Trend-Aware Tensor Factorization for Job Skill Demand Analysis

Xunxian Wu[1] , **Tong Xu**[1,*] , **Hengshu Zhu**[2] , **Le Zhang**[1] , **Enhong Chen**[1] , **Hui Xiong**[1,2,3,*]

[1]Anhui Province Key Lab of Big Data Analysis and Application, School of Computer Science, University of Science and Technology of China

[2]Baidu Talent Intelligence Center, Baidu Inc.

[3]Business Intelligence Lab, Baidu Research

{wjytc, laughing}@mail.ustc.edu.cn, {tongxu, cheneh}@ustc.edu.cn, zhuhengshu@baidu.com, xionghui@gmail.com

Abstract

Given a job position, how to identify the right job skill demand and its evolving trend becomes critically important for both job seekers and employers in the fast-paced job market. Along this line, there still exist various challenges due to the lack of holistic understanding on skills related factors, e.g., the dynamic validity periods of skill trend, as well as the constraints from overlapped business and skill co-occurrence. To address these challenges, in this paper, we propose a trend-aware approach for fine-grained skill demand analysis. Specifically, we first construct a tensor for each timestamp based on the large-scale recruitment data, and then reveal the aggregations among companies and skills by heuristic solutions. Afterwards, the Trend-Aware Tensor Factorization (TATF) framework is designed by integrating multiple confounding factors, i.e., aggregation-based and temporal constraints, to provide more fine-grained representation and evolving trend of job demand for specific job positions. Finally, validations on large-scale real-world data clearly validate the effectiveness of our approach for skill demand analysis.

1 Introduction

With the coming of knowledge economy era, competition for skilled talents becomes extremely severe. Along this line, skill-oriented recruitment market analysis, has become increasingly crucial for both employers and job seekers. On the one hand, it is necessary for employers to understand the skill trend. On the other hand, job seekers have to acquire specific skills for competing against other candidates. Therefore, large efforts have been made on this task.

Traditionally, skill-oriented analyses were conducted by human resource (HR) experts based on simple statistics of historical recruitment records. With the rapid development of online recruitment services, such as LinkedIn and Lagou in China, massive recruitment data have been accumulated to support comprehensive summarization based on data mining techniques. For instance, LinkedIn has published the ranking lists of popular skills for each year [1] to guide the recruitment market. Moreover, some prior arts, e.g., [Xu *et al.*, 2018] and [Zhu *et al.*, 2016] proposed data driven solutions for job skill demand analysis based on various techniques. However, the fine-grained skill demand analysis for specific targets (e.g., positions) has not yet been fully solved.

To achieve the fine-grained skill demand analysis, intuitively, based on the triple form as (*company-position-skill*), classic *Tensor Factorization* (TF) methods might be appropriate due to its effectiveness and interpretability. However, basic TF method may suffer some defects. Firstly, *dynamic validity periods* exists for skills, which reflects the evolving skill trend with temporal correlation. For example, though Deep Learning has become the mainstream technique, traditional solutions like logistic regression are still widely required in industrial applications. Secondly, more constraints could be designed based on various aggregations. For instance, companies with overlapping business may require similar skills, e.g., both Google and Amazon pursue talents on cloud computing, which results in the *competition-based aggregation*. At the same time, co-occurrence of skills may lead to the skill package for the common task, e.g., Ajax and CSS for UI design, which results in the *co-occurrence-based aggregation*. Therefore, basic TF method could be further enhanced with integrating temporal correlation and various aggregations.

To that end, in this paper, we propose a more comprehensive framework called Trend-Aware Tensor Factorization (TATF). Specifically, we firstly estimate the aggregations among companies and skill packages via heuristic solutions, and then constraints for both skill trend (temporal correlation) and various aggregations are integrated into basic TF method to improve the performance. Along this line, more accurate representation for specific skill and position requirement will be achieved to support position-specific skill demand analysis for job seekers. To the best of our knowledge, we are among the first ones who study the job skill demand analysis in the perspective of tensor factorization with comprehensive enhancements. Extensive experiments on real-world data set have demonstrated the effectiveness of our TATF framework compared with several baselines, and further provide some interesting rules via case studies.

*Corresponding Author

[1]https://learning.linkedin.com/blog/top-skills/the-skills-companies-need-most-in-2018–and-the-courses-to-get-t

