CS 6190: Probabilistic Modelling Spring 2019

Homework 2

Handed out: 23 Sep, 2019
Due: 11:59pm, 05 Oct, 2019

- 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 + 55 bonus]

1. [10 points] Given a Gaussian likelihood, $p(x|\mu,\sigma) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\big(-\frac{1}{2\sigma^2}(x-\mu)^2\big)$,

   (a) [5 points] show that given σ fixed, the Jeffery's prior over μ, $\pi_J(\mu) \propto 1$;

$$\log p(x|\mu,\sigma) = -\frac{1}{2}\log(2\pi) - \log\sigma - \frac{1}{2\sigma^2}(x-\mu)^2$$
$$\frac{d}{d\mu}\log p(x|\mu,\sigma) = \frac{1}{\sigma^2}(x-\mu)$$
$$\frac{d^2}{d\mu^2}\log p(x|\mu,\sigma) = -\frac{1}{\sigma^2}$$
$$\mathbf{I}(\mu) = -\mathbb{E}_{p(x|\mu,\sigma)}\left[-\frac{1}{\sigma^2}\right] = \frac{1}{\sigma^2} = \text{constant}$$
$$\pi_J(\mu) \propto \mathbf{I}(\mu)^{1/2} \propto 1$$

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

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

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

$$\begin{aligned}
\log p(x|\lambda) &= -\lambda + x\log\lambda - \log(\mathbf{x}!) \\
\frac{d}{d\lambda}\log p(x|\lambda) &= -1 + \frac{x}{\lambda} \\
\frac{d^2}{d\lambda^2}\log p(x|\lambda) &= -\frac{x}{\lambda^2} \\
\mathbf{I}(\lambda) &= -\mathbb{E}_{p(x|\lambda)}\left[-\frac{x}{\lambda^2}\right] = \frac{1}{\lambda^2}\mathbb{E}_{p(x|\lambda)}[x] = \frac{1}{\lambda^2}\lambda = \frac{1}{\lambda} \\
\pi_J(\lambda) \propto \mathbf{I}(\lambda)^{1/2} &\propto \frac{1}{\sqrt{\lambda}} \qquad (1)
\end{aligned}$$

3. [5 points] Given an infinite sequence of Independently Identically Distributed (IID) random variables, show that they are exchangeable.

Let $x_1, x_2, ...$ be the sequence of iid random variables. Then we know that,

$$p(x_1, x_2, ...) = p(x_1)p(x_2)...p(x_n)...$$

Now for any permutation π ,

$$\begin{aligned}
p(x_1, x_2, ...) &= p(x_{\pi_1}, x_{\pi_2}, ..., x_{\pi_n}...) \\
&= p(x_{\pi_1})p(x_{\pi_2})... \\
&= p(x_1)p(x_2)... \quad \text{using the iid property}
\end{aligned}$$

This shows that an iid sequence is always exchangeable

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.

Let us assume that $x = 0$ refers to drawing a white ball and $x = 1$ refers to drawing a black

ball. Also, assume that initially there were B_0 and W_0 black and white balls respectively

$$\begin{aligned}
p(0,1,0,1) &= \frac{W_0}{B_0+W_0} \times \frac{B_0}{B_0+(W_0+1)} \times \frac{W_0+1}{(B_0+1)+(W_0+1)} \times \frac{B_0+1}{(B_0+1)+(W_0+2)} \\
&= \frac{W_0 B_0 (W_0+1)(B_0+1)}{(B_0+W_0)(B_0+W_0+1)(B_0+W_0+2)(B_0+W_0+3)} \\
p(1,1,0,0) &= \frac{B_0}{B_0+W_0} \times \frac{B_0+1}{(B_0+1)+W_0} \times \frac{W_0}{(B_0+2)+W_0} \times \frac{W_0+1}{(B_0+2)+(W_0+1)} \\
&= \frac{W_0 B_0 (W_0+1)(B_0+1)}{(B_0+W_0)(B_0+W_0+1)(B_0+W_0+2)(B_0+W_0+3)} \\
&= p(0,1,0,1)
\end{aligned} \tag{2}$$

