are updated with gradient decent rules. In particular, to improve the efficiency, we also introduce two variables $AuxiliaryJ$, $AuxiliaryC$ for avoiding the dot production of large-scale matrices in each iteration. Specifically, Algorithm 1 describes the detailed optimization process of the HSBMF model. Note that our software implementation is available from our project website.[2]

Last, we analyze the computation complexity of algorithm 1. There are three layers of iterations in the algorithm. If we don't consider some fast algorithms for matrix multiplication, steps 3-4 need $O(M^2+N^2)K$ time. Steps 6-10 need $O(K)$ time. Steps 12-15 need $O(K)$ time. Steps 6-10 combined with steps 12-15 need $O(|I_J||I_C|)K$ time, and steps 3-4 combined with steps 6-15 need $O\big((M^2+N^2+|I_J||I_C|)\times K\times Max\_Iter\big)$ time, which is the computation complexity of our algorithm.

[2]https://github.com/homeinsky/Salary-Benchmark-With-Matrix-Factorization

Table 3.2. The mathematical notations.

| Symbol | | Description |
|---|---|---|
| $S$ | $\in \mathbb{R}^{MN}$ | The expanded salary matrix |
| $I_s$ | $\in \mathbb{R}^{MN}$ | The indicator matrix of $S$ |
| $J$ | $\in \mathbb{R}^{MK}$ | The latent factor matrix of job position |
| $C$ | $\in \mathbb{R}^{NK}$ | The latent factor matrix of company |
| $S_j$ | $\in \mathbb{R}^{MM}$ | The similarity matrix of job position |
| $S_c$ | $\in \mathbb{R}^{NN}$ | The similarity matrix of company |
| $T$ | $\in \mathbb{R}^{MM}$ | The temporal transition matrix |
| $L$ | $\in \mathbb{R}^{NN}$ | The location awareness matrix |
| $B_j$ | $\in \mathbb{R}^{M1}$ | The bias vector of job position |
| $B_c$ | $\in \mathbb{R}^{N1}$ | The bias vector of company |
| $J^T$, $C^T$ | | The transpose matrix of $J, C$ |
| $\mu$ | | The global mean of expanded salary matrix |
| $\gamma$ | | The learning rate |
| $j, j'$ | | A row in $J$ |
| $c, c'$ | | A row in $C$ |

**Algorithm 1** HSBMF Optimization

**Input:**

$S$,$S_j$,$S_c$,$T$,$L$,$D_{S_j}$, $D_{S_c}$,$D_T$,$D_L$,$\mu$

$\lambda_j$,$\lambda_c$,$\lambda_{S_j}$,$\lambda_{S_c}$,$\lambda_T$,$\lambda_L$,$\lambda_{B_j}$,$\lambda_{B_c}$,$\gamma$,$\alpha$

**Output:** $J$, $C$, $B_j$, $B_c$

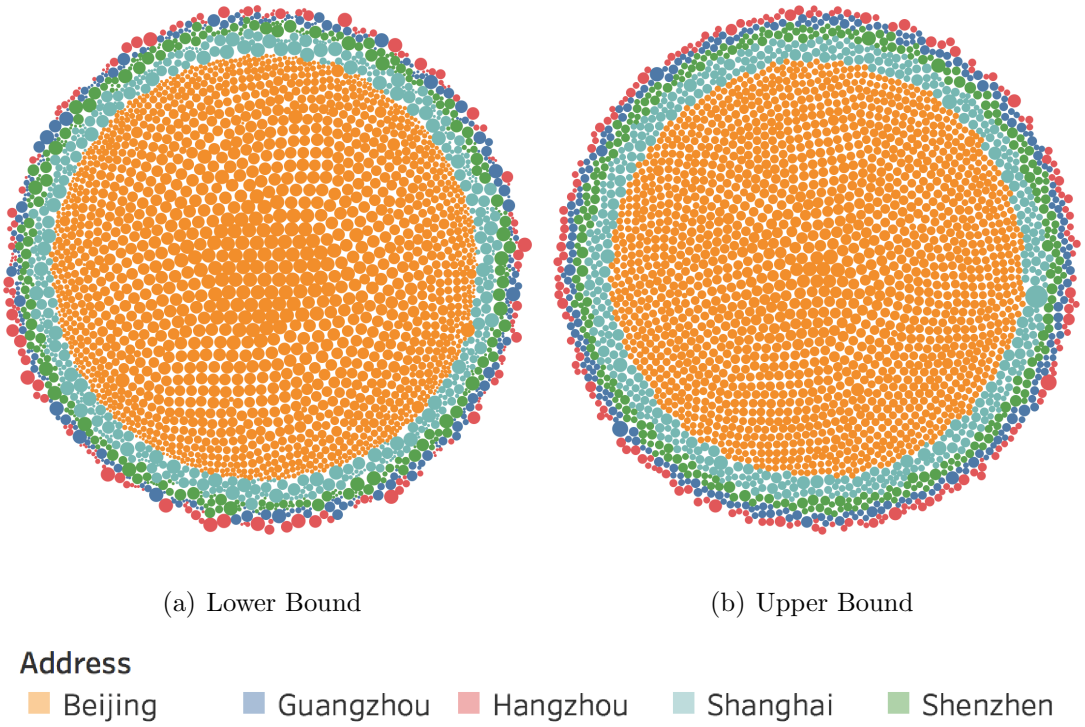
Initialize $J$, $C$, $B_j$, $B_c$ with random values
**while** Iterations $< Max\_Iter$ **do**
  $AuxiliaryJ = \big(\lambda_{S_j}(D_{S_j} - S_j) + \lambda_T(D_T - T) + \lambda_j\big)J$
  $AuxiliaryC = \big(\lambda_{S_c}(D_{S_c} - S_c) + \lambda_L(D_L - L) + \lambda_c\big)C$
  **for** each $(j, c)$ in the $S$ **do**
    $\hat{S} = \mu + B_j(j) + B_c(c) + J(j,:)C(c,:)^T$
    $err = S - \hat{S}$
    # update bias $B_j$ and $B_c$
    $B_j(j) = B_j(j) + \gamma\big(err - \lambda_{B_j}B_j(j)\big)$
    $B_c(c) = B_c(c) + \gamma\big(err - \lambda_{B_c}B_c(c)\big)$
    # update $J$ and $C$
    **for** each k **do**
      $J(j,k) = J(j,k) + \gamma\big(err * C(c,k) - AuxiliaryJ(j,k)\big)$
      $C(c,k) = C(c,k) + \gamma\big(err * J(j,k) - AuxiliaryC(c,k)\big)$
    **end for**
  **end for**
**end while**
**return** $C$,$J$,$B_j$,$B_c$

## 3.4 Experimental Results

In this section, we evaluate the performance of the HSBMF model for salary benchmarking.

### 3.4.1 The Experimental Setup

As introduced in Section 3.2, the real-world dataset was collected from a major online recruitment website in China, which consists of millions of job postings from thousands of high-tech companies from July 2013 to October 2015. To guarantee the effectiveness of our experiments, we preprocessed the data with the following steps. First, we removed the duplicates and structured job postings, and filtered companies that published less than 20 job postings, and job positions that appeared less than five times. Second, we only selected five large work locations in our dataset, including., "Beijing", "Shanghai", "Guangzhou", "Shenzhen" and "Hangzhou", since more than 80% job postings are located in these cities. Third, we grouped the posting time into 5 time periods, i.e., every half year belongs to one time period. Finally, we manually normalized different job titles, and grouped the similar titles into the same job position. After data preprocessing, we kept $132,061$ job postings which belong to $1,795$ job positions from $1,788$ companies. The sparsity of the expanded salary matrix is 99.5%. We can observe the companies' distribution over locations and their salary differences from Figure 3.5. We also plotted the scatter bubble chart for each location and time period in Figure 3.6. The five different colors represent five cities. The bubble scale is proportional to the number of distinct job positions. From the figure, we can observe that as time approaching recent, the number of distinct job

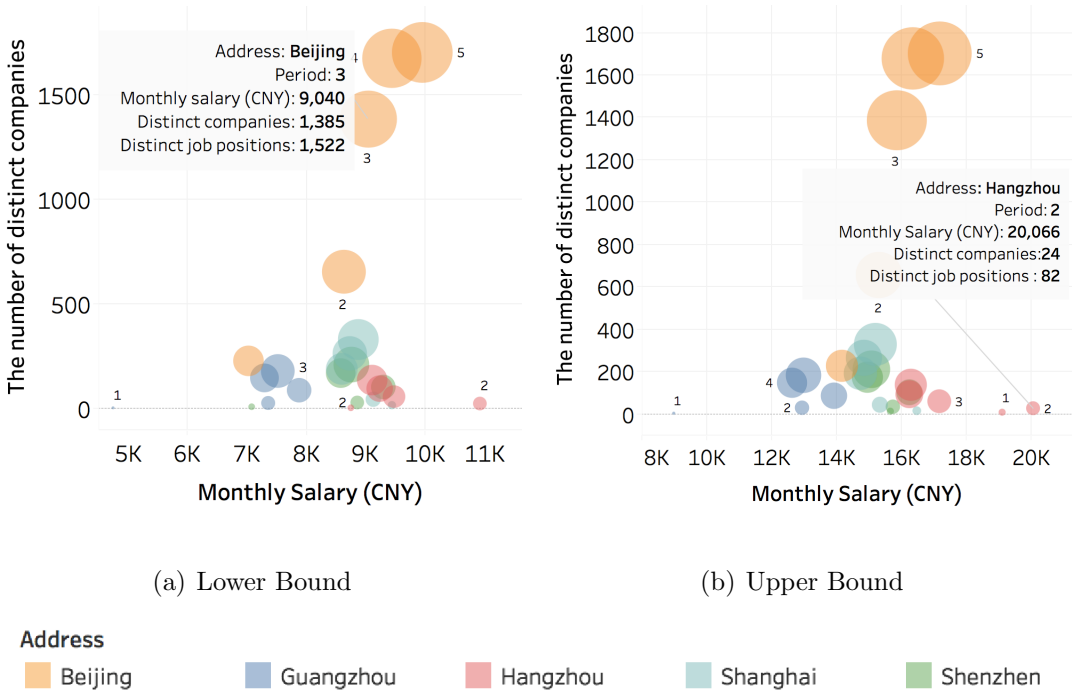



(a) Lower Bound (b) Upper Bound

Figure 3.5. The bubble chart of salary, where each bubble represents a company, and the scale is proportional to the value.

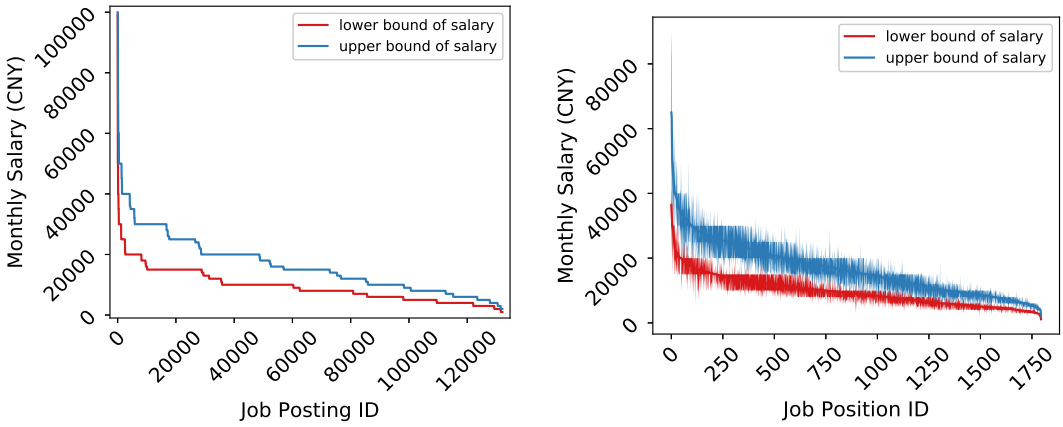
positions and companies arises rapidly in Beijing, while that of the other cities arise mildly, which means the dataset is unbalanced over locations. The salary of Beijing increases along with time, and tend to be the highest, yet the salaries in five cites are close to each other, which are accord with the facts that Beijing has the highest government-released average revenues, but the differences among the five cities are small.

In the experiments, the salary range was segmented into several discrete levels rather than the original values due to the unbalanced long tail distribution of salaries as shown in Figure 3.7, where we can observe that about 80% data records have the salary lower bound below 10K per month and 60% data records have the salary




(a) Lower Bound (b) Upper Bound

Figure 3.6. The scatter bubble chart for each location and time period, where each bubble represents a time-specific city, and the scale is proportional to the number of distinct positions.




(a) Job Posting (b) Job Position

Figure 3.7. The salary distribution in our dataset.

Table 3.3. The segmentation of salary.

| | Lower Bound (CNY) | Upper Bound (CNY) |
|---|---|---|
| Level 1 | $\leq$ 5,000 | $\leq$ 9,000 |
| Level 2 | (5,000, 8,000] | (9,000, 14,000] |
| Level 3 | (8,000, 10,000] | (14,000, 20,000] |
| Level 4 | (10,000, 15,000] | (20,000, 28,000] |
| Level 5 | $>$ 15,000 | $>$ 28,000 |

upper bound below 20K per month. Specifically, we first sorted the salary values and calculated their adjacent differences. Then, we chose four points where the adjacent differences vary dramatically as the segmentation points. After this process, the lower and upper bound of salaries were both classified into 5 levels, which are shown in Table 3.3. Note that, in the experiments, we evaluated the performance of HSBMF on the lower bound and the upper bound of salary, respectively.

### 3.4.2 Benchmark Methods

To evaluate the performance of HSBMF for salary benchmarking, we chose a number of state-of-the-art methods for comparisons. Specifically, we chose four popular MF based approaches, namely SVD, bSVD (Koren, Bell, & Volinsky, 2009), NMF (Luo et al., 2014), PMF (Mnih & Salakhutdinov, 2008), and a Collaborative Filtering (CF) based approach as baselines. Those methods are commonly used in recommender systems and achieved considerable success. We briefly introduce them in the following.

- **SVD**: Derived from Singular Vector Decompose concept in mathematics, SVD

is basically used in the early recommender systems.

- **bSVD**: bSVD refers to SVD with strategy of adding biases in this work.

- **NMF**: NMF factorizes a matrix into two non-negative lower rank latent matrices.

- **PMF**: PMF factorizes a matrix into two matrices, which adopt zero-mean spherical Gaussian priors.

- **CF**: The basic CF method recommends items based on the similarity of users or items. In this research, we utilize the company similarity for salary prediction.

In the experiments, we used Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate each approach. Specifically, the two metrics are defined as

$$RMSE = \sqrt{\frac{\sum_i^{Num}(S_i - \hat{S}_i)^2}{Num}}, \tag{3.20}$$

$$MAE = \frac{\sum_i^{Num}|S_i - \hat{S}_i|}{Num}, \tag{3.21}$$

where $S_i$ is the actual salary value, while $\hat{S}_i$ is the estimated salary value, and $Num$ is the number of test instances.

### 3.4.3 The Overall Performance

We first evaluated the overall performances of HSBMF model compared with other baselines. In the experiments, we empirically set latent dimension $K = 5$ and the

maximum iteration rounds $Max\_Inter = 100$ for all MF based methods. Furthermore, for HSBMF, we set the parameters as $\lambda_j = 0.02$, $\lambda_c = 0.02$, $\lambda_{B_j} = 0.02$, $\lambda_{B_c} = 0.02$, $\lambda_{S_j} = 1 \times 10^{-4}$, $\lambda_{S_c} = 1 \times 10^{-4}$, $\lambda_T = 1 \times 10^{-4}$, $\lambda_L = 1 \times 10^{-4}$, $\gamma = 0.005$, and $\alpha = 2$.

To validate the model performance, we also chose two kinds of sampling strategies. The first one is 5-fold cross validation with random 80%-20% splitting. The other method is only sampling 10% records in the last period as the test data and other historical data for model training. By sampling data as the second way, we can evaluate whether HSBMF model consistently outperforms other baselines for predicting salaries at last period, which is more reasonable and applicable in real-world scenarios.

