Crowd Sourcing, with a Few Answers: Recommending Commuters for Traffic Updates

Elizabeth M. Daly, Michele Berlingerio
Smart Cities Technology Centre
IBM Research - Ireland
dalyeliz, mberling@ie.ibm.com

François Schnitzler
Technion - Israel Institute of Technology
Haifa, Israel
francois@ee.technion.ac.il

ABSTRACT

Real-time traffic awareness applications are playing an ever increasing role understanding and tackling traffic congestion in cities. First-hand accounts from drivers witnessing an incident is an invaluable source of information for traffic managers. Nowadays, drivers increasingly contact control rooms through social media to report on journey times, accidents or road weather conditions. These new interactions allow traffic controllers to engage users, and in particular to query them for information rather than passively collecting it. Querying participants presents the challenge of which users to probe for updates about a specific situation. In order to maximise the probability of a user responding and the accuracy of the information, we propose a strategy which takes into account the engagement levels of the user, the mobility profile and the reputation of the user. We provide an analysis of a real-world user corpus of Twitter users contributing updates to LiveDrive, a Dublin based traffic radio station.

Keywords: Recommender Systems; Crowd-sourcing; Traffic.

1. INTRODUCTION

Crowd sourcing of urban information is becoming increasingly important in the context of smarter cities. For example, Waze[1] provides crowd sourced traffic monitoring. FixMyStreet supports reporting information such as noise pollution, potholes and general repairs for the attention of city councils[2]. Social media analysis has already been used for incident or emergency detection or monitoring, such as earthquakes [9] or floods [5, 12]. It has been used also for traffic monitoring in the city of Dublin [4]. In order to move from passive crowd sourcing to active crowd sourcing, such systems need to actively identify and engage users to provide the right information at the right time, thus encouraging contributions.

Specific solutions have also been developed to estimate user reliability in the context of spatial crowd sourcing, and in particular traffic monitoring. Crowd sourcing can be used to resolve conflicting reports from physical sensors, and these reports can be used as an initial estimate of the true answer [2]. Spatial models have been developed to detect unreliable crowd sourced sensors to measure radioactivity in Japan [11]. When monitoring social media, a related challenge is to detect individuals that are located in the area affected by a problem and these merely talking about it [10].

Sensor or user selection based on space and time constraints have also been considered. In order to maximise the usefulness of the answers received, one can construct probability distributions over sensor locations and optimise the queries based on this distribution [8]. Generic location aware crowd sourcing platforms have also been developed. These platforms distribute tasks to users based on their location [1, 6, 7]. These platforms are different from our setting. Indeed, they either use a pull-based system, where users request tasks based on their current location, continuously monitor the location of users or expect users to move to complete tasks. We, on the other hand, expect our queries to be generated mostly during peak hours, and for users commuting. In our opinion, such users are unlikely to take a detour in a traffic jam to answer a question. However, they can report on a location they've been through recently. We expect a user to provide information only if they have a history of reporting on this particular location of interest.

Our contribution is an algorithm to recommend, based on predicted user engagement, which users to follow or to query to obtain traffic updates around the city. Our algorithm takes into account:

- the engagement level and reputation of the user;
- the time and mobility profile of the user, based on the time of day and locations a user typically communicate about;
- the number of queries previously directed at the user, in order to distribute the load across users;

We evaluate this proposed strategy using a real-world collection of user contributed tweets and traffic related updates of LiveDrive, a Dublin based traffic radio station.

2. EXPERIMENTAL SETUP

LiveDrive[3] CityFM is a radio station run by Dublin County Council. The radio station broadcasts during the morning and evening peak traffic times. Listeners are incentivised to provide real-time updates to the station by the possibility to receive free parking vouchers, request a song of their choice and the general understanding that if they contribute useful information that benefits others, they too may benefit from the contributions of others. The team consists primarily of three members, the radio presenter who communicates the updates to the listeners, the producer who monitors incoming alerts and a camera controller who has access to a number of the traffic cameras in the city when trying to clarify reported issues. Incoming alerts are sent to the team via email, phone calls, text messages and Twitter.

