Intelligent Salary Benchmarking for Talent Recruitment: A Holistic Matrix Factorization Approach

Qingxin Meng[1], Hengshu Zhu[2,*], Keli Xiao[5], Hui Xiong[2,3,4,*]
[1]Rutgers-the State University of New Jersey, [2]Baidu Talent Intelligence Center,
[3]Baidu Business Intelligence Lab, [4]University of Science and Technology of China, [5]Stony Brook University
qm24@rutgers.edu, zhuhengshu@gmail.com, keli.xiao@stonybrook.edu, xionghui@gmail.com

***Abstract*—As a vital process to the success of an organization, salary benchmarking aims at identifying the right market rate for each job position. Traditional approaches for salary benchmarking heavily rely on the experiences from domain experts and limited market survey data, which have difficulties in handling the dynamic scenarios with the timely benchmarking requirement. To this end, in this paper, we propose a data-driven approach for intelligent salary benchmarking based on large-scale fine-grained online recruitment data. Specifically, we first construct a salary matrix based on the large-scale recruitment data and creatively formalize the salary benchmarking problem 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. Indeed, by integrating multiple confounding factors, such as company similarity, job similarity, and spatial-temporal similarity, HSBMF is able to provide a holistic and dynamic view for fine-grained salary benchmarking. Finally, extensive experiments on large-scale real-world data clearly validate the effectiveness of our approach for job salary benchmarking.**

***Index Terms*—Salary Benchmarking, Talent Recruitment, Matrix Factorization**

I. 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 [1]–[3], such as OECD [4]. 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 combinations with respect to many possible work locations and time periods.

*Hui Xiong and Hengshu Zhu are the corresponding authors.

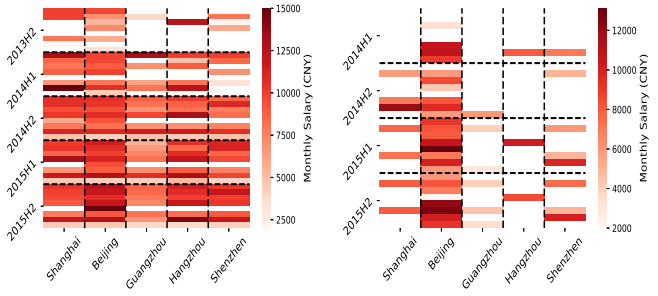
Recently, the prevalence of emerging online recruitment services, such as Glassdoor, Indeed and Lagou, provide opportunities to accumulate massive job related-data from a wide range of companies, and thus enable a new paradigm for salary benchmarking in a data-driven way. To this end, in this paper, 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 paper 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.

II. Preliminaries

In this section, we briefly introduce the recruitment data used in our study and formalize the problem of fine-grained



(a) Position (b) Company

Fig. 1. A snippet of salary distribution in our data. Here, each grid represents a specific job position or company.

salary benchmarking. Also, we discuss the numerical characteristics of the data related to the design of our model.

A. Data Description

In this paper, 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). To facilitate the understanding of our data, we provide several posting examples in Table I. More details of the data will be discussed in Section IV. Indeed, the information similar to our recruitment data is generally available worldwide. Therefore, the method developed in this paper 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 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 1. Therefore, in this paper, we propose a novel approach for fine-grained salary benchmarking to effectively predict expected salaries for unobserved job-company-context combinations.

B. 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 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 II-C. 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.

C. 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 similarities between job positions should be negatively correlated to

TABLE I
SOME TOY EXAMPLES OF JOB POSTINGS IN OUR DATASET.

