Real-time Length of Stay Prediction Using Passive WiFi Sensing

Truc Viet Le[1] Baoyang Song [2] Laura Wynter[3]

[1]School of Information Systems
Singapore Management University, Singapore

[2]Computer Science Department
École Polytechnique, France

[3]IBM Research
Singapore

May 22, 2017

Outline

Outline

Motivation

- Mobile devices are pervasive links between networks & individuals.
- Widespread use of affordable Wi-Fi in many retail settings for customer's convenience (and, more importantly, behavior tracking).
- Human behavior is not random, predictable through pattern recognition.
- Build a system for passive data collection and online learning in real-time

Why length of stay?

Length of stay

Length of stay (LOS), or dwell time, is the duration of time a device (individual) stays active at a specific locality.

- Length of stay (LOS) provides precious information for stores (*e.g.* adjusting service stuffs).
- Previous work shows that LOS is predictable.

Why WiFi?

- WiFi access points (AP) are becoming omnipresent, most of mobile stations today are equipped with WiFi functionality;
- Mobile stations scan periodically the WiFi bands by broadcasting on **all** available channels *probe requests*;
- Association process:

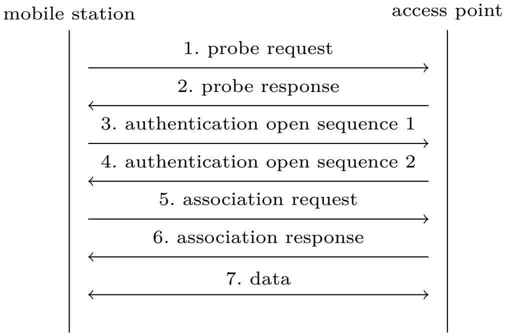


Figure: Association process of a mobile station with an AP

Why WiFi (cont'd)?

Unparalleled advantages of probe request:

- Probe requests are not bound to any specific AP (figure 1).
  - Even if no APs exist at all, probe requests are still sent and can still be recorded/sniffed;
- Probe requests are universally accessible:
  - administrators of APs: querying the system log;
  - anyone: sniff with `tcpdump` or `Wireshark`
- Accessing probe requests is device-free and non-intrusive:

Related Work

Manweiler, J., Santhapuri, N., Choudhury, R. R., & Nelakuditi, S. (2013, April). Predicting length of stay at wifi hotspots. In INFOCOM, 2013 Proceedings IEEE (pp. 3102-3110). IEEE.

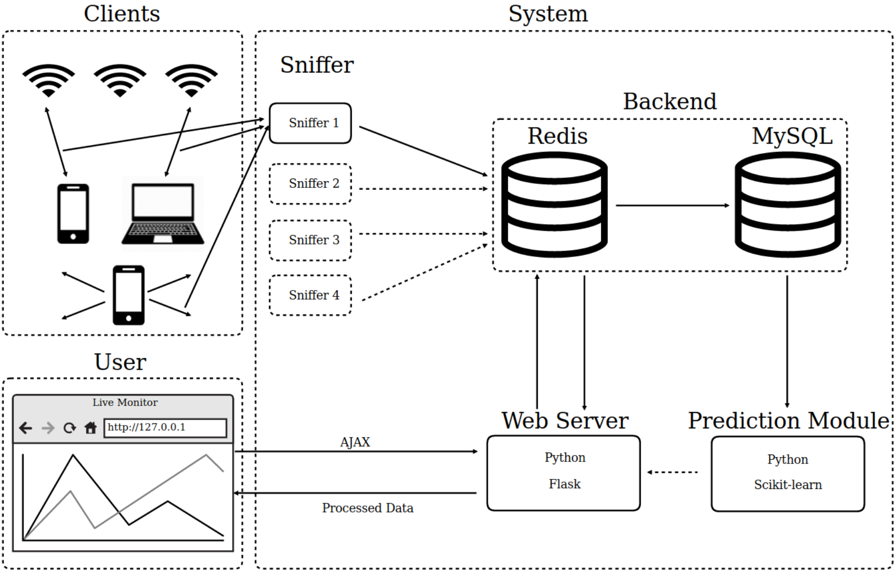
- Real-time classification of dwell time (LOS) into 5 categories using SVM
- Advantages:
  - Live prediction
  - High accuracy
- Disadvantages:
  - Software needs to be installed on mobile devices → intrusive!
  - Many features, e.g., transmission rate, are not available for unassociated devices