[1] https://www.waze.com/

[2] https://www.fixmystreet.ie/

[3] http://dublincityfm.ie/livedrive

RecSys'15, September 2015, Vienna, Austria.
© 2015 ACM. ISBN 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2799673.

We collected the tweets both originating from and directed to the LiveDrive Twitter account using the public API[4] from October 2012 until February 2014 resulting in 38637 tweets from 2240 accounts. As most of the tweets are not geo-located, we generate for each tweet an approximate location based on specific words used, mostly street names, abbreviations of these names or specific area in the city. To do so, we follow the methodology of [4].

Example LiveDrive tweet:

```
Source: LiveDrive
Message: There has been an incident on Nassau St
at the junction of Dawson St. Nassau St is blocked
and the left turn from Dawson St is also blocked
Date: Fri Feb 14 07:06:29
Location Keywords:  nassau junction dawson
Approx. Location: 53.341297, -6.2582927
```

Example tweet from user:

```
Source: @BeastyBoy_
Message: Pedestrian knocked down on Nassau st at
house of frasier - no traffic getting past.
@LiveDrive  @aaroadwatch @gardainfo @GardaTraffic
Date: Fri Feb 14 07:04:03
Location Keywords: nassau
Approx. Location: 53.342697,-6.256634
```

For a broader analysis, we also collected another 943703 tweets geo-located in Dublin, not related to the LiveDrive account.

2.1 Tweet validation

To build the reputation of a user, we must decide whether a tweet sent to LiveDrive is related to a valid incident or not. There are two typical crowd sourcing methods to assess the quality of a tweet: compare to other tweets, or to a gold standard. Since the majority of incidents were reported by a single user or a very small number of users, majority voting strategies are of limited use. Therefore, we rely on a gold standard: the warnings about incidents on the LiveDrive Twitter account.

We first assign an approximate location to both the user tweets and the LiveDrive tweets. We then try to match each single user tweet with a LiveDrive one, based on a spatio-temporal distance. For the spatial dimension, we tried increasing distances from 0km to 5km, and we saw a sharp increase of matches up to 1km, while little difference was noted after that. We therefore selected 1km as spatial threshold. For time, we noticed that there is a linear correlation between temporal distance and number of matched tweets. We arbitrarily chose 2 hours as temporal threshold, based on our personal estimates of how long a traffic problem may last.

2.2 User Reputation

In order to take into account the reputation of a user, we wish to reward users who provide actionable information. If a user consistently provides content that is rebroadcast and shared by the LiveDrive team and therefore of benefit to other users, then this user can be considered a reputable contributor. The reputation of a user is initialised to an input parameter $R_{init} \in [0, 1]$. For every additional informative tweet sent by the user, the value is updated with a reward constant R_{reward}

$$R_{new} = R_{old} + (1 - R_{old}) \times R_{reward}. \qquad (1)$$

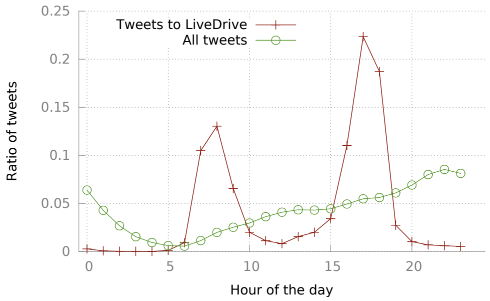

Figure 1: Probability Distribution of User Contributed Tweets over Time

In order to capture the time dynamics and take into account that the last time a user has contributed information also plays a role in the probability as to whether the user will tweet, or respond, about an incident, the reputation value is decayed over time. As a result, users who are continuously providing valuable updates are rewarded, and inactive users are degraded by:

$$R_{new} = R_{old} \times \gamma^k \qquad (2)$$

where γ is the decay coefficient and k is the number of elapsed time units since the user last contributed an update. The selection of the decay coefficient, γ, and time unit, k, should be based on the importance of recency. For the purposes of this paper we use the following values: R_{init}=0.5, R_{reward}=0.1, γ=0,98 and k = number of weeks.

