Location Prediction

Christian Grothoff

Berner Fachhochschule

April 5, 2019

Learning Objectives

- Know why location prediction is important
- Understand basics of human movement
- Creation of data set as basis for prediction
- Understand techniques for location prediction
- Realize simple prediction-based application

Context-Aware Applications

- Applications that adapt to *context*
- Context includes:
  - User behavior
    - Where the user goes
    - What the user does
  - Environment
    - Available connectivity
    - Charging opportunities
    - Location

Context-Aware Applications

- Applications that adapt to *context*
- Context includes:
  - User behavior
    - Where the user goes
    - What the user does
  - Environment
    - Available connectivity
    - Charging opportunities
    - Location
- Observations:
  - *Location* links facts
  - Predicting location $\implies$ predicting future context

Example Use of Context Prediction

- Automatically turn heater on when user heads home
- Predict presence of friends nearby
- Vary QoS according to future energy availability
- Prefetching messages or alerts / delaying uploads

Prefetching

- Potential benefits:
  - Saves energy
  - Conserves data allowance
  - Reduces network congestion
  - Reduces latency
  - Hides spotty network coverage
  - Reduces dependency on centralized infrastructure

---

[1] O'Donnell and Draper. *How Machine Delays Change User Strategies.* SIGCHI Bull., 1996.

[2] Trestian, et al., *Connecting People, Locations And Interests In A Mobile 3g Network.* ACM SIGCOMM, 2009.

Prefetching

- Potential benefits:
  - Saves energy
  - Conserves data allowance
  - Reduces network congestion
  - Reduces latency
  - Hides spotty network coverage
  - Reduces dependency on centralized infrastructure
- More than nice to have: potential *emergent behavior*
  - Reducing latency changes how people interact with programs[1]
  - Users conservative about energy and bandwidth use[2]

---

[1] O'Donnell and Draper. *How Machine Delays Change User Strategies.* SIGCHI Bull., 1996.

[2] Trestian, et al., *Connecting People, Locations And Interests In A Mobile 3g Network.* ACM SIGCOMM, 2009.

Location for Context-aware Applications

- Requirements
  - Current *place* (home, work, etc.)
  - Geographic coordinates not always required

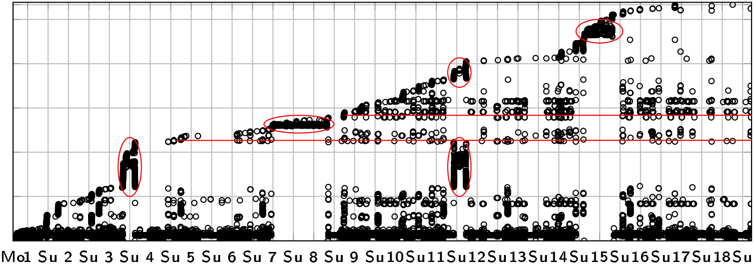
    $\Longrightarrow$ potential for increased privacy / energy savings

Location for Context-aware Applications

- Requirements
  - Current *place* (home, work, etc.)
  - Geographic coordinates not always required

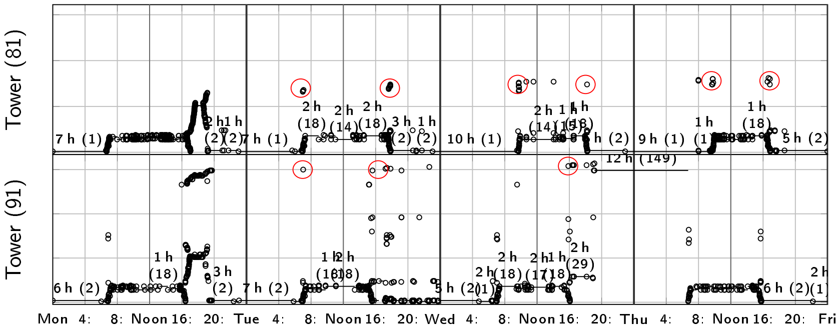
    $\Longrightarrow$ potential for increased privacy / energy savings
- GPS
  - Energy intense
  - No indoor coverage
  - Urban canyons
- Cell towers as landmarks
  - Available for "free"
  - Requires *cleaning*
    - Tower transitions when stationary

Overview



- Main location ("home") appears as a black line (bottom)
- Regular activities appear as dotted lines
- 4 trips - minor *regime changes*
  - Week 3 & week 11 to same location
  - Similar route, but note newly discovered towers
  - Week 7/8 & Week 15 trips probably with plane

