Hierarchical Temporal Mobility Analysis

Adele Hedrick
Faculty of Science
UOIT
adele.hedrick@uoit.ca

Ken Q Pu
Faculty of Science
UOIT
ken.pu@uoit.ca

Ying Zhu
Faculty of Business and IT
UOIT
ying.zhu@uoit.ca

ABSTRACT

It's becoming a commonly accepted standard of living that we constantly have access to a mobile device which is basking in a sea of Wifi hotspots. By recording the observed Wifi signals over time, it is possible for the mobile phone to deduce the salient locations in its environment, and the mobility patterns of the user. In real-life, fluctuations of the Wifi hotspots and the unreliability of mobile phone antenna necessarily creates false readings and missed readings, making the location identification problem and its related problems particularly challenging.

In this paper, we propose a family of algorithms to perform the tasks of location identification, mobility inference, and localization. Our algorithms are able to handle the noisy reading observed in real-life application. Furthermore, our location identification algorithm constructs a hierarchy of salient locations providing a multiresolution model of the environment.

Keywords

ACM proceedings; Mobility; Temporal Analysis; Personal Data

1. INTRODUCTION

WebDB'16 San Francisco, USA

DOI: 10.1145/1235

2. PROBLEM DEFINITION

A mobile device can make a scan. We refer to each scan as a *reading*. Each reading is defined as $\langle t(r), \mathbf{B}(r)\rangle$ where $t(r)$ is the timestamp of the reading, and $\mathbf{B}(r) \subseteq$ BSSID is a set of BSSID of the wifi hotspots that the scan detected. For each BSSID $b \in \mathbf{B}(r)$ detected in the reading, we also have the SSID and the signal strength, written respectively as: BSSID(b) and $s(b|r)$. We assume that each BSSID has a unique SSID, while the strength of a BSSID is specific to a given reading.

2.1 Location identification and inference

Definition 1. *A timeline T is a sequence of readings. We denote T_i as the i-th reading of the timeline T.*

A segment *of the timeline S is a contiguous subsequence of T.*

Let $\mathcal{L}$ be a (unspecified) finite set of *locations*.

Definition 2 (Location identification). *A* location identification *problem consists of several subproblems:*

1. Identification *of the distinct locations $\mathcal{L}$ from a given timeline T.*
2. Inference *of the location of a given reading.*

2.2 Real-life challenges

- Power-aware sensing creates highly irregular sampling intervals.
- False positive in the readings - phantom BSSID
- False negatives in the readings - undetected BSSID
- Transient hotspots - while driving
- Non-stationary hotspots
- Location is naturally a hierarchical concept
- Online algorithm

3. THE ALGORITHM

```
(defn online-cluster
    [H tail C-new]
    (let [root (root H)
          prev (sibling tail)
          parent (parent tail)]
        (cond
            (= tail root)
                (set-root! H (new-cluster root C-new))
            (closer? (C-new tail) :than (prev tail))
                (do
                    (detach! H tail)
                    (promote! H prev)
                    (online-cluster H prev (new-cluster tail C-new)))
            :else
                (do
                    (online-cluster H parent C-new)))))
```

Figure 1: Algorithm to perform online clustering

3.1 Segmentation

Given the timeline T. We want to partition T into a collection of non-overlapping segments $\{S_i\}$, where each S_i is such that all the readings $r \in S_i$ are at the same location.

The challenge is that the locations, at this stage, are still unidentified, and furthemore, as stated in Section 2, we wish to solve the multiresolution localization problem. Thus, instead of partitioning the timeline, our objective is to perform a hierarchical segmentation of the timeline, as defined in Definition 3.

Definition 3 ((Binary) Hierarchical segmentation). *A binary hierarhical segmentation of a timeline T is defined as a binary tree of vertices. The leaf vertices contain a single reading, while the interior vertices have two adjacent vertices as children.*

We define the following:
Given a vertex v, the readings of v, $R(v)$, is given by:

- If $v \in \text{Leaves}(H)$, then $R(v) = \{r(v)\}$.
- Otherwise, $R(v) = R(\text{left}(v)) \cup R(\text{right}(v))$.

The constraint of H is such that:

- H must form a cover of T. Namely, every reading must belong to some $R(v)$.
- For each vertex v, $R(\text{left}(v))$ and $R(\text{right}(v))$ are contiguous segments in T.

