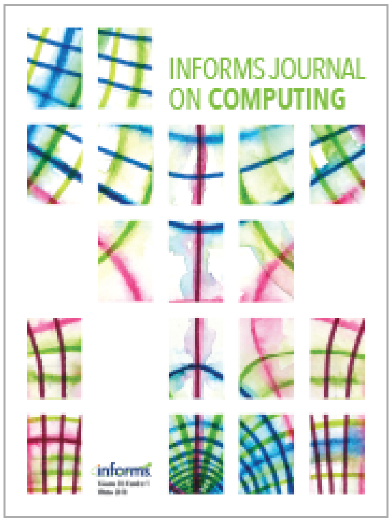


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Fine-Grained Job Salary Benchmarking with a Nonparametric Dirichlet Process-Based Latent Factor Model

Qingxin Meng, Keli Xiao, Dazhong Shen, Hengshu ZhuHui Xiong

informs
http://pubsonline.informs.org/journal/ijoc
INFORMS JOURNAL ON COMPUTING
Articles in Advance, pp. 1–21
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Fine-Grained Job Salary Benchmarking with a Nonparametric Dirichlet Process–Based Latent Factor Model

Qingxin Meng,[a] Keli Xiao,[b] Dazhong Shen,[c] Hengshu Zhu,[d,*] Hui Xiong[e,*]

[a] University of Nottingham Ningbo China, Ningbo, Zhejiang 315104, China; [b] Stony Brook University, Stony Brook, New York 11794; [c] University of Science and Technology of China, Hefei, Anhui 230052, China; [d] Baidu Talent Intelligence Center, Baidu Inc., Haidian, Beijing 100085, China; [e] AI Thrust, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
*Corresponding author

Contact: qingxin.meng@nottingham.edu.cn, https://orcid.org/0000-0002-5792-3311 (QM); keli.xiao@stonybrook.edu, https://orcid.org/0000-0001-6494-1174 (KX); sdz@mail.ustc.edu.cn (DS); zhuhengshu@baidu.com (HZ); xionghui@ust.hk (HX)

Received: May 3, 2020
Revised: July 31, 2021; December 7, 2021; January 29, 2022; February 10, 2022
Accepted: February 12, 2022
Published Online in Articles in Advance: April 18, 2022

https://doi.org/10.1287/ijoc.2022.1182

Abstract. As a key decision-making process in compensation and benefits (C&B) in human resource management, job salary benchmarking (JSB) plays an indispensable role in attracting, motivating, and retaining talent. Whereas the existing research mainly focuses on revealing the essential impacts of personal and organizational characteristics and economic factors on labor costs (e.g., C&B), few studies target optimizing JSB from a practical, data-driven perspective. Traditional approaches suffer from issues that result from using small and sparse data as well as from the limitations of linear statistical models in practice. Furthermore, there are also important technical issues that need to be addressed in the small number of machine learning–based JSB approaches, such as "cold start" issues when considering a brand-new type of company or job or model interpretability issues. To this end, we propose to address the JSB problem with data-driven techniques from a fine-grained perspective by modeling large-scale, real-world online recruitment data. Specifically, we develop a nonparametric Dirichlet process–based latent factor model (NDP-JSB) to jointly model the latent representations of both company and job position and then apply the model to predict salaries based on company and position information. Our model strengthens the usage of data-driven approaches in JSB optimization by addressing the aforementioned issues in existing models. For evaluation, extensive experiments are conducted on two large-scale, real-world data sets. Our results validate the effectiveness of the NDP-JSB and demonstrate its strength in providing interpretable salary benchmarking to benefit complex decision-making processes in talent management.
Summary of Contribution: This paper bridges the cutting-edge machine learning techniques to their implementation in a practical operation research problem in human resources. We focus on optimizing the salary-matching work to help the companies to seek reasonable salaries for their positions by proposing a data-driven approach to capture hidden patterns from user and company profiles. The contributions of this work reside in both operation research and computing. We (1) formulate the JSB optimization problem and (2) solve it by developing a data-driven method along with an effective algorithm optimization. Moreover, the proposed methodology has strengths in addressing the issues of data sparseness and result interpretability.

History: Accepted by Ram Ramesh, Area Editor for Data Science & Machine Learning.
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/ijoc.2022.1182.

Keywords: job salary benchmarking • nonparametric dirichlet process • latent factor model • talent management

1. Introduction

Job salary benchmarking (JSB) refers to the process by which organizations determine the appropriate compensation for their existing and prospective employees by acquiring and analyzing labor market data (Blankmeyer et al. 2011). As such benchmarking is an indispensable step in the compensation and benefits (C&B) process in human resource management, developing an effective JSB method would significantly benefit talent management in attracting, motivating, and retaining talent. Additionally, organizations need to have a good understanding of the labor market to attract talent by offering competitive salary packages. "Good pay" is generally considered as a determining factor leading to successful hiring (Van der Wal and Oosterbaan 2013). On the other hand, individuals can benefit from a well-established JSB system that offers precise salary analytics for target positions under complex social and economic circumstances. For example, it is common for job seekers to inquire about salaries for comparable peers or industries before requesting higher pay or deciding to job hop (Ramasamy 2015). In cases in which comparable peers are inaccessible or exiguous, job seekers may need extra help from an artificial intelligence JSB advisor.

Salary analytics draws extensive attention from researchers because of its importance in talent management. For instance, Barkema and Gomez-Mejia (1998) find that firm performance and its governance structure partially determine managerial pay. Scarpello and Jones (1996) demonstrate that the fairness of a company's compensation method strongly affects the satisfaction of that company's employees. Using principal component analysis and support vector machine methods, Lazar (2004) finds that individuals' income can be predicted based on their demographic characteristics. Studying salaries in nursing facilities, Blankmeyer et al. (2011) find evidence that the compensation of administrators is shaped by their peers. Although there are many studies focusing on investigating relationships between job salaries and various factors, such as age, race, education, performance, and industry, very few of them aim to estimate or predict job salary for practical applications.

Many human resource handbooks summarize general guidance on JSB. For example, Armstrong (2006) and Edwards et al. (2003) emphasize 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 discloses that the current salary services of the company (Kenthapadi et al. 2017a, b) rely on the salary statistics (e.g., first quartile, mean, third quartile, etc.) generated through a Bayesian normal distribution inference. The U.S. Bureau of Labor Statistics[1] also offers similar statistics by simple statistical methods. They aim to provide descriptive salary statistics within a learning unit (i.e., a job or a combination of a job and an industry/area) by obtaining salary observations in the target learning unit. However, such methods may fail to give company-customized salary suggestions as it may be hard to obtain sufficient observations within such a fine-grained learning unit. Moreover, such methods cannot address the bias issue when handling sparse data in which their salary statistic summaries may be distorted. 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 this problem, a recent related study (Meng et al. 2018) proposes to use matrix factorization (MF) techniques (Koren et al. 2009) to solve the JSB problem. This new study shows its strength in estimating company-customized salary for a job, assuming that no observation has been obtained before in a fine-grained learning unit. Meanwhile, the sparseness issue can be handled well using MF techniques. However, classic MF methods usually provide low interpretability and, thus, have very limited practical value when coming to support decision making for talent management. Moreover, this recent study may also fail in handling salary benchmarking when facing cold-start scenarios; that is, there are new job positions or companies without sufficient historical records.

As with other business intelligence (BI) systems, algorithm-based predictions combined with interpretations are important in support of a convincing decision-making procedure of human beings. Shollo and Kautz (2010) conceptualize BI into four products: data, information, knowledge, and decisions. Although data mining and artificial intelligence techniques are developed to reveal underlying patterns from huge data and are capable of handling many prediction tasks, in practice, decision makers still need to understand the insights behind the results to support real actions (Cheng et al. 2006, Steiger 2010).

One of the primary strengths of our model is its ability to offer multiaspect results related to JSB. For instance, nonparametric Dirichlet process–job salary benchmarking (NDP-JSB) can tell the required share of job skills for a job as well as companies acquiring similar talents. Such information can further direct C&B managers to analyze talent competition status in the labor market and make appropriate salary adjustments based on the prediction results. Indeed, it is critical to provide explainable insights into the prediction results for C&B managers in a way such that they have detailed and quantified salary–job patterns to support their final salary decisions.

To address these 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 analyzing a company's 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 two types of real-world recruitment data sets. 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.

The rest of the paper is organized as follows. In the next section, we show the preliminary data analysis that motivates our research. After that, we describe the technical details of our model and the inference procedure in Section 3. Section 4 reports the experimental results based on real-world data sets. Following that, comprehensive case studies showing the evidence of the possible factors that may influence the salary are presented in Section 5. In Section 6, we give a general review of the tasks involved in JSB as well as related work in technical aspects. In Section 7, we summarize the limitations of this paper. Finally, we conclude the paper in Section 8.

2. Preliminary Analysis and Problem Statement

In this section, we first discuss the motivation of this study based on preliminary data analysis and then formalize the JSB problem.

2.1. Statistical Preliminaries

Before jumping into the statistical details of our data, we first define the concept of a "learning unit."

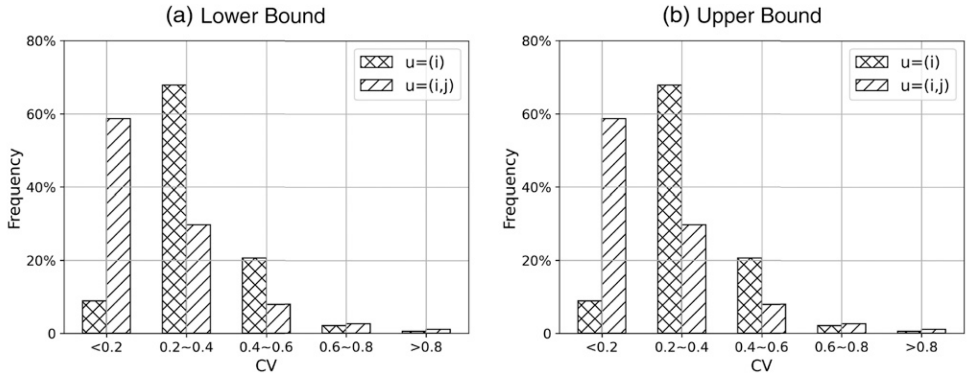
Definition 1 (Learning Unit). If job salary s is recorded with other associated factors, such as job category i, the company j that opened the job, and work location l, the combination of some or all of these factors forms a learning unit u. The learning unit defines the estimation scope of each salary.

For example, $u=(i)$ and $u=(i,j)$ are two possible learning units, where $u=(i)$ has a learning unit size equal to one and $u=(i,j)$ has a learning unit size equal to two. Given $u=(i,j)$, our estimation of the salary of job i at company j is denoted by s_{ij}. The more dimensions that are included in this learning unit, the finer the granularity of our learning scope. For instance, we can say that $u=(i,j)$ has a finer learning granularity than $u=(i)$. Most salary benchmarking applied in practice is based on job category, that is, $u=(i)$, which is due to the difficulty of obtaining comparable data from the job market. However, the job category–based JSB has limited support in the decision-making process for the C&B department, which usually needs to take the company's specific factors, such as its own compensation strategies, into consideration. Additionally, companies may build subsidiaries in different cities. Many companies choose to set up a different compensation structure for their new subsidiary because of varying living costs and the average salary level of that city. Therefore, a finer grained learning unit can provide a more reliable and valuable reference for salary benchmarking.