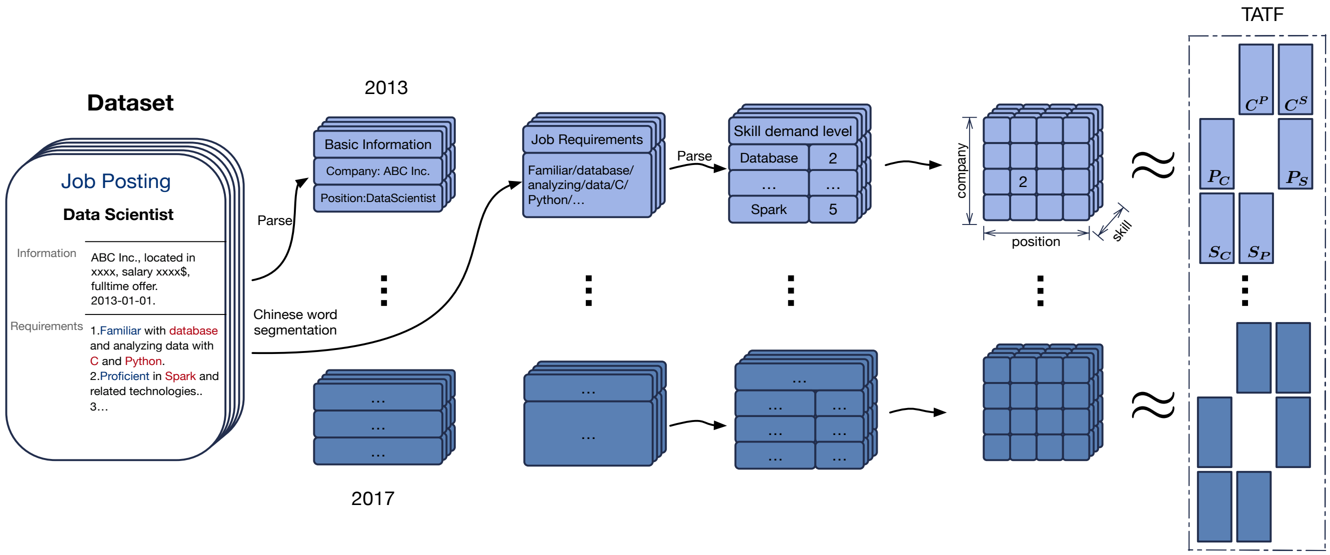


Figure 1: The Overview of Data Flow for Tensor Formation

2 Technical Solution of TATF

In this paper, we attempt to develop a data-driven approach for analyzing demand of job skills with specific target and comprehensive constraints. To deal with this task, we will first formulate our job skill demand analysis in the perspective of *Tensor Factorization*, and then propose our Trend-Aware Tensor Factorization (TATF) in detail.

2.1 Preliminary & Problem Definition

Intuitively, to measure the appropriateness of job skills via Tensor Factorization (TF) methods, we firstly formulate the recruitment records in the triple form as *(c-p-s: l)*, in which:

- $c \in \mathbb{C}$ denotes the specific company;
- $p \in \mathbb{P}$ denotes the specific job position;
- $s \in \mathbb{S}$ denotes the specific job skill;
- $l \in \mathbb{L}$ indicates the level of demand for skill.

Along this line, given a triple formed as *(c-p-s)* $\in \mathbb{C}\times\mathbb{P}\times\mathbb{S}$, we target at measuring the potential level *l* for the given tripe, so that skill s could be recommended to the specific position p in company c, if highly demanded.

Moreover, as mentioned above that the *temporal trend* of job skills may further enhance the performance, we extend the triple form as quadruple with considering the temporal factor as *(t-c-p-s)*, in which $t \in \mathbb{T}$ denotes the timestamp of specific record. Under this situation, our job skill demand analysis will be formally defined as follow:

Definition 1 (Problem Statement)**.** *Given the set of recruitment records $\mathcal{A} \subset \mathbb{T}\times\mathbb{C}\times\mathbb{P}\times\mathbb{S}\times\mathbb{L}$, for each triple (t-c-p), we target measuring the demand level l for each potential skill s, so that highly demanded skills will be revealed.*

Data flow of tensor formation is summarized in Figure 1.

2.2 Basic Tensor Factorization Model

Then, we turn to introduce the basic model of our skill demand analysis. Intuitively, according to the definitions above, we realize that the records $\mathcal{A}$ could be directly formulated as *tensors* in the form of *(t-c-p-s: l)*, and then the *Tensor Factorization* (TF) method will be adapted to deal with this problem. The basic TF model could be formulated as follow:

$$\min_{\mathcal{P}} \mathcal{L}(\mathcal{P}|\mathcal{A}) \triangleq \|\mathcal{A} - \tilde{\mathcal{A}}(\mathcal{P})\| + \lambda\|\mathcal{P}\|, \tag{1}$$

in which $\mathcal{P}$ presents all the parameters used in this model, and $\|\cdot\|$ denotes any reasonable norm here. Besides, λ presents the hyper-parameter of the regularization term.

To optimize the objective function above, we adopt the widely used solution, namely the *Pairwise Interaction Tensor Factorization* (PITF) [Rendle and Schmidt-Thieme, 2010] method, in which the tensors will be transferred as several matrices to ease the computation. Specifically, if we utilize the PITF method for the basic skill demand analysis, i.e., the triple tensor *(c-p-s: l)* without temporal factor, the tensor in PITF will be transferred as follow:

$$\tilde{\mathcal{A}}^{PITF}_{i,j,k} = \langle \boldsymbol{C}^{\boldsymbol{P}}_{i,*}, \boldsymbol{P}^{\boldsymbol{C}}_{j,*}\rangle + \langle \boldsymbol{C}^{\boldsymbol{S}}_{i,*}, \boldsymbol{S}^{\boldsymbol{C}}_{k,*}\rangle + \langle \boldsymbol{P}^{\boldsymbol{S}}_{j,*}, \boldsymbol{S}^{\boldsymbol{P}}_{k,*}\rangle. \tag{2}$$

