Master Project in Sony : Mining smartphone data

Report 2 : 01/05/2015-16/06/2015

Khalil Hajji

EPFL supervisor : Patrick Thiran

Sony supervisor : Fabien Cardinaux

18 June 2015

REMINDER

We have a dataset of various types of users' traces collected from their smartphones during several months. Those traces include the GPS traces, the Wi-Fi names connected to the phone, the applications launched ect… This data is collected each time that one of the following events E occurs: [notification received, application launched, screen is on, screen is off, launcher is on, launcher is off]

The goal of the project is to develop a model that learns the behaviors and the habits of the users starting from this dataset. For example, we would be interested in knowing that each Saturday morning, the user plays football. We would also be interested in to knowing that each morning in the working days, the user reads some news in his smartphone.

Highlights

During the first months, we mainly focused on cleaning the dataset, reordering it, adding information to it, and exploiting the presence of multiple features to end up with more accurate and pertinent information.

In particular, we invested a lot of efforts in extracting consistent and accurate information about the users' locations by combining the GPS, the Wifi routers and the base stations features.

Planned work

After ending the transformation of our data, we planned to do the following:

1- Think on an intelligent way to represent the data as a matrix so that standard algorithms can be applied to it.
2- Start the first Data Analysis algorithms to have some baselines.

REPORT PLAN

In this report, first we present some results that show that the choices of extracting and combining features have a real impact in improving the quality of our data, and especially the location feature.

Second, we describe how we choose to represent our data as a matrix and show some properties of this matrix.

Then, we talk about the first algorithm (SVD) we launched on this dataset and present the first results.

As usual, we end the report by presenting the next things we are planning to do.

A lot of things have been tried, especially different ways to filter some features, different ways to represent the data as matrix and different ways to run the SVD algorithm. In order to keep this report simple and short, we present only on the important things that worked the best.

OVERVIEW

1. IMPROVING DATA QUALITY

1.1- Extracting the Location of the user

In the last report, we explained how we combined the GPS feature, the Wi-Fi to information and the Cell Station data to end up with more accurate information about the location that users visit. We argued that we especially put a lot of efforts in extracting this feature because we believe it is important to describe the behavior of users.

Our work was driven by two goals that we want to achieve:

- *Location consistency:* Build clusters that cover a meaningful geographic zone depending on the place visited by the user.
- *Location accuracy:* Increase the amount of information available that we have about the location of the user

We show below that the results obtained achieve those two goals. Concerning the Location consistency, we observe for all the users that they have two very frequent locations $l1$ and $l2$. For all of the users, we see that during the working days, they are usually in $l2$ during the day (between 10A.M and 8P.M) and in $l1$ during the night. During the week ends, they are very often in $l1$ and rarely in $l2$. An example is the plot showed in Figure 1, where we plot $\Pr(loc = l_i | \, hour)$ for user 1 in the week days (Figure 1.a) and the weekends (Figure 1.b) (where $l1 = 401$ and $l2 = 1105$. Note that $l1$ is the blue curve in Figure 1.a and the red curve in Figure 1.b). Knowing that all the users we have are engineers working in Sony Japan, we plot their clusters $l2$ in a google map. The figure 2 shows that the $l2$ of the 6 users (red circles) covers the Osaki Sony Research Center (small blue circle), where all the users work.
Thus we can conclude that $l2$ is the work place and $l1$ is the home. Note that in Japan, people use to start working around 9A.M, 10 A.M and use to leave work late in the afternoon. This matches with our plots. The distributions of the users' locations in the week days and weekends coupled with the map give us confidence that our location extraction is consistent.

