Crowdsourcing Using Uncertainity Resolution for Query Processing

Sumanth,
M.Tech Scholar in Department of CSE, JNTUA college of Engineering, Ananthapuramu, India

Abstract: Querying uncertain data has become a prominent application. An uncertain data with the help of crowdsourcing for quickly converting the real order of same results. Crowd sourcing which consisting the posting tasks to the human and improving the confidence of the human data value by their human judgement. Crowdsourcing has become an active area of research in data management communites. The main aim of the minimizing the crowd interaction is necessary to get the exact result set. In existing system some important data management and human tasks cannot be completely initiated by automated process. These human tasks such as Entity resolution and image recognition can be overcome through the use of human ability. The main drawback of crowd sourcing is effective techniques are used to achieve high quality. In proposed system uncertainity resolutions which is problem of identifying the minimal set of queries to be submitted to a crowdsource and it reduce the uncertainity in data management. To improve crowdsourced data management and focus on balancing quality.

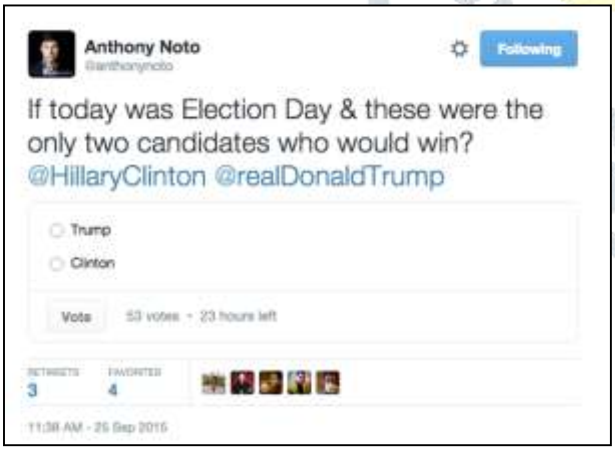
IndexTerms -User/Machine, Systems, Query Processing

I. Introduction

We address the problem of obtaining top-k lists of items out of arbitrarily large item sets using crowdsourcing, a natural and commonly occurring problem. An example application of a crowdsourced top-k query is selecting applications for college admissions. Educational institutions typically receive thousands of applications out of which they select a small set of the higher ranked applicants. Previous studies applicable to the top-k problem in the context of crowdsourcing have major limitations. A natural approach to the top-k problem are tournament style algorithms, as in [Venetis and Garcia-Molina, 2012a] which obtains maxima and [Polychronopoulos et al., 2013] which obtains small top-k lists. The work for the max problem and employs a technique that addresses random spamming but not vandalism, i.e. adversarial spamming by workers who invert the correctness of results. Moreover, the methods cannot tackle top-k lists that are larger than the size of the ranking tasks given to humans. The work in [Marcus et al., 2011] is particularly wasteful in resources. The method studied in [Davidson et al., 2013] invokes the method described in [Feige et al., 1994] to obtain the top- k list for a reduced itemset. The method in [Feige et al., 1994] has a very high latency, since it is essentially a heapsort algorithm for noisy comparisons, where comparisons take place sequentially and not in parallel. Thus, the technique in [Davidson et al. , 2013] is also of high latency. Worker tasks are restricted to pairwise comparisons, while in reality human workers can perform tasks containing significantly more than two items. Moreover, the method's analytic results hold under specific assumptions for the error functions of human workers. In practice, it is difficult to have any knowledge on worker error distribution a priori [Venetis and Garcia-Molina, 2012b]. The work in [Ciceri et al. 2016] assumes prior knowledge on the quality of the items, which is not realistic for many applications. Relevant to our prob- lem is the work in [Ailon, 2012], which presents a way ofsampling a quasilinear number of pair-wise comparison re- sults for the purpose of learning to rank, but performs full sorting with no emphasis on the top-k. A randomized sorting algorithm was studied in [Wauthier et al., 2013], in which predicted permutations are more accurate near the top rather than the middle of the sorted list. Existing proposals in the literature generally lack a robust defense mechanism against the problem of spamming which is rampant in crowdsourcing [Ipeirotis, 2010]. The main contributions of our work are the following: We describe a class of crowdsourced top-k algorithms that have logarithmic latency and require a number of crowdsourcing tasks which grows linearly with the size of the itemset. We propose a budgeting strategy aiming to efficiently address the impact of adversarial users, and a randomized variant of the top-k algorithms that can reduce the required budget drastically by taking a negligible risk of compromising the quality of the output result.We report results from experiments that test the performance of several instantiations of the proposed methods with simulations and real crowds. The results **are** show tolerance of the proposed techniques against errors and vandalism. We draw conclusions on the efficiency of the budgeting strategy, the randomized variant and the trade-off among latency, cost and quality of results. The randomized algorithm is particularly beneficial for very large itemsets and large top-k lists.