2.3 Time Profile

We seek information updates that can be provided in an opportunistic manner, so we need to take into account the time of day we believe the user may be travelling, based on the previous updates provided by the user. Figure 1 shows the probability distribution of when messages are received by the LiveDrive team (crossed red line). The peak hours for the LiveDrive related tweets are in the morning between 7 and 10 followed by an afternoon peak between the hours of 4 and 7. We also report the probability distribution of all other tweets not related to LiveDrive (circled green line) to give a perspective of the relative effects and volume of the LiveDrive service compared to generic Twitter accounts and topics.

In order to compute the time profile of the users we discretise the time of day the users were observed into hourly intervals $t \in [1, 2, \cdot, 24]$. We then bin the frequencies a user has been observed in the respective hourly intervals $c_u(t)$ and compute the corresponding maximum likelihood estimate as:

$$P_u(t) = \frac{c_u(t)}{\sum_t c_u(t)}. \qquad (3)$$

2.4 Location Profile

We also need to take into account the locations each user may be travelling through based on his previous tweets. Figure 2 shows the locations associated to all tweets we use. We construct individual users' location profiles by looking at the location keywords used for the geo-coding of tweets rather than at the coordinates. We believe that these words provide more information about the trajectory of a user than point-wise coordinates. For example, a user reporting a problem on the N4 on the way to the M50 motorway (big road circling Dublin) may be able to report on a problem on the N4 or the M50 far from the crossroad, but not be able to do so for a problem on a road parallel to these even if the incident is closer to the N4-M50 junction.

We compute the location profile in a similar manner as before. We bin the frequencies of a user's observations across all location keywords assuming location keywords are not necessarily dependent on each other. Given an input location query $\mathbf{l} = \{l_i\}$ that gives potentially multiple location keywords is now used to compute the probability that the user has information about those input location keywords:

$$P_u(\mathbf{l}) = \frac{\sum_{l \in \mathbf{l}} c_u(l)}{\sum_l c_u(l)} \ . \qquad (4)$$

[4] http://www.twitter.com

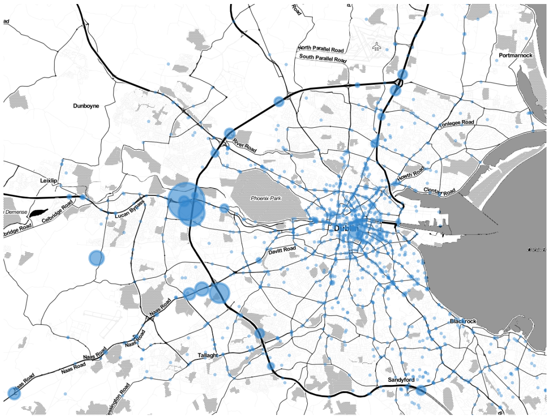

Figure 2: Locations in Dublin that were associated to tweets. The radius of the circles is proportional to the number of tweets at each location.

3. STRATEGY FOR RECOMMENDATIONS

The criteria we consider include the number of tweets contributed by the user, the time profile of the user, the location profile of the user and the reputation. Each of these criteria has a different scale and therefore mapping these objectives to a single utility scale can be problematic, since it may be difficult to provide precise trade-offs between objectives [3]. As a result, it is natural to consider multi-attribute or multi-criteria utility functions to cope with multiple and non-commensurate utility scales on which the decision maker's preferences are expressed. Therefore we propose using Pareto optimisation to determine the non-dominant solution set.