To demonstrate the importance of the consideration of different sizes of learning units, we compute the coefficient of variation (CV) of salaries for different sizes of learning units. The CV can be regarded as the relative variation, which is defined as the ratio of the standard deviation to the mean. The larger the value of the CV, the larger the variation of the salary within the learning unit u; hence, the salary suggestions are less reliable and valuable. Note that our data were collected from a job advertisement website in which companies only provided a job's lower and upper bounds of the salary for compensation privacy protection. Thus, for each job, we recorded the salary's lower and upper bounds and then applied the analysis and algorithm to the salary's lower and upper bounds, respectively. As shown in Figure 1, the CV distribution reaches its peak at approximately 20%–40% for $u=(i)$, which means that the job category–based JSB has a great chance (approximately 70%) of introducing a base error between 20% and 40%. The CV distribution for $u=(i,j)$ shifts to the left slightly, indicating that, if we take the company information into consideration, the JSB carries a lower base error. Similar findings appear for the cases of the lower and upper bounds.

In our framework, we set the learning unit to be $u=(i,j,l)$. Given a combination of a job and a location-specific company, our objective is to estimate its lower and upper salary bounds. Such a fine-granularity setting for salary benchmarking can support the C&B department with reliable salary suggestions by reducing the base error though it may increase the difficulty in collecting and processing external job market data. To be more concise, we denote a location-specific company as $j=(1,2,\ldots,J)$ instead of (j,l) for the rest of the paper. Figure 2 illustrates a general data-sourcing process for JSB. Company j wants to benchmark job i's

Figure 1. Histograms of the CV for Two Kinds of Learning Units $u = (i)$ and $u = (i, j)$



salary s_{ij} as indicated in the crosshatched box. Generally, the C&B department considers factors from two categories: those related to (1) internal salary policy (tendency) and (2) external or market standards. Classic survey-based techniques are based on job category–based data, as indicated with forward-slash in Figure 2. These data usually have large variance; hence, they may easily lead to inappropriate salary benchmarking suggestions. A fine-grained approach that considers both the internal and external salary information should improve the model effectiveness; however, such a fine-grained setting leads to a more severe data sparseness issue. As a result, we want to design a model that not only takes multisource factors into consideration, but also has the ability to handle the data sparseness issue through solutions, such as referring to similar companies or positions.

Thus, to make the most of the data we can obtain from the job market, we propose a solution containing two kinds of functions. First, to jointly consider the internal salary policy and the external market pricing, the JSB process can be modeled as a matrix completion task for which we use a *salary prediction module* to deal. Second, we can take full advantage of the overall external market data rather than job category–based data only by utilizing the correlations among jobs and companies. The process is similar to the strategies that are commonly used in practice: if we cannot source salary records with the same job title, we can source salary records with similar job responsibilities; additionally, if we cannot source the salary records from the given company, we can source them from comparable companies as references. These strategies are based on two assumptions: (1) if two positions have similar responsibilities, their salaries should be close; (2) if two companies share much in common, their salary policies should be similar. We propose *job* and *company representation modules* to model these two observations. In our data, each job is described by a written job description, and each company is

Figure 2. (Color online) Illustration of a Data Sourcing Process for JSB

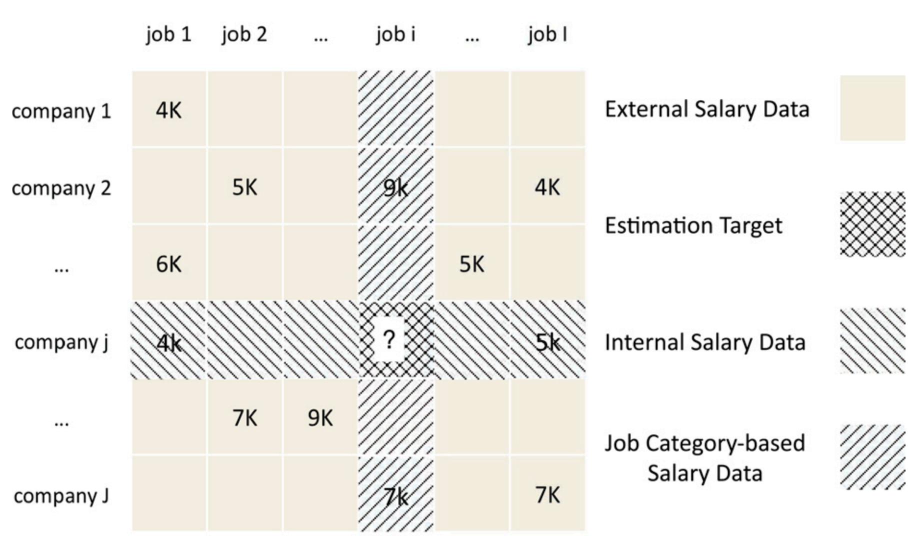
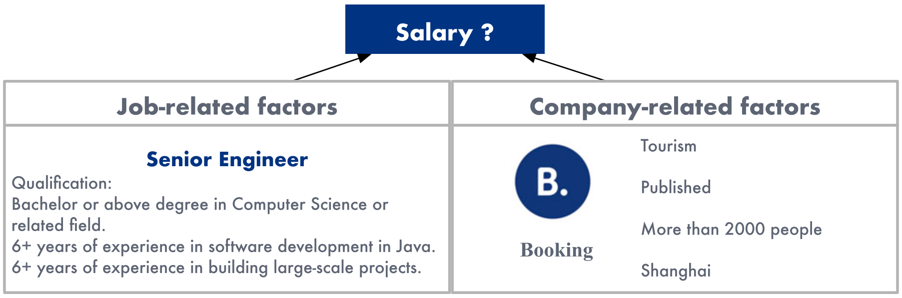


Figure 3. (Color online) An Example of Job Salary Benchmarking



described by a group of features. Those job descriptions and company features are utilized in the job and company representation modules, the technical details of which we explain in Section 3.

2.2. The Problem

Now, we formally define the JSB problem as follows.

Problem 1 (The Job Salary Benchmarking Problem). Suppose there are job positions $i = 1, 2, 3, \ldots, I$ and location-specific company $j = 1, 2, 3, \ldots, J$. Each position i can be described by a bag of words $w_{in}, n = 1, 2, \ldots, N$, where N is the total number of distinct words. Each location-specific company j can be described by a list of features $x_{jd}, d = 1, 2, \ldots, D$ (e.g., work location, industry, and financial type), where D is the total number of features. Given (i, j), a combination of position and company, our objective is to predict its salary $\hat{s}_{ij}$ so that the similarity between $\hat{s}_{ij}$ and real observation s_{ij} is maximized.

Figure 3 offers a straightforward understanding of the problem with a simple example. Booking.com is a public company operating in the tourism industry with more than 2,000 employees located in Shanghai. Suppose the company wants to hire a senior engineer and our objective is to predict the salary range for the position.

3. 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.

3.1. The Model

To address the JSB problem, we construct a Bayesian graphical probabilistic model (Koller and Friedman 2009), which includes three modules: the (1) position representation, (2) company representation, and (3) salary prediction modules. Figure 4 is the graphical representation of the proposed nonparametric Dirichlet process-based latent factor model (NDP-JSB). 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 module. 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,

$$\hat{s}_{ij} = t_i^T c_j. \tag{1}$$

Second, we use the position and company representation modules to learn t_i and c_j, respectively. Specifically, in the position representation module, we learn the topic distribution φ_i from the job descriptions through the latent Dirichlet allocation (LDA) structure (Blei et al. 2003). Job-related latent factors t_i is obtained from the normal distribution with the mean φ_i. Meanwhile, in the company representation module, we segment those companies into several clusters based on their features X by applying an NDP (Blei and Jordan 2006). 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 Equation (1) as

$$\hat{s}_{ij} = t_i^T c_{z_j}. \tag{2}$$

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 have similar latent factors.

Although different modules bare different functions, they are connected in 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 discuss the three modules in detail.

3.1.1. 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 φ_i from a Dirichlet process with the prior parameter α. Second, we assign every word w_{in} in the article to a specific topic g_{in}; the topic g_{in} is selected based on topic proportion φ_i. Finally, 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 are a known variable, the topic proportion φ for every article and the topic–word distributions ϕ are the latent factors we should learn from the model.

3.1.2. Module 2: Company Representation. In the company representation module, we consider both the company's basic features and historic salary observations. The company's historic salary observations can bear the compensation tendency information of that company. For example, compared with small firms, large companies or public corporations usually have more budgets and, hence, can offer higher salaries to seize the top talents in their interested fields, whereas 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 companies within a group share the same parameters, whereas parameters in different groups should fit the similarity relationship. We utilize an NDP to handle the segmentation job. We choose the stick-breaking view to construct an NDP (Ishwaran and James 2001). We first sample $\theta_k, k = 1, 2, \ldots, \infty$ from a beta distribution $B(1, \beta)$. Based on θ_k, we obtain a set of parameters $\pi_k, k = 1, 2, \ldots, \infty$ 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 of which the parameters are formed by $\pi_k, k = 1, 2, \ldots, \infty$. That is, $z_j \sim Multi(1; \pi_1, \pi_2, \ldots, \pi_\infty)$. Because the dimension of π is infinite, the possible group numbers, theoretically, can also be infinite. Meanwhile, we draw the company latent factors $c_k, k = 1, 2, \ldots, \infty$ from a normal distribution $N(0, \lambda_c^{-1})$ for each company group. In parallel with c_k, we draw another set of parameters $\psi_{kd}, k = 1, 2, \ldots, \infty, d = 1, 2, \ldots, D$ as the base parameters of multinomial distributions to generate company features X. That is, company feature $x_{jd} \sim Multi(1; \psi_{z_j, d})$. Based on these procedures, we ensure that companies in the same group share the same parameters, and similar company groups tend to have similar latent factors.

3.1.3. Module 3: Salary Prediction. In the salary prediction module, we follow a matrix factorization formulation. For a (position i, company j) combination, because 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} and then draw the salary values from the normal distribution, in which the mean value is $t_i^T c_{z_j}$, and the variance is h_{ij}^{-1}.

3.2. 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; α, β, λ_t, λ_c, h_{ij}, and γ are hyperparameters that need to be determined before training. Other variables $\Omega = (\varphi, G, \Phi, T, Z, \Theta, C, \Psi)$ are latent variables that need to be trained. We set the maximum group number of companies equal to K, the number of topics equal to L, and the dimension of each company feature equal to M. To get the optimal values of those variables, we maximize the maximum posterior estimation of the model. Thus, our job salary benchmarking problem can be mathematically formalized as follows:

$$\begin{aligned}
\max:\quad & \mathcal{L} = \log\left(\prod_{i,j,n,d} P(s_{ij}, x_{jd}, w_{in}, \Omega)\right) \\
& = \sum_i^I \sum_n^N \log(P(w_{in}, g_{in} \mid \varphi_i, \phi_l)) + \sum_i^I \log(P(\varphi_i \mid \alpha)) \\
& \quad + \sum_i^I \log\left(P\left(t_i \mid \varphi_i, \lambda_t^{-1}\right)\right) \\
& \quad + \sum_{i,j,d} \log\left(P\left(s_{ij}, x_{jd}, C, Z, \Psi, \Theta \mid T, \lambda_c^{-1}, \beta, \gamma, h_{ij}^{-1}\right)\right), \\
\text{s.t.}\quad & \sum_l^L \varphi_{il} = 1 \quad \forall i, \quad \sum_n^N \phi_{ln} = 1 \quad \forall l, \quad \sum_m^M \psi_{kdm} = 1 \quad \forall k, d, \\
& \varphi_{il} > 0, \quad \phi_{ln} > 0, \quad g_{in} > 0, \quad 0 < \theta_k < 1, \quad z_j > 0, \\
& \psi_{kdm} > 0.
\end{aligned} \tag{3}$$

The complete Bayesian generation process of our model is summarized in Online Appendix C, Algorithm 1.