Specifically, the overall RMSE and MAE results of different approaches are shown in Tables 3.4, 3.5, and 3.6 respectively. From the results, we can have the following observations. First, HSBMF consistently achieves the best performance compared with other baselines, which validates the effectiveness of integrating more constraints as side information for salary benchmarking. Second, bSVD is better than SVD and other baselines, which indicates that adding bias is an effective strategy. Indeed, the above results clearly validate the performance of HSBMF model for salary benchmarking.

### 3.4.4 Evaluation on Model Constraints

In order to evaluate the influences of different constraints, we randomly split the dataset into 5 folds for 10 times, and conducted a set of experiments by adding

different regularizer separately. Finally, we compared the average RMSE and MAE with bSVD, which is the preliminary model of HSBMF, and then calculated the paired t-test for validating the improvement significance. The experimental results are shown in Table 3.7. From the table, we can observe that all four constraints can improve the basic bSVD model. Specifically, job position and company similarity constraints can improve the model by around 2.0% to 3.0%, while time and location related constraints can only have slight improvements. It might because that we only use data records in five work locations and five different time periods, where the average salary differences are usually very small, which makes HSBMF not sensitive to $\lambda_T$ and $\lambda_L$. Nonetheless, the p-Values in all experiments are very small, demonstrating that the improvements are statistically significant for all four constraints.

### 3.4.5 Evaluation on Parameter Sensitivity

As discussed above, since HSBMF is not sensitive to $\lambda_T$ and $\lambda_L$, we fixed $\lambda_T = 2\times10^{-4}$ and $\lambda_L = 2\times10^{-4}$, and evaluated the sensitivity of $\lambda_{S_j}$ and $\lambda_{S_c}$ by changing them from 0 to $2 \times 10^{-4}$. Figure 3.8 shows the RMSE and MAE results with parameter tuning. In the figure, we can observe that the performances of RMSE and MAE consistently decrease as the increase of these two parameters. When $\lambda_{S_j}$ and $\lambda_{S_c}$ are approaching to $2 \times 10^{-4}$, the results achieve the best performances. This means the job position similarity and company similarity are effective factors for salary benchmarking.

Table 3.4. The RMSE performance of 5-fold cross validation.

| **Lower Bound** | | | | | | |
|---|---|---|---|---|---|---|
| MODEL | HSBMF | bSVD | SVD | NMF | PMF | CF |
| fold1 | **0.7763** | 0.8091 | 0.8214 | 0.8316 | 0.8287 | 0.8980 |
| fold2 | **0.7803** | 0.8135 | 0.8261 | 0.8421 | 0.8334 | 0.8860 |
| fold3 | **0.7844** | 0.8154 | 0.8312 | 0.8360 | 0.8380 | 0.8912 |
| fold4 | **0.7702** | 0.7982 | 0.8194 | 0.8264 | 0.8265 | 0.8927 |
| fold5 | **0.7799** | 0.8111 | 0.8320 | 0.8408 | 0.8277 | 0.8954 |
| **Upper Bound** | | | | | | |
| fold1 | **0.7750** | 0.8069 | 0.8309 | 0.8368 | 0.8355 | 0.9015 |
| fold2 | **0.7785** | 0.8007 | 0.8188 | 0.8323 | 0.8375 | 0.9005 |
| fold3 | **0.7759** | 0.8070 | 0.8249 | 0.8312 | 0.8363 | 0.9012 |
| fold4 | **0.7738** | 0.8022 | 0.8186 | 0.8293 | 0.8302 | 0.8930 |
| fold5 | **0.7706** | 0.8033 | 0.8213 | 0.8300 | 0.8283 | 0.8968 |

Table 3.5. The MAE performance of 5-fold cross validation.

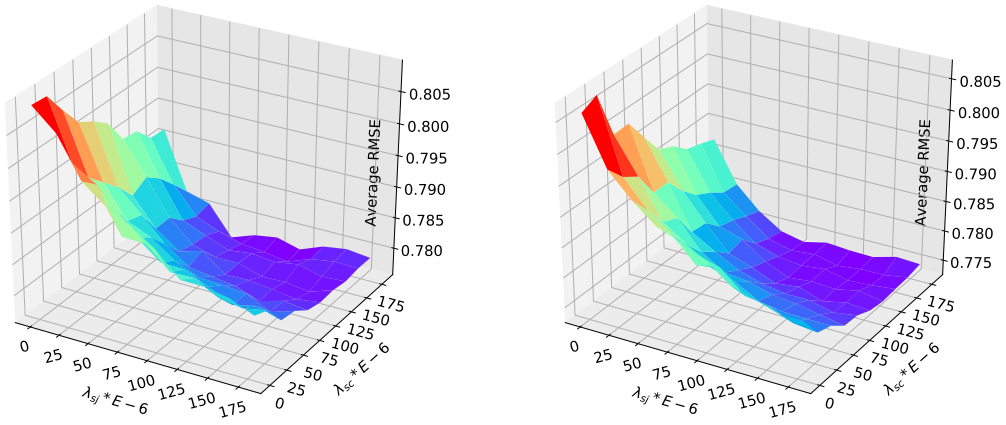
| **Lower Bound** | | | | | | |
|---|---|---|---|---|---|---|
| MODEL | HSBMF | bSVD | SVD | NMF | PMF | CF |
| fold1 | **0.5957** | 0.6165 | 0.6156 | 0.6219 | 0.6288 | 0.6880 |
| fold2 | **0.5990** | 0.6212 | 0.6153 | 0.6277 | 0.6329 | 0.6758 |
| fold3 | **0.6022** | 0.6234 | 0.6197 | 0.6242 | 0.6349 | 0.6789 |
| fold4 | **0.5900** | 0.6072 | 0.6078 | 0.6148 | 0.6269 | 0.6760 |
| fold5 | **0.5981** | 0.6184 | 0.6188 | 0.6262 | 0.6277 | 0.6804 |
| **Upper Bound** | | | | | | |
| fold1 | **0.5927** | 0.6149 | 0.6184 | 0.6232 | 0.6321 | 0.6784 |
| fold2 | **0.5914** | 0.6058 | 0.6069 | 0.6151 | 0.6312 | 0.6791 |
| fold3 | **0.5906** | 0.6109 | 0.6139 | 0.6163 | 0.6298 | 0.6795 |
| fold4 | **0.5899** | 0.6088 | 0.6082 | 0.6164 | 0.6282 | 0.6757 |
| fold5 | **0.5857** | 0.6087 | 0.6094 | 0.6158 | 0.6271 | 0.6768 |

Table 3.6. Predicting salaries of last period.

| **Lower Bound** | | | | | | |
|---|---|---|---|---|---|---|
| MODEL | HSBMF | bSVD | SVD | NMF | PMF | CF |
| RMSE | **0.7122** | 0.7259 | 0.7289 | 0.7259 | 0.7735 | 0.8299 |
| MAE | **0.5410** | 0.5418 | 0.5439 | 0.5418 | 0.5690 | 0.6396 |
| **Upper Bound** | | | | | | |
| RMSE | **0.7363** | 0.7529 | 0.7531 | 0.7529 | 0.7896 | 0.8690 |
| MAE | **0.5628** | 0.5635 | 0.5638 | 0.5635 | 0.5857 | 0.6718 |

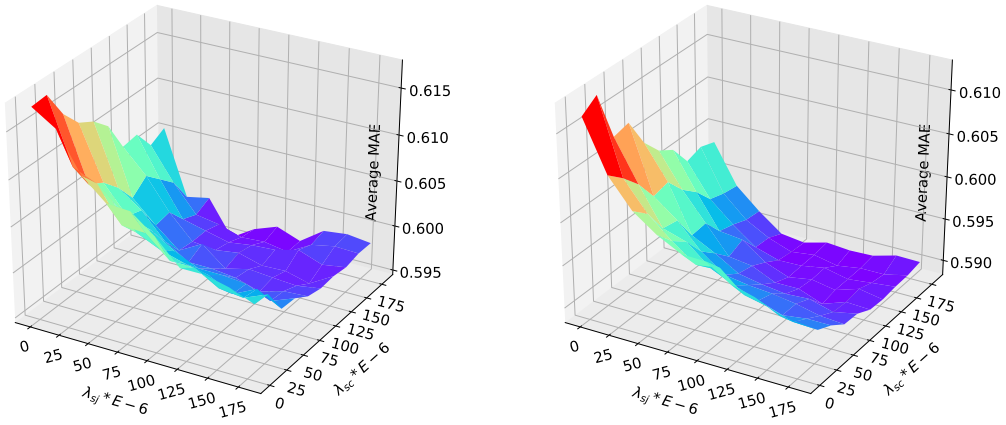
Table 3.7. Evaluation on different constrains.

| MODEL | **Lower Bound** | | | | | |
|---|---|---|---|---|---|---|
| | RMSE | Improvement | P-value | MAE | Improvement | P-value |
| bSVD | 0.8095 | - | - | 0.6174 | - | - |
| bSVD+$S_j$ | 0.7854 | 2.99% | 4.64E-43 | 0.6025 | 2.41% | 9.49E-39 |
| bSVD+$S_c$ | 0.7908 | 2.32% | 1.09E-35 | 0.6043 | 2.12% | 2.38E-31 |
| bSVD+T | 0.8064 | 0.39% | 1.59E-05 | 0.6153 | 0.34% | 1.21E-04 |
| bSVD+L | 0.8043 | 0.65% | 2.76E-12 | 0.6137 | 0.59% | 6.72E-10 |
| HSBMF | 0.7775 | 3.96% | 1.63E-38 | 0.5947 | 3.67% | 4.27E-31 |
| MODEL | **Upper Bound** | | | | | |
| | RMSE | Improvement | P-value | MAE | Improvement | P-value |
| bSVD | 0.8054 | - | - | 0.6111 | - | - |
| bSVD+$S_j$ | 0.7822 | 2.88% | 4.03E-41 | 0.5951 | 2.61% | 2.01E-38 |
| bSVD+$S_c$ | 0.7862 | 2.39% | 4.54E-41 | 0.5978 | 2.17% | 1.33E-39 |
| bSVD+T | 0.8016 | 0.47% | 1.41E-07 | 0.6083 | 0.46% | 1.89E-06 |
| bSVD+L | 0.8000 | 0.68% | 2.29E-12 | 0.6074 | 0.60% | 1.99E-10 |
| HSBMF | 0.7778 | 3.44% | 7.56E-37 | 0.5949 | 2.66% | 5.27E-30 |




(a) RMSE for Lower Bound

(b) RMSE for Upper Bound




(c) MAE for Lower Bound

(d) MAE for Upper Bound

Figure 3.8. The performance of HSBMF with different parameter settings of $\lambda_{S_j}$ and $\lambda_{S_c}$.

## 3.5 Related Work

We organized our related work into two aspects. We first introduce the works related to salary benchmarking problems, then we summarize the literature related to our methodologies.

### 3.5.1 Job Salary Benchmarking

The tasks of salary benchmarking are quite different for high-level managers such as CXOs (*i.e.*, CEO, CFO, CTO *etc.*) from the middle and low level employees. The discrepancy generally comes from the principals of pricing, and the components of salaries. The salaries of CXOs are performance-based, and cash salaries are a small amount of the total incomes, the largest part of them are including bonus, options, equity or non-equity based incentives and others (Lazar, 2004). Companies' performances and "peer group" effect are two main directions used to explain CEO pays in current literature (Frydman & Jenter, 2010; Gong & Li, 2013; Brick, Palmon, & Wald, 2006; Blankmeyer, LeSage, Stutzman, Knox, & Pace, 2011), nonetheless, seldom works in this field aims on estimating the salary range, their research contributions focused on finding the salary determinants or the possible relationships between the salaries and managerial manipulations (Peng & Röell, 2014; Peng & Roell, 2008).

Our research is targeted on pricing of middle and low level employees. Different from CXOs, skill requirements, companies' compensation strategy, work location are key factors determine the salary range of a position. Previous studies intended to understand what kinds of factors will influence the salary level from the individual

perspective, such as age, gender, the timing of motherhood *etc* (Lazar, 2004; Jerrim, 2015; Hamlen & Hamlen, 2016; Correll, Benard, & Paik, 2007). There are a large portion of researches emphasizing on the pay equity (Chang & Hahn, 2006; Berkowitz, Fraser, Treasure, & Cochran, 1987; Scarpello & Jones, 1996; Terpstra & Honoree, 2003). Scholars also investigated the effects that compensation was shaped by peer groups (Blankmeyer et al., 2011; Faulkender & Yang, 2010). (Ferris, Witt, & Hochwarter, 2001) found high social skill and high general mental ability have strong explanation in individuals' job performance and salary levels. Besides, researchers concerned about how to design the compensation structure to boost the performances of firms and their employees (Bergmann & Scarpello, 2002). Recently years, data mining techniques have been applied to salary prediction in variety of scenarios. (Khongchai & Songmuang, 2016b, 2016a) estimated the students' income by their demographic features, and the results can boosting their studying motivations in return. (Lin et al., 2017) proposed a graphical model for company profiling, the model has the abilities to estimate salaries. However, it took the employees' negative and positive comments into considerations rather than the skill requirements and responsibilities, where the application scenarios are different with ours.

### 3.5.2 MF Based Models

MF techniques is widely used in recommender systems, besides that, they also applied to a broad related areas, such as social network analyses (Xiao, Liu, Liu, & Xiong, 2017; L. Zhang, Xiao, Liu, Tao, & Deng, 2015), image tagging (Zhou, Cheung, Qiu, & Xue, 2011), document clustering (W. Xu, Liu, & Gong, 2003) and so

on. The early MF model is based on Singular Vector Decomposition(SVD), which is a well-established technique for identifying latent semantic factors (Adomavicius & Tuzhilin, 2005). The early SVD-based recommendation systems are prone to distort the data and lead to the over-fitting problem, since they applied imputation techniques, which fill the missing values and make the rating matrix dense (Kim & Yum, 2005). As a result, researchers suggest only to model the ratings observed, and add adequate regularizers to avoid over-fitting problems (Paterek, 2007). More recently, researchers proposed various improvements of MF based recommendations. The most representative works include biased SVD (bSVD), SVD++, NMF, and PMF. Specifically, bSVD tries to use bias terms for capturing the latent information associated with users or items (Paterek, 2007; Koren et al., 2009). SVD++ interprets the data with the effect of "implicit" information of users or items (Koren, 2008). In addition, NMF also belongs to MF families. However, different from SVD, NMF constrains latent factors to be non-negative (Lee & Seung, 1999, 2000). Finally, PMF places zero-mean spherical Gaussian priors on user and item feature vectors (Mnih & Salakhutdinov, 2008), which usually passes the estimated values through a logistic function to bound the range of predictions. In order to solve the recommendation systems with additional information, researchers proposed context-aware MF models (Adomavicius & Tuzhilin, 2015), classifying the approaches into three categories: pre-filtering, post-filtering, and contextual modeling. Item-splitting (Baltrunas & Ricci, 2009) is one example of pre-filtering methods. It splits the ratings and corresponding items into multiple virtual ratings and items based on items' subcategories. The post-filtering strategy applies filtering or weighting after the traditional

approaches. (Panniello, Tuzhilin, Gorgoglione, Palmisano, & Pedone, 2009) compared effectiveness and performances of pre-filtering and post-filtering. It states that the better choice of pre-filtering or post-filtering depending on the specific methods. The last category is contextual approach, which uses contextual information directly into a recommender model (Rendle, Gantner, Freudenthaler, & Schmidt-Thieme, 2011; Panniello, Tuzhilin, & Gorgoglione, 2014; H. Zhu et al., 2015; Bao, Cao, Chen, Tian, & Xiong, 2012; Ge, Liu, Xiong, & Chen, 2011). One well-known method is tensor factorization (TF) proposed by (Karatzoglou, Amatriain, Baltrunas, & Oliver, 2010). It factorizes a three-dimension tensor into three feature matrices and one core matrix. However, this method has two drawbacks: one is its rapid growth of parameters and computational complexity; the other is its limited application to categorical contextual variables. In the paper (Baltrunas, Ludwig, & Ricci, 2011), the authors demonstrated that MF-based models can have comparable, and even better performances than TF-based models, especially when data sets are small. Therefore, in this work, HSBMF is MF-based approach that integrates holistic constraints for fine-grained salary benchmarking.

