CS 725: Foundations of Machine Learning (Autumn 2023)

Lecture 1 - 4: Practice Questions

November 16, 2023

Lecturer: Sunita Sarawagi *Scribe: Team ID - 24*[1]

Question 1: Linear Algebra

Suppose x_k is the fraction of IIT Bombay students who prefer Foundations of Machine Learning (FML) to Advanced Machine Learning (AML) at year k. The remaining fraction $y_k = 1 - x_k$ prefers AML. At year $k+1$, $\frac{1}{5}$ of those who prefer FML changed their mind (possibly after taking EE 768). Also at year $k+1$, $\frac{1}{10}$ of those who prefer AML change their mind (possibly because of the exams!). Create the matrix A to give $[x_{k+1} \quad y_{k+1}]^T = [x_k \quad y_k]^T A$ and find the limit of $[1 \quad 0]A^T$ as $k \to \infty$.

Solution:

$$A = \begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.9 \end{bmatrix} \tag{1}$$

The eigenvector with $\lambda = 1$ is $[0.333 \quad 0.667]^T$. This is the steady state starting from $[1 \quad 0]^T$. $\frac{2}{3}$ of all students prefer AML!

Question 2: Linear Regression

The LIBS instrument of India's Chandrayaan-3 rover has made the first-ever in-situ measurements on the elemental composition of the lunar surface near the south pole of Moon. The instrument has just sent back exciting data giving the concentration of water ice y at depth x beneath the surface of the south pole on Moon!

Your task, as one of the mission specialists (back on Earth), is to figure out what hypothesis best models the data, which is shown in Figure 1:

Assume that the datapoints shown in this figure come from three disjoint subsets:

A: Depth $x = 0$ to $x = 6$ (circles)

B: Depth $x = 6$ to $x = 12$ (squares)

C: Depth $x > 12$ (the symbol ×)

[1]Team members: 23m2154, 23m2156, 23m2157, 23m2158, 23m2162, 23d1596

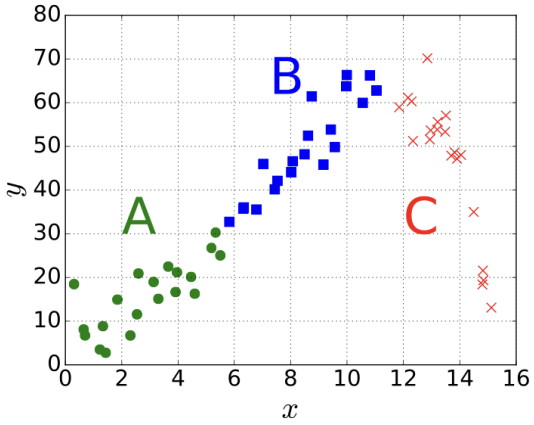

Figure 1: The concentration of water ice y at depth x beneath the surface of the south pole on Moon

And as an ML expert, you know that while you may train your model on one subset of data, you should test it on a different subset of data.

a) Suppose your hypothesis is that ice concentration is linearly related to depth, i.e. $y = \theta^T x + \theta_0$. You employ mean square error (MSE) for the objective function, and use dataset A for training, and dataset B for testing (since they are conveniently disjoint!). Let us say that that MSE below 30 is LOW, and MSE above 100 is HIGH. Judging from the above plot, will the MSE for training be LOW or HIGH? How about for testing? Explain why.

b) Continuing with the hypothesis that ice concentration is linearly related to depth, you now employ datasets A and B (combined) for training, and dataset C for testing. Judging from the above plot, will the MSE for training be LOW or HIGH? How about for testing? Are your choices for training and testing datasets good ones? Explain.

c) Realizing that Moon is unlikely to be a snowball of ice (although it's possible Earth once was!), you switch to a family of hypotheses with nonlinear feature transforms, $y = \theta^T \phi_k(x) + \theta_0$, where $\phi_k(x)$ is a vector of polynomials up to order k. Can you think of any good way to evaluate what order k is the best to choose? Explain.

Solution: a) Training Error: LOW. Testing Error: LOW.
Both errors are LOW because training on dataset A should produce a straight line which fits both A and B very well.

b) Training Error: LOW. Testing Error: HIGH.
Training error will be LOW because training on dataset A and B should produce a straight line which fits both A and B very well. However, extrapolating forward the straight line produced will not be a good fit for dataset C leading to a HIGH testing MSE.