3.3. Inference

To solve the objective function, we use the variational inference (Blei and Jordan 2006) and projection gradient descent method (Duchi et al. 2008) jointly. Because the parameters φ, G, and Φ are disconnected with parameters Z, Θ, C, Ψ in the probabilistic graph, we can solve them separately. We set $\alpha = 1$ and omit some constants. We denote the last term in Equation (3) by $\mathcal{L}_0$, which is irrelevant to φ, G, and Φ. Thus, the objective function can be rewritten as

$$\begin{aligned}
\max:\quad & \mathcal{L} \propto -\frac{\lambda_t}{2} \sum_i^I (t_i - \varphi_i)^T (t_i - \varphi_i) \\
& \quad + \sum_i^I \sum_n^N \log\left(\sum_l^L \varphi_{il} \phi_{l w_{in}}\right) + \mathcal{L}_0.
\end{aligned} \tag{4}$$

The parameters φ, G, and Φ can be solved in a similar way as suggested in Wang and Blei (2011). We extract the terms that contain φ, G, and Φ as follows and define $q(g_{in}=l)=\widetilde{g}_{inl}$. Applying Jensen's inequality, we have

$$\begin{aligned}\mathcal{L}(\varphi_i,g,\phi) &= -\frac{\lambda_t}{2}\sum_i^I (t_i-\varphi_i)^T(t_i-\varphi_i)\\ &\quad+\sum_i^I\sum_n^N \log\left(\sum_l^L \varphi_{il}\phi_{lw_{in}}\right)\\ &\geq -\frac{\lambda_t}{2}\sum_i^I (t_i-\varphi_i)^T(t_i-\varphi_i)\\ &\quad+\sum_i^I\sum_n^N\sum_l^L \widetilde{g}_{inl}(\log(\varphi_{il}\phi_{lw_{in}})-\log\widetilde{g}_{inl})\\ &=\bar{\mathcal{L}}(\varphi_i,\widetilde{g},\phi),\end{aligned} \tag{5}$$

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 $\widetilde{G}$, Φ and then set derivatives to zeros. Then, we get the updating formulas for these two parameters:

$$\widetilde{g}_{inl} \propto \varphi_{il}\phi_{lw_{in}}, \tag{6}$$

and

$$\phi_{lw} \propto \sum_i^I\sum_n^N \widetilde{g}_{inl}1[w_{in}=w]. \tag{7}$$

Different from $\widetilde{G}$ and Φ, the derivative function with respect to φ is quadratic, so we solve it by applying the projection gradient descent method (Duchi et al. 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

$$q(Z,\Theta,C,\Psi)=\prod_j^J q(z_j)\prod_k^K q(\theta_k)\prod_k^K q(c_k)\prod_k^K\prod_d^D q(\psi_{kd}), \tag{8}$$

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:

$$\begin{aligned}\mathcal{L}_0 \geq &\sum_{i,j} E_q[\log(P(s_{ij}\mid t_i,z_j,h_{ij}^{-1},C))]+\sum_j^J E_q[\log(P(z_j\mid\Theta))]\\ &+\sum_k^K E_q[\log(P(\theta_k\mid\beta))]+\sum_k^K E_q[\log(P(c_k\mid\lambda_c^{-1}))]\\ &+\sum_k^K\sum_d^D E_q[\log(P(\psi_{kd}|\gamma))]+\sum_j^J\sum_d^D E_q[\log(P(x_{jd}|z_j,\psi_{*,d}))]\\ &-E_q[\log(q(Z,C,\Theta,\Psi))].\end{aligned} \tag{9}$$

Now, we need to compute all terms in Equation (9). Here, we only show the results, whereas the mathematical details are discussed in Online Appendix A.

$$\begin{aligned}&E_{q(Z,C)}\left[\log(P(s_{ij}\mid t_i,z_j,h_{ij}^{-1},C))\right]\\ &=E_{q(Z,C)}\left[\log\left(\prod_k^K P(s_{ij}\mid t_i,c_k,h_{ij})^{1[z_j=k]}\right)\right]\\ &=\sum_k^K\left\{E_{q(z_j)}[1[z_j=k]]\cdot E_{q(c_k)}[\log(P(s_{ij}\mid t_i,c_k,h_{ij}))]\right\}\\ &=\sum_k^K\{\widetilde{z}_{jk}\cdot E_{q(c_k)}[\log(P(s_{ij}\mid t_i,c_k,h_{ij}))]\}\\ &=\sum_k^K(\widetilde{z}_{jk}\mathcal{L}_1),\end{aligned} \tag{10}$$

where

$$\mathcal{L}_1=-\frac{h_{ij}}{2}\left(s_{ij}^2-2s_{ij}t_i^T\widetilde{\mu}_{c_k}+t_i^T\rho_k t_i\right), \tag{11}$$

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.

$$\begin{aligned}E_q[\log(P(z_j\mid\Theta))]=&\sum_k^K q(z_j>k)E_q[\log(1-\theta_k)]\\ &+q(z_j=k)E_q[\log\theta_k],\end{aligned} \tag{12}$$

where

$$\begin{aligned}&q(z_j=k)=\widetilde{z}_{jk},\\ &q(z_j>k)=\sum_{g=k+1}^K \widetilde{z}_{jg},\\ &E_q[\log\theta_k]=\mathbf{\Psi}(\widetilde{\theta}_{k,1})-\mathbf{\Psi}(\widetilde{\theta}_{k,1}+\widetilde{\theta}_{k,2}),\\ &E_q[\log(1-\theta_k)]=\mathbf{\Psi}(\widetilde{\theta}_{k,2})-\mathbf{\Psi}(\widetilde{\theta}_{k,1}+\widetilde{\theta}_{k,2}).\end{aligned}$$

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

$$E_q[\log(P(\theta_k\mid\beta))]=\log(\beta)+(\beta-1)E_q[(1-\theta_k)]. \tag{13}$$

$$E_q[\log(P(c_k\mid\lambda_c^{-1}))]=\frac{L}{2}\log\left(\frac{\lambda_c}{2\pi}\right)-\frac{\lambda_c}{2}\left(\widetilde{\mu}_{c_k}^T\widetilde{\mu}_{c_k}+\sum_l^L\widetilde{\lambda}_{c_{kl}}^{-1}\right). \tag{14}$$

$$E_q[\log(P(x_{jd}\mid z_j,\psi_{*,d}))]=\sum_k^K\left(\widetilde{z}_{jk}E_q[\log\psi_{kd,x_{jd}}]\right). \tag{15}$$

$$E_q[\log(P(\psi_{kd}\mid\gamma))]=\sum_m^M(\gamma-1)E_q[\psi_{kdm}]-\log\mathbf{B}(\gamma), \tag{16}$$

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

$$E_q[\log(q(c_k\mid\widetilde{\mu}_{c_k},\widetilde{\lambda}_{c_k}))]=\frac{1}{2}\sum_l^L\log\left(\frac{\widetilde{\lambda}_{c_{kl}}}{2\pi}\right)-\frac{L}{2}. \tag{17}$$

$$E_q[\log(q(z_j \mid \widetilde{z}_{jk}))] = \sum_{k}^{K} \widetilde{z}_{jk} \log(\widetilde{z}_{jk}). \tag{18}$$

$$\begin{aligned} E_q[\log(q(\theta_k \mid \widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2}))] = & -\log \mathbf{B}(\widetilde{\theta}_{k,1}, \widetilde{\theta}_{k,2}) \\ & + (\widetilde{\theta}_{k,1} - 1) E_q[\log \theta_k] \\ & + (\widetilde{\theta}_{k,2} - 1) E_q[\log(1 - \theta_k)]. \end{aligned} \tag{19}$$

$$\begin{aligned} E_q[\log(q(\psi_{kd} \mid \widetilde{\psi}_{kd}))] = & \sum_{m}^{M} (\widetilde{\psi}_{kdm} - 1) E_q[\log \psi_{kdm}] \\ & - \log \mathbf{B}(\widetilde{\psi}_{kd}). \end{aligned} \tag{20}$$

3.4. Updating Formulas

We substitute all the terms in Equation (3) based on equations described in Section 3.3. After solving the derivatives in the optimization problem, we get the updating formulas for all corresponding terms. Note that we only show the result for each updating formula, whereas the proofs are demonstrated in Online Appendix B.

1. Updating $q(\theta_k)$,

$$\begin{aligned} \widetilde{\theta}_{k,1} &= 1 + \sum_{j}^{J} \widetilde{z}_{jk}, \\ \widetilde{\theta}_{k,2} &= \beta + \sum_{j}^{J} \sum_{g=k+1}^{K} \widetilde{z}_{jg}. \end{aligned} \tag{21}$$

2. Updating $q(c_k)$,

$$\begin{aligned} \widetilde{\mu}_{c_k} &= (T \mathbf{\Lambda}(H \widetilde{z}_k) T^T + \lambda_c I_l)^{-1} (T(H \odot S) \widetilde{z}_k), \\ \widetilde{\lambda}_{c_k} &= T \odot T H \widetilde{z}_k + \lambda_c I_l, \end{aligned} \tag{22}$$

where $\odot$ denotes the matrix Hadamard product.

3. Updating $q(z_j)$,

$$\begin{aligned} \widetilde{z}_{jk} \propto \exp\Bigg\{ & E_q[\log(\theta_k)] + \sum_{g}^{k-1} E_q[\log(1 - \theta_g)] + \sum_{i}^{I} \mathcal{L}_1 \\ & + \sum_{d}^{D} E_q[\log \psi_{kd, x_{jd}}] \Bigg\}. \end{aligned} \tag{23}$$

4. Updating $q(\psi)$,

$$\widetilde{\psi}_{kdm} = \sum_{j}^{J} \widetilde{z}_{jk} 1[x_{jd} = m] + \gamma. \tag{24}$$

5. Updating t_i,

$$\begin{aligned} t_i = & \left(\mu_c \mathbf{\Lambda}\left(\widetilde{Z}^T h_i \right) \mu_c^T + \mathbf{\Lambda}\left(\widetilde{\lambda}_c^{-1} \widetilde{Z}^T h_i \right) + \lambda_t I_l \right)^{-1} \\ & \times \left(\widetilde{\mu}_c \widetilde{Z}^T (h_i \odot s_i) + \lambda_t \varphi_i \right). \end{aligned} \tag{25}$$

Finally, the overall optimization process is demonstrated in Online Appendix C, Algorithm 2.[2]

4. Data and Experiments

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

4.1. Data

We collected data from two popular online recruiting platforms. The first platform mainly focuses on the companies in the high-tech industry, whereas the second data set primarily contains the companies in the financial industry. We name the first data set as ItDS, the second as FinDS. The time period of ItDS ranges from July 2013 to October 2015 and that of FinDS runs from January 2018 to December 2020. We selected the job postings in five major cities in China, including Beijing, Shanghai, Shenzhen, Guangzhou, and Hangzhou. To avoid noise information, we removed those companies that published job positions fewer than 20 times and the job titles that are rarely offered in the market (e.g., that appear fewer than five times in our data). We grouped and normalized the positions with similar job titles manually. Finally, the processes left us 132,061 job postings that belong to 1,795 job titles from 1,788 companies in the ItDS and 48,434 job postings that belong to 985 job titles from 1,364 companies in the FinDS. We report the basic descriptive statistics of the data in Online Appendix E, Table 3.