## 3.6 Conclusions

In this chapter, we studied the problem of salary benchmarking through the analyses of massive online recruitment data. Specifically, we formalized the problem as a matrix completion task, and then developed a Matrix Factorization (MF) based model named HSBMF for predicting the missing salary information in the expanded salary matrix. A unique perspective of HSBMF is that it can provide a holistic and dynamic

view of salary benchmarking by integrating multiple confounding factors, such as company similarity, job similarity, and spatial-temporal similarity. Finally, extensive experiments were conducted on large-scale real-world data, and the results validated the effectiveness of HSBMF for timely salary benchmarking requirement.

# CHAPTER 4

# FINE-GRAINED JOB SALARY BENCHMARKING WITH NONPARAMETRIC DIRICHLET-PROCESS-BASED LATENT FACTOR MODEL

## 4.1 Introduction

Job salary benchmarking (JSB) refers to the process by which organizations acquire and analyze labor market data to determine appropriate compensation for their actual and prospective employees (Blankmeyer et al., 2011). The importance of this job has been discussed in chapter 3.

Many human resource handbooks summarize general guidance on JSB. For example, (Armstrong, 2006; Edwards, Scott, & Raju, 2003) emphasized the importance of jointly considering internal salary tendencies and external job market rates to address the JSB problem. However, they usually offer solutions based on limited data sources (e.g., questionnaires and survey data) and simple techniques (e.g., job category matching and simple statistical models). In practice, it is highly necessary to have a fine-grained JSB solution to effectively take internal and external factors into consideration in a unified way. LinkedIn disclosed that the current salary services of the company (Kenthapadi, Chudhary, & Ambler, 2017; Kenthapadi, Ambler, Zhang, & Agarwal, 2017) rely on the salary statistics (e.g., 1st quartile, mean, 3rd quartile, etc.) generated through a Bayesian normal distribution inference. However, such

methods cannot address the issue of having bias when handling sparse data. Considering that data sparseness is a common issue in salary data, making predictions with limited data sources is a key challenge in the JSB problem. To address these issues above, we developed a Matrix Factorization (MF) based method in Chapter 3. Although the sparseness issue can be handled this way, the model may still fail in handling salary benchmarking when facing completely new positions or companies without sufficient historical records, leading to cold start issues. Moreover, classic MF methods result in low interpretability and hence weaken the practical value in supporting decision-making for talent management. Explainable insights into the prediction results are appreciated for providing C&B managers information on detailed and quantified salary-job patterns to support their final salary decisions.

To address the above issues, we handle the JSB problem from a fine-grained perspective using data-driven techniques while considering the model interpretability. We design a nonparametric Dirichlet-process-based latent factor model for JSB named the NDP-JSB, which jointly considers internal salary tendencies and the external job market rate through an enhanced MF structure. Specifically, a company representation module is utilized to group companies into different clusters based on location-specific information, and a position representation module is implemented to learn the corresponding job latent parameters based on the job description data. Our model can intelligently refer to similar companies or positions for salary prediction even if the observable data are deficient. Additionally, we can extract features from the job representation and company grouping results for further analysis and then offer certain interpretations for salary prediction.

In summary, this work contributes to the literature in five ways. First, we provide a fine-grained solution to the JSB problem, helping employers make smart salary decisions by analysing companys salary tendency and the job market rate together. Second, we greatly alleviate the data deficiency problem in JSB tasks by taking advantage of the deeply mined patterns among companies and job positions. Third, our method can effectively make predictions for new types of companies when historical salary observations are lacking. Fourth, our model has the strength of being able to offer interpretable results to enhance its value in practice, such as showing the share of a given skill set for a specific job and identifying similar companies for comparison. Finally, we conduct extensive experiments on a large-scale real-world recruitment dataset. By comparing our model with state-of-the-art baselines, the results not only verify the effectiveness of the NDP-JSB model in addressing the JSB problem but also demonstrate its strength in revealing patterns of job categories and companies.

## 4.2 Model Overview

In this section, we discuss the overall structure of the method we propose, the final objective function, model inference, and the updating formulas for optimization.

### 4.2.1 The Model

To address the JSB problem, we construct a Bayesian graphical probabilistic model, which including three modules, (1) the *Position Representation Module*, (2) the *Company Representation Module*, and (3) the *Salary Prediction Module*. First, We utilize a matrix factorization structure to capture the interactions between the company's internal salary policy and the external market pricing in the *Salary Prediction Mod-*

*ule*. In this module, $t_i$ denotes the job-related latent factors, and $c_j$ denotes the company-related latent factors, so the predicted salary $\hat{s}_{ij}$ can be computed as the cross product of $t_i$ and $c_j$. That is,

$$\hat{s}_{ij} = t_i^T c_j. \tag{4.1}$$

Second, we use the *Position Representation Module* and the *Company Representation Module* to learn $t_i$ and $c_j$, respectively. Specifically, In the *Position Representation Module*, we learn the topic distribution $\varphi_i$ from the job descriptions through the Latent Dirichlet Allocation (LDA) structure. $t_i$ is obtained from the normal distribution with the mean $\varphi_i$. Meanwhile, in the *Company Representation Module*, we segment those companies into several clusters based on their features $X$ by applying a Non-parametric Dirichlet Process (NDP). And, the companies in the same cluster share the same latent factors. Letting $z_j$ be the cluster index of each company $j$, we can rewrite the expected salary $\hat{s}_{ij}$ in Eq. (4.1) as:

$$\hat{s}_{ij} = t_i^T c_{z_j}. \tag{4.2}$$

In these ways, our model not only considers multiple sources of job- and company-related information during the learning process, but also are able to ensure that similar jobs and companies will have similar latent factors.

Although different modules bare different functions, they are connected as a joint Bayesian probabilistic structure. The parameters in each module are inferred jointly; thus, the job-related factors $t_i$ are affected not only by job descriptions but also the historical salaries; so do the company-related factors $c_{z_j}$. In the following, we will

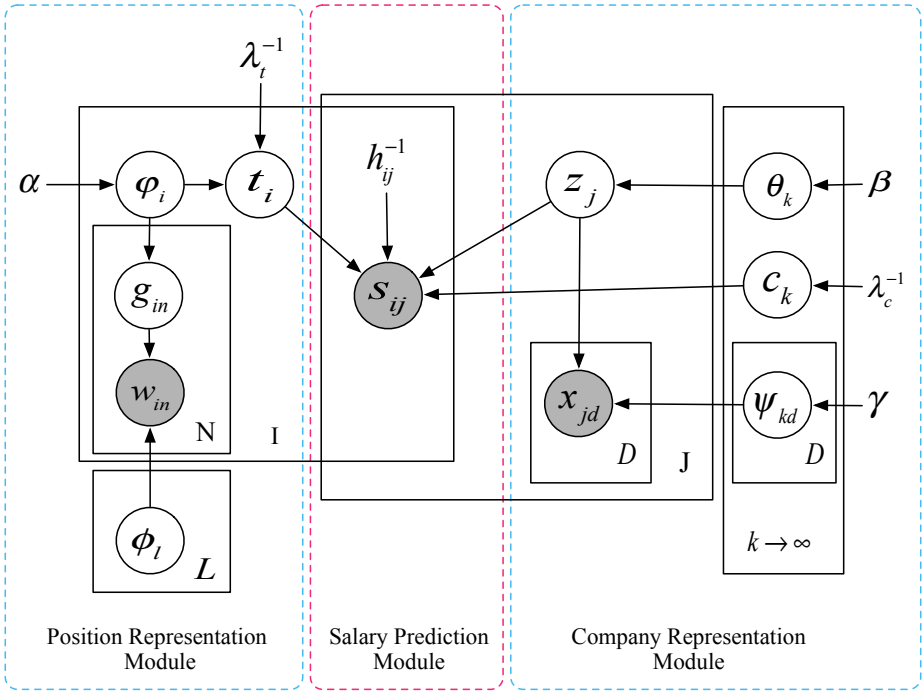



Figure 4.1. The graphical representation of the model.

discuss the three modules in detail.

**Module 1: Position Representation**

In the position representation module, we use an LDA structure to process the job position data (i.e., the job descriptions). LDA models are heavily used in text information retrieval, latent semantic analysis, and text clustering. LDA regards generating an article as the generation of those words in the article, which includes three steps: first, for each article $i$, we generate a topic proportion $\varphi_i$ from a Dirichlet process with the prior parameter $\alpha$. Second, we assign every word $w_{in}$ in the article with a specific topic $g_{in}$; the topic $g_{in}$ is selected based on topic distribution $\varphi_i$. Last, given topic-word distribution parameters $\phi_{g_{in}}$, we generate each word $w_{in}$ from the multinomial distribution with the parameters $\phi_{g_{in}}$. In this process, words $w_{in}$ are known

variables, the topic proportion $\varphi$ for every article and the topic-word distribution $\phi$ are the latent factors we should learn from the model.

**Module 2: Company Representation**

In the company representation module, we consider both company's basic features and the company's historical salary observations. The company's historical salary observations can bear the compensation tendency information of that company. For example, compared to small firms, large companies or public corporations usually have more budgets hence can offer higher salaries to seize the top talents in their interested fields, while tight-budget start-ups may only offer the salaries bordering on the average. A way to investigate the discrepancy among companies is to classify them into different groups. A reasonable principle is that company-related factors within a group share the same parameters, while parameters in different groups should fit the similarity relationship. We utilize the NDP to handle the segmentation job. We choose the stick-breaking view to construct an NDP (Ishwaran & James, 2001). We first sample $\theta_k, k = 1, 2, ..., \infty$ from a Beta distribution $B(1, \beta)$. Based on $\theta_k$, we obtain a set of parameters $\pi_k, k = 1, 2, ..., \infty$ through the calculation $\pi_k = \theta_k \prod_{b=1}^{k-1}(1-\theta_b)$. After that, we draw the group index $z$ from the multinomial distribution where the parameters are formed by $\pi_k$. That is, $z_j \sim Multi(1; \pi_1, \pi_2, ..., \pi_\infty)$. Since the dimension of $\pi$ is infinite, the possible group numbers, theoretically, can also be infinite. Meanwhile, we draw the company latent factors $c_k, k = 1, 2, ..., \infty$ from a normal distribution $N(0, \lambda_c^{-1})$ for each possible group. In parallel with $c_k$, we draw another set of parameters $\psi_{kd}, k = 1, 2, ..., \infty, d = 1, 2, ..., D$, which are used as base

parameters of multinomial distributions to generate features $X$ of each company. That is, company features $x_{jd} \sim Multi(1; \psi_{z_j,d})$. Based on the above procedures, we ensure that companies in the same group share the same parameters, and similar company groups tend to have similar latent factors.

**Module 3: Salary Prediction**

In the salary prediction module, we follow a matrix factorization formulation. For a (position $i$, company $j$) combination, since we know that the group index of the company is $z_j$, we retrieve corresponding factors $t_i$ and $c_{z_j}$, respectively. We first compute the matrix product of $t_i$ and $c_{z_j}$, then draw the salary values from the normal distribution, where the mean value is $t_i^T c_{z_j}$, the variance is $h_{ij}^{-1}$.

### 4.2.2 Objective Function

Now, we can specify the objective function based on the proposed framework. In our model, $w_{in}$, $s_{ij}$, and $x_{jd}$ are visible variables; $\alpha$, $\beta$, $\lambda_t$, $\lambda_c$, $h_{ij}$, and $\gamma$ are hyper parameters that need to be determined before training. Other variables $\Omega = (\varphi, G, \Phi, T, Z, \Theta, C, \Psi)$ are latent variables need to be trained. We set the maximum group number of companies equals $K$, the number of topics equals $L$, and the dimension of each company feature equals $M$. To get the optimal values of those variables, we maximize the Maximum Posterior Estimation (MPE) of the model. Thus, our job salary benchmarking problem can be mathematically formalized as follows:

$$
\begin{aligned}
\text{max: } \mathcal{L} &= \log\left(\prod_{i,j,n,d} P(s_{ij}, x_{jd}, w_{in}, \Omega)\right) \\
&= \sum_{i}^{I}\sum_{n}^{N} \log\left(P(w_{in}, g_{in}|\varphi_i, \phi_l)\right) + \sum_{i}^{I} \log\left(P(\varphi_i|\alpha)\right) + \sum_{i}^{I} \log\left(P(t_i|\varphi_i, \lambda_t^{-1})\right) \\
&+ \sum_{i,j,d} \log\left(P(s_{ij}, x_{jd}, C, Z, \Psi, \Theta|T, \lambda_c^{-1}, \beta, \gamma, h_{ij}^{-1})\right), \\
\text{s.t. } & \sum_{l}^{L} \varphi_{il} = 1 \quad \forall i, \quad \sum_{n}^{N} \phi_{ln} = 1 \quad \forall l, \quad \sum_{m}^{M} \psi_{kdm} = 1 \quad \forall k, d, \\
& \varphi_{il} > 0, \quad \phi_{ln} > 0, \quad g_{in} > 0, \quad 0 < \theta_k < 1, \quad z_j > 0, \quad \psi_{kdm} > 0.
\end{aligned}
\tag{4.3}
$$

The complete Bayesian generation process of our model is summarized in Algorithm 2.

### 4.2.3 Inference

To solve the objective function above, we use the variational inference and projection gradient descent method jointly. Since the parameters $\varphi$, $G$, and $\Phi$ are disconnected with parameters $Z$, $\Theta$, $C$, $\Psi$ in the probabilistic graph, we can solve them separately. We set $\alpha = 1$ and omit some constants. We denote the last term in Eq. (4.3) by $\mathcal{L}_0$, which is irrelevant to $\varphi$, $G$, and $\Phi$. Thus, the objective function can be rewritten as:

$$
\text{max: } \mathcal{L} \propto -\frac{\lambda_t}{2}\sum_{i}^{I}(t_i - \varphi_i)^T(t_i - \varphi_i) + \sum_{i}^{I}\sum_{n}^{N} \log(\sum_{l}^{L} \varphi_{il}\phi_{lw_{in}}) + \mathcal{L}_0. \tag{4.4}
$$

The parameters $\varphi$, $G$, and $\Phi$ can be solved in a similar way as suggested in (Wang & Blei, 2011). We extract the terms that contain $\varphi$, $G$, and $\Phi$ as below, and define

$q(g_{in} = l) = \widetilde{g}_{inl}$. Applying Jensen's inequality, we have

$$\mathcal{L}(\varphi_i, g, \phi) = -\frac{\lambda_t}{2}\sum_i^I (t_i - \varphi_i)^T(t_i - \varphi_i) + \sum_i^I \sum_n^N \log(\sum_l^L \varphi_{il}\phi_{lw_{in}})$$
$$\geq -\frac{\lambda_t}{2}\sum_i^I (t_i - \varphi_i)^T(t_i - \varphi_i) + \sum_i^I \sum_n^N \sum_l^L \widetilde{g}_{inl}(\log(\varphi_{il}\phi_{lw_{in}}) - \log \widetilde{g}_{inl}) = \bar{\mathcal{L}}(\varphi_i, \widetilde{g}, \phi), \tag{4.5}$$

where $\bar{\mathcal{L}}(\varphi_i, \widetilde{g}, \phi)$ is the lower bound of $\mathcal{L}(\varphi_i, g, \phi)$. We compute the partial derivatives of $\bar{\mathcal{L}}$ with respect to $\widetilde{g}$, $\phi$, and then set derivatives to zeros. Then, we get the updating formulas for these two parameters when applying the coordinate ascend method.

$$\widetilde{g}_{inl} \propto \varphi_{il}\phi_{lw_{in}}, \tag{4.6}$$

and

$$\phi_{lw} \propto \sum_i^I \sum_n^N \widetilde{g}_{inl} 1[w_{in} = w]. \tag{4.7}$$

Different from $\widetilde{g}$ and $\phi$, the derivative function with respect to $\varphi$ is quadratic, so we solve it by applying the projection gradient descent method (Duchi, Shalev-Shwartz, Singer, & Chandra, 2008).