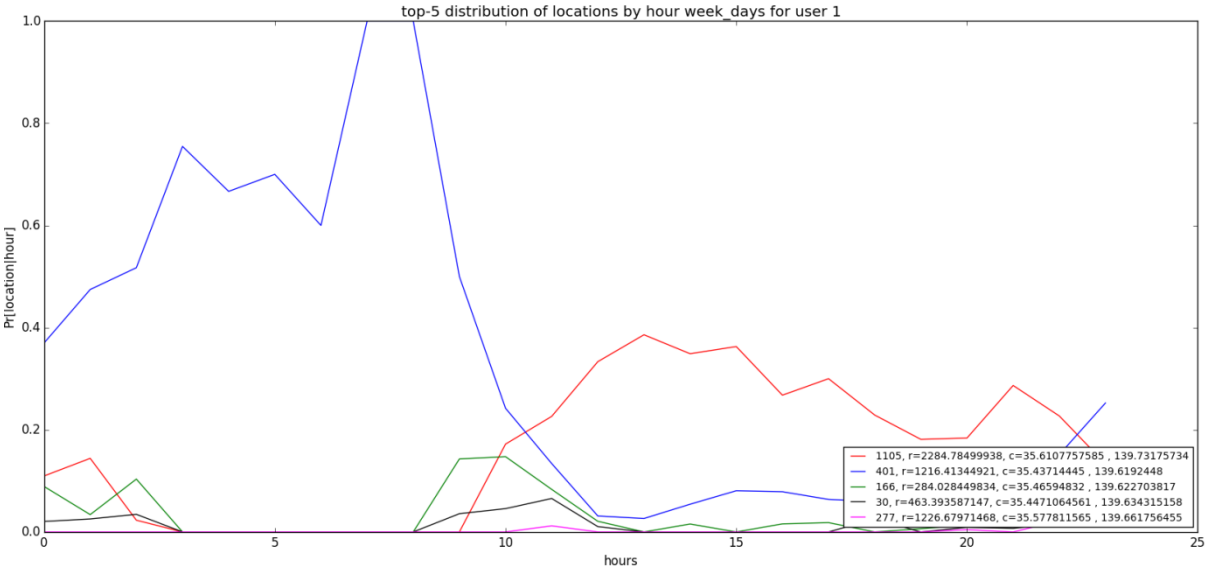

Figure 1(a): Location distribution for user 1 in the week days. In the legends, the first integer represents the id of the location, r the radius of the circle covered by the location (in meters) and c the latitude, longitude coordinates of the center of the circle covered by the location.

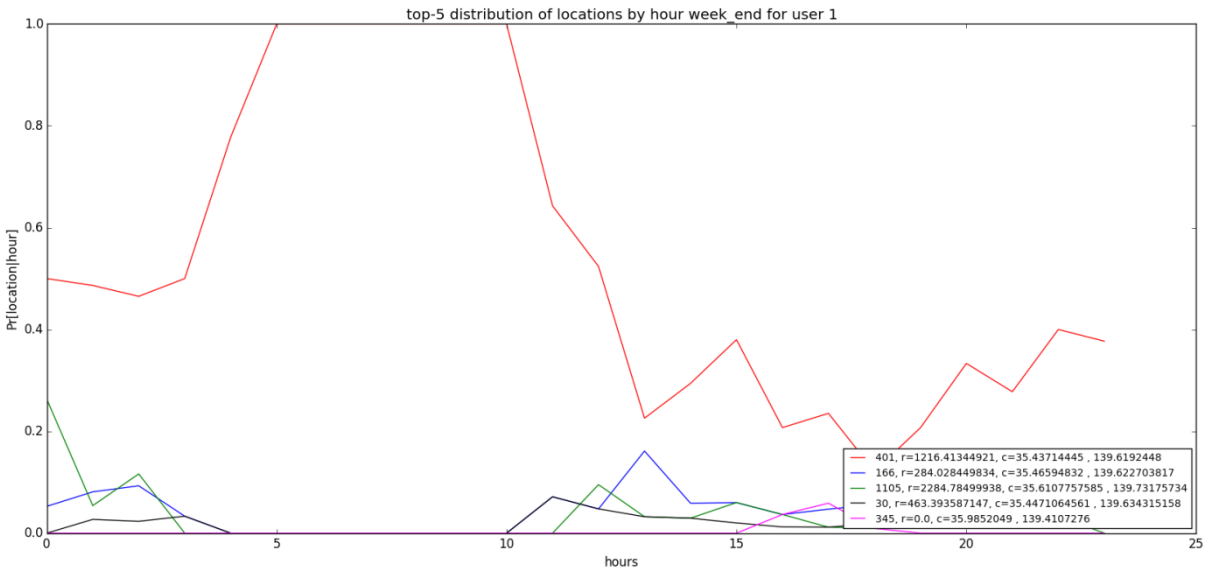

Figure 1(b): Location distribution for user 1 in the weekends. In the legends, the first integer represents the id of the location, r the radius of the circle covered by the location (in meters) and c the latitude, longitude coordinates of the center of the circle covered by the location.