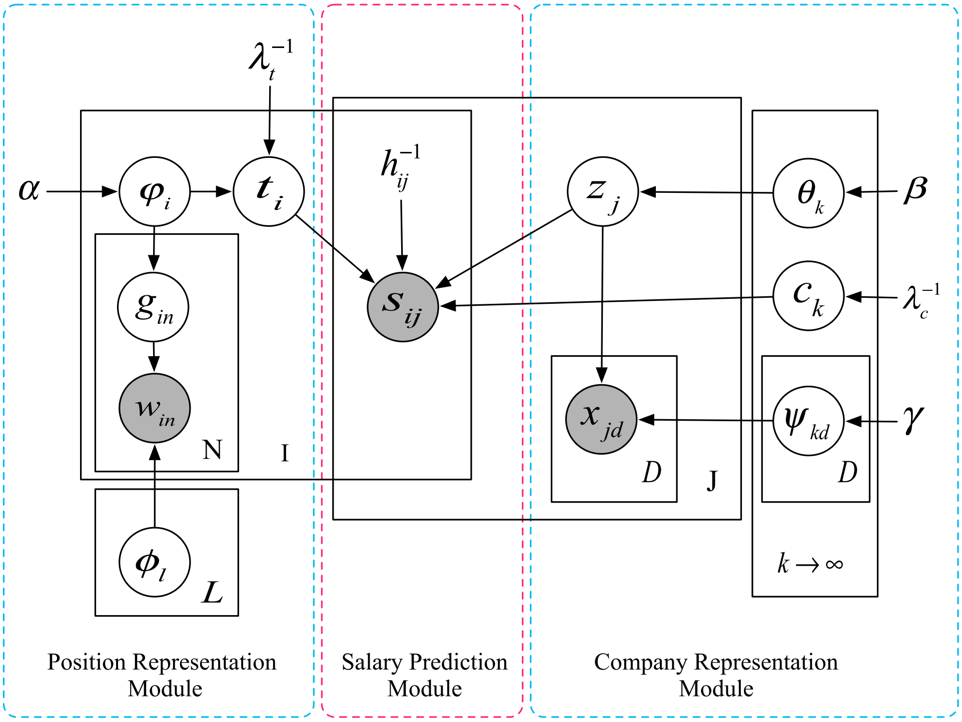
We demonstrate the salary boxplots of two data sets in Figure 5. The large outliers spread most of the places, and the boxes are skewed in the lower positions in the boxplots, indicating real salaries are distributed in a long-tail manner. Meanwhile, we used the logarithmic salary in our model to ensure the values closely follow a normal distribution (see Figure 6). 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. The probability plot for logarithmic salary in FinDS shows a similar situation. We display them in Online Appendix E, Figure 15. Similar processes can also be found in Kenthapadi et al. (2017b).

We report the detailed job- and company-related features we use in Online Appendix E, Table 4. An example of the features is also illustrated in Figure 3.

4.2. Baselines, Settings, and Evaluation Metrics

For validation, because the JSB problem is transformed as a matrix completion task, we compared our method (NDP-JSB) with six powerful MF methods and two advanced language processing neural network–based models in terms of prediction accuracy. They are (1) singular vector decomposition (SVD) (Koren et al. 2009), (2) nonnegative matrix factorization (NMF) (Luo et al. 2014), (3) probabilistic matrix factorization (PMF) (Mnih and Salakhutdinov 2008), (4) collaborative topic model (CTR) (Wang and

Figure 4. (Color online) Graphic Representation of the NDP-JSB



Blei 2011), (5) text-associated deepwalk (TADW) (Yang et al. 2015), (6) holistic salary benchmarking matrix factorization (HSBMF) (Meng et al. 2018), (7) neural network embedding BERT for JSB task (BERT-JSB) (Devlin et al. 2018), and (8) neural network embedding Word2Vec for JSB task (Word2Vec-JSB) (Le and Mikolov 2014). These MF methods are widely used in recommendation systems to address sparse prediction tasks, whereas the neural network–based methods are found effective in language processing tasks. A brief introduction of the baselines is provided in Online Appendix D.

In the experiments, we used the root mean squared error (RMSE), the mean absolute error (MAE) and Pearson relationship (PR) to evaluate each approach. Specifically, the RMSE is defined as $\text{RMSE} = \sqrt{\sum_i^N (s_i - \hat{s}_i)^2/N}$, and MAE is defined as $\text{MAE} = \sum_i^N |s_i - \hat{s}_i|/N$, where s_i is the actual value, $\hat{s}_i$ is the estimated value, and N is the total number of testing instances. PR is applied to test the relationship between predicted and true salaries. A smaller RMSE and MAE or a larger PR indicate better performance.

4.3. Overall Performance and Robustness Tests

Now, we discuss the overall performance of our model in comparison with the baselines. We followed

Figure 5. (Color online) Boxplots of the Salary Distributions

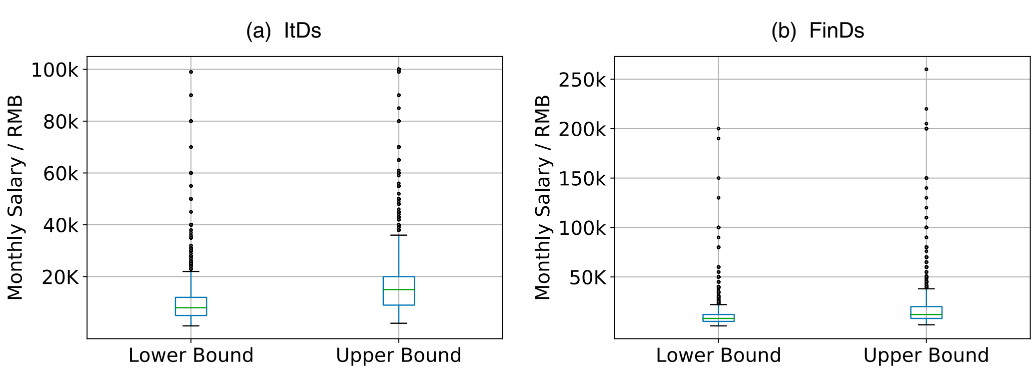
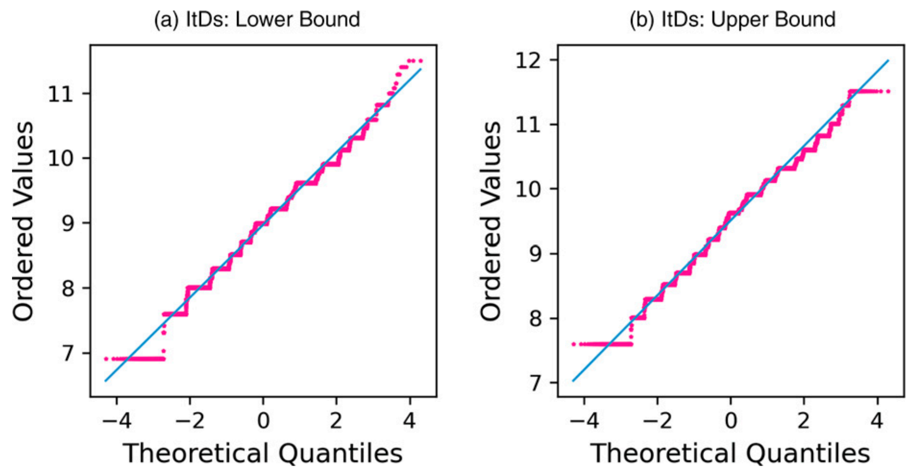


Figure 6. (Color online) Probability Plots of the Logarithmic Salaries in ItDS



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 $K = 60$ for ItDS and $K = 20$ for FinDS. Other hyperparameters were set as follows: $\lambda_t = 1$, $\lambda_c = 1$, $\alpha = 1$, $\beta = 1$, and $\gamma = 1$.

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 and randomly selected some companies or positions of which observations are less than a threshold and padded salaries within those companies or positions with mean salaries. After the imputation process, the salary matrix S includes three kinds of salary instances, namely, real, empty, and padded values. Because 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 influence brought from imputation. The precision parameter h_{ij} can be formulated as follows:

$$h_{ij} = \begin{cases} a, & \text{if the value of } s_{ij} \text{ is real,} \\ b, & \text{if the value of } s_{ij} \text{ is padded,} \\ 0, & \text{if the } s_{ij} \text{ is empty.} \end{cases} \tag{26}$$

We illustrate the function of h_{ij} here. Because s_{ij} is generated from a normal distribution with the variance h_{ij}^{-1}, the model gives 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 $a = 1$ and $b = 0.1$, and the imputation threshold was set to be five. The sampled imputation technique is widely used in the matrix factorization methods. It can not only prevent overflow/underflow problems, but also has the potential to improve the prediction accuracy as demonstrated in Ranjbar et al. (2015) and Ocepek et al. (2015).

To validate the NDP-JSB's performance, we randomly split our data set into five folds to conduct five-fold cross-validation. The overall performance of different approaches is shown in Tables 1 and 2, respectively. NDP-JSB achieves the best performance compared with all the other baselines, indicating that NDP-JSB is a powerful and robust approach in JSB tasks. We also conducted t-tests on the results between NDP-JSB against HSMBF simultaneously. The p-values for RMSE, MAE, and PR are all smaller than 0.01, demonstrating that NDP-JSB's superiority is statistically significant. However, we can see that the performance between NDP-JSB and HSBMF is close. The advantages of our approach mainly rely on the interpretations and the ability to solve the cold-start problem over HSBMF. Also, although BERT and Word2Vec are widely used and found effective in many language processing tasks (Zhang et al. 2018a, Yao et al. 2019), we find that the BERT-JSB and Word2Vec-JSB models are ineffective in the JSB tasks. There are two possible reasons as follows: (1) The JSB tasks in this paper require the identification of the interactions among job- and company-related information, not a pure language processing problem. Moreover, (2) the BERT-JSB has worse performance than Word2Vec-JSB, attributed to the increased difficulties for training with limited sparse data given the high-dimension job description representations learned by BERT.

