The Expected Optimal Labeling Order Problem for Crowdsourced Joins and Entity Resolution

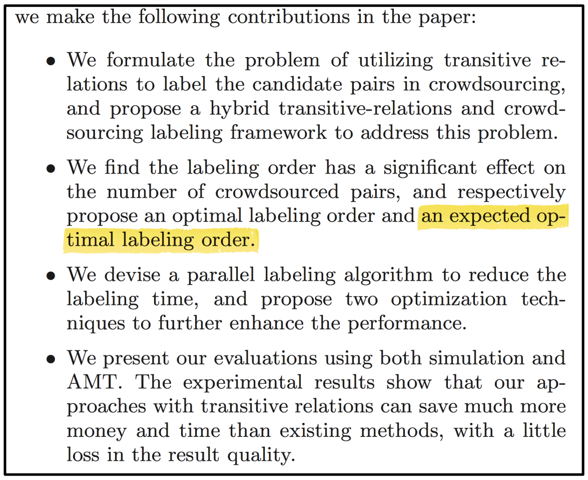
Jiannan Wang[#], Guoliang Li[#], Tim Kraska[†], Michael J. Franklin[‡], Jianhua Feng[#]
[#]Department of Computer Science, Tsinghua University, [†]Brown University, [‡]AMPLab, UC Berkeley
wjn08@mails.tsinghua.edu.cn, ligl@tsinghua.edu.cn, tim_kraska@brown.edu
franklin@cs.berkeley.edu, fengjh@tsinghua.edu.cn

In the SIGMOD 2013 conference, we published a paper [2] extending our earlier work on crowdsourced entity resolution to improve crowdsourced join processing by exploiting transitive relationships. The VLDB 2014 conference has a paper [1] that follows up on our previous work, which points out and corrects a mistake we made in our SIGMOD paper. Specifically, in Section 4.2 of our SIGMOD paper, we defined the "Expected Optimal Labeling Order" (EOLO) problem, and proposed an algorithm for solving it. We incorrectly claimed that our algorithm is optimal. In their paper, Vesdapunt et al. show that the problem is actually NP-Hard, and based on that observation, propose a new algorithm to solve it.

In this note, we would like to put the Vesdapunt et al. results in context, something we believe that their paper does not adequately do.

1. CONTRIBUTIONS OF THE SIGMOD 2013 PAPER

The main contributions of our SIGMOD 2013 paper were to identify the importance of exploiting transitivity to reduce the cost and improve the performance of crowdsourced join processing and to present a new framework for implementing this technique. The issue addressed by Vesdapunt et al. concerns only one aspect of our claimed contributions (shown in Figure 1) namely, one sub-point (highlighted) of the second (of four) contributions listed in the SIGMOD paper. All of the other contributions are unaffected.



we make the following contributions in the paper:

- We formulate the problem of utilizing transitive relations to label the candidate pairs in crowdsourcing, and propose a hybrid transitive-relations and crowdsourcing labeling framework to address this problem.
- We find the labeling order has a significant effect on the number of crowdsourced pairs, and respectively propose an optimal labeling order and an expected optimal labeling order.
- We devise a parallel labeling algorithm to reduce the labeling time, and propose two optimization techniques to further enhance the performance.
- We present our evaluations using both simulation and AMT. The experimental results show that our approaches with transitive relations can save much more money and time than existing methods, with a little loss in the result quality.

Figure 1: Contributions of the SIGMOD 2013 paper.

In particular, the NP-hardness of the EOLO problem does not affect either the correctness of the framework or the correctness of our experimental results. It only means that the efficiency of one of the components in the framework we proposed can be improved, which the authors of the VLDB paper went ahead and did.

2. THE BUG IN THE EOLO OPTIMALITY PROOF

In our proof of optimality for our solution to the EOLO problem, we included several situations that could not occur in practice. Below, we use an example from Vesdapunt et al., to illustrate the problem.

Suppose we have three pairs: (a, b), (a, c), and (b, c). We want to label for each pair whether it refers to the same entity or not. There are two ways to label these pairs. One is to ask the crowd a question such as "whether a pair (e.g., a and b) refers to the same entity". The other way is to use transitive relations to deduce the pair's label. For example, if we know "$a = b$" and "$b = c$", then we can deduce "a = c" without asking the crowd to label it. Similarly, if we know "$a = b$" and "$b \neq c$", then we can deduce "$a \neq c$" using transitive relations as well. For a given pair, if its label is obtained from the crowd, we call it as "crowdsourced pair"; if its label is deduced based on transitive relations, we call it as "deduced pair".

Given three pairs: (a, b), (a, c), and (b, c). Suppose each of the pairs has a probability of 0.5 to refer to the same entity. Consider a labeling strategy, which labels the pairs one by one in the order of (a, b) -> (a, c) -> (b, c), and asks the crowd to label a pair iff. its label cannot be deduced from transitive relations. The question is: "What is the expected number of crowdsourced pairs for the labeling strategy?"