Next, we apply the variational inference to compute the Evidence Lower Bound (ELBO) of $\mathcal{L}_0$ and solve the remaining parameters. We define

$$q(Z, \Theta, C, \Psi) = \prod_j^J q(z_j) \prod_k^K q(\theta_k) \prod_k^K q(c_k) \prod_k^K \prod_d^D q(\psi_{kd}), \tag{4.8}$$

where $q(z_j)$ represents the multinomial distribution with parameters $q(z_j = k) = \widetilde{z}_{jk}$; $q(\theta_k)$ is the Beta distribution with parameters $(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2})$; $q(c_k)$ is the normal distributions with parameters $(\widetilde{\mu}_{c_k}, \widetilde{\lambda}_{c_k}^{-1})$; $q(\psi_{kd})$ is the Dirichlet distribution with

parameters $\widetilde{\psi}_{kd}$. The ELBO of $\mathcal{L}_0$ can be computed as follows:

$$\mathcal{L}_0 \geq \sum_{i,j} E_q[\log(P(s_{ij}|t_i, z_j, h_{ij}^{-1}, C))] + \sum_{j}^{J} E_q[\log(P(z_j|\Theta))] + \sum_{k}^{K} E_q[\log(P(\theta_k|\beta))]$$
$$+ \sum_{k}^{K} E_q[\log(P(c_k|\lambda_c^{-1}))] + \sum_{k}^{K}\sum_{d}^{D} E_q[\log(P(\psi_{kd}|\gamma))] + \sum_{j}^{J}\sum_{d}^{D} E_q[\log(P(x_{jd}|z_j, \psi_{*,d}))]$$
$$- E_q[\log(q(Z, C, \Theta, \Psi))]. \tag{4.9}$$

Now we need to compute all terms in the Eq. (4.9). Here we only show the results, while the mathematical details are discussed in Appendix A.

$$\begin{aligned} E_{q(Z,C)}\left[\log(P(s_{ij}|t_i, z_j, h_{ij}^{-1}, C))\right] &= E_{q(Z,C)}\left[\log\left(\prod_{k}^{K} P(s_{ij}|t_i, c_k, h_{ij})^{1[z_j=k]}\right)\right] \\ &= \sum_{k}^{K}\left\{E_{q(z_j)}[1[z_j = k]] \cdot E_{q(c_k)}[\log(P(s_{ij}|t_i, c_k, h_{ij}))]\right\} \\ &= \sum_{k}^{K}\left\{\widetilde{z}_{jk} \cdot E_{q(c_k)}[\log(P(s_{ij}|t_i, c_k, h_{ij}))]\right\} \\ &= \sum_{k}^{K}\left(\widetilde{z}_{jk}\mathcal{L}_1\right), \end{aligned} \tag{4.10}$$

where

$$\mathcal{L}_1 = -\frac{h_{ij}}{2}\left(s_{ij}^2 - 2s_{ij}t_i^T\widetilde{\mu}_{c_k} + t_i^T\rho_k t_i\right), \tag{4.11}$$

and $\rho_k = \widetilde{\mu}_{c_k}\widetilde{\mu}_{c_k}^T + \mathbf{\Lambda}(\widetilde{\lambda}_{c_k}^{-1})$. $\mathbf{\Lambda}$ is a function transforming a vector into a matrix that the diagonal elements equal to the vector values, and leaving the remaining elements to be zeros.

$$E_q[\log(P(z_j|\Theta))] = \sum_{k}^{K} q(z_j > k)E_q[\log(1 - \theta_k)] + q(z_j = k)E_q[\log\theta_k], \tag{4.12}$$

where

$$q(z_j = k) = \widetilde{z}_{jk},$$

$$q(z_j > k) = \sum_{g=k+1}^{K} \widetilde{z}_{jg},$$

$$E_q[\log \theta_k] = \mathbf{\Psi}(\widetilde{\theta}_{k,1}) - \mathbf{\Psi}(\widetilde{\theta}_{k,1} + \widetilde{\theta}_{k,2}),$$

$$E_q[\log(1 - \theta_k)] = \mathbf{\Psi}(\widetilde{\theta}_{k,2}) - \mathbf{\Psi}(\widetilde{\theta}_{k,1} + \widetilde{\theta}_{k,2}).$$

In the equations above, $\mathbf{\Psi}(\cdot)$ is the Digamma function. The detailed proof can refer to (Blei, Jordan, et al., 2006).

$$E_q[\log(P(\theta_k|\beta))] = \log(\beta) + (\beta - 1)E_q[(1 - \theta_k)]. \tag{4.13}$$

$$E_q[\log(P(c_k|\lambda_c^{-1}))] = \frac{L}{2}\log(\frac{\lambda_c}{2\pi}) - \frac{\lambda_c}{2}(\widetilde{\mu}_{c_k}^T\widetilde{\mu}_{c_k} + \sum_{l}^{L}\widetilde{\lambda}_{c_{kl}}^{-1}). \tag{4.14}$$

$$E_q[\log(P(x_{jd}|z_j, \psi_{*,d}))] = \sum_{k}^{K}\left(\widetilde{z}_{jk}E_q[\log \psi_{kd,x_{jd}}]\right). \tag{4.15}$$

$$E_q[\log(P(\psi_{kd}|\gamma))] = \sum_{m}^{M}(\gamma - 1)E_q[\psi_{kdm}] - \log \mathbf{B}(\gamma), \tag{4.16}$$

where the $\mathbf{B}(\cdot)$ is Multivariate Beta function, and $E_q[\psi_{kdm}] = \mathbf{\Psi}(\widetilde{\psi}_{kdm}) - \mathbf{\Psi}(\sum_m \widetilde{\psi}_{kdm})$.

$$E_q[\log(q(c_k|\widetilde{\mu}_{c_k}, \widetilde{\lambda}_{c_k}))] = \frac{1}{2}\sum_{l}^{L}\log(\frac{\widetilde{\lambda}_{c_{kl}}}{2\pi}) - \frac{L}{2}. \tag{4.17}$$

$$E_q[\log(q(z_j|\widetilde{z}_{jk}))] = \sum_{k}^{K}\widetilde{z}_{jk}\log(\widetilde{z}_{jk}). \tag{4.18}$$

$$E_q[\log(q(\theta_k|\widetilde{\theta}_{k,1},\widetilde{\theta}_{k,2}))] = -\log \mathbf{B}(\widetilde{\theta}_{k,1},\widetilde{\theta}_{k,2}) + (\widetilde{\theta}_{k,1}-1)E_q[\log\theta_k] + (\widetilde{\theta}_{k,2}-1)E_q[\log(1-\theta_k)]. \tag{4.19}$$

$$E_q[\log(q(\psi_{kd}|\widetilde{\psi}_{kd}))] = \sum_{m}^{M}(\widetilde{\psi}_{kdm}-1)E_q[\log\psi_{kdm}] - \log \mathbf{B}(\widetilde{\psi}_{kd}). \tag{4.20}$$

### 4.2.4 Updating Formulas

We substitute all the terms in Eq. (4.3) based on equations described in Section 4.2.3. After solving the derivatives in the optimization problem, we will get the updating formulas for all corresponding terms. Note that we only show the result for each updating formula while the proofs are demonstrated in Appendix B.

1. updating $q(\theta_k)$

$$\begin{aligned} \widetilde{\theta}_{k,1} &= 1 + \sum_{j}^{J}\widetilde{z}_{jk}, \\ \widetilde{\theta}_{k,2} &= \beta + \sum_{j}^{J}\sum_{g=k+1}^{K}\widetilde{z}_{jg}. \end{aligned} \tag{4.21}$$

2. updating $q(c_k)$

$$\begin{aligned} \widetilde{\mu}_{c_k} &= (T\mathbf{\Lambda}(H\widetilde{z}_k)T^T + \lambda_c I_l)^{-1}(T(H\odot S)\widetilde{z}_k), \\ \widetilde{\lambda}_{c_k} &= T\odot TH\widetilde{z}_k + \lambda_c I_l, \end{aligned} \tag{4.22}$$

where $\odot$ denotes matrix Hadamard product.

3. updating $q(z_j)$

$$\widetilde{z}_{jk} \propto \exp\left\{ E_q[\log(\theta_k)] + \sum_{g}^{k-1} E_q[\log(1-\theta_g)] + \sum_{i}^{I} \mathcal{L}_1 + \sum_{d}^{D} E_q[\log \psi_{kd,x_{jd}}] \right\}. \tag{4.23}$$

4. updating $q(\psi)$

$$\widetilde{\psi}_{kdm} = \sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma. \tag{4.24}$$

5. updating $t_i$

$$t_i = (\rho Z^T h_i + \lambda_t)^{-1} (\widetilde{\mu}_c Z^T (h_i \odot s_i) + \lambda_t \varphi_i). \tag{4.25}$$

Finally, the overall optimization process is demonstrated in the Algorithm 3.

**Algorithm 2** The generative process of the NDP-JSB.

**for** Each job $i$ **do**

  Draw topic proportion $\varphi_i \sim Dir(\alpha)$

  Draw job latent offset $\epsilon_i \sim N(0, \lambda_t^{-1} I_l)$

  Job latent vector $t_i = \varphi_i + \epsilon_i$

  **for** Each word $w_{in}$ **do**

    Draw topic assignment $g_{in} \sim Multi(1; \varphi_i)$

    Draw word $w_{in} \sim Multi(1; \phi_{g_{in}})$

  **end for**

**end for**

Draw $\theta_k \sim Beta(1, \beta)$, $k = 1, 2, ..., \infty$.

Group proportion $\pi_k = \theta_k \prod_{b=1}^{k-1}(1 - \theta_b)$, $k = 1, 2, ..., \infty$

Draw company factors for every group $c_k \sim N(0, \lambda_c^{-1} I_l)$, $k = 1, 2, ..., \infty$

Draw company feature distribution parameters for every group $\psi_{kd} \sim Dir(\gamma)$, $k = 1, 2, ..., \infty, d = 1, 2, ..., D$

**for** Each company $j$ **do**

  Draw group indicator

  $z_j \sim Multi(1; \pi_1, \pi_2, ..., \pi_\infty)$, $j = 1, 2, ..., J$

  **for** Each company feature $d$ **do**

    $x_{jd} \sim Multi(1; \psi_{z_j, d})$, $d = 1, 2, ..., D$

  **end for**

**end for**

**for** Each $(i, j)$ combination **do**

  Salary $s_{ij} \sim N(t_i^T c_{z_j}, h_{ij}^{-1})$

**end for**

**Algorithm 3** The optimization process of the NPD-JSB.

**Input:**

$W$, $S$, $H$, $X$, $\alpha$, $\beta$, $\gamma$, $\lambda_c$, $\lambda_t$

**Output:** $T$, $\widetilde{\mu}_c$, $\widetilde{\lambda}_c$, $\widetilde{Z}$, $\varphi$, $\widetilde{G}$, $\phi$, $\widetilde{\Theta}$, $\widetilde{\psi}$

Initialize $T$, $\widetilde{\mu}_c$, $\widetilde{\lambda}_c$, $\widetilde{Z}$, $\widetilde{\Theta}$, $\widetilde{\psi}$ with random values;

Initialize $\varphi$, $\widetilde{G}$, $\phi$ with pre-trained vanilla LDA values to save computation time;

and normalize $\varphi$, $\widetilde{G}$, $\phi$, $\widetilde{Z}$, $\widetilde{\psi}$ to ensure the sum of last dimension equals 1.

**while** Not Converge **do**

  Update $\widetilde{\Theta}$ according to Eq. (4.21)

  Update $\widetilde{\psi}$ according to Eq. (4.24) and normalize $\widetilde{\psi}$

  Update $\widetilde{\mu}_c$, $\widetilde{\lambda}_c$ according to Eq. (4.22)

  Update $T$ according to Eq. (4.25)

  Update $Z$ according to Eq. (4.23) and normalize $Z$

  **while** NOT Converge **do**

    Update $\varphi$ according to projection gradient descent method

  **end while**

  Update $\widetilde{G}$ according to Eq. (4.6), and normalize the $\widetilde{G}$

  Update $\phi$ according to Eq. (4.7), and normalize the $\phi$

**end while**

**return** $T$, $\widetilde{\mu}_c$, $\widetilde{\lambda}_c$, $\widetilde{Z}$, $\varphi$, $\widetilde{G}$, $\phi$, $\widetilde{\Theta}$, $\widetilde{\psi}$

## 4.3 Data and Experiments

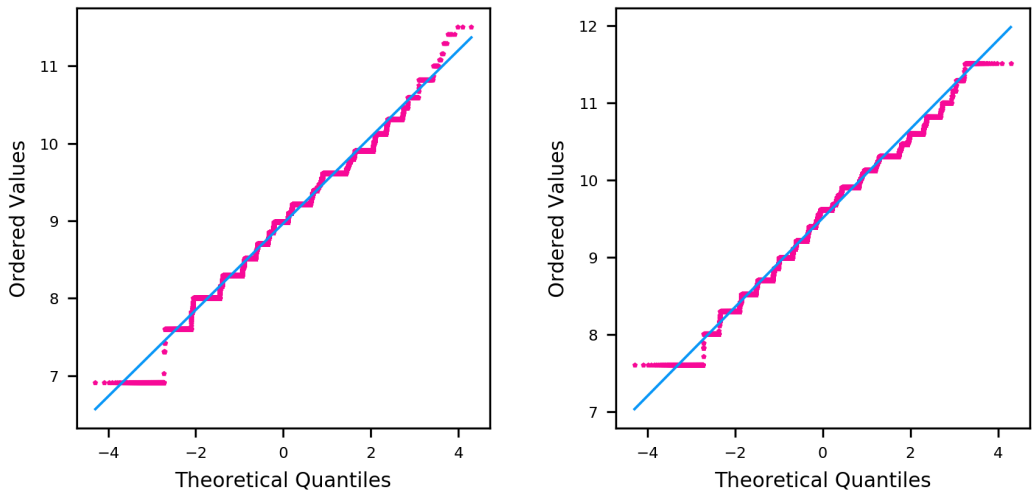
This section discusses data processing, experimental settings, and main results.

### 4.3.1 Data

The data of this work are based on collected job advertisements from a popular online recruiting platform. The dataset contains job postings released by high-tech companies located in five major cities in China, including Beijing, Shanghai, Shenzhen, Guangzhou, and Hangzhou. The time period of our dataset ranges from July 2013 to October 2015. To avoid noise information, we removed those companies that published job positions less than 20 times, and the job titles that are rarely offered in the market (e.g., appear less than five times in our data). We grouped and normalized the positions with similar job titles manually. The processes left us $132,061$ job postings which belong to $1,795$ job titles from $1,788$ companies in the data. Since the real salaries are distributed in a long-tail manner, we used the logarithmic salary in our model to ensure the values closely follow a normal distribution (see Figure 4.2). As can be seen, the scattered points of ordered salary values against the theoretical quantiles are almost in a straight line, indicating a normal distribution is held. Similar processes can also be found in (Kenthapadi, Ambler, et al., 2017).

### 4.3.2 Baselines, Settings, and Evaluation Metrics

For validation, since the JSB problem is transformed as a matrix completion task, we compared our method (NDP-JSB) with five powerful matrix factorization (MF) methods in terms of prediction accuracy. They are (1) Holistic Salary Benchmarking Matrix Factorization (HSBMF) (Meng et al., 2018), (2) Singular Vector Decomposi-




(a) Lower Bound (b) Upper Bound

Figure 4.2. The probability plots of the logarithmic salaries.

tion (SVD) (Koren et al., 2009), (3) Collaborative Topic Model (CTR) (Wang & Blei, 2011), (4) Probabilistic Matrix Factorization (PMF) (Mnih & Salakhutdinov, 2008), and (5) Nonnegative Matrix Factorization (NMF) (Luo et al., 2014). These methods are largely used in recommendation systems to address sparse prediction tasks.

In the experiments, we used the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) to evaluate each approach.

