Worker Skill Estimation from Crowdsourced Mutual Assessments

Shuwei Qiang
The George Washington University

Amrinder Arora
BizMerlin

Current approaches for estimating skill levels of workforce either do not take into account the expertise of the recommender, or require intricate and expensive processes. In this paper, we propose a crowdsourcing algorithm for worker skill estimation based on mutual assessments. We propose a customized version of PageRank algorithm wherein we specifically considered the expertise of the person who made assessments. By implementing our algorithm on 15 real-world datasets from organizations and companies of varying sizes and domains and by using leave-one-out cross validation, we find that the results are highly correlated with the ground truth in datasets.

INTRODUCTION

Problem Space

In this paper, we study expertise management in organizations and focus on worker skill estimation problems. An efficient algorithm which can successfully assess worker abilities has dozens of applications, such as project team formation, resource planning and succession planning.

To clarify the scope of our research, we first define three key areas in the expertise management problem.

1. Skill modeling: In a knowledge-intensive crowdsourcing organization, each worker has different skills in various domains (Roy, et al., 2015, pp. 467-491). In order to accomplish a task which requires some specific skills, we first need to identify which skill each worker has. Skill modeling is a method to label and construct skill models. A naive approach would use succinct descriptions to tag different skills. Other well-designed methods can be constructing hierarchical skill trees (Mavridis, et al., 2016, pp. 843-853).

2. Skill estimation: Workers may have different levels of proficiency for a particular skill. Skill estimation grades a worker's skill by a deterministic value or provides a probability distribution function to describe the worker's performance (Rahman, et al., 2015, pp. 1142-1153).

3. Automated team formation: Given a pool of workers with different skills and a task with some requirements, we need to select right team members to undertake the task. Our choice should satisfy various criteria and give us optimized results. In real implementations, besides skill requirements and skill levels, we need to consider more variables, such as success probability, cost of failure, team coordination scores and cost overheads (Anagnostopoulos, et al., 2012, pp. 839-848).

Although many expertise management processes highly rely on the the second part, worker skill estimation, accurate skill estimation of individual workers is known to be a complex problem. In this paper, we focus solely on solving the skill estimation problem and propose an feasible solution.

Review of Related Work

Skill estimation has various approaches. One recent work is reconstructing workers' skill level based on evaluation results of tasks undertaken by different groups of people (Rahman et al., 2015, pp. 1142-1153). The models considered in that work mainly contain two skill aggregation functions. 1. The SUM function: The skill of a team is defined as the sum of skill levels of individual workers. 2. The MAX function: The skill of a team is defined as the maximum value of skill levels in that team. This work depends on the data of completed tasks, which in their setting come from basketball games and published papers. In many industries, such data may come from the completion of "projects", and the suggested model may then be applicable.

An entirely different mechanism of assessing skills is to allow workers to endorse each other for skills and use this data to estimate skill levels. Popular social media website LinkedIn provides a form of this endorsement functionality by allowing two connected members to endorse each other's skill ("Skill Endorsement - Overall," 2016). The level of skill is assumed to be proportional to the number of endorsements on that skill by the user's first degree connections. However, this model does not take into account the rating of the person giving the endorsement. For example, consider the following situation: User u_1 gets 10 endorsements on a particular skill, suppose, "Impressionist Painting" from his friends who themselves do not have any knowledge of art. User u_2 gets 5 endorsements on the same skill by 5 different world-renowned painters. It certainly appears dubious to conclude that user u_1 has higher level of "impressionist painting" skill than user u_2 .

In Ding (2011), authors apply PageRank algorithm to the field of citation analysis. Their experimental data showed that PageRank algorithm is a robust and accurate measurement of scientific papers. They treat citations in one paper as outgoing links to other papers and citations from other papers as incoming links of this paper. Reagans, and Zuckerman (2001), also cover some similar ideas. Another paper shows that ranking authors in a co-citation network based on PageRank indicators gives us a valid result which is similar to normal citation rank (Ding, et al., 2009, pp. 2229-2243).