In order to test the robustness of NDP-JSB, we held different proportions of the ItDS for testing, that is, 0.1, 0.2, 0.3, 0.4, and 0.5. The results are reported in Figure 7. We can observe that NDP-JSB has the best performance for all different testing proportions. Also, as the training proportion increases, 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 overfitting problem and lose some performance if

Table 1. The Fivefold Cross-Validation Performance for ItDS

	RMSE									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.5750	0.5887	0.5855	0.5867	0.5861	0.5746	0.5875	0.5825	0.5895	0.5812
SVD	0.5952	0.6030	0.5997	0.6023	0.6037	0.5945	0.6016	0.5965	0.6052	0.5983
NMF	0.5810	0.5967	0.5938	0.5933	0.5931	0.5825	0.5968	0.5959	0.6011	0.5880
PMF	0.6021	0.6201	0.6156	0.6083	0.6138	0.5985	0.6169	0.6112	0.6091	0.6089
CTR	0.6442	0.6565	0.6544	0.6484	0.6542	0.6415	0.6608	0.6496	0.6530	0.6514
TADW	0.5795	0.5943	0.5900	0.5919	0.5914	0.5800	0.5935	0.5880	0.5950	0.5871
HSBMF	0.5765	0.5894	0.5863	0.5877	0.5883	0.5753	0.5880	0.5836	0.5900	0.5835
BERT-JSB	0.8459	0.8873	0.9115	0.9721	0.8645	0.8295	0.9006	0.9253	0.8770	0.8677
Word2Vec-JSB	0.6686	0.714	0.6922	0.6942	0.6896	0.6844	0.7021	0.6849	0.7162	0.6818
	MAE									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.4277	0.4313	0.4293	0.4339	0.4272	0.4328	0.4333	0.4308	0.4401	0.4292
SVD	0.4379	0.4408	0.4378	0.4435	0.4364	0.4427	0.4427	0.4385	0.4485	0.4385
NMF	0.4328	0.4393	0.4363	0.4403	0.4331	0.4394	0.4427	0.4420	0.4517	0.4352
PMF	0.4479	0.4565	0.4518	0.4485	0.4496	0.4494	0.4563	0.4512	0.4534	0.4497
CTR	0.4795	0.4858	0.4776	0.4806	0.4812	0.4824	0.4923	0.4799	0.4867	0.4835
TADW	0.4318	0.4365	0.4336	0.4386	0.4325	0.4372	0.4382	0.4364	0.4452	0.4349
HSBMF	0.4292	0.4332	0.4300	0.4354	0.4284	0.4334	0.4349	0.4318	0.4406	0.4311
BERT-JSB	0.6641	0.6937	0.7154	0.7716	0.6612	0.6484	0.7044	0.7277	0.6887	0.6644
Word2Vec-JSB	0.5104	0.5419	0.522	0.528	0.5173	0.5228	0.5317	0.5158	0.5472	0.5112
	PR									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.8156	0.8113	0.8096	0.8107	0.8115	0.8275	0.8240	0.8230	0.8219	0.8266
SVD	0.8033	0.8033	0.8010	0.8017	0.8002	0.8166	0.8167	0.8150	0.8134	0.8164
NMF	0.8113	0.8054	0.8033	0.8056	0.8064	0.8223	0.8180	0.8139	0.8140	0.8221
PMF	0.7957	0.7878	0.7869	0.7945	0.7913	0.8112	0.8038	0.8030	0.8084	0.8080
CTR	0.7618	0.7583	0.7549	0.7623	0.7584	0.7794	0.7705	0.7737	0.7755	0.7762
TADW	0.8146	0.8096	0.8089	0.8093	0.8099	0.8263	0.8224	0.8219	0.8206	0.8248
HSBMF	0.8146	0.8106	0.8090	0.8100	0.8098	0.8272	0.8237	0.8223	0.8216	0.8250
BERT-JSB	0.5963	0.5841	0.5923	0.5785	0.5842	0.6016	0.5861	0.5756	0.6044	0.5849
Word2Vec-JSB	0.7443	0.7330	0.7329	0.7335	0.7338	0.7414	0.7341	0.7327	0.7268	0.7371

Notes. We also investigated the significance of the difference between the performance of NDP-JSB and the second best baseline HSBMF with respect to RMSE, MAE, and PR. All p-values from t-tests are less than 0.01, demonstrating our NDP-JSB significantly outperforms other baselines.

the training proportion is larger than 80%. Also, consistent results are found that BERT-JSB and Word2Vec-JSB fail in offering satisfying performance with varying proportional settings. The same experiments are tested on the FinDS, and similar results are obtained (see Online Appendix E, Figure 16).

4.4. Predicting a New Company

One problem of MF-based methods is their inability to deal with new company situations, which is often referenced as the cold-start problem. 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. Because of the lack of historical observations, those baselines cannot make predictions. However, our NDP-JSB can smartly take advantage of the basic features of the company and find a group to which the company may belong and then provide the estimations. Given only basic company features, the company group index can be inferred as

$$\tilde{z}_{jk} \propto \exp\left\{E_q[\log(\theta_k)] + \sum_{g}^{k-1} E_q[\log(1-\theta_g)] + \sum_{d}^{D} E_q\left[\log\psi_{kd,x_{jd}}\right]\right\}. \tag{27}$$

Based on the obtained $\tilde{z}_{jk}$, the salary can be estimated by Equation (2).

Table 2. The Fivefold Cross-Validation Performance for FinDS

	RMSE									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.5952	0.6097	0.6279	0.6244	0.6256	0.5947	0.6092	0.5990	0.6015	0.5996
SVD	0.6054	0.6160	0.6370	0.6350	0.6364	0.6001	0.6110	0.6057	0.6051	0.6038
NMF	0.6122	0.6129	0.6482	0.6403	0.6448	0.6013	0.6180	0.6098	0.6139	0.6175
PMF	0.6476	0.6569	0.6696	0.6806	0.6768	0.6299	0.6480	0.6322	0.6497	0.6380
CTR	0.7180	0.7171	0.7647	0.7834	0.7726	0.7037	0.7177	0.7121	0.7316	0.7172
TADW	0.6041	0.6206	0.6360	0.6351	0.6343	0.6035	0.6182	0.6075	0.6101	0.6111
HSBMF	0.5999	0.6129	0.6352	0.6266	0.6271	0.5993	0.6109	0.6054	0.6035	0.6017
BERT-JSB	0.8660	0.8110	0.8427	0.8600	1.3757	0.8174	0.7865	0.9282	0.9469	1.2559
Word2Vec-JSB	0.6704	0.6854	0.7091	0.7067	0.7463	0.6738	0.6838	0.7084	0.7082	0.7321
	MAE									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.4464	0.4475	0.4469	0.4478	0.4496	0.4464	0.4488	0.4466	0.4515	0.4508
SVD	0.4549	0.4552	0.4548	0.4590	0.4569	0.4487	0.4502	0.4493	0.4547	0.4513
NMF	0.4627	0.4514	0.4672	0.4610	0.4672	0.4532	0.4609	0.4586	0.4620	0.4667
PMF	0.4826	0.4808	0.4830	0.4894	0.4876	0.4741	0.4813	0.4779	0.4881	0.4825
CTR	0.5254	0.5250	0.5397	0.5456	0.5422	0.5202	0.5264	0.5232	0.5385	0.5341
TADW	0.4537	0.4571	0.4545	0.4578	0.4587	0.4537	0.4561	0.4529	0.4559	0.4588
HSBMF	0.4502	0.4512	0.4511	0.4521	0.4512	0.4492	0.4513	0.4501	0.4535	0.4506
BERT-JSB	0.7109	0.6298	0.6432	0.6614	1.1400	0.6456	0.6134	0.7244	0.7436	1.0366
Word2Vec-JSB	0.5088	0.5149	0.5175	0.5160	0.5569	0.5112	0.5157	0.5158	0.5171	0.5423
	PR									
	Lower bound					Upper bound				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
NDP-JSB	0.7936	0.7894	0.7843	0.7835	0.7916	0.7986	0.7949	0.8025	0.8008	0.8055
SVD	0.7893	0.7875	0.7796	0.7779	0.7847	0.7962	0.7950	0.7990	0.7992	0.8029
NMF	0.7782	0.7848	0.7677	0.7697	0.7773	0.7923	0.7866	0.7939	0.7906	0.7924
PMF	0.7513	0.7498	0.7499	0.7386	0.7521	0.7698	0.7649	0.7768	0.7624	0.7783
CTR	0.6868	0.6927	0.6597	0.6418	0.6622	0.7031	0.7019	0.7080	0.6886	0.7060
TADW	0.7918	0.7860	0.7819	0.7797	0.7883	0.7968	0.7926	0.8000	0.7982	0.8009
HSBMF	0.7922	0.7892	0.7803	0.7835	0.7912	0.7969	0.7955	0.7992	0.8006	0.8046
BERT-JSB	0.5536	0.6120	0.6008	0.6010	0.4584	0.5947	0.6243	0.5817	0.5689	0.5288
Word2Vec-JSB	0.7266	0.7215	0.7127	0.7130	0.7059	0.7235	0.7248	0.7134	0.7143	0.7114

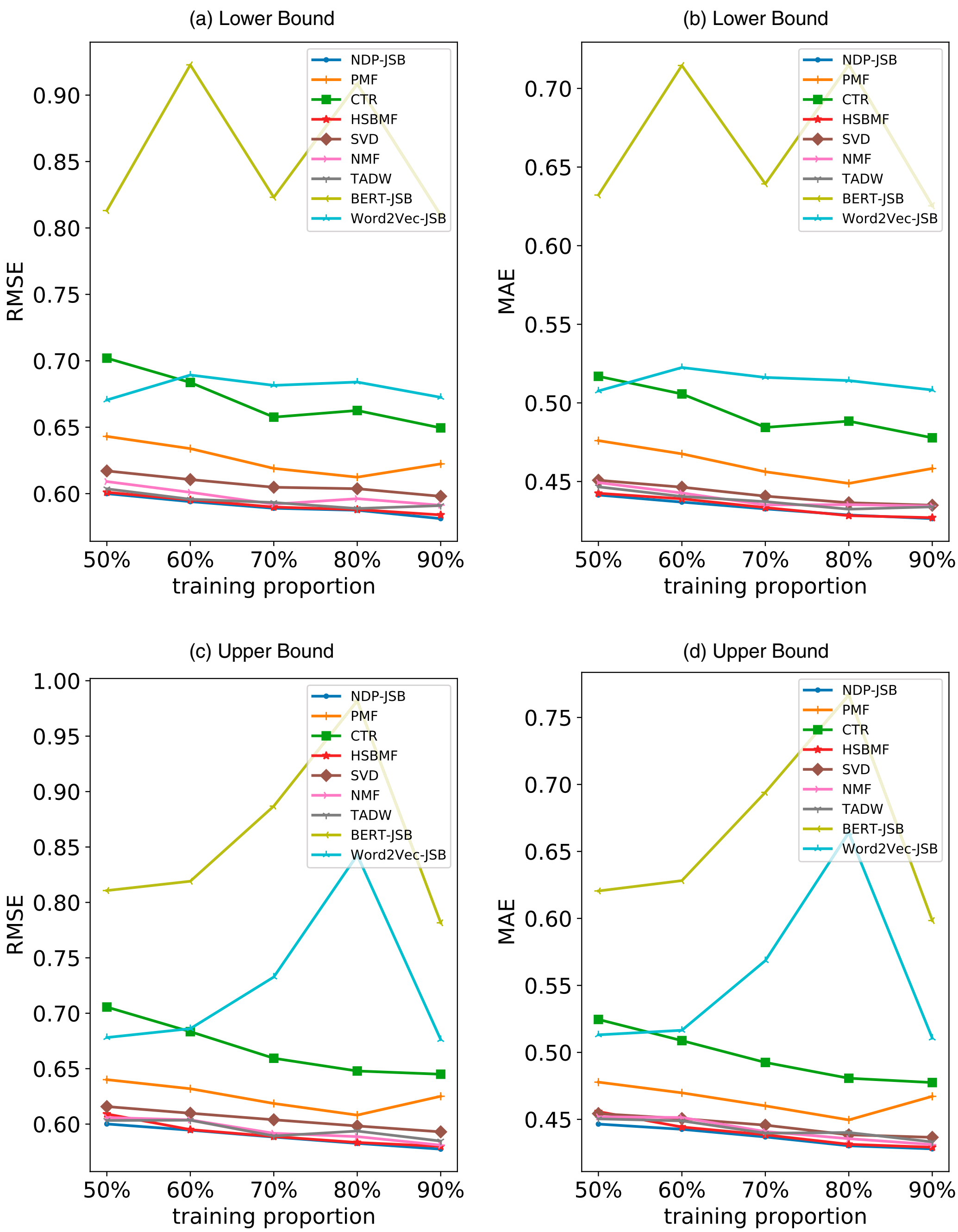
Notes. We also investigated the significance of the difference between the performance of NDP-JSB and the second best baseline HSBMF with respect to RMSE, MAE, and PR. All p-values from t-tests are less than 0.01, demonstrating our NDP-JSB significantly outperforms other baselines.

