Indoor Positioning

Accuracy of a single position estimate for kNN-based fingerprinting indoor positioning applying error propagation theory

Introduction

IEEE Sensors Council

IEEE SENSORS JOURNAL, VOL. XX, NO. XX, XXXX 2017

Accuracy of a single position estimate for kNN-based fingerprinting indoor positioning applying error propagation theory

Antoni Pérez-Navarro, Raúl Montoliu, Emilio Sansano-Sansano,
Marina Martínez-Garcia, Ruben Femenia and Joaquín Torres-Sospedra

Abstract—Indoor Positioning Systems usually consider the average positioning error over a set of evaluation samples, or a quartile of that value, as the global error. However, they do not provide a metric for the uncertainty for each individual position estimation. In this paper, we apply the error propagation theory to the kNN algorithm in Wi-Fi fingerprint-based indoor positioning. Our proposed method does not only retrieve the position estimate but also describes how the uncertainties of the RSSI measurements propagate through the calculations. We have validated our proposed method with two open-access datasets.

Index Terms—Indoor positioning, Fingerprinting methods, Error propagation.

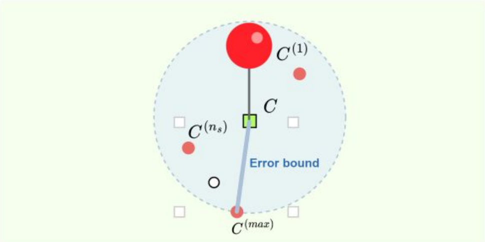


Table of contents

01

About me

About me

01

CFGS

Superior technician in application development

02

Internships

Internship in the GIANT-UJI research group

03

Computational Mathematics

Current student

04

University Scholarship

Scholarship of the University Jaume I

02

Introduction

Introduction to indoor positioning systems

Introduction

Purpose

Positioning and monitoring of people indoors such as:

- Passenger **guidance** at airports
- The **orientation** of visually impaired people
- Support of event management
- Robotics
- etc

Methods

We will use the **kNN** algorithm together with the **error bound** to estimate the actual location. The **error bound** is important because it is a **metric** that represents how reliable is the provided position.

Classic problems

Fluctuations and interferences.

03

Data Structure

What kind of data do we have?

Coordinates and RSSI measures

Coordinates

-25.3, -10.75, 0.0

.

.

.

-17.5, -10.75, 0.0

RSSI values

-53.063636, -55.400000, ...,
-71.590909, -77.663636, ...

.

.

.

-55.436364, -56.800000, ...,
-79.672727, -62.309091, ...

$$\lambda_i^{tr} = \left\{\rho_{i,1}^{tr}, ..., \rho_{i,n_{ap}}^{tr}\right\},\ i \in [1, ..., n_{tr}]$$

Better capture signal behavior at a position

RSSI values received from an AP can **vary** greatly even when the device used to capture the data remains **stationary** in the same position during the capture process...

Multiple measurements can be obtained at each position.

In these cases, we can rewrite the fingerprints as we will show in the next slide.

$$\lambda_i^{tr} = \left\{\rho_{i,1}^{tr}, \ldots, \rho_{i,n_{ap}}^{tr}\right\},\ i \in [1, \ldots, n_{tr}]$$

Better capture signal behavior at a position

$$\lambda_i^{tr} = \left\{\rho_{i,1}^{tr}, \ldots, \rho_{i,n_{ap}}^{tr}\right\}, \; i \in [1, \ldots, n_{tr}]$$

