ALMA MATER STUDIORUM BOLOGNA - University of Bologna
Project work in Machine Learning

Case study: Indoor Localization

Author: Lorenzo Mario Amorosa - lorenzomario.amorosa@studio.unibo.it

1. Overview

This project work consists of a set of tasks regarding indoor localization, such as:

- Room and floor classification using machine learning methods
- WAPs position inference via trilateration techniques
- WAPs coverage analysis using correlation measures

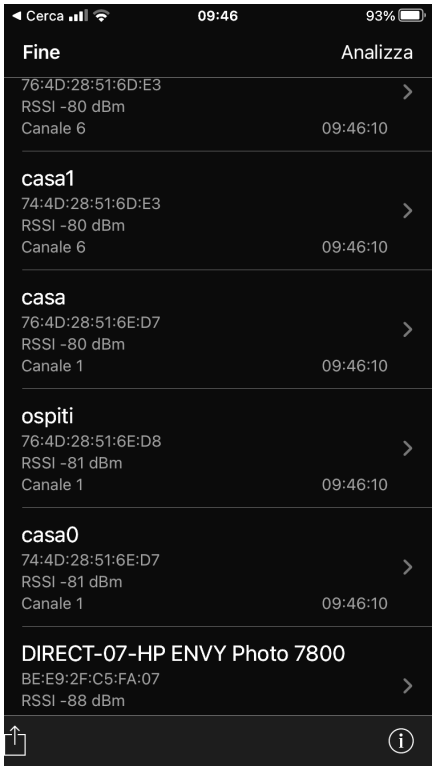
In particular, since the already available wireless signals are used to profile a location, the indoor localization is based on infrastructure-less approaches. On the contrary, if data were collected using a dedicated network (e.g. BLE), we would have talked about infrastructure-based approaches.

The code is available in this Colab notebook and on Github.

2. Dataset

The dataset employed is UJIIndoorLoc [1]: a multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems, and it is available on Kaggle [2].

The dataset consists of almost 20.000 recordings of around 500 WAPs intensity (RSSI, received signal strength indicator). Each recording has been collected using a smartphone app with an interface similar to the one shown in the picture on the right. The user could start a network analysis and all the visible WAPs and GPS user coordinates were recorded as well.



The dataset covers a surface of 108703 m^2 including 3 buildings of the Jaume I University (Spain) with 4 or 5 floors depending on the building. The number of different places (reference points) appearing in the dataset is over 900.

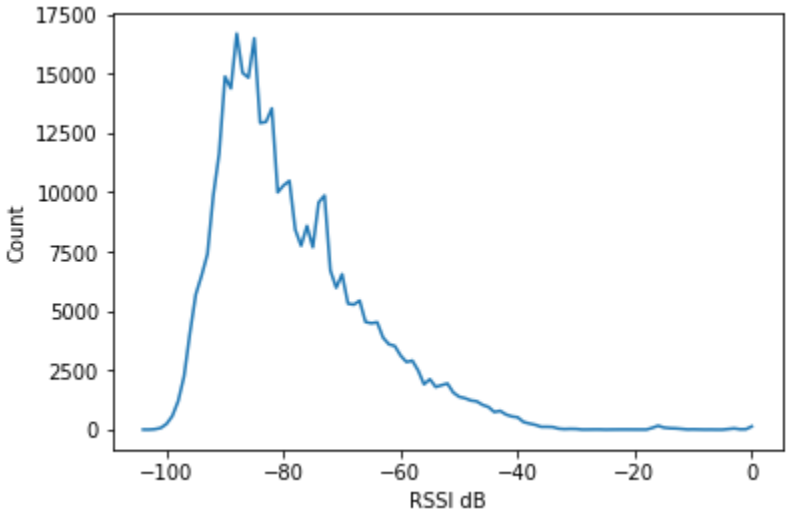
3. Preprocessing

The only columns of the dataset kept are:

- the WAPs detected at least once
- latitude and longitude
- building, floor, spaceID and relative position to the spaceID.

The creators of the dataset did another data collection after several months in the same places for a separate dataset. Several WAPs were detected only in one of these two sessions, and since we consider only the data collected in the first session we discard the WAPs which appeared only in the second session.

