Mining Large-Scale GPS Streams for Connectivity Refinement of Road Maps

Yin Wang
Facebook, Menlo Park
yinwang@fb.com

Hong Wei
Shanghai Jiao Tong University
keith.collens@sjtu.edu.cn

George Forman
HP Labs, Palo Alto
george.forman@hp.com

ABSTRACT

As people increasingly rely on road maps in the digital age, manually maintained maps cannot keep up with the demand for accuracy and freshness, evidenced by the recent iOS map incident and the bidding war for Waze. There are many research works on automatic map inference using GPS data, and some have suggested that Google and Waze automate their map update processes to some degree with user data. However, existing *published* work focuses on refining road geometry. In reality, connectivity issues at intersections, including missing connections and unmarked turn restrictions, are much more prevalent and also more difficult to infer. In this paper, we report on our study on the connectivity issues in the OSM Shanghai map using 21 months of GPS data from over 10,000 taxis. We first adapt a robust *map matching* algorithm to detect missing intersections, and then train a time-series detection model for every turn possibility of every intersection using supervised learning.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Performance

Keywords

spatial data mining, GPS, map inference, road maps

1. INTRODUCTION

As people increasingly rely on navigation using smartphones for everyday activities, inaccurate maps have substantial economic consequences and even threaten safety, evidenced by the recent quality problem of iOS 6 maps, and the subsequent bidding war for Waze. The root cause of the map quality problem is the manual map creation and update process, which is costly, error prone, and cannot keep up with the aggregated rate of changes to the entire road network.

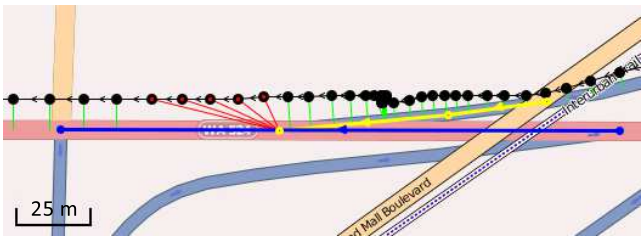


Figure 1: Map connectivity error in SIGSPATIAL Cup 12: Yellow and blue roads are mistakenly disconnected (green line segments connect GPS samples to correctly matched locations, while red line segments connect to incorrectly matched locations due to the missing connection).

In response to the challenge, there is a significant amount of work on automated map inference and update, typically using either aerial imagery [8, 11] or GPS data [6, 9, 5, 12]. The use of aerial imagery for road recognition is more effective in identifying highways; recognizing narrow local roads is very challenging. And it is ineffectual for detecting most turn restrictions. CrowdAtlas [12] focuses on updating roads, including geometry refinement and new road detection. It first employs *map matching* [10] algorithms to break GPS traces into matched segments that are well aligned with the map, and unmatched segments that are discrepant with the existing map. Matched segments are then used to *refine* the map, and unmatched segments are used to *augment* the map.

In practice, we find that intersection errors are more prevalent and are very difficult to detect, including erroneous or wholly missing intersections, as well as incorrect turn restrictions. For example, in the map matching competition of SIGSPATIAL Cup 12 [4], many teams discovered a discrepancy between a training trace and the map, shown in Figure 1. This type of error is almost impossible to detect without trace data, because the roads often touch each other or are extremely close, rendering visual review ineffective.

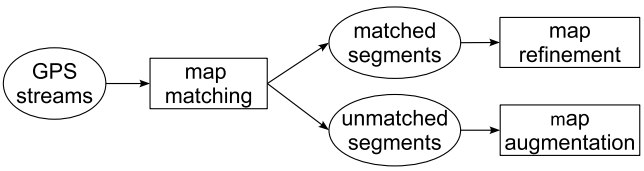


Figure 2: Map update using GPS streams

SIGSPATIAL'13, Nov 05-08 2013, Orlando, FL, USA
ACM 978-1-4503-2521-9/13/11. http://dx.doi.org/10.1145/2525314.2525457

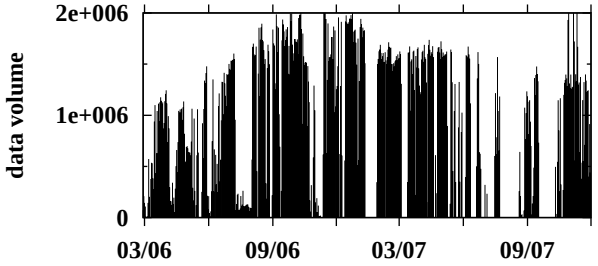


Figure 3: Highly variable data volume per day.

