Indoor Localization

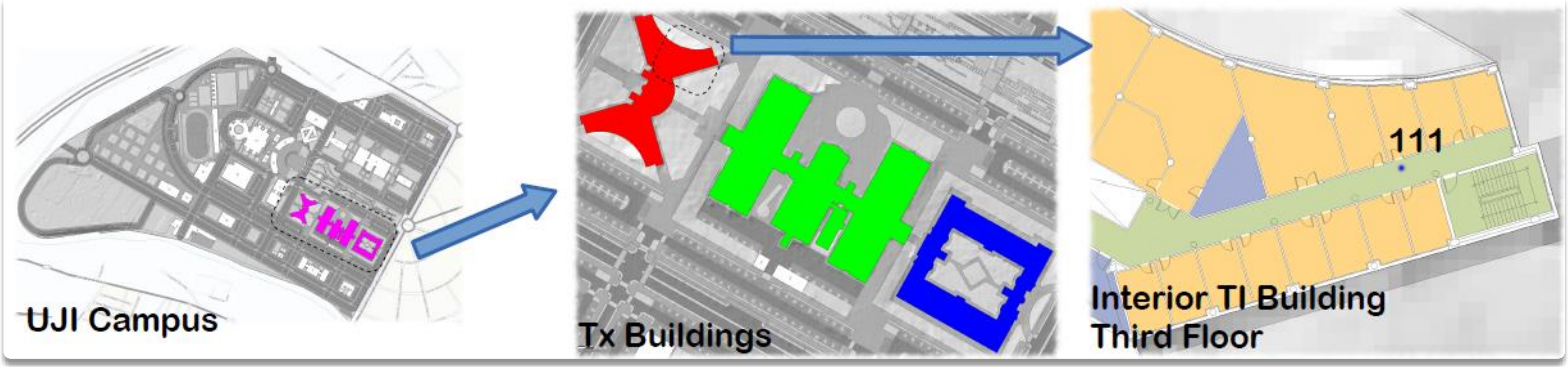
PROJECT WORK IN MACHINE LEARNING — LORENZO MARIO AMOROSA
MASTER DEGREE IN ARTIFICIAL INTELLIGENCE— UNIVERSITY OF BOLOGNA

Overview: Main Tasks

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

Dataset: UJIIndoorLoc

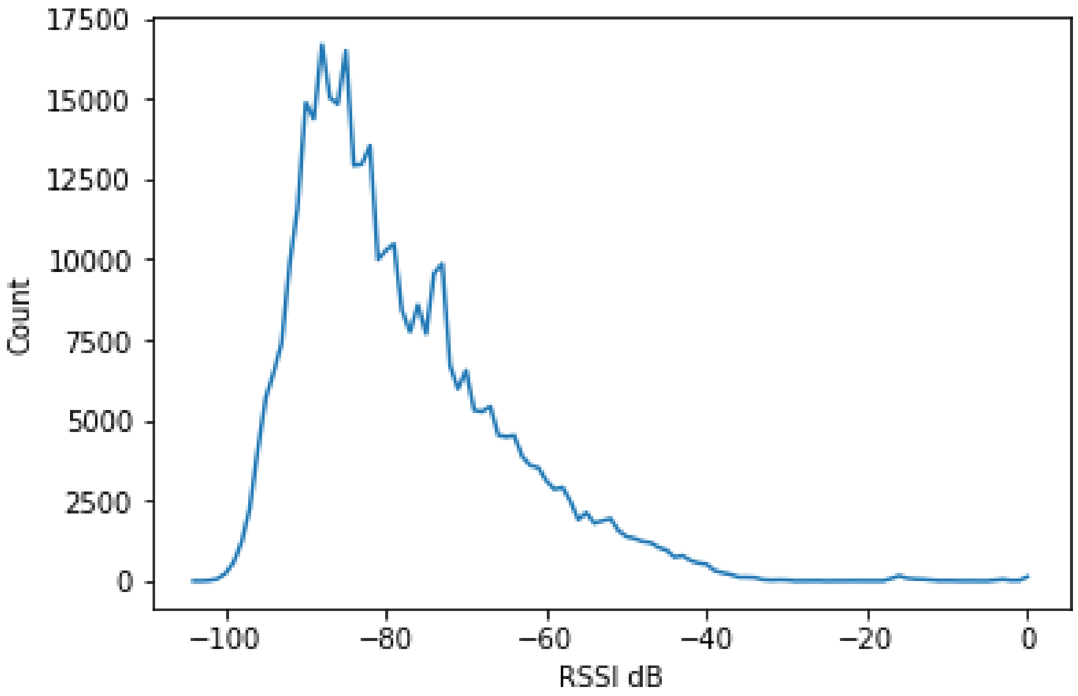
- Multi-building and multi-floor dataset (905 rooms within 13 floors)
- WLAN fingerprint-based → infrastructure-less localization
- 20.000 RSSI recordings within a surface of 108.703 m^2