The dataset is highly sparse, indeed the zero values are 96.13%. Apart from them, the 71.22% of non-null detection are in range [-95, -73] dB. In the picture below it can be seen the overall number of detection for each RSSI intensity in range [-104, 0] dB.



4. Floor and room classification

The first part of the analysis consists in both room and floor prediction on the basis of WAPs' RSSI through machine learning algorithms. Several models are trained

using cross validation. In the end, the best models are compared using statistical methods.

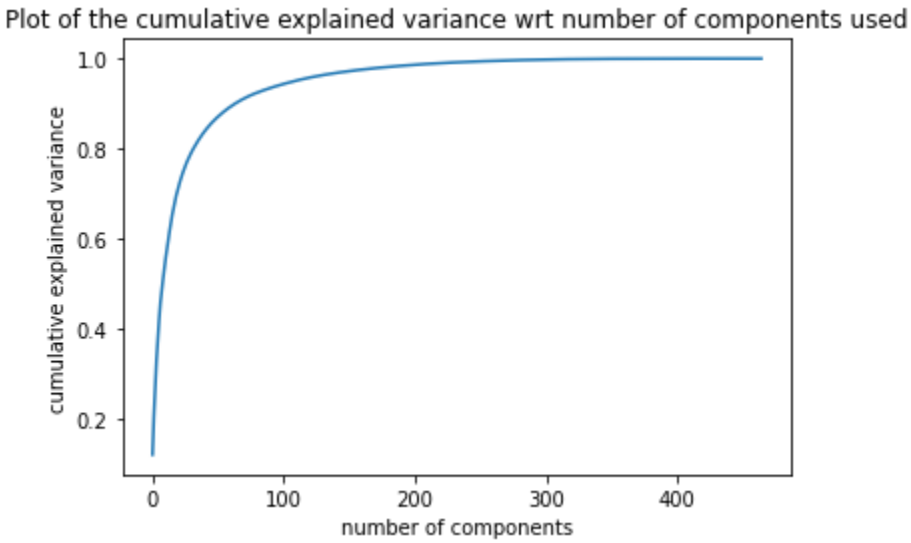
Two sets of labels are firstly generated for each record (one representing rooms, the other floors), then the RSSI negative measures are scaled to positive values.

The models employed in the cross validation are:

- Support Vector Machine, with hyperparameters:
  - kernel: rbf, linear
  - gamma: scale, 1e-3, 1e-4 (for rbf kernel)
  - C: 10, 100, 1000
- K Nearest Neighbor, with hyperparameters:
  - n_neighbors: from 1 to 10
  - metric: euclidean, manhattan, chebyshev
- Random Forest, with hyperparameters:
  - max_depth: from 5 to 50 by steps of 5

The metrics used for tuning the models are accuracy and f1-macro score (in order to take into account possible class imbalances).

In a subsequent section Principal Component Analysis (PCA) is applied on the dataset. Many of the recorded values are zeros (96.13%), therefore a significant dimensionality reduction can be applied, with a negligible loss of information. In particular, 96.03% of the variance is explained using 125 components out of over 450. In the picture below it can be seen the plot of the cumulative explained variance with respect to the number of components used.



In the following tables the results obtained by the best models on the test set are reported:

Predict Room - Accuracy			
Model	Hyperparameters	PCA	Score
Random Forest	max_depth: 50	No	0.84
Support Vector	C: 100, gamma: 0.0001, kernel: rbf	Yes	0.81

Predict Room - F1 Macro			
Model	Hyperparameters	PCA	Score
K Nearest Neighbor	metric: manhattan, n_neighbors: 1	No	0.80
Support Vector	C: 100, gamma: 0.0001, kernel: rbf	Yes	0.79

Predict Floor - Accuracy			
Model	Hyperparameters	PCA	Score
Random Forest	max_depth: 45	No	0.99
Support Vector	C: 10, gamma: 0.0001, kernel: rbf	Yes	0.99

Predict Floor - F1 Macro			
Model	Hyperparameters	PCA	Score
Support Vector	C: 100, gamma: 0.0001, kernel: rbf	No	0.99
Support Vector	C: 10, gamma: 0.0001, kernel: rbf	Yes	0.99

As it can be seen, it is really easy for the models to correctly predict the floor with respect to the room, but this is a consequence of the fact that there are 905

distintinct rooms and only 13 distinct floors. The performances get slightly worse when using PCA, but the training time considerably reduces as well.