Exiting Work Approach

Human Intelligence Tasks were first popularised by Amazon with their Mechanical Turk web service. The online crowdsourcing service was created out of Amazon's need for an efficient way to sort duplicate or similar product entries in their ecommerce database. Amazon created a platform that allowed anyone to register as a worker on their website to get paid (often only a few cents) for the completion of HITs (Human Intelligence Tasks). Although initially developed for Amazon's own purposes, thousands of companies and individuals now submit tasks to Mechanical Turk each day. Crowdsourcing is a specific sourcing model in which individuals or organizations use contributions from characteristics, such as her network centrality, level of activity, expertise, and topical affinity. A viral marketing campaign Internet declarative systems where users to cater necessary services or ideas. Crowdsourcing was coined in 2005 as a portmanteau of crowd and outsourcing. Crowdsourcing is distinguished from outsourcing in that the work can come from an undefined public (instead of being commissioned from a specific, named group) Advantages of using crowdsourcing may include improved costs, speed, quality, flexibility, scalability, or diversity. In the well-known class of applications commonly referred to as "top-K queries", the objective is to find the best K objects matching the user's information need, formulated as a scoring function over the objects' attribute values. If both the data and the scoring function are deterministic, the best K objects can be univocally determined and totally ordered so as to produce a single ranked result set (as long as ties are broken by some deterministic rule). However, in application scenarios involving uncertain data and fuzzy information needs, this does not hold. For example, in a large social network the importance of a given user may be computed as a fuzzy mixture of several may try to identify the "best" K users and exploit their prominence to spread the popularity of a product.



Anthony Noto
@anthonynoto

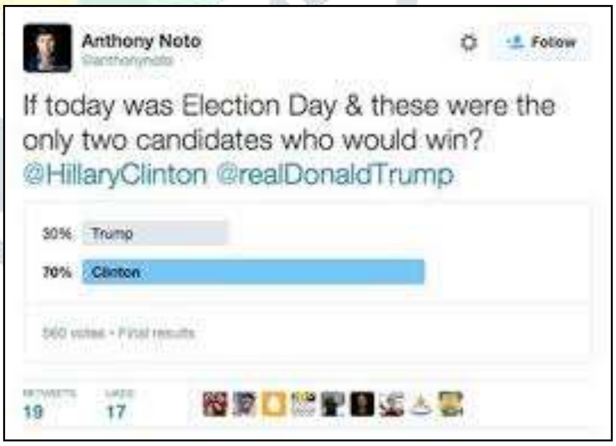
If today was Election Day & these were the only two candidates who would win? @HillaryClinton @realDonaldTrump

○ Trump
○ Clinton

53 votes · 23 hours left

RETWEETS 3
FAVORITES 4

11:38 AM - 25 Sep 2015



Anthony Noto
@anthonynoto

If today was Election Day & these were the only two candidates who would win? @HillaryClinton @realDonaldTrump

30% Trump
70% Clinton

560 votes • Final results

RETWEETS 19
LIKES 17

Current Systems fails to provide the best k objects from the crowd since they don't consider uncertainty of the objects, so we need to develop a system that can do so.

