2023Fall ML Midterm

12110644周思呈

Problem I. Least Square (15 points)

a) Consider $Y = AX + V$ and $V \sim \mathcal{N}(\mathbf{v} \mid \mathbf{0}, Q)$, what is the least square solution of X ?

The least squares solution of X is obtained by minimizing the negative log-likelihood. The likelihood function is given by:

$$P(Y|X) = \frac{1}{(2\pi)^{n/2}|Q|^{1/2}} \exp\left(-\frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX)\right) \tag{1}$$

Taking the negative log-likelihood and minimizing it is equivalent to the least squares problem. The solution can be found by setting the gradient of the negative log-likelihood with respect to X to zero:

$$Q^{-1}A^T(Y - AX) = 0 \tag{2}$$

Multiplying both sides by A and rearranging, we get the normal equation:

$$A^T Q^{-1} A\hat{X} = A^T Q^{-1} Y \tag{3}$$

Assuming $A^T Q^{-1} A$ is invertible, the solution for $\hat{X}$ is given by:

$$\hat{X} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y \tag{4}$$

b) If there is a constraint of $b^T X = c$, what is the optimal solution of X ?

If there is a linear equality constraint $b^T X = c$, the optimization problem becomes a constrained optimization problem. The Lagrange multiplier method can be used to find the optimal solution of X. The Lagrangian function is given by:

$$L(X, \lambda) = \|Y - AX\|_2^2 + \lambda^T(b^T X - c) \tag{5}$$

where λ is the vector of Lagrange multipliers associated with the equality constraint. The optimal solution is found by setting the partial derivatives of the Lagrangian with respect to X and λ to zero.

The partial derivative with respect to X is:

$$\frac{\partial L}{\partial X} = -2A^T(Y - AX) + \lambda b = 0 \tag{6}$$

The partial derivative with respect to λ is:

$$\frac{\partial L}{\partial \lambda} = b^T X - c = 0 \tag{7}$$

Solving these equations simultaneously gives the optimal solution. The solution is given by:

$$\begin{gathered} A^T A\hat{X} + \lambda b = A^T Y \\ b^T \hat{X} = c \end{gathered}$$

c) If there is an additional constraint of $X^T X = d$, in addition to the constraint in b), what is the optimal solution of X ?

The Lagrangian function with both constraints is given by:

$$L(X, \lambda, \mu) = \|Y - AX\|_2^2 + \lambda^T(b^T X - c) + \mu(X^T X - d) \tag{8}$$

where μ is the Lagrange multiplier associated with the quadratic equality constraint.

Setting the partial derivatives of the Lagrangian with respect to X, λ, and μ to zero gives a system of equations to solve for the optimal solution:

$$\begin{aligned} \frac{\partial L}{\partial X} &= -2A^T(Y - AX) + \lambda b + 2\mu X = 0 \\ \frac{\partial L}{\partial \lambda} &= b^T X - c = 0 \\ \frac{\partial L}{\partial \mu} &= X^T X - d = 0 \end{aligned} \tag{9}$$

Solving these equations simultaneously will give the optimal solution to the constrained optimization problem. The solution is obtained by combining the equations:

$$\begin{aligned} A^T A\hat{X} + \lambda b + \mu\hat{X} = A^T Y \\ b^T \hat{X} = c \\ \hat{X}^T \hat{X} = d \end{aligned}$$

d) If both A and X are unknown, how to solve A and X alternatively by using two constraints of $X^T X = d$ and $\operatorname{Trace}\left(A^T A\right) = e$?

Employ the Alternating Least Squares method for optimization.

