Lifespace Tracking and Activity Monitoring on Mobile Phones

Ian Dewancker

idewanck@yelp.com

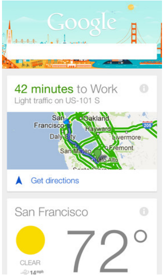
June 23, 2014

Motivation

- Daily **patterns of behavior** are a rich source of information and play an important role in a persons quality of life (mobility, socializing, eating, toileting)
- **Lifespace** is a measure of the frequency, geographic extent and independence of an individuals travels
- While **difficult to measure** and record automatically, lifespace has been shown to correlate to important metrics relating to physical performance, nutritional risk, and community engagement

Motivation

Quantified Self
self knowledge through numbers

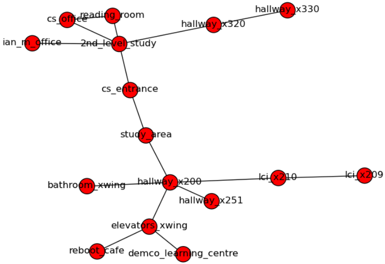


Goals

- Highly **personalized, detailed mobility traces** are not only of interest to researchers, but also individuals motivated to learn more about their own trends and behaviours
- System to track lifespace and monitor mobility of both **ambulating and wheelchair users** with standard **mobile phones**
- Investigation of models and methods to summarize **indoor** and **outdoor** movement as well as **activity** levels. Focus on techniques that perform well on real sensors and in real-world environments

WiFi Indoor Localization

Two main types of maps : metric and **topological**
Pursue topological map localization as it requires only minimal human effort for mapping while preserving useful **semantic information**



Key elements for prob. localization

$$P(x_i \mid x_{i-1}, u) = \text{motion model}$$
$$P(z \mid x_i) = \textbf{observation model}$$

Observation Model

- Vision sensors are low cost and have been well studied in this context, but mounting cameras can be awkward and impose on privacy
- In most situations, GPS will not function indoors
- WiFi has become highly integrated into indoor infrastructure. Signal measurements can be made by any mobile device equipped with WiFi modem including mobile phones

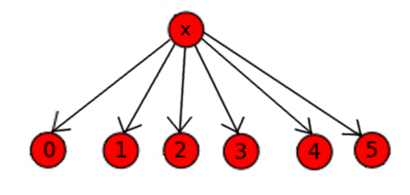
SSID (Name)	BSSID (MAC Address)	RSSI (Signal Strength)
ubc_secure	00:12:44:b0:4f:bb	-73
ubc_secure	00:14:f1:ac:74:23	-36
ubc_visitor	00:1c:0e:42:47:45	-84
bbox2-3964	00:10:7f:13:55:75	-66

$$z = [RSSI_0, RSSI_1, \cdots, RSSI_N], \;\; x_i = room\ i$$

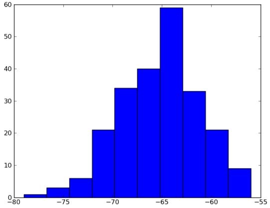
- The signal strength for an absent access point is cast to -100

Naive Bayes

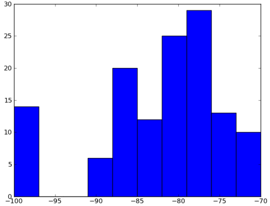
Naive Bayes is a generative model that assumes conditional independence of feature variables



$$P(z[0], z[1], ..., z[N] \mid x) = \prod_{i=0}^{N} P(z[i] \mid x)$$

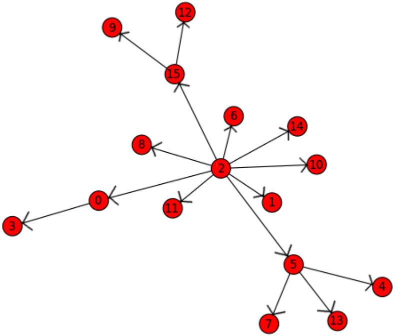


$$P(z[i] \mid x) = \frac{1}{\sqrt{2\pi\sigma_{z[i]}^2}} \exp\left(-\frac{(z[i] - \mu_{z[i]})^2}{2\pi\sigma_{z[i]}^2}\right)$$



Chow-Liu Trees

Chow-Liu algorithm approximates a probability distribution by the closest tree-structured Bayesian network



$$P(z[0], z[1], ..., z[N] \mid x) = \prod_{i=0}^{N} P(z[i] \mid z[parent(i)], x)$$