Moreover, Zhang, Ackerman, and Adamic (2007), use PageRank algorithm and HITS algorithm to compute expertise level of users in a Java question and answer forum. It concludes that network structural characteristic matters when we evaluate expertise of users in a system. Their experimental result also shows that PageRank algorithm does nearly as well as human raters in expertise ranking.

We believe that a precise estimation should use as much information as the crowdsourcing system could directly or potentially provide to us. So to give an efficient algorithm, we not only consider quantity value in each mutual assessmentss, but also the quality of each assessment. Two underlying reasonable assumptions of this work are: 1. Workers with high skill level are likely to receive endorsements with higher scores from other workers. 2. Evaluations from high skill level workers are more important than evaluations from low skill level workers. These assumptions are consistent with real-world data models. To take advantage of our crowdsourcing system, we collect mutual assessment data and derive estimation scores for workers. At a high level, our algorithms take results of crowdsourced mutual assessment as input and output estimation scores on each worker's skill.

Structure of Paper

The rest of the paper is organized as follows. In the following section, we present the underlying data model to formalize the problem. Then in the section for algorithm and implementation, we describe how we apply the original PageRank algorithm to our problems and introduce a customized version. We also discuss implementations and convergence properties of those algorithms in that section. In the experiment section, we use cross validation to evaluate the accuracy of our result and demonstrate how the dumping factor influences the accuracy. Besides, we also present experimental data to show our algorithms is efficient for a large organization. Finally, we make our conclusion, discuss features and limitations of our mechanism and highlight some possible future works.

DATA MODEL

We use the following data model. Assume we have ***n*** workers. We notate them as a set $W = \{w_1, w_2, . . . , w_n\}$, in which w_i represents the i^{th} worker in our worker pool. Similarly, we have a set $S = \{s_1, s_2, . . . , s_m\}$ to represent ***m*** different skills, and a set $T = \{t_1, t_2, . . . , t_l\}$ to represent l different timings at which assessments are made. Following are the input and the output of our algorithm.

Input: A four dimensional array $A_{n \times n \times m \times l}$ where each element is a numerical score in some predetermined scale, recording every assessments. The element a_{ijkq} is the score that w_i gives to w_j for skill s_k at time t_q. If $a_{ijkq} = 0$, it means either w_i does not assess w_j's skill at t_q or w_i does not think w_j has the skill s_k. We call array A the assessment matrix. We consider that clearly the assessment matrix is a very sparse matrix, which reflects the practical observation that only a few workers endorse other workers, and that too, for only a subset of skills.

Output: A two dimensional array $V_{n \times m}$ where each element v_{ij} is the numerical score in some pre-determined scale that represents estimated skill level of worker w_i for skill s_j. We call array V the skill estimation value matrix.

ALGORITHM AND IMPLEMENTATION

Generic mutual assessments include two typical types: one is that workers use 1 or 0 to indicate whether they endorse others' skills or not, another is that each of them gives a numerical evaluation to others' skills in a pre-agreed scale. In our paper, if our input is the first type, we call it the endorsement problem, otherwise we call it the evaluation problem.

In case we have multiple assessments for the same combination of sending worker, receiving worker and skill, we take the latest assessment as the valid assessment, because we are only interested in estimating the current skill level. We also assume skills are independent from each other, meaning an estimation on one skill will not effect estimations on other skills for the same worker. In order to make the description of our algorithm compact, we focus on estimating just one single skill, because the generalization of multiple skills is relatively trivial. Therefore we can simplify our input model as a matrix $A_{n \times n}$ and the output model as one vector $v = \{v_1, v_2, . . . , v_n\}$.

Endorsement Problem

Using the data model built before, we add an additional constraint in this version of problem: $a_{ij} \in \{0, 1\}$. Inspired by the idea of PageRank (Page, et al., 1999), we can define the skill estimation function of our algorithm as:

$$v_i = \beta \sum_{j=1}^{n} \frac{v_j}{d_j} + \frac{1 - \beta}{n} \tag{1}$$

In above equation, d_j means the total number of endorsements made by w_j, so

$$d_j = \sum_{i=1}^{n} a_{ji} \tag{2}$$

and β is called the damping factor which is in the interval $[0, 1)$. Our definition assumes that every worker endorses at least one other worker, such that $d \neq 0$ for any worker. If we present the input as a directed graph, d_j is the out degree of node j. We handle the case in which the out degree of some nodes are zero when we discuss actual implementations. We postpone the choice for factor β in later section.

