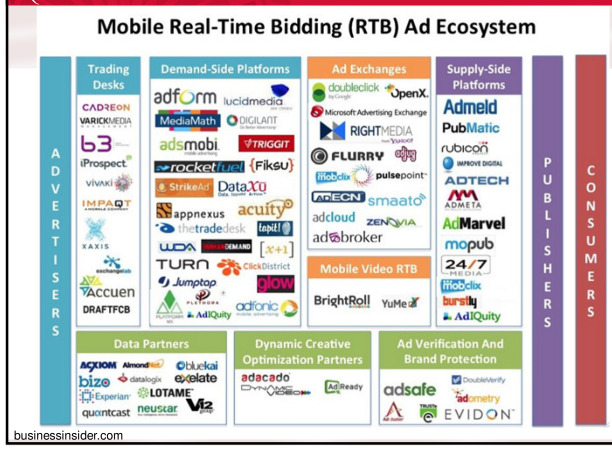
RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

MINING DATA FROM MOBILE DEVICES

Applications: Location, Ads, Privacy

Spiros Papadimitriou, *Tina Eliassi-Rad*
{spiros.papadimtriou, tina.eliassi}@rutgers.edu



businessinsider.com

Ad Targeting on Mobile Devices

- Need reliable location information
  - We collected 21.6M RTB requests on Wed 2/6/2013
  - The majority of them (57%) did **not** have location information
- Reasons behind the missing location information
  1. The RTB system did not forward it
  2. The SSP did not forward it
  3. The device did not capture it
  4. The user did not enable location-based services

- RTB: Real-Time Bid
- SSP: Supply Side Provider

Problem Definition

- *Accurately infer location information for IPs in RTB requests*
- What type of location information should we infer?
  - Latitude, longitude
  - Census Block Groups (CBGs)
  - Zip codes
  - …

privacy-friendly location Information

Census Block Groups (CBGs)

- Assumption
  - *CBGs comprise location information fine-grained enough for useful hyper-local ad targeting, yet coarse-grained enough to avoid major privacy concerns.*
- Why is this reasonable?
  - Covers a contiguous area
  - Never crosses state or county boundaries
  - Contains between 600 and 3,000 people
  - US is divided into ~212K CBGs

- US is divided into ~8.2M CBs
- US is divided into 43K zip codes

Geo-locating IP addresses on mobile networks is hard

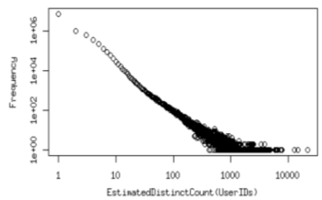
- Balakrishnan et al. (IMC 2009) examined properties of cell-phone IP addresses
- Mobile IPs are ephemeral and their addresses are itinerant
- Example: An individual cell phone can report different IP addresses to various servers within a short time-period

Answers to IP → Location queries provided by 7 geo-location services; the actual cell phone is in Mountain View, CA.

M. Balakrishnan, I. Mohomed, and V. Ramasubramanian. Where's that phone? Geolocating IP addresses on 3G networks. In IMC, pages 294–300, 2009.

Many devices often use the same public IP address

- Metwally & Paduano (KDD 2011) estimated the number of users of an IP address by keeping track of the application-specific traffic
- The primary goal of their work was to combat "abusive" traffic (such as DDoS attacks, ad click fraud and email spam) without violating the user privacy



Estimated Size of 10M Random IPs

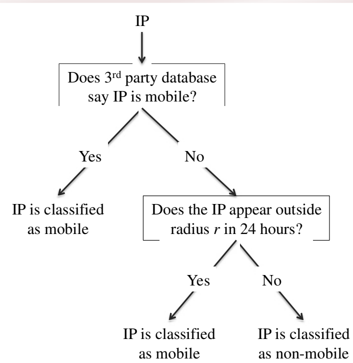
A. Metwally and M. Paduano. Estimating the number of users behind IP addresses for combating abusive traffic. In KDD, pages 249–257, 2011.