Pre-processing

- Data kept:
  - The WAPs detected at least once
  - Latitude and longitude, converted from UTM (Universal Transverse Mercator coordinate system)
  - Building, floor, spaceID and relative position to the spaceID

Data Visualization



Overall number of detection for each RSSI intensity in range [-104, 0] dB

- Highly sparse dataset — the zero values are the 96.13%
- The 71.22% of non-null detection are in range [-95, -73] dB

Floor and room classification

Floor and room classification

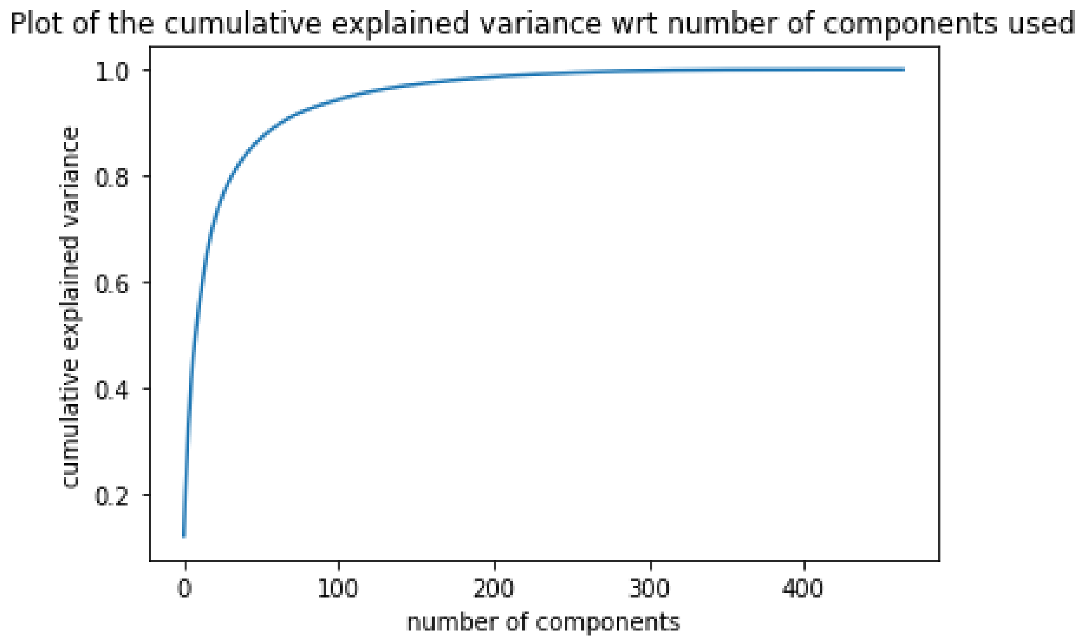
- Room and floor prediction on the basis of **WAPs' RSSI** using cross validation tuning both **accuracy** and **f1-macro score** on:

- **Support Vector Machine**:
  - kernel: rbf, linear
  - gamma: scale, 1e-3, 1e-4 (for rbf kernel)
  - C: 10, 100, 1000

- **K Nearest Neighbor**:
  - n_neighbors: from 1 to 10
  - metric: euclidean, manhattan, chebyshev