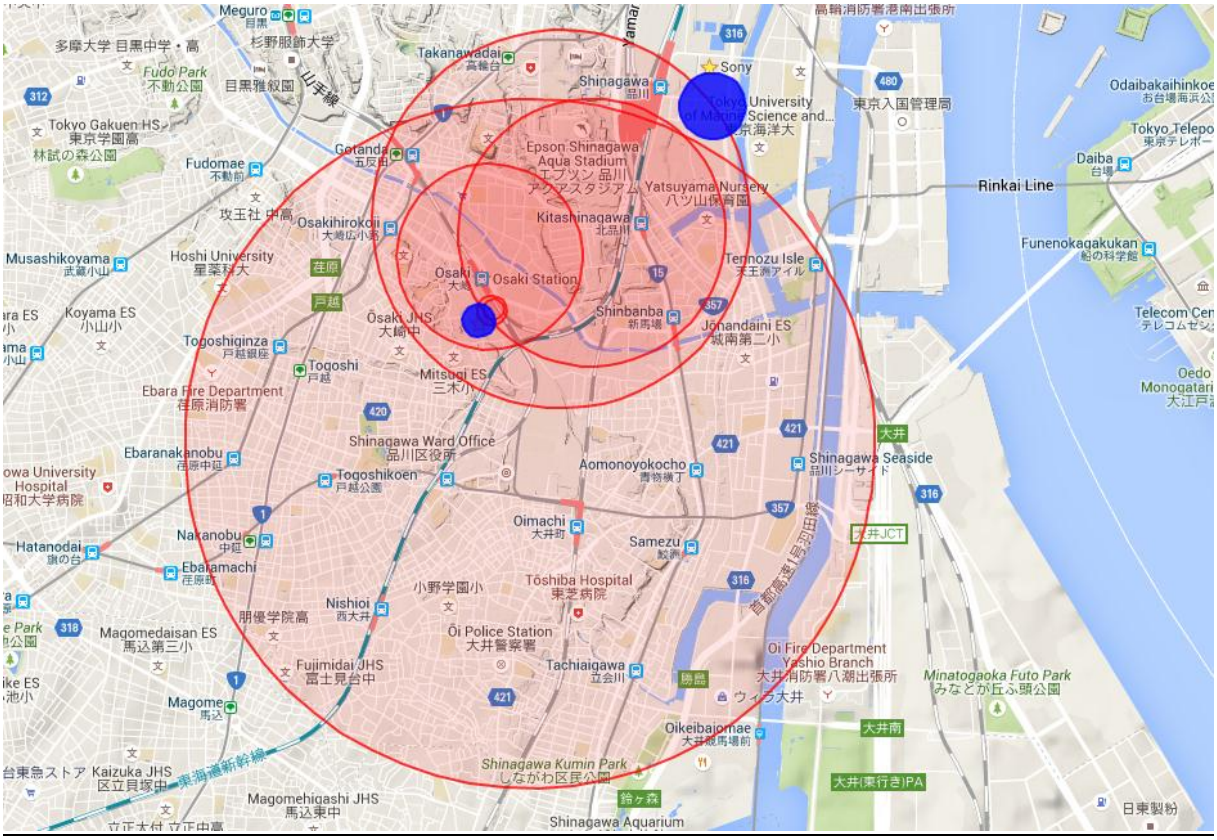

Figure 2: Map representing the Sony Tokyo research center (small blue circle), the Sony Headquarter (big blue circle) and the locations of the l2 clusters of the 6 users (red circles)

Concerning the Location accuracy, we computed for each user, the number of minutes where we have information about his location in the new version of the data (with the current location metric) and in the old one (the one taking into account only the GPS coordinates). Figure 3 shows those results and we can see using the current Location metric enable us to have from 2 (for user 6) up to 6 (for user 2) times more information about the location, which is a huge gain.

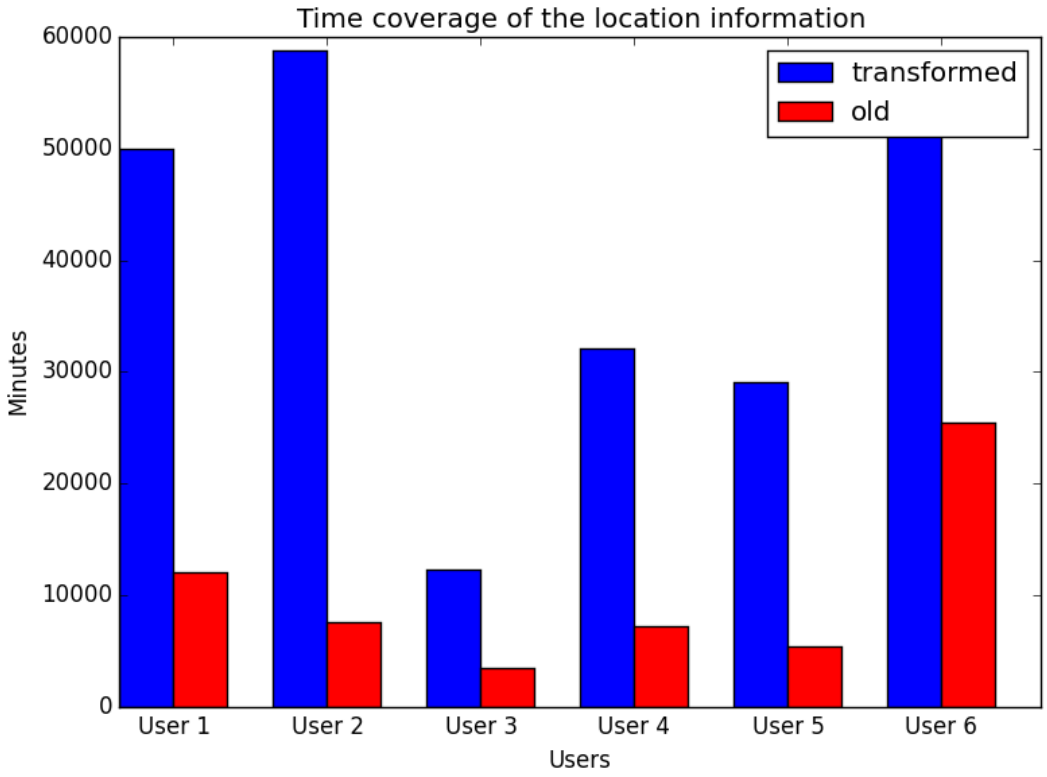

Figure 3: Number of minutes of available information about the location of the user in the old version of the data (red) and the one where the new location metric is used (blue)

