Alternate *ACM* SIG Proceedings Paper in LaTeX

TestName A

ABSTRACT

TO BE DONE

1. INTRODUCTION

TO BE DONE

2. DATASET ANALYSIS

2.1 Data Collection

Zhubajie [1](ZBJ) is a famous online freelancing marketplace platform in China. Employers publish different kinds of human intelligent tasks(HITs) on ZBJ for freelancers to work on. There are three kinds of HITs in this platform: tender, pitch and piece-work. Tender is a kind of task that after the Employers publish a task with content and salary, freelancers submit description of themselves, and employer make a choice to work with one of the candidates. While a pitch task is a task that freelancers directly submit their works for employer to choose, and one or several of the candidates are picked. In a piece-work task, employer need different freelancers to fulfill a lot of micro-tasks. We crawl all the tasks and their corresponding freelancer submissions from ZBJ platform in 2015. We select eight categories of the tender and pitch HITs and tag them with five(need to be alternated) price bins according to the salary of the task.

2.2 Data Features

TO DO

3. CHOSEN WORK PREDICTION

In this section we will formulate the chosen work prediction problem in a tender or pitch HIT. We will then describe features of the freelancers, employers and submitted work.

[1]www.zbj.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2016 ACM. ISBN not known.

DOI: notknown

3.1 Problem Formulation

Considering employer $e \in E$ (Employer set) publishes a HIT, t_{me} which is the m-th HIT from the whole HITs set T. And a freelancer $u \in U$ (Freelancer set) submit the n-th work w_{uemn} to t_{me}. Our work is to predict the $top-k$ works which are going to be rewarded by e in t_{me} and belongs to category $cate_i$ and price bin $price_j$. If a work w_{uemn} is chosen , then the result $r(w_{uemn}) = 1$, otherwise $r(w_{uemn}) = 0$.

3.2 Data features for prediction

Freelancer features.

Freelancer features describe the characters of the freelancer u to enable us to acquire to judge the ability of u to fulfill a HIT.

Category number: We use $cateNum(u)$ to represent the number of categories a freelancer u worked on.

Number of works in related category: We use function $workNum(u, cate)$ to denote describe the category-related interest of the freelancer u.

Fraction of works in related category: Fraction of works in related category is used to describe the category-related interest of the freelancer u. Fraction of work in related category is
$workNumRate(u, cate_i) = \frac{workNum(u,cate_i)}{\sum_i workNum(u,cate_i)}$

Number of works in related price bin: We use function $priceNum(u, price)$ the price-related interest of the freelancer u.

Fraction of work sin related price bin: We use function $priceRate(u, price)$ to describe the price-related interest of the freelancer u. Fraction of work in related price bin is
$priceRate(u, cate_i) = \frac{priceNum(u,price_i)}{\sum_i priceNum(u,price_i)}$

System reputation: ZBJ platform gives a freelancer u a system reputation level $sysLevel(u)$ when u fulfilled certain HITs. The more works, that are submitted by u, get more reward from employers, the higher level $sysLevel(u)$ the u will get from the system.

Efficiency: Efficiency is the number of works that u submits in a limited time period. We use three functions $effi7d(u)$, $effi14d(u)$,$effi30d(u)$ to represent the efficiency of u corresponding to three different time periods which are 7 days, 14 days and 30 days. Efficiency is a character that show us the work ability and passion of a freelancer.

Category-related efficiency and Price-related efficiency: Similarly to efficiency, we calculate category-related efficiency and price-related efficiency in different categories or different price bins for a freelancer u as $cateEffi7d(u, cate)$,

$cateEffi14d(u, cate)$, $cateEffi30d(u, cate)$,
$priceEffi7d(u, price)$,
$priceEffi14d(u, price)$,
$priceEffi30d(u, price)$.

Quality: We calculate the number of works rewarded and the number of works costs and a success rate of a freelancer. $n_{suc}(u)$ denote the number of works worked by u that get rewarded, while $n_{fail}(u)$ denote the number of works worked by u that do not get rewarded even when the task is finished. So the number of works rewarded is $rewardedWorkCost(u) = n_{suc}(u)$; the number of works costs is $workCostNum(u) = n_{suc}(u) - n_{fail}(u)$; success rate is $sucRate(u) = n_{suc}(u)/(n_{suc}(u) + n_{fail}(u))$.

Category-related Quality and Price-related Quality Similarly to Quality, we calculate category-related Quality and price-related Quality in different categories or different price bins for a freelancer u : $cateN_{suc}(u, cate)$, $cateRewardedWorkCost(u, cate)$, $cateSucRate(u, cate)$, $priceN_{suc}(u, price)$, $priceRewardedWorkCost(u, price)$, $priceSucRate(u, price)$, .

Submitted Work Features.

Relative reputation: The system reputation level of a work is the the system reputation level of the freelancer who submits the work. We use $rank(w_{uemn}, t_{me})$ denote the system reputation level rank of work $repRank(w_{uemn})$ compared with other works of the task t_{me}. $repRank(w_{uemn}, t_{me})$ is the number of works whose system reputation level is bigger then the system reputation level of w_{uemn}. And we use $num(t_{me})$ denote the total number of works submitted to the task t_{me}. Relative reputation is then denoted by $relativeRep(w) = \frac{repRank(w_{uemn}, t_{me})}{num(t_{me})}$.