- **Random Forest**:
  - max_depth: from 5 to 50 by steps of 5

Principal Component Analysis (PCA)



Cumulative explained variance wrt number of components used

- Highly sparse dataset — the zero values are the 96.13%

- Dimensionality reduction
- The 96.03% of the variance is explained using 125 components out of over 450
- ML models trained also on PCA dataset

Best models: room prediction

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

Best models: floor prediction

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

Statistical comparison of 2 models

- The error of the metrics of the models e can be approximated by a Normal distribution in case the samples are N > 30:

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

- The difference *d* between two errors e_1 and e_2 can still be approximated by a Normal distribution:

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

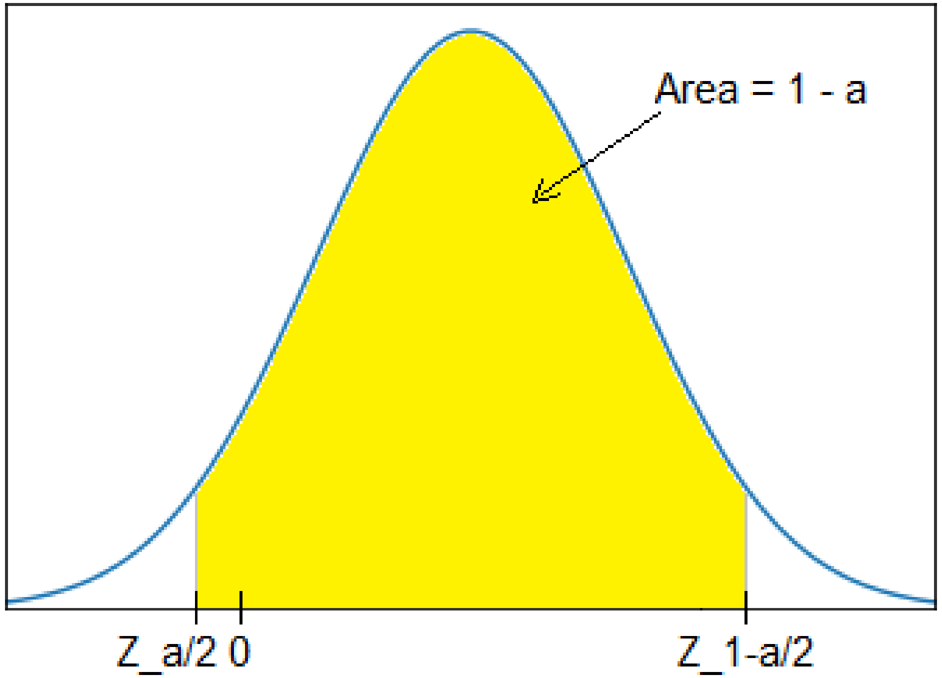
Statistical comparison of 2 models

- The mean d_t is obtained with a confidence of 1 - a:

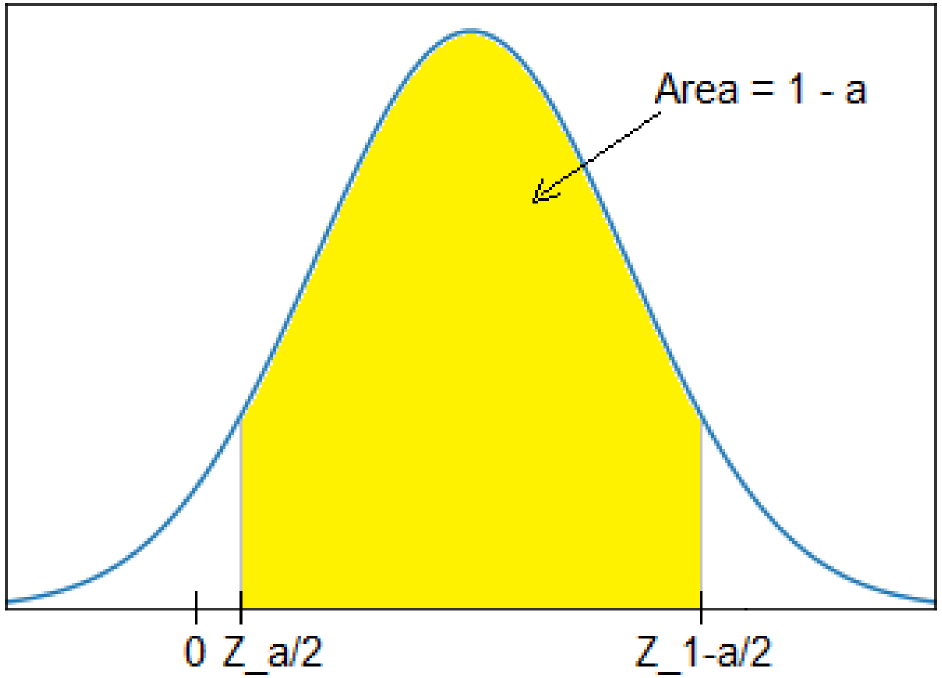
$$d_t = d \pm Z_{\frac{\alpha}{2}} \cdot \sigma_t$$

- If the interval of d_t contains the zero $\Rightarrow$ the difference between the two models is not statistically significant
- Reduce the confidence 1 – a (increase a) $\Rightarrow$ accept the hypothesis that two models are statistically different, smaller $Z_{a/2}$ and narrower interval for d_t

Statistical comparison of 2 models



The confidence interval includes the zero ➡ NO statistical difference



The confidence interval doesn't include the zero ➡ statistical difference

Statistical comparison outcome

- Best models differ for: **PCA**, **tuning metric** and **scope** (floor/room prediction)
- The best models are compared with **confidence 1 – a = 90%** and on a **test set** with cardinality of almost **N = 4000**
- **Floor prediction**: **no statistical difference** between models for both metrics
- **Room prediction**: **Random Forest** model trained without PCA and tuned by accuracy is **statistically better** with respect to the **accuracy** than the other best models tuned for f1-macro and with PCA and it is **equivalent** to others with respect to the **f1-macro score**

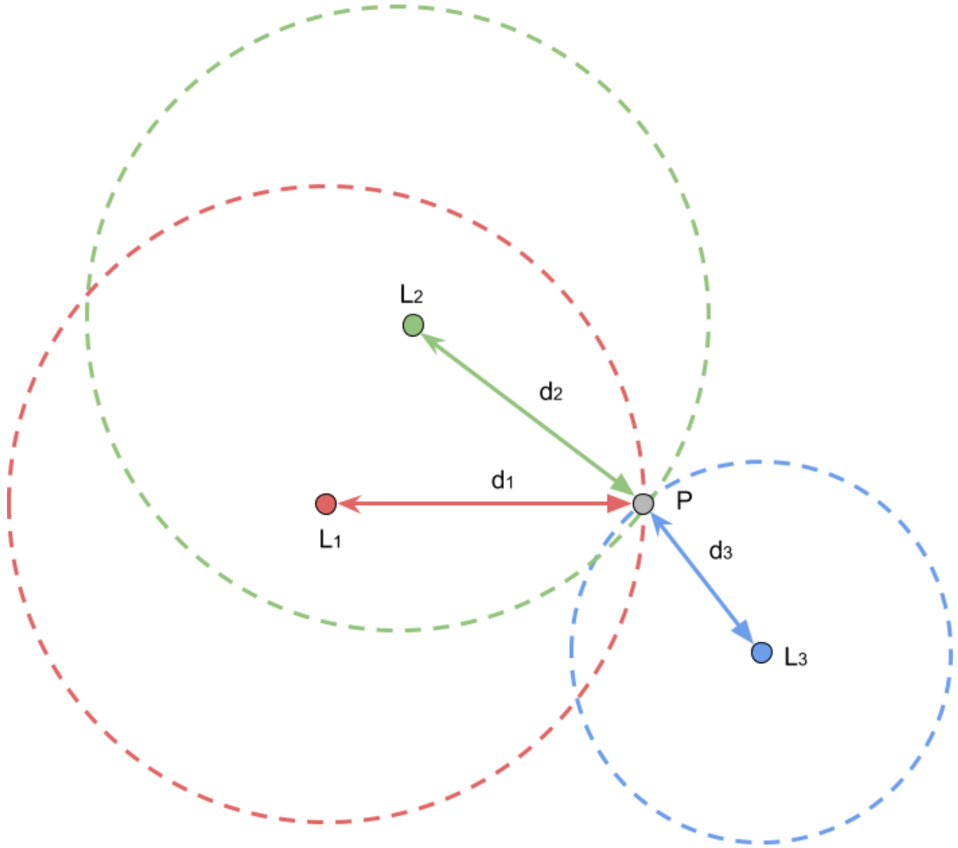
Random Forest model tuned by accuracy and without PCA preferable in room prediction

