CrowdFlow: Integrating Machine Learning with Mechanical Turk for Speed-Cost-Quality Flexibility

Alexander J. Quinn[1], Benjamin B. Bederson[1,2,3], Tom Yeh[3], Jimmy Lin[1,2]
University of Maryland
Human-Computer Interaction Lab
Department of Computer Science[1] || iSchool[2] || Institute for Advanced Computer Studies[3]
College Park, MD 20742 USA
Contact: aq@cs.umd.edu

Many problems involving visual perception, language understanding, or other human abilities can now be solved by computers. This allows the tasks to be done faster and more affordably than humans could do. Furthermore, it means they can be done on-demand, enabling computers to monitor national intelligence information streams, search video footage for missing persons, and perform a wide variety of other services of importance to society. The primary disadvantage is that machines are still not as accurate as humans on many tasks.

The field of human computation - alternately referred to as crowdsourcing [1, 2], distributed human computation [7], or collective intelligence [5]- is concerned with methods and systems for distributing small, independent tasks to anonymous workers connected by large online networks. This enables faster turnaround time and reduces the overhead of hiring humans to do the work in an office. Furthermore, it is typically much less expensive than hiring employees. Human computation employs a variety of paradigms, including games with a purpose (GWAPs) [9, 10] and online task markets, such as Amazon Mechanical Turk (AMT) [4].

Although such systems provide more flexibility than in-person workers, they are still more expensive than running the same jobs with a computer. Furthermore, they human effort in cases where the computer's accuracy is modestly good. Perhaps most importantly, they lack the ability to tune the speed, cost, and quality to the situation's needs.

CrowdFlow is our general toolkit (currently implemented as a Python library) built to solve this problem. The framework blends the flexibility of AMT with the speed and affordability of machine learning systems.

An especially powerful element of this framework is that the humans and machines benefit from one another in a kind of *man-computer symbiosis* [3]. Machines benefit from the human-created training data, which helps boost their accuracy. Human workers benefit from having the machine results as a starting point. When the machine results are correct, the worker need only verify that fact.

The user of CrowdFlow specifies a desired speed-cost-quality tradeoff. The system then allocates tasks to humans and machines in a way that attempts to fulfill the user's specification. By estimating system performance, we can describe a tradeoff space gamut within which it is possible for the human manager to manipulate the system.

Speed may be expressed as a time limit for completing the job. Similarly, cost is the maximum the user is willing to pay to Turkers and/or for any cloud-based computational resources. (The latter is planned but not currently implemented in our toolkit.) Quality is measured relative to some satisfaction criteria the user provides. It could be a heuristic that combines multiple criteria.

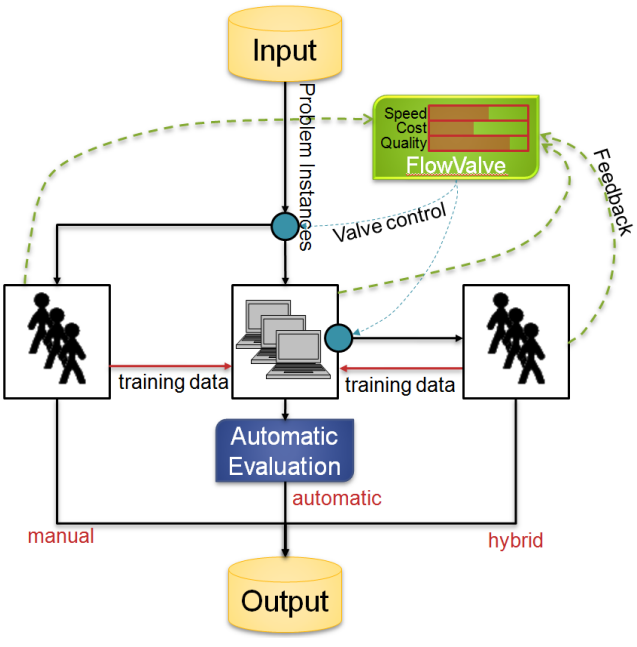

Figure 2. CrowdFlow conceptual model.

Making CrowdFlow work will entail several challenges. First, we must be able to estimate, within some confidence interval, the accuracy of the human workers, even if there is no ground truth. This is doubly important when you consider that without the humans' judgment, it is impossible to estimate the machine's accuracy. Also, the system must keep updating the accuracy estimates and automatically adjust the allocation of tasks as needed. Second, we need to specify a generic architecture that is both easily usable by developers of specific problems while still providing significant value. Third, we need to support a wide range of human capabilities in this context.