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.

let $y_n = \sigma(\mathbf{w}^\top \mathbf{x})$ and $\mathbf{X} \in \mathbf{R}^{n \times d}$

$$\begin{aligned}
\frac{dy_n}{d\mathbf{w}} &= \sigma(\mathbf{w}^\top \mathbf{x})(1-\sigma(\mathbf{w}^\top \mathbf{x}))\mathbf{x} = y_n(1-y_n)\mathbf{x} \\
p(\mathbf{w}|\mathbf{t},\mathbf{X}) &= p(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w}) = \left(\prod_{i=1}^{n} y_n^{t_n}(1-y_n)^{1-t_n}\right) \frac{1}{(2\pi)^{d/2}\lambda^{1/2}} exp(\frac{-\mathbf{w}^\top\mathbf{w}}{2\lambda}) \\
\log p(\mathbf{w}|\mathbf{t},\mathbf{X}) &= \sum_{i=1}^{n} (t_n \log y_n + (1-t_n)\log(1-y_n)) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log\lambda - \frac{\mathbf{w}^\top\mathbf{w}}{2\lambda} \\
\frac{d}{d\mathbf{w}}(-\log p(\mathbf{w}|\mathbf{t},\mathbf{X})) &= -\sum_{i=1}^{n}(t_n - y_n)\mathbf{x} + \frac{\mathbf{w}}{\lambda} \\
&= -\mathbf{X}^\top(\mathbf{t}-\mathbf{y}) + \frac{\mathbf{w}}{\lambda} \\
\frac{d^2}{d\mathbf{w}^2}(-\log p(\mathbf{w}|\mathbf{t},\mathbf{X}) &= \sum_{i=1}^{n} y_n(1-y_n)\mathbf{x}_n\mathbf{x}_n^\top + \frac{\mathbf{I}}{\lambda} \\
&= \mathbf{X}^\top\mathbf{R}\mathbf{X} + \frac{\mathbf{I}}{\lambda} \quad \text{where } \mathbf{R} \text{ is a diagnoal matrix , } R_{ii} = y_i(1-y_i)
\end{aligned}$$

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\mathbf{X}^\top\mathbf{R}\mathbf{X} + \frac{\mathbf{I}}{\lambda}\right)^{-1}\left(\mathbf{X}^\top(\mathbf{y}-\mathbf{t}) + \frac{\mathbf{w}}{\lambda}\right)$$

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.

In probit regrssion, $y_n = \psi(a_n) = \int_{-\infty}^{a_n} \mathcal{N}(x|0,1)dx$ where $a_n = \mathbf{w}^\top\phi(\mathbf{x}_n)$

$$\begin{aligned}
\frac{dy_n}{d\mathbf{w}} &= \frac{exp(-a_n^2/2)}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^\top \\
\frac{d^2y_n}{d\mathbf{w}^2} &= -\frac{exp(-a^2/2)}{\sqrt{2\pi}}\phi(\mathbf{x}_n)\mathbf{w}^\top\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^\top
\end{aligned} \tag{3}$$

$$p(\mathbf{w}|\mathbf{t},\mathbf{X}) = p(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w}) = \left(\prod_{i=1}^{n} y_n^{t_n}(1-y_n)^{1-t_n}\right)\frac{1}{(2\pi)^{d/2}\lambda^{1/2}}exp(\frac{-\mathbf{w}^\top\mathbf{w}}{2\lambda})$$

$$\log p(\mathbf{w}|\mathbf{t},\mathbf{X}) = \sum_{i=1}^{n}(t_n\log y_n + (1-t_n)\log(1-y_n)) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log\lambda - \frac{\mathbf{w}^\top\mathbf{w}}{2\lambda}$$