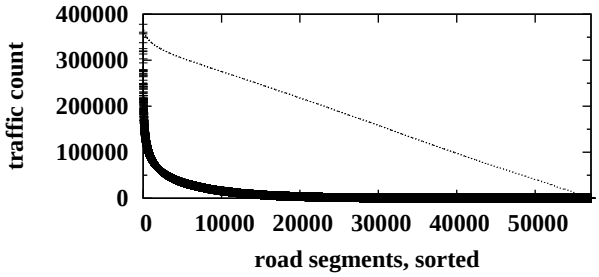


Figure 4: Highly skewed traffic per road.

Turn restriction errors are more difficult to find than missing intersections, and they plague even commercial maps. For example, both Google and Bing maps have serious turn restriction errors near the intersection of Market St. and Freeway 101 in San Francisco, one of the busiest intersections of the city, which has caused fatal accidents in the past due to illegal turns. There is almost no research on map connectivity inference from GPS trace data. A rare exception is [7], which detects the locations of road intersections, absent a base map. It did not consider the detection of turn restrictions or updating existing maps.

In this paper, we extend our map update framework [12] to address connectivity issues in existing road maps. First we use unmatched trace segments to detect missing intersections and automatically add them to the map when sufficient GPS evidence has been collected. Then for every turn possibility of every intersection (both existing and new), we build a time series model to detect its usage over time. This can tell us whether the turn is regularly used and whether we can confidently infer permitted turns and forbidden turn restrictions. We evaluate our approach on 21 months of GPS data from thousands of taxis in Shanghai, using OSM as our base map. A month of data is available online [3].

Contributions: This work is the first to use data mining of large scale GPS traces in order to automatically correct and refine road intersections. We elucidate the value, requirements, and practical challenges for this novel application. And we describe the application of data mining methods to effectively infer updates for the map. Finally, through extensive search and verification, we validated some of the missing intersections that were identified.

Section 2 describes our proposed methods, which are validated by the empirical results in Section 3. And Section 4 concludes, with perspectives on future work.

2. SOLUTION

The core ideas of connectivity refinement are straightforward. An intersection is missing between disconnected roads A and B if there is a vehicle trace that first matches to A and then to B. For every turn possibility of every intersection, we build a time series model for its traffic. If there is never any traffic, the turn is prohibited. If the traffic changes from none to a positive value or vice versa, the turn becomes open or closed, respectively.

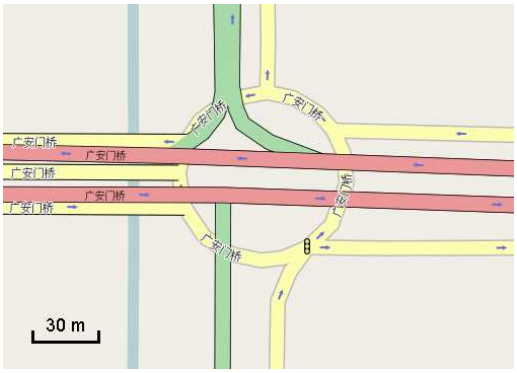


Figure 5: Need to consider every pair of disconnected roads that touch each other for missing intersection detection, at least 8 pairs in this case.

However, we are facing a variety of challenges. First, the data source may have occasional days with missing data, or days having only a fraction of the usual volume (see Figure 3). Second, there is always noise in the data, including inherent GPS noise and drivers making illegal turns. Third, there is a great deal of disparity in data coverage of different types of roads and different areas of the map, as shown by the dark curve in Figure 4. (The thin line shows the log(count) scaled to fill the graph; its good linear fit reveals an exponential distribution.)

