Human-powered Sorts and Joins

Summary

This paper describes how crowds are integrated into a declarative workflow engine called Qurk to reduce the burden on workflow designers. The authors focus on how to use humans to compare items for two of the most common operations in DBMSs: sorting and joining data. Basic query interface and the user interface of the tasks posted to Amazon’s Mechanical Turk were described. They also propose a number of optimizations, including task batching, replacing pairwise comparisons with numerical ratings, and pre-filtering tables before joining them, to reduce the overall cost of running sorts and joins on the crowd. Experiments show that their configuration achieves good results in tasks of joining two sets of images, sorting images according to several criteria, and complex end-to-end query.

Pros

1) Although there are some other groups who have also proposed crowd-oriented database systems motivated by the advantages of a declarative approach, the authors are the first to systematically study the implementation of two of the most important database operators, joins and sorts, in a crowdsourced database.

2) Although intuitive, their finding that ranking costs dramatically less than ordering and produces comparable results in many cases will give guidance in many practical scenarios.

Cons

1) Although different types of experiments are presented in the paper, the dataset the authors used may have some problem. Only one dataset (and this celebrity join dataset was created by the author for experiment) were used to test join implementations and feature filtering, and this may lower the credibility of the conclusion of the paper.

Suggestions

They authors should provide results of their experiments on one or more datasets so as to make their conclusions more convincible.
Max Algorithms in Crowdsourcing Environment

Summary

In this paper, the authors first develop parameterized families of max algorithms, which retrieve the maximum item from a set in crowdsourcing environments. Then, they propose strategies to select appropriate max algorithm parameters. They evaluate their framework using experiments under quantities of quality, monetary cost, and execution time. They also provide insights on the effectiveness of the strategies in selecting appropriate max algorithm parameters and guidelines for choosing max algorithms and strategies for each application.

Pros

1) This paper presents several families of parameterized max algorithms. This middle-of-the-ground approach avoids being too simple or being too specialized to a specific crowdsourcing scenario. It also studies strategies for tuning a max algorithm.

2) The error and cost models the authors consider are more general than previously considered error models.

3) While previous techniques only perform binary comparisons, the authors allow the flexibility of comparing “any” number of items per operation.

Cons

Although the authors used various evaluation metrics in their experiments, they did not use real-world crowdsourcing dataset. This lowers the credibility of their conclusions.

Suggestions

It would be better if the authors could provide results of their algorithms on real-world crowdsourcing datasets in the future.