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 ItDS. We compared the performance with the item-based collaborative filtering (CF) method (Sarwar et al. 2001), 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 are presented in Figure 8, in which we can see that NDP-JSB outperforms CF as we expected. Moreover, the p-values from the t-test are 1.88×10^{-6} and 1.70×10^{-5} for RMSE and MAE, respectively, demonstrating that 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.

4.5. Predicting a New Position

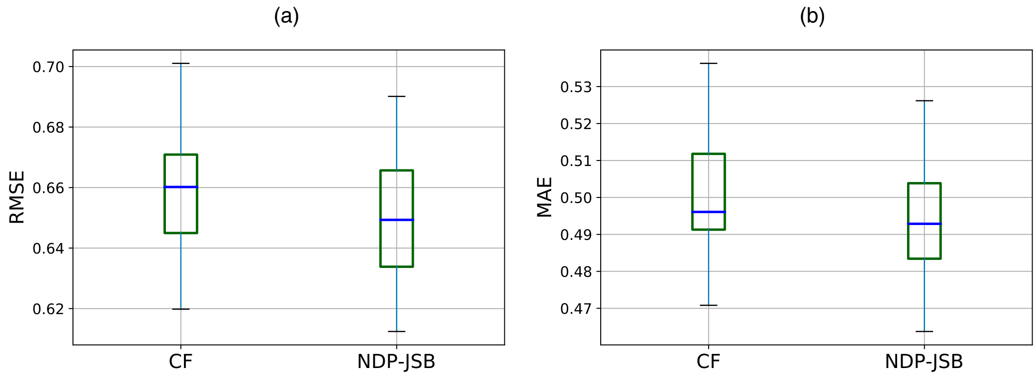
C&B managers may also have the demand to estimate a new job's salary that has never appeared in the job market before. However, this is not an easy task in some situations. For instance, if a job never appears in the job market before, it may indicate a shortage of specific skills related to the position in the labor

Figure 7. (Color online) Robust Testing Results for the Different Splitting Proportions with ItDS



market. The companies need to train the employer to accommodate this job by themselves. Then, salary is a negotiation result from the unique talents' availability and the company's eagerness for the new type of talent. So the salary is usually decided case by case. Another situation is that models can identify similar jobs based on their job descriptions and then make salary predictions through those similar jobs. However, only relying on similar job descriptions may lead to deviations in salary estimation. For instance, the job descriptions for "software developer" and "senior software developer" are quite alike, but their base salaries might have a large distance. The algorithm-based methods can successfully discern the patterns of salary volatility from the job-related information but may fail to capture the base salary for a new job. Nonetheless, we still conducted experiments to predict salary for a new position by NPD-JSB and compared the results with the CTR model. They are two probabilistic models that can use job descriptions for

Figure 8. (Color online) Boxplots of Results for Predicting New Companies



new job salary predictions. The results clearly show that NDP-JSB can predict the salary for a new job more effectively.

Specifically, we randomly selected 10% of instances of which the job positions are new to our data set and repeated the experiments 10 times. We report the comparisons of RMSE and MAE between the NDP-JSB and CTR in Figure 9, and the two main findings are summarized as follows: (1) The performance of the two models decreased significantly as we expected, indicating that an algorithm-based salary for a new job needs to be considered jointly with human judgment. (2) The NDP-JSB shows better performance than the CTR in terms of a new job salary prediction. We ran the *t*-tests on the predictions. The *p*-values are 0.001 and 0.0005 for RMSE and MAE, significantly supporting NDP-JSB's prosperity.

5. Case Studies

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

5.1. Position Grouping

In the position representation module, each position is represented by five latent topics. We plot the *t*-distributed stochastic neighbor embedding (t-SNE) in Figure 10(f), in which the five-dimension job representative vectors are transformed into a two-dimensional space. Five clusters (shown in different colors) can be identified by the job representation module. As can be seen, different clusters are located in different places in Figure 10(f), demonstrating that our job representation module is capable of identifying useful clusters. To facilitate the understanding of the characteristics of these five clusters, we took the top eight keywords in each topic and demonstrate them in Figure 10, (a)–(e). As can be seen, the keywords are skill sets emphasized by different types of professionals. The grouping results are similar to human job categorization practicals, 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 11, we can observe that the technical jobs (i.e., front-end, back-end, and testing) have relatively better compensation. Also, although promotion jobs may have relatively lower salaries, their variation range is the largest,

Figure 9. (Color online) Boxplots of Results for Predicting New Jobs

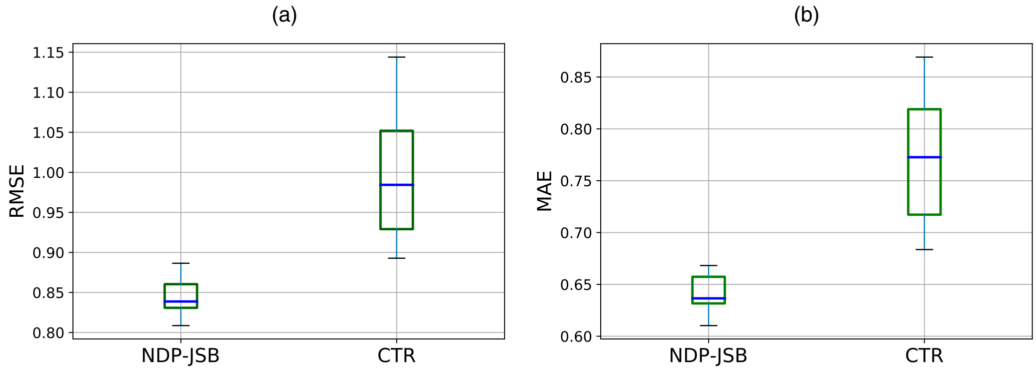
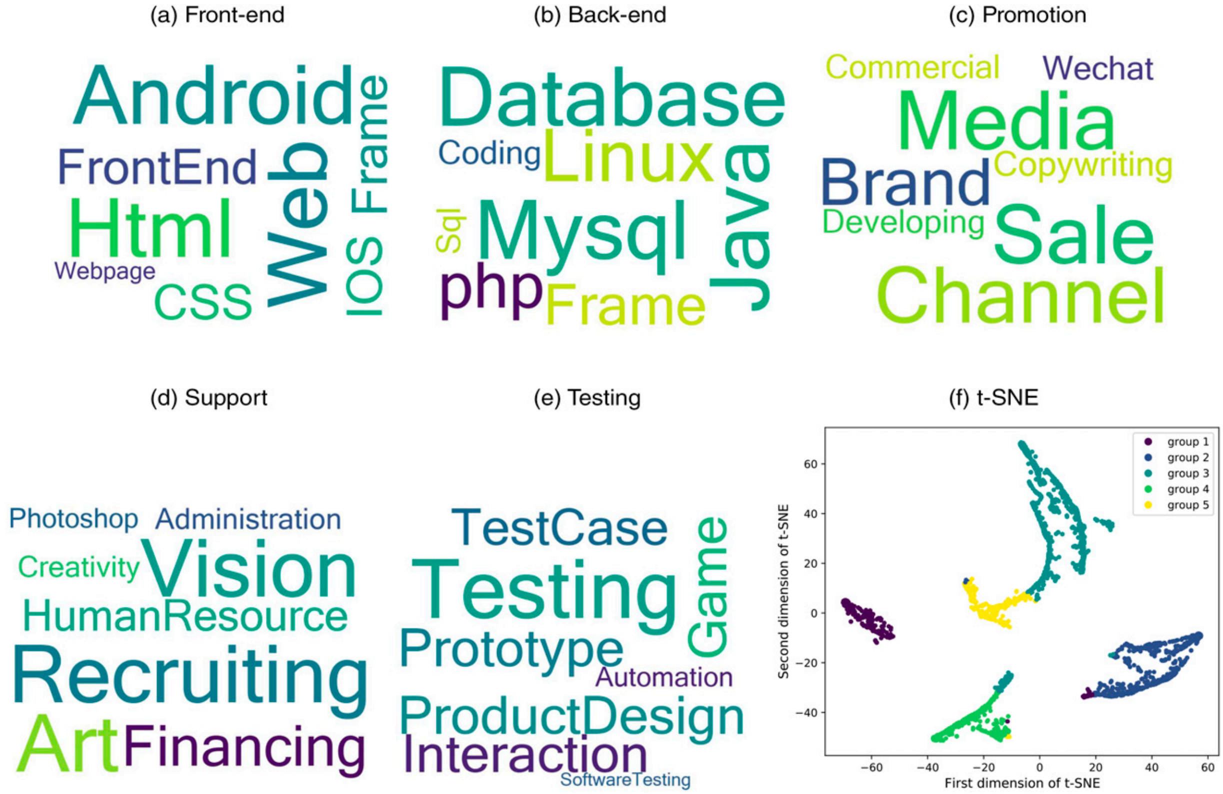


Figure 10. (Color online) Word Clouds and t-SNE for the Five Job Groups



suggesting top promotion people have high potential to earn.