Hyperlocal: A Graph Mining Solution*

1. Classify IPs as mobile vs. non-mobile
2. Construct a movement graph with mobile and non-mobile IP nodes
3. Use a local relational classifier on each unknown node to infer latitude and longitude
4. Assign Census Block Group (CBG) ID to the inferred latitude and longitude using a *k*-nearest neighbor approach

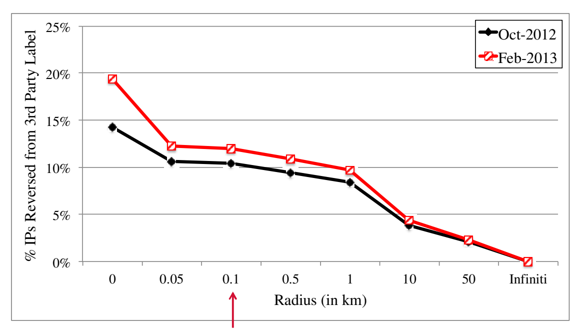
* http://eliassi.org/ESM2013TR.pdf

1. Classifying IPs as Mobile vs. Non-mobile



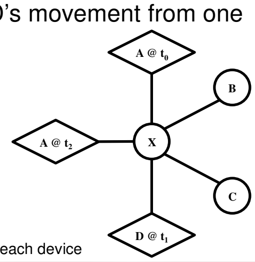
- Mobile IPs tend to change position more quickly than non-mobile IPs
- Default r = 100m (range of current Wi-Fi routers)

Reversing Third Party Decision



2. Constructing IP × IP Movement Graph

- Circle nodes are non-mobile IPs
- Diamond nodes are mobile IPs
  - Mobile IPs are time-stamped because they are transient
- An edge indicates an NUID's movement from one IP to another
  - Each edge stores
    - # of movements between its endpoints, <u>and</u>
    - inter-arrival times (IATs) for all movements across it



NUID: A network exchange ID associated with each device

3. Employing Local Relational Classifiers

- Why local?
  - The farther out one moves in the movement graph, the farther away one gets geographically
  - The movement graph is big ➔ non-local approaches can be computationally burdensome
- Local relational classifier used: *wvRN*
  - *wvRN* stands for weighted-vote Relational Neighbor [Macskassy & Provost, JMLR 2007]
  - *wvRN* estimates the class membership probabilities <u>and</u> assumes homophily in the network data

What weights should we use in *wvRN*?

- Number of movements
  - *Intuition: A node v will be closer in distance to its neighbors with whom it has more movements*
  - The data contains many edges with only one movement
- Minimum IAT
  - *Intuition: The longer the IAT, the longer distance the user has potentially moved (sans traffic)*
  - Uses the normalized minimum IAT between two IPs
  - Weight on the movement between node *v* and its neighbor *i* is

$$w_i = \frac{minIAT_v}{\min(t), \forall t \in IAT(v,i)}$$

- $minIAT_v$ is the minimum IAT across all of a given node *v*'s edge

wvRN Equations

- Inputs
  - Node *v*
  - Its neighbors *Nbr*(*v*)
  - Weights *W* on the edges between *v* and its neighbors
- *wvRN* equations

$$latitude(v) = \frac{\sum_{i \in Nbr(v)} w_i \cdot latitude(i)}{\sum_{i \in Nbr(v)} w_i}$$

$$longitude(v) = \frac{\sum_{i \in Nbr(v)} w_i \cdot longitude(i)}{\sum_{i \in Nbr(v)} w_i}$$

4. Assigning CBGs as Proxies for Location

- Infer location of a hashed public IP address at the CBG level and not at the ⟨latitude, longitude⟩ level
- Why use CBG ?
  1. It provides a more consistent labeling (as in location) for IPs
  2. It allows incorporation of external data that uses census data such as demographics
  3. In the majority of mobile applications, this level of location information is sufficient for a successful campaign

A *k*-Nearest Neighbor Approach for Assigning a CBG ID to a ⟨lat, lon⟩

- Inputs
  - Location of interest
    - *loc* = ⟨*lat*, *lon*⟩
  - For each CBG *i* in the US,
    - *i*'s centroid: $c_i = \langle lat_i, lon_i \rangle$
    - *i*'s area a_i in km
- Output
  - The CBG ID that contains *loc*

- Procedure
  1. *C* = centroids of the *k* nearest CBGs to *loc*
  2. For *j* in *C*
     - Calculate the distance between *loc* & centroid of the j^{th} nearest CBG
       - `d_j = distance(loc, c_j)`
     - Calculate the radius of CBG corresponding to the j^{th} centroid
       - `r_j = sqrt(a_j / π)`
     - Calculate the ratio of distance over radius
       - `ratio_j = d_j / r_j`
  3. Return the CBG ID corresponding to min($ratio_j$), for all *j* in *C*

Recap of *Hyperlocal*

1. Classify IPs as mobile vs. non-mobile
2. Construct a movement graph with mobile and non-mobile IP nodes
3. Use a local relational classifier on each unknown node to infer latitude and longitude
4. Assign Census Block Group (CBG) ID to the inferred latitude and longitude using a *k*-nearest neighbor approach