Our work

- Assumption: LOS can be put into categories
- Goal: at each time t, predict the true LOS label of an active device in *real-time* and as soon as possible based on data frames and features continuously received from the device till t
- Suppose (discrete ordinal labels), *e.g.*, passer-by, short-stay, medium-stay, long-stay, *etc.* specific to the use case.
- Advantages:
  - Low cost
  - Passive
  - Real time

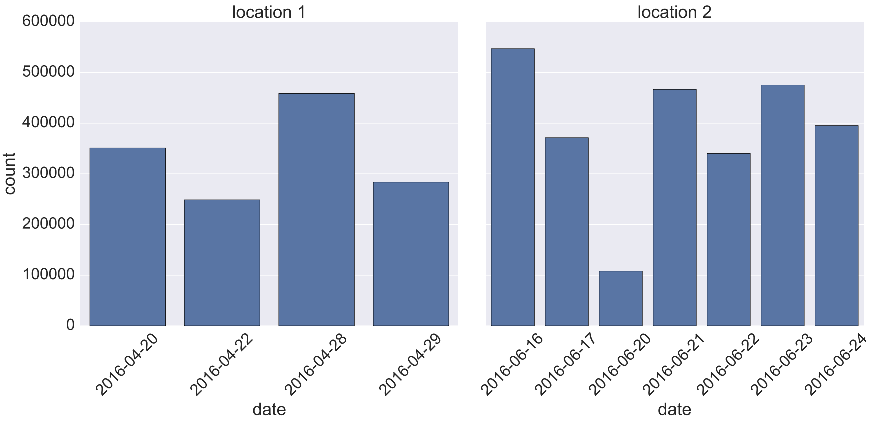
Outline

System Design



Outline

Data acquisition



Data fields

Field	Description
`timestamp`	Date and time of the receipt of the frame
`MAC_addr`	Unique MAC address of the mobile device
`power_mgt`	Power management state (awake/sleep) of the device
`type`	Either 1 (management), 2 (control) or 3 (data)
`subtype`	Additional discrimination between frames
`seq_ctrl`	Counter that identifies message order and eliminates duplicates
`RSSI`	Received Signal Strength Indicator indicating the signal strength
`channel`	Indicates the channel (e.g., ranging 1–14 for 2.4 GHz band)
`data_rate`	Speed of data transmission
`SSID`	Identifier of the AP

Table: The retained data fields of each received data frame.

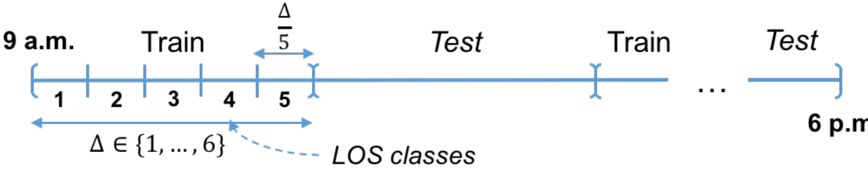
Features

Feature	Description
`begin_hours`	Integer hour of the day when the device was first detected
RSSI	Cumulative mean, stdev and histogram of RSSI
Data rate	Cumulative mean, stdev and histogram of data rate
`time_spent`	Current LOS (so far) of this device (in minutes)
`num_device`	Current number of *other* devices detected at the location
`rssi_grad`	Instantaneous gradient of RSSI
`data_rate_grad`	Instantaneous gradient of data rate

Table: Feature vector $\mathbf{x}(t)$. Calculated every 15 seconds.