Given p attributes, a *multi-attribute utility value* is characterised by a vector $\vec{u} = (u_1, \ldots, u_p) \in \mathbb{R}^p$, where u_i represents the utility with respect to attribute $i \in \{1, \ldots, p\}$. Given $\vec{u}, \vec{v} \in \mathbb{R}^p$, we say that $\vec{u}$ *Pareto dominates* $\vec{v}$ (denoted $\vec{u} \succcurlyeq \vec{v}$) iff $\forall i \in \{1, \ldots, p\}$ $u_i \geq v_i$. As usual, the symbol $\succ$ refers to the asymmetric part of $\succcurlyeq$, namely $\vec{u} \succ \vec{v}$ iif $\vec{u} \succcurlyeq \vec{v}$ and it is not the case that $\vec{v} \succcurlyeq \vec{u}$. Given a finite set of utility vectors $\mathcal{U} \subseteq \mathbb{R}^p$, we define the *maximal set*, denoted by $\max_{\succcurlyeq}(\mathcal{U})$, to be the set consisting of the undominated elements in $\mathcal{U}$, i.e., $\max_{\succcurlyeq}(\mathcal{U}) = \{\vec{v} \in \mathcal{U} \mid \nexists \vec{v} \in \mathcal{U}, \vec{v} \succ \vec{u}\}$. Computing $\max_{\succcurlyeq}(\mathcal{U})$ is quadratic in the size of $\mathcal{U}$.

4. APPLICABILITY STUDY

Our goal here is to study how the main strategy described above performs in an hypothetical application for recommending users to follow, or to query about a specific incident. We imagine to have to query a number of users under budget constraints to maximize the likelihood of their reply, while ensuring load balance across users. Further validation of this methodology involving actual queries is part of our future work in this direction.

We utilise the LiveDrive messages to simulate the workload for the users. This results in 2124 queries about "incidents" (LiveDrive messages) where we try to predict users that contributed tweets about the incident, that is, tweets that were a) geo-coded and contained usable location keywords and b) validated as described in the previous section. User contributions up until the query time (i.e. time of the LiveDrive "incident" tweet) are used to build user profiles. Note that this includes tweets that were not validated by a LiveDrive event and did not necessarily contain location keywords. Users sometimes signal there is no problem (anymore) at a particular location, and while these tweets are usually not validated by LiveDrive, they still provide information about the typical location and time a user is active. User contributed tweets that have been validated are withheld until after the given LiveDrive message has been processed in order to not unfairly advantage the algorithm.

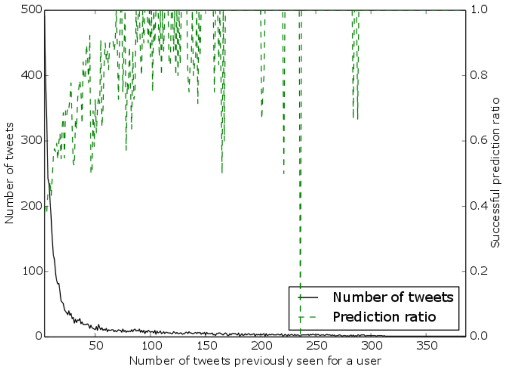

Figure 3: The frequency of a contributing user to be included in the Pareto optimal set of selected users without budget constraint increases with the number of tweets previously seen for that user.

4.1 User Candidate Selection

We first include four metrics as selection criteria: number of tweets, time probability, location probability and the reputation of the user. Our goal is to accurately recommend which users to query based on the time of day and the location keywords of interest. A recommendation is a success if a user who actually did contribute a validated tweet is included in the Pareto optimal solution set.

We achieved an accuracy of 77.16% with an infinite budget. We only consider contributing users who have a history of at least 4 tweets, as with fewer tweets the system may not know enough about a user. Indeed, the accuracy for any particular user increases with the number of tweets from that user that the system as seen before, as shown by Figure 3 (dashed green line). The accuracy starts at 40% and increases to above 60% after the first 20 tweets. This indicates our approach is able to capture the behaviour of users. Depending on the accuracy required from the system, it may be beneficial to increase the minimum number of tweets that must be observed from a user before considering it for a query.