$$-\log p(\mathbf{w}|\mathbf{t},\mathbf{X}) = -\sum_{i=1}^{n}(t_n\log y_n + (1-t_n)\log(1-y_n)) + \frac{\mathbf{w}^\top\mathbf{w}}{2\lambda} + \text{const.}$$

$$\frac{d}{d\mathbf{w}}(-\log p(\mathbf{w}|\mathbf{t},\mathbf{X})) = -\sum_{i=1}^{n}\frac{t_n-y_n}{y_n(1-y_n)}\frac{dy_n}{d\mathbf{w}} + \frac{\mathbf{w}^\top}{\lambda} = -\sum_{i=1}^{n}\frac{t_n-y_n}{y_n(1-y_n)}\frac{exp(-a^2/2)}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^\top + \frac{\mathbf{w}^\top}{\lambda}$$

$$\frac{d^2}{d\mathbf{w}^2}(-\log p(\mathbf{w}|\mathbf{t},\mathbf{X})) = \sum_{i=1}^{n}\left(\left(\frac{1-t_n}{(1-y_n)^2} + \frac{t_n}{y_n^2}\right)\left(\frac{dy_n}{d\mathbf{w}}\right)^2 + \left(\frac{y_n-t_n}{y_n(1-y_n)}\right)\frac{d^2y_n}{d\mathbf{w}^2}\right) + \frac{\mathbf{I}}{\lambda}$$

$$\frac{d^2}{d\mathbf{w}^2}(-\log p(\mathbf{w}|\mathbf{t},\mathbf{X})) = \frac{\mathbf{I}}{\lambda} + \sum_{i=1}^{n}\frac{t_n-y_n}{y_n(1-y_n)}\frac{exp(-a^2/2)}{\sqrt{2\pi}}\phi(\mathbf{x}_n)^\top\mathbf{w}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^\top +$$
$$\left(\frac{1-t_n}{(1-y_n)^2} + \frac{t_n}{y_n^2}\right)\left(\frac{exp(-a^2/2)}{\sqrt{2\pi}}\right)^2\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^\top$$
$$= \Phi^\top\mathbf{R}\Phi + \frac{\mathbf{I}}{\lambda} \tag{4}$$

where $\mathbf{R}$ is a diagonal matrix defined as : $R_{ii} = \frac{t_n-y_n}{y_n(1-y_n)}\frac{exp(-a^2/2)}{\sqrt{2\pi}}\mathbf{w}^\top\phi(\mathbf{x}_n) + \left(\frac{1-t_n}{(1-y_n)^2} + \frac{t_n}{y_n^2}\right)\left(\frac{exp(-a^2/2)}{\sqrt{2\pi}}\right)^2$
The Newton-Raphson update equation is given as :

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\Phi^\top\mathbf{R}\Phi + \frac{\mathbf{I}}{\lambda}\right)^{-1}\left(\mathbf{X}^\top\mathbf{u} + \frac{\mathbf{w}}{\lambda}\right)$$

where u is a vector such that $u_n = \frac{t_n-y_n}{y_n(1-y_n)}\frac{exp(-a^2/2)}{\sqrt{2\pi}}$

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

   (a) [5 points] Logistic regression

$$\eta = \psi(y) = \psi(\sigma(\mathbf{w}^\top\mathbf{x}))$$

For logistic regression, the link function $\psi = \sigma^{-1}$

$$y = \sigma(x) = \left(\frac{1}{1+e^{-x}}\right)$$
$$e^{-x} = \frac{1}{y} - 1$$
$$x = \log\left(\frac{y}{1-y}\right)$$
$$\sigma^{-1}(x) = \log\left(\frac{x}{1-x}\right)$$

(b) [5 points] Poisson regression: $p(x = n) = e^{-\lambda}\frac{\lambda^n}{n!}$ where $\lambda = \exp(\mathbf{w}^\top \boldsymbol{\phi})$.