ID	Job Title	Company Name	Company Size	Work Location	Posting Time	Salary Lower Bound	Salary Upper Bound	Industry	Financial Stage	Job Description
1	Java Developer	Baidu	2,000 and above	Beijing	June. 24, 2015	5,000	10,000	Mobile Internet	Published	Familiar with java ···
2	C++ Engineer	Meituan	2,000 and above	Shenzhen	Aug. 23, 2016	8,000	15,000	E-commerce	Published	Have experience with C++ ···
3	Product Manager	SiyuanTech	50-150	Shanghai	Sep. 10, 2016	15,000	25,000	Mobile Internet	Series A	Responsible for project tasks···

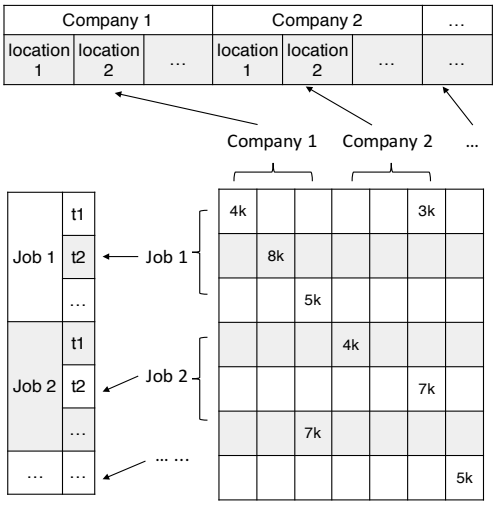
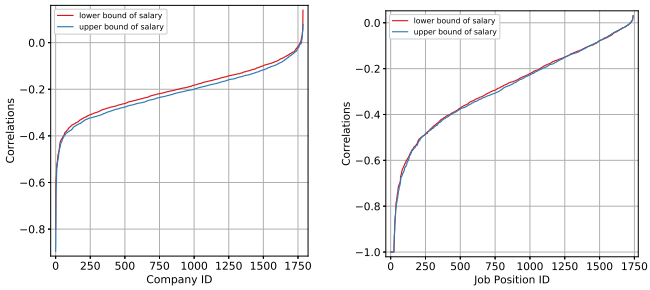


Fig. 2. The structure of the expanded salary matrix.

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 III-B.). Figure 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.

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 (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 II. 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) (b)

Fig. 3. The correlation between job/company similarity and salary difference.

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 II. 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.

Following the above results, we design the expended salary matrix as demonstrated in Fig 2 (i.e., time-specific job po-

TABLE II
THE PEARSON CORRELATION BETWEEN POSTING TIME/WORK LOCATION SIMILARITY AND SALARY DIFFERENCE.

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

sitions 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 III-B, we will provide technical details regarding how we calculate those similarities and integrate them into HSBMF model for higher performance.

III. 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 paper are summarized in Table III.

A. 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 paper, we follow the idea of biased SVD (bSVD) for salary benchmarking as suggested by [5], [6]. Specifically, given an entry $S(j,c)$ in expanded salary matrix S, the predictor is equal to

$$\hat{S}(j,c) \approx \mu + B_j(j) + B_c(c) + J(j,:)C(c,:)^T, \quad (1)$$

where μ, 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 [7], [8], we can formulate the preliminary loss function for salary benchmarking as

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

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

$$I_S(j,c) = \begin{cases} 1, & S(j,c) \text{ exists,} \\ 0, & \text{else.} \end{cases} \quad (3)$$

B. HSBMF with Holistic Constraints

To further refine the performance of salary benchmarking, we integrate more confounding factors as constraints into Equation 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

$$\begin{aligned} R_J &= \frac{1}{2}\sum_{j=1}^{M}\sum_{j'=1}^{M} S_j(j,j')||J(j,:) - J(j',:)||_F^2 \\ &= \sum_{j=1}^{M}\sum_{j'=1}^{M}\sum_{k=1}^{K} S_j(j,j')J(j,k)^2 - \sum_{j=1}^{M}\sum_{j'=1}^{M}\sum_{k=1}^{K} S_j(j,j')J(j,k)J(j',k) \\ &= \sum_{k=1}^{K} J(:,k)^T(D_{S_j} - S_j)J(:,k) \\ &= tr(J^T(D_{S_j} - S_j)J). \end{aligned} \quad (4)$$