To overcome these difficulties, instead of considering the absolute traffic turning from road A to B, we calculate the ratio between the turning traffic and the minimum traffic passing on A and B, i.e.,

$$turnRatio(A,B) = \frac{\text{traces turning from } A \text{ to } B}{\min(\text{traces passing } A, \text{traces passing } B)}$$

Therefore, noisy illegal turns at a busy intersection are distinguished from normal turn traffic at an infrequently traveled intersection. Because of data disparity, we consider the minimum traffic on A and B instead of average or other such measurements.

2.1 Detecting Missing Intersections

As stated at the beginning of this section, there is a missing intersection between two disconnected roads if a trace matches to them in a sequence. In practice with real-world large GPS datasets, traces connect numerous pairs of roads due to noise and sparse sampling [13]. Fortunately, with a base map we only need to focus on two types of potential missing intersections: unconnected roads that touch each other, and roads whose endpoints are extremely close to another road (e.g., within 5 m). Although we consider only these two types of missing intersections, there are a significant number of them in a metropolis. For example in Beijing, there are 15,896 pairs of roads that touch each other, and 263 roads extremely close to other roads. Figure 5 shows a complex highway interchange that includes at least eight potentially missing intersections.

With the $turnRatio(A,B)$ calculation formula, we detect an intersection if the ratio exceeds a threshold, and the

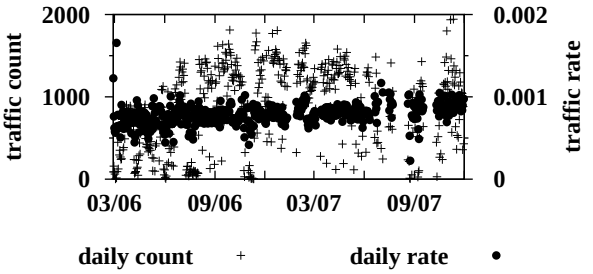


Figure 6: Counts show wider variation than rates.

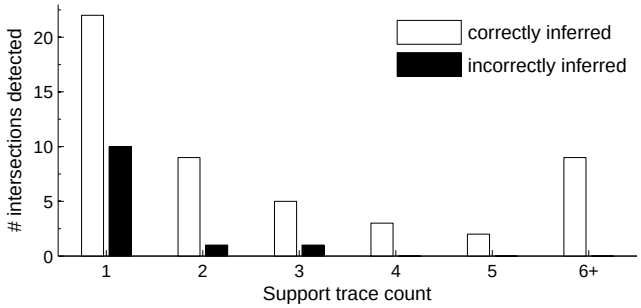


Figure 7: Missing intersections detected in Beijing.

passing traffic on A and B are sufficient. The latter avoids infrequently traveled roads with only noise data.

2.2 Detecting Turn Restrictions

For every turn possibility of every intersection both existing and newly inserted, we use the matched trace segments to detect turn restrictions. More specifically, similar to intersection detection, there is a turn restriction from A to B if $turnRatio(A, B)$ is less than a threshold, and there is sufficient traffic on both A and B. Here the latter requirement is crucial because the data coverage of different roads can be highly skewed (recall Figure 4). We must not declare a turn restriction from A to B if there is simply very little traffic on either road.

With continuous GPS streams, we must be cautious in declaring a permanent turn restriction, because there can be temporary closures due to accidents, maintenance, or events. Next we discuss our time series analysis to detect temporary or permanent changes in turn restriction rules.

2.3 Time Series Analysis

As the stream processing progresses, map-matched GPS traces are tallied for each turn. By considering traces instead of samples, we eliminate some of the variance due to traffic jams and parked vehicles. At the end of each time slot (daily) these counts are digested by all turns in parallel (1) to train traffic models and, once sufficiently trained, (2) to detect changes in the road network. Since we consider $turnRatio$ instead of the absolute trace count, our model is robust against the daily variance of the total data volume. Figure 6 illustrates for an example that the rates show much less variability than the counts. If the minimum trace count of A and B is less than a threshold, the day's count for the turn $A \rightarrow B$ is discarded.

