CS 6190: Probabilistic Machine Learning Spring 2022

Homework 2

Meysam Alishahi (U1323606)

Handed out: 22 Feb, 2022
Due: 11:59pm, 15 March, 2022

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do not need to write it in gory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.

Analytical problems [60 points + 25 bonus]

1. [10 points] Given a Gaussian likelihood, $p(x|\mu,\sigma) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$, following the general definition of Jeffery's prior,

   (a) [5 points] show that given σ fixed, the Jeffery's prior over μ, $\pi_J(\mu) \propto 1$;
   Answer: We first need to sompute Fisher information

$$\begin{aligned} I(\mu) &= -\mathbb{E}_\mu\left[\frac{\mathrm{d}^2 \log p(x|\mu)}{\mathrm{d}\mu^2}\right] \\ &= -\mathbb{E}_\mu\left[\frac{\mathrm{d}^2}{\mathrm{d}\mu^2}\left(C - \frac{1}{2\sigma^2}(x-\mu)^2\right)\right] \\ &= \mathbb{E}_\mu\left[\frac{1}{\sigma^2}\right] = \frac{1}{\sigma^2}. \end{aligned}$$

   Therefore, since σ is considered constant $\pi_J(\mu) \propto \left|\frac{1}{\sigma^2}\right|^{1/2} = \frac{1}{\sigma} \propto 1$.

   (b) [5 points] show that given μ fixed, the Jeffery's prior over σ, $\pi_J(\sigma) \propto \frac{1}{\sigma}$.

Answer:

$$\begin{aligned}
I(\mu) &= -\mathbb{E}_\sigma\left[\frac{\mathrm{d}^2 \log p(x|\sigma)}{\mathrm{d}\sigma^2}\right] \\
&= -\mathbb{E}_\sigma\left[\frac{\mathrm{d}^2}{\mathrm{d}\sigma^2}\left(C - \log\sigma - \frac{1}{2\sigma^2}(x-\mu)^2\right)\right] \\
&= -\mathbb{E}_\sigma\left[\frac{1}{\sigma^2} - \frac{3}{\sigma^4}(x-\mu)^2\right] \\
&= -\left(\frac{1}{\sigma^2} - \frac{3}{\sigma^4}\mathbb{E}_\sigma\left[(x-\mu)^2\right]\right) \\
&= -\frac{1}{\sigma^2} + \frac{3}{\sigma^4}\sigma^2 = \frac{2}{\sigma^2}.
\end{aligned}$$

Therefore, $\pi_J(\mu) \propto \left|\frac{2}{\sigma^2}\right|^{1/2} \propto \frac{1}{\sigma}$.

2. [5 points] Derive the Jeffery's prior for λ in the Poisson likelihood, $p(x=n) = e^{-\lambda}\frac{\lambda^n}{n!}$.
Answer:

$$\begin{aligned}
I(\mu) &= -\mathbb{E}_\lambda\left[\frac{\mathrm{d}^2 \log p(x=n|\lambda)}{\mathrm{d}\lambda^2}\right] \\
&= -\mathbb{E}_\lambda\left[\frac{\mathrm{d}^2}{\mathrm{d}\sigma^2}\left(-\lambda + n\log\lambda - C\right)\right] \\
&= \mathbb{E}_\lambda\left[\frac{n}{\lambda^2}\right] = \frac{1}{\lambda^2}\mathbb{E}_\lambda\left[n\right] = \frac{1}{\lambda^2}\lambda = \frac{1}{\lambda}.
\end{aligned}$$

Therefore, $\pi_J(\lambda) \propto \left|\frac{1}{\lambda}\right|^{1/2} \propto \frac{1}{\sqrt{\lambda}}$.

3. [5 points] Given an infinite sequence of Independently Identically Distributed (IID) random variables, show that they are exchangeable.
Answer: For any finite subsets $x_1, \ldots, x_n$ of these IID random variables, using Bayes rule, we have

$$\begin{aligned}
P(x_1, \ldots, x_n) &= p(x_1)p(x_2|x_1)\cdots p(x_n|x_1, \ldots, x_{n-1}) \\
&= \prod_{i=1}^{n} p(x_i) \qquad \text{since } x_i\text{'s are independent}
\end{aligned}$$

Now, assume that $\sigma : [n] \longrightarrow [n] \in S_n$ is a permutation. Again,

$$\begin{aligned}
P(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) &= p(x_{\sigma(1)})p(x_{\sigma(2)}|x_{\sigma(1)})\cdots p(x_{\sigma(n)}|x_{\sigma(1)}, \ldots, x_{\sigma(n-1)}) \\
&= \prod_{i=1}^{n} p(x_{\sigma(i)}) \qquad \text{since } x_i\text{'s are independent and } \sigma \text{ is a permutation} \\
&= \prod_{i=1}^{n} p(x_i) \qquad \text{since } \sigma \text{ is a permutation} \\
&= P(x_1, \ldots, x_n).
\end{aligned}$$