Response time: Response time $responseDate(w)$ is the count of days after a task t_{me} is published.

Relative response time: Similar to relative reputation, we use $resposeRank(w_{uemn})$ denote the time rank of a work w_{uemn} submiteed to t_{me}. The earlier a work is submitted the lower its ranker is. And the relative response time is $relativeResposeDate(w) = \frac{resposeRank(w_{uemn}, t_{me})}{num(t_{me})}$.

Employer Features.

Employer features describe the preference of a employer e to make a choice.

Reputation preference: We calculate the median and mean system reputation level of the works selected by the employer e: $medianRep(e)$, $meanRep(e)$.

Relative reputation preference: We calculate the median and mean relative system reputation level of the works selected by the employer e: $medianRelativeRep(e)$, $meanRelativeRep(e)$.

Response time preference: We calculate the median and mean system response time works selected by the employer e: $medianResposeDate(e)$, $meanResposeDate(e)$

4. METHODOLOGY

In this section we will firstly describe how we use the features to combine a final input feature matrix. And then we show the baselines we used in this paper to compare with our work. Finally, we will explain our method and the evaluations.

4.1 Feature matrix

Suppose that there is a work w, which is submitted to e's task t by u. And t's category is $cate_i$ and t belongs price bin $price_j$. Then we get the tree type of features $\overrightarrow{w}$, $\overrightarrow{u}$, $\overrightarrow{e}$ seperately.

$$\overrightarrow{w} = [relativeRep(w) \quad responseDate(w) \quad resposeRank(w_{uemn})]$$

$$\overrightarrow{u} = \begin{vmatrix} cateNum(u) \\ workNum(u, cate) \\ workNumRate(u, cate_i) \\ priceNum(u, price) \\ priceRate(u, price) \\ sysLevel(u) \\ effi7d(u) \\ effi14d(u) \\ effi30d(u) \\ cateEffi7d(u, cate) \\ cateEffi14d(u, cate) cateEffi30d(u, cate) \\ priceEffi7d(u, price) \\ priceEffi14d(u, price) \\ priceEffi30d(u, price) \\ n_{suc}(u) \\ rewardedWorkCost(u) \\ sucRate(u) \\ cateN_{suc}(u, cate) \\ cateRewardedWorkCost(u, cate) \\ cateSucRate(u, cate) \\ priceN_{suc}(u, price) \\ priceRewardedWorkCost(u, price) \\ priceSucRate(u, price) \end{vmatrix}^T$$

$$\overrightarrow{e} = \begin{vmatrix} medianRep(e) - sysLevel(u) \\ meanRep(e) - sysLevel(u) \\ medianRelativeRep(e) - relativeRep(w) \\ meanRelativeRep(e) - relativeRep(w) \\ medianResposeDate(e) - responseDate(w) \\ meanResposeDate(e) - responseDate(w) \\ abs(medianRep(e) - sysLevel(u)) \\ abs(meanRep(e) - sysLevel(u)) \\ abs(medianRelativeRep(e) - relativeRep(w)) \\ abs(meanRelativeRep(e) - relativeRep(w)) \\ abs(medianResposeDate(e) - responseDate(w)) \\ abs(meanResposeDate(e) - responseDate(w)) \end{vmatrix}^T$$

The Feature vector of work w is consist of the three vectors:
$vectorW(w_{uemn}) = [\overrightarrow{w} \quad \overrightarrow{u} \quad \overrightarrow{e}]$.
And with certain count of works, we get the final feature matrix : $featureMatrix(W) = \begin{vmatrix} vectorW(w_{uem1}) \\ vectorW(w_{uem2}) \\ ... \\ vectorW(w_{uemn} \end{vmatrix}$.

4.2 Baseline

We use four different means as baselines to compare with our model. We use random select, reputation $sysLevel(u)$, historical success rate $sucRate(u)$ and category-related historical success rate $cateSucRate(u, cate)$ to generate a rank list from all the works submitted to one task t separately. And then we choose the $top - s$ works to be the selected works for the task t, where s is demanded number of works corresponding to t.

4.3 Method and Evaluation

Firstly we use the feature matrix we present in section 3 as features and actually situation that whether a works w is chosen ($r(w_{uemn}) = 1$) or not ($r(w_{uemn}) = 0$) as label data, to train a regression model by three different methods, decision tree, random forest and linear regression. Then similarly to how we deal with the baseline, we use the regression of the model we trained with the feature matrix of test data as input to generate a rank list for the submitted works of each task separately and choose the $top - s$ works according to each rank list.

And to evaluation the result of our feature model and the baseline, we may use precision, accuracy and AUC. Further more we induce NDCG@k(Normalized Discounted Cumulative Gain) and precision@k from information retrieval. And in order to implement the evaluation, we alternate the top-m works to $top - (s + k)$.