Along this line, we have six matrices to present the latent factor for three elements in tensor, namely $\boldsymbol{C}^{\boldsymbol{P}}, \boldsymbol{C}^{\boldsymbol{S}} \in \mathbb{R}^{|\mathbb{C}|\times K}$ for the "company", $\boldsymbol{P}^{\boldsymbol{C}}, \boldsymbol{P}^{\boldsymbol{S}} \in \mathbb{R}^{|\mathbb{P}|\times K}$ for the "position", and $\boldsymbol{S}^{\boldsymbol{C}}, \boldsymbol{S}^{\boldsymbol{P}} \in \mathbb{R}^{|\mathbb{S}|\times K}$ for the "skill". Here K indicates the hyper-parameter for the dimension of latent factor. In this case, pairwise relations among elements could be clearly reflected and estimated. Then, the objective function will be adapted as follow:

$$\begin{aligned}\mathcal{L}^{PITF}(\mathcal{P}|\mathcal{A}) &\triangleq \sum_{i,j,k}(\mathcal{A}_{i,j,k} - \tilde{\mathcal{A}}^{PITF}_{i,j,k})^2 \\ &+ \lambda_0(\|\boldsymbol{C}^{\boldsymbol{P}}\|_F^2 + \|\boldsymbol{C}^{\boldsymbol{S}}\|_F^2 + \|\boldsymbol{P}^{\boldsymbol{C}}\|_F^2 \\ &+ \|\boldsymbol{P}^{\boldsymbol{S}}\|_F^2 + \|\boldsymbol{S}^{\boldsymbol{C}}\|_F^2 + \|\boldsymbol{S}^{\boldsymbol{P}}\|_F^2).\end{aligned}$$

2.3 Enhanced Tensor Factorization with Aggregation-based Constraint

Based on the basic TF model, we then turn to introduce our TATF framework with enhanced constraints. Specifically, we will first introduce the comprehensive constraints based on two *aggregations*, i.e., *competition* (among companies) and *co-occurrence* (among skills) based aggregations. Technical details to achieve the aggregations will be introduced in Section 3.1 in experimental part. In this section, we will focus on how to design the constraints based on aggregations.

Constraint for Competition-based Aggregation

First, we will introduce the constraint for competition-based aggregation. Intuitively, companies with overlapping business may probably seek for talents with similar positions or skills, which could be easily reflected by their similar job postings. Correspondingly, companies with similar job postings should have similar preference on positions or skills, i.e., similar features for each company. Therefore, we could reveal the competition-based aggregation among companies via measuring the similarities of their job postings, and then enhance the TF model with the aggregation.

Specifically, given the $\mathcal{C}$ categories of business, considering that each company may involve multiple business categories, we have $P(i|c)$ to denote the probability of company c which belongs to the i-th category. Along this line, we design the constraint as follow:

$$\begin{aligned}\mathcal{L}^1(\mathcal{P}|\mathcal{A}) \triangleq\ & \lambda_0(\|\mathcal{G}^{1,\boldsymbol{P}}\|_F^2 + \|\mathcal{G}^{1,\boldsymbol{S}}\|_F^2) \\ & + \lambda_1 \sum_{c=0}^{|\mathbb{C}|-1} \sum_i P(i|c)(\|\mathcal{G}^{1,\boldsymbol{P}}_{i,*} - \boldsymbol{C}^{\boldsymbol{P}}_{c,*}\|^2 \\ & + \|\mathcal{G}^{1,\boldsymbol{S}}_{i,*} - \boldsymbol{C}^{\boldsymbol{S}}_{c,*}\|^2).\end{aligned} \tag{3}$$

Here, two new parameter matrices are utilized, namely $\mathcal{G}^{1,\boldsymbol{P}}, \mathcal{G}^{1,\boldsymbol{S}} \in \mathbb{R}^{\mathcal{C}\times K}$ to present the features of "*center company*" of each category, which indicates the latent preference for position and skill, separately. In this case, companies which belong to the same business category should be mutually similar, which leads to the constraint.

Constraint for Co-occurrence-based Aggregation

Second, we will introduce the constraint for co-occurrence-based aggregation. As mentioned above that co-occurrence of skills may lead to the "skill package" for the common task. Correspondingly, skills in the same package could probably satisfy the similar companies or positions. Therefore, similar with the constraint above, given the $\mathcal{S}$ packages of skills, we have $P(i|s)$ to denote the probability of skill s which belongs to the i-th package, and then design the constraint as follow:

$$\begin{aligned}\mathcal{L}^2(\mathcal{P}|\mathcal{A}) \triangleq\ & \lambda_0(\|\mathcal{G}^{2,\boldsymbol{C}}\|_F^2 + \|\mathcal{G}^{2,\boldsymbol{P}}\|_F^2) \\ & + \lambda_2 \sum_{s=0}^{|\mathbb{S}|-1} \sum_i P(i|s)(\|\mathcal{G}^{2,\boldsymbol{C}}_{i,*} - \boldsymbol{S}^{\boldsymbol{C}}_{c,*}\|^2 \\ & + \|\mathcal{G}^{2,\boldsymbol{P}}_{i,*} - \boldsymbol{S}^{\boldsymbol{P}}_{c,*}\|^2).\end{aligned} \tag{4}$$

Here, we have two more new parameter matrices as $\mathcal{G}^{2,\boldsymbol{C}}, \mathcal{G}^{2,\boldsymbol{P}} \in \mathbb{R}^{\mathcal{S}\times K}$ to present the features of "*center skill*" of each package, which indicates the latent preferences for company and position, separately.