4. [10 points] We discussed Polya's Urn problem as an example of exchangeability. If you do not recall, please look back at the slides we shared in the course website. Now, given two finite sequences $(0,1,0,1)$ and $(1,1,0,0)$, derive their probabilities and show they are the same.
Answer:

$$\begin{aligned}
P(0,1,0,1) &= p(x_1=0)p(x_2=1|x_1=0)p(x_3=0|x_1=0,x_2=1)p(x_4=1|x_1=0,x_2=1,x_3=0,x_4=1) \\
&= \frac{W_0}{B_0+W_0} \times \frac{B_0}{B_0+W_0+a-1} \times \frac{W_0+a-1}{B_0+W_0+2(a-1)} \times \frac{B_0+a-1}{B_0+W_0+3(a-1)} \\
&= \frac{B_0(B_0+a-1)W_0(W_0+a-1)}{\prod_{i=0}^{3}(B_0+W_0+i(a-1))}
\end{aligned}$$

$$
\begin{aligned}
P(1,1,0,0) &= p(x_1=1)p(x_2=1|x_1=1)p(x_3=0|x_1=1,x_2=1)p(x_4=0|x_1=0,x_2=1,x_3=1,x_4=0) \\
&= \frac{B_0}{B_0+W_0} \times \frac{B_0+a-1}{B_0+W_0+(a-1)} \times \frac{W_0}{B_0+W_0+2(a-1)} \times \frac{W_0+a-1}{B_0+W_0+3(a-1)} \\
&= \frac{B_0(B_0+a-1)W_0(W_0+a-1)}{\prod_{i=0}^{3}(B_0+W_0+i(a-1))} = P(0,1,0,1).
\end{aligned}
$$

5. [10 points] For the logistic regression model, we assign a Gaussian prior over the feature weights, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$. Please derive the Newton-Raphson updates.
Answer: Our objective is to find $\mathbf{w}$ maximizing

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{t}) &= p(\mathbf{w})p(\mathbf{t}|\mathbf{w},\mathbf{x}) = p(\mathbf{w})\prod_{i=1}^{n} p(\mathbf{t}_i|\mathbf{w},\mathbf{x}_i) \\
&= p(\mathbf{w})\prod_{i=1}^{n} y_n^{t_n}(1-y_n)^{1-t_n} \qquad \text{where } y_n = \sigma(\mathbf{w}^\top\phi_n).
\end{aligned}
$$

It is equvalent to find $\mathbf{w}$ minimizing

$$
\begin{aligned}
E(w) &= -\log p(\mathbf{w}|\mathbf{t}) \\
&= -\log p(\mathbf{w}) - \sum_{i=1}^{n}\left(t_n \log y_n + (1-t_n)\log(1-y_n)\right) \\
&= C + \frac{1}{2\lambda}\mathbf{w}^\top\mathbf{w} - \sum_{i=1}^{n}\Big(t_n \log y_n + (1-t_n)\log(1-y_n)\Big)
\end{aligned}
$$

Consequently, (in details, it was computed in the class!)

$$
\nabla E(\mathbf{w}) = \frac{1}{\lambda}\mathbf{w} + \sum_{i=1}^{n}(y_n - t_n)\phi_n = \frac{1}{\lambda}\mathbf{w} + \boldsymbol{\phi}^\top(\mathbf{y}-\mathbf{t})
$$

Therefore,

$$
\begin{aligned}
\mathbf{H} = \nabla^2 E(\mathbf{w}) &= \frac{1}{\lambda}\mathbf{I}_d + \sum_{i=1}^{n} y_n(1-y_n)\phi_n\phi_n^\top \\
&= \frac{1}{\lambda}\mathbf{I}_d + \boldsymbol{\phi}^\top\mathbf{R}\boldsymbol{\phi},
\end{aligned}
$$

where $\mathbf{R}$ is an $n \times n$ diagonal matrix whose (i,i) enrty is $y_n(1-y_n)$ and $\boldsymbol{\phi}$ is a $n \times d$ matrix whose i-th row is the i-th data point x_i. Finally, we can derive the Newton-Raphson updates as follows:

$$
\mathbf{w}^{old} = \mathbf{w}^{old} - \mathbf{H}^{-1}\left[\frac{1}{\lambda}\mathbf{w} + \boldsymbol{\phi}^\top(\mathbf{y}-\mathbf{t})\right]
$$

6. [**Bonus**][20 points] For the probit regression model, we assign a Gaussian prior over the feature weights, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$. Please derive the Newton-Raphson updates.
Answer: Our objective is to find $\mathbf{w}$ minimizing