Figure 3 also shows the number of users that have received a query with a given number of previously observed tweets. That curve is not monotonically decreasing because no query is emitted for non-validated tweets. Because users emit tweets at different rates, the overall accuracy does not increase much over time.

4.2 Budget Constraints

We now investigate the performance of our approach under a budget constraint. The initial candidates are selected using the Pareto optimisation approach and users are then ranked based on three possible strategies. Figure 4 shows the performance with a decreasing budget ranking the candidates based on time probability, location probability and reputation. As can be seen the budget does not have an impact until it drops below 50 queries. The reputation ranked strategy however, only drops 2% when the budget is reduced to 25 and still maintains a success rate of 65.34% with a budget of 10. The success rate drops to only 21.56% with a budget of one, however it should be noted that this measure of success is whether we accurately predicted which user contributed a relevant tweet. Our end goal is to identify users who are good candidates to poll for updates taking into account the time of day and location, meaning those recommended with a budget of 1 may still be valid candidates who could respond if probed with a high probability. Ranking based on location and time on the other hand suffer a large drop in performance as the budget available decreases, demonstrating that once users with a reasonable probability based on time and location profiles are identified, taking into account the reputation of the user becomes the more successful strategy.

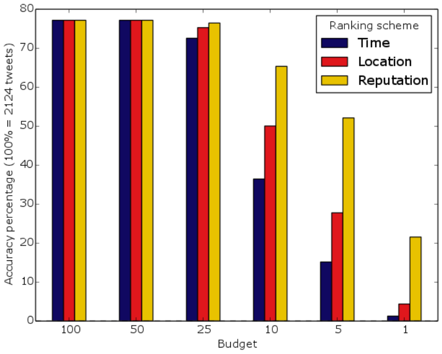

Figure 4: Success Rate for varying budget and candidates ranking schemes

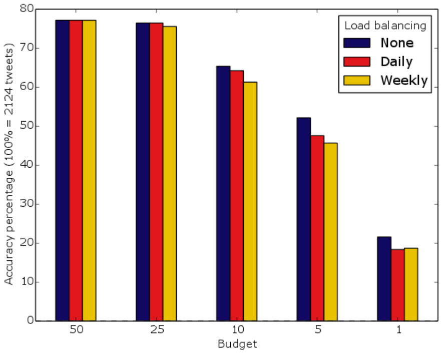

Figure 5: Success Rate with varying budget and load balancing constraints

4.3 User Query Load

Figure 6 shows a violin plot[5] demonstrating the distribution of queries across the user population with varying budget constraints. As can be seen, when no load balancing strategy is in place, a single user could potentially be queried 2124 times when considering a budget of 25 over the course of the 15 month experiment. This is clearly too much load to place on a single user in the system and has the potential to cause this user to opt out of the system. As a result we introduce an additional decay factor in the user reputation as proposed in equation 2, however the value of k is determined by the number of times the user was queried in the given time unit. Figure 6 shows the impact on load distribution when considering the number of queries the user has received that day, or that week. Depending on the budget constraints the additional decay factor helps to distribute the load more evenly across the available users.

Figure 5 shows the impact on accuracy when predicting the exact user who contributed the tweet when employing the load balancing schemes. As can be seen, there is only a small reduction in accuracy due to the decision to distribute the load across users. When considering a budget of 5 the accuracy drops from 52.11% to 47.55% and 45.66% when considering the number of queries a user has already been given that day and that week respectively.