* http://eliassi.org/ESM2013TR.pdf

Experiments

- Experiments are divided into nine combinations of *infer location for* X *using* Y
- Values for *X* are 'all IPs', 'mobile IPs', and 'non-mobile IPs'
- Values for *Y* are 'all neighbors', 'mobile neighbors', and 'non-mobile neighbors'
- Measure accuracy by checking the predicted CBG ID vs. the actual CBG ID of an IP

Implementation & Runtime

- Hardware & OS: Macbook Pro with
  - CPU 2.66 GHz Intel Core i7
  - RAM 8 GB DDR3
  - hard drive 500 GB SSD
  - OS X 10.8
- Language: Python
- Supporting Software: NetworkX & MongoDB
- Runtime: On average 1.2 milliseconds to process each RTB request

Data

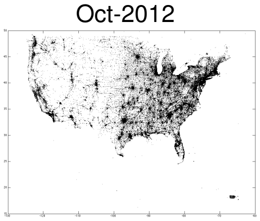
Data Name	Collection Date	# RTB Requests with Valid US NUIDs	% RTB Requests **without** Location	% RTB Requests from **Mobile** IPs
Oct-2012	Mon 10/01/2012	44.1M	36.5%	57.3%
Feb-2013	Wed 02/06/2013	21.6M	56.7%	47.7%

- From Oct-2012 to Feb-2013
  - # of RTB requests decreased by ~50%
    - Due to reductions from SSPs
  - # of requests without location information increased by ~55%
  - # of requests from mobile IPs decreased by ~17%

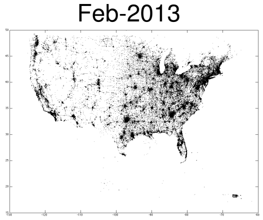
All IPs are hashed public IP addresses.

IPs on the US Map

Oct-2012

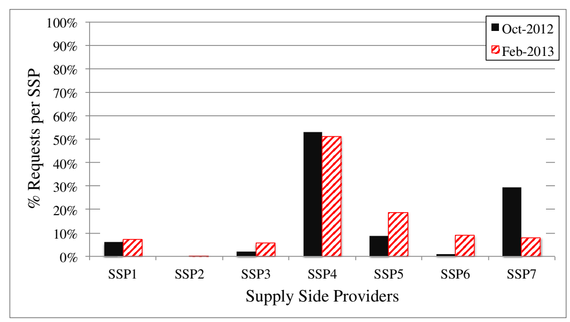


Feb-2013



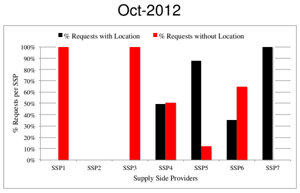
Data Characteristics per SSP

SSP4 provides about 50% of the requests for both datasets.

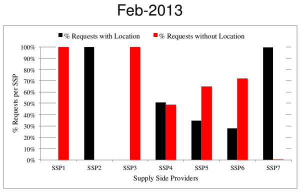


% Requests with and without Location Information per SSP

Oct-2012

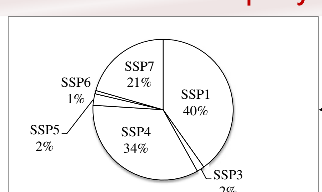


Feb-2013



- None of the requests from SSP1 and SSP3 have location information.
- All the requests from SSP7 have location information.

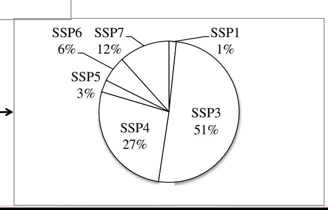
Oct-2012 Homophily per SSP



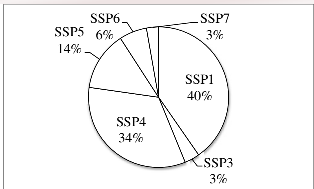
There is a considerable overlap with other SSPs that provide location information

% IPs from SSP1 requests that connect to other SSPs

% IPs from SSP3 requests that connect to other SSPs



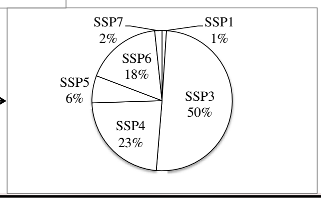
Feb-2013 Homophily per SSP



There is a considerable overlap with other SSPs that provide location information

% IPs from SSP1 requests that connect to other SSPs

% IPs from SSP3 requests that connect to other SSPs



Homophily in IP×IP Movement Graphs

$$homophily = \frac{\text{\# of movements whose IP endpoints are from the same SSP}}{\text{total \# of movements}}$$