Integrated Model & Complexity Analysis

With integrating two constraints above, we could now enhance the basic TF model with the loss function as follow:

$$\mathcal{L}(\mathcal{P}|\mathcal{A}) \triangleq \mathcal{L}^{PITF}(\mathcal{P}|\mathcal{A}) + \mathcal{L}^1(\mathcal{P}|\mathcal{A}) + \mathcal{L}^2(\mathcal{P}|\mathcal{A}). \tag{5}$$

Compared with the traditional PITF method, the Equation 5 require additional time complexity as $\Theta((\mathcal{C} + \mathcal{S} + |\mathbb{C}| + |\mathbb{S}|)K)$, which results in the *overall time complexity* as $\Theta((|\mathcal{A}| + |\mathbb{C}| + |\mathbb{P}| + |\mathbb{S}| + \mathcal{C} + \mathcal{S})K)$.

Considering that $|\mathcal{A}|$, i.e., the amount of records, could be much larger than the rest part, i.e., $|\mathbb{C}| + |\mathbb{P}| + |\mathbb{S}| + \mathcal{C} + \mathcal{S}$. Thus, the overall time complexity is mainly determined by the amount of records $|\mathcal{A}|$, and the dimension of latent vector K. In this case, time complexity of our enhanced model may not be significantly higher than the basic PITF model.

2.4 Trend-aware Tensor Factorization (TATF)

Finally, based on the constraint-enhanced TF model, in this subsection, we will introduce the temporal factor to propose the complete TATF framework, so that the evolving trend of job skills will be captured.

Intuitively, supposing that the recruitment records are denoted as $\{\mathcal{A}_0, ..., \mathcal{A}_{|\mathbb{T}|-1}\}$, with $|\mathbb{T}|$ timestamps in total. To integrate the temporal factor, for each time t, we could individually estimate the corresponding parameters as $\mathcal{P}_t$, so that those time-sensitive features could be captured. Moreover, considering the *dynamic validity period* exists for skills with temporal correlation, which means that skill requirements in adjacent timestamps should be mutually similar. Therefore, we further design the *temporal constraint* based on previous models, which could be formulated as follows:

$$\begin{aligned}\mathcal{P}_0^{opt} &= \arg\min_{\mathcal{P}} \mathcal{L}(\mathcal{P}|\mathcal{A}_0), \\ \mathcal{P}_t^{opt} &= \arg\min_{\mathcal{P}} \mathcal{L}(\mathcal{P}|\mathcal{A}_t) + \lambda_3 \sum_{i=0}^{t-1} \varphi_t(i)\|\mathcal{P} - \mathcal{P}_i^{opt}\|^2.\end{aligned} \tag{6}$$

Based on the formulations above, for the first timestamp, i.e., $t = 0$, parameters will be estimated without temporal constraint. Then, as time goes by, newly estimated parameters will be constrained by all the previous models with the hyper-parameter λ_3 to measure the weight. Also, we have $\varphi_t(i)$ to measure the time decay effects, so that influence from earlier models will be limited. $\varphi_t(i)$ is formulated as follow:

$$\varphi_t(i) = \frac{b}{|t-i|}, b > 0. \tag{7}$$

Convergence Analysis. For both Equation 5 and 6, only two norms, namely *L2-norm* and *F-norm* are used in the loss functions. Therefore, for any row vector in any parameter matrix, the loss functions $\mathcal{L}$ satisfies the Lipschitz continuous condition. Along this line, for all the parameters $\mathcal{P}$, the loss function still satisfies the Lipschitz continuous condition. According to the prior art [Ghadimi and Lan, 2013], for a non-convex and smooth loss function, if the SGD algorithm is used for training, it will converge to the critical point at the convergence speed of $\|\nabla f(x)\|^2 \leq O(1/\sqrt{T})$. In summary, the convergence of our TATF method could be ensured.

3 Experiments

In this section, we will evaluate the effectiveness of our purposed TATF framework. After that, we will discuss some interesting rules via intuitive case studies, which further confirm our basic assumptions.

3.1 Experimental Setup

Firstly, we will summarize the details of experimental setup in this subsection, including data description, pre-processing, aggregation method, baseline methods and metrics.