We aspire to keep CrowdFlow as general as possible. It should be applicable to problems with these properties:

- Solvable by humans but at unacceptable speed or cost.
- Solvable by computers but with unacceptable quality.
- Divisible into small, independent subtasks.

In particular, we expect it to be especially useful for tasks with this additional optional property:

- Cannot be solved by machine learning algorithms previously trained for other problems.

CrowdFlow currently uses Turkers in two roles (although we envision using a much richer model in the future):

- Worker answers the question from scratch.
- Fixer corrects a machine-created answer.

Which role is used depends on the specifics of the domain and the machine learning system. If the cognitive cost of fixing an incorrect result is low, and/or if the machine's accuracy is high, then the fixer role is preferred. This gives Turkers the benefit of the machine results, reducing their effort, and potentially reducing the required price and/or time required to get the work done. Otherwise the worker role is used. Currently, the user decides, but ultimately, we envision that the valve will decide automatically.

ANALYSIS

We wanted to better understand CrowdFlow's ability to flexibly target a specific point in the speed-cost-quality tradeoff space. Running CrowdFlow repeatedly for every possible point in the space would consume time and money needlessly. Instead, we ran the HITs once and used simulation to explore the space. We chose two domains.

Human detection. In this task, the goal is to identify human subjects in a photograph and determine their bounding boxes. Turkers did this using a web interface. The machine ran the DetectorPLS human detection software [8], which implements a partial least squares algorithm. We found this to be 60% accurate on the data we used. The web interface was pre-populated with the machine's results so that if the machine was correct, the Turker could simply press a button to confirm that.

Turkers did 120 tasks in a total of 3 hours 42 minutes (1 minute 51 seconds per task) for $2.40 ($0.02 per task) with 90% accuracy, using standard cheating deterrence methods.

We can extrapolate to other scenarios. For example, had we needed to do 1000 such tasks under a time constraint of 10 hours, the Turkers would have done only $3.24 tasks, with the remainder done by the machine. That would yield a combined accuracy of 70% and a cost of $6.48.

Sentiment polarity of movie reviews. The goal of this task is to determine if a movie review (typically several paragraphs) is more positive or more negative. Turkers read the review and answered by selecting a radio button. The machine used a classifier based on the algorithm, described by Pang and Lee with the implementation provided by the publicly available LingPipe NLP toolkit. Accuracy on our data was 83.5%. Since the answer format was simple, the worker role was used.

Turkers did a total of 1083 movie reviews (in batches of 3) over a period of 8 hours 7 minutes for $18.05 with overall accuracy of 91%. Extrapolating, we can estimate that had the user needed only 90% accuracy (for example), 1083 movie reviews could have been done for $15.64 in 7 hours 1 minute. Although the cost and time savings here is modest, CrowdFlow lets the user control the tradeoff.

FUTURE WORK

The CrowdFlow toolkit demonstrates the ability to blend human and machine resources to achieve a desired tradeoff point. In the future, we hope to achieve even greater flexibility using more human roles (e.g. verifying correct or incorrect, appraising whether an incorrect answer would be worth fixing as opposed to doing it from scratch, etc.). Also, although our current experiments relied on some existing ground truth to measure accuracy, we hope to develop methods in the future that can estimate within some confidence interval even without the ground truth.

REFERENCES

1. Hoffmann, L. *Crowd control*. CACM 52, 3 (2009), 16-17.
2. Howe, Jeff, *The Rise of Crowdsourcing*. Wired Mag. June 2006.
3. Licklider, J. C. R. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, 1960.
4. Mechanical Turk. http://mturk.com.
5. Malone, T.W., Laubacher, R., and Dellarocas, C. Harnessing Crowds: Mapping the Genome of Collective Intelligence. (February 3, 2009). MIT Sloan Research Paper No. 4732-09.
6. Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. *ACL '02*
7. Quinn, A. J., Bederson, B. (2009) A Taxonomy Of Distributed Human Computation. Tech. Rpt, Univ. of Maryland, HCIL-2009-23.
8. Schwartz, W. R., Kembhavi, A., Harwood, D., Davis, L. S.. Human Detection Using Partial Least Squares Analysis. *ICCV* '09
9. von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. *CHI '04*.
10. von Ahn, L. v. 2006. Games with a Purpose. *Computer* (Jun. 2006)