$$\mu_{i,r}^{tr} = \frac{\sum_{m=1}^{n_m} \rho_{i,r}^{tr}(m)}{n_m}$$

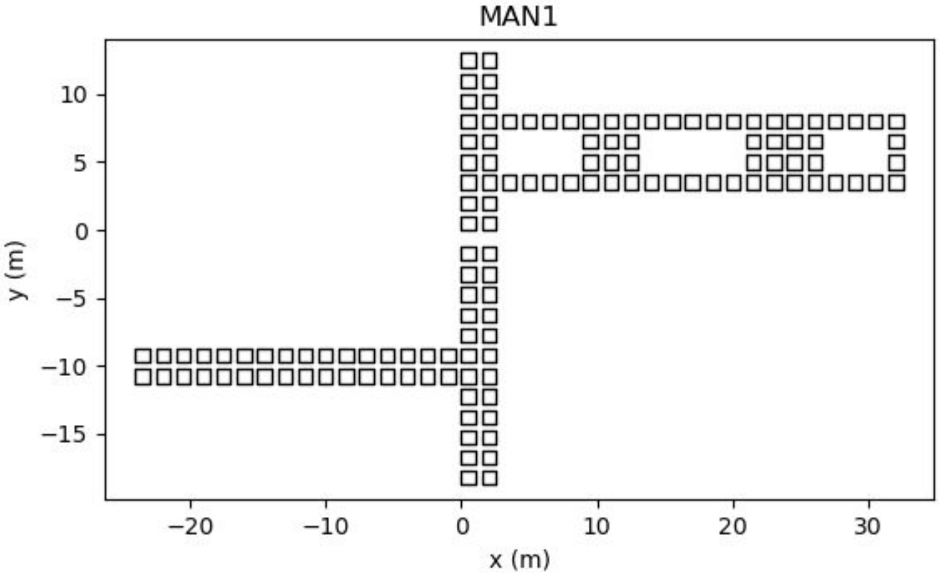
$$\sigma_{i,r}^{tr} = \sqrt{\frac{\sum_{m=1}^{n_m} (\rho_{i,r}^{tr}(m) - \mu_{i,r}^{tr})^2}{n_m - 1}}$$

Calculating the **mean** and **standard deviation** of each measurement

$$\lambda_i^{tr} = \left\{(\mu_{i,1}^{tr}, \sigma_{i,1}^{tr}), \ldots, (\mu_{i,n_{ap}}^{tr}, \sigma_{i,n_{ap}}^{tr})\right\}, \; i \in [1, \ldots, n_{tr}]$$

Data Visualization

We will work on this scenario for the rest of the presentation



Representation of the training coordinates of the MAN1 dataset

Where do we want to go?

Coordinates

-25.3, -10.75, 0.0

.

.

.

-17.5, -10.75, 0.0

Unknown

RSSI values

-53.063636, -55.400000, ...,
-71.590909, -77.663636, ...

.

.

.

-55.436364, -56.800000, ...,
-79.672727, -62.309091, ...

$$\lambda_i^{tr} = \left\{\rho_{i,1}^{tr}, ..., \rho_{i,n_{ap}}^{tr}\right\}, \; i \in [1, ..., n_{tr}]$$

04

kNN

Proposed mechanism to calculate a position using kNN

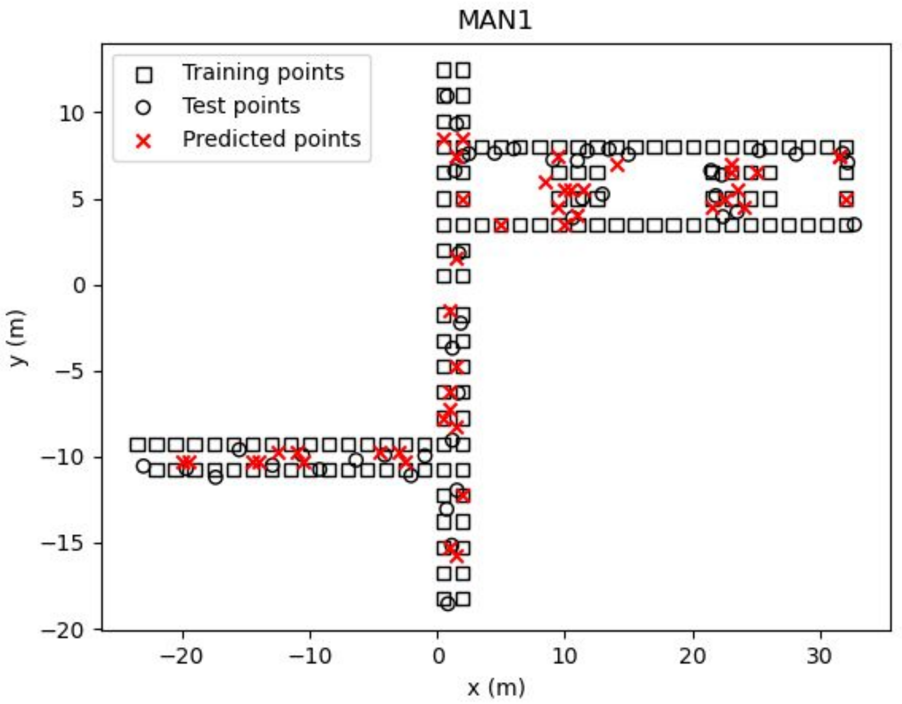
The kNN algorithm for indoor positioning

We use the **kNN** algorithm with **k = 3** to make **coordinate predictions** about where the user will be using the test set.