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{t}) &= p(\mathbf{w})p(\mathbf{t}|\mathbf{w},\mathbf{x}) = p(\mathbf{w})\prod_{i=1}^{n} p(\mathbf{t}_i|\mathbf{w},\mathbf{x}_i) \\
&= p(\mathbf{w})\prod_{i=1}^{n} y_n^{t_n}(1-y_n)^{1-t_n} \qquad \text{where } y_n = \psi(\mathbf{w}^\top\phi_n) = \int_{-\infty}^{\mathbf{w}^\top\phi_n} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\,\mathrm{d}x.
\end{aligned}
$$

It is equvalent to find $\mathbf{w}$ maximizing

$$\begin{aligned} E(w) &= -\log p(\mathbf{w}|\mathbf{t}) \\ &= -\log p(\mathbf{w}) - \sum_{i=1}^{n} (t_n \log y_n + (1-t_n)\log(1-y_n)) \\ &= C + \frac{1}{2\lambda}\mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{N} \Big(t_n \log y_n + (1-t_n)\log(1-y_n)\Big). \end{aligned}$$

Using the Leibniz integral rule,

$$\frac{\partial y_n}{\partial \mathbf{w}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\mathbf{w}^\top \boldsymbol{\phi}_n)^2}{2}} \phi_n,$$

which implies

$$\begin{aligned} \nabla E(\mathbf{w}) &= \frac{1}{\lambda}\mathbf{w} - \sum_{i=1}^{N} \Big(\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\Big)\frac{\partial y_n}{\partial \mathbf{w}} \\ &= \frac{1}{\lambda}\mathbf{w} - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{N} \Big(\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\Big) e^{-\frac{(\mathbf{w}^\top \boldsymbol{\phi}_n)^2}{2}} \phi_n \\ &= \frac{1}{\lambda}\mathbf{w} + \sum_{i=1}^{N} \underbrace{\frac{1}{\sqrt{2\pi}} \Big(\frac{y_n - t_n}{y_n(1-y_n)}\Big) e^{-\frac{(\mathbf{w}^\top \boldsymbol{\phi}_n)^2}{2}}}_{=z_n} \phi_n \\ &= \frac{1}{\lambda}\mathbf{w} + \boldsymbol{\phi}^\top \mathbf{z}, \end{aligned}$$

where $\mathbf{z}$ is a $N \times 1$ vector whose i^{th} entry is z_n. Moreover,

$$\begin{aligned} \mathbf{H} = \nabla\nabla E(\mathbf{w}) &= \nabla \left(\frac{1}{\lambda}\mathbf{w} - \frac{1}{\sqrt{2\pi}} \sum_{n=1}^{N} \Big(\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\Big) e^{-\frac{(\mathbf{w}^\top \phi)^2}{2}} \phi_n \right) \\ &= \frac{1}{\lambda}\mathbf{I}_d - \frac{1}{2\pi} \sum_{n=1}^{N} \Big(\frac{-t_n}{y_n^2} - \frac{1-t_n}{(1-y_n)^2}\Big) e^{-(\mathbf{w}^\top \phi_n)^2} + \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{N} \Big(\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\Big)(w^\top \phi_n) e^{-\frac{(\mathbf{w}^\top \phi)^2}{2}} \phi_n^\top \phi_n \\ &= \frac{1}{\lambda}\mathbf{I}_d + \sum_{n=1}^{N} \left\{ \underbrace{\frac{1}{2\pi} \Big(\frac{t_n}{y_n^2} + \frac{1-t_n}{(1-y_n)^2}\Big) e^{-(\mathbf{w}^\top \phi_n)^2} + \frac{1}{\sqrt{2\pi}} \Big(\frac{-t_n}{y_n} + \frac{1-t_n}{1-y_n}\Big)(w^\top \phi_n) e^{-\frac{(\mathbf{w}^\top \phi)^2}{2}}}_{=r_n} \right\} \phi_n^\top \phi_n \\ &= \frac{1}{\lambda}\mathbf{I}_d + \boldsymbol{\phi}^\top \mathbf{R} \boldsymbol{\phi}, \end{aligned}$$

where R is a diagonal $N \times N$ matrix whose (n,n)-th value is R_n. Finally, we can derive the Newton-Raphson updates as follows:

$$\mathbf{w}^{old} = \mathbf{w}^{old} - \mathbf{H}^{-1} \left[\frac{1}{\lambda}\mathbf{w} + \boldsymbol{\phi}^\top \mathbf{z} \right]$$

7. [10 points] What are the link functions of the following models?