TABLE III
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
μ		The global mean of expanded salary matrix
γ		The learning rate
j, j'		A row in J
c, c'		A row in C

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

$$D_{S_j}(u,v) = \begin{cases} \sum_{v=1}^{M} S_j(u,v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \quad (5)$$

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

$$\begin{aligned} R_C &= \frac{1}{2}\sum_{c=1}^{N}\sum_{c'=1}^{N} S_c(c,c')||C(c,:) - C(c',:)||_F^2 \\ &= tr(C^T(D_{S_c} - S_c)C), \end{aligned} \quad (6)$$

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

$$D_{S_c}(u,v) = \begin{cases} \sum_{v=1}^{N} S_c(u,v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \quad (7)$$

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 [9], [10], 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

$$\rho(j, j') = exp(-\alpha|\tau_j - \tau_{j'}|), \tag{8}$$

where α is a positive parameter that controls the temporal evolutionary process, and τ_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

$$\begin{aligned} R_T &= \frac{1}{2}\sum_{j=1}^{M}\sum_{j'=1}^{M} T(j, j')||J(j,:) - J(j',:)||_F^2 \\ &= Tr(J^T(D_T - T)J), \end{aligned} \tag{9}$$

$$D_T(u, v) = \begin{cases} \sum_{v=1}^{M} T(u, v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{10}$$

T is a temporal transition matrix, which is defined as

$$T = \begin{bmatrix} 1 & \rho(1,2) & \cdots & \rho(1,M) \\ \rho(2,1) & 1 & \cdots & \rho(2,M) \\ \vdots & \vdots & \vdots & \vdots \\ \rho(M,1) & \rho(M,2) & \vdots & 1 \end{bmatrix}_{MM}. \tag{11}$$

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:

$$\varphi_(c, c') = 1 - \frac{|AS_c - AS_{c'}|}{max(AS_c, AS_{c'})}, \tag{12}$$

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

$$\begin{aligned} R_L &= \frac{1}{2}\sum_{c=1}^{N}\sum_{c'=1}^{N} L(c, c')||C(c,:) - C(c',:)||_F^2 \\ &= Tr(C^T(D_L - L)C), \end{aligned} \tag{13}$$

$$D_L(u, v) = \begin{cases} \sum_{v=1}^{N} L(u, v), & \text{if } u = v, \\ 0, & \text{else.} \end{cases} \tag{14}$$

With above holistic constraints, we can obtain the final loss function of our HSBMF model by integrating Equation 2 with all regularizers. That is,

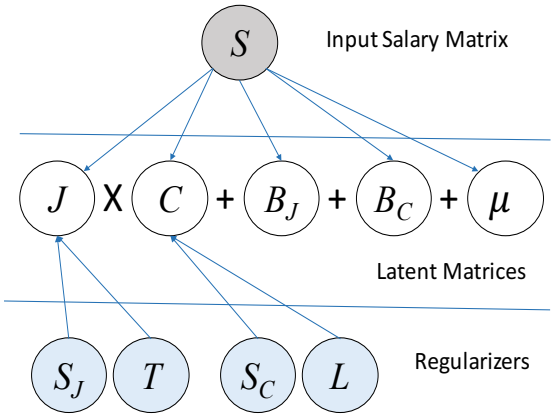


Fig. 4. The graphical representation of our HSBMF model.

$$\begin{aligned} min: \mathcal{F} = \frac{1}{2}\Big(&\sum_{j=1}^{M}\sum_{c=1}^{N}(I_s(j,c) \circ (S(j,c) - \hat{S}(j,c)))^2 \\ &+\lambda_J||J||_F^2 + \lambda_C||C||_F^2 + \lambda_{B_j}||B_j||_F^2 + \lambda_{B_c}||B_c||_F^2 \\ &+\lambda_{S_j}tr(J^T(D_{S_j} - S_j)J) + \lambda_{S_c}tr(C^T(D_{S_c} - S_c)C) \\ &+\lambda_T tr(J^T(D_T - T)J) + \lambda_L tr(C^T(D_L - L)C)\Big). \end{aligned} \tag{15}$$

In summary, Figure 4 shows the graphical representation of the HSBMF model.