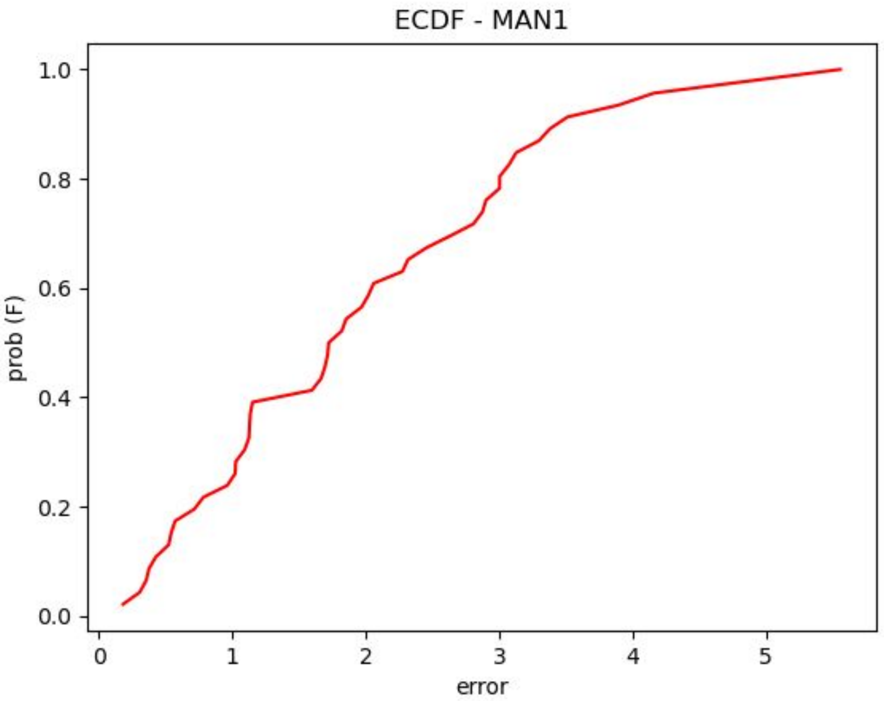
To estimate the distance used by the algorithm between the test set and the train set we use the popular **Euclidean distance**:

$$d_i = d(\lambda^{ts}, \lambda_i^{tr}) = \sqrt{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2}$$

Algorithm results



Cumulative distribution function *F(x)*



The **median** would be about **2 meters** approximately

05

Error bound

Estimate the error bound associated to a position

Error propagation

To take into account how uncertainty propagates through calculations, we use **EPT** (Error Propagation Theory).

We consider a magnitude ***z*** that is given by a function ***f*** (f is supposed to have a **Gaussian** distribution) that depends on n independent variables:

$$z = f(x_1, \ldots, x_e, \ldots, x_{n_e})$$

The **error of the z function** will be given by:

$$(\Delta z)^2 = \sum_{e=1}^{n_e} \left(\frac{\partial f}{\partial x_e} \Delta x_e \right)^2$$

Where Δx_e is the error obtained when measuring the variable x_e.

Estimating the error in the fingerprint distance

* Standard deviation of these measurements are used as the error when measuring these variables.

f → Distance metric used in the kNN algorithm $d(\lambda^{ts}, \lambda_i^{tr})$

x_e → The RSSI obtained at each location expressed by their mean values.