1. **Initialization:** Set initial values for A and X.
2. **Alternating Optimization:**
   - **Update X:** Fix A and update X by solving the optimization problem:
     $\min_X \|Y - AX\|_2^2 \quad \text{subject to } X^T X = d.$

     Using the Lagrange multiplier method, construct the Lagrangian function:
     $L(X, \lambda) = \|Y - AX\|_2^2 + \lambda(X^T X - d).$

     Take the partial derivative of L with respect to X and set it to zero:
     $-2A^T(Y - AX) + 2\lambda X = 0.$
     Solve this equation to obtain the update formula for X.
   - **Update A:** Fix X and update A by solving the optimization problem:
     $\min_A \|Y - AX\|_2^2 \quad \text{subject to } \operatorname{Trace}(A^T A) = e.$

     Similarly, using the Lagrange multiplier method, construct the Lagrangian function:
     $L(A, \mu) = \|Y - AX\|_2^2 + \mu(\operatorname{Trace}(A^T A) - e).$

     Take the partial derivative of L with respect to A and set it to zero:
     $-2X^T(Y - AX) + 2\mu A = 0.$
     Solve this equation to obtain the update formula for A.
3. **Iteration:** Repeat the update process from step 2 until convergence.

Problem II. Linear Gaussian System (10 points)

Consider $Y = AX + V$, where X and V are Gaussian, $X \sim \mathscr{N}\left(\boldsymbol{x} \mid \boldsymbol{m}_0, \boldsymbol{\Sigma}_0\right), V \sim \mathscr{N}\left(\boldsymbol{v} \mid \mathbf{0}, \boldsymbol{\beta}^{-1}\boldsymbol{I}\right)$. What are the conditional distribution, $p(Y \mid X)$, the joint distribution $p(Y, X)$, the marginal distribution, $p(Y)$, the posterior distribution, $p\left(X \mid Y = \boldsymbol{y}, \beta, \boldsymbol{m}_0, \Sigma_0\right)$, the posterior predictive distribution, $p\left(\hat{Y} \mid Y = \boldsymbol{y}, \beta, \boldsymbol{m}_0, \Sigma_0\right)$, and the prior predictive distribution, $p\left(Y \mid \beta, \boldsymbol{m}_0, \Sigma_0\right)$, respectively?

1. Conditional Distribution $p(Y \mid X) = \mathcal{N}(AX, \boldsymbol{\beta}^{-1}\mathbf{I})$
2. Joint Distribution $p(Y, X) = p(Y \mid X) \cdot p(X) = \mathcal{N}(AX, \boldsymbol{\beta}^{-1}\mathbf{I}) \cdot \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$
3. Marginal Distribution $p(Y) = \int p(Y, X)\, dX$
4. Posterior Distribution $p(X \mid Y = \boldsymbol{y}, \beta, \boldsymbol{m}_0, \Sigma_0) \propto p(Y \mid X) \cdot p(X)$
5. Posterior Predictive Distribution $p(\hat{Y} \mid Y = \boldsymbol{y}, \beta, \boldsymbol{m}_0, \Sigma_0) = \int p(\hat{Y} \mid X) \cdot p(X \mid Y = \boldsymbol{y}, \beta, \boldsymbol{m}_0, \Sigma_0)\, dX$
6. Prior Predictive Distribution $p(Y \mid \beta, \boldsymbol{m}_0, \Sigma_0) = \int p(Y \mid X) \cdot p(X)\, dX$

Problem III. Linear Regression (10 points)

Consider $y = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + v$, where v is Gaussian, i.e., $v \sim \mathscr{N}\left(v \mid 0, \beta^{-1}\right)$, and $\boldsymbol{w}$ has a Gaussian priori, i.e., $\boldsymbol{w} \sim \mathscr{N}\left(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha^{-1}\boldsymbol{I}\right)$. Assume that $\phi(\boldsymbol{x})$ is known, please derive the posterior distribution, $p\left(\boldsymbol{w} \mid D, \beta, \boldsymbol{m}_0, \alpha\right)$, the posterior predictive distribution, $p\left(\hat{y} \mid \hat{x}, D, \beta, \boldsymbol{m}_0, \alpha\right)$, and the prior predictive distribution, $p\left(D \mid \beta, \boldsymbol{m}_0, \alpha\right)$, respectively, where $D = \{\phi_n, y_n\}, n = 1, \ldots, N$, is the training data set and $\phi_n = \phi\left(\mathbf{x}_n\right)$.