5.2. 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 cities 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 that belong to a company should have different grouping results. We display the grouping results in Figure 12, in which each block represents a location-specific company, and a different shape filled represents a group ID. The 15 branches are classified into three main groups, and

Figure 11. (Color online) Salary Distributions for the Five Job Groups

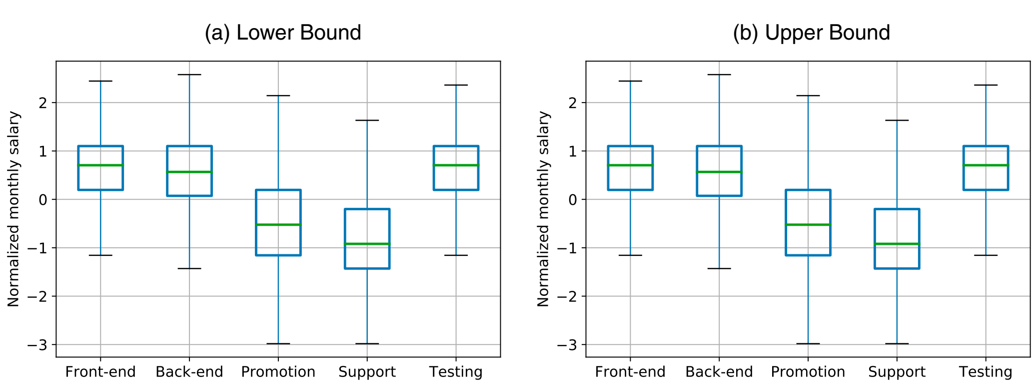
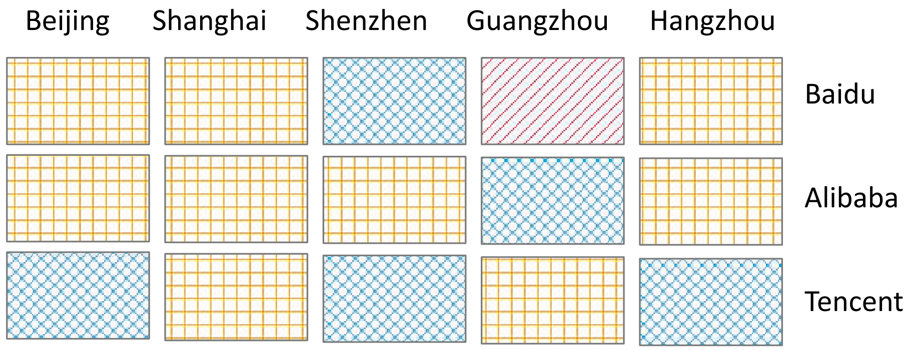


Figure 12. (Color online) Grouping Results for Three Famous Companies



each company has two to three classes across the five cities. 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 finds the optimal group number on the whole. We display the clustering results for all companies and compare them with another commonly used clustering method, *K*-modes (Chaturvedi et al. 2001) in Figure 13. 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 lie in the same row with the same color. The points in Figure 13(a) are more compacted than Figure 13(b). NDP-JSB can intelligently figure out the optimal group number is less than 60, whereas *K*-modes is incapable of deciding the reasonable group number by itself.

5.3. 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 as well as other similar companies in the job market. Those similar companies can be used for further data sourcing and competition analysis. Figure 14 shows an example of job profiling, which is a real case in our data set. Alibaba (Hangzhou) wanted to hire a Java engineer in the job market. Learning from the NDP-JSB, Java engineer emphasizes the professionals of the back-end for around 85% and front-end for around 15%. The competitive companies in the job market include Taobao (Beijing), Yibao Pay (Shanghai), and Sina Weibo (Hangzhou).

6. 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.

6.1. Job Salary Benchmarking

Salary estimation has drawn much attention from human resource management because of its key role in attracting, motivating, and 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, Correll et al. 2007, Jerrim 2015, Hamlen and Hamlen 2016). Frydman and Jenter (2010), Gong and Li (2013), and Brick et al. (2006) try to understand what determines

Figure 13. (Color online) Grouping Results for All Companies

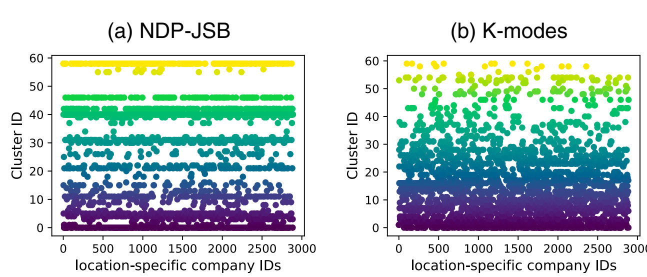
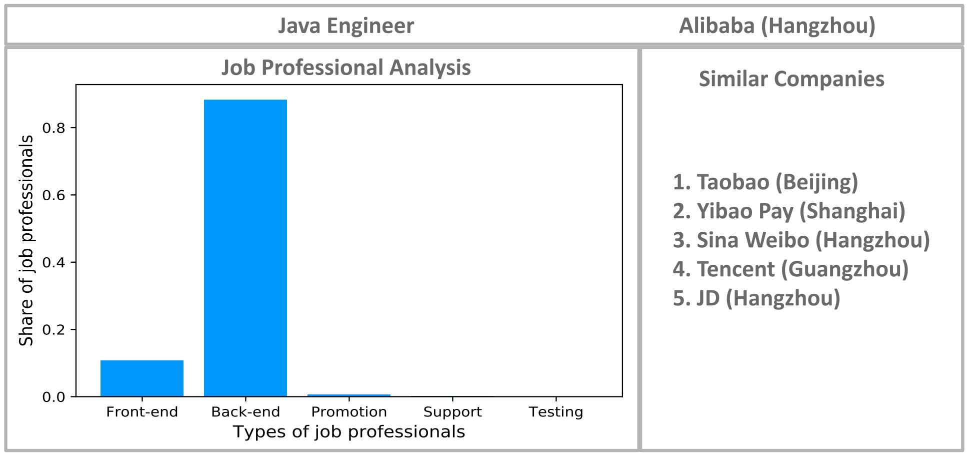


Figure 14. (Color online) An Example of Job Profiling



the high revenues of CEOs, whereas Peng and Röell (2008, 2014) discover indications that CEOs intend to raise their revenues through managerial manipulations. There are also a large number of studies emphasizing pay equity (Berkowitz et al. 1987, Scarpello and Jones 1996, Terpstra and Honoree 2003, Chang and Hahn 2006). Still other researchers investigate the ways compensation is shaped by peer comparative organizations and individuals (Faulkender and Yang 2010, Blankmeyer et al. 2011). Ferris et al. (2001) finds that excellent social skills and related general mental ability serve as strong explanations for individuals' job performance and salary levels. Khongchai and Songmuang (2016a, b) predict students' income by examining their demographic features and state 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 and Scarpello 2002).

The existing work mainly focuses on understanding the determinants of the salary range, whereas how to benchmark salary by jointly considering internal compensation policies and external market pricing from the C&B department's perspective is not well-addressed. As a widely applied process in practice, some human resource handbooks (Edwards et al. 2003, Armstrong) provide 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) proposes a framework for company profiling that can simultaneously predict job salary; however, their framework is based on a data set 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 start-ups.

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.

6.2. 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; the nodes in the graph represent variables (observable or unobservable), and the edges represent the interactions between them (Koller and Friedman 2009). Because of 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 MF method, the topic model, and the nonparametric Dirichlet process. In the following, we 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 because of its effectiveness in "guessing" the missing values in S by the cross-product procedure (Adomavicius and 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 early imputation technique may distort the actual distribution and easily lead to overfitting (Kim and 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) suggests using regularizers to address the overfitting issue by constraining the values in A and B. After that, Koren (2008) proposes a method of integrating the implicit neighborhood 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 nonnegative, demonstrating that the constraints are able to learn the parts-based representations (Lee and Seung 1999, 2001). PMF places zero-mean spherical Gaussian priors on matrices A and B (Mnih and Salakhutdinov 2008). In our framework, we adopt the PMF structure in the prediction module because it belongs to the probabilistic graphical model and is easy to extend in a more complicated graphic structure.

An early developed topic model named probabilistic latent semantic indexing (pLSI) (Hofmann 1999) is a probabilistic model with three layers. The first layer generates documents, the second layer generates topics of each document, and the last layer describes the word selection process–based topic-word occurrence frequency distributions. Later, Blei et al. (2003) propose the famous 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 and Blei (2011) incorporate LDA into an MF framework for scientific article recommendations.

Additionally, LDA models are implemented broadly in the areas of text mining, document classification (Chen et al. 2015, Pavlinek and Podgorelec 2017, Wan et al. 2020), image recognition (Rasiwasia and Vasconcelos 2013, Gomez et al. 2017), and brand management (Tirunillai and Tellis 2014, Guo et al. 2017). In our model, we adopt 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 because of 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) and Antoniak (1974) provide an answer by proposing the NDP. The word "nonparametric" 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 and James (2001) described it with a stick-breaking view, and the development of Gibbs and Monte Carlo Markov chain sampling methods enabled it to be solved in an approximate way (Neal 2000, Ishwaran and James 2001). Afterward, Blei and Jordan (2006) propose a variational inference (VI) technique to solve the algorithms that can mitigate the computational complexity caused by sampling methods. The NDP is widely applied in machine learning tasks, especially for density estimation and clustering (Escobar and West 1995, Teh et al. 2005, Zhang et al. 2005, Dahl 2006, Xue et al. 2007, Nguyen et al. 2016, Zhang et al. 2018b). 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 predefining the number of mixtures. We adopt an NDP structure to learn the latent company representations and the VI technique to solve the algorithms efficiently.

7. Limitations

Our proposed framework has limitations regarding the three aspects. First, our model's performance is affected by the data size. If the data set size is as small, such as containing hundreds of job postings, and every job or company only has a few observations, the performance decrease a lot. The same effects apply to other baselines as well. The phenomenon is due to model complexity. The NDP-JSB is a relatively complicated model with large amount parameters. The increase of learnable parameters amount in a model demands increased training data size as the data-driven machine learning model relies on the data to learn meaningful and useful data patterns (L'heureux et al. 2017). Adding regularizers, such as Lasso and Ridge regularizers, are commonly used techniques to control the model complexity and reduce the generalization error (Bühlmann and Van De Geer 2011). As a probabilistic model, the regularizers are naturally embedded in our model because each parameter is constrained by its prior and distribution assumption. For instance, in our approach, we assume the prediction salary is normally distributed, and the variance is h_{ij}^{-1}. So tuning h_{ij}^{-1} is equivalent to a salary bound within a specific range, leading to a similar effect to Lasso and Ridge regularizers. In our future work, we will explore more priors and possible regularization methods to improve the performance. Second, we chose to use the grid-search method to tune the hyperparameters in our model, and the searching space of the best hyperparameter combinations is proportional to the exponential function (Andonie 2019), which took us a lot of time. On the other hand, as long as the hyperparameters are well set, the NDP-JSB takes fewer training epochs to reach the best performance. That is because variation inference optimization is an efficient method to learn complicated probabilistic graphical models. Finally, the probabilistic graphic-based model such as NDP-JSB is hard to extend or change structures. Adding extra modules or changing the structures slightly might cause the inference to be quite different. The inference formulas need to be derived from the beginning, and so does the hyperparameter tuning process (Wainwright and Jordan 2008).

8. Conclusions

In this paper, we address the JSB problem from a more fine-grained and data-driven perspective by modeling large-scale, real-world online recruitment data. Specifically, we design a nonparametric 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 two real-world data sets. The experimental results clearly validate the effectiveness of the NDP-JSB in terms of salary prediction and also demonstrate 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.

Endnotes

[1] See https://www.bls.gov/bls/blswage.htm.

[2] Readers can access the source code of this algorithm, the data sets, and the appendix file via the link https://github.com/qingxin-meng/NDP-JSB.

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge Data Engrg.* 17(6):734–749.

Andonie R (2019) Hyperparameter optimization in learning systems. *J. Membrane Comput.* 1(4):279–291.

Antoniak CE (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.* 2(6): 1152–1174.

Armstrong M (2006) *A Handbook of Human Resource Management Practice* (Kogan Page Publishers, London).

Barkema HG, Gomez-Mejia LR (1998) Managerial compensation and firm performance: A general research framework. *Acad. Management J.* 41(2):135–145.

Bergmann T, Scarpello V (2002) *Compensation Decision Making* (South-Western Co., Mason, OH).

Berkowitz L, Fraser C, Treasure FP, Cochran S (1987) Pay, equity, job gratifications, and comparisons in pay satisfaction. *J. Appl. Psych.* 72(4):544–551.

Blankmeyer E, LeSage JP, Stutzman J, Knox KJ, Pace RK (2011) Peer-group dependence in salary benchmarking: A statistical model. *Managerial Decision Econom.* 32(2):91–104.

