Quality-controlling real-time crowdsourced data

First Draft

Aaron Steele

Abstract

TODO

Introduction

Crowdsourcing has become a much more popular way of both getting and processing data in recent years. In processing data, crowdsourcing is primarily helpful for tasks which are easy for humans to do but hard to machines. Getting data, on the other hand, is a very different process. With the advent of the ubiquity of smartphones, it has become much easier to gather various types of data from the crowd, as it were.

Quality controlling crowdsourced processing is, in a lot of ways, easier than doing the same with crowdsourced data. We have created a taxonomy of quality in crowdsourced systems. There are various approaches to quality controlling the processed data, but we believe the most relevant one for our purposes is Contributor Evaluation, which is where a contribution is assessed based on the quality of the contributor's previous contributions.

Body

Thanks to the rise of smartphones being abundant, a new type of crowdsourcing has been created. Ubiquitous crowdsourcing is smartphone owners contributing data about their outside world, such as GPS location, or ambient noise level. For example, Google Maps, or Waze, both rely on crowdsourced data to give information about traffic conditions, wrecks, and other events that happen on the road.

A major issue that faces developers who wish to use ubiquitous crowdsourcing is quality control. Just like in crowdsourced processing, the data gathered from ubiquitous applications must be controlled for quality. For ubiquitous systems, quality control is even more important than processing systems, in a lot of cases. On top of dealing with outright malicious users submitting bad data, the "fuzziness" of the real world requires us to deal with a truly huge amount of possible edge cases. Getting usable data out of the mess is what this paper will be discussing.

Two other issues that ubiquitous crowdsourcing faces that crowdsourced processing doesn't have to deal with are Real-time Events, and Dynamic Crowds.

- *Real-time Events*: ubiquitous crowdsourcing inherently deals with real-time events. The data is gathered in real time, and in many instances, real-time processing of the data is expected as well. In crowdsourced processing, on the other hand, quality control can often be delayed by some amount of time, if needed, to be checked and flagged by an authorized or more credible user.

- *Dynamic Crowds*: since ubiquitous crowdsourcing deals in real-time, the crowd itself is also often dynamic. People start and stop driving all the time, for example. The challenge of a dynamic crowd is that there are times when the number of contributors might not reach criti-

Conference: UbiComp'11 (`https://dl.acm.org/citation.cfm?id=2030100&picked=prox`)

Papers: [1][2]

cal mass. Having too few points of data makes quality control even more important, on top of increasing the challenge of getting the proper results from the program.

Participatory sensing is the concept of communities (or other groups of people) contributing sensory information to form a body of knowledge. A common problem with participatory sensing is data corruption, or malicious users intentionally sending invalid or fallacious data. The paper [] shows a new concept of controlling for this by allowing consumers of the crowdsourced data to assign trust scores to specific sources of data.

The only other paper that is related to our work is []. In that paper, the authors consider the scenario where users deliberately try to confuse the sensor and send false data. The solution the authors propose is using a trust-based rating system, where each contributor has a reputation, or trust, rating and processed the data sent from users based on this rating. While their system shows an improvement over other trust-based systems, we propose an improvement to that system.

Our proposed method is to track the users' mobility patterns. Studies have shown that data can be very consistent when cross-referencing with a users' commute. For example, there is a high degree of regularity in a weekday commute; most people tend to make a schedule and stick to it when traveling. [] We look at taking a public bus, and users' traveling patterns to determine when the busses will arrive at given stops. First we define, for each user, Points of Interest (POIs) with tuples, $T(loc_{POI_x}, t_i)$. loc_{POI} is the specific point where the contribution was submitted. t_i is a logical time as opposed to a timestamp. We define a logical time as something application-relevant, such as *morning*, *afternoon*, or *late night*. While processing the data, we consider patterns differently for weekdays vs. weekends, since people generally have more regular schedules on the weekdays.

Based on this, and with this framework, we define a regularity function $Reg(T_j)$ whose value is determined based on implicit location readings from the user's device. While there is obviously no history data from the start of the logging process, after just a few days of logging data we can beign to put together a fairly accurate regularity function for the user. We start to define POIs for the user as *local*, *familiar*, and *stranger*.

On top of the regularity function, we also define a reputation score. When the user submits data, it is checked against other users submissions and also their own regularity function to determine a trustworthiness score, $Trust(u_i)$. For crowdsourced processing, this kind of trust function is usually created by reviews on the users' input from experts, but in ubiquitous crowdsourcing the rating must be different, as there is both too much data and not enough time for éxpertsśo rate each user. On top of that, using the formula we created, it is still possible to use data from locations where the user-submitted data is very sparse. Based on the regularity and trust functions, we can compute a credibility weight for user u_i:

$$\text{credibility weight}(T_j) =$$

$$\alpha \cdot Reg(T_j) + (1 - \alpha) \cdot Trust(u_i)$$

α can be adjusted on the fly to give more weight to either the trust or regularity functions for the overall credibility weight. For example, if there is an area with a large amount of submissions, the regularity function of each user might not be as important as the trust score, and visa-versa in low-data areas.

Related Works

WIP: Wikipedia, Reddit, Slashdot, IMDB, Voluntary Geographical Information (VGI). In [14] Authors are investigating the quality of voluntarily submitted POIs.

Conclusion

WIP: The authors are creating an Android-based app to give more accurate bus-tracking data. The app also includes a game-like element, which tends to increase user engagement and participation [source?].

References

[1] A. J. Mashhadi and L. Capra, "Quality control for real-time ubiquitous crowdsourcing," in *Proceedings of the 2nd international workshop on Ubiquitous crowdsouring*, pp. 5–8, ACM, 2011.

[2] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: Issues and directions," *IEEE Internet Computing*, vol. 17, no. 2, pp. 76–81, 2013.