Because our algorithm is defined in a recursive way like PageRank, a naive recursive implementation will fail. To implement the algorithm, we first need to preprocess our data into the form of Markov chains and then use power iteration to compute results (Leskovec, Rajaraman, & Ullman, 2014). Now we briefly describe the preprocessing step for primary data.

Given input matrix A, we define A' as:

$$\mathrm{A}' = \mathrm{A} + h^T e \tag{3}$$

The row vector h is totally dependent on A:

$$h_i = \begin{cases} 1 & if \quad \sum_{j=0}^{n} a_{ij} = 0 \\ 0 & if \quad \sum_{j=0}^{n} a_{ij} \neq 0 \end{cases} \tag{4}$$

and h^T is its transpose matrix. The vector e is a special $1 \times n$ matrix in which all elements is equal to 1. Then we can compute a column vector d:

$$d = \mathrm{A}' \cdot e^T \tag{5}$$

We take the reciprocal of each element in d to define matrix B and let element

$$b_{ij} = \frac{1}{d_i} \tag{6}$$

so we have a $n \times n$ matrix B. Then we define another $n \times n$ matrix C as the Hadamard product of matrix A and B:

$$\mathrm{C} = \mathrm{A} \circ \mathrm{B} \tag{7}$$

Because it is an element-wise product, it defines element:

$$c_{ij} = a_{ij} \cdot b_{ij} \tag{8}$$

Let our initial state vector be a column vector $r^{(0)}$ in which elements are all $1/n$. Define Markov transition matrix be M:

$$\mathrm{M} = \beta \mathrm{C} + \frac{1-\beta}{n} e \cdot e^T \tag{9}$$

Then our goal is to find $r^{(t+1)}$, the state vector after t iterations, which satisfies:

$$r^{(t+1)} = \mathrm{M} \cdot r^{(t)} \tag{10}$$

Now it is the same as finding an non-zero eigenvector for matrix M. Since it can be computed efficiently by power iteration algorithm, we skip those implementation details.

Because our Markov transition matrix is stochastic, aperiodic and irreducible, so vector r will always converge to a unique positive stationary vector, which is exactly equal to the vector v we are urging to compute.

Evaluation Problem

In this problem, for the same data model, a_{ij} is an arbitrary real number in a given scale based on different scenarios. Now the element a_{ij} does not just show a simple endorsement from w_i to w_j, instead it means the weight of that endorsement. In another word, given an evaluation scale, a_{ij} is the score w_i graded w_j on a specific skill. We customize the original PageRank algorithm and define the new estimation function as:

$$v_i = \beta \sum_{j=1}^{n} \frac{v_j \times a_{ji}}{d_j} + \frac{1-\beta}{n} \tag{11}$$

Now d_j denotes the summation of evaluation scores made by w_j. Again we made a similar assumption for this definition: every worker evaluates at least one worker with a nonzero score. Compared to the endorsement problem, v_j is not distributed evenly to his co-workers, instead v_j is given to other workers differently according to their skill levels. Because the data processing and implementation are similar with the first problem, we will not go through them again. The convergence property remains for the same argument.

EXPERIMENTAL EVALUATION

We conduct experiments on real-world datasets for the evaluation problem. The main purposes are testing the correctness of our algorithm and also trying to find out how the damping factor β dominates the accuracy of our method.

Implementation

We first collect mutual assessment data of 15 different organizations and companies for various skills on more than 2000 workers from BizMerlin database. Each dataset contains records of assessment score from one worker to other workers in the same organization for one single skill.

Before we take those datasets as input, we process them in the following way. On one hand, since different organizations have their own evaluation scales, we first normalize all scores to the range $[0, 10]$. On the other hand, because there might be multiple scores from one worker to another worker in one dataset from different time periods, we pick the latest one.