$$
\begin{aligned}
p(x = n) &= e^{-\lambda}\frac{\lambda^n}{n!} \\
\mathbb{E}(x) &= \lambda \\
\lambda &= \exp(\mathbf{w}^\top \boldsymbol{\phi}) = f(\mathbf{w}^\top \boldsymbol{\phi}) \\
\log \lambda &= \mathbf{w}^\top \boldsymbol{\phi} = f^{-1}(\lambda) \\
\eta &= \psi(\lambda) = f^{-1}(f(\mathbf{w}^\top \boldsymbol{\phi})) = \mathbf{w}^\top \boldsymbol{\phi} \\
\psi(\lambda) &= f^{-1}(\lambda) = \log \lambda \quad \text{where } \psi \text{ is the link function}
\end{aligned}
$$

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, $p(t|z) = I(t = 0)I(z < 0) + I(t = 1)I(z \geq 0)$. Show if we marginalize out z, we recover the original likelihood of the probit regression.

$$
\begin{aligned}
p(t = 1|z) &= I(t = 1)I(z \geq 0) \\
p(t = 1) &= \int_0^\infty \mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1)dz \\
&= \int_0^\infty \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(z - \mathbf{w}^\top \boldsymbol{\phi})^2}{2}\right) dz \qquad (5)
\end{aligned}
$$

Let $z = y + \mathbf{w}^\top \boldsymbol{\phi})$ which gives $dz = dy$

$$
\begin{aligned}
p(t = 1) &= \int_{-\mathbf{w}^\top \boldsymbol{\phi}}^\infty \frac{1}{\sqrt{2\pi}} exp\left(-\frac{y^2}{2}\right) dy \\
&= \int_{-\mathbf{w}^\top \boldsymbol{\phi}}^\infty \mathcal{N}(y|0, 1)dy \\
&= \int_{-\infty}^{\mathbf{w}^\top \boldsymbol{\phi}} \mathcal{N}(y|0, 1)dy \qquad (1) \\
&= \psi(\mathbf{w}^\top \boldsymbol{\phi}) \qquad (6)
\end{aligned}
$$

In equation (1), the symmetric nature of gaussian distribution is used. The area of the Gaussian curve between $[-\infty, \mathbf{w}^\top \boldsymbol{\phi}]$ and $[-\mathbf{w}^\top \boldsymbol{\phi}, \infty]$ will be same. Similarly for $p(t = 0)$,

$$
\begin{aligned}
p(t = 0) &= \int_{-\infty}^0 \mathcal{N}(z|\mathbf{w}^\top \boldsymbol{\phi}, 1)dz \\
&= \int_{-\infty}^{-\mathbf{w}^\top \boldsymbol{\phi}} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{y^2}{2}\right) dy \\
&= 1 - \int_{-\infty}^{\mathbf{w}^\top \boldsymbol{\phi}} \mathcal{N}(y|0, 1)dy \quad \text{symmetric nature of gaussian distribution} \qquad (7) \\
&= 1 - \psi(\mathbf{w}^\top \boldsymbol{\phi}) \qquad (8)
\end{aligned}
$$

Thus, by marginalising z, the likelihood function can be written as :

$$
p(t|\mathbf{w}, \boldsymbol{\phi}) = \prod_i \psi(\mathbf{w}^\top \boldsymbol{\phi})^{t_i}(1 - \psi(\mathbf{w}^\top \boldsymbol{\phi}))^{1-t_i} \qquad (9)
$$

9. [**Bonus**][5 points] For polynomial regression, 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.

$$y_n = \beta_0 + \beta_1 x_n + \beta_2 {x_n}^2 + ... + \beta_k {x_n}^k \tag{1}$$

The training loss is given by : $\sum_i (t_n - y_n)^2$. To have a training error zero, we need for each data point x_i, $(t_i - y_i) = 0$. This implies that all the data points (x_i, t_i) should lie on the curve y. Thus, for the curve y, all the n data points should satisfy the equation (1). With n different points lying on a curve, we can always learn a polynomial of degree $n-1$ such that the training error is zero.