1.2- Other features tuning

Concerning the other features (as Bluetooth, User Activity, Notifications), we also made some minor modifications to make them more accurate. For example, Concerning the Activity, our original dataset contains two sources that outputs the activity that the user is doing (Running, Walking, Driving,…). One source is the Android activity recognition and the other is the Sony activity recognition. After having exploring them, trying to combine them in different ways, we decided to drop the Sony activity feature (due to the huge noise present in it) and to rely only on the Android activity feature.

However, we do not detail in this report the operations that were done on those features to clean and to make them more accurate.

2. REPRESENTING THE DATA AS A MATRIX

2.1- Matrix Representation

We decide to represent our data as a binary matrix where the rows represent the features and the columns the different time frames (see Figure 4). To that end, we split our features into smaller features: a feature i represent a binary entry that contains 1 if it is present at the corresponding time j, 0 if it is absent or unknown. A feature could be Location_0, notification_7, Application_launch_5 or Bluetooth_paired_with_device_1, … We choose the time frames to represent 1 hour. We can see this representation as the document corpus representation where words are features and documents are a set of features that happened in one specified hour.

Representing the features as binary values enable us to interpret the results coming from matrix decomposition methods where a big absolute value of one feature projected on a given concept means that the presence of this feature is important in that concept.

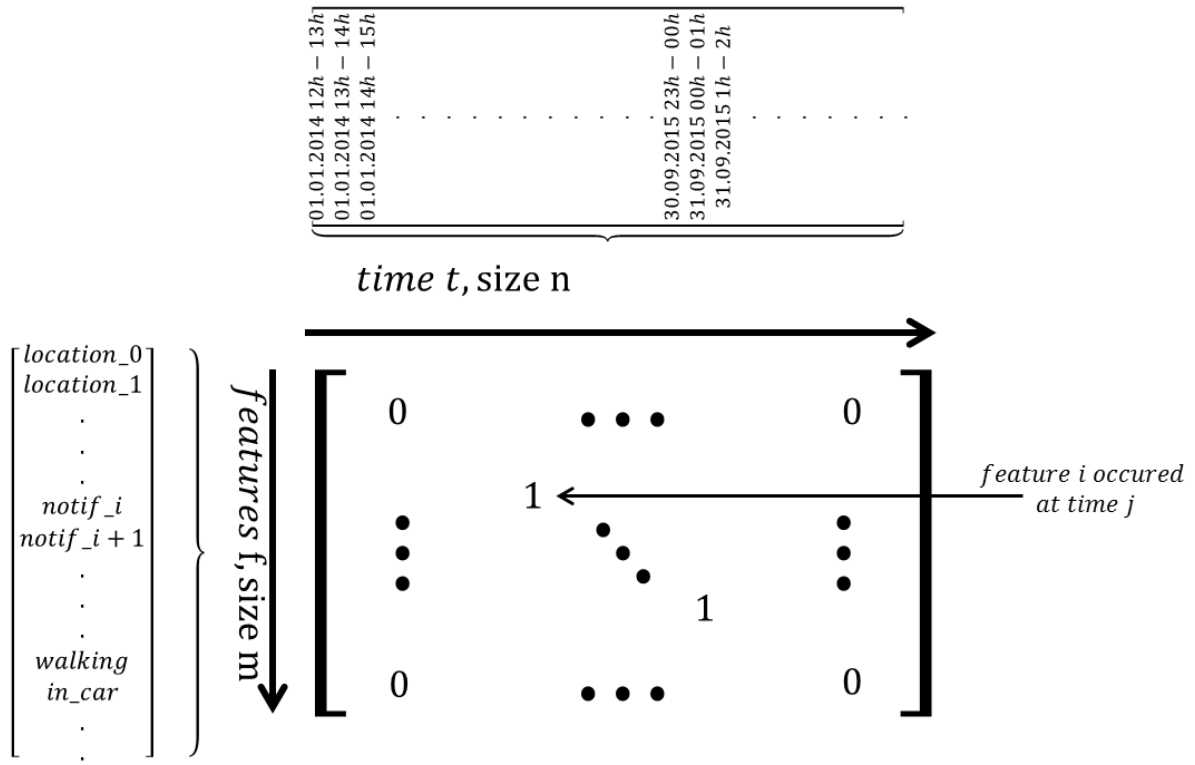

Figure 4: Matrix representation of the data

An exhaustive list of the binary features represented is presented in Appendix.A. Moreover, we reduce the size of the features by selecting only the $top - k$ locations and regrouping all the others in a feature "other_locations". We do the same for the notifications, the application launches, and the seen bleutooth devices. For now, we choose $k = 20$ for all the features enumerated above.

