5 Probabilistic Learning

Course: COMPSCI 589 Machine Learning, Spring, 2021

Instructor: Justin Domke

Assignment: 5

Group work policy: You are allowed to complete this homework in teams of at most 3 students. However, each student must submit their own individual .pdf and .zip files to Gradescope. List your team members at the beginning of your `report.pdf`. You may verbally discuss the assignment with course staff or students outside your group. Please also list any students or course staff (separately) at the beginning of your `report.pdf`. However, you may not *look, copy, or show* any part of another student's assignment. Copying any part of another assignment — even a single sentence or line of code — from anyone outside your team is considered plagiarism. We use sophisticated tools to detect this. Please do not do it.

Due date: April 28, 2021, 5:00 PM

Submission instructions:

- For this assignment, you should prepare your solutions in one of three formats:
  - Latex (any style)
  - Markdown
  - Jupyter notebook
- Regardless of how you prepared the solutions, you should export a single .pdf file that you upload to Gradescope. The .pdf should be submitted to Assignment 5: Probabalistic Learning. Most coding question will ask you to include your code as text in the solution .pdf.

- Additionally, you **must submit a .zip file** to Assignment 5: ZIP File in Gradescope. Your .zip file should contain four things:
  - `report.pdf` - Your report.
  - `report_src/` - A directory containing all source files for the report.
  - `code/` - A directory containing Python code for all parts of the assignment.
  - `code/run_me.py` - A single Python file that will generate all figures included in your report.
- If you use a Jupyter notebook, nothing changes. You still must put your code in external files, and you still must submit both a single .pdf and a .zip file containing the above components to the respective assignments in Gradescope.
- When you submit the .pdf to Gradescope, you must must mark page numbers for the different questions. We hate to do it, but we will penalize anyone who does not do this, as it creates a huge amount of difficulty for the graders.
- For the purpose of late days, the later of your two submissions will be considered the submission time for your assignment. E.g., if you submit your .pdf on time, but the .zip is two days late, the assignment will be considered two days late.

Code:

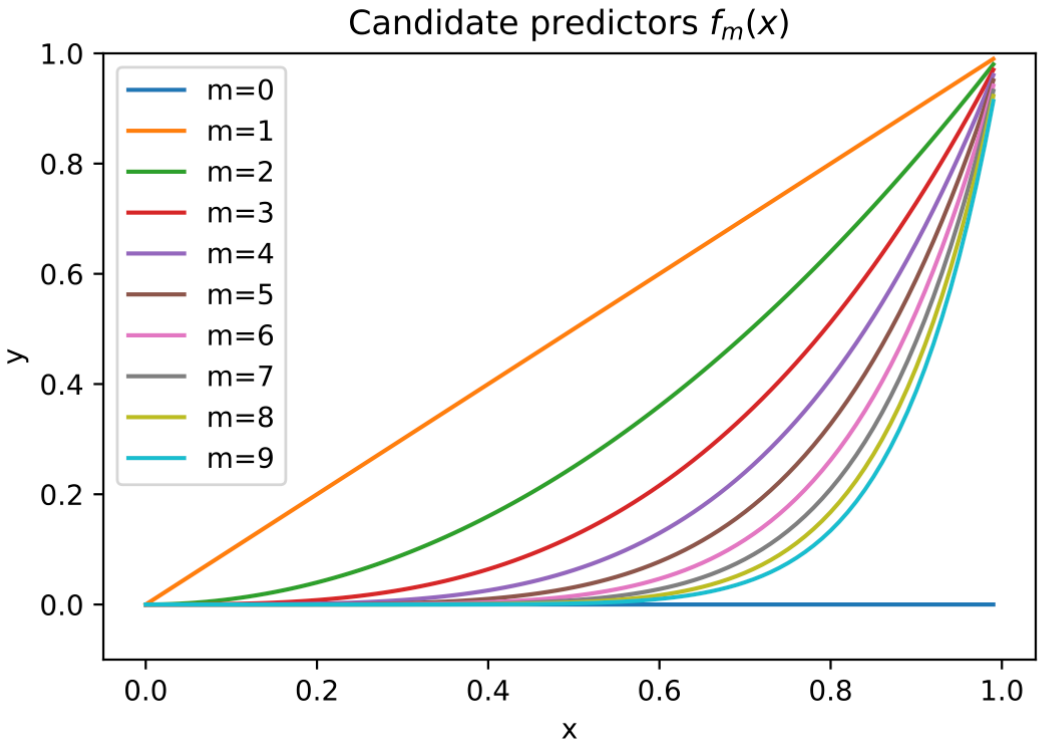
- For this assignment you may **NOT** use `sklearn`.
- You may not use any external libraries except `numpy`, `matplotlib`, and `scipy.stats`.