1. Posterior Distribution

$$\begin{gathered}
p(\boldsymbol{w} \mid D, \beta, \boldsymbol{m}_0, \alpha) \propto p(D \mid \boldsymbol{w}, \beta) \cdot p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha) \\
p(D \mid \boldsymbol{w}, \beta) \propto \exp\left(-\frac{\beta}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})\right) \\
p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha) \propto \exp\left(-\frac{\alpha}{2}(\boldsymbol{w} - \boldsymbol{m}_0)^T(\boldsymbol{w} - \boldsymbol{m}_0)\right)
\end{gathered} \tag{10}$$

Take the logarithm:

$$\log p(\boldsymbol{w} \mid D, \beta, \boldsymbol{m}_0, \alpha) \propto -\frac{\beta}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) - \frac{\alpha}{2}(\boldsymbol{w} - \boldsymbol{m}_0)^T(\boldsymbol{w} - \boldsymbol{m}_0) \tag{11}$$

Expanding and collecting terms, we get a quadratic form. Completing the square and rearranging, we obtain the parameters of the Gaussian distribution:

$$\begin{gathered}
\boldsymbol{m}_N = \beta\boldsymbol{S}_N(\beta\boldsymbol{\Phi}^T\boldsymbol{y} + \alpha\boldsymbol{I}\boldsymbol{m}_0) \\
\boldsymbol{S}_N^{-1} = \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \alpha\boldsymbol{I}
\end{gathered} \tag{12}$$

Thus, the posterior distribution is a Gaussian distribution with mean $\boldsymbol{m}_N$ and precision matrix $\boldsymbol{S}_N^{-1}$.

2. Posterior Predictive Distribution:

$$p(\hat{y} \mid \hat{x}, D, \beta, \boldsymbol{m}_0, \alpha) = \int p(\hat{y} \mid \hat{\boldsymbol{w}}) \cdot p(\hat{\boldsymbol{w}} \mid D, \beta, \boldsymbol{m}_0, \alpha) \, d\hat{\boldsymbol{w}} \tag{13}$$

where, $p(\hat{y} \mid \hat{\boldsymbol{w}})$ is the likelihood of the new data point given $\hat{\boldsymbol{w}}$.

3. Prior Predictive Distribution:

$$p(D \mid \beta, \boldsymbol{m}_0, \alpha) = \int p(D \mid \boldsymbol{w}, \beta) \cdot p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha) \, d\boldsymbol{w} \tag{14}$$

Problem IV. Logistics Regression (10 points)

Consider a two-class classification problem with the logistic sigmoid function, $y = \sigma\left(\boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x})\right)$, for a given data set $D = \{\phi_n, t_n\}$, where $t_n \in \{0, 1\}, \phi_n = \phi\left(\mathbf{x}_n\right), n = 1, \ldots, N$, and the likelihood function is given by

$$p(\boldsymbol{t} \mid \boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n} \tag{15}$$

where $\boldsymbol{w}$ has a Gaussian priori, i.e., $\boldsymbol{w} \sim \mathscr{N}\left(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha^{-1} \boldsymbol{I}\right)$. Please derive the posterior distribution, $p\left(\boldsymbol{w} \mid D, \boldsymbol{m}_0, \alpha\right)$, the posterior predictive distribution, $p\left(t \mid x, D, \boldsymbol{m}_0, \alpha\right)$, and the prior predictive distribution, and $p\left(D \mid \boldsymbol{m}_0, \alpha\right)$, respectively. (Hint: using Delta approximation and Laplace approximation properly).