Let $\mathbf{X} = \begin{bmatrix} 1 & 1 & ... & 1 \\ x_1 & x_2 & ... & x_n \\ x_1^2 & x_2^2 & ... & x_n^2 \\ ... & ... & ... & ... \\ x_1^{n-1} & x_2^{n-1} & ... & x_n^{n-1} \end{bmatrix}$ and

$\beta = \begin{bmatrix} \beta_0 & \beta_1 & ... & \beta_n \end{bmatrix}^\top$

then we can write the model with zero training error as follows :

$$\mathbf{t} = \mathbf{X}^\top \beta$$
$$\beta = {\mathbf{X}^\top}^{-1} \mathbf{t} \tag{10}$$

10. [**Bonus**][30 points] Consider the decision problem for continuous variables. Given the decision variable t and input variable $\mathbf{x}$, we know $p(t, \mathbf{x})$, and we want to make an optimal decision $y(\mathbf{x})$ to predict t. To this end, we need to define a loss $L(t, y(\mathbf{x}))$ and then find $y(\mathbf{x})$ by minimizing

$$\mathbb{E}[L] = \int \int L(t, y(\mathbf{x})) p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t.$$

(a) [5 points] We mentioned in the class that if we choose the square loss, $L(t, y(\mathbf{x})) = (y(\mathbf{x}) - t)^2$, the optimal decision is given by $y(\mathbf{x}) = \mathbb{E}(t|\mathbf{x})$. We did not show it in the lecture. Show it now.

$$\begin{aligned} \mathbb{E}[L] &= \int \int (y(\mathbf{x}) - t)^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t \\ &= \int \left(\int (y(\mathbf{x}) - t)^2 p(t|\mathbf{x}) \mathrm{d}t \right) p(\mathbf{x}) \mathrm{d}\mathbf{x} \end{aligned}$$

To minimise $\mathbb{E}[L]$, for all values of $\mathbf{x}$, $\int (y(\mathbf{x}) - t)^2 p(t|\mathbf{x}) \mathrm{d}t$ should be minimum.

$$\begin{aligned} u &= \int (y(\mathbf{x}) - t)^2 p(t|\mathbf{x}) \mathrm{d}t \\ &= \int y(\mathbf{x})^2 p(t|\mathbf{x}) \mathrm{d}t + \int t^2 p(t|\mathbf{x}) \mathrm{d}t - \int 2y(\mathbf{x}) t p(t|\mathbf{x}) \mathrm{d}t \\ &= y(\mathbf{x})^2 + \int t^2 p(t|\mathbf{x}) \mathrm{d}t - 2y(\mathbf{x}) \int t p(t|\mathbf{x}) \mathrm{d}t \end{aligned}$$

We need to make an optimal decision $y(\mathbf{x})$ such that u is minimum.

$$\frac{du}{dy(\mathbf{x})} = 2y(\mathbf{x}) - 2 \int t p(t|\mathbf{x}) \mathrm{d}t = 0$$
$$y(\mathbf{x}) = \int t p(t|\mathbf{x}) \mathrm{d}t = \mathbb{E}(t|\mathbf{x})$$

(b) [10 points] Let us define a more general loss, $L(t, y(\mathbf{x})) = |y(\mathbf{x}) - t|^q$. Show that when $q = 1$ (mean absolute error), the optimal decision is the conditional median of t given $\mathbf{x}$. Given a random variable Y, its median is the value θ such that $p(Y \leq \theta) = p(Y \geq \theta)$. *Hint: please refer to PRML book page 703, Appendix D, for functional derivative.*

$$\begin{aligned}\mathbb{E}[L] &= \int\int |y(\mathbf{x}) - t| p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t \\ &= \int \left(\int |y(\mathbf{x}) - t| p(t|\mathbf{x}) \mathrm{d}t \right) p(\mathbf{x}) \mathrm{d}\mathbf{x}\end{aligned}$$