If we are trying to model all of the data (i.e. the data in subsets A, B, and C), the union

of subsets A and B is not representative; it misses out on the behavior in subset C. Similarly, subset C is not representative; it misses out on the behavior in subsets A&B. A better choice of training data would be one that has points from every subset; similarly, a better choice of testing data would have points from every subset.

c) Training Set: randomly select data points from across all three datasets (A, B, C). A good percentage could be 80% data for training.
Testing Set: use the remaining 20% points not chosen for training to be part of the test set.

The reason one would want to choose randomly from across all datasets is because the data for training and for testing should come from the same sample distribution, even if they are disjoint datapoints. Alternatively, use cross-validation. With cross-validation, you could use all the data for training then determine the best k by minimizing the error output by cross-validation. This would mean no need for a single separate test set.

Question 3: Logistic Regression

For the binary logistic regression problem, the target values are encoded as $t^i \in 0, +1$. For a dataset $D_N = (x^{(i)}, t^{(i)})$ with $t^i \in 0, +1$, the logistic regression is defined using the following steps:

$$z = w^T x + b \tag{2}$$
$$y = \sigma(z) \tag{3}$$
$$L(y, t) = -t \log(y) - (1 - t) \log(1 - y) \tag{4}$$

Show that if $t^i \in -1, +1$ then the minimization problem takes the following form where w and b are the weights parameters:

$$min_{w,b} \sum_{i=1}^{N} \log(1 + \exp(-t^i(w^T x^{(i)} + b))) \tag{5}$$

Solution:

We can substitute the expression for y, then later substitute for z:

$$L(z, D) = -t \log(\sigma(z)) - (1-t) \log(1 - \sigma(z)) \tag{6}$$

$$= -t \log\left(\frac{1}{1+\exp(-z)}\right) - (1-t) \log\left(1 - \frac{1}{1+\exp(-z)}\right) \tag{7}$$

$$= -t \log\left(\frac{1}{1+\exp(-z)}\right) - (1-t) \log\left(\frac{1}{1+\exp(z)}\right) \tag{8}$$

$$= t \log\left(1+\exp(-z)\right) + (1-t) \log\left(1+\exp(z)\right) \tag{9}$$

$$L(w, b, D) = t \log\left(1+\exp(-(w^T x + b))\right) + (1-t) \log\left(1+\exp(w^T x + b)\right) \tag{10}$$

$$= \sum_{i=1}^{N} t^{(i)} \log\left(1+\exp(-(w^T x^{(i)} + b))\right) + (1-t^{(i)}) \log\left(1+\exp(w^T x^{(i)} + b)\right) \tag{11}$$

Thus the cost minimization problem when $t^i \in 0, +1$ is formulated as:

$$\min_{w,b} \sum_{i=1}^{N} t^{(i)} \log\left(1+\exp(-(w^T x^{(i)} + b))\right) + (1-t^{(i)}) \log\left(1+\exp(w^T x^{(i)} + b)\right) \tag{12}$$

When $t^i \in -1, +1$ then we can substitute $t^{(i)} = \frac{t^{(i)}+1}{2}$ into the expression above:

$$L(w, b, D) = \sum_{i=1}^{N} \frac{t^{(i)}+1}{2} \log\left(1+\exp(-(w^T x^{(i)} + b))\right) + \frac{1-t^{(i)}}{2} \log\left(1+\exp(w^T x^{(i)} + b)\right) \tag{13}$$

When $t^{(i)} = +1$ for the i^{th} training example, the second term disappears, leading to the remaining term. When $t^{(i)} = -1$ for the i^{th} training example, the first term disappears, leading to the remaining term. therefore, the only difference between the two cases in the sign inside the exponential term, which has the opposite sign as $t^{(i)}$. We can simplify to the desired expression:

$$min_{w,b} \sum_{i=1}^{N} \log(1+\exp(-t^i(w^T x^{(i)} + b))) \tag{14}$$

Question 4: Optimization

Assume that you are minimizing a cost function which can be written as:

$$J(w) = \frac{1}{N}\sum_{i-1}^{N} L(w, x_i, t_i) \tag{15}$$

Where $N = 1,000,000$.

a) Write the one-step update rules for gradient descent (GD), stochastic GD and mini-batch SGD with batch size of 100. You can denote the gradient of the loss with respect to w for each sample with $g_i = \nabla L(w, x_i, t_i)$ and your learning rate is η.

b) Rank the computational cost of each iteration for GD, SGD and mini-batch SGD (with batch size of 100) from smallest to largest.

