Task Recommendation in Reward-Based Crowdsourcing Systems

Ayswarya R Kurup *, G P Sajeev†
Dept of Computer Science and Engineering
Amrita School of Engineering, Amritapuri
Amrita Vishwa Vidyapeetham
Amrita University
India
Email: *ayswaryarkurup16@gmail.com, †sajeevgp@am.amrita.edu

Abstract—**Crowdsourcing systems are distributed problem solving platforms, where small tasks are channelled to a crowd in the form of open calls for solutions. Reward based crowdsourcing systems tries to attract the interested and capable workers to provide solutions in return for monetary rewards. We study the task recommendation problem in reward based crowdsourcing platforms, where we leverage both implicit and explicit features of the worker-reward and worker-task attributes. Given a set of workers, set of tasks, participation, winner attributes, we intend to recommend tasks to workers by exploiting interactions between tasks and workers. Two models based on worker-reward based features and worker task based features are presented. The proposed approach is compared with multiple related techniques using real world dataset.**

Index Terms—**Task Recommendation, Crowdsourcing, task identification, probabilistic matrix factorization.**

I. Introduction

Crowdsourcing refers to the process of seeking inputs from a large group of online users called crowd via the Web-based platforms [1]. Facilitated by information technology, crowdsourcing platforms implement socio-technical systems to direct the benefaction of human resources, knowledge, or skillfulness into the creation of digital information products and solutions [2]. Reward-based Crowdsourcing systems are crowdsourcing models in which workers solve tasks in order to gain rewards, such as a certain amount of money. By rewarding for their successfully completed tasks, the workers gets a motivation to participate and attempt to produce good quality work. For example Mechanical Turk provides a way to pay workers on their platforms to help with a task [3]. A generic architecture for a reward based crowdsourcing system is depicted in Fig.1.

In reward based crowdsourcing systems employers submits their tasks to crowdsourcing platforms, crowd participates in tasks by selecting their preferred tasks and submitting their finished work to the crowdsourcing systems. The employer examines the submission and chooses the top one and the corresponding worker gets the reward.

In general, crowdsourcing systems are built upon the self selection principle, Organizations publicize their tasks to an open public on the web and interested individuals contributes for the completion of the task [4]. Apart from the conventional

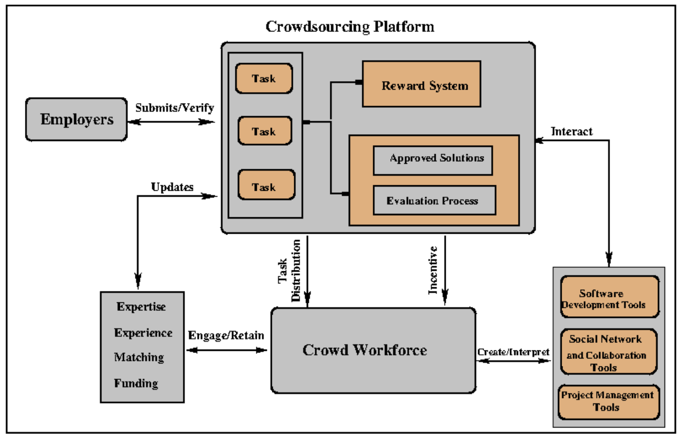

Fig. 1. A General Architecture for a Reward Based Crowdsourcing System.

models, this concept enables crowdsourcing systems to retrieve the extensive potential of contributors with diverse backgrounds. The value of this scalability and diversity provided by crowdsourcing approaches is utilized by many organizations for surplus applications. To make crowdsourcing more effective, several considerations are made. First, communication between the crowd and the requester must be clear and easy to follow. Since crowds need to have a common purpose, that is to earn some reward, the chances of a particular worker get paid depends on the quality of the work they submits [5]. Second, the problem or task being given to the crowd must also be broken down easily into independent subtasks. Third, crowdsourcing is most effective when firms have a recognized knowledge gap and can benefit from the collective insight of the crowd [6]. All together a good crowdsourcing platform should be equally beneficial to both the requester and crowd.

Eventhough the self selection principle allows crowds to select the tasks according to their preferences, it is tedious since the crowdsourcing platforms are in a information overloaded situation. Several studies [7], [8] show that the search costs of the crowd to find a task that matches their preferences is comparatively higher than that of complete a selected task. This reduces the motivation of the contributors to participate and make them to compromise with less suitable tasks which

978-1-5090-6367-3/17/$31.00 ©2017 IEEE

results in less quality works. Hence, it is significant for workers to recognize tasks in which they have more chances to have a success. Clearly, for an efficient crowdsourcing system it is not admirable that the time spend by a worker to choose a task is comparatively higher than that of time spend on completing a task, and the renumeration is just a pittance [1].

