

SEARCH



RESOURCES

CONCEPTS

- ✓ 1. Introduction and Overview
- ✓ 2. I/O Recap
- ✓ 3. Model-Based vs Data-Driven A...
- ✓ 4. Which is Best?
- ✓ 5. Data Driven Example - Trajecto...
- ✓ 6. Trajectory Clustering 2 - Online...
- ✓ 7. Thinking about Model Based A...
- ✓ 8. Frenet Coordinates
- ✓ 9. Process Models
- ✓ 10. More on Process Models
- ✓ 11. Multimodal Estimation
- ✓ 12. Summary of Data Driven and ...
- ✓ 13. Overview of Hybrid Approach...
- ✓ 14. Intro to Naive Bayes
- ✓ 15. Naive Bayes Quiz
- ✓ 16. Implement Naive Bayes C++

17. Implement Naive Bayes C++ (...)

18. Conclusion

Knowledge

Get learning questions answered

Student Hub

Chat with peers and mentors

2. Implement the `predict(observation)` method in `classifier.cpp`.

Given a new data point, prediction requires two steps:

1. **Compute the conditional probabilities for each feature/label combination** and label C with mean μ and standard deviation σ (computed in training). The probability can be computed using the formula [here](#):

$$p(x = v|C) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(v-\mu)^2}{2\sigma^2}}$$

Here v is the value of feature x in the new data point.

2. **Use the conditional probabilities in a Naive Bayes classifier.** This can be computed using the formula [here](#):

$$y = \underset{k \in (1, \dots, K)}{\operatorname{argmax}} p(C_k) \prod_{i=1}^n p(x_i = v_i | C_k)$$

In this formula, the argmax is taken over all possible labels C_k and the all features x_i with values v_i .

3. When you want to test your classifier, run `Test Run` and check out the results.

NOTE: You are welcome to use some existing implementation of a Gaussian Naive Bayes classifier. However, to get the **best** results you will still need to put some thought into what **features** you use when classifying. Though you will only be given the 4 coordinates listed that by "engineering" features you may get better performance. For example: the coordinate may not be that useful. But `d % lane_width` might be helpful since it gives the position of a vehicle in its lane regardless of which lane the vehicle is in.

Helpful Resources

- [sklearn documentation on GaussianNB](#)
- [Wikipedia article on Naive Bayes / GNB](#)

Extra Practice

Provided in one of the links below is `python_extra_practice`, which is the same code written in Python that you can optionally go through for extra coding practice. The code is available at the `python_solution` link. If you get stuck on the quiz see if you can write a solution to C++ and pass the classroom quiz with it. The last link `Nd013_Pred_Data` contains training and testing data for this problem in case you want to run the problem off

```

main.cpp  classifier.cpp  classifier.h
1  #include <fstream>
2  #include <iostream>
3  #include <string>
4  #include <vector>
5  #include "classifier.h"
6
7  using std::cout;
8  using std::endl;
9  using std::ifstream;
10 using std::string;
11 using std::vector;
12
13 // Helper functions to load .txt files
14 vector<vector<double>> Load_State(string file_name);

```