Solution:

a) The gradient update rule for GD, SGD and mini-batch SGD are as follows:

1. GD:

$$w \leftarrow w - \eta \sum_{i=1}^{N} g_i \tag{16}$$

2. SGD:

$$\text{Choose} \quad i \approx Uniform[1, N], w \leftarrow w - \eta g_i \tag{17}$$

3. mSGD:

$$\text{Choose a subset} M \subset 1, 2, \ldots, N, w \rightarrow w - \eta \sum_{i \in M}^{|M|} g_i \tag{18}$$

b) From smallest to largest $SGD < mSGD < GD$.

SGD only requires processing of 1 training example, mSGD requires 100 batch examples and GD requires 1000000 training examples.

Question 5: Optimization

Let $n \geq 1$ be an integer and let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix (non necessarily positive definite) for which all of its eigenvalues are non-zero. Let $a \in \mathbb{R}^n$ be a given vector and we consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$, defined as:

$$f(x) = \frac{1}{2}(x - a)^T A^2 (x - a) \tag{19}$$

a) Using first and second order optimality conditions show that f has a unique global minimizer on $\mathbb{R}^n$ and determine this optimizer. Denote it by x^*.

b) Write the updates in the gradient descent algorithm with optimal step size starting from a point $x^0 \in \mathbb{R}^n$ to appropriate the optimizer x^* of f that has been determined in (a). Determine the step size α_k in each step.

Solution:
Notice first the since A is symmetric, so is ^{2}. Moreover since A has non-zero eigenvalues, A^2 has all its eigenvalues positive, hence it is a positive definite matrix. Let us define $Q := A^2$. Observe also that the function can be rewritten as:

$$f(x) = \frac{1}{2}x^T Q x - x^T b + c \quad \text{Where } b := Qa, c := \frac{1}{2}a^T Q a \tag{20}$$

a) Since the optimization problem is without constraints, the first order necessary optimality condition for the minimizer reads as $\nabla f(x^*) = 0$, that is $Qx^* = b$, from where $x^* = Q^{-1}b = Q^{-1}Qa = a$. All these computations are meaningful because Q^-1 exists. The second order sufficient condition of minimality reads as $D^2 f(x^*) = Q = A^2 > 0$ which is true. Hence $x^* = a$ is the unique global minimizer of f on $\mathbb{R}^n$.

b) The updates in the gradient descent starting from x^0 are:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) = x^k - \alpha_k (Qx^k - b) \tag{21}$$

Where $\alpha_k = argmin_{\alpha \in \mathbb{R}} f(x^k - \alpha \nabla f(x^k))$.

$$\alpha_k = \frac{||\nabla f(x^k)||^2}{\nabla f(x^k)^T Q \nabla f(x^k)} \tag{22}$$

Question 6: Mixed - Optimization and Linear Algebra

Let $n \geq 1$ be an integer and let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix (non necessarily positive definite) for which all of its eigenvalues are non-zero. Let $a \in \mathbb{R}^n$ be a given vector and we consider the function $f : \mathbb{R} \to \mathbb{R}$, defined as:

$$f(x) = \frac{1}{2}(x - a)^T A^2 (x - a) \tag{23}$$

a) Imagine that one wants to use a fixed step gradient algorithm too, to approximate x^*. Which is maximal range for the step size α in terms of the eigenvalues of A that ensures global convergence for the algorithm?

b) Give an example of $A \in \mathbb{R}^{2\times 2}$ diagonal matrix that has a zero and a non-zero eigenvalue. Take $a \in \mathbb{R}^2$. Determine the global minimizers of f in $\mathbb{R}^2$ in this case. What can we say about the uniqueness of them?

Solution:
a) For the fixed step size algorithm global convergence is equivalent to $0 < \alpha < \frac{2}{\lambda_{max}(Q)}$. The maximal eigenvalue of Q actually can be written in terms of the maximal (in absolute value) eigenvalue of A i.e. $\lambda_{max}(Q) = \max \lambda_i^2 : i = 1, 2, \ldots, n$, where the λ_i are the eigenvalues of A counted with multiplicity.

b) An example of such a matrix is:

$$A = \begin{bmatrix} \gamma & 0 \\ 0 & 0 \end{bmatrix} \tag{24}$$

Where $\gamma \neq 0$. The other option is when the elements on the main diagonal are exchanged. In this case,

$$Q = A^2 = \begin{bmatrix} \gamma^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{25}$$

and the function can be written as $f(x_1, x_2) = \frac{1}{2}\gamma_2(x_1 - a_1)^2$, hence it is independent of the second variable. Setting $\nabla f(x) = 0$ one finds that the candidates for the optimizers are $x^* = (a_1, x_2)$, where $x_2 \in \mathbb{R}$ is arbitrary. Since the function is independent of the second variable and $f(a_1, x_2) = 0 \leq f(y_1, y_2)$ for any $(y_1, y_2) \in \mathbb{R}^2$, one has that all of them are global minimizers that have the same objective function value, hence they are not unique.