Homophily	Oct-2012	Feb-2013
All movements	96.5%	90.8%
Mobile to mobile movements	98.6%	99.2%
Non-mobile to non-mobile movements	86.9%	78.1%

There is a considerable homophily in the IP×IP movement graph

Core Results: wvRN(minIAT) vs. wvRN(numMoves)

Infer location for **all** IPs . . .	Accuracy wvRN(minIAT) Oct-2012 | Feb-2013	Accuracy wvRN(numMoves) Oct-2012 | Feb-2013	Number of Predictions Oct-2012 | Feb-2013	% Mobile in Predictions Oct-2012 | Feb-2013
using all neighbors	71.6 | 70.8	69.7 | 69.3	1,189,679 | 328,015	91.0% | 83.8%
using mobile neighbors	**74.4** | **74.8**	**72.6** | **73.5**	1,077,644 | 278,674	93.2% | 87.4%
using non-mobile neighbors	51.9 | 57.7	51.7 | 57.3	98,338 | 40,777	68.7% | 62.1%
Infer location for **mobile** IPs . . .				
using all neighbors	74.2 | 75.0	72.3 | 73.6	1,082,566 | 274,900	100% | 100%
using mobile neighbors	**76.6** | **79.1**	**74.7** | **77.9**	1,004,601 | 243,630	100% | 100%
using non-mobile neighbors	50.4 | 54.7	50.2 | 54.5	67,553 | 25,323	100% | 100%
Infer location for **non-mobile** IPs . . .				
using all neighbors	45.5 | 49.0	44.0 | 46.8	107,113 | 53,115	0% | 0%
using mobile neighbors	44.0 | 45.1	42.5 | 43.0	73,043 | 35,044	0% | 0%
using non-mobile neighbors	**55.4** | **62.7**	**55.0** | **62.0**	30,785 | 15,454	0% | 0%

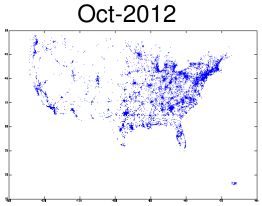
- Differences between the two methods are not statistically significant at the 0.05 level
- Number of predictions varies depending on the particular inference and the neighbor types used in the inference process

IATs on movement edges are correlated with distances

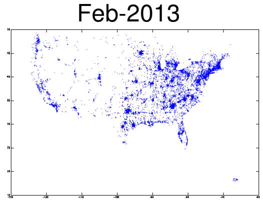
- Shorter IAT, shorter distance
- For IPs with only one known neighbor, restricting IAT to ≤ 60 minutes
  - Improves accuracy by an average of 12% on Oct-2012 and 23% on Feb-2013 data
  - Reduces the number of predictions by an average of 4 times for Oct-2012 and 5 times for Feb-2012
- Restricting IATs to > 60 minutes decrease accuracy

Correct (Blue) & Incorrect (Red) Classification

Oct-2012



Feb-2013

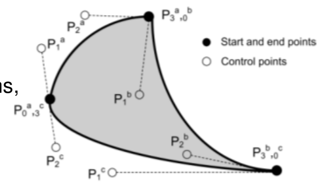


Limitations of a Graph Mining Appraoch

- Cannot infer location for IPs with no neighbors
  - Use other info – e.g., site visits; subnet info, etc.
- Cannot infer location for IPs with no known neighbors
  - Use collective classification.

Related Work [Wong et al. NSDI 2007]

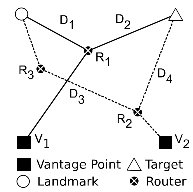
- Locate IPs by
  - representing node positions through regions,
  - expressing constraints as areas, and
  - computing locations by solving a system of geometric constraints
- Relies on pings to estimate the round-trip time between two IPs



B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A comprehensive framework for the geolocalization of internet hosts. In NSDI, pages 23–23, 2007.

Related Work [Wang et al. NSDI 2011]

- A client-independent geo-location system
- Like [Wong et al. NSDI 2007] relies on pings to estimate the round-trip time between two IPs
- Also relies on landmarks, which are collected manually



Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. Towards street-level client-independent IP geolocation. In NSDI, pages 27–27, 2011.

Recap & Open Problems

- Graph mining on just the structure of an IP×IP movement graph to infer locations, in terms of CBGs, for hashed public IP addresses produces an accuracy of ~75%
- Results are impressive since estimating the correct CBG is out of 212K possibilities
- **Open problems**
  - Inference on truncated IP addresses
  - Constrained collective classification

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

MINING DATA FROM MOBILE DEVICES

Applications: Location, Ads, Privacy

Spiros Papadimitriou, *Tina Eliassi-Rad*
{spiros.papadimtriou, tina.eliassi}@rutgers.edu

License