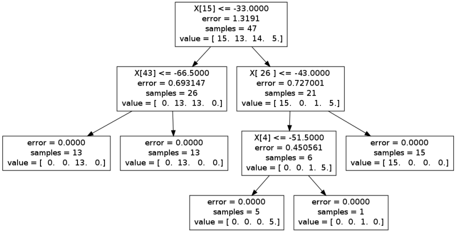
The topology is determined by the maximum-weight spanning tree of the complete graph of the N random variables. The weights are calculated as the mutual information

$$I(z[i], z[j]) = \sum_{z[i]\in\Omega,\ z[j]\in\Omega} p(z[i], z[j]) \log \frac{p(z[i], z[j])}{p(z[i])p(z[j])}$$

This structure saw interest again in robotics research in for visual appearance models

Random Forests

Random forests are ensemble learners that use a collection of decision trees to classify test input. Each tree in the forest is learned on a random subset of the training data



$$P(x = r \mid z) = \frac{|\text{Trees voting for } r|}{\text{Total Trees}}$$

splitting candidate $\phi = (\theta, \tau)$

$$Q = \{(z, x)\}$$
$$Q_l(\phi) = \{\ (z, x) \mid z[\theta] < \tau\ \}$$
$$Q_r(\phi) = Q \setminus Q_l(\phi)$$

$$\phi^* = \arg\max_{\phi} G(\phi)$$

$$G(\phi) = H(Q) - \sum_{s \in \{l, r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi))$$

$$H(Q) = - \sum_{r \in \{rooms\}} p(x = r) \log p(x = r)$$

K-Nearest Neighbours

The k-nearest neighbour classifier is a non-parametric classifier that compares test input to training examples and finds the k most similar. It then returns the majority class label of those k training examples

$$NN(z) = \mathrm{k} \underset{(z_i, x_i) \in O}{\arg\min} ||z_i - z||_2$$

$$P(x = r \mid z) = \frac{|\{(z, x) \in NN(z) \mid x = r\ \}\ |}{\mathrm{k}}$$

It can be converted into a probabilistic model by returning the percentage of the k neighbours voting for each room

Novelty Detection

Localizing to an indoor room is important, however so is knowing when to report that a user is probably **not** located in one of the trained rooms

$$f(z) = \begin{cases} 1, & \text{inlier, observation from known room} \\ -1, & \text{outlier, novel observation} \end{cases}$$

Investigation of useful distance metrics for comparing test observation vectors to the training data

$$z_{neg} = [\ -100,\ -100,\ \cdots,\ -100\ ]$$

$$f(z) = \begin{cases} 1, & \text{if} \quad dist(z,\ \{z_i\}) < \tau \\ -1, & \text{otherwise} \end{cases}$$

$$\tau = \alpha \cdot (\ dist(z_{neg},\ \{z_i\}))$$

Novelty Detection

Nearest Neighbour Threshold

$$f(z) = \begin{cases} 1, & \text{if } \min_{i \in O} ||z_i - z||_2 < \tau \\ -1, & \text{otherwise} \end{cases}$$

$$\tau = \alpha \cdot (\min_{i \in O} ||z_i - z_{neg}||_2)$$

Nearest Centroid Thresholds

$$\mu_i = \frac{1}{|R_l|} \sum_{i \in R_l} z_i$$

$$f(z) = \begin{cases} 1, & \text{if } ||\mu_i - z||_2 < \tau_i \\ -1, & \text{otherwise} \end{cases}$$

$$\tau_i = \alpha \cdot ||\mu_i - z_{neg}||_2$$

$$i = \arg\min_{i \in U} ||\mu_i - z||_2$$

$$d(z,\ \mu_i) = \sqrt{(z - \mu_i)^T \Sigma_i^{-1} (z - \mu_i)}$$

$$f(z) = \begin{cases} 1, & \text{if } d(z,\ \mu_i) < \tau_i \\ -1, & \text{otherwise} \end{cases}$$

$$\tau_i = \alpha \cdot d(z_{neg},\ \mu_i)$$

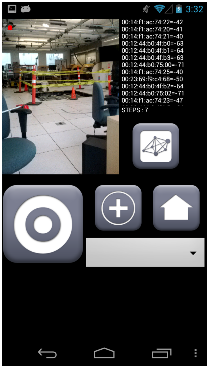
$$i = \arg\min_{i \in U}\ d(z,\ \mu_i)$$