WAPs position estimation via trilateration

WAPs position estimation via trilateration

- Compute the **coordinates** (latitude, longitude) of the **WAPs**, not provided within the dataset
- Mathematical method: **Trilateration** ➡ solved with **optimization** technique
- Trilateration aims to **reconstruct the position** starting **from several measured distances** between the devices and the WAPs

Trilateration: Mathematical formulation



- At least 3 devices for unique positioning
- WAP *P* in unknown position (*x*, *y*)
- Devices L_i in postion (x_i, y_i)

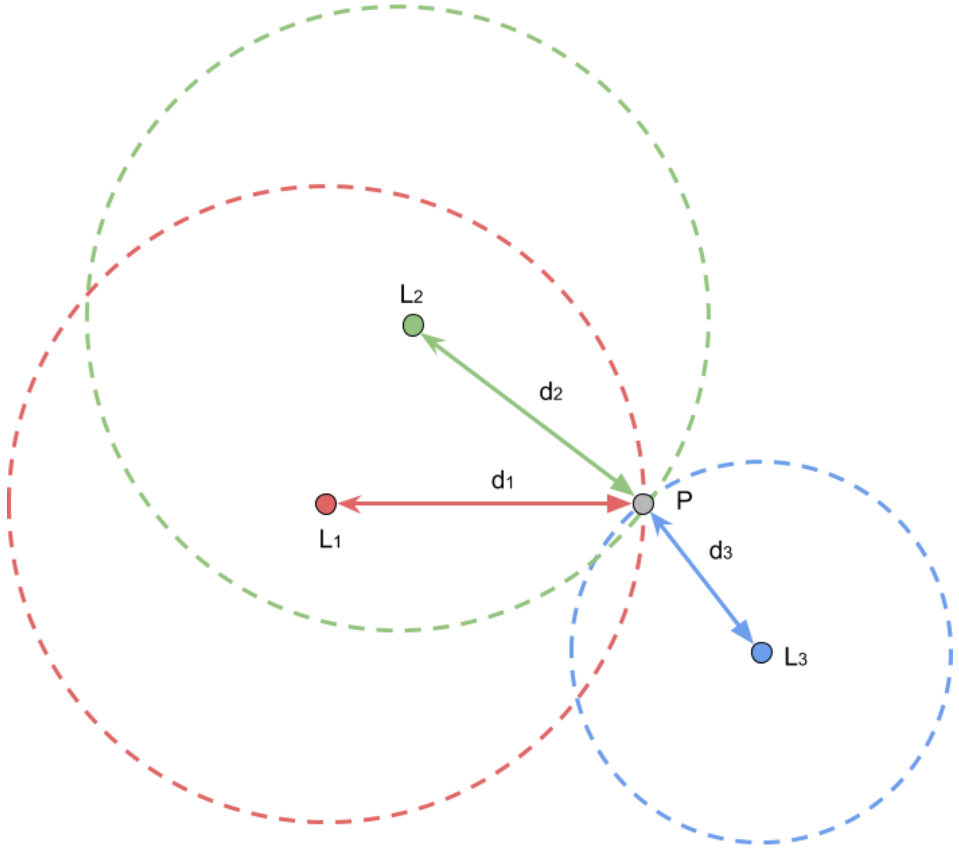
$$(x - x_1)^2 + (y - y_1)^2 = d_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 = d_3^2$$

