Title: Human-powered Sorts and Joins:

Summary:

In this paper, authors are considering the highly amount of reimplementations done for common tasks in crowdsourcing marketplaces, and they try to reduce the burden on developers using an engine called Qurk, which is run on top of the crowd sourcing platforms such as Mechanical Turk. They provide an optimized model for doing common tasks, with focus on data sorting and data joining. Using this model they are reducing the costs massively.

Comments:

Pros

The paper is well written and is easy to be understood. The approach is novel and it is really easy to learn the approach they are suggesting. They have used great ways to demonstrate their results and comparison. Their tables and charts are self-explanatory.

The flow of the paper is well and they explanations are complete. They present great discussions and provide good suggestions about the future works, which shows that they have looked deeply into the problem with a critical view.

Cons

In my idea there are too many detailed codes and query examples and giving such detailed examples and demonstrations were not necessary. However, despite giving many examples, they did not provide us with any examples about HIT generation.

They do not give reasons about why they have a fixed value for all the HITs and why they have chosen this amount. I also suggest use of a larger dataset for celebrities’ dataset, with more pictures for the same person, and I think that 25 images for the animal data set is little.

I suggest changing the order of sections in a way that “Related Works” appear sooner than it is now. Furthermore, I believe that the number of references is little for this paper.
Questions:

Why have they chosen this fixed amount for HITs? Why is this amount the same for all the HITs?

How possible is it to extend these models to other tasks?

(Please find the critiques for the next paper on the following pages)
Title: Executing SQL over Encrypted Data in the Database-Service-Provider Model:

Summary:

This paper focuses on the data privacy challenge of online software products, and focuses on protection of data from the service providers. They use SQL queries that can be executed on data that is encrypted. So they suggest a strategy to split data and process queries on encrypted data at the provider's site. Then they send the results to the client side, and use client site to decrypt and execute the rest of data to do the rest of processing. They present a framework to use minimum computation at the client side. This study can help more organizations to use service providers instead of spending time on preparing their own software and hardware.

Comments:

Pros

They study a really important issue from an aspect that although is also very important, is less studied compared to other aspects of data protection and privacy.

The paper is well written and it is understandable. The way they demonstrate the query tree helps to make their approach more comprehensible. The results are shown clearly and the charts are perfectly representing the results.

The relational operators are implemented and explained well over the encrypted data. Mapping functions, storage of encrypted data, and decryption functions are explained clearly.

Cons

I believe that they have not explained their experiments enough and they could have dedicated a longer part to their experiments and analysis. Moreover, there is a lack of discussion part at the end. Although they have talked about some ideas of future work inside the paper, it would have been better if they mentioned future work and discussed it at the end as well.

The conclusion part does not actually present a conclusion of experiments and approach; it is more giving a summary of what is said in the beginning of the paper, in abstract and introduction parts. There are many parts in the paper that worth to be included in the conclusions part, such as “These experiments show that our architecture does not introduce significant overhead due to the proposed communication protocol between client and server” and I think the conclusion part should be more dedicated to these parts than what it is currently dedicated to.

They are showing the results on the last page in figure 10 and figure 11, it would have been better if they moved these figures to page 11.
Questions:

Does execution at both client side and server side cause the process to be less fault-tolerant?

I think it might be hard for server providers to manage different types of servers at different client sides as they might vary a lot. Would it actually be an efficient way for the service providers?