Observation Model Thresholds

$$f(z) = \begin{cases} 1, & \text{if } -\log(\ P(z \mid \{z_i\})\ ) < \tau \\ -1, & \text{otherwise} \end{cases}$$

$$\tau = -\alpha \cdot \log(\ P(z_{neg} \mid \{z_i\})\ )$$

Dataset	Building	Multi-floor	Rooms	WiFi APs	Readings
(Al)	house	yes	6	15	2100
(Ar)	aptm	yes	5	42	2094
(Fe)	house	yes	4	7	1269
(UBC)	multi-bldg	yes	4	189	3138
(Va)	house	no	5	20	1140



WiFi environments are not static and can easily change over time. Access points may be moved or shut off after the training data has been collected. Created five simulated datasets where each observation vector has 20% of its access point readings set to -100

Evaluation

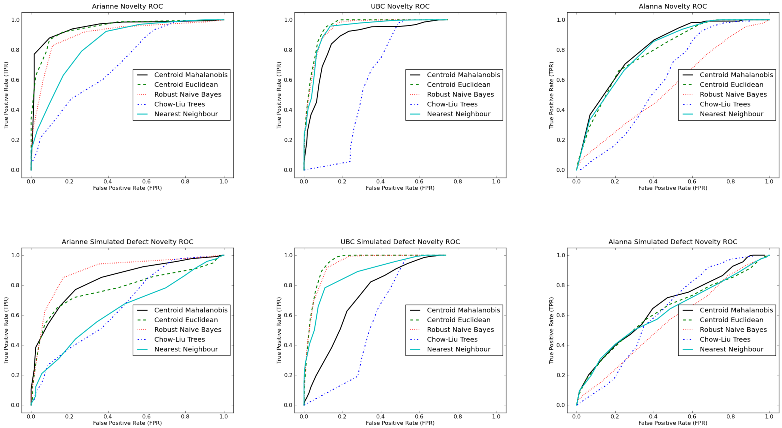
Random forests have strong performance on both the normal dataset and the simulated defect dataset

Classifier	**Avg.**	**Al**	**Ar**	**Fe**	**UBC**	**Va**	**Al**$_S$	**Ar**$_S$	**Fe**$_S$	**UBC**$_S$	**Va**$_S$
Random Forest	0.916	0.939	0.969	0.898	0.984	0.955	0.863	0.896	0.76	0.98	0.917
Robust NB	0.890	0.789	0.936	0.843	0.992	0.940	0.759	0.922	0.817	0.987	0.910
Chow-Liu	0.870	0.892	0.888	0.897	0.985	0.960	0.713	0.774	0.747	0.981	0.854
K-NN	0.826	0.926	0.796	0.96	0.982	0.955	0.727	0.46	0.757	0.953	0.744
Naive Bayes	0.193	0.189	0.116	0.313	0.216	0.120	0.189	0.116	0.313	0.237	0.120

Parameter study for k-nearest neighbour and random forests
Random forest (number of trees, maximum features, maximum depth)

Evaluation

ROC curves are a useful visualization for information retrieval systems. They are 2D projections of a systems performance in the space of true positive rate and false positive rates



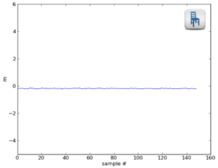
Actvity Recognition

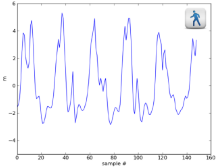
Context awareness is an ever improving feature of assistive technologies and ubiquitous computing. Inference of user activities and environmental interactions remains an important problem if only to serve higher level goals of a particular system or application

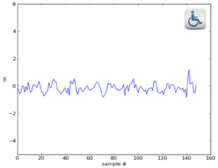
Actigraphy (non-invasive activity monitoring) has become a common theme in mobile health applications

Most of these systems have focused on ambulating users only, wheelchair users would certainly also benefit from similar summaries.

