Discussion 5

Note: *Your TA will probably not cover all the problems on this worksheet. The discussion worksheets are not designed to be finished within an hour. They are deliberately made slightly longer so they can serve as resources you can use to practice, reinforce, and build upon concepts discussed in lectures, discussions, and homework.*

This Week's Cool AI Demo/Video:
Deep Dream: `https://www.youtube.com/watch?v=DgPaCWJL7XI`

Explanation for Deep Dream: `https://youtu.be/ta5fdaqDT3M?si=TfGM_QnBBHoJ4sbl&t=2590`

1 Maximum Likelihood Estimation with Laplace Noise

Suppose we have a regression model with parameters $w \in \mathbb{R}^D$ and observations (x_n, t_n) for $n = 1, \dots, N$. We assume that the observed targets are generated as

$$t_n = x_n^\top w + \epsilon_n,$$

where the noise terms ϵ_n are i.i.d. from a Laplace distribution:

$$p(\epsilon) = \frac{1}{2b} \exp\left(-\frac{|\epsilon|}{b} \right).$$

(a) Write down the likelihood $p(\mathcal{D}|w)$ for the parameters w given the dataset $\mathcal{D} = \{(x_n, t_n)\}_{n=1}^N$.

Solution:

Goal. Derive the likelihood $p(\mathcal{D} \mid w)$ for the dataset $\mathcal{D} = \{(x_n, t_n)\}_{n=1}^N$ under the model

$$t_n = x_n^\top w + \epsilon_n, \quad \epsilon_n \overset{\text{i.i.d.}}{\sim} \text{Laplace}(0, b).$$

Step 1: Single-observation conditional density. Since $t_n = x_n^\top w + \epsilon_n$ and $\epsilon_n \sim \text{Laplace}(0, b)$, it follows that

$$t_n \,\big|\, x_n, w \;\sim\; \text{Laplace}(x_n^\top w, b),$$

whose pdf is

$$p(t_n \mid x_n, w) = \frac{1}{2b} \exp\left(-\frac{|t_n - x_n^\top w|}{b} \right).$$

Step 2: Independence across data points. Because the noise variables $\epsilon_1, \dots, \epsilon_N$ are i.i.d., the targets $t_1, \dots, t_N$ are conditionally independent given w. Thus:

$$p(t_1, \dots, t_N \mid X, w) = \prod_{n=1}^{N} p(t_n \mid x_n, w).$$

Step 3: Full likelihood. Substituting the pdf from Step 1, the likelihood is

$$p(\mathcal{D} \mid w) = \prod_{n=1}^{N} \frac{1}{2b} \exp\left(-\frac{|t_n - x_n^\top w|}{b} \right).$$

Final expression.

$$\boxed{p(\mathcal{D} \mid w) \;=\; \prod_{n=1}^{N} \frac{1}{2b} \exp\bigl(-\tfrac{1}{b} \,|t_n - x_n^\top w| \bigr)}$$

(b) What is the log-likelihood? Simplify your answer as much as possible.

Solution:

Step 1: Recall the likelihood. From the previous part we have

$$p(\mathcal{D} \mid w) = \prod_{n=1}^{N} \frac{1}{2b} \exp\left(-\frac{|t_n - x_n^\top w|}{b} \right).$$

Step 2: Take the logarithm. The log-likelihood is

$$\log p(\mathcal{D} \mid w) = \log \prod_{n=1}^{N} \left[\frac{1}{2b} \exp\left(-\frac{|t_n - x_n^\top w|}{b} \right) \right].$$

Step 3: Expand the product into a sum. Using log rules, we get

$$\log p(\mathcal{D} \mid w) = \sum_{n=1}^{N} \log\left(\frac{1}{2b}\right) \;+\; \sum_{n=1}^{N} \left(-\frac{|t_n - x_n^\top w|}{b} \right).$$

Step 4: Simplify. The first sum can be expressed more simply as

$$\sum_{n=1}^{N} \log\left(\frac{1}{2b}\right) = N \log\left(\frac{1}{2b}\right) = -N \log(2b).$$

The second sum can be expressed more simply as

$$-\frac{1}{b} \sum_{n=1}^{N} |t_n - x_n^\top w|.$$

Final expression. Putting these together,

$$\boxed{\log p(\mathcal{D} \mid w) = -N \log(2b) \;-\; \frac{1}{b} \sum_{n=1}^{N} |t_n - x_n^\top w|.}$$

(c) Show that maximizing the log-likelihood is equivalent to minimizing the **mean absolute error (MAE)**, defined as

$$\mathrm{MAE}(w) = \frac{1}{N}\sum_{n=1}^{N} |t_n - x_n^\top w|.$$

Solution:

From the previous step, the log-likelihood of the dataset is

$$\log p(\mathcal{D} \mid w) = -N\log(2b) - \frac{1}{b}\sum_{n=1}^{N} |t_n - x_n^\top w|.$$

Step 1: Identify the terms that depend on w. The first term, $-N\log(2b)$, is constant with respect to w and can be ignored when optimizing over w. This leaves

$$-\frac{1}{b}\sum_{n=1}^{N} |t_n - x_n^\top w|.$$

Step 2: Simplify the optimization objective. Because $\frac{1}{b}$ is a positive constant, maximizing the log-likelihood is equivalent to minimizing

$$\sum_{n=1}^{N} |t_n - x_n^\top w|.$$

Step 3: Interpret the objective. This objective is simply the sum of absolute prediction errors. Dividing by N gives the average error:

$$\boxed{\mathrm{MAE}(w) = \frac{1}{N}\sum_{n=1}^{N} |t_n - x_n^\top w|.}$$

Remark. This function is also known as the Laplace loss.