The best models are compared in order to see whether there is a statistical significant difference between them. The method employed is explained in this pdf [3]. In brief, the error of the metrics of the models *e* can be approximated by a Normal distribution in case the metrics are assessed using a sufficiently large enough test set (with number of samples $N > 30$).

$$e \sim N(\mu, \sigma) \qquad \sigma^2 = \frac{e \cdot (1 - e)}{N}$$

We can compute the difference *d* between two errors e_1 and e_2, which can still be approximated by a Normal distribution.

$$d \sim N(d_t, \sigma_t) \qquad \sigma_t^2 = \sigma_1^2 + \sigma_2^2 = \frac{e_1 \cdot (1 - e_1)}{N_1} + \frac{e_2 \cdot (1 - e_2)}{N_2}$$

Finally, d_t is obtained with a confidence of 1 - α:

$$d_t = d \pm Z_{\frac{\alpha}{2}} \cdot \sigma_t$$

If the interval of d_t contains the zero then the difference between the two models is not statistically significant. It is possible to reduce the confidence (resulting in a smaller $Z_{\alpha/2}$ and consequently in a narrower interval for d_t) to accept the hypothesis that two models are statistically different.

In the experiment the accuracy and f1-macro score obtained by the best models are compared with a confidence of 90%. The best models are tuned both by accuracy and f1-macro score, and some of them also use PCA. This kind of analysis is useful to identify those models which have good performances in both metrics.

All the best models resulted statistically equivalent in predicting the floor on test data both for accuracy and f1-macro score, so it makes no difference whether to use or not PCA or whether the tuning is done by accuracy or f1-macro.

Whereas it resulted that in predicting the room the Random Forest model trained without PCA and tuned by accuracy was statistically better with respect to the accuracy than the other best models tuned for f1-macro and with PCA. Additionally, all the models which predict the room resulted statistically equivalent with respect to the f1-macro score. As a result, it would be advisable to employ in production the Random Forest model tuned by accuracy and without PCA since it is better in terms of accuracy and as good as the others in terms of f1-macro score.

5. WAPs position estimation via trilateration

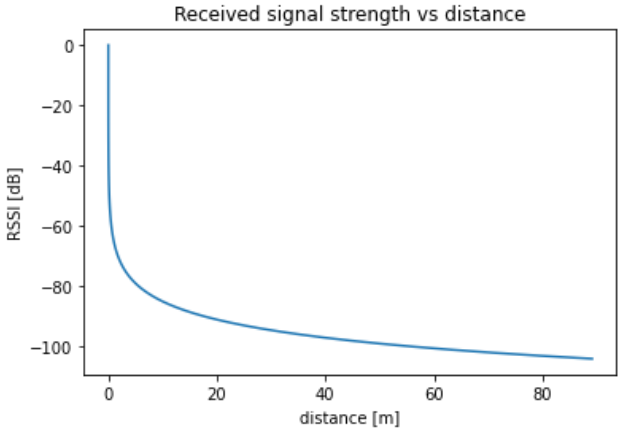
In this section it is addressed the problem of computing the coordinates (latitude, longitude) of the WAPs, which are not provided within the dataset. The method employed is trilateration [4], in particular the solution to the mathematical formulation is provided through an optimization method rather than a geometrical solution.

The main problem of position estimation is that it has to reconstruct a complex information, i.e. the position, starting from several simple elements, i.e. the measured distances between the devices and the WAPs.

In particular, we do not have the measured distances but only the RSSI. It is possible to go to the distance from the RSSI following this reasoning [5]: supposing a calibration power *Tx* (e.g. -65 dB) obtained at 1 meter from the WAP, we can derive *d_dB*, namely the difference between *Tx* and the measured RSSI, which is in decibel. After we convert *d_dB* to a linear ratio *d_linear* and if we assume conservation of energy, then the signal strength must fall off as $1/r^2$. Consequently we obtain the distance *r*:

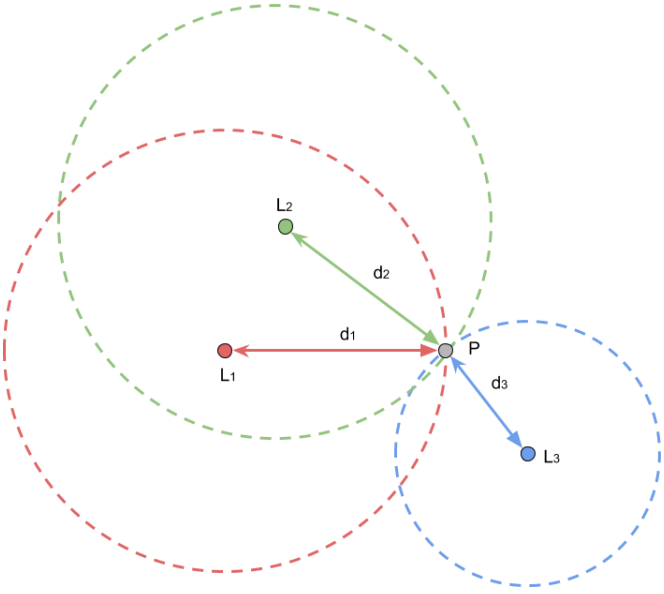
$$power = \frac{power\_at\_1\_meter}{r^2} \qquad r = \sqrt{d\_linear}$$

It has to be noticed that this modeling is only theoretically good [6]. For example, if you are inside a steel building, then perhaps there will be internal reflections that make the signal decay slower than $1/r^2$. If the signal passes through a human body (water) then the signal will be attenuated. It is very likely that the antenna does not have equal gain in all directions. Metal objects in the room may create strange interference patterns and so on. Another strong assumption that we make is that each WAPs shares the same *Tx*. However, this modelling is the best effort we can do given the absence of additional information. In the picture below it can be seen which is the RSSI for a given distance in meters.



Another important step is to convert the coordinates expressed in UTM (Universal Transverse Mercator coordinate system) in the dataset, which basically represent a point by means of its distance in meters from specific reference points, into latitude and longitude.

Once we transform all the RSSI values to distances, we convert UTM to latitude-longitude and we have at least 3 measurements (ideal case), we can determine the position of the hidden WAP.



In the picture above the devices are marked as L_i and the inferred WAP as P. In mathematical terms we can express the distances between the elements of the system with a set of equations, given (x, y) the unknown coordinates of the WAP and (x_1, y_1), (x_2, y_2), (x_3, y_3) the known coordinates of the devices.

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2$$
$$(x - x_2)^2 + (y - y_2)^2 = d_2^2$$
$$(x - x_3)^2 + (y - y_3)^2 = d_3^2$$

In the ideal case, we can obtain (*x*, *y*) by solving this system. Of course the environment impacts on the signal propagation, and it can happen that there is no solution to the system. In fact many more measurements are often collected to determine the actual position of the hidden WAP, resulting in an overdetermined system.

The problem of trilateration can be approached from an optimization point of view. Given a point *X*, we can estimate how well it replaces the WAP *P*. We can do this simply by calculating its distance from each device L_i. If those distances perfectly match with their respective distances d_i, then *X* is indeed *P*. The more *X* deviates from these distances, the further it is assumed from *P*.

We need to find the point *X* that minimizes a certain error function. We have a distinct source of error e_i for each device:

$$e_i = d_i - dist(X, L_i)$$

A very common way to merge these different contributions is through Mean Squared Error (MSE). This solution can take into account an arbitrary number of points. The higher is the difference between the expected distances and the computed distances, the higher is the MSE.