Dataset Description & Preprocessing

In our research, the experiments were conducted on a real-world data set collected from an online recruiting market. Specifically, we have 2,284,903 job postings in total, which were posted by 147,690 companies during the year from 2013 to 2017. To ensure the data quality, we set threshold for companies, positions and skills as 20, 10 and 10, respectively. In other words, companies with less than 20 job postings will be removed, and so do the rest. Related statistics for both raw and filtered data set are summarized in Table 1.

Along this line, for each job posting, some detailed information, including company, position, timestamp and skill requirements were extracted. For describing skills in a unified form, we utilized the pre-defined job skill list, which is widely used in prior arts like [Xu *et al.*, 2018]. Then, we matched the list with each job posting to extract mentioned skills.

At the same time, for the level of demand for each skill, i.e., the l for each quadruple *(t-c-p-s)*, we rated the level based on textual description in skill requirements. For example, for skill *A*, we have "expert in *A*" in the requirement of job posting corresponds to the quadruple *(t-c-p-s)*, then the level here is set as "Very High", i.e., $l = 5$. Similarly, "familiar with *A*" will lead to $l = 3$, namely "Medium". Based on the ratings, we divided the demand level as 1-5 for each quadruple.

Solutions for Aggregation

Then, we turn to introduce the solutions for aggregations of companies and skills, as mentioned in Section 2.3.

To be specific, we captured the skill packages based on their co-occurrence in job postings. Then, we intuitively used the classic Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] to reveal the latent aggregation among skills. For each job posting, we treat it as a document, in which the extracted skills are treated as words. Besides, the skills will be repeated based on their demand level, e.g., a document as "*Java, Java, MySQL*" indicates that *Java* is demanded in level 2, and level 1 for *MySQL*. Along this line, LDA could reveal the latent topic, i.e., skill package for the common task. Intuitively, we have $\mathcal{S}$ topics (skill packages) for LDA, and the hyper-parameters were set as $\alpha = 50/\mathcal{S}, \beta = 0.1$. Sensitiveness of $\mathcal{S}$ will be discussed in Section 3.2.

At the same time, for the business overlapping among companies, we clustered the companies based on their preference vectors, which were generated via connecting each row vector in the matrices as (position-skill) for each company. Along this line, we utilized the classic K-means method with $\mathcal{C}$ clusters, so that similar companies will be aggregated based on their Euclidean distance of preference vectors. Sensitiveness of $\mathcal{C}$ will be also discussed in Section 3.2.

	Raw Data	Filtered data
Unique Companies	147,690	24,531
Unique Positions	500,941	11,826
Unique Skills	1,832	681
Elements in Tensor	5,897,739	2,151,445
Sparsity of Tensor	4.38×10^{-8}	1.08×10^{-4}

Table 1: The Statistics of Data Set.

Baseline Methods & Evaluation Metrics

To validate the performance of TATF for skill demand analysis, several typical baseline methods are selected as baseline methods as follows:

- Market Average (**MA**), which intuitively counts the most popular skill for specific position in training data (without considering the companies).
- Tucker Decomposition (**TD**) [Tucker, 1966], which is a classic method to treat the original sparse tensor as product of a small tensor and several matrices.
- Candecomp/Parafac Decomposition (**CP**) [Harshman, 1970], a variant of TD but removes the small tensor.
- **PITF** [Rendle and Schmidt-Thieme, 2010], which is the basic model of our TATF framework.

At the same time, to evaluate the performance, three metrics were utilized as follows:

- Mean Square Error (**MSE**), which is the mean value of squared error for all samples, as $\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$.
- Mean Absolute Percentage Error (**MAPE**), which focuses on the absolute ratio between the prediction error and ground truth value, as $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{f(x_i)-y_i}{y_i}\right|$,
- R-Square (**RS**), which measures the how the proposed model is better than the "mean" value, compared with the ground truth value separately, as $1 - \frac{\sum_{i=1}^{n}(f(x_i)-y_i)^2}{\sum_{i=1}^{n}(\bar{y}-y_i)^2}$.

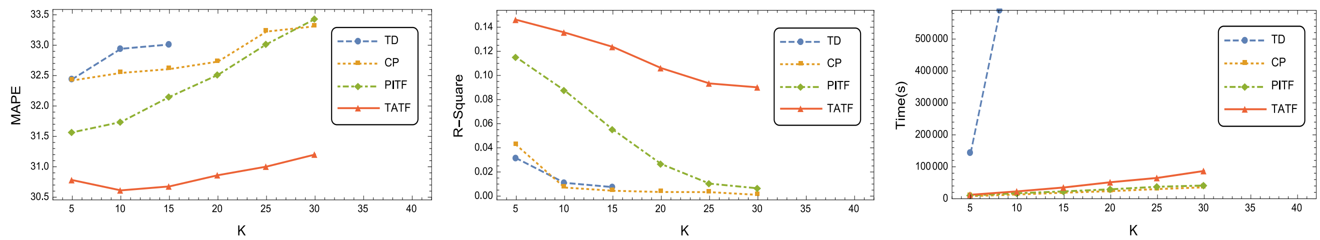
3.2 Experimental Results

Then, we turn to introduce the experimental results. Specifically, one experiment was executed for records in one year. Along this line, to integrate the trend-aware constraint as mentioned in Section 2.4, model parameters of previous years were also utilized following Equation 7.