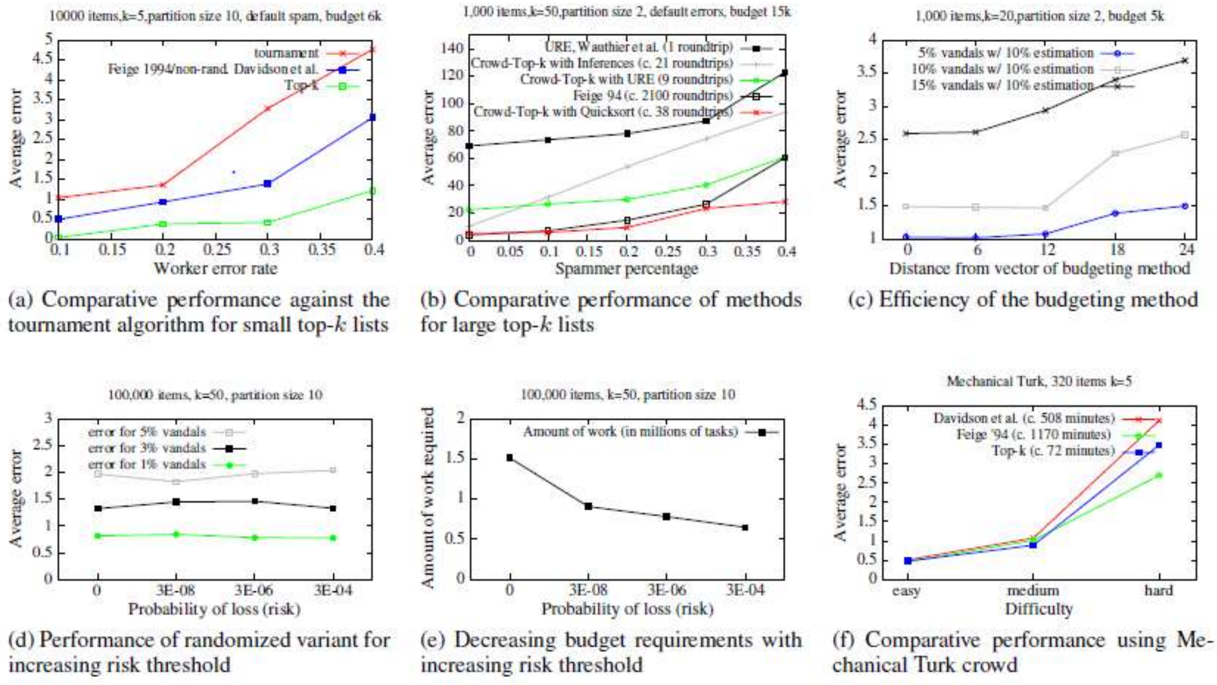
Proposed Work Approach

We propose to define and compare task selection policies for uncertainty reduction via crowdsourcing, with emphasis on the case of top-K queries. Given a data set with uncertain values, our objective is to pose to a crowd the set of questions that, within an allowed budget, minimizes the expected residual uncertainty of the result, possibly leading to a unique ordering of the top K results. We formalize a framework for uncertain top

K query processing, adapt to it existing techniques questions that, when submitted to the crowd, ensures the convergence to a unique, or at least more determinate, sorted result set. We for computing the possible orderings, and introduce a procedure for removing unsuitable orderings, given new knowledge on the relative order of the objects considering several measures of uncertainty, either agnostic (Entropy) or dependent on the structure of the orderings, by proposing algorithms that avoids the materialization of the entire space of possible orderings to achieve even faster results.

- Algorithm 1:Top-1 online algorithm
- Algorithm 2:Incremental algorithm

We formulate the problem of Uncertainty Resolution (UR) in the context of top-K query processing over uncertain data with crowd support. The UR problem amounts to identifying the shortest sequence of questions that, when submitted to the crowd, ensures the convergence to a unique, or at least more determinate, sorted result set. We show that no deterministic algorithm can find the optimal solution for an arbitrary UR problem. We introduce two families of heuristics for question selection: offline, where all questions are selected prior to interacting with the crowd, and online, where crowd answers and question selection can intermix. We conduct an extensive experimental evaluation of several algorithms on both synthetic datasets, and with a real crowd, in order to assess their performance and scalability. The main drawback of above approach is that the crowd source query initiator(CSQI) has no control over query execution area whether it should be in their inner circle or the entire social network. So in case of a more refined query pool there is no means for a specification social network selection. So we would like to hide the query specifications and access policy of a user using a dynamic access policy deriving solution based on CSQI configurations. It's algorithmic implementation is as follows It first calls the policy comparing algorithm PolicyCompare to compare the new access policy with the previous one, and outputs three sets of row indexes which are shuffled to create a perturbed access policy which cannot be reconstructed by the server but yet stored at the server. It adapts based on the owner, receiver, content attributes along with access configurations initiated for the data content by the CSQI. Considering its dynamic efficient nature while upholding privacy and refinement with respect to CSQ it is a much better system compared to prior approaches and it can be extended to more filters such age group specifications, gender specifications, education, designations etc which can be regarded as a future work.