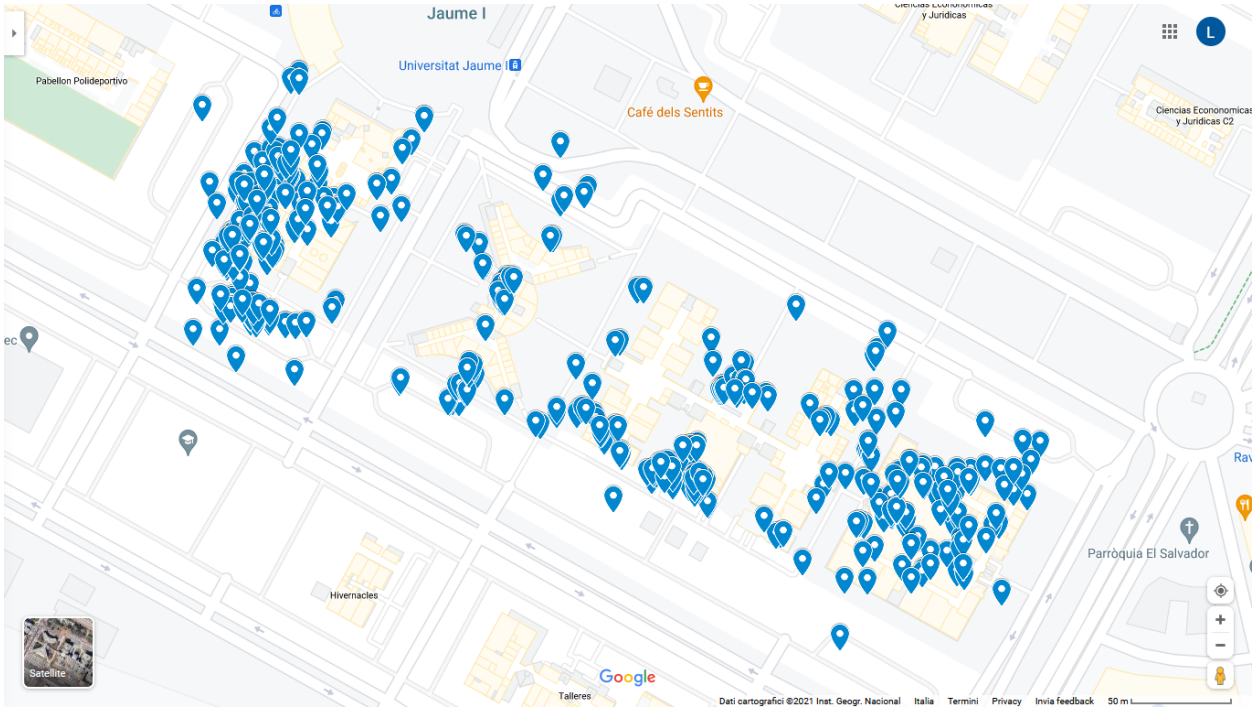
$$MSE = \frac{\sum [d_i - dist(X, L_i)]^2}{N}$$

What is left now is to find the point *X* that minimizes the MSE. Luckily, scipy comes with several optimization algorithms that we can use. While optimization algorithms can solve many problems, it is unrealistic to expect them to perform well if we provide little to no additional information to them. One of the most important aspects is the initial guess. Providing a good starting point could indeed reduce its execution time significantly. There are several initial guesses that one could use, a sensible choice is to pick as starting point the centroid of the devices which detected the WAP.

In brief, the main steps to compute the WAP location are the following:

1. all the recording where the WAP is detected are taken
2. for each recording, the latitude, the longitude of the device and its distance from the WAP are kept
3. we end up with a tuple containing a list of distances device-WAP and a list of device coordinates
4. this information are fed to the function in charge of minimizing the MSE, in order to obtain the point *P* of the WAP whose distances with the devices are the most similar to the ones expected

In the following image it is possible to visualize where the inferred positions of the WAPs are. Notice that the records of the dataset are taken only in the 3 right most buildings, but many access points in a fourth building (the leftmost one) were recorded.



