Competition II: CS4786/5786-Machine Learning for Data Science (Fall 2017)

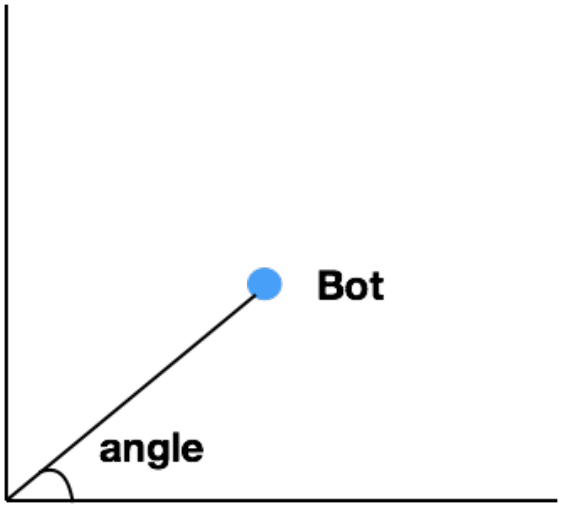
Find the bot Challenge

In-Class Kaggle competition for CS4786: Machine Learning for Data Science

Fall 2017

The second in-class Kaggle competition for our class:

This competition is based on the find the bot example we covered in class for HMM's. Only this is a continuous version. A bot moves around in a 2D plane following some probabilistic pattern unknown to you. You don't observe this bot's location on every time-step. What you do observe is the angle of the box to x axis at every time step (see figure below).



On every run, we place the bot at its starting location (fixed at same starting location for all runs) and let it run for 1000 +1 steps. We perform 10000 runs each with 1000+1 steps.

You are given observations for 10000 runs of the angle observed at each step. You are additionally provided exact location of the bot at some random time steps on every round for the first 6000 rounds only.

Your goal: predict the final location of the bot at the 1001'th step for rounds 6001 to 10000

Here is what you are provided with

- The **Observations of the bots**: You are given a 10000x1000 matrix where each row of the matrix is observations made in one run (for runs from 1 to 10000). That is, row 20 column 40 specifies, the angle of the bot to X axis on the 20th run and 40th step. This data is provided to you in the **O bervations.csv** file in the comma separated values format.
- A few **labeled example**: You are also provided the location of the bot on some random subset of steps on every run. **Label.csv** consists of 600000 example locations. It consists of 600000 rows and 4 columns. Each row is one example location of the bot.
    - For instance, a row in Label.csv of form "201,333,1.2,-0.8" means that on run 201, step 333, the bot was at location (1,2,-0.8).
    - Label.csv only has locations for runs from 1 to 6000

Task: For each of the remaining 4000 runs (from 6001 to 10000), predict location of the bot at step 1001. The competition will be hosted on in-class-Kaggle. Kaggle will be intialized soon.

For this competition, unlike competition 1, you can use the Label.csv to evaluate your solutions by yourself without requiring Kaggle to check how good your method is. Hence Kaggle can be used sparsely and you can use label.csv to develop your method.

Kaggle Link:

https://www.kaggle.com/t/a159a375ad704dba8a233abd2340f729

Download the data below as zip file. When unzipped you will find the two files, obervations.csv and labels.csv

http://www.cs.cornell.edu/courses/cs4786/2017fa/code/fa2017_comp_2.zip

Group size: Group of size 1-4 students.

Due date The deadline is **11:59 pm, Thursday, 30th November**. The due date for the report on CMS will be announced soon and is on 2nd Dec. Submit what you have at least once by an hour before that deadline, even if you haven't quite added all the finishing touches — CMS allows resubmissions up to, but not after, the deadline. If there is an emergency such that you need an extension, contact the professors.

> 1. Footnote: *The choice of the number "four" is intended to reflect the idea of allowing collaboration, but requiring that all group members be able to fit "all together at the whiteboard", and thus all be participating equally at all times. (Admittedly, it will be a tight squeeze around a laptop, but please try.)*

Deliverables:

1. **Report:** In the end of the competition each group should submit a 5-15 page writeup that includes visualization, clear explanation of methods etc. See grading guidelines for details about what is expected from the writeup. **(worth 50% of the competition grade)**
2. **Predictions:** Competition is held on Kaggle in-class as a competition. You can submit your predictions to kaggle to compete with your friends. You should also submit your predictions on CMS. **(worth 50% of the competition grade)**
3. **Code:** Submit the code you used for kaggle as a zip file.

Collaboration and academic integrity policy

Students may discuss and exchange ideas with students only within their group.

We distinguish between "merely" violating the rules for a given assignment and violating academic integrity. To violate the latter is to commit fraud by claiming credit for someone else's work. For this assignment, an example of the former would be getting detailed feedback on your approach from person X who is not in your group but stating in your homework that X was the source of that particular answer. You would cross the line into fraud if you did not mention X. The worst-case outcome for the former is a grade penalty; the worst-case scenario in the latter is academic-integrity hearing procedures.

The way to avoid violating academic integrity is to always document any portions of work you submit that are due to or influenced by other sources, even if those sources weren't permitted by the rules.[2]

> *2. Footnote: We make an exception for sources that can be taken for granted in the instructional setting, namely, the course materials. To minimize documentation effort, we also do not expect you to credit the course staff for ideas you get from them, although it's nice to do so anyway.*

Grading Guidelines:

Grading:

- Clear explanation of your main model (20 points)
  - Explain any preprocessing you did, explain clearly what your model takes as input
  - Explain clearly what algorithm you used to train and not just the model
- How does your model fit the problem description, how does it deal with the continuous location case? (15 points)
- How were parameters chosen in a principled fashion? (10 points)
- Failed attempts. (Have a clear flow of your reasoning for why you tried various models and how their failure guided you to pick next one) Give clear comparison of things you tried. Dont go for numbers but rather clear progression of thought and how each model guided the next. (15 points)
- Visualization (what did you learn from them and how they guided you). This includes tables, plots, graphs etc. (10 points)

- Supervision: How did you use the labeled examples given in your model. Did you use these to minimize kaggle submissions? (10 points)
- Unlabeled examples: How were the unlabeled data points part of you model (10 points)
- Understanding data, what did you learn from the observations and how was it used in your approach? (10 points)

Bonus (at the discretion of the graders):

- Tried new or more methods not necessarily covered in class
- Developed new algorithm or methods, tweaked existing methods to fit the problem better