This situation make workers to suffer huge latency to find suitable tasks, which creates dissatisfaction and eventually leads to the abandonment of the platform. To handle this problem of incorrect task identification, task recommendation is used. Task recommendation problems is challenging in crowdsourcing platforms, since the objective is to recommend a set of tasks to each worker such that these tasks are best suited for them. Appropriate matching between workers and tasks is also a problem for crowdsourcing platforms, because workers get disappointed due to the low chance of winning the reward which results in significant differences in the worker's actual income. This may results in dissatisfaction of workers which makes them to leave crowdsourcing platforms results in future depopulation [9]. By using recommender approaches for personalizing the tasks, both the task identification and task matching problem is minimized.

Recommender systems are information filtering tools, which provide users with information which they are interested in based on their preferences. It reduces information overload as well as the problem due to great internet expansion by estimating relevance of an information. In our work, we deals with the recommender approaches to solve the issue of incorrect task identification and task matching problem in crowdsourcing systems.

In crowdsourcing systems, the straight forward utilization of the recommendation methods developed for traditional e-commerce systems would not result in satisfactory performance due to several reasons.

- The tasks in crowdsourcing systems may be completed at varied levels of quality, and thus could be rejected by the requesters of the tasks if the completion does not fulfill the required quality standard.
- The tasks in a crowdsourcing system usually have pretty short life span, of which many can be completed in minutes or seconds by experienced workers.
- Generally, a worker only picks up tasks that fit his/her skills and interest, typically are a much restricted subset of a potentially huge set of all tasks available in a crowd sourcing system.

To address the above mentioned, apart from the conventional recommendation systems, a novel approach is much needed, to deliver the required high real time performance of recommendations for both tasks and workers in crowdsourcing systems. When compared to unlimited mapping between products and buyers in an e-commerce system, the mapping between the tasks and the workers is much restricted in crowdsourcing systems. This helps in avoiding computing recommendations from the huge set of all possible pairing of tasks and workers. So for crowdsourcing systems, a reliable task recommender system is needed. The task recommendation is accurate and effective, if for the reommended tasks the worker is able to meet the submission quality and has a chances to get paid. Such a task recommendation system can be mutually favourable for both requester and worker.

To assist the potential workers to find the appropriate tasks following their personal choices, new perspectives to support the self-identification process are needed. In this paper we proposes a probabilistic latent factor based task recommendation model to suggest suitable tasks to workers based on their past performance history, preferences and participation information. In order to accurately recommend user preferred and trustful tasks to workers, rather than considering worker or task alone features we took the worker-reward and worker-task participation information for probability computation. This helps to minimize the problem of incorrect task identification.

Workers can choose suitable tasks from recommended lists which is provided based on their chances to win a task. Here in our work we exploits the features based on the workers ability to get a reward and preferences over a task. The idea is if the recommended list of tasks are similar to one which he has finished successfully, then the chances of completing the newly assigned task is higher. Our objective is to derive a probability model for predicting a task to be assigned for a worker by considering both worker-task and worker-reward as factors. Based on the probability, a latent factor model [10] associated with BPR, is used to recommend a list of suitable tasks. The proposed model is evaluated for performance by simulations.

The rest of the paper is structured as follows. Section II briefly describes the related research. Task identification and recommendation problem is formalized in Section III. The proposed solution is analysed in Section IV. In Section V, our method is compared with some baseline methods and the simulation results with real dataset are given. We conclude in Section VI.

II. State Of the Art

Several studies have been conducted regarding what motivates the workers in a crowdsourcing system to produce their best results. Silberman et al. [11] suggested that rather than other motivations remuneration plays a significant role, with most responders reporting that they do tasks for earn some reward and not for fun or to kill time. Various studies have been conducted regarding incentives in reward-based crowdsourcing, examines design of tasks with different available rewards in the form of contests. Yang et al [12] investigates how rewards influences the task selection process by workers. Harris [13] described that renumerations actually increases the quality of the job, if the task is assigned suitably based on skills.

Task suggestion in crowd sourcing have also attracted considerable interest [14] which optimizes the matching between tasks and workers. With the help of recommendation systems requesters can design the tasks with respect to their search target and the expected costs. In improvised participation context it is crucial yet challenging task to predict and manage