1. Posterior Distribution:

$$\begin{aligned} p(\boldsymbol{w} \mid D, \boldsymbol{m}_0, \alpha) &= p(D \mid \boldsymbol{w}) \cdot p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha) \\ &= \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n} N(w|m_0, \alpha^{-1} I) \end{aligned} \tag{16}$$

$$\begin{aligned} E(\boldsymbol{w}) &= -\ln p(\boldsymbol{w}|D, \boldsymbol{m_0}, \alpha) \\ &= \frac{1}{2}(\boldsymbol{w} - \boldsymbol{m_0})^T \alpha I (\boldsymbol{w} - \boldsymbol{m_0}) - \sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) ln(1 - y_n)] \\ b &= \nabla E(\boldsymbol{w}) = \alpha I (\boldsymbol{w} - \boldsymbol{m_0}) + \sum_{n=1}^{N} (y_n - t_n) \Phi_n \\ H &= \nabla\nabla E(\boldsymbol{w}) = \alpha I + \sum n = 1^N y_n (1 - y_n) \Phi_n \Phi_n^T \end{aligned} \tag{17}$$

$$\begin{aligned} \boldsymbol{w_{MAP}} &= \boldsymbol{w_{old}} - H^{-1} b \\ p(\boldsymbol{w}|D, \boldsymbol{m_0}, \alpha) &= N(\boldsymbol{w}|\boldsymbol{w_{MAP}}, H^{-1}) \end{aligned} \tag{18}$$

2. Posterior Predictive Distribution:

$$\begin{aligned} p(t \mid x, D, \boldsymbol{m}_0, \alpha) &= \int p(t \mid \boldsymbol{w}) \cdot p(\boldsymbol{w} \mid D, \boldsymbol{m}_0, \alpha) \, d\boldsymbol{w} \\ &= N(t|y, y(1-y)) \end{aligned} \tag{19}$$

3. Prior Predictive Distribution:

$$p(D \mid \boldsymbol{m}_0, \alpha) = \int p(t \mid \boldsymbol{w}) \cdot p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha)\, d\boldsymbol{w} \tag{20}$$

Problem V. Neural Network (10 points)

Consider a two-layer neural network described by following equations:

$$a_1 = \boldsymbol{w}^{(1)}\boldsymbol{x}, a_2 = \boldsymbol{w}^{(2)}\boldsymbol{z}, z = h\left(a_1\right), y = \sigma\left(a_2\right) \tag{21}$$

where $\boldsymbol{x}$ and y are the input and output, respectively, of the neural network, $h(\bullet)$ is a nonlinear function, and $\sigma(\bullet)$ is the sigmod function.
(1) Please derive the following gradients: $\frac{\partial y}{\partial \mathbf{w}^{(1)}}$, $\frac{\partial y}{\partial w^{(2)}}$, $\frac{\partial y}{\partial a_1}$, $\frac{\partial y}{\partial a_2}$, and $\frac{\partial y}{\partial x}$.

1. $\frac{\partial y}{\partial \mathbf{w}^{(1)}}$:

Using the chain rule:

$$\frac{\partial y}{\partial \mathbf{w}^{(1)}} = \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial \mathbf{w}^{(1)}} \tag{22}$$

Since $a_2 = \boldsymbol{w}^{(2)}\boldsymbol{z}$, and $\boldsymbol{z} = h(a_1)$, we have:

$$\frac{\partial y}{\partial a_2} = \sigma'(a_2) \quad \text{and} \quad \frac{\partial a_2}{\partial \mathbf{w}^{(1)}} = \boldsymbol{w}^{(2)} \cdot \frac{\partial \boldsymbol{z}}{\partial \mathbf{w}^{(1)}} \tag{23}$$

Therefore,

$$\frac{\partial y}{\partial \mathbf{w}^{(1)}} = \sigma'(a_2) \cdot \boldsymbol{w}^{(2)} \cdot \frac{\partial \boldsymbol{z}}{\partial \mathbf{w}^{(1)}} \tag{24}$$

2. $\frac{\partial y}{\partial w^{(2)}}$:

Using the chain rule:

$$\frac{\partial y}{\partial w^{(2)}} = \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial w^{(2)}} \tag{25}$$

Since $a_2 = \boldsymbol{w}^{(2)}\boldsymbol{z}$, we have:

$$\frac{\partial y}{\partial a_2} = \sigma'(a_2) \quad \text{and} \quad \frac{\partial a_2}{\partial w^{(2)}} = z \tag{26}$$