2.2- Matrix Properties

We make efforts to have a matrix that have the same properties than the ones that represent a large corpus of documents. This means that we want to have a sparse matrix, with positive values, where

the values are proportional to the relevance of words (features) in the documents (records) ($1 \rightarrow present => relevant, 0 \rightarrow absent => irrelevant$) and where the number of documents is highly bigger than bigger than the number of words (20 times). This is to ensure that the algorithms that we are going to run on our data are used in conditions where they already shown that they could produce good results.

Concerning the sparcity, we end up with matrices that contains around 5.5% of non-zero values. Concerning the records size and the feature size, we have in average 3000 records per user for 125 features ($records_size = 24 * features_$size).

3. SINGULAR VALUE DECOMPOSITION (SVD)

We decide to run the SVD algorithm in the matrix data as a first baseline. Then we try to tune our matrix to improve the SVD results. Below, we explain the main transformations we did to the matrix data and compare their effects on the SVD. Second, we show some results that started to show some meaningful users' behaviors.

3.1- Transforming the Matrix

The SVD we launched in the initial version of the data (without any transformation) resulted in having the very recurrent features to dominate all the concepts. For example, the feature $Battery\_Health\_Good$ appeared in the top of all of the main concepts which is not descriptive of the behavior of the user.

Starting from the intuition that the very frequent features do not describe the behavior of the user (if a feature is always present then it does not impact the behavior of the user), we apply some matrix transformations to our data to decrease the gap between the importance of the very frequent features and the less important ones.

Let X be the data matrix and $\widetilde{X_k} = U_k S_k V_k$ the approximation of X that results from the truncated SVD by selecting the k most important singular values. SVD constructs $\widetilde{X_k}$ s.t the error $\|X - \widetilde{X_k}\|^2$ is minimized. This is nothing than the norm of the distance between the two matrices. Thus, a very big value in X (comparing to the others) makes the $U_k S_k V_k$ to converge to that value. This means that a metric that can express the importance of a feature i in the SVD decomposition is the sum of the values taken by this feature over all the records $\sum_{j=0,..,n-1} x_{i,j}$. The more this sum is important, the more the corresponding features takes importance in the SVD decomposition.

The Figure 5 shows the effect of different transformations we used on the features importance (for user 4). The $x - axis$ represents the ranks of the different features. $x = 0$ represents the most frequent feature in the data, $x = 1$ represents the second most frequent and $x = n$ the n^{th} most frequent feature in the data. The $y - axis$ represents the importance $\sum_{j=0,..,n-1} x_{i,j}$ that each feature gets depending on the transformation used. The $presence\_count$ curve is the initial matrix where no transformation has been applied to the matrix. The idf is the well-known inverse document frequency transformation (the others were tuned by ourselves). We can see from the graph that with using the idf transformation, the most frequent feature gets less importance that the second one. Moreover, the difference between the importance of the features is smaller in the idf curve that in $presence\_count$ curve. This means that less frequent features get nearly the same importance in the SVD decomposition than more frequent ones. This is the effect aimed by the idf transformation.

When applying the idf to the data and then running SVD, the results observed looks random. The top features in the concepts do not give any meaning to what the user is doing or uses to do. This is explained by the fact that the idf equilibrates too much the importance between the different features (as shown in Figure 5, the variation is very small). While in a document context the 20^{th} most frequent word in the corpus can express a lot the meaning of a document, the behavior of the user is much more described by the top-features (but not the too frequent ones). The ldc (stands for linear document count) transformation penalizes the most frequent features but then allows the other frequent features to have more importance than the less frequent one (see Figure 5). It is with this transformation that the results are the most meaningful when applying the SVD. The ldc transformation is discussed in more details in Appendix.B.