Given these sequences of rates with missing values, we considered a variety of time-series analysis methods in the literature, including regression models, step detection, etc. Most traditional methods expect model normality and report statistically significant changes. In our application, however, there are a great many substantial changes in the traffic which are not useful for detecting openings/closures.

Although we desired to apply supervised machine learning methods, we first needed to build a labeled training set. Since historical turn restriction changes are almost impossible to verify, we develop our model using traffic data matched to each road and consider the detection of road closures instead. For long term closures, we can manually verify the results through the historical aerial imagery of Google Earth, as well as news archives. To help identify likely positives, we first used a 5-month data sample to develop various step-down detection models that sought, e.g., a 90% drop in traffic, modeling the expected rate with a moving average or with a control chart using the lower 3σ estimate. This was a laborious process, involving much iteration and visual performance inspection (as yet having no training set). Eventually we developed a satisfactory model that seeks an 80% drop in traffic from the minimum of a moving window of recent history that excludes any data that were considered closures. Rather than trigger on any day with a zero count, we compared the expected traffic threshold against the *maximum* of the most recent traffic rates. We used this handcrafted model to identify closure segments having periods of $\geq$2 weeks that closed and remained so through the end of the dataset.

Using the labeled series, we ran a sliding window across each, generating snapshots as training examples for supervised learning. The sliding window consisted of a month of history adjacent to a two week window of the most current data. For series labeled with a single closure, we took a single snapshot with the history/present window boundary aligned with the beginning of the labeled closure. For series with no closures, we generated snapshots iteratively with no overlap. This is consistent with the preponderance of negatives in this domain.

For each snapshot, we composed a feature vector of the mean, stdev., and various quantiles of the history window and the present window. We directly apply the above model to turn restriction change detection.

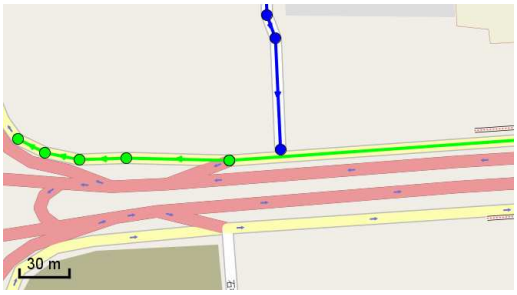
3. EMPIRICAL RESULTS

We use two taxi datasets for evaluation. For missing intersection detection, we use a relatively small dataset of 70 taxis from Beijing for one week in 2008. Turn restriction detection and time series analysis need datasets of longer durations. Therefore we use 21 months of taxi traces from Shanghai for evaluation, 02/27/2006–11/30/2007. Both the Beijing dataset and one month of the Shanghai dataset are available online [1]. Our base map is from OSM [2].

3.1 Intersection Detection

In Beijing, there are 15,896 pairs of roads that touch each other but are not connected, and 263 roads whose endpoints are within 5 m to other roads. Our Beijing dataset has traces that match to 57 and 5 of these two types of potential missing intersections, respectively. Among them, 43 and 4, respectively, are manually confirmed as real missing intersections using the Baidu map. Figure 7 shows all 57 + 5 = 62 intersections and their support trace count. By requiring at least three support traces, all false positives were eliminated. Most of these incorrectly

(a) touching but not connected



(b) close but not connected

Figure 8: Example missing intersections in Beijing

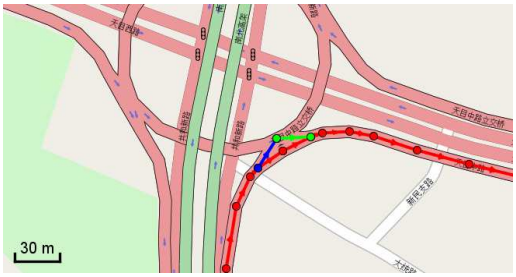


Figure 9: Turn restriction on bypass?

inferred intersections were due to noise. Figure 8 shows examples of the two types of missing intersections detected by our algorithm. In Figure 8a, the node of the blue-highlighted road missed the green-highlighted road slightly. In Figure 8b, the end node of the blue-highlighted road is less than 5 m from the green-highlighted road.