Often NO solution because of the environement

Trilateration: Optimization

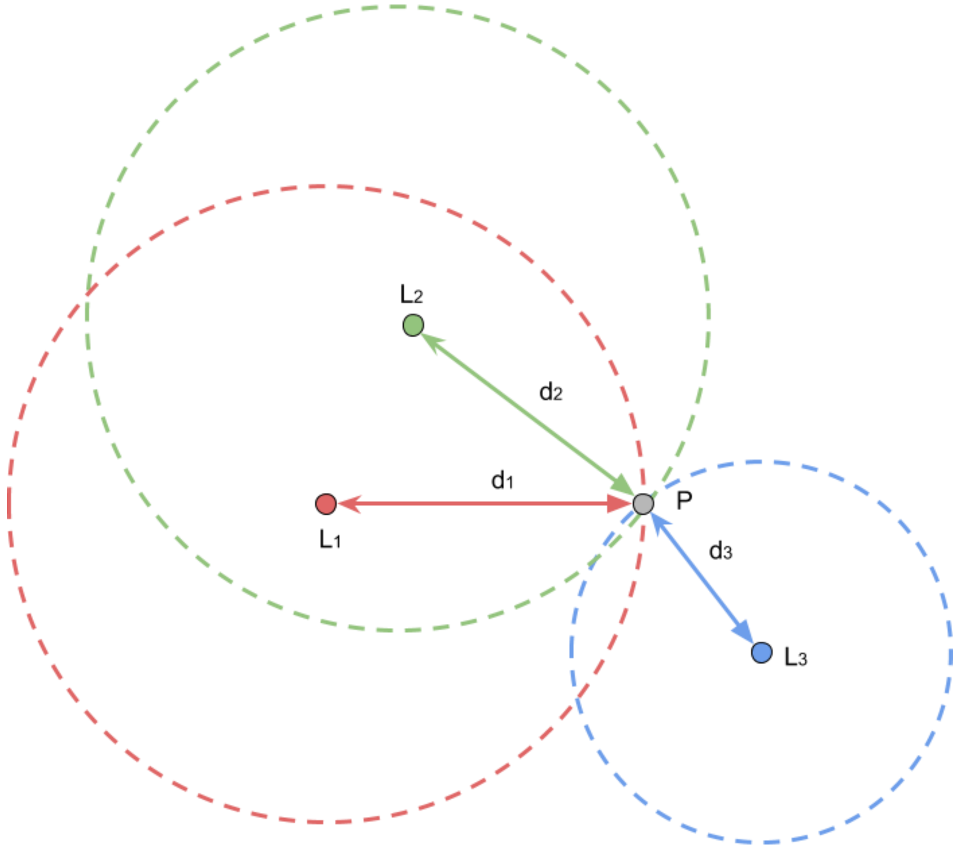


- Instead of solving the system of equations

⇩

- Find point *X* that better replaces *P*
- If the distances between *X* and the devices perfectly match with the respective distances d_i, then *X* is indeed *P*
- The more *X* deviates from these distances, the further it is assumed from *P*

Trilateration: Optimization



- Minimization of error function:

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

- For all devices:

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

- Minimized with `scipy.optimize.minimize` to obtain the estimated position for each WAP