To minimise $\mathbb{E}[L]$, for all values of $\mathbf{x}$, $\int |y(\mathbf{x}) - t| p(t|\mathbf{x}) \mathrm{d}t$ should be minimum.

$$\begin{aligned}u &= \int |y(\mathbf{x}) - t| p(t|\mathbf{x}) \mathrm{d}t \\ &= \int_{t<y(\mathbf{x})} (y(\mathbf{x}) - t) p(t|\mathbf{x}) \mathrm{d}t - \int_{t>y(\mathbf{x})} (y(\mathbf{x}) - t) p(t|\mathbf{x}) \mathrm{d}t \\ &= y(\mathbf{x}) \left(\int_{t<y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t - \int_{t>y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t \right) + \left(\int_{t>y(\mathbf{x})} t p(t|\mathbf{x}) \mathrm{d}t - \int_{t<y(\mathbf{x})} t p(t|\mathbf{x}) \mathrm{d}t \right)\end{aligned}$$

for minimum u, $\frac{du}{dy(\mathbf{x})} = 0$:

$$\begin{aligned}\int_{t<y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t - \int_{t>y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t &= 0 \\ \int_{t<y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t = \int_{t>y(\mathbf{x})} p(t|\mathbf{x}) \mathrm{d}t \quad &(1)\end{aligned}$$

The equation(1) will be true only when $y(\mathbf{x})$ will be the conditional median of $p(t|\mathbf{x})$

(c) [15 points] Show that when $q = 0$, the optimal decision is the conditional mode (i.e., MAP estimation).
when $q = 0$, $L(t, y(\mathbf{x})) = |y(\mathbf{x}) - t|^0 = \mathbf{1}(y(\mathbf{x}) = t)$ where $\mathbf{1}$ is the indicator function.

$$\begin{aligned}\mathbb{E}[L] &= \int\int |y(\mathbf{x}) - t|^0 p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t \\ &= \int\int_{y(\mathbf{x})=t} \mathbf{1}(y(\mathbf{x}) = t) p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t + \int\int_{y(\mathbf{x})\neq t} \mathbf{1}(y(\mathbf{x}) = t) p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t \\ &= \int\int p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t - \int\int_{y(\mathbf{x})=t} \mathbf{1}(y(\mathbf{x}) = t) p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t\end{aligned} \quad (11)$$

$\mathbb{E}[L]$ will be minimum when $\int\int_{y(\mathbf{x})=t} \mathbf{1}(y(\mathbf{x}) = t) p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t$ will be maximum.
$\int\int_{y(\mathbf{x})=t} \mathbf{1}(y(\mathbf{x}) = t) p(\mathbf{x}, t) \mathrm{d}\mathbf{x}\mathrm{d}t$ will be maximum when $y(\mathbf{x})$ is the conditional mode as for mode $p(x, t)$ will be maximum.

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 $\mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$, 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?

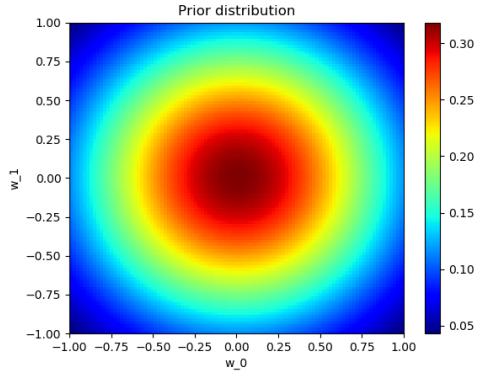
(b) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $(\mathbf{x}_1, y_1)$. Now draw the heat map of the distribution. Also draw the ground-truth of w_0 and w_1 in the heat map. Then from the posterior distribution, sample 20 instances of $\mathbf{w}$, for each of which 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?

(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?