Question 7: Mixed - Optimization and Linear Algebra

Let $n \geq 1$ be an integer and let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix (non necessarily positive definite) for which all of its eigenvalues are non-zero. Let $a \in \mathbb{R}^n$ be a given vector and we consider the function $f : \mathbb{R} \to \mathbb{R}$, defined as:

$$f(x) = \frac{1}{2}(x - a)^T A^2 (x - a) \tag{26}$$

a) Explain what will happen if we want to proceed with a fixed step size gradient algorithm for Question 6 (b). Does an algorithm like this converge globally? If yes, for which values of the step size α and to which limit point x^*?

b) Explain what is the major difference between the cases when A has at least one zero eigenvalue and when it does not, from the point of view of the gradient descent algorithms.

Solution:
a) In the case of Question 6 (b) the problem is reduced to a 1D problem, hence a fixed step size gradient algorithm converges globally if and only if the step size α is in the range $0 < \alpha < \frac{2}{\gamma^2}$. From the 2D point of view what is happening is the following: choosing any initial guess $x^0 = (x_1^0, x_2^0)$, since f is independent of the second variable (hence the second coordinate of its

gradient is always zero), during each update in $x^{k+1} = (x_1^{k+1}, x_2^{k+1})$ the second coordinate x_2^{k+1} remains unchanged. Hence the algorithm actually converges to a global minimizer namely the one (a_1, x_2^0).

b) If some of the eigenvalues of A are zero, $Q = A^2$ will have also the corresponding eigenvalues 0. On the other hand, since Q is symmetric, it is diagonalizable, so we can see it up to a change of coordinates as a diagonal matrix with the eigenvalues on the main diagonal. As we have seen in (a), the coordinates (in the new system of coordinates, if Q was not diagonal at the first place) corresponding to the zero eigenvalues are unaffected by the gradient algorithms. And the dimension of the problem can be reduced by the number of zero eigenvalues. While for positive definite Q, i.e. if A does not have zero eigenvalues, the problem is full dimensional. This is a major difference between the two cases.

Question 8: Linear Algebra

If A is 3×3 symmetric positive definite, then $Aq_i = \lambda_i q_i$ with positive eigenvalues and orthonormal eigenvectors q_i. Suppose $x = c_1q_1 + c_2q_2 + c_3q_3$.

a) Compute x^Tx and also x^TAx in terms of the c and λ.

b) Looking at the ratio of x^TAx in part (a) divided by x^Tx in part (a), what c would make that ratio as large as possible ? You can assume $\lambda_1 < \lambda_2 < .\cdots < \lambda_n$. Also find x where the ratio $\frac{x^TAx}{x^Tx}$ is a maximum.

Solution: a)

$$x^Tx = \left(\sum_{i=1}^{3} c_i q_i^T\right)\left(\sum_{j=1}^{3} c_j q_j\right) \tag{27}$$

$$= c_1^2 q_1^T q_1 + c_1c_2 q_1^T q_2 + \cdots + c_3^2 q_3^T q_3 \tag{28}$$

$$= c_1^2 + c_2^2 + c_3^2 \tag{29}$$

$$x^TAx = \left(\sum_{i=1}^{3} c_i q_i^T\right)\left(\sum_{j=1}^{3} c_j Aq_j\right) \tag{30}$$

$$= \left(\sum_{i=1}^{3} c_i q_i^T\right)\left(\sum_{j=1}^{3} c_j \lambda_j q_j\right) \tag{31}$$

$$= c_1^2\lambda_1 q_1^T q_1 + c_1c_2\lambda_2 q_1^T q_2 + \cdots + c_3^2\lambda_3 q_3^T q_3 \tag{32}$$

$$= c_1^2\lambda_1 + c_2^2\lambda_2 + c_3^2\lambda_3 \tag{33}$$

b) We maximize $\frac{\sum_{i=1}^{3} c_i^2 \lambda_i}{\sum_{i=1}^{3} c_i^2}$ when $c_1 = c_2 = 0$ so $x = c_3 q_3$ is a multiple of the eigenvector q_3 with the largest eigenvalue λ_3.