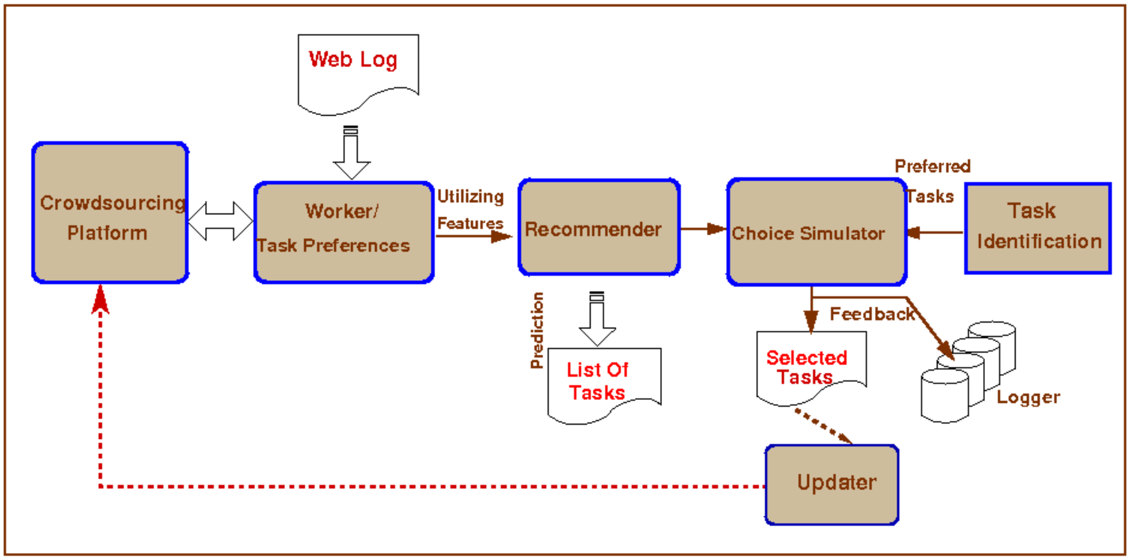

Fig. 2. Task Identification and Recommendation framework for a crowdsourcing system based on worker Preferences.

user involvement where users attention is limited. Several crowdsourcing platforms like mCrowd [3] and Amazon Mechanical Turk are pull-based systems which requires the user to actively search and select suitable tasks. In such systems user's attention is limited, furthermore it is an inappropriate and time consuming task for a worker since they have to search for appropriate tasks. Apart from pull-based systems, another is push sensing tasks to users by notifying potential participants with tasks recommendations.

Estimation of chances to get a worker rewarded is tedious because of the two significant features of crowd sourcing platforms:

1) Only one or few workers can be rewarded for a particular job,
2) And for the tasks of current interest no reward or winner is decided.

This is similar to the cold-start problem in recommender systems where information is very sparse to calculate affinity between workers or tasks, which degradates the performance prediction [15], [16]. To address this data sparsity problem in task recommendation the reserved information including features of workers and tasks, participation histories can be deployed [17]. To measure the similarity between workers and tasks, participation history is a good source of information.

Current Recommender Approaches For Crowd Sourcing Systems: The current recommendation methods centered around the two main recommendation approaches.

1) The Content based filtering,
2) And the Collaborative filtering.

In content-based recommendation approach, a set of descriptions of tasks previously rated by a worker is analyzed and forms a profile of user interests based on the features of the tasks that has been rated by that worker. Based on the profile new interesting tasks are recommended. The recommendation approach includes matching up the features of the content object against the attributes of a user profile. The result represents the workers level of interest in that task. The problems this approach faces include the following:

1) Some tasks may not be easily described using task descriptors,
2) Distinct tasks may share the same set of features described by the same keywords,
3) Profile information is not always available,
4) And poor performance scalability.

The collaborative filtering approach is probably the most successful and popular approach used in recommendation systems [18]. Comparing to content filtering, they depends only on the workers past behaviour. This method identifies hidden worker-task relationships by analyzing the relationships among workers and the correlations among tasks. This approach usually generates pretty accurate results because the learned worker-task relationships implicitly incorporate many subtle aspects that are hard to be explicitly profiled. The major drawback of this approach is the cold start problem because it relies on collected historical information that new workers/tasks do not yet have. For future crowdsourcing systems the number of workers and the tasks will be huge. Therefore, in principle this attractive approach will not result in real-time recommendation which is very much needed by future, very large-scale crowdsourcing systems.

When the feedback data is sparse, the conventional recommendation models gives limited performance, in recent works several authors proposed latent factor model with worker and task attributes [19], [20]. One of the work uses very high-dimensional task and worker attributes and their feature vectors are used to project to the latent space [21]. While worker ability and task difficulty are arguably important factors, many other factors may exist that previous models ignore. Latent factors are extremely predictive of the correctness probability that emerge naturally from the data [22]. From the literature survey we observe that, eventhough a lot of studies are conducted in this area none of them are considering probability of worker's ability to get a reward for the task and participation information as features for the prediction. Thus there is a need of further study in this area.