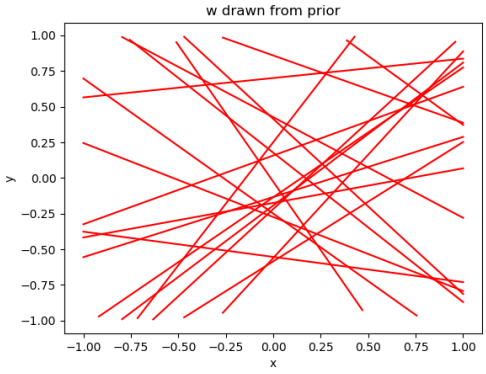
(d) [3 points] Calculate and report the posterior distribution of $\mathbf{w}$ given $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_5, y_5)\}$. Then draw the plots as the above. What do you observe now?

(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?

- All the weights generated from gaussian distribution $\mathcal{N}(0, \alpha^{-1}\mathbf{I})$ are equally likely as no data is seen.



(a) Prior($\mathbf{w}$)

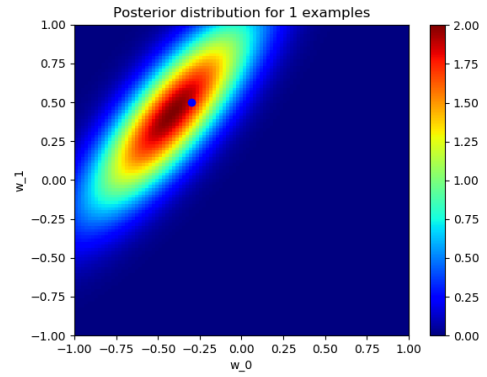


(b) Samples of $\mathbf{w}$ drawn from prior($\mathbf{w}$)

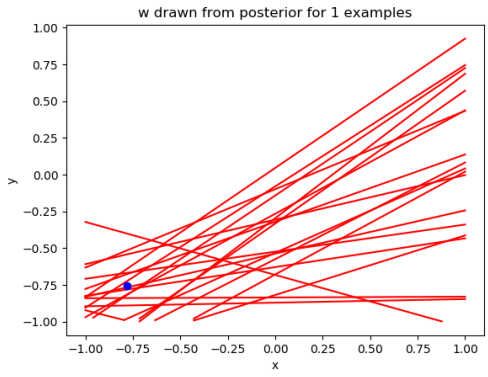
- As a single data-point (x_1, y_1) is seen by the model, the distribution on $\mathbf{w}$ is changed as per the seen point. This makes all those $\mathbf{w}$ more likely which can fit the seen data-point (x_1, y_1).

- As more data-points (x_1, y_1) are seen by the model, the distribution on $\mathbf{w}$ changes such that the variance decreases with the mean approaching the true $\mathbf{w}$. Thus, all those $\mathbf{w}$ which can fit the seen data-points become more likely.

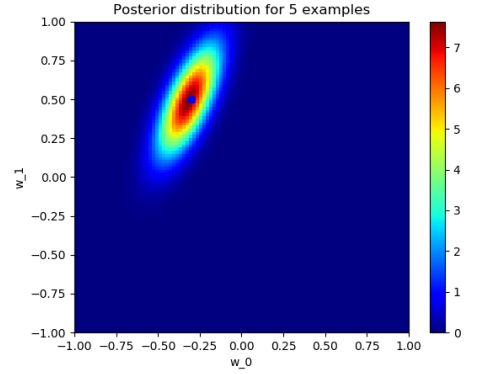
- As all the 20 data-points are seen by the model, the posterior distribution on $\mathbf{w}$ becomes more spiky at the true $\mathbf{w}$ with decreased variance.