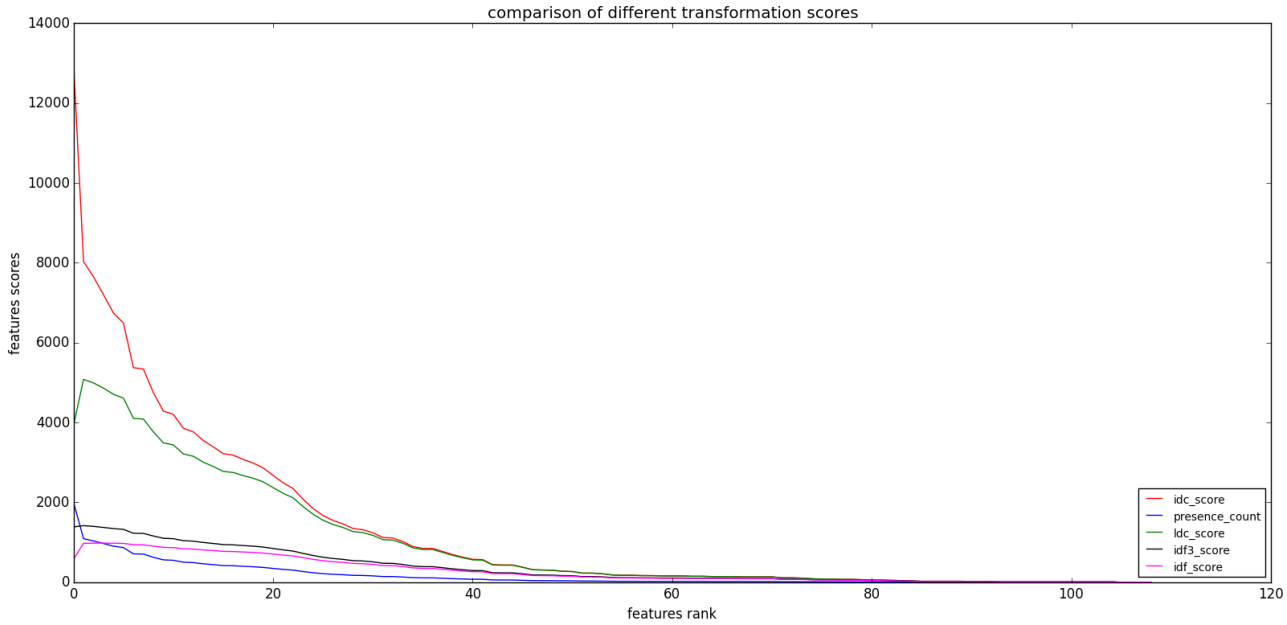

Figure 5: Effect of different transformations on the importance of features in the SVD decomposition

3.2- Results

Below (see Figures 6(a), 6(b), 6(c)) we present the results of the SVD obtained for user 4 when applying the ldc transformation. We display the $top - 20$ features for the $top - 3$ concepts. In the displaying, the singular values are normalized such that they sum to 1. This is to quantify the importance of each concept (to see how much variance of the data it expresses). The same is done for the values of the features in each concept.

The concept 0 shows the following: the user goes to work during the week days. He is more often at work during the second part of the day (from 12 A.M to 0 A.M). He uses his car to reach work. He use to plug his phone in the power while working.
The concept 1 shows the following: During the week ends mornings, he uses to be either at home or visit another place. He sometimes does some velo at that moment. He likes also to read some news with his smartphone at that time.
The concept 2 shows us that starting from the afternoon, he is very often at home in the week end.

3.3- Critics & Remarks

Applying some transformations to the data matrix enabled us to get some meaningful results using the SVD decomposition. However, we can still observe a lot of noise in our results and a lot of features that do not give meaning to the clusters they are in. We can also note the presence of features like $tilting$ or $plugged\ in\ usb$ that are much less expressive of the behavior of the user that other features like $time$ and $location$. Moreover, the behaviors we are catching are not enough precise to be relevant and too few to represent the habits of the users.
Anyway, we knew from the beginning that SVD would not give us amazing results and that is should be considered as primary results.
An idea we are considering (for the next algorithms) is to give initial weights to the features to indicate the features that should have more importance than the others (for example classifying the features into important, neutral, not_relevant).

When dealing with a corpus of documents, looking to the clusters of words and the sense of the top words gives a quite precise idea about the quality of the clustering and the words that should not belong to the cluster. In our case looking to the clusters of features tells us if the clusters could represent a behavior of the user. However, they do not tell us if those features really express a habit of the user or if they were just randomly clustered together. For example, is it true that the user 4 uses to do some bicycle in the week ends mornings? Or is the feature $bicycle$ was there by chance? Thus, one of the main challenges we currently have is to find a way to evaluate the relevance of our results.

Those remarks close this section and lead us to talk about what we are planning to do next.

