**Crowdsourcing for Top-K Query Processing over Uncertain Data**

**ABSTRACT:**

Querying uncertain data has become a prominent application due to the proliferation of user-generated content from social media and of data streams from sensors. When data ambiguity cannot be reduced algorithmically, crowdsourcing proves a viable approach, which consists of posting tasks to humans and harnessing their judgment for improving the confidence about data values or relationships. This paper tackles the problem of processing top-K queries over uncertain data with the help of crowdsourcing for quickly converging to the realordering of relevant results. Several offline and online approaches for addressing questions to a crowd are defined and contrasted on both synthetic and real data sets, with the aim of minimizing the crowd interactions necessary to find the realordering of the result set.

**EXISTING SYSTEM:**

* Query processing over uncertain data has become an active research field, where solutions are being sought for coping with the two main uncertainty factors inherent in this class of applications: the approximate nature of users’ information needs and the uncertainty residing in the queried data.
* In existing system, the quality score for an uncertain top-K query on a probabilistic (i.e., uncertain) database is computed. Moreover, the authors address the problem of cleaning uncertainty to improve the quality of the query answer, by collecting multiple times data from the real world (under budget constraints), so as to confirm or refute what is stated in the database.

**DISADVANTAGES OF EXISTING SYSTEM:**

* The output of humans is uncertain, too, and thus additional knowledge must be properly integrated, notably by aggregating the responses of multiple contributors.
* These amounts to asking many questions that are irrelevant for the top-K prefix, since they could involve tuples that are ranked in lower positions.
* The wasted effort grows exponentially as the dataset cardinality grows.

**PROPOSED SYSTEM:**

* The goal of this paper is 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.
* The main contributions of the paper are as follows:
* We formalize a framework for uncertain top-K query processing, adapt to it existing techniques for computing the possible orderings, and introduce a procedure for removing unsuitable orderings, given new knowledge on the relative order of the objects.
* We define and contrast several measures of uncertainty, either agnostic (Entropy) or dependent on the structure of the orderings.
* 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 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.
* For the offline case we define a relaxed, probabilistic version of optimality, and exhibit an algorithm that attains it as well as sub-optimal but faster algorithms. We also generalize the algorithms to the case of answers collected from noisy workers.

**ADVANTAGES OF PROPOSED SYSTEM:**

* We show that no deterministic algorithm can find the optimal solution for an arbitrary UR problem.
* We propose an algorithm that avoids the materialization of the entire space of possible orderings to achieve even faster results.
* We conduct an extensive experimental evaluation of several algorithms on both synthetic and real datasets, and with a real crowd, in order to assess their performance and scalability.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System                           :         Pentium Dual Core.
* Hard Disk                      :         120 GB.
* Monitor                         :         15’’ LED
* Input Devices                 :         Keyboard, Mouse
* Ram                               :         1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system           :   Windows 7.
* Coding Language           :         JAVA/J2EE
* Tool                               :         Eclipse
* Database                        :         MYSQL