Blei DM, Jordan MI (2006) Variational inference for Dirichlet process mixtures. *Bayesian Anal.* 1(1):121–143.

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J. Machine Learn. Res.* 3(Jan):993–1022.

Brick IE, Palmon O, Wald JK (2006) CEO compensation, director compensation, and firm performance: Evidence of cronyism? *J. Corporate Finance* 12(3):403–423.

Bühlmann P, Van De Geer S (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications* (Springer Science & Business Media, Berlin).

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

Chaturvedi A, Green PE, Caroll JD (2001) K-modes clustering. *J. Classification* 18(1):35–55.

Chen X, Xia Y, Jin P, Carroll J (2015) Dataless text classification with descriptive LDA. Bonet B, Koenig S, eds. *Proc. 29th AAAI Conf. Artificial Intelligence*, (AAAI Press, Austin, TX), 2224–2231.

Cheng S, Dai R, Xu W, Shi Y (2006) Research on data mining and knowledge management and its applications in China's economic development: Significance and trend. *Internat. J. Inform. Tech. Decision Making* 5(4):585–596.

Correll SJ, Benard S, Paik I (2007) Getting a job: Is there a motherhood penalty? *Amer. J. Sociol.* 112(5):1297–1338.

Dahl DB (2006) Model-based clustering for expression data via a Dirichlet process mixture model. Do K, Müller P, Vannucci M, eds. *Bayesian Inference for Gene Expression and Proteomics* (Cambridge University Press, Cambridge, UK), 201–218.

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint, submitted October 11, https://arxiv.org/abs/1810.04805.

Duchi J, Shalev-Shwartz S, Singer Y, Chandra T (2008) Efficient projections onto the l 1-ball for learning in high dimensions. Zhang Z, ed. *Proc. 25th Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York, NY), 272–279.

Edwards JE, Scott JC, Raju NS (2003) *The Human Resources Program-Evaluation Handbook* (SAGE Publications, Inc., Thousand Oaks, CA).

Escobar MD, West M (1995) Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.* 90(430):577–588.

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

Ferguson TS (1973) A Bayesian analysis of some nonparametric problems. *Ann. Statist.* 1(2):209–230.

Ferris GR, Witt LA, Hochwarter WA (2001) Interaction of social skill and general mental ability on job performance and salary. *J. Appl. Psych.* 86(6):1075–1082.

Frydman C, Jenter D (2010) CEO compensation. *Annual Rev. Financial Econom.* 2(1):75–102.

Ghahramani Z (2015) Probabilistic machine learning and artificial intelligence. *Nature* 521(7553):452–459.

Gomez L, Patel Y, Rusiñol M, Karatzas D, Jawahar C (2017) Self-supervised learning of visual features through embedding images into text topic spaces. Bertino E, Chen Y, eds. *Proc. IEEE Conf. Comput. Vision Pattern Recognition* (IEEE Computer Society, Los Alamitos, CA), 4230–4239.

Gong JJ, Li S (2013) CEO incentives and earnings prediction. *Rev. Quant. Finance Accounting* 40(4):647–674.

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

Hamlen KR, Hamlen WA (2016) Faculty salary as a predictor of student outgoing salaries from MBA programs. *J. Ed. Bus.* 91(1):38–44.

Hofmann T (1999) Probabilistic latent semantic indexing. Lang J, ed. *Sigir '99: Proc. 22nd Annual Internat. ACM SIGIR Conf. Res. Development in Informat. Retrieval* (Association for Computing Machinery, New York, NY), 50–57.

Ishwaran H, James LF (2001) Gibbs sampling methods for stick-breaking priors. *J. Amer. Statist. Assoc.* 96(453):161–173.

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

Kenthapadi K, Chudhary A, Ambler S (2017a) LinkedIn salary: A system for secure collection and presentation of structured compensation insights to job seekers. Bertino E, Chen Y, eds. *2017 IEEE Sympos. Privacy-Aware Comput.* (IEEE, Computer Society, Los Alamitos, CA), 13–24.

Kenthapadi K, Ambler S, Zhang L, Agarwal D (2017b) Bringing salary transparency to the world: Computing robust compensation insights via LinkedIn salary. Lim E-P, Winslett M, eds. *Proc. 2017 ACM Conf. Inform. Knowledge Management* (ACM, New York, NY), 447–455.

Khongchai P, Songmuang P (2016a) Implement of salary prediction system to improve student motivation using data mining technique. Gaol FL, Theeramunkong T, Papadopo GA, eds. *2016 11th Internat. Conf. Knowledge Inform. Creativity Support Systems* (IEEE, Yogyakarta, Indonesia), 1–6.

Khongchai P, Songmuang P (2016b) Random forest for salary prediction system to improve students' motivation. Manfredi G, Pietro G, eds. *2016 12th Internat. Conf. Signal-Image Tech. Internet-Based Systems* (IEEE, Los Alamitos, CA), 637–642.

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

Koller D, Friedman N (2009) *Probabilistic Graphical Models: Principles and Techniques* (MIT Press, Cambridge, MA).

Koren Y (2008) Factorization meets the neighborhood: A multifaceted collaborative filtering model. Li Y, Liu B, eds. *Proc. 14th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining*, (ACM, New York, NY), 426–434.

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Comput.* 42(8):30–37.

Lazar A (2004) Income prediction via support vector machine. Tourassi GD, ed. *Internat. Conf. Machine Learning and Applications*, (ICMLA), 143–149.

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. Xing EP, Jebara T, eds. *Internat. Conf. Machine Learn.* (PMLR), 1188–1196.

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791.

Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. *Adv. Neural Inform. Processing Systems* 13: 556–562.

L'heureux A, Grolinger K, Elyamany HF, Capretz MA (2017) Machine learning with big data: Challenges and approaches. *IEEE Access* 5:7776–7797.

Lin H, Zhu H, Zuo Y, Zhu C, Wu J, Xiong H (2017) Collaborative company profiling: Insights from an employee's perspective. Singh S, Markovitch S, eds. *31st AAAI Conf. Artificial Intelligence* (AAAI Press, San Francisco, CA), 1417–1423.

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

Meng Q, Zhu H, Xiao K, Xiong H (2018) Intelligent salary benchmarking for talent recruitment: A holistic matrix factorization approach. Zhu F, Yu J, eds. *2018 IEEE Internat. Conf. Data Mining* (IEEE Computer Society, Los Alamitos, CA), 337–346.

Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. Platt JC, Koller D, Singer Y, Rowies ST, eds. *Proc. 20th Internat. Conf. Adv. Neural Inform. Processing Systems NIPS'07* (Curran Associates Inc., Red Hook, NY), 20:1257–1264.

Neal RM (2000) Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graphical Statist.* 9(2): 249–265.

Nguyen V, Gupta S, Rana S, Li C, Venkatesh S (2016) A Bayesian nonparametric approach for multi-label classification. Durrant RJ, Kim K, eds. *Asian Conf. Machine Learn.*, vol. 63 (University of Waikato, Hamilton, New Zealand), 254–269.

Ocepek U, Rugelj J, Bosnić Z (2015) Improving matrix factorization recommendations for examples in cold start. *Expert Systems Appl.* 42(19):6784–6794.

Paterek A (2007) Improving regularized singular value decomposition for collaborative filtering. Liu B, ed. *Proc. KDD Cup and Workshop* (New York, NY), 5–8.

Pavlinek M, Podgorelec V (2017) Text classification method based on self-training and LDA topic models. *Expert Systems Appl.* 80: 83–93.

Peng L, Röell A (2008) Manipulation and equity-based compensation. *Amer. Econom. Rev.* 98(2):285–290.

Peng L, Röell A (2014) Managerial incentives and stock price manipulation. *J. Finance* 69(2):487–526.

Ramasamy R (2015) The production of salary profiles of ICT professionals: Moving from structured database to big data analytics. *Statist. J. IAOS* 31(2):177–191.

Ranjbar M, Moradi P, Azami M, Jalili M (2015) An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. *Engrg. Appl. Artificial Intelligence* 46:58–66.

Rasiwasia N, Vasconcelos N (2013) Latent Dirichlet allocation models for image classification. *IEEE Trans. Pattern Anal. Machine Intelligence* 35(11):2665–2679.

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. Shen VY, Saito N, eds. *Proc. 10th Internat. Conf. World Wide Web* (Association for Computing Machinery, New York, NY), 285–295.

Scarpello V, Jones FF (1996) Why justice matters in compensation decision making. *J. Organ. Behav.* 17(3):285–299.

Shollo A, Kautz K (2010) Toward an understanding of business intelligence. Green P, Rosemann M, eds. *Proc. 21st Australasian Conf. Inform. Systems*. (ACIS 2010), Brisbane, Australia.

Steiger DM (2010) Decision support as knowledge creation: A business intelligence design theory. *Internat. J. Bus. Intelligence Res.* 1(1):29–47.

Teh YW, Jordan MI, Beal MJ, Blei DM (2005) Sharing clusters among related groups: Hierarchical Dirichlet processes. Saul L, Weiss Y, Bottou L, eds. *Adv. Neural Inform. Processing Systems* (MIT Press, Cambridge, MA), 17:1385–1392.

Terpstra DE, Honoree AL (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 Rev.* 35(6):67–74.

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

Van der Wal Z, Oosterbaan A (2013) Government or business? Identifying determinants of MPA and MBA students' career preferences. *Public Personnel Management* 42(2):239–258.

Wainwright MJ, Jordan MI (2008) *Graphical Models, Exponential Families, and Variational Inference* (Now Publishers, Inc., Delft, Netherlands).

Wan C, Peng Y, Xiao K, Liu X, Jiang T, Liu D (2020) An association-constrained LDA model for joint extraction of product aspects and opinions. *Inform. Sci.* 519:243–259.

Wang C, Blei DM (2011) Collaborative topic modeling for recommending scientific articles. Apte C, Ghosh J, eds. *Proc. 17th*

ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining (ACM, New York, NY), 448–456.

Xue Y, Liao X, Carin L, Krishnapuram B (2007) Multi-task learning for classification with Dirichlet process priors. *J. Machine Learn. Res.* 8:35–63.

Yang C, Liu Z, Zhao D, Sun M, Chang E (2015) Network representation learning with rich text information. Yang Q, Wooldridge M, eds. *24th Internat. Joint Conf. Artificial Intelligence* (AAAI Press, San Francisco, CA), 2111–2117.

Yao S, Yu D, Xiao K (2019) Enhancing domain word embedding via latent semantic imputation. *Proc. 25th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York, NY) 557–565.

Zhang J, Ghahramani Z, Yang Y (2005) A probabilistic model for online document clustering with application to novelty detection. Saul L, Weiss Y, Bottou L, eds. *Adv. Neural Inform. Processing Systems* (MIT Press, Cambridge, MA), 17:1617–1624.

Zhang X, Li W, Nguyen V, Zhuang F, Xiong H, Lu S (2018b) Label-sensitive task grouping by Bayesian nonparametric approach for multi-task multi-label learning. Lang J, ed. *Proc. 27th Internat. Joint Conf. Artificial Intelligence IJCAI* (AAAI Press, San Francisco, CA), 3125–3131.

Zhang L, Xiao K, Zhu H, Liu C, Yang J, Jin B (2018a) Caden: A context-aware deep embedding network for financial opinions mining. Zhu F, Yu J, eds. *2018 IEEE Internat. Conf. Data Mining (IEEE* Computer Society, Los Alamitos, CA), 757–766.