Applying EPT to the Euclidean distance, we obtain: →

$$(\Delta d_i)^2 = (\Delta d(\lambda^{ts}, \lambda_i^{tr}))^2 = \\ = \frac{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2 \left[(\sigma_r^{ts})^2 + (\sigma_{i,r}^{ts})^2\right]}{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2}$$

Proposed method to estimate the error bound

To estimate the error bound, we extract ns distance samples among the candidates inside the interval

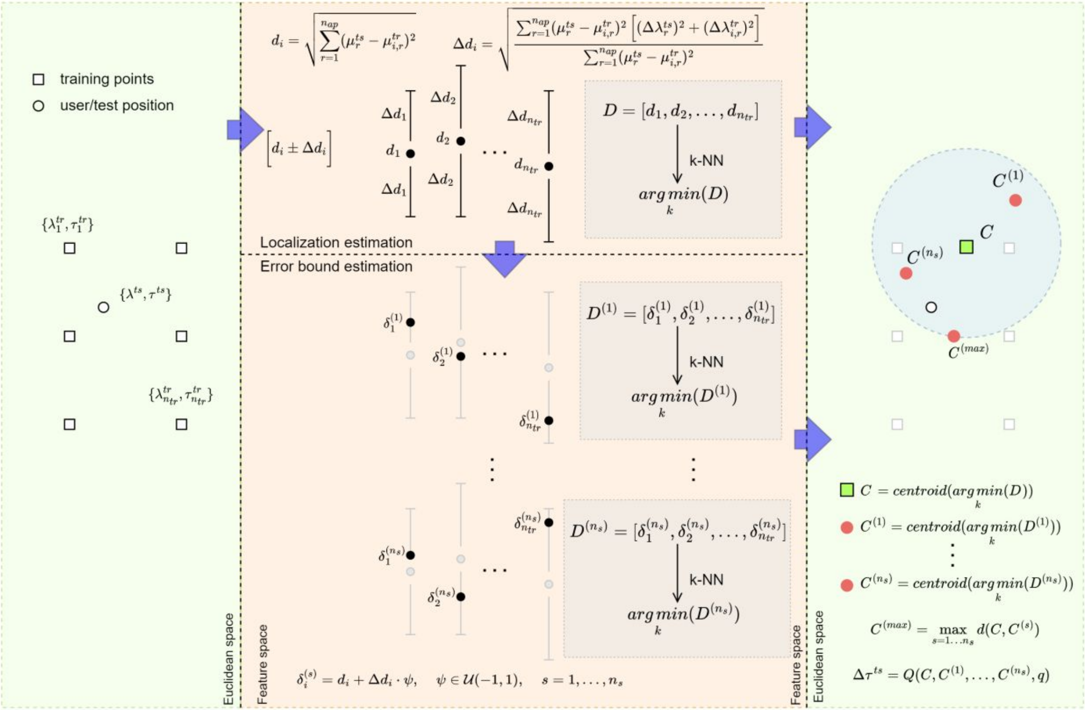
$$[d_i - \Delta d_i, d_i + \Delta d_i]$$

Each sample are called $\delta_i^{(s)}$ and can be calculated as follows:

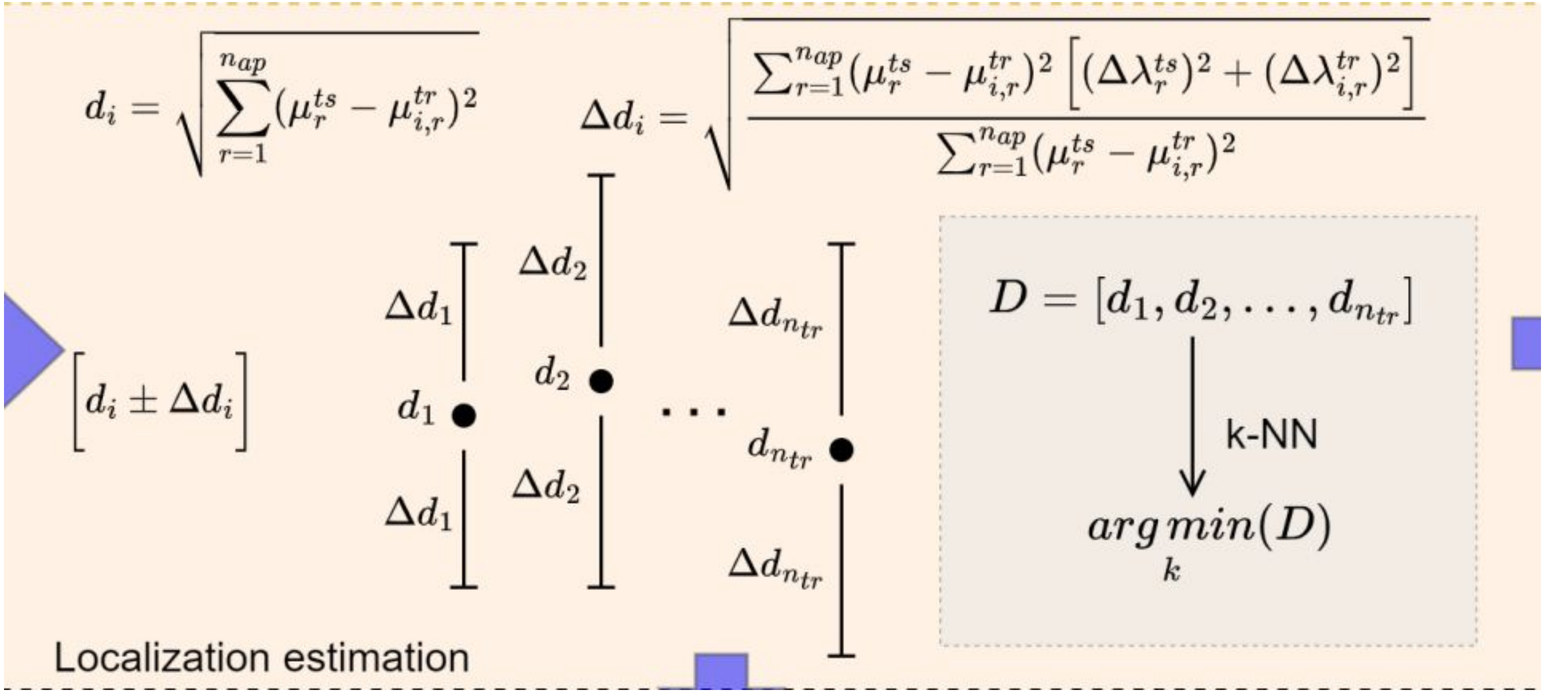
$$\delta_i^{(s)} = d_i + \Delta d_i \cdot \psi$$

Where ψ gives a number between -1 and 1 using a **uniform distribution**.

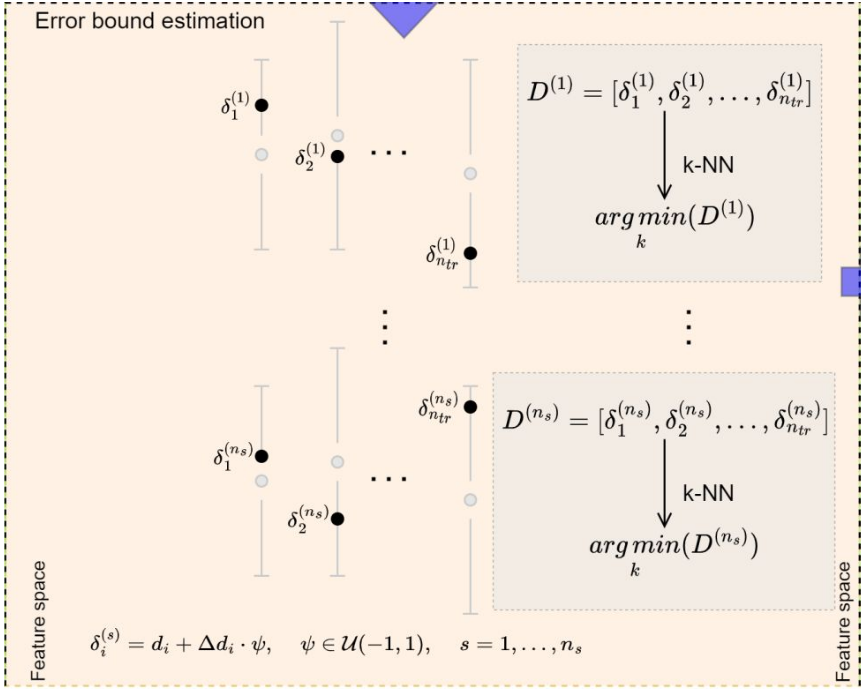
Proposed method to estimate the error bound



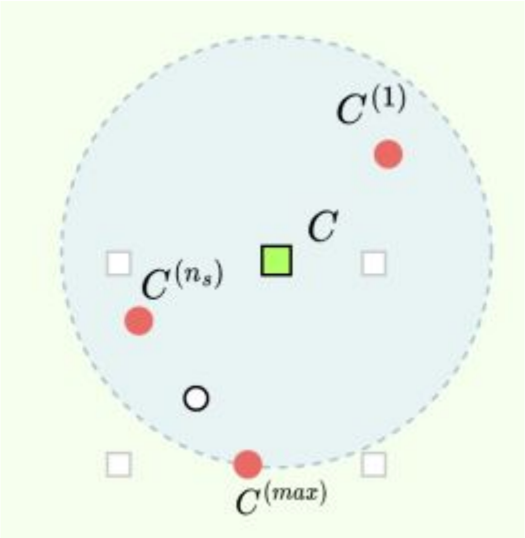
Proposed method to estimate the error bound



Proposed method to estimate the error bound

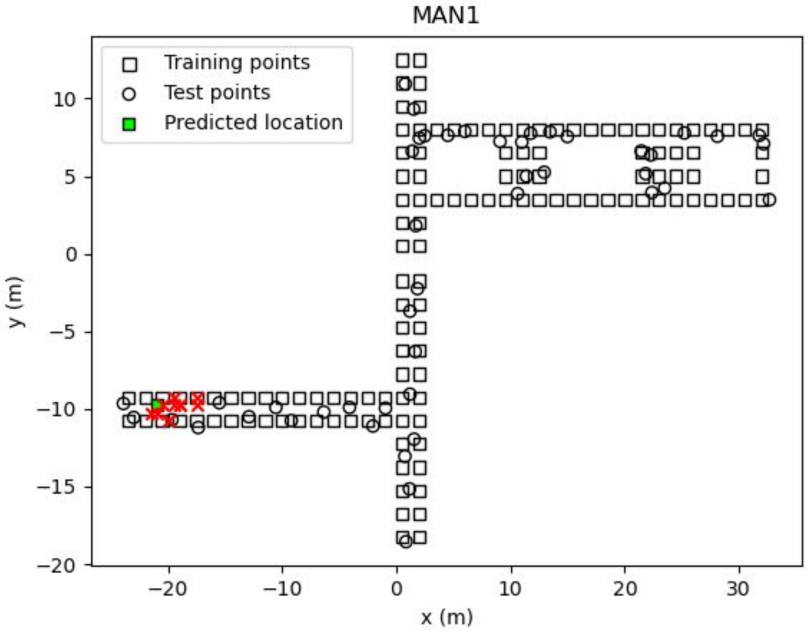


Proposed method to estimate the error bound



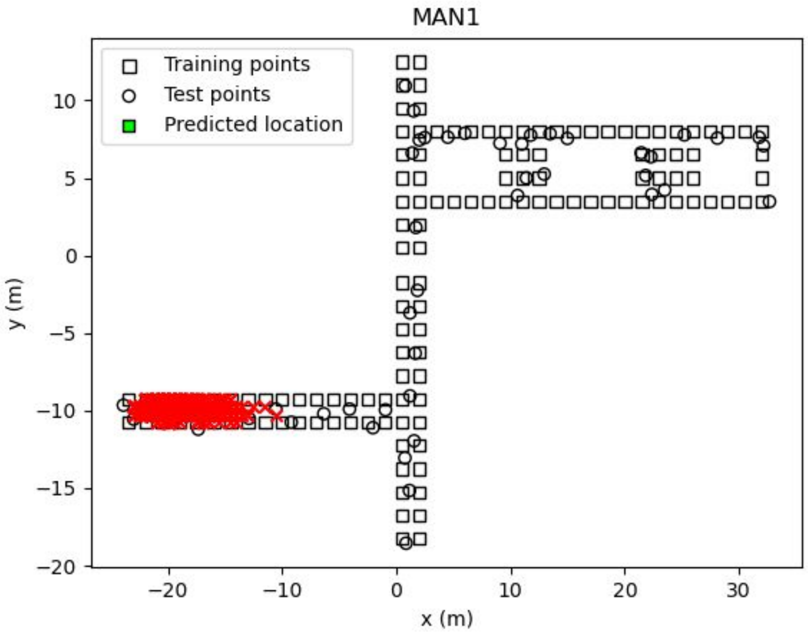
Samples among the candidates inside the interval

20 or 30 samples approx. for only one point

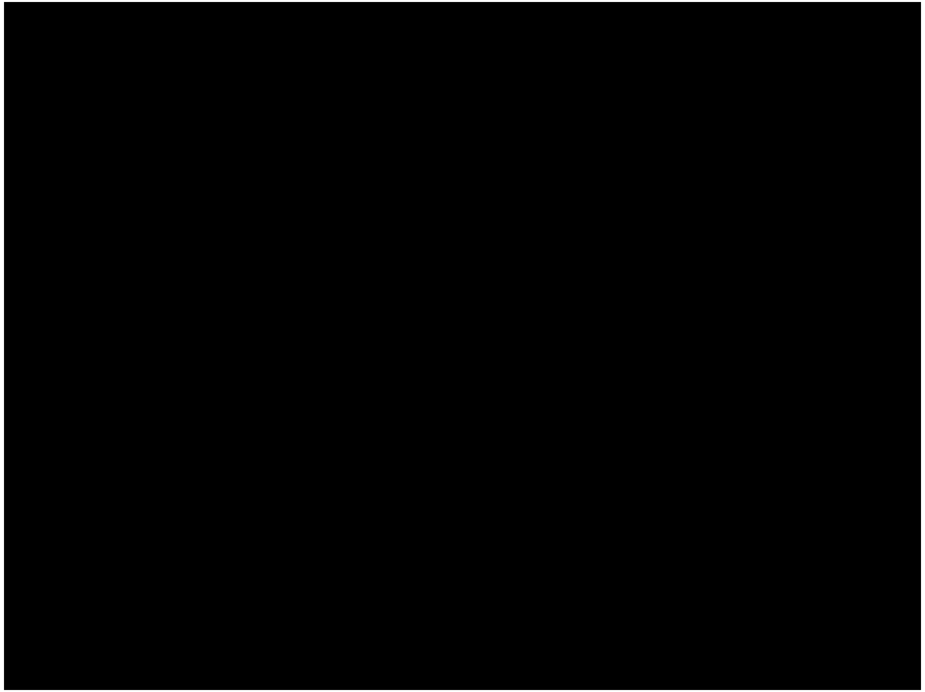


Samples among the candidates inside the interval

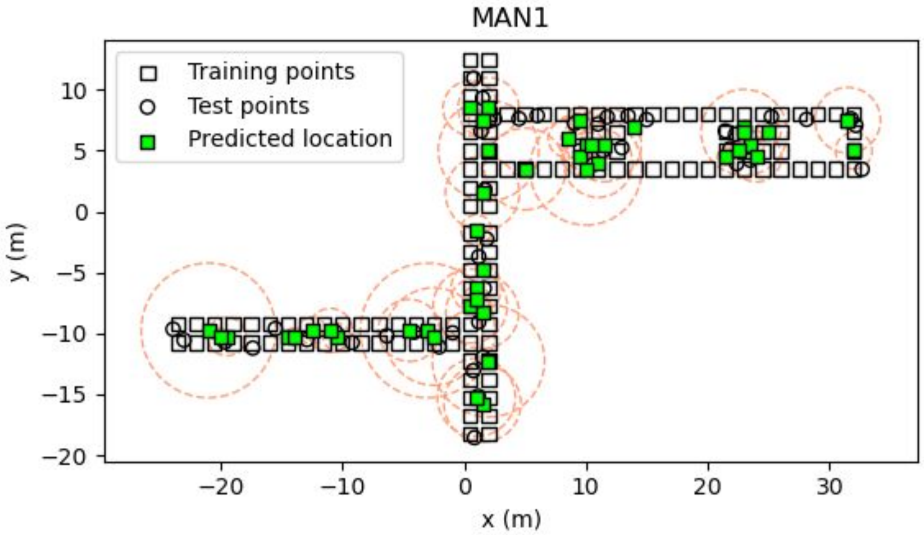
1k samples approx. for only one point



How to create the error bound circle?



Error bound



Not all are displayed

06

Future work

Adapt the shape of the bound error to the environment

Future work

It can also be of interest to study how to adapt the shape of the bound error to the environment, exploring non-symmetric ways of **characterizing the error bound**, such as **ellipses**, that reflect both the uncertainties and the **spatial distribution** of the training set.

- Finding walls
- Find objects
- etc

Thanks!

Do you have any questions?

al428586@uji.es

https://github.com/crisvalab

https://www.linkedin.com/in/cristianvaleroabundio/