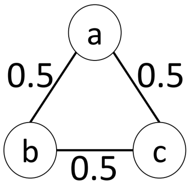


Figure 2: The probabilities of (a, b), (b, c) and (a, c).

To solve this problem, we first compute the probability of each pair being a crowdsourced pair, and then sum up their probabilities.

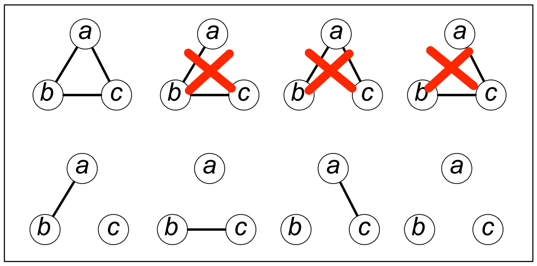


Figure 3: Five possible cases of the labels of (a, b), (a, c), and (b, c).

- For the first pair "(a, b)", as there is no labeled pair before it, we have to ask the crowd to label it, thus the probability of (a, b) being a crowdsourced pair is equal to 1.
- For the second "(a, c)", since its label cannot be deduced from the first pair (a, b), we have to ask the crowd to label it, thus the probability of (a, c) being a crowdsourced pair is equal to 1 as well.
- For the third pair "(b, c)", it needs to be crowdsourced only when both of "$a \neq b$" and "$b \neq c$" hold. Since the probability of "a = b" is 0.5 and the probability of "b = c" is 0.5, the probability of the event that both of "$a \neq b$" and "$b \neq c$" hold is (1-0.5)*(1-0.5) = 0.25. Thus, we calculate that the probability of (b, c) being a crowdsourced pair is 0.25.

By summing up the three probabilities, the expected number of crowdsourced pairs was computed as 1 + 1 + 0.25 = 2.25. This calculation turns out to be incorrect. The correct answer, as pointed out by Vesdapunt et al., is 2.4.

Our error was in the computation of the probability for the third pair (b, c). Consider the five possible cases of the labels of (a, b), (a, c), and (b, c) in Figure 3. In the figure, an edge between a pair of nodes means that the two nodes represent the same entity. Note that the 2nd, 3rd, and 4th graphs are impossible since they violate the transitivity assumption. For the five possible cases, we can compute that each case has a probability of 1/5 = 0.2. Since the third pair "(b, c)" needs to be crowdsourced only when both of "$a \neq b$" and "$b \neq c$" hold, i.e., the 6th or the 8th graphs, the probability of (b, c) being a crowdsourced pair should be 0.2+0.2 = 0.4 instead of 0.25 as computed above.

The incorrect calculation above led us to an incorrect proof of optimality for our solution to the EOLO problem and Vesdapunt et al. rightly show that under the transitivity assumptions we made, the problem is in fact NP-Hard. Furthermore, based on this insight they propose a new algorithm for this aspect of our framework.

3. ALGORITHMIC COMPARISONS

In the VLDB paper, the authors first compared their algorithm with our algorithm on three real data sets. Their experimental results showed that on one dataset, their algorithm is preferable; on a second dataset, our algorithm performs better; on the third dataset, a simple random algorithm performs the best (Figure 12 in their paper). Next, the authors constructed a worst case (for our algorithm), and then showed that in this case their algorithm performed much better than ours (Figure 13). Unfortunately, in the introduction to their paper, Vesdapunt et al. make the claim that the performance of their algorithm is **an order of magnitude** better than ours **in practice**. We do not believe that their experimental results support this claim. In fact, that result was obtained only on an artificial scenario specifically created to defeat our algorithm.

4. UPDATING OUR SIGMOD 2013 PAPER

Based on the analysis of Vesdapunt et al., we revised our SIGMOD 2013 to remove the claim of optimality for our EOLO solution and placed the revised paper on arXiv (`http://tiny.cc/revised`). The following are the changes that we made to the paper:

- We corrected the example of computing the expected optimal labeling order in Section 4.2.
- We cited the new VLDB paper in Section 4.2 to clarify that the EOLO problem is NP-hard.
- We removed the claim that "our algorithm can identify the expected optimal labeling order" from the paper.

4.1 Summary

We appreciate that Vesdapunt et al. discovered flaw in an important aspect of our SIGMOD 2013 paper and were able to publish a full VLDB paper to address this issue. We do feel strongly, however, that the introduction section of that paper overstates the relative benefits of their proposed algorithm for the EOLO problem relative to our original algorithm in practice. That being said, it is clear that the research topic of crowdsourced query processing is gaining increasing attention and that there are a wide range of open challenges to be researched. For interested readers, we collected a list of papers published recently in this topic and put them on this link (`http://tiny.cc/crowdpaper`).

5. REFERENCES

[1] N. Vesdapunt, K. Bellare, and N. Dalvi. Crowdsourcing algorithms for entity resolution. PVLDB, 7(12):1071 – 1082, 2014.

[2] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In SIGMOD Conference, pages 229–240, 2013.