CS 274A Homework 4

Probabilistic Learning: Theory and Algorithms, CS 274A, Winter 2023

Due: 12 noon, Monday February 27th, submit via Gradescope

Instructions and Guidelines for Homeworks

- Please answer all of the questions and submit your solutions to Gradescope (either hand-written or typed are fine as long as the writing is legible).
- All problems are worth equal points (10 points) unless otherwise stated. All homeworks will get equal weight in computation of the final grade for the class (with lowest-scoring homework being dropped).
- The homeworks are intended to help you better understand the concepts we discuss in class. It is important that you solve the problems yourself to help you learn and reinforce the material from class. If you don't do the homeworks you will likely have difficulty in the exams later in the quarter.
- In problems that ask you to derive or prove a result you should submit a complete mathematical proof (i.e., each line must follow logically from the preceding one, without "hand-waving"). Be as clear as possible in explaining your notation and in stating your reasoning as you go from line to line.
- If you can't solve a problem, you can discuss the high-level concepts *verbally* with another student (e.g., what concepts from the lectures or notes or text are relevant to a problem). However, you should not discuss any of the details of a solution with another student. In particular note that you are not allowed to view (or show to any other student) *any written material* directly related to the homeworks, including other students' solutions or drafts of solutions, solutions from previous versions of this class, etc. The work you hand in should be your own original work.
- If you need to you can look up standard results/definition/identities from textbooks, class notes, textbooks, other reference material (e.g., from the Web). If you base any part of your solution on material that we did not discuss in class, or is not in the class notes, or is not a standard known result, then you may want to rovide a reference in terms of where the result is from, e.g., "based on material in Section 2.2 in ....." or a URL.

Recommended Reading for Homework 4: Class Notes on Regression; and from the online Mathematics for Machine Learning (MML) text; Chapter 7 (Optimization) pages 225-238; Chapter sections 8.1, 8.2; and Chapter sections 9.1, 9.2.

Note: For this homework its important to go through the MML readings above in particular since they will cover concepts required for some of the problems below that will not be discussed in any detail in class. Its fine to use any results from the MML text without citing them (but please cite any other sources you use in your solutions).

Problem 1: Maximum Likelihood Estimation for Linear Regression

Assume we have IID training data in the form $D = \{(x_i, y_i)\}, i = 1, \ldots, N$, where x_i and y_i are both one-dimensional and real-valued. Say we assume that y given x is a conditional Gaussian density with mean $E[y|x] = ax + b$ and with variance σ^2 (see example 8.4 in the MML text). Assume that a, b, and σ^2 are unknown.

Show from first principles that the maximum likelihood estimates for each of a, b, and σ^2 can be written as:

$$\hat{a} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\overline{x})^2}$$

$$\hat{b} = \overline{y} - \hat{a}\overline{x}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_i \big(y_i - [\hat{a}x_i + \hat{b}]\big)^2$$

where terms such as $\bar{x}, \bar{y}$ represent empirical averages over the N datapoints, and terms like $\hat{a}$ represent maximum likelihood estimates.

Problem 2: Normal Equations for Least Squares (MSE) Regression

Assume we have training data in the form $D = \{(\underline{x}_i, y_i)\}, i = 1, \ldots, N$, where each $\underline{x}_i$ is a d-dimensional real-valued vector (with one component set to the constant 1 to allow for an intercept term) and where each y_i is a real-valued scalar. Assume we wish to fit a linear model of the form $\underline{\theta}^T \underline{x}$ where $\underline{\theta}$ is a d-dimensional parameter vector, where by "fit" we mean here that we want to find $\hat{\underline{\theta}}$ that minimizes $MSE(\underline{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\big(y_i - \underline{\theta}^T \underline{x}_i\big)^2$.

1. Prove that the solution to this problem can be written as the solution of a system of d linear equations (often referred to as the "normal equations") that can be written in the form $\mathbf{A}\underline{\theta} = \underline{b}$ where $\underline{\theta}$ has dimension $d \times 1$, $\mathbf{A}$ is a $d \times d$ matrix, and $\underline{b}$ is a $d \times 1$ vector. Starting from the definition of $MSE(\underline{\theta})$ above, carefully write out all steps in your proof, and clearly show how $\mathbf{A}$ and $\underline{b}$ are defined. If you need to assume as part of your solution that a particular matrix is full rank then assume so and state that you have assumed this.

2. Define the time complexity of minimizing $MSE(\underline{\theta})$ (i.e., solving for $\arg\max_{\underline{\theta}} MSE(\underline{\theta})$) using the normal equations given a dataset $D = \{(\underline{x}_i, y_i)\}, i = 1, \ldots, N$. Time complexity is defined as being "on the order of" (i.e., "big O") of some function of d and N, e.g., computing a sum of N terms has time complexity $O(N)$, multiplying a $1 \times N$ vector by a $N \times d$ matrix has time complexity $O(Nd)$, etc.

3. Prove that minimizing the MSE is equivalent to maximizing the conditional log-likelihood with a linear model, Gaussian noise, and IID data.

Problem 3: Computational Complexity for Fitting Linear Models using MSE

Consider the optimization problem in Problem 2: fitting a linear model by minimizing MSE, with d parameters and a d-dimensional input $\underline{x}$, with N IID data points. Answer the following questions below:

1. Assume we are using gradient descent algorithm (Section 7.1 in MML) to solve this problem. Define the time complexity of doing one gradient update (using all N data points).

2. Assume that instead of the full gradient method, we use instead the stochastic gradient method (see Section 7.1.3 in MML) for this problem, where we use M randomly selected datapoints as the mini-batch size for each stochastic gradient update. Define the time complexity of doing one such stochastic gradient update.

3. In the context of this problem (i.e., linear model, MSE loss, IID data) write a few sentences (or bullet points) comparing the strengths and weaknesses of the following optimization methods: Normal Equations, gradient descent, stochastic gradient. Focus specifically on the computational complexity and numerical stability of each method, as d increases relative to fixed N and fixed M. Since the number of iterations before convergence for the iterative methods will depend on the data D, on learning rates, on batch sizes, on convergence criteria, etc, its difficult to make any precise statements involving the number of iterations to convergence for these methods: instead its fine in your comments to focus primarily just on computational complexity and numerical stability per iteration.

Problem 4: Gradients for MAP Gaussian Regression

Consider a regression problem with data $D = \{(\underline{x}_i, y_i)\}, i = 1, \ldots, N$, where $\underline{x}$ is a d-dimensional real-valued vector (where one component is set to the constant 1 to allow for an intercept term). Consider a linear model in the form $f(\underline{x}; \underline{\theta}) = \underline{\theta}^T \underline{x}$ where $\underline{\theta}$ is a d-dimensional parameter vector with one weight for each component of $\underline{x}$. Consider a Gaussian regression model of the form $y|\underline{x} \sim N(\underline{\theta}^T \underline{x}, \sigma^2)$ where σ^2 is assumed known. Assume that we have independent priors on each weight of the form $\underline{\theta}_j \sim N(\underline{\theta}; 0, s^2)$ with prior mean 0 and where the prior variance s^2 is assumed known.

1. Define the posterior $\log P(\underline{\theta}|D)$ for this problem

2. Derive the gradient $\nabla_{\underline{\theta}}$ with respect to the parameters $\underline{\theta}$ for this problem.

3. Prove that minimizing $-\log P(\underline{\theta}|D)$ is a convex optimization problem.

Problem 5: L1 or Lasso Regression

Consider a squared error loss function $MSE(\underline{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\underline{x}_i;\underline{\theta})^2$ with training data $D = \{(\underline{x}_i, y_i)\}, i = 1, \ldots, N$ and where f is some prediction model with unknown parameters $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_p)$. A popular regularization method takes the form $r(\underline{\theta}) = \sum_{j=1}^{p} |\underline{\theta}_j|$, resulting in an optimization problem where we minimize $MSE(\underline{\theta}) + \lambda r(\underline{\theta})$, where λ is the relative weight of the regularization term (this is known as L1 or Lasso regularization).

Clearly show how we can interpret L1 regularization in terms of a prior on $\underline{\theta}$ (by viewing this optimization problem from a Bayesian MAP perspective). Be sure to state clearly what distributional form this prior is, i.e., what name it has.

Problem 6: Poisson Regression

Consider a problem where we have a data set $D = \{(\underline{x}_i, y_i)\}, i = 1, \ldots, N$ where $\underline{x}_i$ are real-valued d-dimensional vectors and $y_i \in \{0, 1, 2, \ldots, \}$, i.e., the y_i's are non-negative integers, e.g., a count of the number of purchases an individual i makes on a Website given that they visit the site. In a Poisson regression model we build a model where the conditional distribution of y, $P(y|\underline{x};\underline{\theta})$, is assumed to be a Poisson distribution with mean $E[y|\underline{x}] = \lambda(\underline{x}) = f(\underline{x};\underline{\theta})$ where the mean varies as a function of $\underline{x}$, for some fixed value of parameters $\underline{\theta}$, rather than being having a fixed mean value λ. To ensure that $\lambda(\underline{x}) > 0$, a common parametrization is $\lambda(\underline{x}) = exp(\underline{\theta}^T \underline{x})$, which is what we will use in this problem.

1. Derive the log-likelihood for this problem

2. Derive the gradient of the log-likelihood with respect to $\underline{\theta}$ for this problem

Problem 7: Convexity for Logistic Classifiers

Consider a classification problem where we have training data $D = \{(\underline{x}_i, y_i)\}, 1 \ldots, i, \ldots, N$ where $\underline{x}_i$ are real-valued d-dimensional vectors and $y_i \in \{0, 1\}$ are binary class labels. We will assume for convenience that the first component of $\underline{x}$ always takes value 1, allowing us to have an intercept (or bias) term in our model. Let $f(\underline{x};\underline{\theta})$ be a logistic regression model, where $\underline{\theta}$ is a d-dimensional parameter vector (set of weights), and our predictive model is

$$f(\underline{x};\underline{\theta}) \;=\; \frac{1}{1 + \exp(-\underline{\theta}^T \underline{x})}.$$

where $f(\underline{x};\underline{\theta})$ is our estimate of $p(y = 1|\underline{x})$.

Let the objective function (that we want to minimize) be the cross-entropy loss (also known as the log-loss), defined as

$$CE(\underline{\theta}) \;=\; -\frac{1}{N}\sum_{i=1}^{N} y_i \log f(x_i;\underline{\theta}) \;+\; (1-y_i)\log\big(1-f(x_i;\underline{\theta})\big).$$

1. Derive the equation for the gradient for $\underline{\theta}$ for this optimization problem

2. Prove that $CE(\underline{\theta})$ has a single global minimum and no local minima by proving that it is a convex function of $\underline{\theta}$.

3. Consider a second-order (Newton) method for optimization of the cross-entropy loss with a logistic model. Let $\mathbf{H}_{\underline{\theta}}$ be the Hessian matrix, defined as the $d \times d$ of partial second derivatives of the objective function evaluated at the current parameter values $\underline{\theta}^{(t)}$. Each second-order iteration is defined as

$$\underline{\theta}^{(t+1)} \;=\; \underline{\theta}^{(t)} \;-\; \mathbf{H}_{\underline{\theta}}^{-1}\nabla_{\underline{\theta}} \quad t = 1, 2, \ldots$$

   where $\nabla_{\underline{\theta}}$ is the gradient of the cross-entropy loss evaluated at the current parameter values $\underline{\theta}^{(t)}$.

   - Derive an expression for the i, j th element, $h_{i,j}$, of the Hessian matrix $\mathbf{H}_{\underline{\theta}}$ for this problem, $1 \leq i, j, \leq d$.
   - Define the time complexity of doing one such second order update and write a sentence or two comparing the computational efficiency of this second-order Newton method with first-order gradient descent.

4. Prove from first principles that the $CE(\underline{\theta})$ function above, corresponds to a negative conditional log-likelihood $L(\underline{\theta}) = p(D_y|D_x, \underline{\theta})$ for an IID dataset $D = \{(\underline{x}_i, y_i)\}$. Here the model $f(x_i;\underline{\theta})$ can be any model where $f(x_i;\underline{\theta})$ is bounded between 0 and 1, e.g., a neural network with logistic function at the output.