Then we implement our algorithm on them with the damping factor $\beta = 0.85$ as original PageRank. Table 1 describes our 15 input datasets from a statistics perspective and their convergence properties.

TABLE 1
STATISTICS ON DATASETS

OID	n	m	RECORD DENSITY	NUMBER OF ITERATIONS
1	50	86	1.72	6
2	93	137	1.47	5
3	180	813	4.51	5
4	196	641	3.27	9
5	170	456	2.68	7
6	196	886	4.52	10
7	121	241	1.99	9
8	59	132	2.24	8
9	184	604	3.28	7
10	84	241	2.87	9
11	169	296	1.75	5
12	194	1809	9.32	3
13	89	208	2.34	5
14	70	191	2.73	10
15	152	665	4.38	7

We use OID to denote different organizations, use ***n*** to demote the number of workers and ***m*** to denote the number of assessment records. Because we use power iteration to compute results, we also record the total number of iterations each dataset took before the state vector ***r*** gets stationary. Furthermore, we compute the ratio ***m*** to ***n*** and call it the record density for later convenience.

In these experiments, we observed that the proposed algorithm never takes more than 10 iterations to finish on the datasets despite the size of input.

Correctness

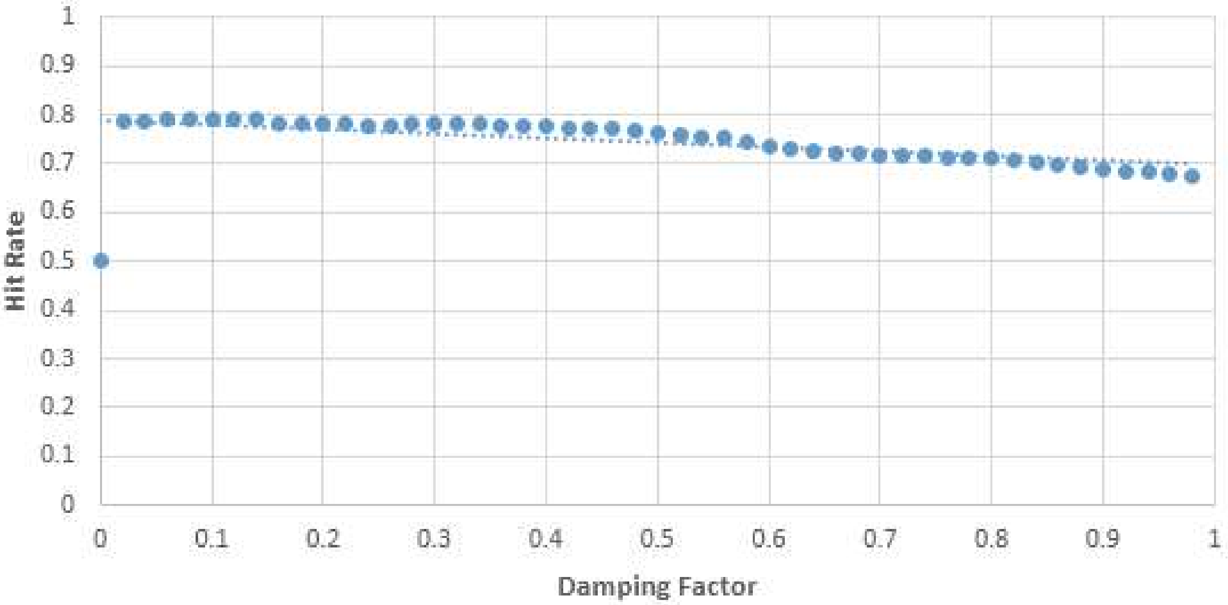
Now we adopt leave-one-out cross validation to evaluate the accuracy of our results. For the dataset of n workers, if we take worker w_i as the test element who made assessments to more than one workers, we pick the worker received the highest score from w_i as the most skilled worker w_H and the worker received the lowest score from w_i as the worst skilled worker w_L. We assume the ground truth for this test element is that the estimation score for w_H should be higher than the estimation score for w_L. If the ground truth is hold in estimations from the rest $n-1$ workers, we say it is a hit, otherwise, we call it an error. If the total number of hits is h, we define the hit rate p as:

$$p = \frac{h}{n} \tag{12}$$

The higher p we get, higher is the accuracy of our results.