C. 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 (15), we have

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

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

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

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

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 γ, we get the updating rules of HSBMF as follows:

$$J(j,k) \leftarrow J(j,k) + \gamma\Big(\sum_{c \in I_J(j)} \big(S(j,c) - \hat{S}(j,c)\big) C(c,k) - |I_J(j)| \times \big(\lambda_{S_j}(D_{S_j} - S_j) J(j,k) + \lambda_T (D_T - T) J(j,k) + \lambda_j J(j,k)\big)\Big), \tag{16}$$

$$C(c,k) \leftarrow C(c,k) + \gamma\Big(\sum_{j \in I_C(c)} \big(S(j,c) - \hat{S}(j,c)\big) J(c,k) - |I_C(c)| \times \big(\lambda_{S_c}(D_{S_c} - S_c) C(c,k) + \lambda_L (D_L - L) C(c,k) + \lambda_c C(c,k)\big)\Big), \tag{17}$$

$$B_j(j) \leftarrow B_j(j) + \gamma\Big(\sum_{c \in I_J(j)} \big(S(j,c) - \hat{S}(j,c)\big) - |I_J(j)| \lambda_{B_j} B_j(j)\Big), \tag{18}$$

$$B_c(c) \leftarrow B_c(c) + \gamma\Big(\sum_{j \in I_C(c)} \big(S(j,c) - \hat{S}(j,c)\big) - |I_C(c)| \lambda_{B_c} B_c(c)\Big). \tag{19}$$

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 μ. 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[1].

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.

IV. Experimental Results

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

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

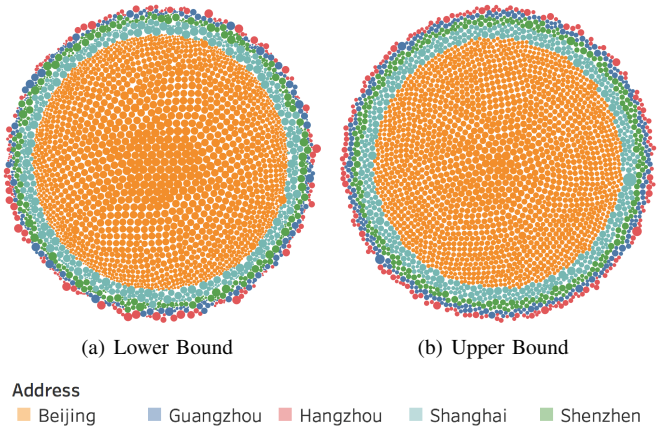
Algorithm 1 HSBMF Optimization

Input:
S,S_j,S_c,T,L,D_{S_j}, D_{S_c},D_T,D_L,μ
λ_j,λ_c,λ_{S_j},λ_{S_c},λ_T,λ_L,λ_{B_j},λ_{B_c},γ,α
Output: J, C, B_j, B_c

1: Initialize J, C, B_j, B_c with random values
2: **while** Iterations $< Max\_Iter$ **do**
3: $AuxiliaryJ = \big(\lambda_{S_j}(D_{S_j} - S_j) + \lambda_T(D_T - T) + \lambda_j\big)J$
4: $AuxiliaryC = \big(\lambda_{S_c}(D_{S_c} - S_c) + \lambda_L(D_L - L) + \lambda_c\big)C$
5: **for** each (j,c) in the S **do**
6: $\hat{S} = \mu + B_j(j) + B_c(c) + J(j,:)C(c,:)^T$
7: $err = S - \hat{S}$
8: # update bias B_j and B_c
9: $B_j(j) = B_j(j) + \gamma\big(err - \lambda_{B_j} B_j(j)\big)$
10: $B_c(c) = B_c(c) + \gamma\big(err - \lambda_{B_c} B_c(c)\big)$
11: # update J and C
12: **for** each k **do**
13: $J(j,k) = J(j,k) + \gamma\big(err * C(c,k) - AuxiliaryJ(j,k)\big)$
14: $C(c,k) = C(c,k) + \gamma\big(err * J(j,k) - AuxiliaryC(c,k)\big)$
15: **end for**
16: **end for**
17: **end while**
18: **return** C,J,B_j,B_c