6. WAPs coverage analysis

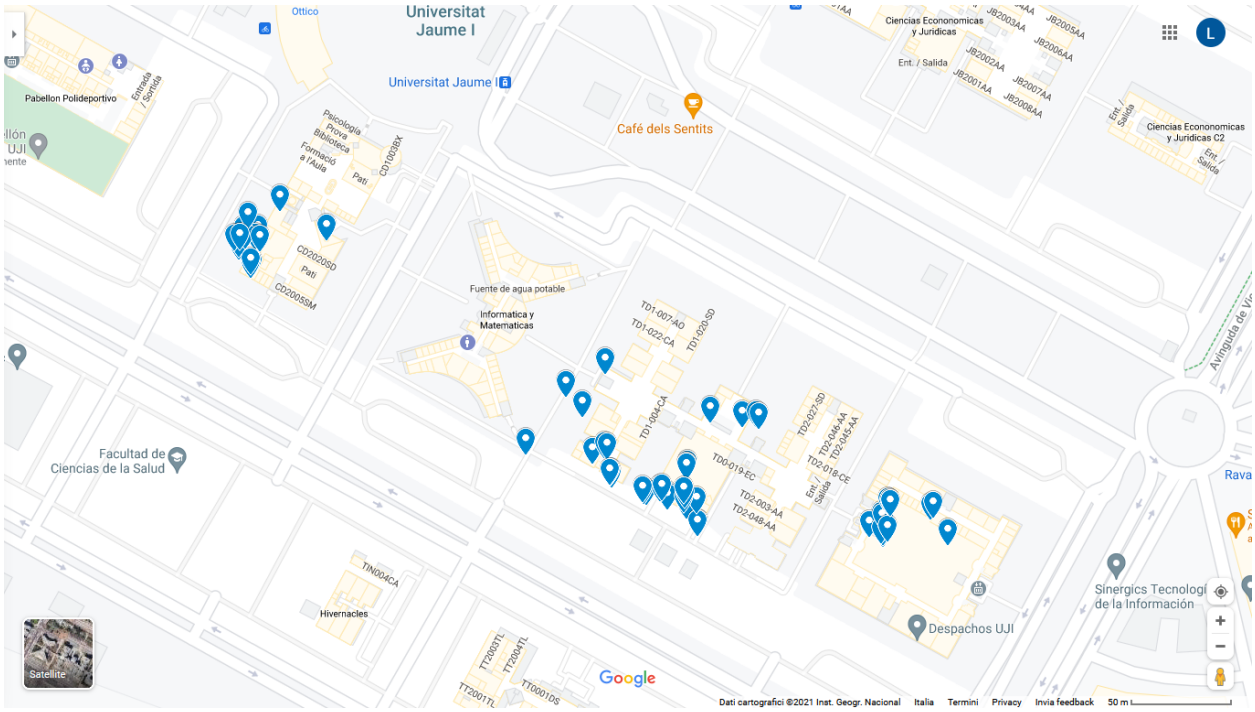
In this section the WAPs reciprocal coverage is analysed through Spearman's correlation [7]. This is a correlation coefficient that assesses how well the relationship between two variables can be described using a monotonic function. While Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not).

The sign of the Spearman correlation indicates the direction of association between *X* (the independent WAP) and *Y* (the dependent WAP). If *Y* tends to increase when *X* increases, the Spearman correlation coefficient is positive. If *Y* tends to decrease when *X* increases, the Spearman correlation coefficient is negative. A Spearman correlation of zero indicates that there is no tendency for *Y* to either increase or decrease when *X* increases. The Spearman correlation increases in magnitude as *X* and *Y* become closer to being perfectly monotone functions of each other.

In particular, the Spearman correlation function of the scipy library provides also the confidence (i.e. *1 - p-value*) according to which the null hypothesis (i.e. two WAPs are not correlated) can be rejected. So the higher the *p-value*, the higher the probability that two WAPs are not correlated. Conversely, the lower the *p-value*, the higher the probability that two WAPs are correlated.

In order to deal with the high sparsity of the dataset, when evaluating the correlation between two WAPs only those records where at least one of the two WAPs compared is not null are taken. Otherwise all WAPs would result much more correlated because of all the many detections taken far away from them, where none of the two WAPs was recorded and their respectively RSSI is set to zero.

Moreover, the Spearman correlation is computed pairwise between all the WAPs, and the WAPs that correlated with the highest number of WAPs are highlighted. For each WAP, it is counted the number of times in which it results positively correlated with another WAP with a confidence of 99%. The WAPs which correlate with at least another 50 WAPs (which are 63 in total) are marked in the picture below. Therefore they are the WAPs with the worst coverage.



7. Bibliography

[1] Joaquin Torres-Sospedra et al., 2014, UJIIndoorLoc: A New Multi-building and Multi-floor Database for WLAN Fingerprint-based Indoor Localization Problems

[2] UJIIndoorLoc Kaggle dataset

[3] Statistical model comparison

[4] True-range multilateration

[5] Obtaining the distance from RSSI

[6] Ambili Thottam Parameswaran et al, 2011. Is RSSI a Reliable Parameter in Sensor Localization Algorithms

[7] Spearman's rank correlation coefficient