   (a) [5 points] Logistic regression **Answer:** The Logistic regression model is a binary model in which for a feature vector $\boldsymbol{\phi}$ and a weight vector $\mathbf{w}$,

$$p(t|\mathbf{w}, \boldsymbol{\phi}) = \sigma(\mathbf{w}^\top \boldsymbol{\phi})^t \big[1 - \sigma(\mathbf{w}^\top \boldsymbol{\phi})\big]^{1-t},$$

where $t \in \{0, 1\}$. Writting $p(t|\mathbf{w}, \boldsymbol{\phi})$ in a form of exponentioal family distribution member, we will have

$$\begin{aligned} p(t|\mathbf{w}, \boldsymbol{\phi}) &= \exp\left\{t \log \sigma(\mathbf{w}^\top \boldsymbol{\phi}) + (1-t) \log\left(1 - \sigma(\mathbf{w}^\top \boldsymbol{\phi})\right)\right\} \\ &= \left(1 - \sigma(\mathbf{w}^\top \boldsymbol{\phi})\right) \exp\left\{\log\left(\frac{\sigma(\mathbf{w}^\top \boldsymbol{\phi})}{1 - \sigma(\mathbf{w}^\top \boldsymbol{\phi})}\right) t\right\} \\ &= \sigma(-\mathbf{w}^\top \boldsymbol{\phi}) \exp\left\{(\mathbf{w}^\top \boldsymbol{\phi}) t\right\} \\ &\exp\left\{(\mathbf{w}^\top \boldsymbol{\phi}) t - \log \sigma(-\mathbf{w}^\top \boldsymbol{\phi})\right\} \end{aligned}$$

Comparison to the sandard form of exponentioal family distribution

$$p(t|\eta) = h(t) \exp\left\{\eta t - g(\eta)\right\},$$

we obtain

$$\mathbf{u}(t) = t \qquad h(t) = 1 \qquad \eta = \mathbf{w}^\top \boldsymbol{\phi} \qquad \text{and} \qquad g(\eta) = -\log(\sigma(-\eta)).$$

Therefore,

$$\begin{aligned} y = \mathbb{E}_\eta(t|\eta) = \frac{\partial g(\eta)}{\partial \eta} &= \frac{\sigma(-\eta)(1 - \sigma(-\eta))}{\sigma(-\eta)} \\ &= 1 - \sigma(-\eta) = \sigma(\eta) \end{aligned}$$

which implies $\eta = \sigma^{-1}(y) = \log(\frac{y}{1-y})$ is the link function.

(b) [5 points] Poisson regression: $p(x = n) = e^{-\lambda} \frac{\lambda^n}{n!}$ where $\lambda = \mathbf{w}^\top \boldsymbol{\phi}$. **Answer:** Rewritting Poisson distributions in the form of exponentioal family distributions, we obtain

$$p(x = n|\mathbf{w}, \phi) = e^{-\lambda} \frac{\lambda^n}{n!} = \frac{1}{n!} \exp\left\{n \log \lambda - \lambda\right\}.$$

Therefore,

$$h(x) = \frac{1}{x!} \qquad \mathbf{u}(x) = x \qquad \eta = \log \lambda = \log \mathbf{w}^\top \boldsymbol{\phi} \qquad \text{and} \qquad g(\eta) = \lambda = e^\eta = \mathbf{w}^\top \boldsymbol{\phi}.$$

Accordingly, $y = \mathbb{E}_\eta(x|\eta) = \frac{\partial g(\eta)}{\partial \eta} = e^\eta = \mathbf{w}^\top \boldsymbol{\phi}$ which implies $\eta = \log y$ is the link function.

8. [10 points] As we discussed in the class, the probit regression model is equivalent to given each feature vector $\boldsymbol{\phi}$, sampling a latent variable z from $\mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1)$, and then sampling the binary label t from the step distribution, $p(t|z) = \mathbb{1}(t = 0)\mathbb{1}(z < 0) + \mathbb{1}(t = 1)\mathbb{1}(z \geq 0)$ where $\mathbb{1}(\cdot)$ is the indicator function. Show that if we marginalize out z, we recover the original likelihood of the probit regression. **Answer:**

$$\begin{aligned} p(t|\phi) &= \int_{-\infty}^{+\infty} p(t, z|\mathbf{w}, \boldsymbol{\phi}) \mathrm{d}z = \int_{-\infty}^{+\infty} p(z|\mathbf{w}, \boldsymbol{\phi}) p(t|z) \mathrm{d}z \\ &= \int_{-\infty}^{+\infty} \mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1) \Big[\mathbb{1}(t = 0)\mathbb{1}(z \leq 0) + \mathbb{1}(t = 1)\mathbb{1}(z \geq 0)\Big] \mathrm{d}z \\ &= \mathbb{1}(t = 0) \int_{-\infty}^{0} \mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1) \mathrm{d}z + \mathbb{1}(t = 1) \int_{0}^{+\infty} \mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1) \mathrm{d}z \\ &= \mathbb{1}(t = 0) \int_{-\infty}^{-w^\top \boldsymbol{\phi}} \mathcal{N}(u|\mathbf{0}, 1) \mathrm{d}u + \mathbb{1}(t = 1) \int_{-w^\top \boldsymbol{\phi}}^{+\infty} \mathcal{N}(u|0, 1) \mathrm{d}u \qquad \text{changing variable } u = z - w^\top \boldsymbol{\phi} \\ &= \mathbb{1}(t = 0)(1 - \psi(\mathbf{w}^\top \boldsymbol{\phi})) + \mathbb{1}(t = 1)\left[\psi(\mathbf{w}^\top \boldsymbol{\phi})\right] \qquad \text{symetry of Normal distribution} \\ &= \psi(\mathbf{w}^\top \boldsymbol{\phi})^t \big(1 - \psi(\mathbf{w}^\top \boldsymbol{\phi})\big)^{1-t}. \end{aligned}$$