(a) Comparative performance against the tournament algorithm for small top-k lists

(b) Comparative performance of methods for large top-k lists

(c) Efficiency of the budgeting method

(d) Performance of randomized variant for increasing risk threshold

(e) Decreasing budget requirements with increasing risk threshold

(f) Comparative performance using Mechanical Turk crowd

Figure 2: Results of experimental study

We need significantly lower budget to achieve comparable or even superior quality. We can see the use of budget, expressed in required amount of work, in the Figure 2(e). We achieve a budget save that approaches

57% for a very low risk of 3 _ 104, without noticeable change in the quality of results.pairwise comparison HITs. The results in Figure 2(f) demonstrate a comparable performance of the three tested methods in terms of quality of results for all three levels of difficulty. The difference in latencies is significant. The Crowd-Top-k has an approximate latency of 13 roundtrips and 72 minutes, while the algorithm of [Davidson et al., 2013] has a latency of 214 roundtrips and 508 minutes (there is only one task in each roundtrip but the total time is significantly higher than the Crowd-Top-k's latency, because of the parallel execution of tasks in each roundtrip of Crowd-Top-k). The latency of those methods would get prohibitively high for even bigger datasets as it would increase linearly, while the Crowd-Topk algorithm would scale as the latency increases logarithmically.

CONCLUSIONS AND FUTURE WORK

In this paper we have introduced Uncertainty Resolution (UR), which is the problem of identifying the minimal set of questions to be submitted to a crowd in order to reduce the uncertainty in the ordering of top-query results. First of all, we proved that measures of uncertainty that take into account the structure of the tree in addition to ordering probabilities (i.e., UMPO, UHw and UORA) achieve better performance than stateof- the-art measures (i.e., UH). Moreover, since UR does not admit deterministic optimal algorithms, we have introduced two families of heuristics (offline and online, plus a hybrid thereof) capable of reducing the expected residual uncertainty of the result set. The proposed algorithms have been evaluated experimentally on both synthetic and real data sets, against baselines that select questions either randomly or focusing on tuples with an ambiguous order. The experiments show that offline and online best-first search algorithms achieve the best performance, but are computationally impractical. Conversely, the T1−on and C−off algorithms offer a good tradeoff between costs and performance. With synthetic datasets, both the T1−on and C−off achieve significant reductions of the number of questions wrt. the Naive algorithm. The proposed algorithms have been shown to work also with non-uniform tuple score distributions and with noisy crowds. Much lower CPU times are possible with the incr algorithm, with slightly lower quality (which makes incr suited for large, highly uncertain datasets). These trends are further validated on the real datasets. Future work will focus on generalizing the UR problem and heuristics to other uncertain data and queries, for example in skill-based expert search, where queries are desired skills and results contain sequences of people sorted based on their topical expertise and skills can be endorsed by community peers.

REFERENCES

[1] M. Allahbakhsh et al. Quality control in crowdsourcing systems: Issues and directions. IEEE InternetComp., 17(2):76– 81, 2013.

[2] A. Amarilli et al. Uncertainty in crowd data sourcing under structural constraints. In DASFAA, pages 351–359, 2014.

[3] A. Anagnostopoulos et al. The importance of being expert: Efficient max- finding in crowdsourcing. In SIGMOD, 2015.

[4] M. Cha et al. Analyzing the video popularity characteristics of large-scale user generated content systems. IEEE/ACM Trans. Netw., 17(5):1357–1370, 2009.

[5] R. Cheng et al. Efficient join processing over uncertain data. In 15th ACM international conference on Information and knowledge management, pages 738–747. ACM, 2006.

[6] N. N. Dalvi et al. Aggregating crowdsourced binary ratings. In WWW, pages 285–294, 2013.

[7] A. Das Sarma et al. Crowd-powered find algorithms. In ICDE, pages 964–975. IEEE, 2014.

[8] S. B. Davidson et al. Top-k and clustering with noisy comparisons. ACM Trans. Database Syst., 39(4):35:1–35:39, 2014.

[9] J. Fan et al. A hybrid machine-crowdsourcing system for matching web tables. ICDE, 2014.

[10] C. Gokhale et al. Corleone: hands-off crowdsourcing for entity matching. In SIGMOD, pages 601–612, 2014.