Overall Performance

First, we will report the overall performance of TATF framework compared with baseline methods. To be specific, here $\mathcal{C}$ and $\mathcal{S}$ for aggregations were set as 20, and K for the dimension of vector was set as 10. Besides, we have $\lambda_0 = 0.5$, and $\lambda_1 = \lambda_2 = \lambda_3 = 1$ for the regularization term.

The results are summarized in Table 2, in which the improvement of our TATF method compared with baselines are also listed. According to the results, we realized that our TATF outperforms all the baselines, which verified that our assumptions of various enhancement could really benefit the skill demand analysis. At the same time, we found that the heuristic method performed the worst, which indicated that the performance is sensitive to the company factor.



(a) (b) (c)

Figure 2: Sensitiveness of Vector Dimension K, (a) MAPE, (b) R-Square, (c) Time Spending

	MA	TD	CP	PITF	TATF
Time complexity	-	$\Theta(\|\mathcal{A}\|K^3)$	$\Theta(\|\mathcal{A}\|K)$	$\Theta(\|\mathcal{A}\|K)$	$\Theta(\|\mathcal{A}\|K)$
CPU Time(s)	< 100	851746	13345	17260	23164
MSE	1.3897	1.1923	1.1868	1.0906	1.0332
Improve(%)	25.653	13.341	12.936	5.2567	-
MAPE	35.111	32.940	32.543	31.735	30.611
Improve(%)	17.920	7.0726	5.9367	3.5418	-
R-Square	-0.162	0.0110	0.0071	0.0875	0.1355
Improve(%)	-	1125.1	1808.4	54.857	-

Table 2: Overall Performance

	2013	2014	2015	2016	2017
Basic Model	1.99781	1.24219	1.07473	1.07103	1.06157
BM+1	1.89819	1.21796	1.05541	1.04773	1.03919
BM+2	1.52479	1.15705	1.05505	1.04445	1.03687
BM+3	1.99781	1.21293	1.05290	1.04211	1.02761
BM+1+2	1.33991	1.15614	1.03655	1.02855	1.01332
BM+1+3	1.89819	1.19058	1.04758	1.03748	1.02034
BM+2+3	1.52479	1.16272	1.03404	1.01788	0.99985
TATF	1.33991	1.13541	1.02820	1.01374	0.99629

Table 3: Ablation Study with Three Constraints: 1) Competition-based Aggregation, 2) Co-occurrence-based Aggregation, 3) Trend-aware Temporal Constraint.

		$\mathcal{C}=20$	40	60	80	100
MSE	$\mathcal{S}=20$	1.03320	1.03367	1.02982	1.03115	1.03069
	40	1.02975	1.03554	1.03209	1.03183	1.03196
	60	1.03374	1.03646	1.03376	1.03309	1.03194
	80	1.03323	1.03485	1.03746	1.03552	1.03772
	100	1.04158	1.03597	1.03571	1.03474	1.03713
MAPE	$\mathcal{S}=20$	30.6113	30.6342	30.5719	30.6063	30.5712
	40	30.5497	30.6430	30.5934	30.6054	30.5557
	60	30.6264	30.6600	30.6264	30.5834	30.5733
	80	30.6043	30.6206	30.6770	30.6792	30.6418
	100	30.6931	30.6636	30.6766	30.6354	30.6336

Table 4: Experimental results of different hyperparametric.

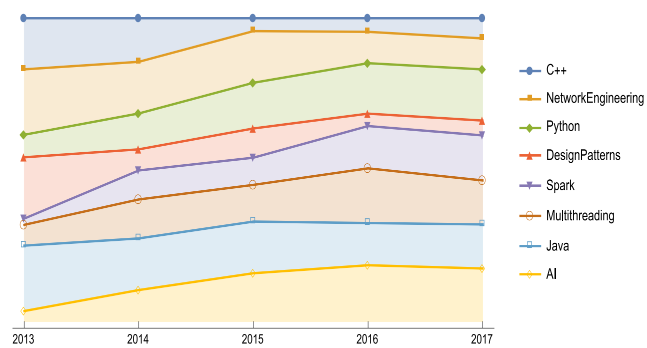


Figure 3: Trend of Skills Popularity Revealed by TATF.

Ablation Study

Then, for better understanding which kind of constraint contributes the most in the improvement, we turned to conduct the ablation study, which is summarized in Table 3 for the *MSE* metric. According the results, we realized that the co-occurrence-based aggregation, i.e., constraint based on skill packages could better improve the overall results. Besides, we found that with longer period, the strength of temporal constraint could be more significant. Obviously, more sufficient trend-aware factors lead to better performance.

Parameter Sensitiviness

Finally, we turn to evaluate the parameter sensitiveness. To be specific, two sets of parameters were discussed, namely $\mathcal{C}$ and $\mathcal{S}$ for aggregations, and K for the latent vector dimension.