9. [**Bonus**][5 points] For polynomial regression (1d feature vector), show that given N training points, you can always choose the highest order M for the polynomial terms such that your model results in 0 training error (e.g., mean squared error or mean absolute error). Please give the corresponding regression function as well. **Answer:** Assume that our data set is $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. Define

$$P(x) = \sum_{i=1}^{N} y_i \underbrace{\prod_{j=1, j\neq i}^{N} \frac{x - x_j}{x_i - x_j}}_{=Q_i(x)}$$

Note that for each $i, k \in [N]$,

$$Q_i(x_k) = \begin{cases} 0 & i \neq k \\ 1 & i = k. \end{cases}$$

Thus, for each $k \in [N]$, $P(x_k) = \sum_{i=1}^{N} y_i Q_i(x_k) = y_k$. Accordingly, $P(x)$ is a polynomial of degree $N - 1$ which perfectly fits the data.

Practice [40 points + 45 Bonus]

1. [15 Points] Let us generate a simulation dataset for fun. We consider a linear regression model $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x$. We set the ground-truth $w_0 = -0.3$ and $w_1 = 0.5$. We generate 20 samples $[x_1, \ldots, x_{20}]$ from the uniform distribution in $[-1, 1]$. For each sample x_n, we obtain an sample y_n by first calculating $w_0 + w_1 x_n$ with the ground-truth values of w_0 and w_n, and then adding a Gaussian noise with zero mean, standard deviation 0.2. Now let us verify what we have discussed in the class. We use a Bayesian linear regression model. The prior of $\mathbf{w}$ is $\mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha\mathbf{I})$, and the likelihood for each sample is $p(y_n|\mathbf{x}_n, \mathbf{w}) = \mathcal{N}(y_n|w_0 + w_1 x, \beta^{-1}\mathbf{I})$. Here we set $\alpha = 2$ and $\beta = 25$.

   (a) [3 points] Draw the heat-map of the prior $p(\mathbf{w})$ in the region $w_0 \in [-1, 1]$ and $w_1 \in [-1, 1]$, where you represent the values of $p(\mathbf{w})$ for different choices of $\mathbf{w}$ with different colors. The darker some given color (e.g., red), the larger the value; the darker some the other given color (e.g., blue), the smaller the value. Most colors should be in between. Then sample 20 instances of $\mathbf{w}$ from $p(\mathbf{w})$. For each w, draw a line $y = w_0 + w_1 x$ in the region $x, y \in [-1, 1]$. Ensure these 20 lines are in the same plot. What do you observe?
   Answer: The prior contours are circles as it is a Gaussian distribution with zero mean and multiplication of identity matrix as its covariance matrix. Also, in the region we plot the 20 lines, these lines seems have be completely random slopes.

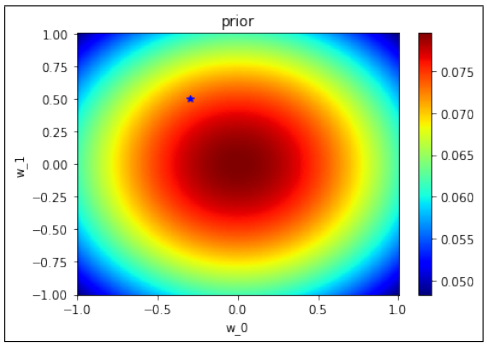

Figure 1: Prior

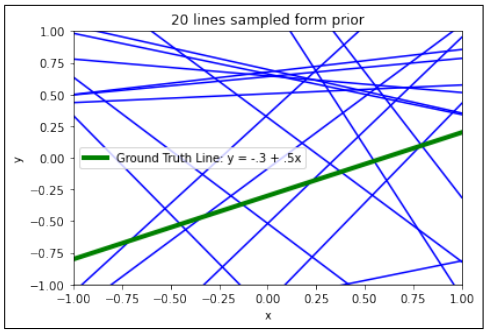

Figure 2: 20 lines whose coefficients are sampled from prior

(b) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $(\mathbf{x}_1, y_1)$. Now draw the heat map of the distribution. Also draw the ground-truth of w_0 and w_1 in the heat map. Then from the posterior distribution, sample 20 instances of $\mathbf{w}$, for each of which draw a line $y = w_0 + w_1 x$ in the region $x, y \in [-1, 1]$. Ensure these 20 lines are in the same plot. Also draw (x_1, y_1) as a circle in that plot. What do you observe? Why?