III. Proposed Work

Based on the historical data available on how many tasks a worker has successfully performed and those observations on what types of tasks and its features, we can lower the risk of incorrect task identification by asking workers to identify tasks from a list of recommended tasks based on their prior experience. The proposed model based on worker-reward probability and participation probability is depicted in Fig. 3

Inorder to predict the probability of a worker's chance to get success, we have to consider the preferences of the workers which is retrieved from worker's task performance records and worker's task searching history. From performance records we collects number of browsed tasks, number of selected tasks, number of tasks which are succesfully finished, total reward, number of tasks that is accepted by the employer and total percentage of accepted tasks for each worker. Worker task searching history gives the relationship between a worker and a task, which measures the worker's interest in a particular task. These information can accurately reflect the worker's preference and the features can be represented as Worker-Task matrix. For eg, the Worker-Task matrix given in Fig. 3 illustrates the extent of the favor of each task for each worker on a five-point integer scale. The Worker-Task matrix gives partial observations, which we decomposes using Probabilistic Matrix Factorization to get complete features of the matrix.

A. Problem Definition

Assume that there are J tasks and I workers in a crowdsourcing system. We have to consider two types of logged data, information about the tasks that the worker successfully finished (got a reward) and the lists of tasks in which the worker participated. Let $R_{I\times J}$ denote the matrix which represents that ith worker got a reward for jth task. $R \in \{0,1\}^{I\times J}$ denotes if the ith worker got rewarded for jth task ($R_{ij} = 1$) or not ($R_{ij} = 0$). Let J_f denotes the list of finished tasks where the approved job is already declared. ie $R_{ij} \in \{0,1\}$ for all $j \in J_f$. J_n represents the set of unfinished tasks which have not yet been approved. Let σ_j denotes the index of the worker who successfully finished the jth task.

Let P be the participation matrix $P \in \{0,1\}^{I\times J}$ which denotes if the ith worker participated in a jth task ($P_{ij} = 1$) or ($P_{ij} = 0$). Let I_i be the list of workers who participate in the jth task, for all $j \in J_f$, $P_{ij} \in \{0,1\}$ and $P_{\sigma ij} = 1$, since the worker who got reward is selected from the set of participants of each task.

Each worker has his own profile and each task has its own description. Let worker's profile description is associated with an N dimensional feature vector and task with an M dimensional feature vector. Let $\delta \in \{0,1\}$ denotes the feature

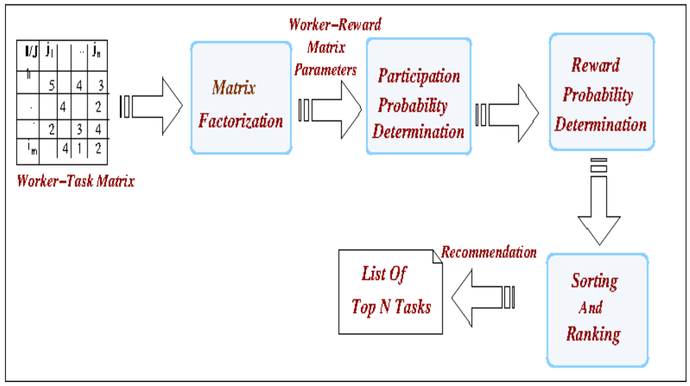

Fig. 3. Proposed Recommendation Model based on worker-reward probability.

matrix of worker, where δ_{in} represents the value of the nth feature of the ith worker and $\gamma \in [0,1]$ is a task-feature matrix where γ_{jm} denotes the value of mth feature of the jth task. Here our aim is to:

- Anticipate the probability of worker i getting a success for a jth task for all $j \in J_n$,
- Recommend a ranked list of tasks to each worker according to this probability.

B. Design approaches for encouraging user contributions

Based on the assumption that if tasks similar to one in which the worker previously won can be recommended, then chances for worker to gain is high. We starts with a determination model to anticipate the probability of the worker getting a reward, which has a crucial role in the recommendation process. Based on the assumption that the probability of a worker winning the task depends on the comparative performance of the worker to that of the other workers who participated in that particular task. In that case, we need a participation probability model which considers the cases where the profile of workers in a crowdsourcing system is partially known.

1) Worker-Reward based Probability determination: First, consider the expected performance of worker i for task j, denoted by $L_{ij} \in R$. The performance of a worker on a particular task depends on two factors: the affinity A_{ij} between worker i and task j and the potential b_i of worker i. The performance L_{ij} can be modelled as the summation of the worker potential and affinity between the worker and task which can be expressed as

$$L_{ij} = A_{ij} + b_i$$

The worker bias, b_i, captures potential of the worker i which is task-independent, and A_{ij} models the diverse worker performance depending on the combination of worker i and task j. Here task factor is not considered because it is common to all workers who are participating in that task.