We introduce a homogeneity measure for each vertex to measure whether the vertex covers a segment of readings at a single location.

Definition 4 (Vertex minsim). *Given vertex v covering a range of readings $R(v)$, we define the minsim as the mininum similarity:*

$$\text{minsim}(v) = \min\{\text{sim}(r, r') : r, r' \in R(v)\}$$

Proposition 1. minsim *is monotonic with respect to the hierarchy H, with the root of H having the minimal* minsim *value.*

```
(defn minsim [v]
    (if (or (zero? (minsim (left v)))
            (zero? (minsim (right v))))
        0
        (min (minsim (left v))
             (minsim (right v))
             (minsim (R v)))))
```

Figure 2: Minsim computation

We can efficiently compute the minsim(v) in a bottom-up fashion.

For overall large vertices, we rely on an approximation based on sampling:
minsim$(R(v))$ can be approximated by:

- Sample K_1 from $R(\text{left}(v))$ and K_2 from $R(\text{right}(v)$.
- Compute the minsim of the $K_1 \times K_2$ pairs, and use it as an estimation of minsim(v).

Definition 5 (Segmentation). *Given a threshold c, a vertex is a* maximally *homogeneous if:*

- minsim$(v) \geq c$, *and*
- minsim$(\text{parent}(v)) < c$

Let $\mathbf{S}(H|c)$ be the maximally homogeneous vertices in H with respect to threshold c.

The next stop is the clean $\mathbf{S}(H|c)$. For each vertex $v \in \mathbf{S}(H|c)$, we assert that it must have sufficient temporal and reading support. Let $\mathbf{S}^*(H|c)$ be the vertices v such that $\Delta t(v) \geq \tau$.

3.2 Location Identification

We need to construct a set of locations $\mathcal{L}$, and a mapping

$$h : \mathbf{S}^*(H|c) \to \mathcal{L}$$

that identifies the locations of the segments in $\mathbf{S}^*(H|c)$.

This is done using a fast clustering algorithm.

```
(loop [segments (segmentation H)
       locs     []]
    (cond
        (empty? segments) locs
        (empty? locs) (recur (rest segments)
                             (conj locs (first segments)))
        :else
            (let [seg (first segments)
                  segments (rest segments)]
                (if-let [loc (nearest locs seg)]
                    (recur segments
                           (assoc (:name loc) (augment loc segment)))
                    (recur segments
                           (conj locs (new-location seg)))))))
```

In hierarchical localization, we wish to generate a hierarchical organization of the fundamental locations. We already have a hierarchical organization of $\mathbf{S}^*(H|c)$. So, we want to use it to *induce* a hierarchy of locations.

4. RELATED WORK

[8, 5, 4, 3, 7, 1, 6, 2]

5. REFERENCES

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[2] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: human localization using mobile phones. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 149–160. ACM, 2010.

[3] B. Ferris, D. Fox, and N. D. Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[4] T. Garcia-Valverde, A. Garcia-Sola, H. Hagras, J. A. Dooley, V. Callaghan, and J. A. Botia. A fuzzy logic-based system for indoor localization using wifi in ambient intelligent environments. *Fuzzy Systems, IEEE Transactions on*, 21(4):702–718, 2013.

[5] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy. Precise indoor localization using smart phones. In *Proceedings of the international conference on Multimedia*, pages 787–790. ACM, 2010.

[6] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive computing*, pages 152–169. Springer, 2011.

[7] O. Serrano, L. Rodero Merino, V. Matellán Olivera, J. M. Cañas, et al. Robot localization using wifi signal without intensity map. 2012.

[8] S. Siddiqi, G. S. Sukhatme, and A. Howard. Experiments in monte-carlo localization using wifi signal strength. In *Proceedings of the International Conference on Advanced Robotics*, pages 210–223, 2003.

```
segments is a queue of S*(H|c)
forest
while (not-empty? segments)
     seg <- (pop segments)
     (add seg :to forest)
     parent = (parent-of seg)
     if (not-in-queue? parent)
        (push parent :into segments)

(defn add [seg :to forest]
    (let [subF (for [F forest :when (is-subset? F seg)
                    (location-of F))]
        (merge subF :in forest)))
```