Answer: The posterior distribution is gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = [-0.09630267, -0.01547873] \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.03684579 & -0.02622379 \\ -0.02622379 & 0.19578505 \end{bmatrix}.$$

The posterior distribution is more concentrated compared to prior distribution and moreover, its contours are ellipsoid affected by the observed data point (x_1, y_1). Also, the 20 sample lines all are tend to pass through to the data point (x_1, y_1), they all pass close to this point.

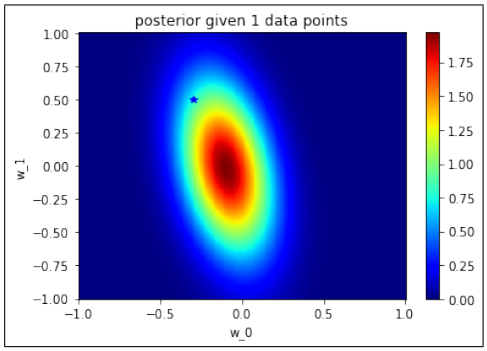

Figure 3: Posterior distribution given (x_1, y_1)

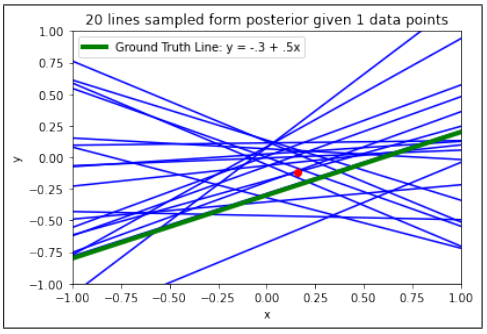

Figure 4: 20 lines whose coefficients are sampled from posterior given (x_1, y_1)

(c) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$. Then draw the plots as the above. What do you observe now?

Answer: The posterior distribution is gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = [-0.05922021, 0.20500141] \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.03430475 & -0.04133198 \\ -0.04133198 & 0.10595668 \end{bmatrix}.$$

The posterior mean is getting closer to the ground truth $(-0.3, 0.5)$ and it is now even more concentrated compared to the previous step because of the observed data points $(x_1, y_1), (x_2, y_2)$. The 20 sample lines all tend to pass through to the two data points $(x_1, y_1), (x_2, y_2)$, they all pass close these points. Also, these lines are now somehow concentrated around the ground truth line.

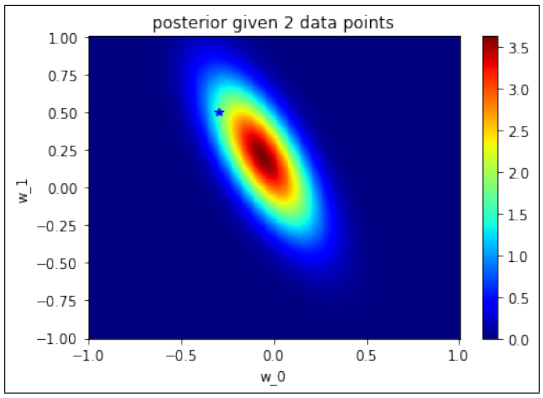

Figure 5: Posterior distribution given (x_1, y_1) and (x_2, y_2)

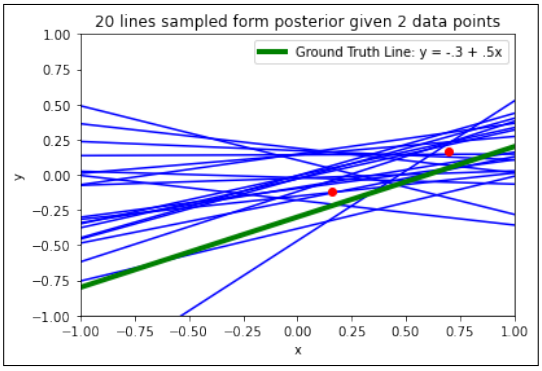

Figure 6: 20 lines whose coefficients are sampled from from posterior given (x_1, y_1) and (x_2, y_2)

(d) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_5, y_5)\}$. Then draw the plots as the above. What do you observe now?
Answer: The posterior distribution is gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = [-0.14286727, 0.42610391] \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.00817504 & -0.00511181 \\ -0.00511181 & 0.05413014 \end{bmatrix}.$$

Similarly, the posterior mean is getting, even more, closer to the ground truth $(-0.3, 0.5)$ and it is now even more concentrated compared to the previous step because of the observed data points $(x_1, y_1), \ldots, (x_5, y_5)$. The 20 sample lines are now really concentrated around the ground truth line.