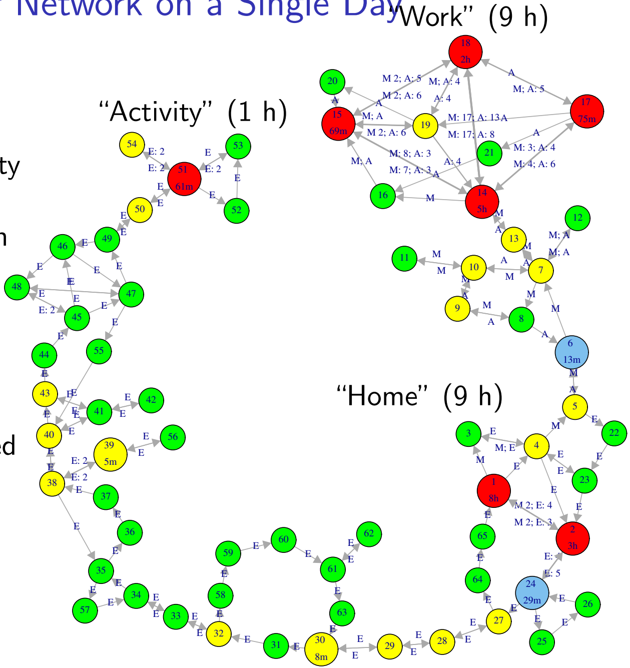
Week-by-Week View



- "home" at night (except second Wed. night)
- At "work" during the day
- Path to and from "work" is similar, but see new towers
- "activity" Monday evenings

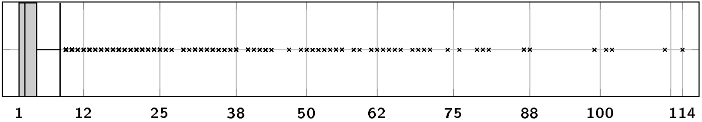
Induced Cell Tower Network on a Single Day

- 3 major areas of activity
- Device connects to multiple towers in each area; many transitions
  - Device samples nearby cells
  - Core & periphery towers
- Effect more pronounced with more data



Tower Sampling Experiment

- Is the device really stationary during some transitions?

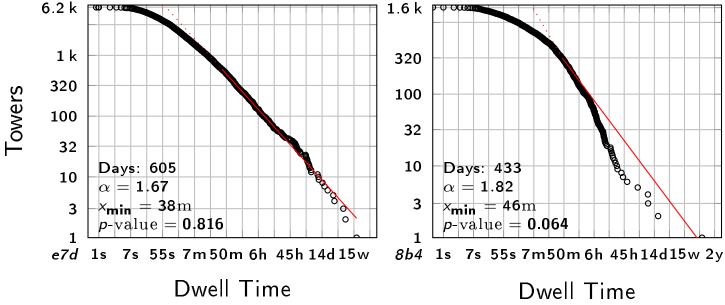


- # Towers seen during each **wall charge** (> 30 min.)
  - 6289 wall charges across all traces
  - Median: 2 (MAD: 1.48)
  - Upper quartile: 4

Tower Sampling is Real

- Locations not covered by a single tower
- Phone appears to sample towers in its vicinity
- Conclusions
  - Cell tower data higher resolution than places
  - Tower transitions do not correspond to user movement
  - Need to aggregate towers to identify:
    - Landmarks
    - User movement

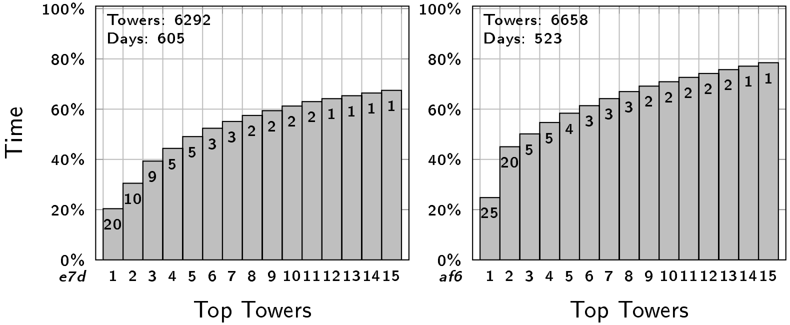
Time Spent at Each Tower (1/2)



- CCDF plots of the total time spent at a tower (by user)
- 44 of the fits (75%) are significant fits to a power law