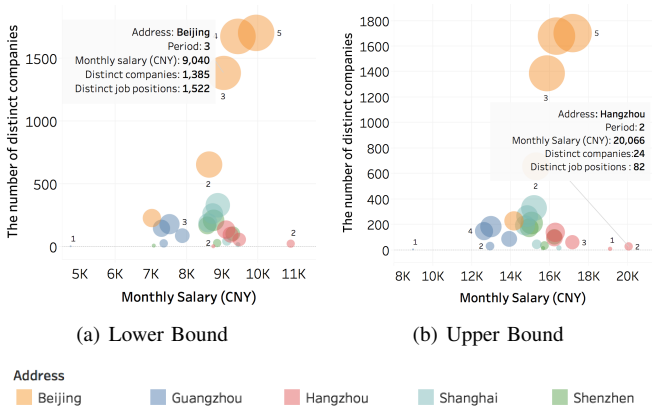
A. The Experimental Setup

As introduced in Section II, 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 5. We also plotted the scatter bubble chart for each location and time period in Figure 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 positions and companies arises rapidly in Beijing, while that of the other cities arise mildly, which means the dataset is unbalanced over



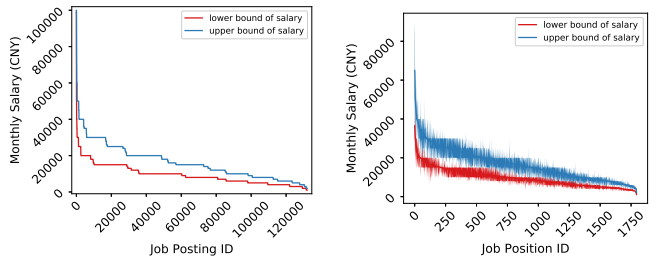
(a) Lower Bound (b) Upper Bound

Fig. 5. The bubble chart of salary, where each bubble represents a company, and the scale is proportional to the value.



(a) Lower Bound (b) Upper Bound

Fig. 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

Fig. 7. The salary distribution in our dataset.

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 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 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 IV. Note that, in the experiments, we evaluated the performance of HSBMF on the lower bound and the upper bound of salary, respectively.

TABLE IV
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

B. 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 [11], NMF [7], PMF [12], 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 paper.
- **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

$$RMSE = \sqrt{\frac{\sum_i^{Num}(S_i - \hat{S}_i)^2}{Num}}, \tag{20}$$

$$MAE = \frac{\sum_i^{Num}|S_i - \hat{S}_i|}{Num}, \tag{21}$$

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.

TABLE V
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 VI
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 VII
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

C. 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 V, VI, and VII 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.

D. 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 VIII. 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 λ_T and λ_L. Nonetheless, the p-Values in all experiments are very small, demonstrating that the improvements are statistically significant for all four constraints.

E. Evaluation on Parameter Sensitivity

As discussed above, since HSBMF is not sensitive to λ_T and λ_L, we fixed $\lambda_T = 2 \times 10^{-4}$ and $\lambda_L = 2 \times 10^{-4}$, and evaluated the sensitivity of λ_{S_j} and λ_{S_c} by changing them from 0 to 2×10^{-4}. Figure 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 λ_{S_j} and λ_{S_c} are approaching to 2×10^{-4}, the results achieve the best performances. This means the job position similarity and company similarity are effective factors for salary benchmarking.