(d) Now consider instead a standard regression model with *Gaussian* noise. Let $t = (t_1, \ldots, t_N)^\top \in \mathbb{R}^N$ denote the vector of observed targets, and let $X \in \mathbb{R}^{N\times D}$ be the design matrix whose rows are the feature vectors $x_n^\top$. Now, we assume our observations follow the model:

$$t_n = x_n^\top w + \varepsilon_n, \qquad \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2).$$

The likelihood of our data under Gaussian noise is

$$p(t \mid X, w) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t_n - x_n^\top w)^2}{2\sigma^2}\right).$$

Place an independent Laplace prior on the coefficients w using $\mu = 0$ and $b = 2\sigma^2/\lambda$ such that

$$p(w) = \prod_{d=1}^{D} \frac{\lambda}{4\sigma^2} \exp\big(-\frac{\lambda}{2\sigma^2}|w_d|\big).$$

Recall that the **Maximum a Posteriori (MAP) estimate** chooses the parameter w that maximizes the posterior density:

$$\hat{w}_{\text{MAP}} = \arg\max_{w}\ p(w \mid t, X) \quad = \quad \arg\max_{w}\ p(t \mid X, w)\, p(w).$$

Show that this MAP estimate is equivalent to the solution to the Lasso (L_1 regularized) regression problem.

Solution:

The negative log-likelihood (ignoring additive constants that don't depend on w) is

$$-\log p(t \mid X, w) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - x_n^\top w)^2 + \text{const.}$$

The negative log-prior (ignoring additive constants that don't depend on w) is

$$-\log p(w) = \frac{\lambda}{2\sigma^2}\|w\|_1 + \text{const.}$$

Therefore the MAP estimate is

$$\begin{aligned}\hat{w}_{\text{MAP}} &= \arg\max_{w}\ p(t \mid X, w)p(w)\\ &= \arg\min_{w}\left[\frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - x_n^\top w)^2 + \frac{\lambda}{2\sigma^2}\|w\|_1\right]\\ &= \arg\min_{w}\left[\frac{1}{2}\sum_{n=1}^{N}(t_n - x_n^\top w)^2 + \frac{\lambda}{2}\|w\|_1\right].\end{aligned}$$

This minimization problem is exactly the **Lasso regression** objective.

(e) Briefly contrast the two estimators:

1. **Laplace-noise MLE**: Minimizes the MAE (L_1 loss on residuals)
2. **Gaussian-noise + Laplace-prior MAP (Lasso)**: Minimizes squared error with an L_1 penalty on coefficients

Solution:

The Laplace-noise MLE changes the *model error/data-fit term* to penalize large residuals less heavily than squared error, so the estimator is robust to outliers in the targets.

Lasso (Laplace prior on w with Gaussian noise) changes the *regularization term*, promoting sparse w by shrinking some coefficients to zero.

2 Bias-Variance Trade-off

As we saw in lecture, the expected error for a model f_w at a test point x can be decomposed as:

$$\mathbb{E}\left[(t - f_w(x))^2\right] = \underbrace{\mathbb{E}\left[(t - h(x))^2\right]}_{(i)} + \underbrace{\left(h(x) - \mathbb{E}[f_w(x)]\right)^2}_{(ii)} + \underbrace{\mathbb{E}\left[\left(\mathbb{E}[f_w(x)] - f_w(x)\right)^2\right]}_{(iii)}, \tag{1}$$

where $f_w(x)$ is the prediction of our model, parameterized by w, trained on a particular dataset.

We set t to be the true label for the test point, and assume it comes from some underlying ground-truth function h such that

$$t = h(x) + \epsilon,$$

where ϵ is random noise inherent in the system. Assume $\mathbb{E}[\epsilon] = 0$ and $\text{Var}(\epsilon) = \sigma^2$.

(a) From the decomposition above, consider the following symbols:

- t: the true test label
- w: the weights of the fitted model
- x: the test input

Which of these are random variables, and which are not? Briefly explain.

Solution:

- t: Random variable due to the noise ϵ added to the ground-truth function $h(x)$.
- w: Random variable, since the weights are fitted to randomly sampled training data.
- x: Not a random variable (in this context); we are interested in the test error at a fixed x.

(b) Rewrite equation (1) in terms of σ^2, b (the **model's bias**), and v (the **model's variance**).

Solution:

Term (i) can be rewritten as $\mathbb{E}[\epsilon^2] = Var(\epsilon) + \mathbb{E}[\epsilon]^2 = \sigma^2 + 0^2 = \sigma^2$.

Term (ii) is the square of the difference between the ground-truth function and the model's expected prediction. Thus, it represents the model's bias squared, b^2.

Recall that $f_w(x)$ is a random variable because it is dependent on w, which is random. From that perspective, term (iii) is just the definition of variance.

Our final equation is then

$$\mathbb{E}\left[(t - f_w(x))^2\right] = \sigma^2 + b^2 + v$$

(c) Suppose you're tasked with predicting housing prices using a regression model.

(i) If you move from linear regression to a neural network model with many more parameters, how do you expect the bias and variance to change?

(ii) What if you keep linear regression but add a ridge (L_2) regularization term. How does this affect bias and variance?

Solution:

(i) Moving to a neural network with more parameters:

- **Bias:** *Decreases* (model is more flexible, can fit more complex patterns)
- **Variance:** *Increases* (More parameters so model's predictions change a lot depending on the training data)

(ii) Adding ridge (L_2) regularization to linear regression:

- **Bias:** *Increases* (model is more restricted, less flexible)
- **Variance:** *Decreases* (Restriction on change in parameters, so model's predictions become less sensitive to training data variations)