5. CONCLUSION AND FUTURE WORK

We present a strategy to recommend users who may be probed in order to provide crowd-sourced traffic updates. We evaluate our strategy using a novel dataset consisting of real user-service interactions over the course of 15 months. We demonstrate the ability to accurately predict which user is appropriate to provide information depending on the time of day and location of interest. We study the behaviour of our algorithm with a budget of maximum users that can be queried. We also consider different load balancing approaches to better spread queries between users. While this can lead to a decrease in accuracy, it can both keep medium users engaged in the system and avoid sending too many queries to good users, which may make them leave our crowd sourcing scheme. Using real tweets addressed to a Dublin traffic radio, our empirical evaluation shows our approach can accurately predict who will contribute tweets about an incident at a particular place and time. While these first results are promising, we believe there is room for improvement. From an algorithmic point of view, more advanced method from reinforcement learning could increase accuracy or the learning rate. From a practical point of view, we hope to test our approach by actively querying users. Human traffic controllers could trigger the queries. However, automatically sending queries based on sensor measurements may be a more promising approach.

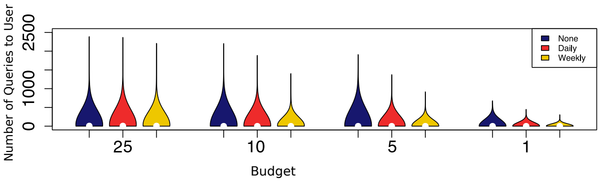

Figure 6: Effect of load balancing on the number of queries per users

Acknowledgements

This work is partly funded by the EU FP7 INSIGHT project (project number 318225).

6. REFERENCES

[1] Alt, F., Shirazi, A.S., Schmidt, A., Kramer, U., Nawaz, Z.: Location-based crowdsourcing: extending crowdsourcing to the real world. In: NordCHI'10. pp. 13–22. ACM (2010)

[2] Artikis, A., Weidlich, M., Schnitzler, F., Boutsis, I., Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., Kalogeraki, V., Marecek, J., Gal, A., Mannor, S., Gunopulos, D., Kinane, D.: Heterogeneous stream processing and crowdsourcing for urban traffic management. In: EDBT. pp. 712–723 (2014)

[3] Brink, T., Keeney, R., Raiffa, H.: Decisions with multiple objectives, preferences and value tradeoffs. Behavioral Science 39(2), 169–170 (1994), http://dx.doi.org/10.1002/bs.3830390206

[4] Daly, E.M., Lecue, F., Bicer, V.: Westland row why so slow?: Fusing social media and linked data sources for understanding real-time traffic conditions. In: IUI '13. pp. 203–212 (2013)

[5] Fuchs, G., Andrienko, N., Andrienko, G., Bothe, S., Stange, H.: Tracing the german centennial flood in the stream of tweets: First lessons learned pp. 31–38 (2013)

[6] Kakantousis, T., Boutsis, I., Kalogeraki, V., Gunopulos, D., Gasparis, G., Dou, A.: Misco: A system for data analysis applications on networks of smartphones using mapreduce. In: MDM '12. pp. 356–359 (2012)

[7] Kazemi, L., Shahabi, C.: Geocrowd: enabling query answering with spatial crowdsourcing. In: SIGSPATIAL '12. pp. 189–198 (2012)

[8] Krause, A., Horvitz, E., Kansal, A., Zhao, F.: Toward community sensing. In: IPSN '08. pp. 481–492 (2008)

[9] Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW '10. pp. 851–860 (2010)

[10] Starbird, K., Muzny, G., Palen, L.: Learning from the crowd: Collaborative filtering techniques for identifying on-the-ground twitterers during mass disruptions. In: Proc. 9th Int. Conf. Inf. Syst. Crisis Response Manag. Iscram (2012)

[11] Venanzi, M., Rogers, A., Jennings, N.R.: Crowdsourcing spatial phenomena using trust-based heteroskedastic gaussian processes. In: HCOMP. pp. 182–189 (2013)

[12] Victorino, J.N.C., Estuar, M.J.E.: Profiling flood risk through crowdsourced flood level reports. In: ICITCS'14. pp. 1–4 (2014)

[5] http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/violplot.htm