We take the average hit rate of 15 datasets to show the correctness of our algorithm. At the same time, we change the value of damping factor from 0 to 0.98 with 0.02 as the interval. We represent results in a scatter diagram (see Figure 1).

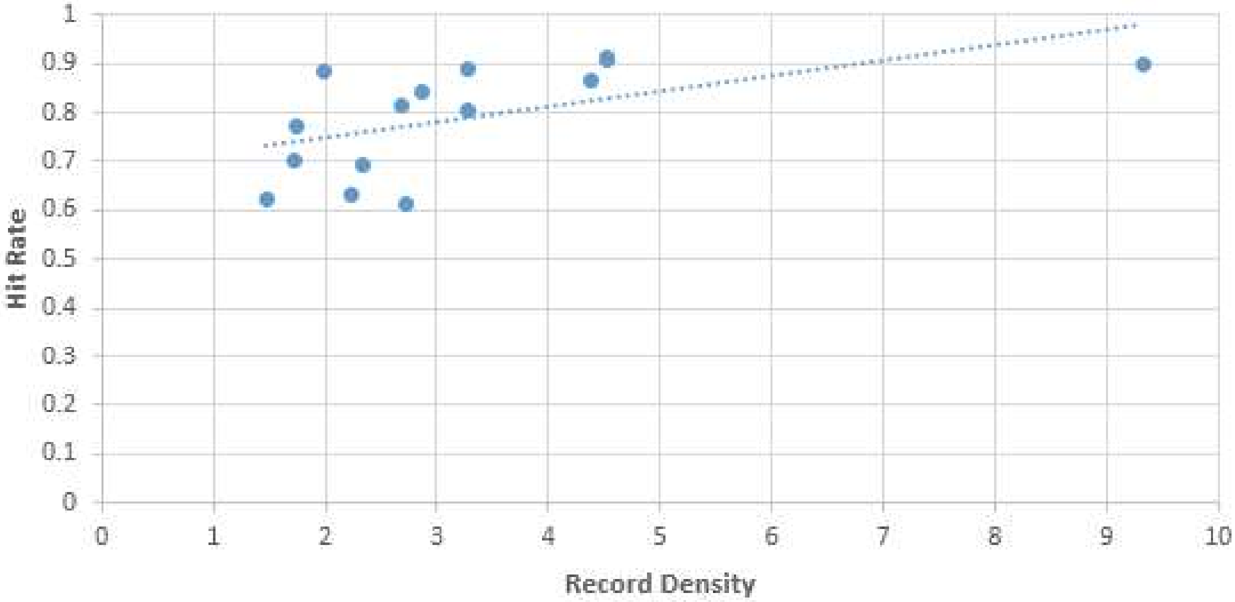
FIGURE 1
RELATIONSHIP BETWEEN OVERALL HIT RATE AND DAMPING FACTOR.



From Figure 1, we make two observations: 1. Our algorithm is valid when the damping factor is in the interval $[0.02, 0.98]$. When $\beta = 0$, our results are independent from mutual assessments and the expectation value of hit rate is 0.5 which is also reflected by experimental data. For other damping factor values, the overall hit rate keeps relatively high, which means our results are accurate and meaningful. 2. Overall hit rate is decreasing when the damping factor is increasing. The maximum hit rate 0.79 is reached when $\beta = 0.1$ and the minimum 0.68 is reached when $\beta = 0.98$.

Finally, we pick the ideal value 0.1 for factor β and test how the change of record density effects the hit rate. We also depict results for our datasets in a scatter diagram (see Figure 2).

FIGURE 2
RELATIONSHIP BETWEEN OVERALL HIT RATE AND RECORD DENSITY



The linear regression function shows a distinct increase of hit rate when the record density gets higher. An intuitive interpretation is that when more feedback are collected from the crowdsourcing system, our algorithm will give us a more accurate result.

