CS 475/675 Machine Learning: Homework 1
Analytical Questions
25 Points Total Version 1.1

Alexandra Szewc, Cassie Parent (aszewc1, cparent5)

Instructions

We have provided this LaTeX document for completing the analytical portion of the assignment. We give you one or more boxes to answer each question. The question to answer for each box will be noted in the title of the box.

Other than your name, do not type anything outside the boxes. Leave the rest of the document unchanged.

Do not change any formatting in this document, or we may be unable to grade your work. This includes, but is not limited to, the height of textboxes, font sizes, and the spacing of text and tables. Additionally, do not add text outside of the answer boxes. Entering your answers are the only changes allowed.

We strongly recommend you review your answers in the generated PDF to ensure they appear correct. We will grade what appears in the answer boxes in the submitted PDF, NOT the original latex file.

Notation

$\mathbf{x_i}$	One input data vector. $\mathbf{x_i}$ is M dimensional. $\mathbf{x_i} \in \mathbb{R}^{1\times M}$. We assume $\mathbf{x_i}$ is augmented with a 1 to include a bias term.
$\mathbf{X}$	A matrix of concatenated $\mathbf{x_i}$'s. There are N input vectors, so $\mathbf{X} \in \mathbb{R}^{N\times M}$
y_i	The true label for input vector $\mathbf{x_i}$. In regression problems, y_i is continuous. In general ,y_i can be a vector, but for now we assume it's a scalar: $y_i \in \mathbb{R}^1$.
$\mathbf{y}$	A vector of concatenated y_i's. There are N input vectors, so $\mathbf{y} \in \mathbb{R}^{N\times 1}$
$\mathbf{w}$	A weight vector. We are trying to learn the elements of $\mathbf{w}$. $\mathbf{w}$ is the same number of elements as $\mathbf{x_i}$ because we will end up computing the dot product $\mathbf{x_i} \cdot \mathbf{w}$. $\mathbf{w} \in \mathbb{R}^{M\times 1}$. We assume the bias term is included in $\mathbf{w}$.
$h((x))$	The true regression function that describes the data.
i.i.d.	Independently and identically distributed.
Bias-variance decomposition	We can write $E_D[(f(x, D) - h(x))^2] =$ $(E_D[f(x, D) - h(x))^2 + E_D[(f(x, D) - E_D[f(x, D)])^2]$ where the first term is the bias squared, and the second term is the variance.
Notes:	In general, a lowercase letter (not boldface), a, indicates a scalar. A boldface lowercase letter, $\mathbf{a}$, indicates a vector. A boldface uppercase letter, $\mathbf{A}$, indicates a matrix.

1) Loss/Error (4 points)

For each question, write out the probability density of each model and the referenced error function to answer (yes/no) and explain.

(1) In linear regression, with fixed σ^2, is the mean squared error equivalent to the negative log likelihood?

Yes, the result of minimizing the negative log likelihood will be the same as the result of minimizing the mean squared error. The probability density function for the linear regression model and resulting negative log likelihood are:

$$p(\mathbf{y}|\mathbf{x}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - \mathbf{w}^T\mathbf{x_i})^2}$$

$$-\log p(\mathbf{y}|\mathbf{x}, \sigma^2) = \frac{N}{2}\log(2\pi) + N\log\sigma + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x_i})^2$$

The mean squared error is:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x_i})^2$$

Since differentiation will result in the removal of some terms, we can see that the MSE is equivalent to the relevant component of the negative log likelihood. Therefore, minimizing one will also minimize the other resulting in the same set of weights no matter which approach is used.

(2) In logistic regression is the mean squared error equivalent to the negative log likelihood?

No. The probability density function for the linear regression model and resulting negative log likelihood are:

$$p(\mathbf{y}|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

$$-\log p(\mathbf{y}|\mathbf{x}; \mathbf{w}) = -\sum_{i=1}^{N}\log\left(\frac{1}{1 + e^{-\mathbf{wx_i}}}\right)^y\left(\frac{1 + e^{-\mathbf{wx_i}}}{1 + 1 + e^{-\mathbf{wx_i}}}\right)^{1-y}$$

The mean squared error is:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x_i})^2$$

By inspection, we see that these are distinct functions from each other. Therefore, minimizing the negative log likelihood and minimizing the MSE will not yield the same set of weights.

2) Linear Regression (7 points)

Consider the simple linear regression model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, ..., n$ where $\epsilon_i | x_i$ iid $N(0, \sigma^2)$. Suppose that σ^2 is known. Let $\hat{\beta} = (\hat{\beta}_0 \;\; \hat{\beta}_1)^T$ denote the maximum likelihood estimator.