First, discussion for aggregation parameters is shown in Table 4. According to the results, it seems that for both $\mathcal{C}$ and $\mathcal{S}$, a medium value around 40-60 might be the best choice, since more clusters (topics) lead to more overlapping, while less clusters (topics) may interfere the distinction.

Second, discussion for vector dimension K is shown in Figure 2(a), including performance of MAPE and R-Square, as well as time spending. According to the results, we interestingly found that generally a smaller K leads to a better result for all the three aspects, which may reflect that higher K could result in overfitting problem. Besides, our TATF performed the best on MAPE when $K = 10$, which was selected for the overall performance validation.

3.3 Discussion and Insights with Case Studies

As the effectiveness of TATF has been validated, we will further discuss about two interesting rules as follows.

Trend of Popular Skills. To better understand the evolving trend of skill popularity, we manually selected several representative skills as example, which are shown in Figure 3. Unsurprisingly, we realized that Big Data and AI-related skills are increasingly demanded in recent years, such as "*Python*" for data analytics, and "*Spark*" for large-scale data processing. On the contrary, some classic skills, e.g., "*C++*" and

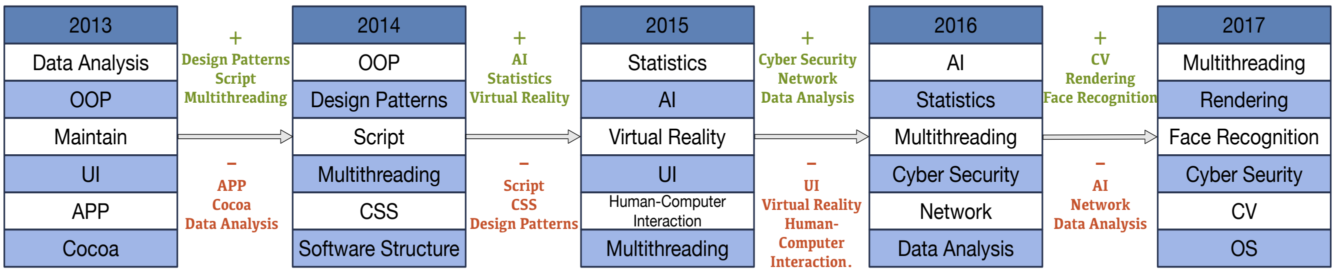


Figure 4: Evolution of "C/C++" related Skill Package during 2013-2017

"*Design Patterns*" are much less required. At the same time, some other skills like "*Java*" generally keep their demand level, as it is still widely used in Web and Android applications. In a sense, the popularity of skills are highly correlated to the importance of business field, as well as the diversity of application range, as skills with more diversified applications are more difficult to be replaced.

Evolving Skill Package. Similar phenomenon could be found in the evolving skill packages. As shown in Figure 4, the evolution of "*C/C++*" related skills are summarized year by year. Interestingly, we realized that though "*AI*" occurred during 2015-2016 as a novel technique, it was replaced by "*CV*" and even "*Face Recognition*" in 2017. To a certain degree, this case reflected the fine-grained trend of AI-related techniques. Recently, traditional statistical-based machine learning methods are facing the impact of Deep Learning based techniques, especially for CV and NLP related researches. At the same time, "*Cyber Security*" attracted wide attention to protect the data, which also supports the main idea that Data-related skills are highly demanded.

4 Related Work

In this section, we summarize related works in two aspects, namely *Recruitment Analysis* and *Tensor Factorization*.

4.1 Recruitment Analysis

As an essential part to support the success of organizations, recruitment analysis has attracted tremendous efforts. Traditional recruitment analysis highly depend of personal experience and simple statistics. For instance, [Maringe, 2006] investigated the influence factor of skill learning in different universities to provide recruitment strategies. However, they could only provide qualitative analysis without solid quantitative evidences.

Recently, thanks to the accumulation of recruitment data resource, data analytic techniques are now deeply integrated to benefit the recruitment market. On the one hand, job seekers are guided by personalized career path planning [Li *et al.*, 2017a] and talent circles analysis [Xu *et al.*, 2016; Xu *et al.*, 2015]. On the other hand, the employers are supported by job interview assessment [Shen *et al.*, 2018], talent flow analysis [Zhang *et al.*, 2019] and person-job fit [Wilden *et al.*, 2010; Zhu *et al.*, 2018; Qin *et al.*, 2018]. Besides, some skill-oriented researches were proposed, including the popularity ranking of skills [Xu *et al.*, 2018] and skill requirement trends analysis [Zhu *et al.*, 2016]. However, few of them have focused on the fine-grained skill demand analysis task, which could be effectively achieved by our TATF framework with comprehensive constraints.

4.2 Tensor Factorization

Due to its effectiveness and interpretability, Tensor Factorization (TF) has been widely used in many research fields, e.g., tag recommendation [Karatzoglou *et al.*, 2010; Yu *et al.*, 2018], social media analysis [Hong and Jung, 2018] and patent litigation[Liu *et al.*, 2018]. In order to capture the unique characteristics of different data source, TF methods were adapted following different assumptions. For instance, similar with the aggregation constraints utilized in our TATF framework, in BTF-GR [Wang *et al.*, 2018] method, user ratings are corrected based on the inner product between latent vectors of user and added group. Moreover, [Jiang *et al.*, 2014] utilized a flexible and dynamic factorization scheme to improve the effectiveness and decomposition efficiency. At the same time, comprehensive constraints, e.g., temporal correlations are always utilized in variant TF models to enhance the effectiveness, such as [Li *et al.*, 2017b].