Figure 6(a): SVD result for user 4 using ldc transformation : most important features for the concept 0

Figure 6(b): SVD result for user 4 using ldc transformation : most important features for the concept 1

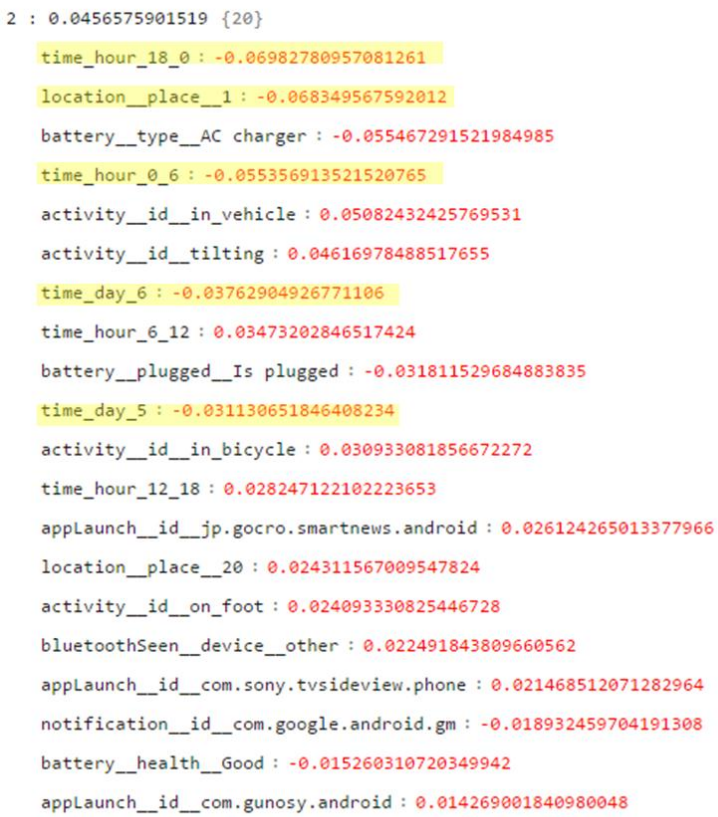

Figure 6(c): SVD result for user 4 using ldc transformation : most important features for the concept 2

4. NEXT STEPS

We have now available the beginning of meaningful results. Now, our goal is to consider the different approaches we could take to improve our work. One approach is to consider probabilistic models with latent variables. Another is to consider more sophisticated matrix decomposition techniques.

Moreover, we are thinking about the point raised in 3.3, and trying to find a way to be able to evaluate accurately our results.

APPENDIX A. Exhaustive list of the represented features in the matrix

The features represented in the data matrix are the following:

- Activity in vehicle
- Activity in bicycle
- Activity on foot
- Activity still
- Activity tilting
- Time day number (0 is Monday, 6 is Sunday)
- Hour range 0am-6am
- Hour range 6am-12pm
- Hour range 12pm-18pm
- Hour range 18pm-0am
- Location number (0 is the most frequent location, $k-1$ the k^{th} frequent, k the others)
- Application Launch name (only the k most frequent represented)
- Notification name (only the k most frequent represented)
- Bluetooth paired device name (the device name that was paired with the smartphone)
- Bluetooth seen device name (detected by the phone but not paired. only the k most frequent represented)
- Battery health is good
- Battery health is cold
- Battery health is overheat
- Battery health is dead
- Battery plugged (binary feature that takes 1 if a battery is plugged in the phone)
- Battery plugged type usb charger
- Battery plugged type AC charger
- Battery plugged type wireless plugging
- Headset plugged (binary feature that takes 1 if a headset is plugged in the phone)
- Headset microphone plugged (binary feature that takes 1 if a headset that contains a microphone is plugged in the phone)

APPENDIX B. Linear Document Count (LDC) Transformation

The linear document count (ldc)applies the following transformation to the data:

Let n_i be the number of records where the feature i appears and n the total number of features. Then the feature i gets the score:

$$\mathrm{l}dc(n_i) = -\frac{\ln(n)}{n-1}.n_i + \frac{nln(n)}{n-1}$$ (see Figure 7)