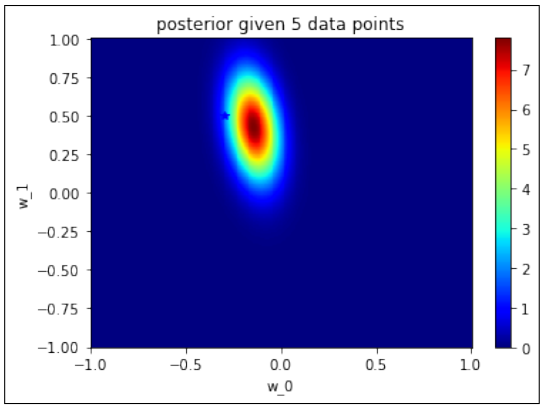

Figure 7: Posterior distribution given $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_5, y_5)\}$

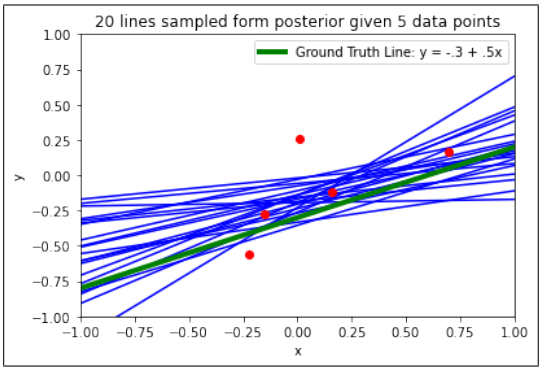

Figure 8: 20 lines whose coefficients are sampled from posterior given $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_5, y_5)\}$

(e) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given all the 20 data points. Then draw the plots as the above. What do you observe now?
Answer: The posterior distribution is gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = [-0.23654566, 0.45238796] \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.00204578 & -0.00065104 \\ -0.00065104 & 0.00646252 \end{bmatrix}.$$

Similarly, the posterior mean is now really close to the ground truth $(-0.3, 0.5)$ and posterior is now very concentrated around its mean because of the observed data points $(x_1, y_1), \ldots, (x_{20}, y_{20})$. The 20 sample lines are now completely concentrated around the ground truth line. Indeed, each of them can be a very good approximation of the ground truth line.

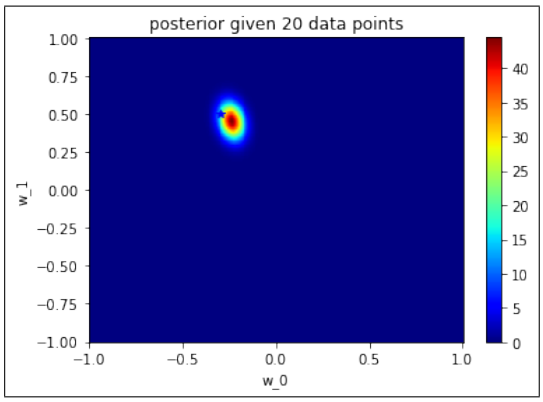

Figure 9: Posterior distribution given all the 20 data points

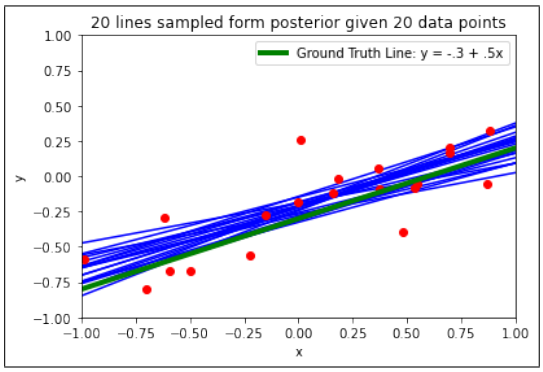

Figure 10: 20 lines whose coefficients are sampled from posterior given all the 20 data points

2. [25 points] We will implement Logistic regression and Probit regression for a binary classification task — bank-note authentication. Please download the data "bank-note.zip" from Canvas. The features and labels are listed in the file "bank-note/data-desc.txt". The training data are stored in the file "bank-note/train.csv", consisting of 872 examples. The test data are stored in "bank-note/test.csv", and comprise of 500 examples. In both the training and testing datasets, feature values and labels are separated by commas. To ensure numerical stability and avoid overfitting, we assign the feature weights a standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