Table 4.1. The RMSE performance for the 5-fold cross validation.

| **Lower Bound** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **RMSE** | | | | | **MAE** | | | | |
| | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 |
| NDP-JSB | **0.5755** | **0.5886** | **0.5848** | **0.5865** | **0.5869** | **0.4280** | **0.4316** | **0.4281** | **0.4340** | **0.4267** |
| HSBMF | 0.5851 | 0.5969 | 0.5962 | 0.5979 | 0.5987 | 0.4368 | 0.4405 | 0.4383 | 0.4440 | 0.4383 |
| SVD | 0.5971 | 0.6078 | 0.6030 | 0.6089 | 0.6108 | 0.4413 | 0.4467 | 0.4435 | 0.4506 | 0.4438 |
| CTR | 0.6475 | 0.6660 | 0.6572 | 0.6506 | 0.6558 | 0.4808 | 0.4912 | 0.4790 | 0.4808 | 0.4812 |
| PMF | 0.6021 | 0.6201 | 0.6156 | 0.6083 | 0.6138 | 0.4479 | 0.4565 | 0.4518 | 0.4485 | 0.4496 |
| NMF | 0.6191 | 0.6196 | 0.6135 | 0.6270 | 0.6204 | 0.4628 | 0.4580 | 0.4516 | 0.4677 | 0.4544 |

Table 4.2. The MAE performance for the 5-fold cross validation.

| **Upper Bound** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **RMSE** | | | | | **MAE** | | | | |
| | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 |
| NDP-JSB | **0.5578** | **0.5694** | **0.5650** | **0.5718** | **0.5649** | **0.4198** | **0.4199** | **0.4178** | **0.4267** | **0.4174** |
| HSBMF | 0.5851 | 0.5970 | 0.5943 | 0.6003 | 0.5948 | 0.4331 | 0.4339 | 0.4314 | 0.4403 | 0.4309 |
| SVD | 0.5818 | 0.5913 | 0.5835 | 0.5932 | 0.5854 | 0.4371 | 0.4349 | 0.4302 | 0.4412 | 0.4305 |
| CTR | 0.6233 | 0.6436 | 0.6313 | 0.6387 | 0.6287 | 0.4663 | 0.4799 | 0.4658 | 0.4759 | 0.4665 |
| PMF | 0.5985 | 0.6169 | 0.6112 | 0.6091 | 0.6089 | 0.4494 | 0.4563 | 0.4512 | 0.4534 | 0.4497 |
| NMF | 0.5914 | 0.6011 | 0.5930 | 0.6047 | 0.5971 | 0.4459 | 0.4473 | 0.4374 | 0.4525 | 0.4443 |

### 4.3.3 Overall Performance and Robustness Tests

Now we discuss the overall performance of our model in comparison with the baselines. We followed the experimental settings $L = 5$ on job and company latent dimensions in (Meng et al., 2018). Also, we set the maximum number of company groups $K = 60$. Other hyperparameters were set as follows: $\lambda_t = 1$, $\lambda_c = 1$, $\alpha = 1$, $\beta = 1$ and $\gamma = 1 \times 10^5$.

When some positions or companies only contain a few observations, it easily results in overflow and underflow problems in the optimization process. To solve this, we adopted the imputation technique in our model – randomly selected some companies or positions of which observations are less than a threshold, and padded salaries within that companies or positions with mean salaries. After the imputation process, the salary matrix $S$ will include three kinds of salary instances, namely, real values, empty values, padding values. Since the padding salaries are unreal and may introduce larger bias than real values, we should set different scales on the precision parameters to control the errors brought from imputation. The precision parameter $h_{ij}$ can be formulated as below:

$$h_{ij} = \begin{cases} a, & \text{if the value of } s_{ij} \text{ is real,} \\ b, & \text{if the value of } s_{ij} \text{ is padded,} \\ 0, & \text{if the } s_{ij} \text{ is empty.} \end{cases} \tag{4.26}$$

We illustrate the function of $h_{ij}$ here. Since $s_{ij}$ is generated from a normal distribution with the variance $h_{ij}^{-1}$, the model will give a less weight on $s_{ij}$ if $h_{ij}$ is smaller. Scilicet, the $h_{ij}$ can be regarded as the confidence level we believe the $s_{ij}$ is close to the true value. Intuitively we should assign less confidence on the padded salaries

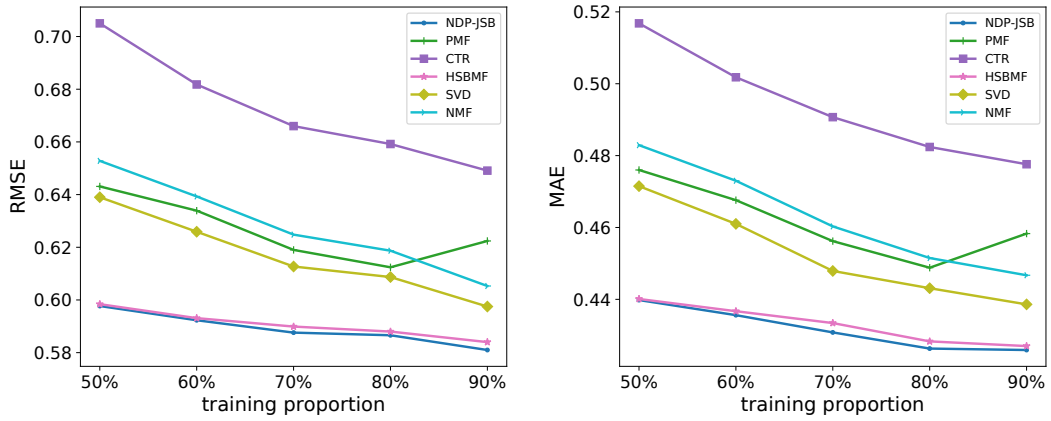
than real observations, so we should set $a > b$. In our experiments, we set $a = 5$ and $b = 1$, and the imputation threshold was set to be 10.

To validate the NDP-JSB's performance, we randomly split our dataset into 5 folds to conduct the 5-fold cross-validation. The overall RMSE and MAE results of different approaches are shown in Table 4.1 and 4.2, respectively. NDP-JSB achieve the best performance compared with all the other baselines consistently, suggesting NDP-JSB is a strong and robust approach in JSB tasks.

In order to test the robustness NDP-JSB, we held different proportions of the dataset for testing, *i.e.,* 0.1, 0.2, 0.3, 0.4, and 0.5. The results are reported in Figure 4.3. We can observe that NDP-JSB has the best performance for all different testing proportions. Also, as the training proportion increasing, the performance of the NDP-JSB model and all baselines are steadily increasing accordingly, except for PMF. It suggests that all models are stable, and NDP-JSB is a robust framework with superior performance. In addition, the PMF model may be subject to the over-fitting problem and lose some performance if the training proportion is larger than 80%.

### 4.3.4 Predicting New Company

One problem of MF-based methods is its inability to deal with new company situations, which is often referenced as "cold-start" problems. For example, a start-up company wants to hire employees in the job market, or an existing company wants to set up a branch company in a new city. Due to the lack of historical observations, those baselines can not make predictions. However, our NDP-JSB can smartly take advantage of the basic features of the company, and find a group the company may




(a) Lower Bound (b) Lower Bound

(c) Upper Bound (d) Upper Bound

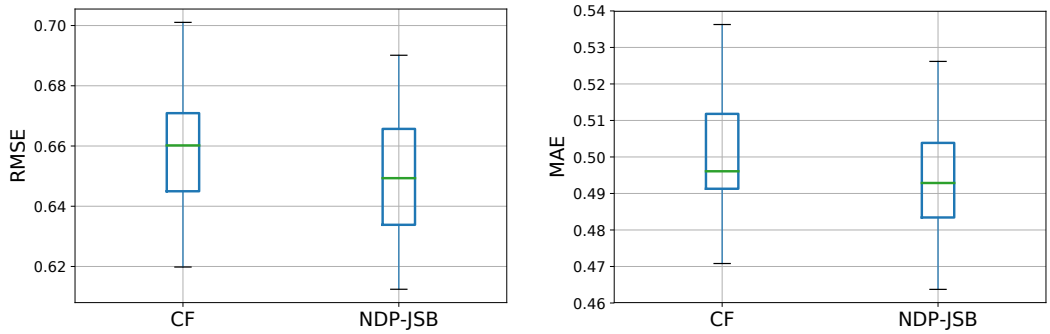
Figure 4.3. Robust testing results for the different splitting proportions.

belong to, then provide the estimations. Given only basic company features, the company group index can be inferred as

$$\widetilde{z}_{jk} \propto \exp\left\{E_q[\log(\theta_k)] + \sum_{g}^{k-1} E_q[\log(1-\theta_g)] + \sum_{d}^{D} E_q[\log \psi_{kd,x_{jd}}]\right\}. \tag{4.27}$$

Based on the obtained $\widetilde{z}_{jk}$, the salary can be estimated by equation (4.2).

To test whether NDP-JSB can give reasonable estimations for a new company, we randomly selected 0.5% instances that belong to the new companies in our dataset.




(a) RMSE (b) MAE

Figure 4.4. The box plots of results for predicting new companies.

We compared the performance with the Collaborative Filtering (CF) method, which also make use of similarity relationships of company features for salary prediction. The comparative experiments were conducted 10 times independently. The average RMSE and MAE were presented in Figure 4.4, in which we can see that NDP-JSB outperforms CF as we expected. Moreover, the p-values from the t-test are $1.88 \times 10^{-6}$ and $1.70 \times 10^{-5}$ for RMSE and MAE, respectively, demonstrating the superiority of NDP-JSB against CF is statistically significant. The competitive strength comes from the joint learning process – the model not only can make use of the company features but also gain extra information from salaries in the job market.

## 4.4 Case Studies

As a generative model, NDP-JSB can also provide multiple distribution information regarding positions and companies; hence, give valuable advice related to salary benchmarking. Based on case studies, we will show useful findings in three aspects, including position grouping, company grouping, and job profiling.

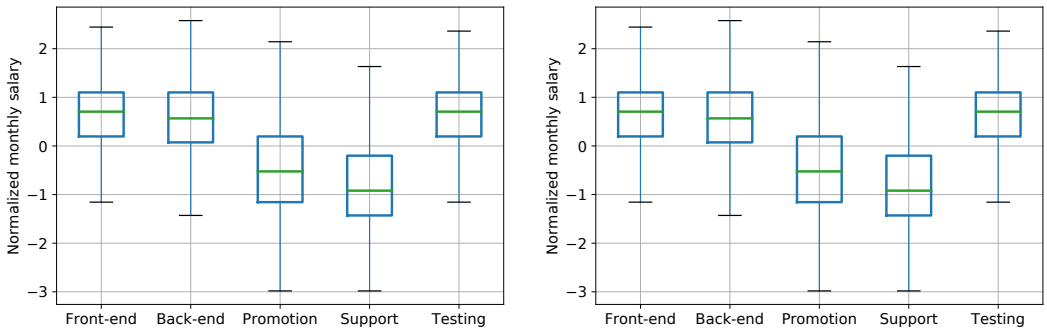

(a) Front-end (b) Back-end (c) Promotion

(d) Support (e) Testing

Figure 4.5. Word clouds for the five job groups.

### 4.4.1 Position Grouping

In the position representation module, each position is represented by five latent topics. To understand what are the main characteristics of those topics, we took the top eight keywords in each topic and demonstrated them in Figure 4.5. As can be seen, the keywords are skill sets emphasized by different types of professionals, including "Front-end", "Back-end", "Testing", "Support", and "Promotion". Based on the clustering results, we compared the salary distributions for the five types of jobs. In Figure 4.6, we can observe that the technical jobs (*i.e., front-end, back-end, and testing*) have relatively better compensations. Also, although Promotion jobs may have relatively lower salaries, their variation range is the largest, suggesting top promotion people have high potential to earn much.




(a) Lower Bound (b) Upper Bound

Figure 4.6. Salary distributions for the five job groups.

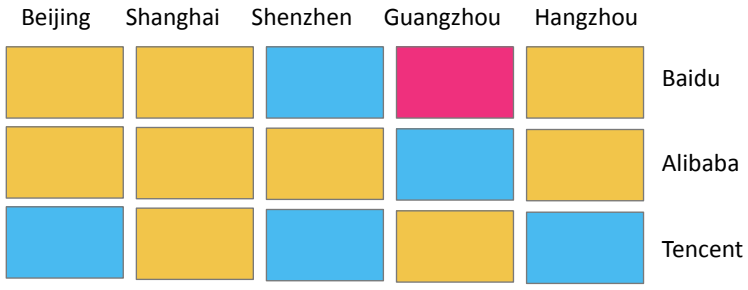
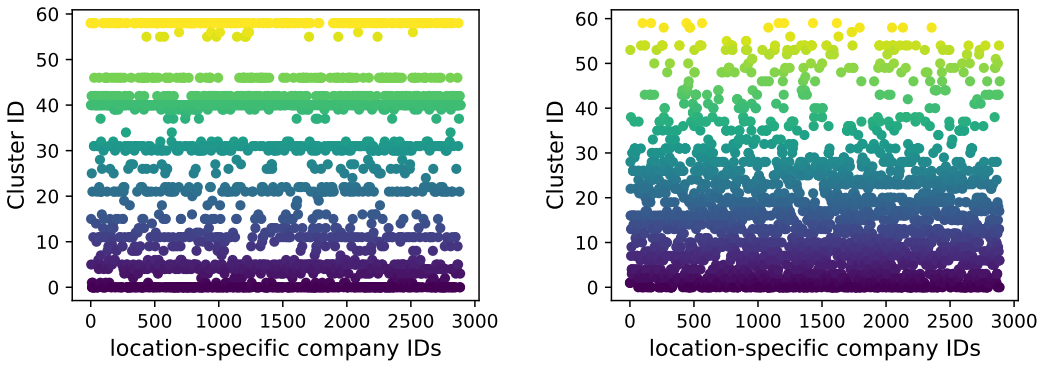



Figure 4.7. Grouping results for 3 famous companies.

### 4.4.2 Company Grouping

In the company representation module, every company was assigned to a group. We selected three famous companies –"Baidu", "Alibaba", and "Tencent"– to study the rationality of grouping results. These companies are the biggest in the field of Mobile Internet. They share much in common, and all of them set subsidiaries in the five cites we study. Based on the domain knowledge, we expect two findings from the clustering results. First, as the companies are similar in many ways, they are supposed to be grouped together. Second, the subsidiaries in different cities bear different functions and deal with different businesses, so the subsidiaries belong to a company should have different grouping results. We displayed the grouping results in Figure 4.7, in




(a) NDP-JSB (b) K-modes

Figure 4.8. Grouping results for all companies.

which each block represents a location-specific company, and each color represents a group ID. The 15 branches are classified into 3 main groups, and each company has 2 to 3 classes across the five cites. The results are consistent with our expectations and verify the effectiveness of NDP-JSB in terms of company grouping.

The companies are grouped on the basis of the NDP module. One advantage of NDP is that we do not need to know the group number in advance. NDP will find the optimal group number on the whole. We displayed the clustering results for all companies and compared it with another commonly used clustering method K-modes in Figure 4.8. We set the group number of K-modes equal to the maximum group number of NDP-JSB. K-modes make use of the company features to perform the clustering. Every row in the figure represents a group ID, and every point represents a company. If the points belong to the same group, they will lie in the same row with the same color. The points in Figure 4.8 (a) are more compacted than Figure 4.8 (b). NDP-JSB can intelligently figure out the optimal group number is less than 60, while K-modes is incapable of deciding the reasonable group number by itself.