The winning probability of worker i in a task j is defined as a normalized exponential function of the expected performance. Let the participation information be denoted by $Z_{ij} \in \left[0,1\right]$ for all j, then the probability of a worker get rewarded for a particular task is defined as

$$W\left[\ R_{ij} = 1\right] = \frac{Z_{ij}exp(L_{ij})}{\sum_p Z_{pj}exp(L_{pj})} \quad (1)$$

If $Z_{ij} = 1$ it is expressed as

$$= \frac{exp(L_{ij})}{\sum_p \in z_j exp(L_{pj})}$$

otherwise 0. ie unless a worker engages in a task the chance to get a reward is zero. The value of Z_{ij} depends up on the comparative performance of the worker to that of others who are particpating in that task. So if the number of workers is small the probabilty of getting a reward for that worker will be high. Another possibility is that the participating workers may not finish the tasks, in that case the probability is expressed as

$$\left[Z_{ij} = 1\right] = \frac{W\left[Z_{ij} = 1\right]exp(L_{ij})}{\sum_p (W\left[Z_{ij} = 1\right])exp(L_{pj})} \quad (2)$$

$W\left[Z_{ij} = 1\right]$ denotes the probability of worker i participating in task j.

2) Worker-Task participation probability determination: The willingness of a worker to choose a particular task is not influenced by other workers. In that case the participation probability is expressed as

$$W\left[Z_{ij} = 1\right] = \frac{1}{1 + exp(-B_{ij})} \quad (3)$$

where B_{ij} is the willingness of a worker i to choose a task j. B_{ij} is composed of several factors,

$$B_{ij} = A_{ij} + b_i + c_j + d$$

A_{ij} is the worker i's preference of contest j, b_i is the efficiency of worker i, c_j is the benefits or attractiveness of task j, the bias constant d. We uses the worker bias b_i and task bias c_j to get the worker and task preferences (independent) respectively. To overcome the data sparsity problem the feature vectors of workers, tasks and participation information is used as auxilliary information. Here we presented two probability models to determine participation probability and winning probabilty, further these models are analysed based on latent factor model.

IV. Analysis of the Proposed Solution

A. *Probability based Latent Factor Models*

To get the worker-task matrix information we employ a probabilistic matrix factorization model. Latent factor models have gained popularity due to their scalability and accuracy. The main advantage is a latent factor model maps a worker i and task j into the same $I \times J$ dimensional latent factor space. The precise estimation of the parameters of worker who not yet got a reward is challenging here. This cold start problem can be tackled by using the auxilliary information. To reduce the sparsity in worker-reward matrix R we can use the participation matrix Z because Z is relatively denser than R. Our goal is to predict for each user i, a ranked list of tasks j. The prediction for a worker-task pair is given by the dot product of the worker-reward factor vector and the worker-participation factor vector. Assume that each user i and task j can be represented by latent factors v_i and v_j respectively, which are vectors of $1 \times N$. Worker's affinity A_{ij} is expressed as $A_{ij} = \left[\ V_i, V_j\right]$.

Here the problem is to calculate the ideal values for v_i and v_j based on the given worker-task(participation) matrix Z_{ij}. While traditional approaches to matrix factorization attempt to revert over the known entries of the matrix [23], a better approach is the Bayesian personalized ranking (BPR) matrix [24]. Here, the idea is to perform regression straightly over the ranks of the tasks, because our goal is to build a ranked lists of n tasks. Also, we have an implicit feedback from the workers. The main objective to use BPR is to discriminate between tasks that the worker won from those tasks that were not. ie, a ranking function R_i for each worker i has to be learned for that i's interesting tasks higher than the non-interesting items. Particularly, if task i appears in worker is matrix R_{ij} and task k does not appear in R_{ij}, then we must have $R_i(j) > R_i(k)$. For this, we need to have the affinity $A_i \in Iij > A_j \in Iij$.

To estimate both Worker-Reward based probability and Worker-Task based probability we can use MAP estimation (Maximum a Posteriori Estimation) method. The likelihood function $D(R_i \mid Z)$ defined as

$$D(R_i \mid Z) = \Pi_{i \in I} \Pi_{j \in R_i} \Pi_{k \in R_i} \sigma(A_{ij} - A_{ik})$$

Using the Gaussian prior $N(0, \sigma)$ over all the factors in Z the MAP estimate of Z can be computed. The posterior over Z is given by

$$D(Z \mid R_i) = D(Z)D(R_i \mid Z)$$

As mentioned earlier to tackle the data sparsity we use the auxilliary information for the MAP estimation. Here the parameters we consider are A (worker-task preference) and b (worker-efficiency) for worker-reward probability determination model. Therefore

$$D(A, b) = \sum_{j \in J_f} \log(W\left[\ R_{\sigma i,j} = 1\right]$$

$$= \sum_{j \in J_f} (A_{\sigma,j} + b_{\sigma,j} - \log \sum_{p \in I} exp(A_{pj} + b_p))$$

Here the parameters are determined for only the set of finished tasks j_f using MAP estimation. For tasks in the set of unfinished tasks j_n it is hard to precisely determine the parameters. To tackle this problem we uses the auxilliary information from the feature vectors.