$\Longrightarrow$ Users may visits thousands of towers, but spend nearly all of their time at a few locations

Time Spent at Each Tower (2/2)



- CDF of time at top towers
- Top few towers *dominate*

Approaches for mapping Tower → Place

- Collapse oscillation sequences
  - Relatively simple heuristic
- Place detection

Place Detection

- Observation:
  - Places automatically carved out via user movement
  - Places are islands of high dwell time

Related Work (1/2): Geography

- Scellato et al., Kim et al.
  - Place a 2-D Gaussian at each GPS sample & normalize
  - Islands above a threshold (15% of max) $\implies$ place
  - Recall:
    - Tower dwell time consistent with a power law

  $\implies$ 15% will only identify 1–3 places!

Related Work (2/2): Network Theory

- Eagle et al.
  - Community detection
    - Partition graph such that the number of edges between subgraphs is lower than expected
  - Makes places too large (include routes)
  - Computationally expensive and thus can only be run offline

A Good Strategy

- Identify graph structures that are typical of places:
  - Primary characteristic: High density subgraphs
  - Look for cliques, size ≥ 4

When to Run

Look for new subgraphs *whenever* there is a new edge

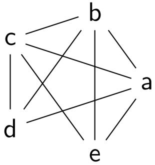
- Only need to look near the edge

$\Longrightarrow$ fast

$\Longrightarrow$ appropriate for online use

Naming

- When merging, use longest used name
  - Example: *a*: 3 h, *b*: 4 h $\implies$ *b*
  - But, if tower *b* called *b* for 0.5 h and *x* for 3.5 h, then use *x*
  - Overlapping clusters usually share a name
  - Example: 5-clique missing 1 edge $\implies$ two 4-cliques



Project

You implemented an (background) App that:

- Frequently records user's location[3]
- Location method is up to you (Cell tower, WLAN, GPS, multiple)
- Possibly record auxiliary data (power status, usage, etc.)
- Exported data via CSV, MQTT or HTTP
- Imported GPS locations into GIS database

You should have 2-3 week mobility data by now!

Now implement tools to (if applicable):

- Convert CSV to (graphviz/dot) transition graph
- Cluster cell towers into locations
- Compute location dwell time statistics

[3] Ideally, batch write to disk to safe battery!

Location Prediction: Related Work Overview

- Most work focused on predicting *next* tower
  - Relevant to network management
  - Approaches:
    - Markov chain (François et al., Song et al.)
    - Graphical models (Eagle et al.)
    - LZ-based predictor (Song et al.)
- Location in x hours:
  - Non-linear time series (NextPlace, Scellato et al.)
  - Recognize routes (Laasonen)
  - $P(\mathrm{place}|\mathrm{tod})$ (Burbey and Martin)
    - Simple scheme
    - Extensions by Walfield et al. work best
    - ⇒ We will focus on this scheme

Simple Idea

- Solve: $argmax_{t \in T} P(t|\mathbf{c})$
  - t: tower aggregate
  - $\mathbf{c}$: a set of conditions

Simple Idea

- Solve: $argmax_{t \in T} P(t|\mathbf{c})$
  - t: tower aggregate
  - $\mathbf{c}$: a set of conditions
- Why argmax?
  - Simplicity of evaluation
  - NextPlace does it
  - Easily modified to return rank or whole CPT, if appropriate

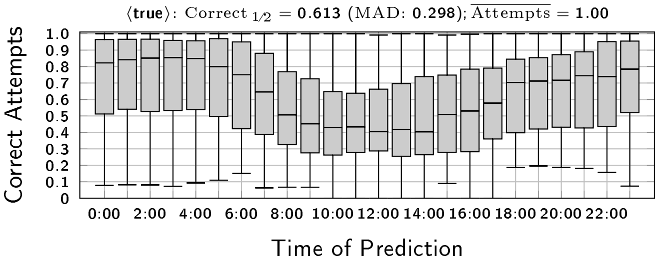
Evaluation

- Every half hour, make a series of predictions
- Predictions for 0.5 h, 1.5 h, . . . , 23.5 hours in the future
- Prediction correct if predicted aggregate visited within ± 15 min of prediction time