3.2 Turn Restriction Detection

Using the full 21 months of data in Shanghai, there are 103 turns without any trace matched at all where the incoming and outgoing branches both have at least 30 traces matched. Among these 103 turns, 70 are indeed turn restrictions not annotated by OSM. For the remaining 33 turns, 17 of them are false positives due to either insufficient data coverage or possible map changes since the GPS data was taken (we didn't have the map of 2007 available, so we verified using the latest Baidu online map). The last 16 turns are on bypasses or detours. Figure 9 is an example. Baidu and Google maps use a different representation for this highway interchange. Our turn restriction algorithm can ignore this type of turn by detecting whether there is a shortcut. If we set the *turnRatio* threshold to be 10% instead of 0, there are 81 turns detected and 26 of them are false positives. Figure 10 is an example turn restriction detected by our algorithm. Vehicles cannot make turns from the upper right road to the orthogonal road in either direction, or vice versa.

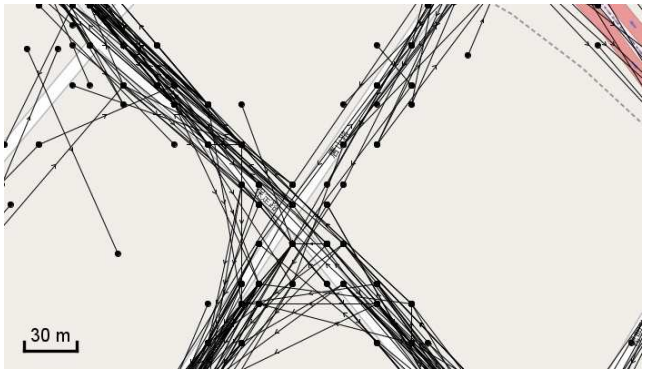


Figure 10: Example turn restriction in Shanghai

4. CONCLUSION

Lack of intersection connectivity can result in poor route planning and confused GPS navigation aids; and missing turn restrictions can advise drivers to take illegal and dangerous forbidden turns. In this paper, we have shown the feasibility of a prototype application that detects missing intersections and turn restrictions in order to automate the update of digital maps, which have become a cornerstone resource for so many applications. While in this research we have used historical data from a fleet of over 10,000 taxis, we envision future streaming implementations that scavenge the data byproducts of a variety of fleet management deployments as well as mobile smartphones, given sufficiently anonymized, delayed, and/or opt-in data sources.

5. REFERENCES

[1] GPS dataset. `grid.sjtu.edu.cn/mapmatching/data`.

[2] OpenStreetMap. `www.openstreetmap.org`.

[3] SUVnet-Trace data. `wirelesslab.sjtu.edu.cn`.

[4] M. Ali, T. Rautman, J. Krumm, and A. Teredesai. ACM SIGSPATIAL Cup 2012. In *GIS*, 2012.

[5] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *ACM GIS*, 2012.

[6] L. Cao and J. Krumm. From GPS traces to a routable road map. In *ACM GIS*, 2009.

[7] A. Fathi and J. Krumm. Detecting road intersections from GPS traces. In *6th International Conference on Geographic Information Systems*, 2010.

[8] J. Hu et al. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE T. Geoscience and Remote Sensing*, 45(12-2):4144–4157, 2007.

[9] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu. Mining large-scale, sparse GPS traces for map inference: Comparison of approaches. In *KDD*, 2012.

[10] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *ACM GIS*, 2009.

[11] Y.-W. Seo, C. Urmson, and D. Wettergreen. Exploiting publicly available cartographic resources for aerial image analysis. In *SIGSPATIAL GIS*, 2012.

[12] Y. Wang et al. Crowdatlas: Self-updating maps for cloud and personal use. In *MobiSys*, 2013.

[13] Y. Wang, Y. Zhu, Z. He, Y. Yue, and Q. Li. Challenges and opportunities in exploiting large-scale GPS probe data. Technical Report HPL-2011-109, HP Labs, 2011.