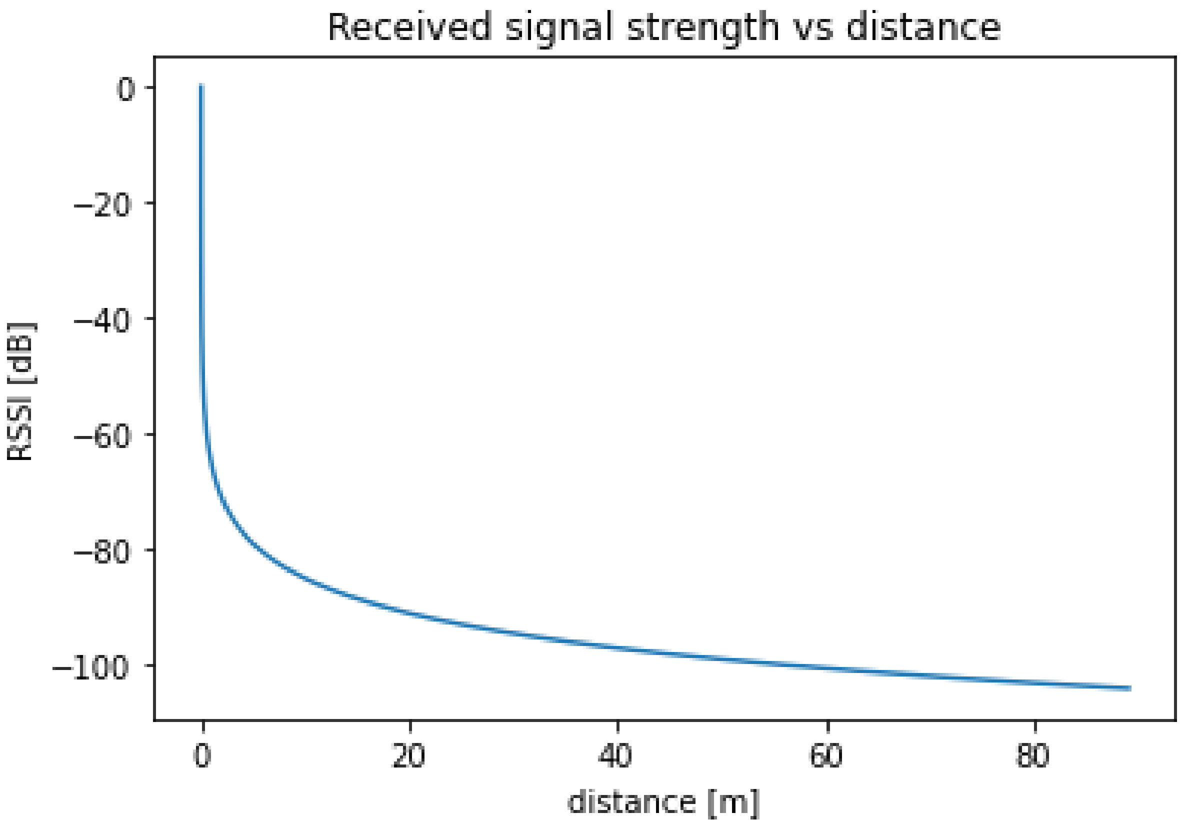
WAPs inferred positions

Appendix - From RSSI to distance

- In the dataset we have RSSI only ➡ need to compute distances
- Two assumptions needed: WAP calibration power T_x (e.g. -65 dB) and conservation of energy, so signal strength falls off as $1/r^2$ (no refraction, etc.)
- We can get: $d\_dB = T_x - RSSI$ [dBm] ➡ $d\_linear = 10^{d\_dB/10}$ [mW], consequently:

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

Appendix - From RSSI to distance



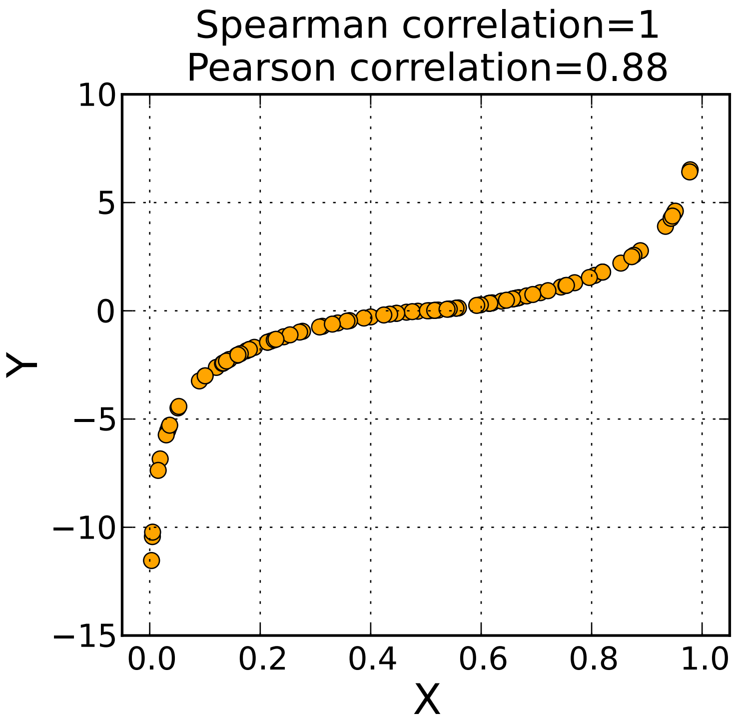
Received signal strength vs the distance