In this paper, we intuitively proposed the TATF framework based on comprehensive analysis, constraints for temporal correlation and various aggregations, so that fine-grained job skill demand analysis will be achieved.

5 Conclusion

In this paper, we propose a comprehensive data-driven framework called Trend-Aware Tensor Factorization (TATF) to deal with the job skill demand analysis, in which constraints for temporal correlation and various aggregations are integrated to improve the performance. Along this line, more accurate representation for skill requirement, as well as the evolving trend of skill package will be accurately achieved. Extensive experiments on real-world dataset demonstrated the effectiveness of our TATF framework, and further provide some interesting rules via intuitive case studies.

Acknowledgements

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 91746301, 61703386 and U1605251), and the Anhui Provincial Natural Science Foundation (Grant No. 1708085QF140).

References

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Ghadimi and Lan, 2013] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[Harshman, 1970] Richard A Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[Hong and Jung, 2018] Minsung Hong and Jason J Jung. Multi-sided recommendation based on social tensor factorization. *Information Sciences*, 447:140–156, 2018.

[Jiang *et al.*, 2014] Meng Jiang, Peng Cui, Fei Wang, Xinran Xu, Wenwu Zhu, and Shiqiang Yang. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1186–1195. ACM, 2014.

[Karatzoglou *et al.*, 2010] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[Li *et al.*, 2017a] Huayu Li, Yong Ge, Hengshu Zhu, Hui Xiong, and Hongke 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*, pages 917–925. ACM, 2017.

[Li *et al.*, 2017b] Xin Li, Mingming Jiang, Huiting Hong, and Lejian Liao. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. *ACM Transactions on Information Systems (TOIS)*, 35(4):31, 2017.

[Liu *et al.*, 2018] Qi Liu, Han Wu, Yuyang Ye, Hongke Zhao, Chuanren Liu, and Dongfang Du. Patent litigation prediction: A convolutional tensor factorization approach. In *IJCAI*, pages 5052–5059, 2018.

[Maringe, 2006] Felix Maringe. University and course choice: Implications for positioning, recruitment and marketing. *International journal of educational management*, 20(6):466–479, 2006.

[Qin *et al.*, 2018] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui 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*, pages 25–34. ACM, 2018.

[Rendle and Schmidt-Thieme, 2010] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 81–90. ACM, 2010.

[Shen *et al.*, 2018] Dazhong Shen, Hengshu Zhu, Chen Zhu, Tong Xu, Chao Ma, and Hui Xiong. A joint learning approach to intelligent job interview assessment. In *IJCAI*, pages 3542–3548, 2018.

[Tucker, 1966] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[Wang *et al.*, 2018] Jinkun Wang, Yuanchun Jiang, Jianshan Sun, Yezheng Liu, and Xiao Liu. Group recommendation based on a bidirectional tensor factorization model. *World Wide Web*, 21(4):961–984, 2018.

[Wilden *et al.*, 2010] Ralf Wilden, Siegfried Gudergan, and Ian Lings. Employer branding: strategic implications for staff recruitment. *Journal of Marketing Management*, 26(1-2):56–73, 2010.

[Xu *et al.*, 2015] Huang Xu, Zhiwen Yu, Hui Xiong, Bin Guo, and Hengshu Zhu. Learning career mobility and human activity patterns for job change analysis. In *2015 IEEE International Conference on Data Mining*, pages 1057–1062. IEEE, 2015.

[Xu *et al.*, 2016] Huang Xu, Zhiwen Yu, Jingyuan Yang, Hui Xiong, and Hengshu Zhu. Talent circle detection in job transition networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 655–664. ACM, 2016.

[Xu *et al.*, 2018] Tong Xu, Hengshu Zhu, Chen Zhu, Pan Li, and Hui Xiong. Measuring the popularity of job skills in recruitment market: A multi-criteria approach. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Yu *et al.*, 2018] Runlong Yu, Yunzhou Zhang, Yuyang Ye, Le Wu, Chao Wang, Qi Liu, and Enhong Chen. Multiple pairwise ranking with implicit feedback. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1727–1730. ACM, 2018.

[Zhang *et al.*, 2019] Le Zhang, Chen Zhu, Hengshu Zhu, Tong Xu, Enhong Chen, Chuan Qin, and Hui Xiong. Large-scale talent flow forecast with dynamic latent factor model. In *Proceedings of The Web Conference 2019*, pages 2312–2322, 2019.

[Zhu *et al.*, 2016] Chen Zhu, Hengshu Zhu, Hui Xiong, Pengliang Ding, and Fang 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*, pages 383–392. ACM, 2016.

[Zhu *et al.*, 2018] Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)*, 9(3):12, 2018.