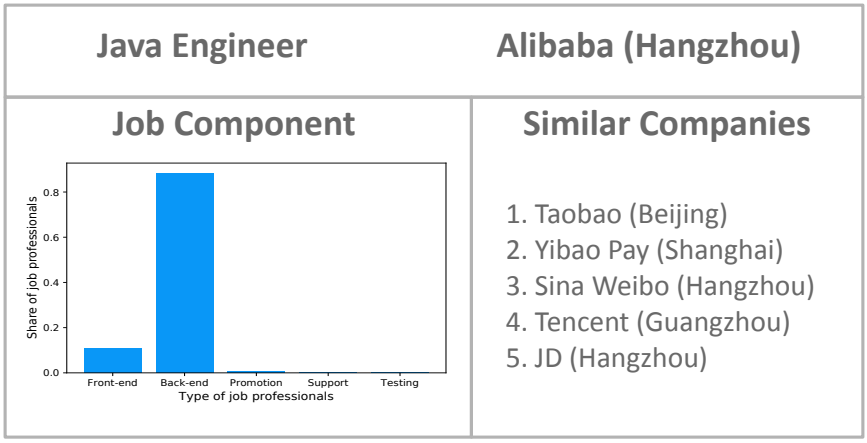



Figure 4.9. An example of job profiling.

### 4.4.3 Job Profiling

NDP-JSB can provide certain explanations along with salary estimations, which can benefit inexperienced C&B managers for profiling a job. In particular, NDP-JSB can provide the share of job professionals that each position emphasizes on, as well as other similar companies in the job market. Those similar companies can be used for further data sourcing and competition analysis. Figure 4.9 shows an example of job profiling, which is a real case in our dataset. "Alibaba (Hangzhou)" wanted to hire a Java Engineer in the job market. Learned from the NDP-JSB, Java Engineer emphasizes on the professionals of the back-end for around 85% and frond-end for around 15%. The competitive companies in the job market include "Taobao (Beijing)", "Yibao Pay (Shanghai)", and "Sina Weibo (Hangzhou)".

## 4.5 Related Work

We summarize related work into two categories. We (1) discuss related research on the job salary benchmarking problem and (2) summarize related methodologies with data-driven techniques.

### 4.5.1 Job Salary Benchmarking

Salary estimation has drawn much attention from human resource management due to its key role in attracting, motivating, retaining talent, as well as in reducing operating costs for organizations.

Some studies intend to understand the essential factors that influence salary level from an individual perspective, such as age, gender, and the timing of motherhood (Lazar, 2004; Jerrim, 2015; Hamlen & Hamlen, 2016; Correll et al., 2007). (Frydman & Jenter, 2010; Gong & Li, 2013; Brick et al., 2006) tried to understand what determines the high revenues of CEOs, while (Peng & Röell, 2014; Peng & Roell, 2008) discovered indications that CEOs intend to raise their revenues through managerial manipulations. There are also a large number of studies emphasizing pay equity (Chang & Hahn, 2006; Berkowitz et al., 1987; Scarpello & Jones, 1996; Terpstra & Honoree, 2003). Still, other researchers investigate the ways compensation is shaped by peer comparative organizations and individuals (Blankmeyer et al., 2011; Faulkender & Yang, 2010). (Ferris et al., 2001) found that excellent social skills and related general mental ability serve as strong explanations for individuals job performance and salary levels. (Khongchai & Songmuang, 2016a, 2016b) predicted students income by examining their demographic features and stated that students would be

motivated to study hard if they learned about their salary prediction results. In addition, researchers are concerned about how to design compensation structures to boost the performance of employees (Bergmann & Scarpello, 2002).

The existing work mainly focused on understanding the determinants of the salary range, while how to benchmark salary by jointly considering internal compensation policies and external market pricing from the C&B departments perspective has not been well addressed. As a widely applied process in practice, some human resource handbooks (Edwards et al., 2003; Armstrong, 2006) have provided guidance on how to conduct JSB using surveys and statistical methods, although they emphasize the importance of designing a self-consistent and justifiable internal compensation structure; meanwhile, they have not provided a unified solution for internal and external factors. (Lin et al., 2017) proposed a framework for company profiling that can simultaneously predict job salary; however, their framework is based on a dataset of employees positive and negative comments about their employers; thus, their method cannot predict salary based on job responsibilities or company information or provide advice for new startups.

Our NDP-JSB method not only makes effective use of the correlations among positions and companies but also has the ability to conduct JSB for new companies.

### 4.5.2 Data-Driven Predictive Models

Our method for addressing the JSB problem can be classified as a probabilistic graphic model. Probabilistic graphical models use a graph-based representation to encode a complex distribution over a high-dimensional space, where the nodes in the graph

represent variables (observable or unobservable), and the edges represent the interactions between them (Koller & Friedman, 2009). Due to their strong ability to model the complex relationships between features with uncertainty, as well as their explanatory-friendly characteristics, probabilistic graphical models are broadly used in a variety of machine learning tasks (Ghahramani, 2015). There are three modules in our framework, which are associated with the matrix factorization (MF) method, the topic model, and the non-parametric Dirichlet process. In the following, we will present multiple relative techniques for them. The MF family is a technique factorizing a high-dimension sparse matrix $S$ into two lower rank matrices, $A$ and $B$, and the cross product $\hat{S}$ of $A$ and $B$ is close to the original matrix $S$. As an early technique in the MF family, SVD was first proposed to identify latent semantic factors carried in $S$, and then it was applied to the recommendation applications due to its effectiveness in "guessing" the missing values in $S$ by the cross product procedure (Adomavicius & Tuzhilin, 2005). First, to calculate the distance between $S$ and $\hat{S}$ in the optimization process, the researcher adopts an "imputation" technique in which the missing values in $S$ are filled by guessing the values. However, the "imputation" technique may distort the actual distribution and easily lead to overfitting (Kim & Yum, 2005), in which case, the researcher can replace "imputation" by integrating an auxiliary indication matrix to mark the positions of the existing values in $S$. Moreover, (Paterek, 2007) suggested using regularizers to address the overfitting issue by constraining the values in $A$ and $B$. After that, (Koren, 2008) proposed a method of integrating the implicit neighbourhood information in $A$ and $B$ to improve the prediction efficiency for recommender systems. Another two commonly used MF techniques are NMF and

PMF. NMF adopts the MF structure but constrains the variables to be non-negative, demonstrating that the constraints are able to learn the parts-based representations (Lee & Seung, 1999, 2001). PMF places zero-mean spherical Gaussian priors on matrices $A$ and $B$ (Mnih & Salakhutdinov, 2008). In our framework, we adopted the PMF structure in the prediction module because it belongs to the probabilistic graphical model and is easy to extend in a more complicated graphical structure.

An early developed topic model named pLSI (Hofmann, 2017) is a probabilistic model with three layers. The first layer is used to generate documents, the second layer generates topics of each document, and the last layer describes the word selection process based on a topic-word occurrence frequency distribution. Later, (Blei, Ng, & Jordan, 2003) proposed the famous Latent Dirichlet Allocation (LDA) model, which is similar to pLSI with its three-layer structure. In contrast, LDA places Dirichlet priors on both document-topic and topic-word distributions, and the refined architecture is demonstrated to be more effective in learning the document-topic and topic-word distributions. Afterward, (Wang & Blei, 2011) incorporated LDA into an MF framework for scientific article recommendations.

Additionally, LDA models have been implemented broadly in the areas of text mining, document classification (Chen, Xia, Jin, & Carroll, 2015; Pavlinek & Podgorelec, 2017), image recognition (Rasiwasia & Vasconcelos, 2013; Gomez, Patel, Rusiñol, Karatzas, & Jawahar, 2017), and brand management (Guo, Barnes, & Jia, 2017; Tirunillai & Tellis, 2014). In our model, we adopted the LDA structure to learn the latent job representations from job descriptions.

The Dirichlet process (DP) is commonly used to generate a set of values to form a

simplex, and the simplex can be used for the parameters of a multinomial distribution. As DP is conjugated with the multinomial distribution, we normally place a DP prior on a multinomial distribution for a Bayesian probabilistic model in practice due to its mathematics-friendly characteristics. If the parameters of the multinomial distribution are drawn not from one DP but from more than one DP, namely, it is a DP mixture, what kind of process can represent the generation procedure of a DP mixture? (Ferguson, 1973; Antoniak, 1974) provided an answer by proposing the non-parametric Dirichlet process (NDP). The word non-parametric can be interpreted as an infinite number of mixtures. The NDP is generated from a base distribution and a positive parameter. There was no explicit form for the posterior distribution of the NDP, so the application was limited until (Ishwaran & James, 2001) described it with a stick-breaking view, and the development of Gibbs and Monte-Carlo Markov Chain (MCMC) sampling methods enabled it to be solved in an approximate way (Ishwaran & James, 2001; Neal, 2000). Afterward, (Blei et al., 2006) proposed a variational inference (VI) technique to solve the algorithms that can mitigate the computational complexity caused by sampling methods. The NDP has been widely applied in machine learning tasks, especially for density estimation and clustering (J. Zhang, Ghahramani, & Yang, 2005; Teh, Jordan, Beal, & Blei, 2005; Dahl, 2006; Escobar & West, 1995; Nguyen, Gupta, Rana, Li, & Venkatesh, 2016; Xue, Liao, Carin, & Krishnapuram, 2007; X. Zhang et al., 2018). The merit of NDP in the clustering task is that people do not need to know the number of clusters, and the model can learn an optimal number of clusters by itself. In this way, people bypass the potential error caused by incorrectly pre-defining the number of mixtures. We

adopted an NDP structure to learn the latent company representations and the VI technique to solve the algorithms efficiently.

## 4.6 Conclusions

In this paper, we addressed the job salary benchmarking (JSB) problem from a more fine-grained and data-driven perspective by modelling large-scale real-world online recruitment data. Specifically, we designed a non-parametric Dirichlet-process-based latent factor model for JSB, namely, the NDP-JSB, which can jointly model the latent representations of both company and job position. Our method can effectively predict job salaries for each company and job position with rich contexts. We evaluated our model with extensive experiments on a large-scale real-world dataset. The experimental results clearly validated the effectiveness of the NDP-JSB in terms of salary prediction and also demonstrated its strength in revealing patterns between job categories and companies, which makes our prediction results more interpretable and can further benefit the decision-making process of talent management.

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

With the development of big data techniques and the cumulation of digital data from talents, we believe it is an inevitable trend to reform the managerial patterns about talents, from more subjective to more objective. Along this line, in this dissertation, we developed a few data-driven techniques for solving the practical issues related to talent recruitment, such as job mobility prediction and salary benchmarking.

We first developed a neural-network-based model in order to predict an individual's future career transitions in terms of the next employer and stay duration by learning one's historical working experiences. Also, we proposed a Matrix Factorization (MF) based framework to estimate job salaries based on information provided in the typical job advertisements. Furthermore, we developed a Non-Parametric Dirichlet (NPD) based model aimed at solving the "cold-start" problems in MF-based model, and providing more interpretation for Salary Benchmarking tasks.

Here we illustrate several future research directions along with this dissertation. Job mobility prediction is a valuable research area that deserved to be studied thoroughly. We have shed light on job transition at the company level, other possible research directions include understanding the job transition at the position level. *i.e.,* 1) How to get a promotion along with a special job track? 2) What factors have impacts on the individual's shift of career interests, and *etc.* Another important topic

about the position is to compare the job title ranks between different companies.

Talent management involves many important subjects that can be viewed in new ways and solved under information technology progresses. Along with line, we will thrive on the way to put forward new methods associated with efficient talent selection, job-person fit, personal career advance lesson recommendation, employee's performance prediction, and *etc.*

# BIBLIOGRAPHY

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734–749.

Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In *Recommender systems handbook* (pp. 191–226). Springer.

Alic, B., et al. (2016). Talent recruitment and selection–issue and challenge for organizations in the republic of moldova. *Annals-Economy Series*, *1*, 62–68.

Anderson, W. J. (2012). *Continuous-time markov chains: An applications-oriented approach*. Springer Science & Business Media.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, 1152–1174.

Armstrong, M. (2006). *A handbook of human resource management practice*. Kogan Page Publishers.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baltrunas, L., Ludwig, B., & Ricci, F. (2011). Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth acm conference on recommender systems* (pp. 301–304).

Baltrunas, L., & Ricci, F. (2009). Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the third acm conference on recommender systems* (pp. 245–248).

Bao, T., Cao, H., Chen, E., Tian, J., & Xiong, H. (2012). An unsupervised approach to modeling personalized contexts of mobile users. *Knowledge and Information Systems*, *31*(2), 345–370.

Bartel, A. P. (1979). The migration decision: What role does job mobility play? *The American Economic Review*, *69*(5), 775–786.

Bergmann, T., & Scarpello, V. (2002). Compensation decision making, 4e. *Ohio: South-Western Co*.

Berkowitz, L., Fraser, C., Treasure, F. P., & Cochran, S. (1987). Pay, equity, job gratifications, and comparisons in pay satisfaction. *Journal of Applied Psychology*, *72*(4), 544.

Blankmeyer, E., LeSage, J. P., Stutzman, J., Knox, K. J., & Pace, R. K. (2011). Peer-group dependence in salary benchmarking: a statistical model. *Managerial and Decision Economics*, *32*(2), 91–104.

Blei, D. M., Jordan, M. I., et al. (2006). Variational inference for dirichlet process mixtures. *Bayesian analysis*, *1*(1), 121–143.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Brick, I. E., Palmon, O., & Wald, J. K. (2006). Ceo compensation, director compensation, and firm performance: Evidence of cronyism? *Journal of Corporate Finance*, *12*(3), 403–423.

Chang, E., & Hahn, J. (2006). Does pay-for-performance enhance perceived distributive justice for collectivistic employees? *Personnel Review*, *35*(4), 397–412.

Chen, X., Xia, Y., Jin, P., & Carroll, J. (2015). Dataless text classification with descriptive lda. In *Twenty-ninth aaai conference on artificial intelligence.*

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Collings, D. G., Wood, G. T., & Szamosi, L. T. (2018). Human resource management: A critical approach. In *Human resource management* (pp. 1–23). Routledge.

Correll, S. J., Benard, S., & Paik, I. (2007). Getting a job: Is there a motherhood penalty? *American journal of sociology*, *112*(5), 1297–1338.

Dahl, D. B. (2006). Model-based clustering for expression data via a dirichlet process mixture model. *Bayesian inference for gene expression and proteomics*, *4*, 201–218.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., & Song, L. (2016). Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1555–1564).

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on machine learning* (pp. 272–279).

Edwards, J. E., Scott, J. C., & Raju, N. S. (2003). *The human resources program-evaluation handbook*. SAGE Publications, Incorporated.

Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, *90*(430), 577–588.

Farndale, E., Scullion, H., & Sparrow, P. (2010). The role of the corporate hr function in global talent management. *Journal of world business*, *45*(2), 161–168.

Faulkender, M., & Yang, J. (2010). Inside the black box: The role and composition of compensation peer groups. *Journal of Financial Economics*, *96*(2), 257–270.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, 209–230.

Ferris, G. R., Witt, L. A., & Hochwarter, W. A. (2001). Interaction of social skill and general mental ability on job performance and salary. *Journal of Applied Psychology*, *86*(6), 1075.

Frydman, C., & Jenter, D. (2010). Ceo compensation. *Annu. Rev. Financ. Econ.*, *2*(1), 75–102.

Gao, H., Tang, J., Hu, X., & Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th acm conference on recommender systems* (pp. 93–100).

Ge, Y., Liu, C., Xiong, H., & Chen, J. (2011). A taxi business intelligence system. In *Proceedings of the 17th acm sigkdd international conference on knowledge discovery and data mining* (pp. 735–738).

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452.

Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D., & Jawahar, C. (2017). Self-supervised learning of visual features through embedding images into text topic spaces. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4230–4239).

Gong, J. J., & Li, S. (2013). Ceo incentives and earnings prediction. *Review of Quantitative Finance and Accounting*, *40*(4), 647–674.

Grant, E. A. (2008). How to retain talent in india. *MIT Sloan Management Review*, *50*(1), 6.

Guo, Y., Barnes, S. J., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management*, *59*, 467–483.

Hamlen, K. R., & Hamlen, W. A. (2016). Faculty salary as a predictor of student outgoing salaries from mba programs. *Journal of Education for Business*, *91*(1), 38–44.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hofmann, T. (2017). Probabilistic latent semantic indexing. In *Acm sigir forum* (Vol. 51, pp. 211–218).

Ishwaran, H., & James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, *96*(453), 161–173.

Jerrim, J. (2015). Do college students make better predictions of their future income than young adults in the labor force? *Education Economics*, *23*(2), 162–179.