Training

- For each day, divide a 9-hour timeline into $\frac{9}{\Delta}$ intervals
- The first interval is used for training, second for testing, *etc.* The last interval is *always* for testing.
- Each interval is divided equally into 5 sub-intervals.
- Only stays that starts and ends in the sub-intervals are retained.
- The label is the number of sub-intervals covered.
- Training using classical SVM and online SVM (detailed later).



Interlude - SVM

- (Soft) maximal margin:

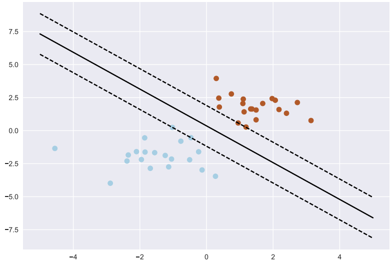


Figure: Linear SVM - Separable case

- Hinge-loss: the SVM classifier $x \mapsto \text{sign}(\omega x + b)$ minimizes the (regularized) *hinge-loss*

$$\frac{1}{n}\sum_{i=1}^{n} \max(0, y_i(\omega x_i + b)) + \lambda\|\omega\|^2 \qquad (1)$$

Interlude - Stochastic gradient descent

- Function to minimize: $Q(\omega) = \frac{1}{n}\sum_{i=1}^{n} Q_i(\omega)$.
- Gradient descent:

$$\omega := \omega - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla_i Q_i(\omega)$$

- Stochastic gradient descent:

$$\omega := \omega - \eta \nabla_i Q_i(\omega)$$

Testing and evaluation

For each present mobile device i

- $\hat{C}_i(t)^{\text{pred}}$: the prediction of its *final* LOS class.
- $C_i(t)^{\text{pred}} = \max\{\hat{C}_i(t)^{\text{pred}}, C_i(t)^{\text{current}}\}$.

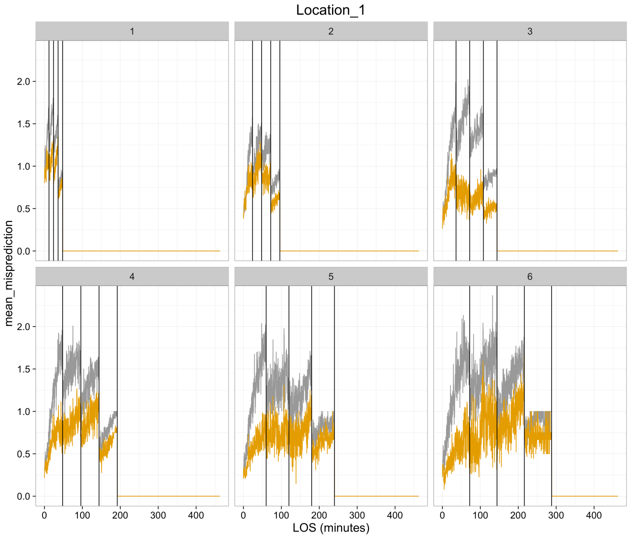
Mean mis-prediction

Given device i, suppose that its final true class of LOS is C_i^{true}. At any time t, our adjusted prediction of i's true class is $C_i(t)^{pred}$.
The mean mis-prediction error at time t is defined as

$$\texttt{mean\_misprediction}(t) = \frac{\sum_{i=1}^{N} |D_i(t)|}{N(t)}. \tag{2}$$