V. RELATED WORK

In general, the related work of this study can be grouped into two categories, namely salary analytics and MF based recommendation models.

Salary analytics is a popular research topic in both management science and econometrics. The objectives of salary analytics usually focus on the salary equity, satisfaction of employees, and the confounding factors which potentially influence the salary structures [1]–[3]. Traditional approaches for salary analytics rely heavily on the experiences of domain

TABLE VIII
EVALUATION ON DIFFERENT CONSTRAINS.

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

experts and the limited survey data from third parties. For instance, [13] proposed to use Support Vector Machine based on survey data for salary prediction. [14] proposed a Bayesian regression model to predict salary with peer group effects on a data set collected from nursing facilities. Recently, the prevalence of online recruitment data has draw big attention for recruitment analyses [15]–[19]. Specially, [20] proposed a collaborative topic regression model, which could integrate online public opinions for predicting job salaries. However, this approach needs additional review data from former employees, and does not consider the influence of latent factors such as job similarity and other contextual information.

MF techniques is widely used in recommender systems, besides that, they also applied to a broad related areas, such as social network analyses [21], [22], image tagging [23], document clustering [24] 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 [25]. 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 [26]. As a result, researchers suggest only to model the ratings observed, and add adequate regularizers to avoid over-fitting problems [6]. 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 [6], [11]. SVD++ interprets the data with the effect of "implicit" information of users or items [5]. In addition, NMF also belongs to MF families. However, different from SVD, NMF constrains latent factors to be non-negative [27], [28]. Finally, PMF places zero-mean spherical Gaussian priors on user and item feature vectors [12], 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 [29], classifying the approaches into three categories: pre-filtering, post-filtering, and contextual modeling. Item-splitting [30] 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. [31] 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 [32]–[36]. One well-known method is tensor factorization (TF) proposed by [37]. 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 [38], 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 paper, HSBMF is MF-based approach that integrates holistic constraints for fine-grained salary benchmarking.

VI. CONCLUSIONS

In this paper, 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.

VII. ACKNOWLEDGMENT

This work was supported in part by the grant from the National Natural Science Foundation of China (Grant No.91746301).

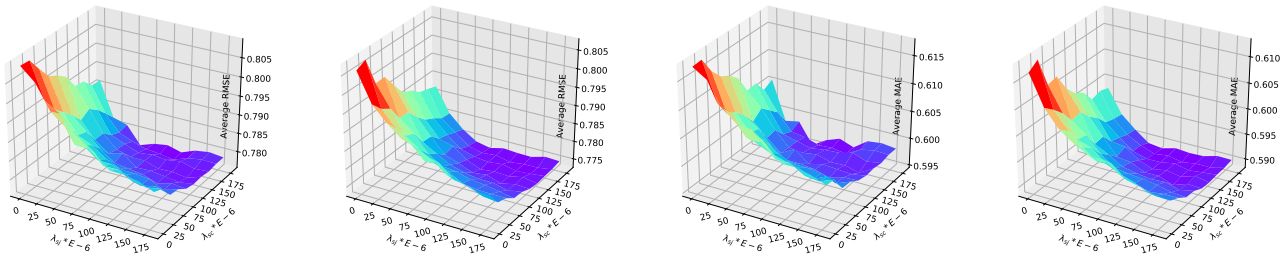
REFERENCES

[1] C. B. Johnson, M. L. Riggs, and R. G. Downey, "Fun with numbers: Alternative models for predicting salary levels," *Research in Higher Education*, vol. 27, no. 4, pp. 349–362, 1987.

[2] C. G. Schau and V. H. Heyward, "Salary equity: Similarities and differences in outcomes from two common prediction models," *American Educational Research Journal*, vol. 24, no. 2, pp. 271–286, 1987.