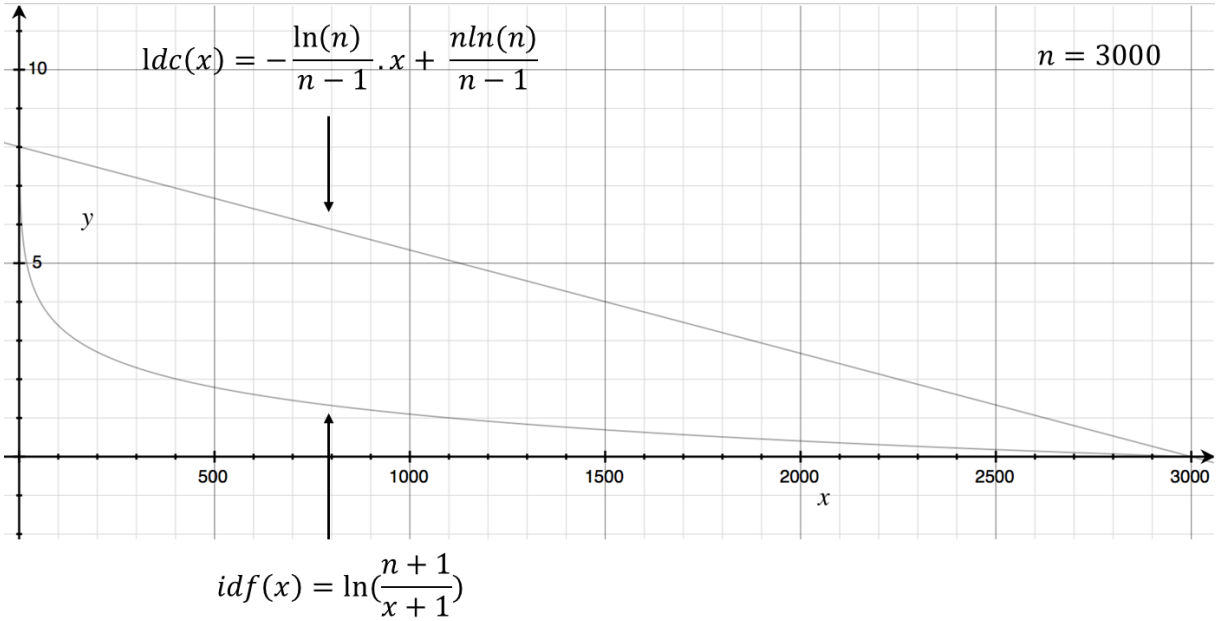

Figure 7: idf and ldc functions for 3000 records

The figure 7 shows the curves of both ldc and idf in function of x, where x represents the number of records where a feature is present from a total of $n = 3000$ records. We can see that ldc is a decreasing linear curve where the most frequent features get smaller score than the less frequent ones.

For all of the users, most of the features are present in less than 10% of the records and very few of them are contained in more than 30% of the records (see Figure 8). This means that most of the features are too rare and will get a score $\geq f(\frac{n}{100}.10)$ where f is the function we choose for scoring ($ldc, idf, \ldots$). The more representative features will get a score $\leq f(\frac{n}{100}.30)$.

By looking to the Figure 6, we can understand the effect of idf and $\mathrm{l}dc$ on the data. For idf the too rare features gets a score of $\geq idf\left(\frac{3000}{10}\right) = 2.1$ and the most representative ones $\leq idf\left(\frac{3000}{10}.3\right) = 1.1$.

Thus the too rare features are increased at least by a factor of $\frac{2.1}{1.1} = 1.9$ with respect to the most frequent ones. The effect of ldc is to decrease this factor ($\frac{ldc\left(\frac{3000}{10}\right)}{ldc\left(\frac{3000}{10}.3\right)} = \frac{7.1}{5.6} = 1.26$) so that the most frequent features still gets more importance that the less frequent ones.

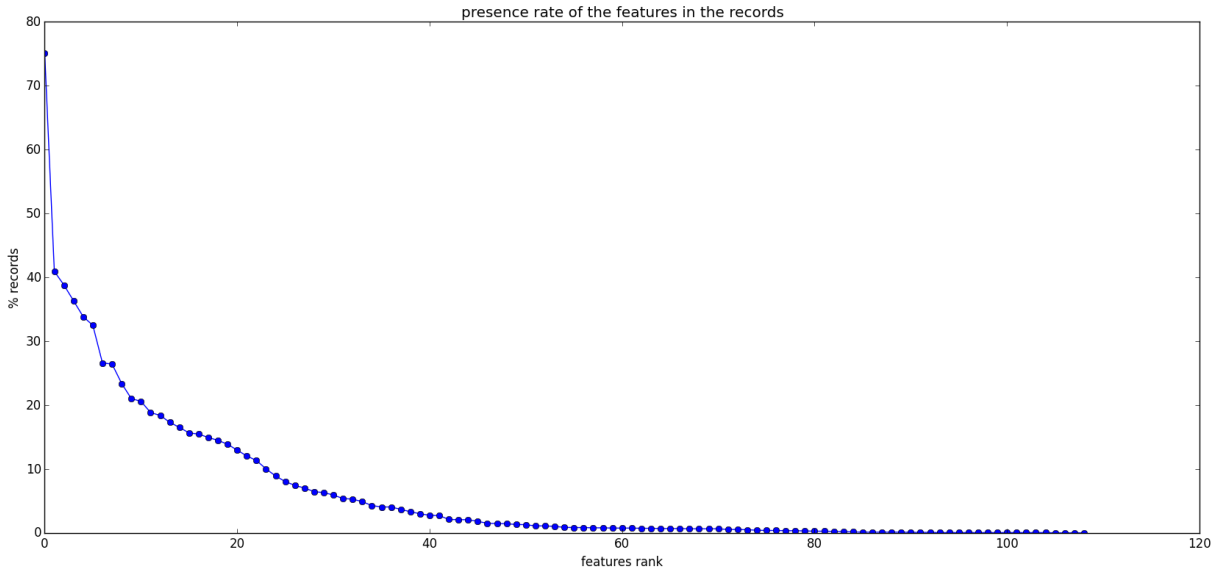

Figure 8: Features presence percentage for user 4