   (a) [15 points] Implement Newton-Raphson scheme to find the MAP estimation of the feature weights in the logistic regression model. Set the maximum number of iterations to 100 and the tolerance level to be $1e-5$, i.e., when the norm of difference between the weight vectors after one update is below the tolerance level, we consider it converges and stop updating the weights any more. Initially, you can set all the weights to be zero. Report the prediction accuracy on the test data. Now set the initial weights values be to be randomly generated, say, from the standard Gaussian, run and test your algorithm. What do you observe? Why?
   Answer: Prediction accuracy on the test data when we initiated all the weights to be zero is 99%. But when the values of the initial weights are randomly generated from the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, the model is unstable and the accuracy sometimes drops down to even 50% and sometimes we get the same accuracy of 99%. The reason might be that when we randomly generate w, it is possible that $|w^\top \phi_n|$ is a large number (for example more than 10) which makes either $y_n = 0$ or $y_n = 1$. This vanishes the gradient since the sigmoid function has almost zero gradients for numbers with large absolute values. It causes almost no meaningful updates to w. We tried some ideas to overcome this issue. First, we randomly generated w from the standard Gaussian with zero mean and $10^{-5}\mathbf{I}$ covariance. This method works almost all the time. The other idea is that, since multiplicating w by a positive number has no effect on the model accuracy, during training we can always keep tracking the L_2-norm of w and when it is larger than 1, we can project it on the unit ball. This method works pretty well.

   (b) [10 points] Implement MAP estimation algorithm for Probit regression model. You can calculate the gradient and feed it to any optimization algorithm, say, L-BFGS. Set the maximum number of iterations to 100 and the tolerance level to $1e-5$. Initially, you can set all the weights to zero. Report the prediction accuracy on the test data. Compared with logistic regression, which

one is better? Now set the initial weights values be to be randomly generated, say, from the standard Gaussian, run and test your algorithm. What do you observe? Can you guess why?
Answer: The prediction accuracy on the test data when we initiated all the weights to be zero is 98.8%. Compared with the logistic regression model, we can conclude that the logistic regression performs a bit better rather than Probit regression model. Using L-BFGS, it seems that the model is stable even when we randomly initiate w where we obtain the same accuracy of 98.8%. Probably, since L-BFGS is a built-in model, it takes care of the problems we encountered in the previous part.

(c) [**Bonus**][15 points]. Implement Newton-Raphson scheme to find the MAP estimation for Probit regression. Report the prediction accuracy.
Answer: We implimented Newton-Raphson scheme. The prediction accuracy on the test data when we initiated all the weights to be zeros is 98.8% (see the end of uploaded file HW2Q2.ipynb). Again here we check the randomized initialization and we have some sort of the problems we described in Part(a). Here also computing the Hessian matrix is numerically unstable since we have y_n and $1 - y_n$ in the denominator and when the initial weights are random, as described in Part (a), most likely either $y_n = 0$ or $y_n = 1$. However, using the same methods as described in Part (a), we could get a stable model resulting in the same accuracy of 98.8%.

3. [**Bonus**][30 points] We will implement a multi-class logistic regression model for car evaluation task. The dataset is from UCI repository(`https://archive.ics.uci.edu/ml/datasets/car+evaluation`). Please download the processed dataset (car.zip) from Canvas. In this task, we have 6 car attributes, and the label is the evaluation of the car. The attribute and label values are listed in the file "data-desc.txt". All the attributes are categorical. Please convert each categorical attribute into binary features. For example, for "safety: low, med, high", we convert it into three binary features: "safety" is "low" or not, "safety" is "med" or not, and "safety" is "high" or not. The training data are stored in the file "train.csv", consisting of $1,000$ examples. The test data are stored in "test.csv", and comprise 728 examples. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column. To ensure numerical stability and avoid overfitting, we assign the feature weights a standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

(a) [15 points] Implement MAP estimation algorithm for multi-class logistic regression model. To do so, you can calculate the gradient and feed it to some optimization package, say, L-BFGS. Report the prediction accuracy on the test data.
Answer: Using gradient descent to train, either with zero or random initialization, the prediction accuracy on the test data is: 91.7%. Also, using L-BFGS method, either with zero or random initialization, the prediction accuracies on the test data are: 90.7% and 90.1% respectively.

(b) [15 points] Let us use an "ugly" trick to convert the multi-class classification problem into a binary classification problem. Let us train four logistic regression models, where each model predicts one particular label, i.e., "unacc" or not, "acc" or not, "good" or not, and "vgood" or not. Then for each test example, we run the models to get four logistic scores, i.e., the probability that each label is one. We choose the label with the highest score as the final prediction. Report the prediction accuracy on the test data. As compared with multi-class logistic regression,which one is better?
Answer: Using gradient descent to train, either with zero or random initialization, the prediction accuracy on the test data is : 85.3%. However, we need to initiate the random weights to be very small, say from $\mathcal{N}(0, 10^{-4})$.

Also, using L-BFGS method, either with zero or random initialization, the prediction accuracy on the test data is : 86.1%.