2

In this assignment you will experiment with making Bayesian predictions in a simple toy model. The inputs x and outputs y are both 1D. The possible predictors are the functions

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

This function is plotted below:



We take the following prior over m:

$$p(m) = \begin{cases} .2 & m = 0 \\ .2 & m = 1 \\ .2 & m = 2 \\ .1 & m = 3 \\ .1 & m = 4 \\ .05 & m = 5 \\ .05 & m = 6 \\ .05 & m = 7 \\ .025 & m = 8 \\ .025 & m = 9 \end{cases}.$$

The observation model is

6

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

where $\sigma = .1$ is fixed and $\mathcal{N}(y|\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2.

Question 1 (5 points) Implement a function which returns the prior $p(m)$ for each value of $m \in \{0, 1, \cdots, 9\}$. Your function should have the following signature: 1

```
def prior(m): # do stuff return p
```

Give your function directly in your solutions.

Question 2 (5 points) Run the following code, using your prior from the previous question

```
for m in range(10): print("m", m, "prior(m)", prior(m))
```

Give the output directly in your solutions. Also, create a bar-chart of these values, with m on the x-axis and $p(m)$ on the y-axis. Include that chart here.

Question 3 (10 points) Implement a function that computes the likelihood $p(y|x, m)$ for single input/output pair (x, y) (both scalars). Your function should have the signature 4

```
def likelihood_single(x,y,m): # do stuff return p
```

Give your function directly in your solutions.

Question 4 (10 points) Implement a function that computes the likelihood of a dataset

5

$$p(\text{Data}|m) = \prod_{n=1}^{10} p(y^{(n)}|x^{(n)}, m).$$

Your function should have the following signature:

```
def likelihood(X,Y,m): # do stuff return p
```

Here `X` is a 1D array of input values, `Y` is a 1D array of output values, and `m` is an integer. Give your function directly in your solutions.

x.csv 0.2KB

y.csv 0.2KB

Question 5 (5 points) You are given a set of 10 training inputs and outputs. Load these as `X` and `Y`. Then, make a bar chart of the likelihood, with m on the x-axis and the likelihood on the y-axis. 4

Question 6 (5 points) Give a mathematical equation for the posterior $p(m|\text{Data})$ in terms of the likelihood and the prior. Make sure your equation is correctly normalized. 2

Question 7 (10 points) Implement a function to compute the posterior given the data. Your function should have the following signature:

```
def posterior(X,Y,m): # do stuff return p
```

Give your function directly in your solutions. 2

Question 8 (5 points) Make a bar chart of your posterior evaluated on each value of m. Make sure your posterior is correctly normalized. 2

Question 9 (5 points) Make a function to compute the MAP estimate

$$m_{\text{MAP}} = \arg\max_m p(m|\text{Data}).$$

Your function should have the following signature:

```
def MAP(X,Y): # do stuff return m
```

Give your function directly in your solutions.

Question 10 (5 points) What value of m do you find, and what is its posterior probability? (Give specific numbers.)

2

Question 11 (5 points) Implement a function $f_{\text{MAP}}(x) = f_{m_{\text{MAP}}}(x)$ to make predictions using your MAP estimate of m. Your function should have the following signature:

4

```
def predict_MAP(x,X,Y): # do stuff return f
```

Here, `X` and `Y` are 1D arrays containing training data, and `x` is a scalar representing a point you want to make a prediction for.

x_test.csv 2.4KB

y_test.csv 2.5KB

Question 12 (10 points) You are given a set of 100 test inputs and outputs. Using your estimate of m_{MAP} generated from the training data in `x.csv` and `y.csv`, compute $f_{\text{MAP}}(x)$ for each element of the test set. Make a plot with the test inputs values x in the x-axis, and the predictions $f_{\text{MAP}}(x)$ on the y-axis, plotted as circles. Also, plot the true output values y as crosses. Make sure to label all axes and have a legend for the markers.

Question 13 (5 points) Compute the mean-squared test error of your MAP estimate,

$$\frac{1}{100}\sum_{n=1}^{100}\left(y^{(n)} - f_{\text{MAP}}(x^{(n)})\right)^2.$$

Give this error as a number.

Question 14 (10 points) Implement a function to make Bayes predictions

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

Your function should have the following signature:

```
def predict_Bayes(x,X,Y): # do stuff return f
```

Give your function directly in your solutions.

Question 15 (10 points) Compute predictions using the Bayes predictor for each element of the test set (again, using the training data to compute the posterior). Plot these as circles, comparing against the true outputs plotted with crosses, as in the question above. 1

Question 16 (5 points) Compute the mean-squared test error of your Bayes estimate,

$$\frac{1}{100} \sum_{n=1}^{100} \left(y^{(n)} - f_{\text{Bayes}}(x^{(n)}) \right)^2.$$

Give the final error as a number.

Question 17 (5 points) Is the Bayes or MAP error lower? Given your posterior from the earlier question, can you reason informally about why this might be true?