WAPs coverage analysis

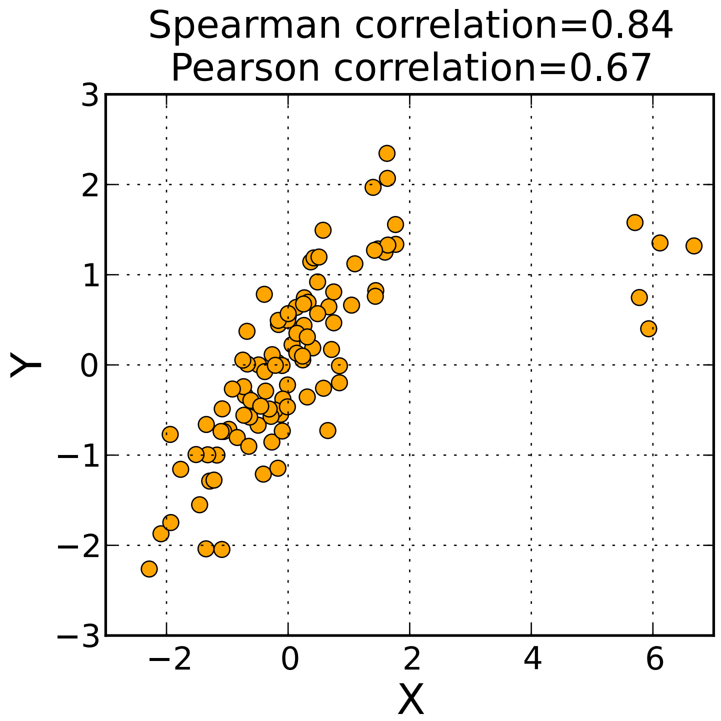
Spearman's correlation

- **WAPs reciprocal coverage** is analysed through **Spearman's correlation**
- It assesses how well the **relationship** between two variables (i.e WAPs' RSSI) can be described using a **monotonic function**
- Correlation **ρ > 0** ➡ the RSSI *Y* tends to increase when the RSSI *X* increases
- Correlation **ρ < 0** ➡ the RSSI *Y* tends to decrease when the RSSI *X* increases
- Correlation **ρ = 0** ➡ the RSSI *Y* is not correlated with the RSSI *X*
- The correlation ρ is associated with a **confidence *1 - p-value*** according to which the null hypothesis (i.e. two WAPs are not correlated) can be rejected.

Spearman vs Pearson Correlations



A Spearman correlation of 1 results when the two variables are monotonically related, even if their relationship is not linear



The Spearman correlation is less sensitive than the Pearson correlation to strong outliers

WAPs coverage analysis

- Main idea: if two WAPs (i.e. their RSSI) are correlated then they have a similar coverage
- The Spearman correlation is computed **pairwise** between all the WAPs
- For each pair of WAPs, only those records where **at least the RSSI of one** WAP is **not null** are taken, to deal with high data sparsity and reduce correlation
- For each WAP, it is **counted** the number of times in which it results positively correlated with another WAP with a confidence of 99%

WAPs which correlate with at least other 50 WAPs (63)

Thank you for your attention