Assignment 2

Due Nov 18th, 2024 at 11:59pm

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment.

Submitting Your Assignment

The assignment must be submitted online on Coursys. You must submit a report in **PDF format**. You may typeset your assignment in LaTeX or Word, or submit neatly handwritten and scanned solutions. We will not be able to give credit to solutions that are not legible.

In addition to the PDF report, you must submit to Coursys a zip file containing **three Jupyter notebooks for Questions 2, and 3**. **Make sure your code can run** in the collab without any changes.

---

1 Probability

Let's consider a scenario where we have a binary classification problem in machine learning, and we want to calculate the KL divergence between two probability distributions: the true distribution $P(Y)$ and the predicted distribution $Q(Y)$, where Y represents the class labels (0 or 1).

Suppose we have a dataset with 100 samples, where the true distribution P of labels is as follows:

- Class 0: 60 examples
- Class 1: 40 examples

Let's say our machine learning model makes predictions on this dataset, and the predicted probabilities Q for each class are as follows:

- Predicted probabilities for Class 0: p
- Predicted probabilities for Class 1: $1 - p$

with $0 < p < 1$.

a) Calculate the value of true distribution Entropy $H(P)$ and write down the Entropy of the predicted distribution $H(Q)$.

b) Calculate the minimum cross-entropy $H_{min}(P, Q)$ and find the corresponding probability p.

$$H_{min}(P, Q) := \min_p [-\sum_{i=0}^{1} P(Y = i) \log Q(Y = i)].$$

c) Prove that KL divergence can be found with the following relationship between the cross entropy $H(P, Q)$ and the entropy $H(P)$

$$D_{KL}(P||Q) = H(P, Q) - H(P).$$

d) What is the minimum KL divergence of the prediction $D_{KL_{min}}(P||Q)$?

2 Bayesian Inference

Consider a simple linear model $y = wx + b + \epsilon$, where $x, y, w, b \in \mathbb{R}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Assume prior information such that $w \sim \mathcal{N}(0, \sigma_w^2)$ and $b \sim \mathcal{N}(0, \sigma_b^2)$. The regularization parameters λ_w and λ_b are defined as $\lambda_w = \frac{1}{\sigma_w^2}$ and $\lambda_b = \frac{1}{\sigma_b^2}$, respectively. The training data consists of 20 data points, with code provided in this Jupyter Notebook.

(a) **Identify the prior means and covariances.**

State the prior means and covariance values for both w and b based on the given prior information.

(b) **Explain and implement the formula for $[w_{\text{MAP}}, b_{\text{MAP}}]$ (Maximum A Posteriori estimation).**

State the expression for $[w_{\text{MAP}}, b_{\text{MAP}}]$ in terms of the design matrix X (with an added column of ones for the intercept), the output vector y, and regularization terms. Provide a high-level explanation of the derivation steps. A detailed derivation is not required as it is available in the provided slides. Focus on summarizing the intuition behind the MAP estimate and implement the formula in code.

(c) **Calculate the posterior means and covariances for w and b.**

Provide expressions for the posterior mean and covariance matrix for the parameter vector $[w, b]$. For a detailed derivation and further explanation on Bayesian linear regression, see *Probabilistic Machine Learning: An Introduction* by Kevin P. Murphy, Chapter 7, Section 7.6.1, noting V_N as posterior covariance matrix.

(d) **Interpret the posterior mean and covariance.**

Explain what the posterior mean and covariance represent for w and b after observing the data, and describe the practical significance of w.

(e) **Sample models from the posterior distribution and plot the results.**

Use the posterior mean and covariance to sample multiple sets of parameters $[w, b]$ from the multivariate normal distribution. Plot the training data along with multiple linear models represented by these samples. Display these sampled models as semi-transparent lines to visualize the uncertainty in the predictions.

3 Nonlinear Optimization

In this question, we implement iterative algorithms to solve a nonlinear optimization problem. As practical application we will optimize a nonlinear model of chess win probability given ELO ratings of the players, using data from lichess.org.

Please provide the code for your answers in this Jupyter Notebook (there are total 4 "`#<<TODO#x>>`" in this question).

a) Define the objective function.

Implement your chosen objective function. You may choose between the following:

- Option (i): A simpler quadratic function:

$$f(x, y) = x^2 + \frac{(y-2)^2}{2}$$

- Option (ii): The cross-entropy loss for the sigmoid model, which better represents the chess ELO win probability data.

Implement this function and ensure the code correctly calculates the chosen loss value and its gradient wrt the two model parameters for the given data.

Note: You have the option to proceed with either one of the two objective functions. The first one is an optional, simpler fallback that might help you to get started. There is a -5% penalty on this question, if you choose not implement the cross-entropy loss option (ii).

b) Implement a basic Gradient Descent algorithm for the objective function (i) or (ii).

- Use step size of 0.2, total iterations of 100, and initial point of $(-10, 10)$ for the objective function.
- Change the step size to 0.01. Report and plot your observation.

c) Implement the Adam algorithm for the chosen objective function.

- Use step size of 0.2, total iterations of 150, and initial point of $(-10, 10)$ for the objective function. Use default values for other parameters. Does the solution converge to the minimal objective value? Report and plot your observation.

d) Finally, use the best parameters that you found via Adam (or gradient descent) to draw the best fitting sigmoid function model on top of the data points. Briefly discuss this result in comparison with the linear fit from the earlier question.