Accelerometer Features







An accelerometer is a device capable of measuring acceleration forces

Means of controlling the user interface, sensing display orientation or detecting falls for hard drive protection.

$$m = \sqrt{X^2 + Y^2 + Z^2} - g$$
$$g = 9.81\ m/s^2$$

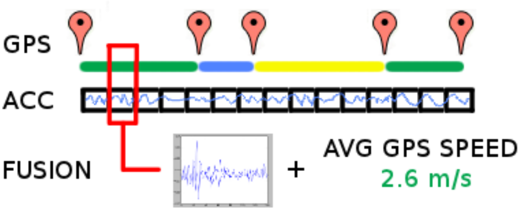
Poll at 20Hz for 3 second windows

$$f_0 = \frac{1}{N-1} \sum_{n=0}^{N-1} (m_n - \mu_m)^2$$

$$X_k = \sum_{n=0}^{N-1} m_n \cdot e^{-i2\pi kn/N}$$
$$|X_k| = \sqrt{X_k.Re^2 + X_k.Im^2}$$

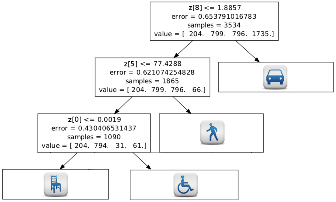
Sensor Fusion

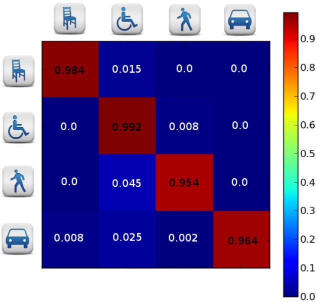
The accelerometry features seemed suitable for differentiating between the first three motion classes, however the classifier required a way to distinguish vehicle motion from the other profiles.



Final feature vector produced for each signal window by combinining accelerometer features with the average GPS speed recorded during the same time interval

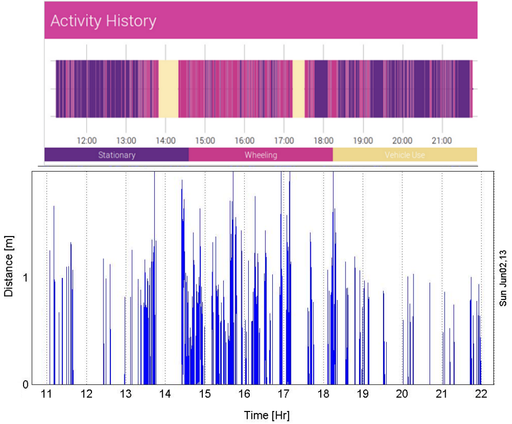
Evaluation


z[8] <= 1.8857
error = 0.653791016783
samples = 3534
value = [204. 799. 796. 1735.]
z[5] <= 77.4288
error = 0.621074254828
samples = 1865
value = [204. 799. 796. 66.]
z[0] <= 0.0019
error = 0.430406531437
samples = 1090
value = [204. 794. 31. 61.]


0.984
0.015
0.0
0.0
0.0
0.992
0.008
0.0
0.0
0.045
0.954
0.0
0.008
0.025
0.002
0.964
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

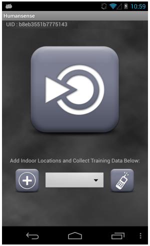
Evaluation

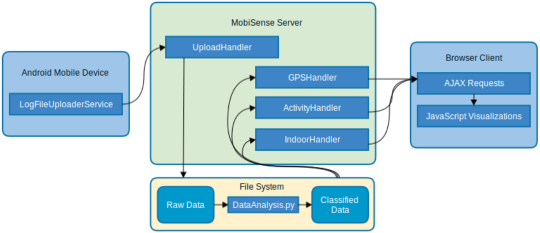
To further validate our classifier, we compared our results with a summary produced by another wheelchair activity measurement system. This system used an accelerometer mounted to a wheel.



MobiSense System

Data collection on the phone with the ability to upload data to a web service for summarizing and visualizing reports for users.
The centralized processing and storage of the data streams is done on a single Amazon EC2 instance running Ubuntu. The Tornado web server is used to handle data uploads and requests for lifespace summaries





App is a modifcation of HumanSense project. ML and Data pipeline in python, mostly sci-kit learn

Sensor Streams

- The accelerometer sensor stream creates by far the most data, recorded constantly at 20Hz
- The WiFi modem is polled every 10 seconds
- The GPS sensor is polled once a minute
- 18 hour recording period with full accelerometer, WiFi and GPS logging 15 MB of compressed sensor data collected
- Files are only uploaded to the web service when WiFi connection present
- Summarization format per day only 100 KB
- With little screen use and cell antenna disabled, MobiSense was able to run on a single battery charge on a Nexus 4 phone for 22 hours
- With antenna enabled, close to 14 hours.

Demo!

Demo

Questions? Comments?