Jing, H., & Smola, A. J. (2017). Neural survival recommender. In *Proceedings of the tenth acm international conference on web search and data mining* (pp. 515–524).

Johnson, C. B., Riggs, M. L., & Downey, R. G. (1987). Fun with numbers: Alternative models for predicting salary levels. *Research in Higher Education*, *27*(4), 349–362.

Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth acm conference on recommender systems* (pp. 79–86).

Karr, A. (2017). *Point processes and their statistical inference*. Routledge.

Keith, K., & McWilliams, A. (1995). The wage effects of cumulative job mobility. *ILR Review*, *49*(1), 121–137.

Kenthapadi, K., Ambler, S., Zhang, L., & Agarwal, D. (2017). Bringing salary transparency to the world: Computing robust compensation insights via linkedin salary. In *Proceedings of the 2017 acm on conference on information and knowledge management* (pp. 447–455).

Kenthapadi, K., Chudhary, A., & Ambler, S. (2017). Linkedin salary: A system for secure collection and presentation of structured compensation insights to job seekers. In *2017 ieee symposium on privacy-aware computing (pac)* (pp. 13–24).

Khongchai, P., & Songmuang, P. (2016a). Implement of salary prediction system to improve student motivation using data mining technique. In *2016 11th international conference on knowledge, information and creativity support systems (kicss)* (pp. 1–6).

Khongchai, P., & Songmuang, P. (2016b). Random forest for salary prediction system to improve students' motivation. In *2016 12th international conference on signal-image technology & internet-based systems (sitis)* (pp. 637–642).

Kim, D., & Yum, B.-J. (2005). Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, *28*(4), 823–830.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining* (pp. 426–434).

Koren, Y., & Bell, R. (2015). Advances in collaborative filtering. In *Recommender systems handbook* (pp. 77–118). Springer.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8).

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lawler, E. E. (2017). *Reinventing talent management: Principles and practices for the new world of work*. Berrett-Koehler Publishers.

Lazar, A. (2004). Income prediction via support vector machine. In *Icmla* (pp. 143–149).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Proceedings of the 13th international conference on neural information processing systems* (pp. 535–541).

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556–562).

Li, H., Ge, Y., Zhu, H., Xiong, H., & Zhao, H. (2017). Prospecting the career development of talents: A survival analysis perspective. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 917–925).

Li, L., Jing, H., Tong, H., Yang, J., He, Q., & Chen, B.-C. (2017). Nemo: Next career move prediction with contextual embedding. In *Proceedings of the 26th international conference on world wide web companion* (pp. 505–513).

Lin, H., Zhu, H., Zuo, Y., Zhu, C., Wu, J., & Xiong, H. (2017). Collaborative company profiling: Insights from an employee's perspective. In *Aaai* (pp. 1417–1423).

Liu, Y., Zhang, L., Nie, L., Yan, Y., & Rosenblum, D. S. (2016). Fortune teller: Predicting your career path. In *Aaai* (Vol. 2016, pp. 201–207).

Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, *10*(2), 1273–1284.

Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. *ACL*, 11–19.

Mei, H., & Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in neural information processing systems* (pp. 6754–6764).

Meng, Q., Zhu, H., Xiao, K., & Xiong, H. (2018). Intelligent salary benchmarking for talent recruitment: A holistic matrix factorization approach. In *2018 ieee international conference on data mining (icdm)* (pp. 337–346).

Miller, A. R. (2011). The effects of motherhood timing on career path. *Journal of population economics*, *24*(3), 1071–1100.

Mnih, A., & Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems* (pp. 1257–1264).

Muriithi, F. W., & Makau, M. S. (2017). Talent management: A conceptual framework from review of literature and a research agenda. *Journal of Human Resource Management*, *5*(6), 90–94.

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, *9*(2), 249–265.

Nguyen, V., Gupta, S., Rana, S., Li, C., & Venkatesh, S. (2016). A bayesian nonparametric approach for multi-label classification. In *Asian conference on machine learning* (pp. 254–269).

Panniello, U., Tuzhilin, A., & Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, *24*(1-2), 35–65.

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., & Pedone, A. (2009). Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third acm conference on recommender systems* (pp. 265–268).

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of kdd cup and workshop* (Vol. 2007, pp. 5–8).

Pavlinek, M., & Podgorelec, V. (2017). Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, *80*, 83–93.

Peng, L., & Roell, A. (2008). Manipulation and equity-based compensation. *American Economic Review*, *98*(2), 285–90.

Peng, L., & Röell, A. (2014). Managerial incentives and stock price manipulation. *The Journal of Finance*, *69*(2), 487–526.

Porter, C. O., Cordon, D. E., & Barber, A. E. (2004). The dynamics of salary negotiations: Effects on applicants'justice perceptions and recruitment decisions. *International Journal of Conflict Management*, *15*(3), 273–303.

Qin, C., Zhu, H., Xu, T., Zhu, C., Jiang, L., Chen, E., & Xiong, H. (2018). Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *The 41st international acm sigir conference on research & development in information retrieval* (pp. 25–34).

Rasiwasia, N., & Vasconcelos, N. (2013). Latent dirichlet allocation models for image classification. *IEEE transactions on pattern analysis and machine intelligence*, *35*(11), 2665–2679.

Řehůřek, R., & Sojka, P. (2010, May 22). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA. (`http://is.muni.cz/publication/884893/en`)

Ren, K., Qin, J., Zheng, L., Yang, Z., Zhang, W., Qiu, L., & Yu, Y. (2019). Deep recurrent survival analysis. *AAAI*.

Rendle, S., Gantner, Z., Freudenthaler, C., & Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international acm sigir conference on research and development in information retrieval* (pp. 635–644).

Rosenfeld, R. A. (1992). Job mobility and career processes. *Annual review of Sociology*, *18*(1), 39–61.

Scarpello, V., & Jones, F. F. (1996). Why justice matters in compensation decision making. *Journal of organizational behavior*, *17*(3), 285–299.

Schau, C. G., & Heyward, V. H. (1987). Salary equity: Similarities and differences in outcomes from two common prediction models. *American Educational Research Journal*, *24*(2), 271–286.

Shen, D., Zhu, H., Zhu, C., Xu, T., Ma, C., & Xiong, H. (2018). A joint learning approach to intelligent job interview assessment. In *Ijcai* (pp. 3542–3548).

Stone, D. L., & Rosopa, P. J. (2017). *The advantages and limitations of using meta-analysis in human resource management research.* Elsevier.

Sullivan, S. E., & Al Ariss, A. (2019). Making sense of different perspectives on career transitions: A review and agenda for future research. *Human Resource Management Review*, 100727.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems* (pp. 1385–1392).

Terpstra, D. E., & Honoree, A. L. (2003). The relative importance of external, internal, individual and procedural equity to pay satisfaction: Procedural equity may be more important to employees than organizations believe. *Compensation & Benefits Review*, *35*(6), 67–74.

Tirunillai, S., & Tellis, G. J. (2014). Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research*, *51*(4), 463–479.

Topel, R. H., & Ward, M. P. (1992). Job mobility and the careers of young men. *The Quarterly Journal of Economics*, *107*(2), 439–479.

Vance, C. M. (2005). The personal quest for building global competence: A taxonomy of self-initiating career path strategies for gaining business experience abroad. *Journal of World Business*, *40*(4), 374–385.

Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th acm sigkdd international conference on knowledge discovery and data mining* (pp. 448–456).

Wegener, B. (1991). Job mobility and social ties: Social resources, prior job, and status attainment. *American Sociological Review*, 60–71.

Xiao, K., Liu, Q., Liu, C., & Xiong, H. (2017). Price shock detection with an influence-based model of social attention. *ACM Transactions on Management Information Systems (TMIS)*, *9*(1), 2.

Xu, H., Yu, Z., Xiong, H., Guo, B., & Zhu, H. (2015). Learning career mobility and human activity patterns for job change analysis. In *2015 ieee international conference on data mining (icdm)* (pp. 1057–1062).

Xu, H., Yu, Z., Yang, J., Xiong, H., & Zhu, H. (2016). Talent circle detection in job transition networks. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 655–664).

Xu, H., Yu, Z., Yang, J., Xiong, H., & Zhu, H. (2018). Dynamic talent flow analysis with deep sequence prediction modeling. *IEEE Transactions on Knowledge and Data Engineering*.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international acm sigir conference on research and development in informaion retrieval* (pp. 267–273).

Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, *8*(Jan), 35–63.

Yao, Y., Zhao, W. X., Wang, Y., Tong, H., Xu, F., & Lu, J. (2017). Version-aware rating prediction for mobile app recommendation. *ACM Transactions on Information Systems (TOIS)*, *35*(4), 38.

Ye, Z., Zhang, L., Xiao, K., Zhou, W., Ge, Y., & Deng, Y. (2018). Multi-user mobile sequential recommendation: An efficient parallel computing paradigm. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2624–2633).

Zhang, J., Ghahramani, Z., & Yang, Y. (2005). A probabilistic model for online document clustering with application to novelty detection. In *Advances in neural information processing systems* (pp. 1617–1624).

Zhang, L., Xiao, K., Liu, Q., Tao, Y., & Deng, Y. (2015). Modeling social attention for stock analysis: An influence propagation perspective. In *Data mining (icdm), 2015 ieee international conference on* (pp. 609–618).

Zhang, L., Xiao, K., Zhu, H., Liu, C., Yang, J., & Jin, B. (2018). Caden: A context-aware deep embedding network for financial opinions mining. In *2018 ieee international conference on data mining (icdm)* (pp. 757–766).

Zhang, X., Li, W., Nguyen, V., Zhuang, F., Xiong, H., & Lu, S. (2018). Label-sensitive task grouping by bayesian nonparametric approach for multi-task multi-label learning. In *Ijcai* (pp. 3125–3131).

Zhou, N., Cheung, W. K., Qiu, G., & Xue, X. (2011). A hybrid probabilistic model for unified collaborative and content-based image tagging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(7), 1281–1294.

Zhu, C., Zhu, H., Xiong, H., Ding, P., & Xie, F. (2016). Recruitment market trend analysis with sequential latent variable models. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 383–392).

Zhu, H., Chen, E., Xiong, H., Yu, K., Cao, H., & Tian, J. (2015). Mining mobile user preferences for personalized context-aware recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *5*(4), 58.

# APPENDIX

# APPENDIX A

# REPLICATIONS AND PROOFS

## A.1 Experimental settings for replications in Chapter 2

### A.1.1 Position Normalization

The position names in the dataset are not standardized, they are not even in one kind of language. So we used multiple keywords matching method to normalize those names into 26 categories. The normalized types of positions are listed in Table A.1.

Table A.1. Position name normalization.

| **Position Types** | | | |
|---|---|---|---|
| Accounting | Sales | Administrative | Supporter |
| Consulting | Social Service | Engineering | Education |
| Entrepreneurship | Finance | Health Care | Human Resources |
| Information Technology | Law | Military | Marketing |
| Media | Operation | Real Estate | Purchaser |
| Product Management | Quality Assurance | Researcher | Program Management |
| Arts and Design | Business Development | | |

### A.1.2 Data Preprocessing

Our dataset contains sequential data with different length. In order to fit to the non-sequential models (*i.e.* Logistic regression, Decision Tree and Random Forest), we

have to transform the input features into a vector with a fixed length. To deal with the problem, we used a *Bag-of-Companies* model, which is similar to the concept of *Bag-of-words*. We ignored the sequential information among companies, and only counted the occurrence number of each company, and calculated the cumulative duration in that company. Except for companies and durations, we only recorded the last values of the sequential features. The non-sequential features remained the same with what we used in HCPNN. At last, we concatenated all the features into one vector with a fixed length. In this way, we fit our sequential data to the non-sequential models.

### A.1.3 Baseline Setting

We summarize the details of baseline methods, especially the modified CRF and MHP methods as follows:

- **CTMC** (Anderson, 2012): It is a stochastic model to describe a series of events which the state spaces are discrete, yet the time is continuous. It also called the memoryless process, because the future states are solely dependent on the present state. This is a model that can predict the next state and the duration of the next state simultaneously. In our experiment, we set the state to be working in a specific company.

- **PP** (Karr, 2017): It defines the occurrence probability of an event over a real-time line. When the instantaneous occurrence probability $\lambda$ is a constant, we call it stationary or homogeneous Poisson Process, which we deployed in this paper.

- **CRF**(Lafferty et al., 2001): It is an undirected graphical and discriminative model allowing long-distance dependencies and integration of rich features. The nodes in the graph denote the random variables, while the edges denote the direct influence or dependency relations between the variables. We used the linear conditional field in our experiment. We have

$$P(Y|X) = exp\left(\sum_{i,k} \lambda_k t_k(Y_{i-1}, Y_i, X, i) + \sum_{i,l} u_l s_l(Y_i, X, i)\right), \tag{A.1}$$

where $t_k(Y_{i-1}, Y_i, X, i)$ denotes the transition probability transferred from $Y_{i-1}$ to $Y_i$ at the sequence position $(X, i)$, which corresponding the transfer probabilities from one company to another company. $s_l(Y_i, X, i)$ represents the probability of $Y_i$ at the sequence position $(X, i)$. $\lambda_k$ and $u_l$ are two weight coefficients for these two functions. In our problem, $Y_i$ means a specific company in the position $i$. The train process is the same with what is broadly applied in NLP tasks [1] , but the predicting process is different, since the viterbi algorithm will use the future information to deduce the historical sequence. To handle this problem, we used the original definition as described in Equation A.1 to calculate the probabilities of next employer. More specifically, we used the parameters $s_l(Y_i, X, i)$, $t_k(Y_{i-1}, Y_i, X, i)$ learned from training process, combining with the known historical company sequence $\overrightarrow{Q(u)}$ to calculate the probabilities. We set the tunning parameters $\lambda_k$ and $u_l$ to be 1.

- **MHP**(Mei & Eisner, 2017): We used a multi-variable Hawkes process defined in (Mei & Eisner, 2017) in this paper with modifications. It assumes the event

[1] https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/crf

intensity rate is not only caused by a self-excited rate $\mu$, but also influenced by the events happened before, and the influence degree is proportional to the time span between the events and event types. Suppose a company $c \in C$, where $C$ denotes the whole company set, and we have $N$ companies in total. We want to simulate the instantaneous occurrence probability $\lambda(\tau)$ over time $(0, \infty)$ by training three parameters, namely self-excited intensity rate $\mu$, the event influence parameter $\sigma$ and the time decay parameter $\delta$. To facilitate the understanding of MHP, we first summarize the notation descriptions and their dimensions in Table A.2.

The key process of the algorithm can be described in equation:

$$\lambda_{c_{g+1}}(\tau) = \mu_{c_{g+1}}(\tau) + \sum_{i=1}^{g} \sigma_{c_i, c_{g+1}} \exp\left(-\delta_{c_i, c_{g+1}}((g+1) - i)\right), \tag{A.2}$$

where $\lambda_{c_{g+1}}(\tau)$ is the individual turnover probability when she works for her $(g+1)$-th company $c_{g+1}$. For every person, the turnover probability is influenced by two factors, one is the self-excited factor of $\mu_{c_{g+1}}(\tau)$, the other is all the employers he/she worked for before. The influence degree is controlled by the company type and the time index distance. The longer of the time distance, the weaker of the influence. The objective function is the maximize the log-likelihood of predicting duration as describe in Equation 2.10. When we get the $\lambda_{c_{g+1}}(\tau)$ for every person for the next company, we can use the survival analysis integration function described in Equation 2.12 to compute the expectation of the duration.

### A.1.4 T-test for HCPNN and HCPNO

The results of standard student t-test on comparing HCPNN and HCPNO are summarized in Table A.3.