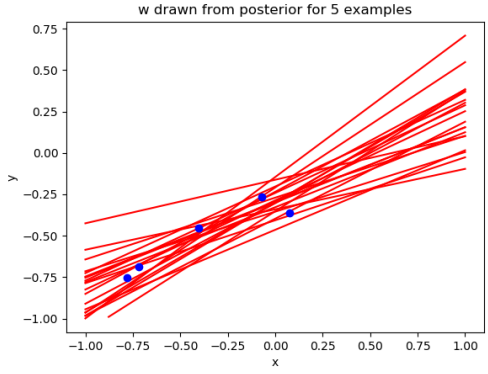
(a) Posterior($\mathbf{w}$) given (x_1, y_1)



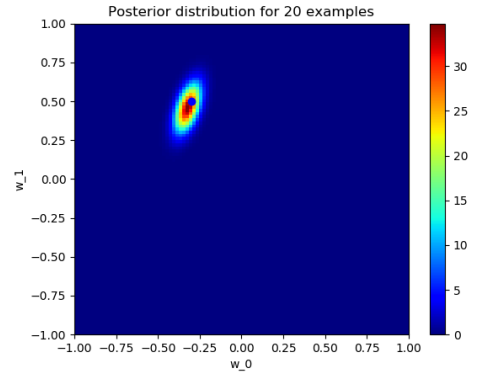
(b) Samples of $\mathbf{w}$ drawn from Posterior($\mathbf{w}$) given (x_1, y_1)



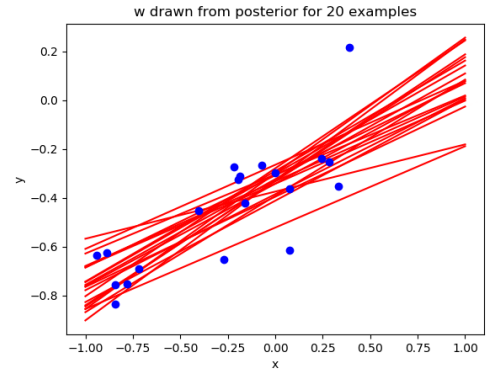
(a) Posterior($\mathbf{w}$) given 5 data-points



(b) Samples of $\mathbf{w}$ drawn from Posterior($\mathbf{w}$) given given 5 data-points



(a) Posterior($\mathbf{w}$) given 5 data-points



(b) Samples of $\mathbf{w}$ drawn from Posterior($\mathbf{w}$) given given 5 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?

- With zero initialisation of weight vector $\mathbf{w}$, the model has a test accuracy of 99.0%. The tolerance reached in 9 iterations.
- With random initialisation from standard gaussian of weight vector $\mathbf{w}$, the model has a test accuracy of 48.80% after 100 iterations. The model has not converged.
  If initial weights are too large, the gradients back-propagated through the model will be too large resulting in model's divergence. If weights are too small, the model suffers from vanishing gradient problem and does not train properly. Thus in neural networks or models trained using back-propagation algorithm, proper weight initialisation is very important.
- With xavier initialisation of weight vector $\mathbf{w}$, the model has a test accuracy of 99.0%. The tolerance reached in 9 iterations.

(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?

- With zero initialisation of weight vector $\mathbf{w}$, the model has a test accuracy of 99.0%.
- With random initialisation from standard gaussian of weight vector $\mathbf{w}$, the model has a test accuracy of 28.0%. The model has not converged.
  If initial weights are too large, the gradients back-propagated through the model will be too large resulting in model's divergence. If weights are too small, the model suffers from vanishing gradient problem and does not train properly. Thus in neural networks or models trained using back-propagation algorithm, proper weight initialisation is very important.
- With xavier initialisation of weight vector $\mathbf{w}$, the model has a test accuracy of 99.0%.
- With random weights, logistic seems to be a better option as it suffers less from vanishing gradient problem in comparison to probit. However, with proper initialisation of weight vector, both probit and logistic works same.

(c) [**Bonus**][15 points]. Implement Newton-Raphson scheme to find the MAP estimation for Probit regression. Report the prediction accuracy

- With zero/xavier initialisation of weight vector $\mathbf{w}$, the model has a test accuracy of 98.8%. The tolerance reached in 9 iterations.

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.
Prediction accuracy on the test data is 91.0 %.

(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?
Prediction accuracy on the test data is 87.0 %. Training multi-class classifier is better than training K one class classifier.