Similarly for estimation of Worker-Task based probability determination model, the auxilliary information A, b, c (attractiveness of task), and the bias constant d are utilized. Therefore the likelihood function is defined as

$$D(A, b, c, d) = \sum_{i,j \in J_f} \log(W[\ Z_{\sigma i,j} = 1])$$
$$+ \sum_{i,j \in J_f} \log(1 - W[\ Z_{\sigma i,j} = 1])$$

Let P denote a set of (i, j) for which $Z_{\sigma i,j} = 1$, and N denote a set of (i, j) such that $Z_{\sigma i,j} = 0$.

$$= \sum_{i,j \in P} \log(1 + exp(B_{ij})$$
$$- \sum_{i,j \in N} (B_{ij} + \log(1 + exp(-B_{ij})))$$

We apply matrix factorization on D which is obtained from Worker-Task based probability determination model. This results is used for determining the probabilty of Worker-Reward.

An abstract view of task recommendation can be stated as:

1) Apply the matrix factorization based on the feature-based matrix to the worker-task matrix Z by maximizing D,
2) obtain the worker-reward probability determination matrix parameters by using D,
3) determine the probability of worker i getting a reward for task j,
4) the tasks of present interest $(j \in J_u)$ can be sorted for each worker, based on the probability and recommend the top n tasks.

The parameters of worker-reward probabilty determination model will get updated when a new worker or a new task arrives to the system. It is computationally expensive to recompute all the parameters for each time. The parameters pertaining to worker-task determination need to be updated periodically since the parameters of worker and task are stable.

V. Simulations

Inorder to assess the proposed model, we conduct simulations on an Intel i7 quad core CPU running at 2.8 GHz, 16GB of RAM with Ubuntu 16.04 Operating System. The code was written and compiled in Python environment. We used a real crowdsourcing dataset containing details of 423 workers along with number of tasks and their participation details in each task.

TABLE I
Statistics of datasets

Parameters	Dataset summary
Number Of Workers(I)	423
Number of Tasks(J)	5613
Worker Feature's dimension(N)	2560
Task Feature's dimension(M)	6321
Number of elements in worker-reward matrix	5613
Number of elements in Worker-Task Matrix	17, 109
Average of the number of tasks each worker participated	23.85
Average number of workers in each task	2.90

A. Dataset

Data is collected from CrowdFlower [22], which is a well known crowdsourcing platform. CrowdFlower dataset is open source from which the observations from 2014 onwards has been collected and a set of 423 workers is randomly choosen for the analysis along with 5613 jobs. From this we summarized our required data that correspond to the features of the matrices used. The details of the dataset are depicted in Table I.

Task and worker features are included in the dataset. The task features taken into account are category of the task, employer ID, reward amount, worker skill options, job description. Worker features includes worker ID, number of tasks finished, worker skills, number of tasks get paid. From this details we extract the binary values corresponding to feature vectors of both tasks and workers. To balance the values we used $one - of - N$ coding [17].

B. Performance Comparison with Other Models

The proposed approach is compared with one of the baseline algorithm, and two variants of probabilistic matrix factorization model. We choose Linear Regression Based Latent Factor Model (feature-based) as the baseline algorithm and the Matrix Factorization model includes Worker-based and Task-based approaches. For the baseline algorithm, we assume that task feature matrix is given, based on which the list of tasks is predicted. The results of this model compared with our proposed solution using PR (Precision-Recall) Curve.

For the Matrix factorization models, the simulations are carried out using implicit feedback dataset. In W-based model, matrix factorization is performed on the Worker-preference matrix and in T-based model, matrix factorization is performed on Task-preference matrix utilizing [25]. For a worker $i \in I$

and task $j \in J$, let the worker-factor vector be x_i and task-facor vector be y_j. Then the prediction is given by the dot product as $r_{ij} = {x_i}^T.y_j$. Based on [25] matrix factorization equation for worker-preference and task preference is taken as

$$min_{xy} \sum f_{ui}(r_{ij} - {x_i}^T.y_j)^2 + \lambda((| \, x_i \, |)^2 + (| \, y_j) \, |)^2$$

where λ is the regularization parameter and f_{ui} is the confidence for each observation. The above mentioned models are compared with our algorithm using MPR and the results are depicted in Fig. 4.

C. Evaluation Metric

Evaluating the implicit feedback recommender approaches requires appropriate measures. An evaluation methodology based on Mean Percentile Ranking (MPR) can be used since we have to take in account competition between workers over others, participation and repeat feedback. MPR is a recall-based evaluation metric that evaluates a worker's preferences over a list of recommended tasks [22].