Table A.2. Notation description in MHP.

| **Notation** | **Description** |
|---|---|
| $i, g \in N^+$ | the time index in the company sequence. |
| $c_i \in C$ | a person's $i$-th employer in his/her career path. |
| $\lambda(\tau)$ | the individual turnover probability. |
| $\mu \in \mathbb{R}^N$ | the self-excited turnover probability |
| $\sigma \in \mathbb{R}^{NN}$ | the intensity influence rate between pair-wised companies |
| $\delta \in \mathbb{R}+^{NN}$ | the time decay parameter between pair-wised companies |

Table A.3. The results of standard student t-test with 95% confidence interval.

| **Model** | HCPNN | HCPOP | p-value |
|---|---|---|---|
| **Acc@1** | $0.0726 \pm 0.0004$ | $0.0712 \pm 0.0003$ | 1.3e-5 |
| **Acc@15** | $0.4039 \pm 0.0009$ | $0.3995 \pm 0.0010$ | 4.9e-8 |
| **Acc@30** | $0.5353 \pm 0.0010$ | $0.5308 \pm 0.0009$ | 4.3e-8 |
| **MRR** | $0.1555 \pm 0.0004$ | $0.1534 \pm 0.0004$ | 9.6e-7 |
| **MAE** | $2.7288 \pm 0.0056$ | $2.7357 \pm 0.0043$ | 5.7e-4 |
| **RMSE** | $3.8846 \pm 0.0084$ | $3.8925 \pm 0.0073$ | 1.0e-2 |

## A.2 Proof of the Variational Inference Process in Section 4.2.3

Before showing the proof of each term in the Evidence Lower Bound (ELBO) of $\mathcal{L}_0$, we first summarize some formulas which will be used in related proofs.

$$\frac{\partial(\log[\mathbf{B}(X)])}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\sum_i^I \log[\Gamma(x_i)] - \log\left[\Gamma\sum_i^I x_i\right]\right) = \mathbf{\Psi}(x_i) - \mathbf{\Psi}(\sum_i x_i), \tag{A.2.1}$$

where $X = x_1, x_2, ..., x_I$.

If $\Theta \sim Dirichlet(\alpha_1, \alpha_2, ..., \alpha_i, ..., \alpha_I)$, where $\theta$ is a simplex, then we have

$$E[\log(\theta_i)] = \frac{\partial \log(\mathbf{B}(\alpha))}{\partial \alpha} = \mathbf{\Psi}(\alpha_i) - \mathbf{\Psi}(\sum_k^K \alpha_k). \tag{A.2.2}$$

$$f = \sum_i^I \sum_{g=i+1}^I A_i Z_g = \sum_i^I \sum_g^{i-1} A_g Z_i. \tag{A.2.3}$$

### A.2.1 Proof of Eq. (4.11)

**Proof** Given that

$$\begin{aligned} E_q[c_{kl}] &= \widetilde{\mu}_{c_{kl}}, \\ E_q[c_{kl}^2] &= \widetilde{\mu}_{c_{kl}}^2 + \widetilde{\lambda}_{c_{kl}}^{-1}, \end{aligned} \tag{A.2.4}$$

we have

$$\begin{aligned} \mathcal{L}_1 &= -\frac{1}{2}E_q[h_{ij}(s_{ij} - t_i^T c_k)^2] \\ &= -\frac{1}{2}h_{ij}\left\{s_{ij}^2 - 2s_{ij}t_i^T E_q[c_k] + E_q[c_k^T t_i t_i^T c_k]\right\} \\ &= -\frac{1}{2}h_{ij}[s_{ij}^2 - 2s_{ij}t_i^T\widetilde{\mu}_{c_k} + (t_i^T\widetilde{\mu}_{c_k})^2 + \sum_l^L \widetilde{\lambda}_{c_{k,l}}^{-1} t_{il}^2] \\ &= -\frac{1}{2}h_{ij}\left\{s_{ij}^2 - 2s_{ij}t_i^T\widetilde{\mu}_{c_k} + t_i^T[\widetilde{\mu}_{c_k}\widetilde{\mu}_{c_k}^T + \mathbf{\Lambda}(\widetilde{\lambda}_{c_k}^{-1})]t_i\right\} \\ &= -\frac{1}{2}h_{ij}\left\{s_{ij}^2 - 2s_{ij}t_i^T\widetilde{\mu}_{c_k} + t_i^T\rho_k t_i\right\}. \end{aligned} \tag{A.2.5}$$

■

### A.2.2 Proof of Eq. (4.15)

**Proof**

$$\begin{aligned} E_q[\log(P(x_{jd}|z_j,\psi_{kd}))] &= E_q[\log\left(\prod_k^K P(x_{jd}|\psi_{kd})^{1[z_j=k]}\right)] \\ &= E_q[\sum_k^K 1[z_j=k]\cdot\log P(x_{jd}|\psi_{kd})] \\ &= \sum_k^K \left(E_q[1[z_j=k]]\cdot E_q[\log\psi_{kd,x_{jd}}]\right) \\ &= \sum_k^K \widetilde{z}_{jk} E_q[\log\psi_{kd,x_{jd}}]. \end{aligned} \tag{A.2.6}$$

■

### A.2.3 Proof of Eq. (4.16)

**Proof**

$$\begin{aligned} E_q[\log(P(\psi_{kd}|\gamma))] &= E_q[\log\frac{1}{\mathbf{B}(\gamma)}\prod_m^M \psi_{kdm}^{\gamma-1}] \\ &= \sum_m^M(\gamma-1)E_q[\psi_{kdm}] - \log\mathbf{B}(\gamma). \end{aligned} \tag{A.2.7}$$

■

### A.2.4 Proof of Eq. (4.18)

**Proof**

$$\begin{aligned} E_q[\log(q(z_j|\widetilde{z}_j))] &= E_q[\log\sum_k^K \widetilde{z}_{jk}^{1[z_j=k]}] \\ &= \sum_k^K E_q[1[z_j=k]]\log(\widetilde{z}_{jk}) \\ &= \sum_k^K \widetilde{z}_{jk}\log(\widetilde{z}_{jk}). \end{aligned} \tag{A.2.8}$$

■

## A.3 Proof of Updating Formulas in Section 4.2.4

This appendix shows the mathematical proof of updating strategies in the optimization of our model.

### A.3.1 Proof of Eq. (4.21)

**Proof** We first extract all the terms contain $\widetilde{\theta}_k$ and get

$$\mathcal{L}\left(\widetilde{\theta}_k\right) = \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) E_q[\log(\theta_k)] + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) E_q[\log(1-\theta_k)] + \log\left(\mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2}\right). \tag{A.3.1}$$

To present the proof process more concisely, we substitute some terms with simple notations. They are:

$$\begin{gathered} E_q[\log(\theta_k)] = f_1, \quad E_q[\log(1-\theta_k)] = f_2, \\ \frac{\partial f_1}{\partial \widetilde{\theta}_{k,1}} = f_{11}, \quad \frac{\partial f_1}{\partial \widetilde{\theta}_{k,2}} = f_{12}, \quad \frac{\partial f_2}{\partial \widetilde{\theta}_{k,1}} = f_{21}, \quad \frac{\partial f_2}{\partial \widetilde{\theta}_{k,2}} = f_{22}. \end{gathered} \tag{A.3.2}$$

And we can get

$$\frac{\partial \mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2})}{\partial \widetilde{\theta}_{k,1}} = f_1, \quad \frac{\partial \mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2})}{\partial \widetilde{\theta}_{k,2}} = f_2. \tag{A.3.3}$$

Now, we can calculate the deviations of $\mathcal{L}(\widetilde{\theta}_k)$ with the above substitutional notations. By setting the deviations to zeros, we get

$$\begin{aligned} \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) f_{11} + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) f_{21} = 0, \\ \left(\sum_j^J \widetilde{z}_{jk} - \widetilde{\theta}_{k,1} + 1\right) f_{12} + \left(\sum_j^J \sum_{g=k+1}^K \widetilde{z}_{jg} - \widetilde{\theta}_{k,2} + \beta\right) f_{22} = 0. \end{aligned} \tag{A.3.4}$$

After solving the equations above, we can get the updating formulas as follows.

$$\begin{aligned}\widetilde{\theta}_{k,1} &= 1 + \sum_{j}^{J} \widetilde{z}_{jk} \\ \widetilde{\theta}_{k,2} &= \beta + \sum_{j}^{J} \sum_{g=k+1}^{K} \widetilde{z}_{jg}\end{aligned} \tag{A.3.5}$$

■

### A.3.2 Proof of Eq. (4.22)

**Proof** We first extract all the terms containing $\widetilde{\mu}_{c_k}$ and $\widetilde{\lambda}_{c_k}$:

$$\begin{aligned}\mathcal{L}(\widetilde{\mu}_{c_k}) &= \sum_{i}^{I} \sum_{j}^{J} \widetilde{z}_{jk} \left(-\frac{h_{ij}}{2}\right) \left[-2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + (t_i^T \widetilde{\mu}_{c_k})^2\right] - \frac{\lambda_c}{2} \widetilde{\mu}_{c_k}^T \widetilde{\mu}_{c_k}, \\ \mathcal{L}(\widetilde{\lambda}_{c_k}) &= \sum_{i}^{I} \sum_{j}^{J} \widetilde{z}_{jk} \left(-\frac{h_{ij}}{2}\right) \sum_{l}^{L} \widetilde{\lambda}_{c_{kl}}^{-1} t_{il}^2 - \frac{\lambda_c}{2} \sum_{l}^{L} \widetilde{\lambda}_{c_{kl}}^{-1} - \sum_{l}^{L} \frac{1}{2} \log\left(\widetilde{\lambda}_{c_{kl}}\right).\end{aligned} \tag{A.3.6}$$

Then, we calculate the deviations of Eq. (A.3.6):

$$\begin{aligned}\frac{\partial}{\partial \widetilde{\mu}_{c_k}} \mathcal{L}\left(\widetilde{\mu}_{c_k}\right) &= \sum_{i,j} \widetilde{z}_{jk} h_{ij} s_{ij} t_i^T - \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_i^T t_i + \lambda_c\right) \widetilde{\mu}_{c_k}, \\ \frac{\partial}{\partial \widetilde{\lambda}_{c_{kl}}} \mathcal{L}(\widetilde{\lambda}_{c_{kl}}) &= \frac{1}{2} \left(\left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_{il}^2 + \lambda_c\right) \widetilde{\lambda}_{c_{kl}}^{-2} - \widetilde{\lambda}_{c_{kl}}^{-1}\right).\end{aligned} \tag{A.3.7}$$

By setting the deviations above to be zeros, we can get

$$\begin{aligned}\widetilde{\mu}_{c_k} &= \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} t_i^T t_i + \lambda_c I_l\right)^{-1} \left(\sum_{i,j} \widetilde{z}_{jk} h_{ij} s_{ij} t_i^T\right) \\ &= (T \mathbf{\Lambda}(H \widetilde{z}_k) T^T + \lambda_c I_l)^{-1} (T(H \odot S) \widetilde{z}_k), \\ \widetilde{\lambda}_{c_{kl}} &= \sum_{i,j} \widetilde{z}_{jk} h_{ij} t_{il}^2 + \lambda_c, \\ \widetilde{\lambda}_{c_k} &= T \odot T H \widetilde{z}_k + \lambda_c I_l.\end{aligned} \tag{A.3.8}$$

■

### A.3.3 Proof of Eq. (4.23)

**Proof** We extract all the terms containing $\widetilde{z}_{jk}$, notice that we use the Eq. (A.2.3) to change the subscript during the extraction, then we have

$$\begin{aligned}\mathcal{L}(\widetilde{z}_{jk}) = &\sum_i \widetilde{z}_{jk}\mathcal{L}_1 + \sum_g^{k-1} \widetilde{z}_{jk} E_q[\log(1-\theta_g)] \\ &+ \widetilde{z}_{jk} E_q[\log(\theta_g)] + \sum_d \widetilde{z}_{jk} E_q[\log \psi_{kd,x_{jd}}] - \widetilde{z}_{jk}\log \widetilde{z}_{jk}.\end{aligned} \tag{A.3.9}$$

After that, we calculate the deviations of Eq. (A.3.9), which is

$$\frac{\partial}{\partial \widetilde{z}_{jk}}\mathcal{L}(\widetilde{z}_{jk}) = \sum_i \mathcal{L}_1 + \sum_g^{k-1} E_q[\log(1-\theta_g)] + E_q[\log(\theta_g)] + \sum_d E_q[\log \psi_{kd,x_{jd}}] - \log \widetilde{z}_{jk} - 1. \tag{A.3.10}$$

By setting the deviation above to zero, we get the updating formula as follows:

$$\widetilde{z}_{jk} \propto \exp\left\{ E_q[\log(\theta_k)] + \sum_g^{k-1} E_q[\log(1-\theta_g)] + \sum_i^{I} \mathcal{L}_1 + \sum_d^{D} E_q[\log \psi_{kd,x_{jd}}] \right\}. \tag{A.3.11}$$

■

### A.3.4 Proof of Eq. (4.24)

**Proof** We first extract all the terms containing $\widetilde{\psi}_{kd}$:

$$\mathcal{L}(\widetilde{\psi}_{kdm}) = \left( \sum_j^J \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm} \right) E_q[\log \psi_{kdm}] + \log \mathbf{B}(\widetilde{\psi}_{kd}). \tag{A.3.12}$$

Then, we calculate the deviation of Eq. A.3.12:

$$\frac{\partial}{\partial \widetilde{\psi}_{kdm}}\mathcal{L}(\widetilde{\psi}_{kdm}) = \left( \sum_j^J \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm} \right) \frac{\partial E_q[\log \psi_{kdm}]}{\partial \widetilde{\psi}_{kdm}} - E_q[\log \psi_{kdm}] + \frac{\partial \mathbf{B}(\widetilde{\psi}_{kd})}{\partial \widetilde{\psi}_{kdm}}. \tag{A.3.13}$$

From Eq. (A.2.1) and (A.2.2), it is easy to see that

$$\frac{\partial}{\partial \widetilde{\psi}_{kdm}}\mathcal{L}(\widetilde{\psi}_{kdm}) = \left( \sum_j^J \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma - \widetilde{\psi}_{kdm} \right) \frac{\partial E_q[\log \psi_{kdm}]}{\partial \widetilde{\psi}_{kdm}}. \tag{A.3.14}$$

Finally, we get the updating formula by setting the deviation above to be zero as follow:

$$\widetilde{\psi}_{kdm} = \sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma. \tag{A.3.15}$$

■

### A.3.5 Proof of Eq. (4.25)

**Proof** We first extract all the terms containing $t_i$:

$$\mathcal{L}(t_i) = -\frac{\lambda_t}{2}(t_i - \varphi_i)^T(t_i - \varphi_i) + \sum_{j}^{J}\sum_{k}^{K} \widetilde{z}_{jk} h_{ij} \left( -2 s_{ij} t_i^T \widetilde{\mu}_{c_k} + (t_i^T \widetilde{\mu}_{c_k})^2 + \sum_{l}^{L} \widetilde{\lambda}_{c_k}^{-1} t_{il}^2 \right). \tag{A.3.16}$$

Then, we calculate the deviation of Eq. (A.3.16) and get

$$\frac{\partial}{\partial t_i}\mathcal{L}(t_i) = -\lambda_t(t_i - \varphi_i) - \sum_{j}^{J}\sum_{k}^{K} \widetilde{z}_{jk} h_{ij} (-s_{ij}\widetilde{\mu}_{c_k} + \widetilde{\mu}_{c_k}\widetilde{\mu}_{c_k}^T t_i + \mathbf{\Lambda}(\widetilde{\lambda}_{c_k}^{-1}) t_i). \tag{A.3.17}$$

By setting the deviation above to zero, we get the updating formula

$$\begin{aligned} t_i &= \left( \sum_{j}^{J}\sum_{k}^{K} \widetilde{z}_{jk} h_{ij} s_{ij} \rho_k + \lambda_t I_l \right)^{-1} \left( \sum_{j}^{J}\sum_{k}^{K} \widetilde{z}_{jk} h_{ij} s_{ij} \widetilde{\mu}_{c_k} + \lambda_t \varphi_i \right) \\ &= \left( \rho \widetilde{Z}^T h_i + \lambda_t I_l \right)^{-1} \left( \widetilde{\mu}_c \widetilde{Z}^T (h_i \odot s_i) + \lambda_t \varphi_i \right). \end{aligned} \tag{A.3.18}$$

■