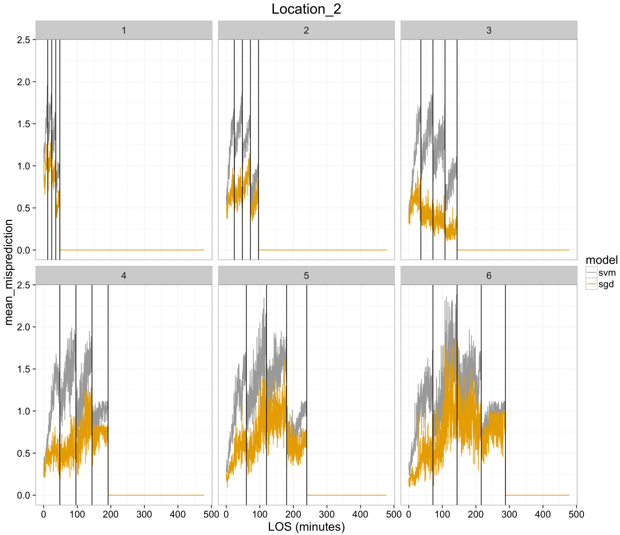
where $D_i(t) = |C_i^{true} - C_i(t)^{pred}|$ is the instantaneous misclassification error for i and $N(t)$ the number of active device at t.

Evaluation

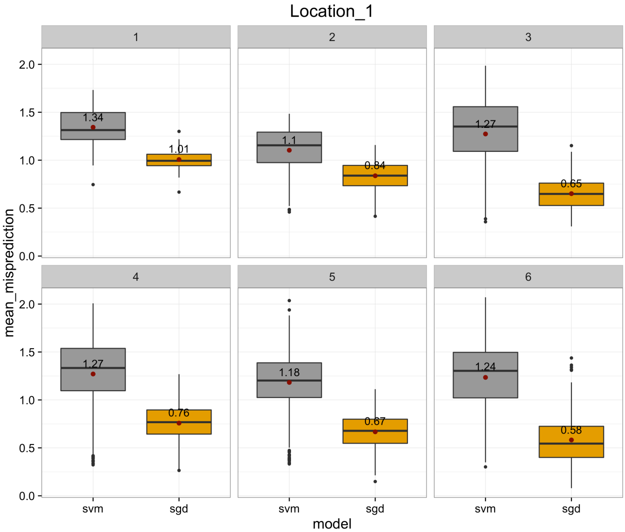


Location_1
1
2
3
4
5
6
mean_misprediction
0.0
0.5
1.0
1.5
2.0
0
100
200
300
400
LOS (minutes)

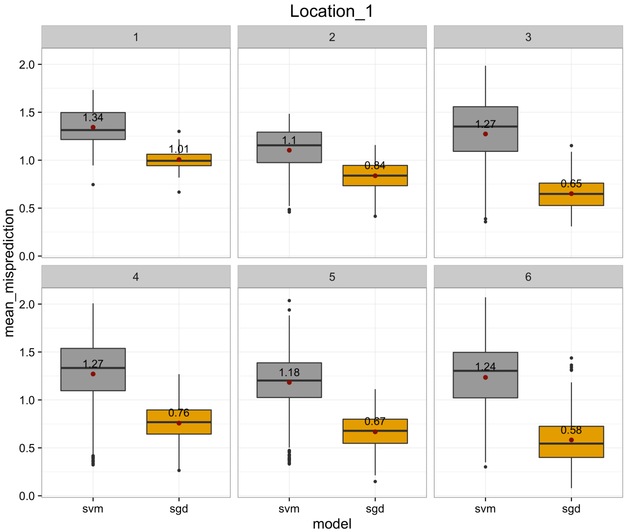
Evaluation (cont'd)



Evaluation (cont'd)



Evaluation (cont'd)



Conclusion

- Design and test a passive Wi-Fi sensing system to monitor and predict in real-time information about people's movements.
- Many interesting applications in retail settings.
- For length of stay, the system automatically generates a number of features.
- The features are used to train a linear SVM classifier as well as an online SGD update mechanism to take into account the dynamics of the environment and adaptive changes to the classification parameters.
- Future work: explore other applications that can be derived from this type of system.

Questions?

Thank you for your attention!