Therefore, we draw the conclusion that our algorithm is efficient and accurate for the skill estimation problem and our mechanism expects an input dataset with a high record density.

CONCLUSIONS AND DISCUSSION

In this paper, we propose a new mechanism to solve skill estimation problem. Our algorithm compute estimation scores on workers' skills based on mutual assessments in a crowdsourced system. We treat an organization as a directed graph and each assessment as an weighted edge from one worker to another. Borrowing the idea of PageRank, we specifically consider the weight of each endorsement and develop a customized version algorithm fitted in our scenario. Then we implement the proposed algorithm on 15 real-world datasets collected from organizations of different skill domains and sizes. Using leave-one-out cross validation, we find that experimental results are highly correlated with the ground truth, so we conclude our proposed algorithm is efficient and meaningful.

Future works on skill estimation can cover many of the following issues.

1. There are harsh raters and mild raters in our system. Building an evaluation base line database may allow an estimation system to take account of differences between raters more effectively.
2. Some users are more likely to participate in mutual assessments than others. It might lower the importance of scores from those active raters.
3. In practice, a few workers may have very large variance in their performance when they work on different tasks, which means their skill level can be better represented by a probability distribution function instead of a deterministic value.
4. The algorithms proposed in this work give a result of one snapshot in time. In some real world applications, the skill level of users varies fast, and the evaluation from one user to another also changes quickly. Therefore, an algorithm that can effectively use the different endorsements (for the same combination of workers and skill at different time), may be of significant practical value.

5. When a worker leaves from one organization and joins another, their skill estimation data in previous organization is lost and is not correlated with the corresponding worker identity in the next organization. This may be more of an implementation consideration and not a research topic, but may have significant practical value nevertheless.
6. Customized PageRank algorithm proposed in this work is one type of weighted PageRank but is different from other published versions (Xing, & Ghorbani, 2004, pp. 305-314). It may be interesting to consider various versions and see if one works better than others.

REFERENCES

Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., & Leonardi, S. (2012, April). Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web* (pp. 839-848). Association for Computing Machinery, New Jersey, USA.

Ding, Y. (2011). Applying weighted PageRank to author citation networks. *Journal of the American Society for Information Science and Technology*, *62*(2), 236-245.

Ding, Y., Yan, E., Frazho, A., & Caverlee, J. (2009). PageRank for ranking authors in co-citation networks. *Journal of the American Society for Information Science and Technology*, *60*(11), 2229-2243.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge University Press, Cambridge, England.

Mavridis, P., Gross-Amblard, D., & Miklós, Z. (2016, April). Using Hierarchical Skills for Optimized Task Assignment in Knowledge-Intensive Crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web* (pp. 843-853). International World Wide Web Conferences Steering Committee, Geneva, Switzerland.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web. *Technical Report of Stanford InfoLab.* Retrieved from: http://ilpubs.stanford.edu:8090/422/

Rahman, H., Thirumuruganathan, S., Roy, S. B., Amer-Yahia, S., & Das, G. (2015). Worker skill estimation in team-based tasks. *Proceedings of the VLDB Endowment*, *8*(11), 1142-1153.

Reagans, R., & Zuckerman, E. W. (2001). Networks, diversity, and productivity: The social capital of corporate R&D teams. *Organization science*, *12*(4), 502-517.

Roy, S. B., Lykourentzou, I., Thirumuruganathan, S., Amer-Yahia, S., & Das, G. (2015). Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal*, *24*(4), 467-491.

Skill Endorsements – Overview. (2016, April). Retrieved from https://www.linkedin.com/help/linkedin/answer/31888

Xing, W., & Ghorbani, A. (2004, May). Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on* (pp. 305-314). Institute of Electrical and Electronics Engineers, New Jersey, USA.

Zhang, J., Ackerman, M. S., & Adamic, L. (2007, May). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web* (pp. 221-230). Association for Computing Machinery, New Jersey, USA.