Evaluation

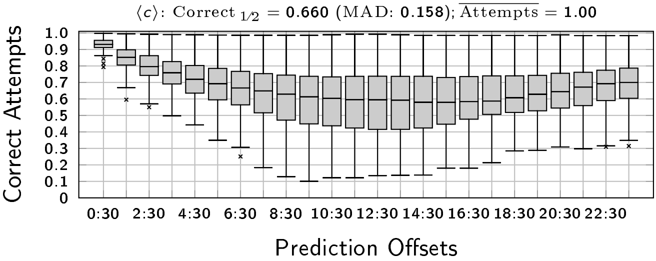
- Every half hour, make a series of predictions
- Predictions for 0.5 h, 1.5 h, . . . , 23.5 hours in the future
- Prediction correct if predicted aggregate visited within ± 15 min of prediction time
- Only attempt a prediction if **c** has $\geq 2h$ of data
  - Larger reduces attempts
  - Too large ($\geq 8h$) also reduces precision
    - Rarely visited locations apparently highly predictive
    - e.g., after shopping, user goes home

Baseline 1/2



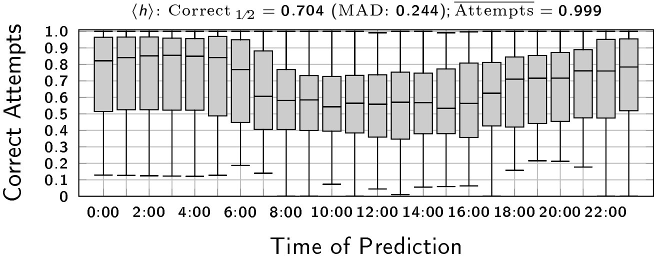
- Unconditional predictor
  - Current dominant aggregate
  - Probably where user usually sleeps
- Results consistent with power law behavior

Baseline 2/2



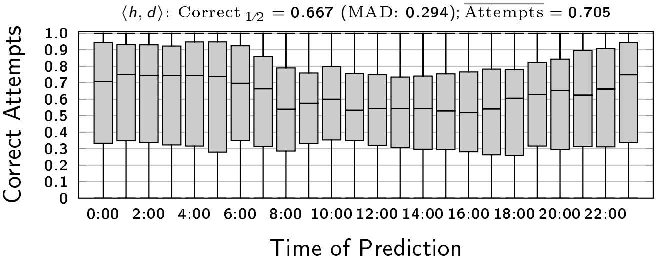
- Note: x axis is prediction offset, not time of day
- Current tower predictor
- Strong tendency to stay at a location for at least half an hour
- Increase for $\Delta > 17h \implies$ diurnal behavior

Time of Day



- Idea: daily routines
- Condition on current hour or half hour
  - Both perform similarly
  - We prefer hour due to smaller CPT

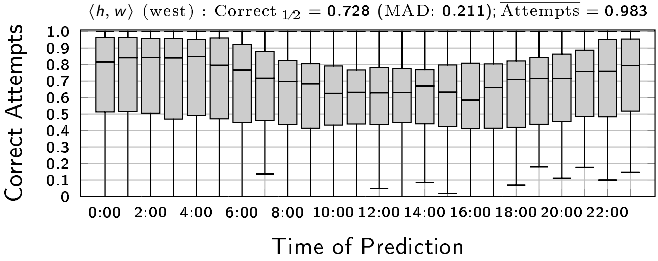
Time of Day and Day of Week



- Reduction in performance! (66.7% vs. 70.4% for $\langle h \rangle$)
- Low number of attempts (70.5%)
- Just considering long traces (¿ 16 weeks):
  - Score: 74.1%
  - Attempts: 95.8%

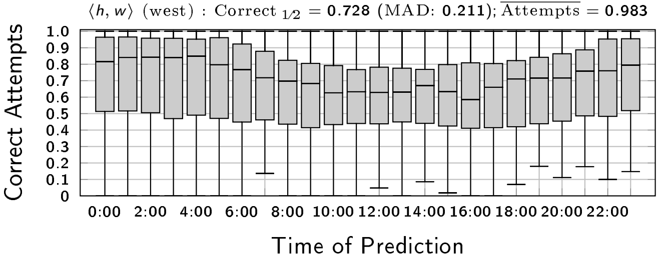
$\Longrightarrow$ data too spread out!