(1) Derive the maximum likelihood estimates $\hat{\beta}$ by differentiating the log likelihood, and then derive the variance covariance matrix $Var(\hat{\beta}) = \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{bmatrix}$.

$$N(\mathbf{y}|\mathbf{x}; \beta_0, \beta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - \mathbf{x_i})^2}$$

$$p(\mathbf{y}|\mathbf{x}; \beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{-\frac{1}{2\sigma^2}(y_i - (\beta_0 + \beta_1 \mathbf{x_i}))^2}$$

$$\mathcal{L} = \log p(\mathbf{y}|\mathbf{x}; \beta_0, \beta_1, \sigma^2) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - (\beta_0 + \beta_1 \mathbf{x_i}))^2$$

Starting with $\hat{\beta}_0$:

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \beta_1 \mathbf{x_i} - \beta_0) = 0$$

$$\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} y_i - \frac{\beta_1}{N}\sum_{i=1}^{N} \mathbf{x_i}$$

Now with $\hat{\beta}_1$:

$$\frac{\partial \mathcal{L}}{\partial \beta_1} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \beta_1 \mathbf{x_i} - \beta_0)\mathbf{x_i} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \beta_1} = -\frac{1}{\sigma^2}\sum_{i=1}^{N} y_i \mathbf{x_i} + \frac{1}{\sigma^2}\beta_1 \sum_{i=1}^{N} \mathbf{x_i}^2 + \frac{1}{\sigma^2}\beta_0 \sum_{i=1}^{N} \mathbf{x_i}$$

$$= -\frac{1}{\sigma^2}\sum_{i=1}^{N} y_i \mathbf{x_i} + \frac{1}{\sigma^2}\beta_1 \sum_{i=1}^{N} \mathbf{x_i}^2 + \frac{1}{\sigma^2}(\frac{1}{N}\sum_{i=1}^{N} y_i - \frac{\beta_1}{N}\sum_{i=1}^{N} \mathbf{x_i})\sum_{i=1}^{N} \mathbf{x_i}$$

$$= \frac{1}{\sigma^2}\beta_1(\sum_{i=1}^{N} \mathbf{x_i}^2 - \frac{1}{N}(\sum_{i=1}^{N} \mathbf{x_i})^2) + \frac{1}{N\sigma^2}\sum_{i=1}^{N} y_i \sum_{i=1}^{N} \mathbf{x_i} - \frac{1}{\sigma^2}\sum_{i=1}^{N} y_i \mathbf{x_i} = 0$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} y_i \mathbf{x_i} - \frac{1}{N}\sum_{i=1}^{N} y_i \sum_{i=1}^{N} \mathbf{x_i}}{\sum_{i=1}^{N} \mathbf{x_i}^2 - \frac{1}{N}(\sum_{i=1}^{N} \mathbf{x_i})^2}$$

$$\nabla \log(\mathcal{D}|\beta) = \sum_{i=1}^{N} (y_i - \beta^T \mathbf{x_i}) \mathbf{x_i}^T$$

$$\sum_{i=1}^{N} (y_i \mathbf{x_i}^T) - \beta^T \sum_{i=1}^{N} \mathbf{x_i} \mathbf{x_i}^T = 0$$

$$\sum_{i=1}^{N} (y_i \mathbf{x_i}^T) = \beta^T \sum_{i=1}^{N} \mathbf{x_i} \mathbf{x_i}^T$$

$$(\mathbf{Y}^T \mathbf{X} = \beta^T \mathbf{X}^T \mathbf{X})^T$$

$$\mathbf{X}^T \mathbf{Y} = \mathbf{X}^T \mathbf{X} \beta$$

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

$Var(\hat{\beta}) = E(\hat{\beta}^2) - E(\hat{\beta}) E(\hat{\beta}^T)$
$= E[((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y})^2] - \beta\beta^T$
$= E[((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{X}\beta + \epsilon))^2] - \beta\beta^T$
$= E[((\beta + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon)^2] - \beta\beta^T$
$= \beta\beta^T + E[2(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \beta\epsilon] + E[((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon)^2] + \beta\beta^T$
$= E[((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon)^2]$ since $E[\epsilon] = \mu = 0$
$= E[\epsilon^2]((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T)^2$ since $E[\epsilon^2] = \sigma^2$
$= \sigma^2((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$
Using this result, we expand the matrix form to derive the full covariance variance matrix:

