Summary

“Max Algorithms in Crowdsourcing Environments” describes and traverses the subject matter of designing hybrid human/computer decision algorithms designed to identify a specific item from a group of items based on a relative criteria spectrum. In this particular article, the various factors that impact how well such an algorithm performs are identified. The factors contributing to these outcomes which are investigated through experimentation, and all relate to themes such as task type, workload (human), error types and rates, and precision required from results. Through experimentation manipulating these factors the authors quantify impact and identify tradeoffs related to the quality of the algorithm results (correctness), the cost of the algorithm in terms of computing or decision processes, and finally the total execution time (latency) required to complete the decision algorithm.

Pros

This article covers a number of pragmatic issues critical to developing and implementing Max algorithms in crowd sourced workload execution environment. Though the conclusion seems to be rather obvious, the breadth and depth of experimentation that precedes the conclusion lends great creditability. All in all, the work represents a very pragmatic and realistic bridge between computational science of this subject matter and knowledge needed for implementation that ultimately develops knowledge into viable commercial value.

Cons

Though the experimentation phase of the research was very broad and well designed, it does seem as if the VaryingS algorithm stood out early. Given this early outcome, some discussion as to why this is the case or if it was expected would have been appropriate.

Suggested Improvements

Some suggestions for improvement or follow on research might include discussion of why VaryingS stood out as superior in all regards. In the future, such “whys” might inform better decision making especially if circumstances are uncovered where this generalization no longer holds true. From a formatting and prose standpoint, some of the diagrams would benefit from some formatting work as a few of them were questionably readable or useful to the audience.
Summary
“Human-powered Sorts and Joins” investigates the process of optimizing simple sort and join Human Intelligence tasks (HITS) used to populate databases using crowd sourced applications. Task batching and feature filtering strategies are proposed to increase efficiency of human work execution. Through experimentation various forms of batching and filtering are tested to quantify the success of each algorithm. These experiments ultimately measure how well these strategies integrate with the human component of the overall algorithm, or more simply, how good they are at creating more value from individual HITS. In many ways, this manifests as compressing more work into the same task, and thus a certain workload tolerance on the part of the human component is also investigated.

Pros
Though this paper investigates some very interesting dynamics in term of computational informatics, it also robustly addresses the human component of the algorithms and overall data development system. Through the discussion sections, it identifies interface design and intensity of workload compression present in certain batching strategies as important to the human element which executes the task. Nonetheless, the experiments are able to identify batching and sorting as strategies to improve efficiency in crowd sourced work environments, without a substantial loss of accuracy or work quality. All in all, the work represents a very pragmatic and realistic bridge between computational science of this subject matter and knowledge needed for implementation that ultimately develops knowledge into viable commercial value.

Cons
Given the authors discussion on issues and lessons learned task work on Mechanical Turk, it would be interesting to see some more elaboration on how the human factors might adversely impact accuracy of work or success of Join and Sort strategies.

Suggested Improvements
There were some very interesting points made through the discussion section of the paper that would be nice to see touched on in the conclusion section. The value of the work in terms of computational informatics is interesting, but there is a deeply integrated human component discussed heavily in the discussion section but ignored in the conclusion.