

There are a lot of papers to analyze users by their web sites visits while more than half of digital traffic online now comes from mobile devices and through mobile apps (based on [comScore report](#)).

The goal is to predict the demographic and life style profiles of users based on their previous locations and past behavior at a certain hour of a day.

In case if we have additional context (like any truth set, or application used, user's tweets, etc.) we could tune the model.

As a first step, let's imagine we have a data set that contains user id, timestamp and location (latitude/longitude pair).

1. Detect "frequent spots":

- cluster data using KMeans algorithm (represent users trajectories as fixed-length vectors of coordinates and then compare such vectors by means of Euclidean distance) or (as another approach) using Hidden Markov models
- detect multiple interleaved periods using Fourier Transform and autocorrelation

record	user	timestamp	latitude	longitude
r1	u1	2016-05-09 09:00:00	37.786137	-122.409143
r2	u1	2016-05-09 09:30:00	37.785737	-122.410922
r3	u1	2016-05-09 13:00:00	37.787011	-122.406039
r4	u2	2016-03-26 12:45:00	37.786200	-122.409600
r5	u3	2016-03-01 17:15:00	37.785934	-122.411144

2. Label the spots based on timestamps and external context available (like type of location from GooglePlacesAPI): "Office", "Home", "Shopping Mall" etc.

record	annotation
r1	San Francisco, Starbucks, coffeehouse, working hours
r2	San Francisco, Hilton, hotel, working hours
r3	San Francisco, Macy's, department store, lunch time
r4	San Francisco, road
r5	San Francisco, FedEx

3. Predict user profiles using decision trees with generative grammar component (associative rules, NLP are applicable).

*High-level examples:*

Frequent visits to "Victoria Secret" => Gender: female

Frequent visits to Chinese, Japanese restaurants => Food interest: Asian

*Notations:*

Let's use [a formal grammar](#) as a tuple of

- a finite set  $V_T$  (terminal or main vocabular),
- a finite set  $V_A$  (auxillary vocabular),
- an axiom  $S \in V_A$
- a finite set of production rules  $A \rightarrow \varphi$ , where  $A \in V_A$ ,  $\varphi \in \{V_A \cup V_T\}^*$

Let's represent a grammar as a graph, whose vertices correspond to the nonterminal symbols of the dictionary  $S = A_1, A_2, \dots, A_P$ , and to the symbols appearing in the terminal rules  $a_1, a_2, \dots, a_q$ .

We connect  $A_i$  and  $A_j$  (correspondingly in  $a_k$ ) if there is a rule in the grammar in the form of:

$$A_i \rightarrow \varphi A_j \psi \text{ (correspondingly } A_i \rightarrow a_k \text{ )}.$$

If the graph is not connected, then any of its connected components that doesn't contain any axiom will not have any effect on the generative grammar, so the correspondent rules and symbols can be removed.

Thus, we can always assume that the above grammar is connected, i.e. that its graph is connected.

*Example:*

Suppose we have keywords " $A_1 A_2 A_3 A_4 A_5$ " and the following rules:

Conditional rules	Decision rules
$A_1 \mid (s_1 = "+" \wedge s_2 = "-") \rightarrow \varphi_{11}$ $A_1 \mid (s_1 = "+" \wedge s_2 = "+") \rightarrow \varphi_{12}$	$A_1 \mid (\varphi_1 = \varphi_{11}) \Rightarrow (s_3 := "+")$ $A_1 \mid (\varphi_1 = \varphi_{12}) \Rightarrow (s_3 := "-")$
$A_2 \mid (s_3 = "+" \wedge s_4 = "+") \rightarrow \varphi_{21}$	$A_2 \mid \varphi_2 = \varphi_{21} \Rightarrow (s_5 = "+")$
$A_3 \mid (s_4 = "+") \rightarrow \varphi_{31}$ $A_3 \mid (s_4 = "-") \rightarrow \varphi_{32}$	$A_3 \mid (\varphi_3 = \varphi_{31}) \Rightarrow (s_2 := "-")$
$A_4 \mid (s_6 = "+") \rightarrow \varphi_{41}$ $A_4 \mid (s_6 = "-") \rightarrow \varphi_{42}$	$A_4 \mid (\varphi_4 = \varphi_{41}) \Rightarrow (s_1 := "-")$ $A_4 \mid (\varphi_4 = \varphi_{42}) \Rightarrow (s_1 := "+" \wedge s_4 := "+")$
$A_5 \mid (s_1 = "+") \rightarrow \varphi_{51}$ $A_5 \mid (s_1 = "-") \rightarrow \varphi_{52}$	$A_5 \mid (\varphi_5 = \varphi_{51}) \Rightarrow (s_6 := "-")$ $A_5 \mid (\varphi_5 = \varphi_{52}) \Rightarrow (s_6 := "+")$

Then the algorithm is as follows:

Setting						Profile $A_i$ , Rule type	Hypothesis
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	I	
.	.	.	.	.	.	-	
+	-	.	.	.	.	1, cond.	$H_1 : s_1="+" \wedge s_2="-"$
+	-	+	.	.	.	1, cond.	
+	-	+	+	.	.	2, cond.	$H_2 : s_4="+"$
+	-	+	+	+	.	2, cond.	
+	-	+	+	+	.	3, cond.	
+	-	+	+	+	.	3, cond.	Confirmation for $s_2="-"$ in $H_1$
+	-	+	+	+	+	4, cond.	$H_3 : s_6="+"$
-	-	+	+	+	+	4, cond.	Rejection for $s_1="+"$ in $H_1$
+	-	+	+	+	-	4, cond.	$H_3 : s_6="-"$
+	-	+	+	+	-	4, cond.	Confirmation for $s_1="+"$ in $H_1$ and $s_4="+"$ in $H_2$
+	-	+	+	+	-	5, cond.	
+	-	+	+	+	-	5, cond.	Confirmation for $s_6="-"$ in $H_3$

Therefore we obtain the following classification:

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$
$\varphi_{11}$	$\varphi_{21}$	$\varphi_{31}$	$\varphi_{42}$	$\varphi_{51}$

### Improvements and known issues:

- GPS accuracy: The United States government currently [claims](#) 4 meter RMS (**7.8 meter 95%** Confidence Interval) horizontal accuracy for civilian (SPS) GPS. Vertical accuracy is worse. So in step 2, we need to use not latitude/longitude pair, but a circle with radius at least 8 meters (we choose 10 meters).
- For demographic profiles some open data sets can be used as the truth sets like: [http://proximityone.com/location\\_based\\_demographics.htm](http://proximityone.com/location_based_demographics.htm)  
<http://www.census.gov/topics/income-poverty/income.html>
- To smooth our probabilities in case of high deviations it worst to add some weights to every profile. As a first approach, for this step we need to estimate the overall population in the area using [deep learning model](#).