$$\sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} = \sigma^2 \begin{bmatrix} N & \sum_{i=1}^{N} \mathbf{x_i} \\ \sum_{i=1}^{N} \mathbf{x_i} & \sum_{i=1}^{N} \mathbf{x_i}^2 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \frac{\sigma^2 \sum_{i=1}^{N} \mathbf{x_i}^2}{N(\sum_{i=1}^{N} \mathbf{x_i}^2 - (\sum_{i=1}^{N} \mathbf{x_i})^2)} & -\frac{\sigma^2 \sum_{i=1}^{N} \mathbf{x_i}}{N(\sum_{i=1}^{N} \mathbf{x_i}^2 - (\sum_{i=1}^{N} \mathbf{x_i})^2)} \\ -\frac{\sigma^2 \sum_{i=1}^{N} \mathbf{x_i}}{N(\sum_{i=1}^{N} \mathbf{x_i}^2 - (\sum_{i=1}^{N} \mathbf{x_i})^2)} & \frac{\sigma^2}{\sum_{i=1}^{N} \mathbf{x_i}^2 - (\sum_{i=1}^{N} \mathbf{x_i})^2} \end{bmatrix}$$

$$= \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{bmatrix}$$

(2) Verify that $Var(\hat{\beta}) = I^{-1}(\beta)$ where $I(\beta)$ is the expected information matrix with elements $I_{ij}(\beta) = \mathbb{E}[-\frac{\partial^2 l(\beta)}{\partial\beta_i \partial\beta_j}]$

Beginning with the first partial derivatives of $\mathcal{L}$ with respect to β_0 and β_1 from the first part, we can compute the following:

$$I_{00}(\beta) = \frac{\partial^2 \mathcal{L}}{\partial\beta_0 \partial\beta_0} = \frac{\partial}{\partial\beta_0}[\frac{1}{\sigma^2}\sum_{i=1}^{N}\beta_0] = \frac{N}{\sigma^2}$$

$$I_{01}(\beta) = \frac{\partial^2 \mathcal{L}}{\partial\beta_0 \partial\beta_1} = \frac{\partial}{\partial\beta_1}[\frac{1}{\sigma^2}\beta_0\sum_{i=1}^{N}\mathbf{x_i}] = \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i}$$

$$I_{10}(\beta) = \frac{\partial^2 \mathcal{L}}{\partial\beta_1 \partial\beta_0} = \frac{\partial}{\partial\beta_0}[\frac{1}{\sigma^2}\beta_0\sum_{i=1}^{N}\mathbf{x_i}] = \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i}$$

$$I_{11}(\beta) = \frac{\partial^2 \mathcal{L}}{\partial\beta_1 \partial\beta_1} = \frac{\partial}{\partial\beta_1}[\frac{1}{\sigma^2}\beta_1\sum_{i=1}^{N}\mathbf{x_i}\mathbf{x_i}^T] = \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i}\mathbf{x_i}^T$$

Therefore, $I(\beta)$ is:

$$I(\beta) = \begin{bmatrix} -\frac{N}{\sigma^2} & -\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i} \\ -\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i} & -\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i}\mathbf{x_i}^T \end{bmatrix}$$

Taking the inverse, we have:

$$I(\beta)^{-1} = \frac{\sigma^4}{N\sum_{i=1}^{N}\mathbf{x_i}^2 - (\sum_{i=1}^{N}\mathbf{x_i})^2}\begin{bmatrix} -\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i}\mathbf{x_i}^T & \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i} \\ \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathbf{x_i} & -\frac{N}{\sigma^2} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sigma^2\sum_{i=1}^{N}\mathbf{x_i}^2}{N(\sum_{i=1}^{N}\mathbf{x_i}^2 - (\sum_{i=1}^{N}\mathbf{x_i})^2)} & -\frac{\sigma^2\sum_{i=1}^{N}\mathbf{x_i}}{N(\sum_{i=1}^{N}\mathbf{x_i}^2 - (\sum_{i=1}^{N}\mathbf{x_i})^2)} \\ -\frac{\sigma^2\sum_{i=1}^{N}\mathbf{x_i}}{N(\sum_{i=1}^{N}\mathbf{x_i}^2 - (\sum_{i=1}^{N}\mathbf{x_i})^2)} & \frac{\sigma^2}{\sum_{i=1}^{N}\mathbf{x_i}^2 - (\sum_{i=1}^{N}\mathbf{x_i})^2} \end{bmatrix}$$

$$= \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{bmatrix}$$

3) Ranking (6 points)

In regression tasks a function predicts real valued numbers for each instance. In classification tasks a function predicts labels for each instance.
In ranking, each instance considers a *set* of candidates and a function must order (rank) the candidates. This is a common supervised machine learning setting, for example, ranking search engine results. We haven't discussed how to design or train ranking algorithm, but some of the algorithms we have learned about can be used within a ranking task.