MPR is defined as

$$\frac{\sum_{ij} C_{ij} P_{ij}}{\sum_{ij} C_{ij}}$$

where p_{ij} is the percentile ranking of the task j for worker i. In our model recommendation is based on worker-reward probability matrix which is estimated from worker-participation probability matrix, $P = RZ$, where R is the worker-reward matrix and Z is the worker-task matrix. For evaluating our methods, 90 percentage of the data is choosen as the training set and the left over 10 percentage as the test set. All the results are an average of four runs. Fig. 4 gives the compared results of the W-based, T-based and Reward-based models. As a second evaluation method PR curve (Precision Recall curve) is used. This method is used to evaluate for how many tasks in the test set, the probability can be correctly anticipated by taking top n percentage from the recommended tasks. Value of n had been changed accordingly to obtain the PR curve. The results of PR curve for Feature-based model and Reward-based model is given in Fig. 5. PR curve for Matrix Factorization models and Reward-based model is depicted in Fig. 6. From PR curve it is observed that Reward-based model gives better precisions than other models.

D. Results

We simulate the Feature-based-regression model, Matrix Factorization models, and our proposed model, with varying number of latent factors. Table. II gives the MPR results for Feature-based-regression model, Matrix Factorization models, and our proposed approach. It is observed that reward based model gives better performance compared to other models.

The results obtained for the two matrix factorization models (W-based, T-based), Feature-based-regression model and

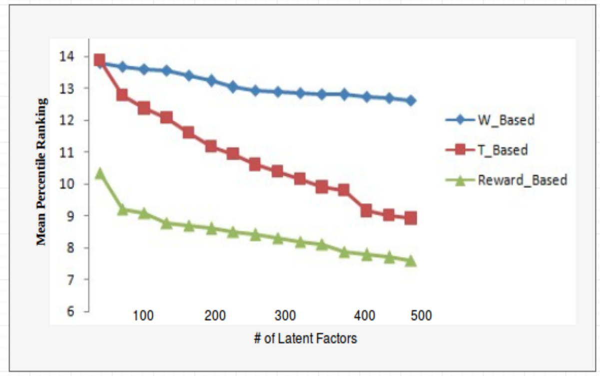

Fig. 4. A comparison of Reward Based Model with Matrix Factorization models based on Worker preference and Task preference.

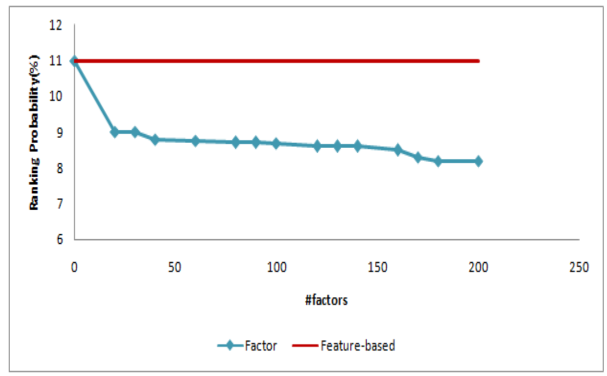

Fig. 5. Comparison of Reward based latent model with Feature-regression based model using PR Curve.

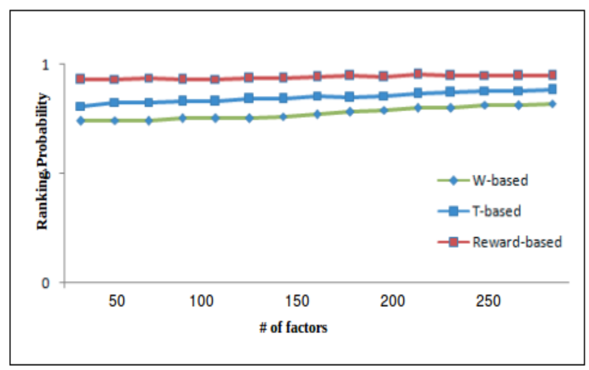

Fig. 6. Comparison of Reward-based model with Matrix Factorization models using PR Curve.

TABLE II
MPR RESULTS

Algorithm	MPR
Feature-based_regression	12.906
Reward-based	**7.68**
Matrix Factorization W-Based	11.60
Matrix Factorization T-Based	8.91

Measure	Prediction(N=100)	Percentage of completed tasks				
		15%	30%	50%	75%	90%
MAP_2	Feature_ based	0.767	0.812	0.822	0.832	0.861
	Factor_based	**0.891**	**0.920**	**0.912**	**0.909**	**0.902**
	W_based	0.857	0.905	0.876	0.882	0.891
	T_based	0.838	0.917	0.899	0.901	0.903
MAP_3	Feature_ based	0.803	0.816	0.835	0.847	0.851
	Factor_based	**0.856**	**0.873**	**0.882**	**0.893**	**0.903**
	W_based	0.832	0.843	0.851	0.857	0.863
	T_based	0.838	0.862	0.873	0.884	0.897
MAP_5	Feature_ based	0.798	0.812	0.818	0.827	0.843
	Factor_based	**0.814**	**0.827**	**0.858**	**0.887**	**0.910**
	W_based	0.802	0.815	0.829	0.854	0.892
	T_based	**0.814**	0.819	0.837	0.863	0.902

Fig. 7. The results obtained for task recommendation using MAP estimation for varying percentages of completed tasks. The results shown is for $N = 100$, reward based method outperformed other methods.