[3] C. O. Porter, D. E. Cordon, and A. E. Barber, "The dynamics of salary negotiations: Effects on applicants'justice perceptions and recruitment decisions," *International Journal of Conflict Management*, vol. 15, no. 3, pp. 273–303, 2004.

[4] "The organisation for economic co-operation and development: http://www.oecd.org/."



(a) RMSE for Lower Bound (b) RMSE for Upper Bound (c) MAE for Lower Bound (d) MAE for Upper Bound

Fig. 8. The performance of HSBMF with different parameter settings of λ_{S_j} and λ_{S_c}.

[5] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[6] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, pp. 5–8.

[7] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[8] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender systems handbook*. Springer, 2015, pp. 77–118.

[9] Y. Yao, W. X. Zhao, Y. Wang, H. Tong, F. Xu, and J. Lu, "Version-aware rating prediction for mobile app recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, p. 38, 2017.

[10] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 93–100.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[12] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[13] A. Lazar, "Income prediction via support vector machine." in *ICMLA*, 2004, pp. 143–149.

[14] E. Blankmeyer, J. P. LeSage, J. Stutzman, K. J. Knox, and R. K. Pace, "Peer-group dependence in salary benchmarking: a statistical model," *Managerial and Decision Economics*, vol. 32, no. 2, pp. 91–104, 2011.

[15] T. Xu, H. Zhu, C. Zhu, P. Li, and H. Xiong, "Measuring the popularity of job skills in recruitment market: A multi-criteria approach," *arXiv preprint arXiv:1712.03087*, 2017.

[16] D. Shen, H. Zhu, C. Zhu, T. Xu, C. Ma, and H. Xiong, "A joint learning approach to intelligent job interview assessment." in *IJCAI*, 2018, pp. 3542–3548.

[17] C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "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*. ACM, 2018, pp. 25–34.

[18] C. Zhu, H. Zhu, H. Xiong, P. Ding, and F. Xie, "Recruitment market trend analysis with sequential latent variable models," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 383–392.

[19] H. Li, Y. Ge, H. Zhu, H. Xiong, and H. Zhao, "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*. ACM, 2017, pp. 917–925.

[20] H. Lin, H. Zhu, Y. Zuo, C. Zhu, J. Wu, and H. Xiong, "Collaborative company profiling: Insights from an employee's perspective." in *AAAI*, 2017, pp. 1417–1423.

[21] K. Xiao, Q. Liu, C. Liu, and H. Xiong, "Price shock detection with an influence-based model of social attention," *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 1, p. 2, 2017.

[22] L. Zhang, K. Xiao, Q. Liu, Y. Tao, and Y. Deng, "Modeling social attention for stock analysis: An influence propagation perspective," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 609–618.

[23] N. Zhou, W. K. Cheung, G. Qiu, and X. Xue, "A hybrid probabilistic model for unified collaborative and content-based image tagging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1281–1294, 2011.

[24] W. Xu, X. Liu, and Y. Gong, "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*. ACM, 2003, pp. 267–273.

[25] G. Adomavicius and A. Tuzhilin, "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*, vol. 17, no. 6, pp. 734–749, 2005.

[26] D. Kim and B.-J. Yum, "Collaborative filtering based on iterative principal component analysis," *Expert Systems with Applications*, vol. 28, no. 4, pp. 823–830, 2005.

[27] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[28] ——, "Algorithms for non-negative matrix factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*. MIT Press, 2000, pp. 535–541.

[29] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 191–226.

[30] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 245–248.

[31] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 265–268.

[32] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 635–644.

[33] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35–65, 2014.

[34] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 58, 2015.

[35] T. Bao, H. Cao, E. Chen, J. Tian, and H. Xiong, "An unsupervised approach to modeling personalized contexts of mobile users," *Knowledge and Information Systems*, vol. 31, no. 2, pp. 345–370, 2012.

[36] Y. Ge, C. Liu, H. Xiong, and J. Chen, "A taxi business intelligence system," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 735–738.

[37] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[38] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 301–304.