
CS589: Machine Learning - Fall 2019

Homework 4: Bayesian Prediction

Assigned: Apr 19th, 2019 Due: Apr 25th, 2019

Getting Started: In this assignment, you will train and evaluate making Bayesian predictions in a simple toy model. Please install Python 3.6 via Anaconda on your personal machine. For this homework you may only use **numpy**, **scipy.stats.norm** (optional) and **matplotlib** packages. In particular you may not use any part of sklearn. Download the homework file HW04.zip via Moodle. Unzipping this folder will create the directory structure shown below,

HW04

```
--- HW04.pdf
--- Data
--- Submission
    |--Code
    |--Figures
```

The data files for each data set are in 'Data' directory respectively. You will write your code under the Submission/Code directory. Make sure to put the deliverables (explained below) into the respective directories.

Deliverables: This assignment has two types of deliverables:

- **Report:** The solution report will give your answers to the homework questions (listed below). Try to keep the maximum length of the report to 5 pages in 11 point font, including all figures and tables. Reports longer than five pages will only be graded up until the first five pages. You can use any software to create your report, but your report must be submitted in PDF format. You can use an additional two pages to answer extra credit questions only.
- **Code:** The second deliverable is the code that you wrote to answer the questions, which will involve implementing kernel methods. Your code must be Python 3.6 (no iPython notebooks or other formats). You may create any additional source files to structure your code. However, you should aim to write your code so that it is possible to re-produce all of your experimental results exactly by running *python run_me.py* file from the Submissions/Code directory.

Submitting Deliverables: When you complete the assignment, you will upload your report and your code using the Gradescope.com service. Place your final code in Submission/Code. If you used Python to generate report figures, place them in Submission/Figures. Finally, create a zip file of your submission directory, Submission.zip (NO rar, tar or other formats). Upload this single zip file on Gradescope as your solution to the 'HW04-Bayesian-Programming' assignment. Gradescope will run checks to determine if your submission contains the required files in the correct locations. Finally, upload your pdf report to the 'HW04-Bayesian-Report' assignment. When you upload your report please make sure to select the correct pages for each question respectively. Failure to select the correct pages will result in point deductions. The submission time for your assignment is considered to be the later of the submission timestamps of your code and report

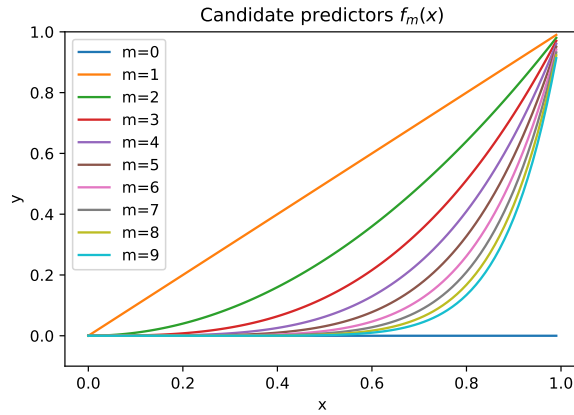
submissions.

Academic Honesty Statement: Copying solutions from external sources (books, internet, etc.) or other students is considered cheating. Sharing your solutions with other students is also considered cheating. Posting your code to public repositories like GitHub, stackoverflow is also considered cheating. Any detected cheating will result in a grade of -100% on the assignment for all students involved, and potentially a grade of F in the course.

Task: In this homework you will experiment with making Bayesian predictions in a simple toy model. The model contains inputs x in 1-d and outputs y in 1-d. The possible predictors are the functions

$$f_m(x) = \begin{cases} 0 & m = 0 \\ x^m & m > 0 \end{cases}.$$

These are plotted below.



We take the following prior over $m \in \{0, 1, \dots, 9\}$:

m	0	1	2	3	4	5	6	7	8	9
$p(m)$.2	.2	.2	.1	.1	.1	.025	.025	.025	.025

We take the following probabilistic model, with $\sigma = .1$ fixed:

$$p(y|x, m) = \mathcal{N}(y|\mu = f_m(x), \sigma^2).$$

1. **Prior:** Implement a function which returns the given prior $p(m)$ for each value of $m \in \{0, \dots, 9\}$. Make a bar chart of the values (with m on the x-axis) and plot it.
2. **Likelihood:** You are given a set of 10 training inputs and outputs, stored as x.csv and y.csv. Load this data and implement a function to compute the likelihood

$$p(\text{Data}|m) = \prod_{i=1}^{10} p(y_i|x_i, m).$$

Make a bar chart of the values (with m on the x-axis) and plot it.

3. **Posterior Equation:** Give a mathematical equation for the posterior $p(m|\text{Data})$ in terms of the prior $p(m)$ and the likelihood $p(\text{Data})$. Make sure that this equation is normalized.
4. **Posterior Implementation.** Implement a function to compute the posterior $p(m|\text{Data})$. Make sure that this is normalized (sums to one). Make a bar chart of the values (with m on the x-axis) and plot it.
5. **MAP estimate:** Implement a function to compute the “MAP” estimate $\hat{m}_{\text{MAP}} = \arg \max_m p(m|\text{Data})$. What value of m do you find, and what is the posterior probability of it? (Give numbers.)
6. **MAP predictions:** You are given a set of 100 test inputs and outputs, stored as `x_test.csv`. Using your map estimate \hat{m}_{MAP} plot your predicted outputs with $f_{\hat{m}_{\text{MAP}}}(x_i)$ as circles. Then, load the 100 test outputs, stored as `y_test.csv`. Add these to the same plot, labeled with crosses. Show the plot. Make sure to label all axes, and to have a legend for the two types of markers.
7. **MAP error:** Compute the mean-squared error of your MAP estimate, $\frac{1}{100} \sum_{i=1}^{100} (y_i - f_{\hat{m}_{\text{MAP}}}(x_i))^2$, where x and y represent test data. Give the final error as a number.
8. **Bayes estimate:** Implement a function to compute the Bayes predictions

$$f_{\text{Bayes}}(x) = \sum_{m=0}^9 p(m|\text{Data}) f_m(x).$$

Please note, $p(m|\text{Data})$ is the posterior you computed in (4). As in step (6) above, make a plot comparing these to the true outputs in the test data.

9. **Bayes error:** Compute the mean-squared error of your Bayes estimate, $\frac{1}{100} \sum_{i=1}^{100} (y_i - f_{\text{Bayes}}(x_i))^2$, where $x_i, y_i \in \text{test_data}$. Give the final error as a number.
10. **Thoughts:** Is the Bayes or the MAP error lower? Given your plot of the posterior from step (4), can you reason informally about why this might be the case?