Reward-based model using PR are given in Fig. 7. For varying percentages of tasks completed by the workers, the values are estimated for all the four approaches. And Reward-based approach gives comparatively better performance.

VI. Conclusion

This paper has proposed a novel task recommendation model for reward based crowdsourcing applications, considering both implicit feedback and explicit features. The task based and reward based probabilities are derived and analysed its effectiveness. Simulations are carried out in comparison with matrix factorization models (W-based, T-based), Feature-based-regression model and Reward-based model and it is observed that the proposed model performs better and offers reduced data sparsity. The participation information could be effectively utilized for predicting the reward gain, using feature vectors for the worker and task. Adding new features to our proposed model may improve the accuracy of the recommendations.

References

[1] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio, "Crowdsourcing: a review and suggestions for future research," *International Journal of Management Reviews*, 2017.

[2] D. Geiger and M. Schader, "Personalized task recommendation in crowdsourcing information systemscurrent state of the art," *Decision Support Systems*, vol. 65, pp. 3–16, 2014.

[3] S. Schnitzer, S. Neitzel, S. Schmidt, and C. Rensing, "Perceived task similarities for task recommendation in crowdsourcing systems," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 585–590.

[4] D. Geiger, *Personalized task recommendation in crowdsourcing systems*. Springer, 2016.

[5] M. Yin, M. L. Gray, S. Suri, and J. W. Vaughan, "The communication network within the crowd," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 1293–1303.

[6] A. Jain, A. D. Sarma, A. Parameswaran, and J. Widom, "Understanding workers, developing effective tasks, and enhancing marketplace dynamics: a study of a large crowdsourcing marketplace," *Proceedings of the VLDB Endowment*, vol. 10, no. 7, pp. 829–840, 2017.

[7] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot, "Task search in a human computation market," in *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 2010, pp. 1–9.

[8] V. Chandrasekaran, S. V. Rajan, R. K. Vasani, A. Menon, P. B. Sivakumar, and C. S. Velayutham, "A crowdsourcing-based platform for better governance," in *Proceedings of the International Conference on Soft Computing Systems*. Springer, 2016, pp. 519–527.

[9] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM, 2010, pp. 13–22.

[10] P. Nedungadi and T. Smruthy, "Personalized multi-relational matrix factorization model for predicting student performance," *Intelligent Systems Technologies and Applications*, vol. 1, p. 163, 2015.

[11] M. Silberman, L. Irani, and J. Ross, "Ethics and tactics of professional crowdwork," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, pp. 39–43, 2010.

[12] J. Yang, L. A. Adamic, and M. S. Ackerman, "Crowdsourcing and knowledge sharing: strategic user behavior on taskcn," in *Proceedings of the 9th ACM conference on Electronic commerce*. ACM, 2008, pp. 246–255.

[13] C. Harris, "Youre hired! an examination of crowdsourcing incentive models in human resource tasks," in *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 2011, pp. 15–18.

[14] J. Shu and X. Jia, "Secure task recommendation in crowdsourcing," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[15] C. Shyamala, S. Hemaashri, and R. Swetha, "An improved recommendation system for social networks," *International Journal of Control Theory and Applications.(ICSCS). IJCT A International Science Press*, vol. 8, no. 5, pp. 1903–1910, 2015.

[16] V. Kavinkumar, R. R. Reddy, R. Balasubramanian, M. Sridhar, K. Sridharan, and D. Venkataraman, "A hybrid approach for recommendation system with added feedback component," in *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*. IEEE, 2015, pp. 745–752.

[17] Y. Baba, K. Kinoshita, and H. Kashima, "Participation recommendation system for crowdsourcing contests," *Expert Systems with Applications*, vol. 58, pp. 174–183, 2016.

[18] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[19] M.-C. Yuen, I. King, and K.-S. Leung, "Taskrec: A task recommendation framework in crowdsourcing systems," *Neural Processing Letters*, vol. 41, no. 2, pp. 223–238, 2015.

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

[21] M.-C. Yuen, I. King, and K.-S. Leung, "An online-updating algorithm on probabilistic matrix factorization with active learning for task recommendation in crowdsourcing systems," *Big Data Analytics*, vol. 1, no. 1, p. 14, 2016.

[22] H. Rahman, L. Joppa, and S. B. Roy, "Feature based task recommendation in crowdsourcing with implicit observations," *arXiv preprint arXiv:1602.03291*, 2016.

[23] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 19–28.

[24] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. Smola, "Scalable inference in latent variable models," in *International conference on Web search and data mining (WSDM)*, vol. 51, 2012, pp. 1257–1264.

[25] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 2008, pp. 263–272.