Time of Day and Work Day



- Just distinguish between workdays and days of rest
- Increase in performance (72.8% vs. 70.4% for $\langle h \rangle$)
- High portion of attempts

Time of Day and Work Day



- Just distinguish between workdays and days of rest
- Increase in performance (72.8% vs. 70.4% for $\langle h \rangle$)
- High portion of attempts
- Also tried country-specific days of rest, same performance
- Perhaps bias?
- Ideally learn workdays from data

Regimes

- Weak location-dependent predictor
- Regime classification
  0 Dominant tower over past 24 hours
  1 Current tower's primary regime ($argmax\, P(r|t)$)

Regime-based Predictor

Predictor	Correct Attempts	Attempts
$\langle h, w \rangle$	72.8%	98.3%
$\langle r, h \rangle$	77.6%	91.7%
$\langle r, h, d \rangle$	76.7%	55.8%
$\langle r, h, w \rangle$	81.1%	86.9%

- Significant improvement in correct attempts
- Trade-off: Fewer attempts

Current Tower Aggregate-based Predictor

- Strong location-dependent predictor
- Note:
  - $\langle c \rangle$ is the current tower predictor (our baseline)
  - Need a temporal reference, e.g., the prediction offset (Δ)

Evaluation

Predictor	Correct Attempts	Attempts
$\langle r, h, w \rangle$	81.1%	86.9%
$\langle h, \Delta, c \rangle$	81.3%	73.8%
$\langle h, d, \Delta, c \rangle$	81.4%	38.2%
$\langle h, w, \Delta, c \rangle$	83.0%	66.1%

- Slight improvement in correct attempts
- Tradeoff: significant decrease in portion of attempts

Aging

- Idea: Adapt to changes in behavior
- Approaches:
  1. Keep last x days of data
  2. Keep last x days per *primary* condition
     - Idea: behavior at secondary regimes likely stable
     - Example: parents' or remote office visited every few months

Aging Evaluation

- Used approach 2
- Tried different amounts of aging
  - 1, 2, 3, 4, 6 weeks

Aging Evaluation

- Used approach 2
- Tried different amounts of aging
  - 1, 2, 3, 4, 6 weeks
- Results:
  - Aging $\langle h \rangle$: 1 week improved precision from 70.4% to 75%
  - Aging r-based predictors: status quo for 3–4 weeks
  - Aging c-based predictors: status quo for 3–4 weeks
- Conclusion:
  - Conditioning on r or c already captures dynamic behavior
- Recommendation:
  - 3–4 weeks of aging to reduce amount of data stored

Combining Predictors

- If a predictor doesn't have enough data, fallback to another
- Prefer high precision predictors
- Results:
  - $> 99\%$ attempts
  - $\langle r, h, w\rangle$,$\langle r, h\rangle$,$\langle r\rangle$: 80% correct
  - $\langle h, w, c, \Delta\rangle$,$\langle h, c, \Delta\rangle$,$\langle r, h, w\rangle$,$\langle r, h\rangle$,$\langle r\rangle$: 79% correct

Combining Predictors

- If a predictor doesn't have enough data, fallback to another
- Prefer high precision predictors
- Results:
  - $> 99\%$ attempts
  - $\langle r, h, w\rangle$,$\langle r, h\rangle$,$\langle r\rangle$: 80% correct
  - $\langle h, w, c, \Delta\rangle$,$\langle h, c, \Delta\rangle$,$\langle r, h, w\rangle$,$\langle r, h\rangle$,$\langle r\rangle$: 79% correct
- Per prediction offset-based predictors:
  - $0.5h$: Current tower aggregate baseline (93%)
  - $1.5h - 2.5h$: Current tower-aggregate based (80% – 85%)
  - $> 2.5h$: Regime-based (78% – 80%)
  - Result ($24h$): 82%

Project

1. Design high-level location prediction API
2. Implement baseline predictor
3. Implement current tower-aggregate predictor $(1 - 2.5h)$
4. Use prediction to:
   - Enable/disable home heating (project!)
   - Prefetch weather data (going to Bern or skiing?)
   - Disable GSM/WLAN ("user rarely uses it on the train")
   - Make suggestions for when to schedule appointments
   - ...

API Design Hints

- Start by defining “Location” abstraction
- Input for location prediction is time in future
- Plan for “no prediction” as possible answer!
- Output may include level of uncertanity or multi-set with probabilities

Your final design will likely depend on your method to record locations and your application!

Exam reminder

1. Submit your code ≈ 1 week before the oral exams
2. In that case, you will be asked questions about the project:
   - What your project does (explain to co-examiners!)
   - Examination on how it works in depth
   - Critical discussion based on my code review prior to the exam
3. Otherwise, any theory that was taught is fair game

Acknowlegements

This presentation used material from: