Noisy Data

By Sophia Seitz
& Claire Diehl

Computational Bayesian - 2014

The Question

The output of a sensor, in our case an infrared range finder, is often not as accurate as we would like it to be. In previous classes, we many methods to account for this like averaging many points together. This does not work all that well, and we still wonder whether or not there is a better way, or whether we can do this in a Bayesian way.

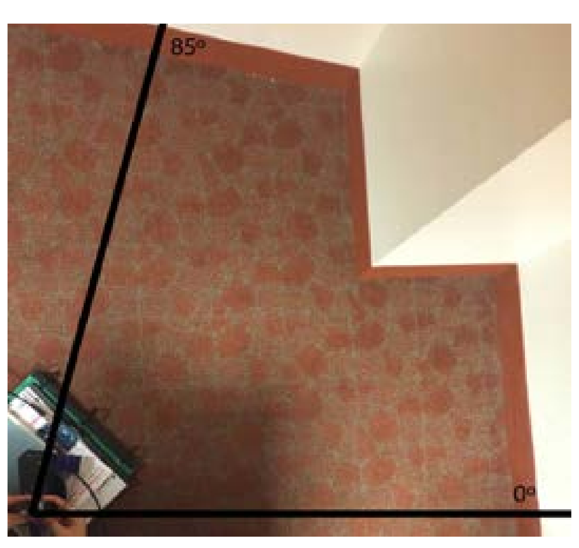
We take inspiration from another Bayesian problem, one that uses altered GPS data to better characterize the actual position of a person. After solving this problem, we wanted to see whether this solution would also work to characterize the actual location of a perceived object.

The System

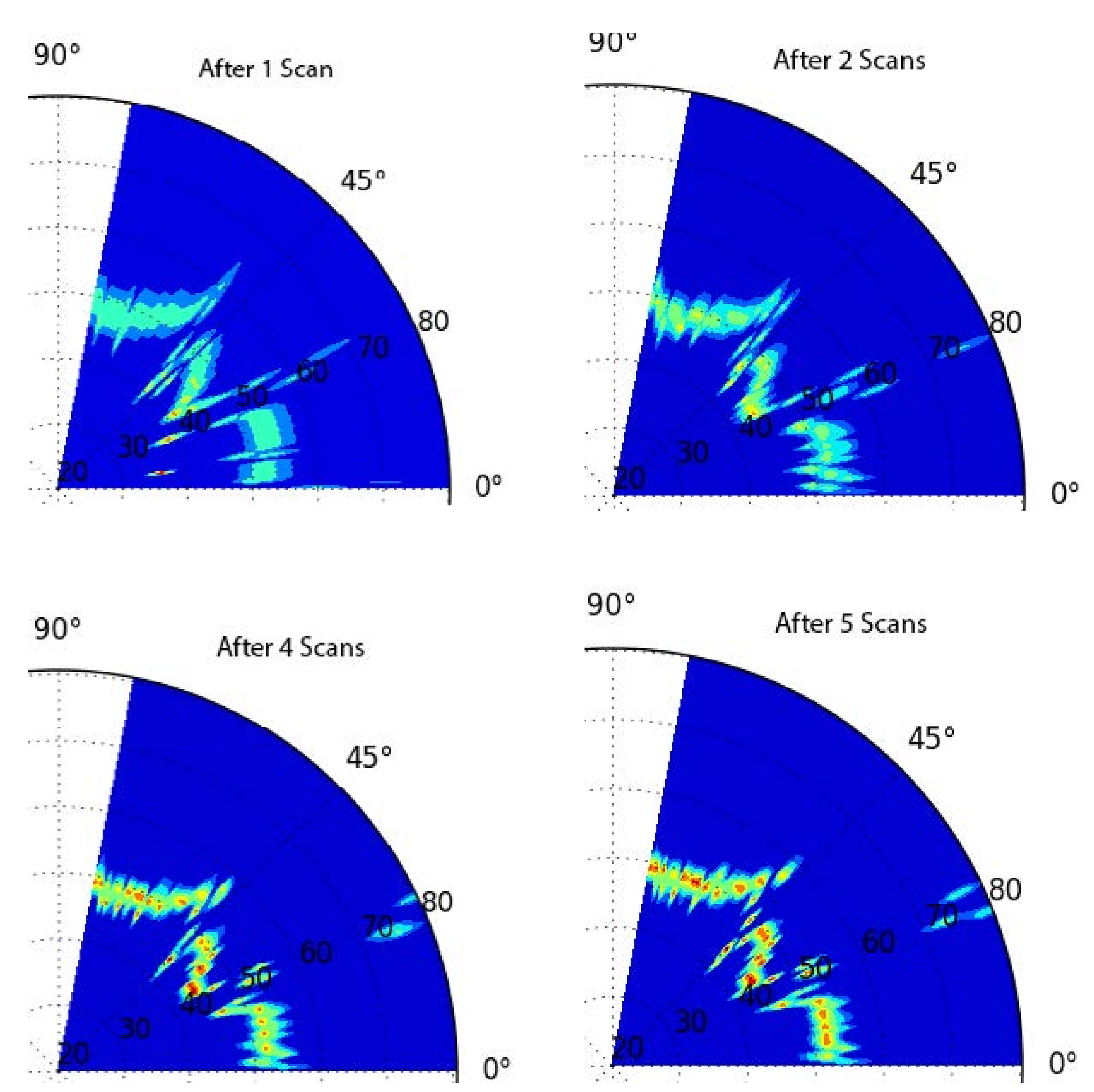
Sensor Setup

For this case study, we use a range finder mounted to a servo that sends range readings to an arduino. The arduino then communicates the data to python via the serial port. From there, we take the data and update our hypothesis about where there might be something and then finally, we plot a representation of where there are probably objects.

To get the actual sensor data, we mounted an infrared range finder to a servo. We then rotate the servo through a range of angles, collect the output of the range finder and the angle, and pass it to our python code running the Bayesian update.



The testing setup



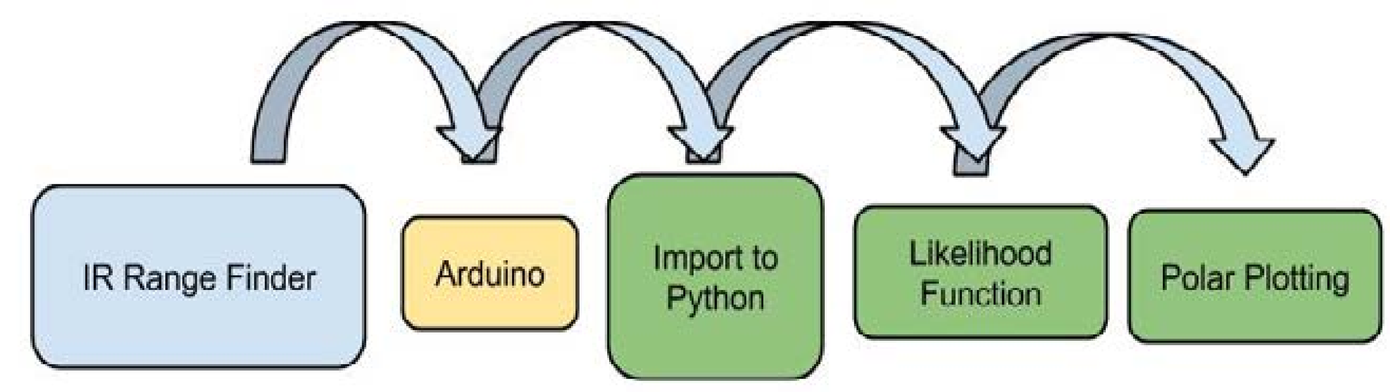
Our scans of a set of corners in an East Hall team room

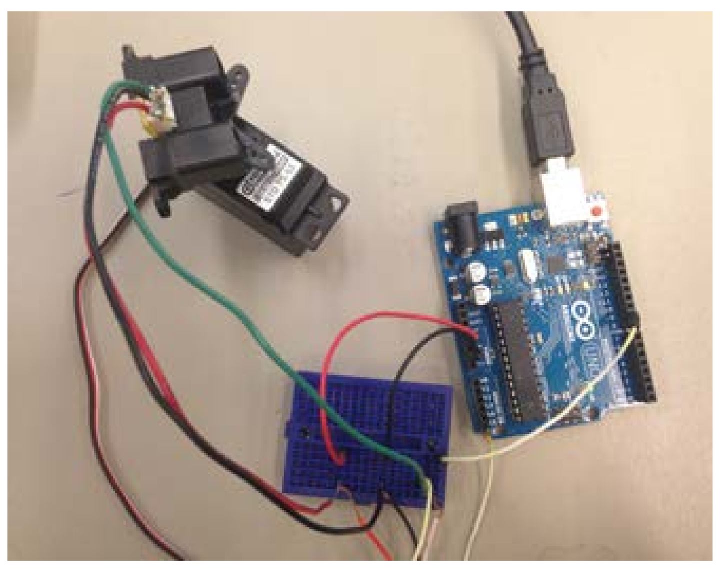
Characterizing Error

We characterize the error for the data by calculating the standard deviation and mean of the error at a series of distance points. Here, we assume that the error is normally distributed. This allows us to create a error function based on distance. We use this to calculate potential errors and likelihood of the data.

For this case study, we are representing all of our data and hypothesis as radius, angle pairs. It is also worth noting that we assume that there is no error in the angle of the sensor.

System





The testing system we used to collect the data

The Bayesian

Hypotheses

In this case study, we have a separate suite of hypotheses for each angle. Within each suite our hypotheses correspond to the distance at which there might be an object.

Quick segue back to non-Bayesian things. The data we pass into our likelihood function is all the distances for one sweep of the sensor. This data is of the form of a large tuple containing (85) smaller tuples, where each of those consists of a measured distance and angle. As a note, the sensor does not respond well to distances that are outside its maximum range and often returns “Not a Number”, or “NaN”. When this happens, we simply set the measured distance to the maximum distance.

Because we have a suite of hypotheses for each angle, we have to iterate through our data, and choose the appropriate angle suite to update for each piece of data. We use the error that we characterized above to compute the likelihood that we get a given measurement if we have an object at our hypothesis.

Our Interpretation

Bayesian is the best!

Maybe. We do not compare this data to another method of calculating the probable location of the wall and are not completely convinced that this solution is better than an average of multiple scans or even just the raw data. However, in a short number of scans we can relatively accurately map a difficult set of corners with decent probability. This proves that Bayesian is, at very least, a usable solution to this question.

References

Our repo:

{https://github.com/phiaseitz/CompBayesCaseStudy

Reference for arduino control of IR sensor and motor, as well as communication with python:

https://github.com/JPWS2013/PoE_F2013/tree/master/Lab_1