(1) How would you use a trained regression or classification algorithm to rank a set of instances? Assume that the model has already been trained for this purpose. You only need to describe how it will be used at test time.

Suppose there is a position q with c possible candidates that need to be ranked. Assuming we had a trained regression model, we will use a point-wise approach at testing time. Given each candidate, the model will predict its relevance into a numerical score. The candidates are then sorted by this scoring metric.

(2) Provide a loss function suitable for ranking. This loss function need not be related to your answer in the first part of the question. Explain how your loss function works.

One loss function suitable for ranking is the mean reciprocal rank (MRR). The idea that for position q, the ranked spot of the first relevant candidate is denoted r(q). Then the MRR is 1/r(q). So, if you pick the most relevant candidate c, you get a reciprocal rank of 1, if you picked the second you get a score of 1/2 and so on. You then average the ranks for all of your samples to get a final score. You are trying to maximize this: the best algorithm would rank the best candidate first every time.

4) Regularization. (8 points)

Linear models are popular because they are simple, powerful and interpretable. A standard formulation of a linear model is an error function combined with a regularization term.

Consider these three objective functions: that use least squares as the error function:

$$\hat{\theta}_0 = \underset{\theta_0}{\operatorname{argmin}} \|y - X_1\theta_0\|_2^2, \tag{1}$$

$$(\hat{\theta}_1, \hat{\theta}_2) = \underset{\theta_1,\theta_2}{\operatorname{argmin}} \|y - X_1\theta_1 - X_2\theta_2\|_2^2, \tag{2}$$

$$\hat{\theta}_3 = \underset{\theta_3}{\operatorname{argmin}} \|y - X_1\theta_3\|_2^2 + \lambda\|\theta_3\|_2^2, \tag{3}$$

where $\lambda > 0$, $y \in \mathbb{R}^n$, $X_1 \in \mathbb{R}^{n\times d_1}$, and $X_2 \in \mathbb{R}^{n\times d_1}$.

You probably recognize (3) as the objective function for Ridge regression. The square norm acts as a penalty function to reduce overfitting.

Prove that

$$\|y - X_1\hat{\theta}_3\|_2^2 \geq \|y - X_1\hat{\theta}_0\|_2^2 \geq \|y - X_1\hat{\theta}_1 - X_2\hat{\theta}_2\|_2^2. \tag{4}$$

FTSOC assume that $||y - X_1\hat{\theta}_3||_2^2 < ||y - X_1\hat{\theta}_0||_2^2$. The only difference in these two equations is the parameter $\hat{\theta}$. θ_3 must must be closer to y than θ_1. This means that (3) must provide a better optimization for the training than (1). Because (3) better fits the training data, (3) has more overfitting than (1). To create this overfitting, we will set $\lambda < 0$ so that optimizing θ_3 decreases the overall error and promotes the overfitting of the data. This is a contradiction! We know that $\lambda > 0$ so there is a penalty for overfitting. Because (1) does not penalize for overfitting, $\hat{\theta}_0$ can be chosen to make $||y - X_1\hat{\theta}_0||_2^2$ as small as possible with no restrictions. This makes $||y - X_1\hat{\theta}_3||_2^2 > ||y - X_1\hat{\theta}_0||_2^2$.

Now, we will prove that $||y - X_1\hat{\theta}_0||_2^2 \geq ||y - X_1\hat{\theta}_1 - X_2\hat{\theta}_2||_2^2$. For minimal error, when we predict y we should take into account errors from both X_1 and X_2. In (1), only X_1 is taken into account. This will lead to a larger error then in (2) where both X_1 and X_2 are taken into account. Because (2) has this extra parameter, the overall error will be lower and $||y - X_1\hat{\theta}_0||_2^2 \geq ||y - X_1\hat{\theta}_1 - X_2\hat{\theta}_2||_2^2$.

Via the transitive property since we know $||y - X_1\hat{\theta}_3||_2^2 \geq ||y - X_1\hat{\theta}_0||_2^2$ and that $||y - X_1\hat{\theta}_0||_2^2 \geq ||y - X_1\hat{\theta}_1 - X_2\hat{\theta}_2||_2^2$, then $||y - X_1\hat{\theta}_3||_2^2 \geq ||y - X_1\hat{\theta}_0||_2^2 \geq ||y - X_1\hat{\theta}_1 - X_2\hat{\theta}_2||_2^2$.