Therefore,

$$\frac{\partial y}{\partial w^{(2)}} = \sigma'(a_2) \cdot z \tag{27}$$

3. $\frac{\partial y}{\partial a_1}$:

Using the chain rule:

$$\frac{\partial y}{\partial a_1} = \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial a_1} \tag{28}$$

Since $a_2 = \boldsymbol{w}^{(2)}\boldsymbol{z}$, and $\boldsymbol{z} = h(a_1)$, we have:

$$\frac{\partial y}{\partial a_2} = \sigma'(a_2) \quad \text{and} \quad \frac{\partial a_2}{\partial a_1} = \boldsymbol{w}^{(2)} \cdot h'(a_1) \tag{29}$$

Therefore,

$$\frac{\partial y}{\partial a_1} = \sigma'(a_2) \cdot \boldsymbol{w}^{(2)} \cdot h'(a_1) \tag{30}$$

4. $\frac{\partial y}{\partial a_2}$:

$$\frac{\partial y}{\partial a_2} = \sigma'(a_2) \tag{31}$$

5. $\frac{\partial y}{\partial x}$:

Using the chain rule:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial a_1} \cdot \frac{\partial a_1}{\partial x} \tag{32}$$

Since $a_1 = \boldsymbol{w}^{(1)}\boldsymbol{x}$, we have:

$$\frac{\partial a_1}{\partial x} = \boldsymbol{w}^{(1)} \tag{33}$$

Therefore,

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial a_1} \cdot \boldsymbol{w}^{(1)} \tag{34}$$

(2) Please derive the updating rules for $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$ given the classification errors between y and t, where t is the ground truth of the output y.

1. Updating Rule for $\boldsymbol{w}^{(2)}$:

Using the chain rule:

$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{w}^{(2)}} &= \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial \boldsymbol{w}^{(2)}} \\ \frac{\partial E}{\partial y} &= -\frac{t}{y} + \frac{1-t}{1-y} \\ \frac{\partial y}{\partial a_2} &= \sigma'(a_2) \\ \frac{\partial a_2}{\partial \boldsymbol{w}^{(2)}} &= \boldsymbol{z}\end{aligned} \tag{35}$$

Therefore,

$$\frac{\partial E}{\partial \boldsymbol{w}^{(2)}} = \left(-\frac{t}{y} + \frac{1-t}{1-y}\right)\sigma'(a_2)\boldsymbol{z} \tag{36}$$

The updating rule becomes:

$$\boldsymbol{w}^{(2)} \leftarrow \boldsymbol{w}^{(2)} + \eta \left(\frac{t}{y} - \frac{1-t}{1-y} \right) \sigma'(a_2) \boldsymbol{z} \tag{37}$$

2. Updating Rule for $\boldsymbol{w}^{(1)}$:

Using the chain rule:

$$\frac{\partial E}{\partial \boldsymbol{w}^{(1)}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial a_2} \cdot \frac{\partial a_2}{\partial \boldsymbol{z}} \cdot \frac{\partial \boldsymbol{z}}{\partial a_1} \cdot \frac{\partial a_1}{\partial \boldsymbol{w}^{(1)}} \tag{38}$$

$$\frac{\partial \boldsymbol{z}}{\partial a_1} = h'(a_1)$$

$$\frac{\partial a_1}{\partial \boldsymbol{w}^{(1)}} = \boldsymbol{x}$$

Therefore,

$$\frac{\partial E}{\partial \boldsymbol{w}^{(1)}} = \left(-\frac{t}{y} + \frac{1-t}{1-y} \right) \sigma'(a_2) \boldsymbol{w}^{(2)} h'(a_1) \boldsymbol{x} \tag{39}$$

The updating rule becomes:

$$\boldsymbol{w}^{(1)} \leftarrow \boldsymbol{w}^{(1)} + \eta \left(\frac{t}{y} - \frac{1-t}{1-y} \right) \sigma'(a_2) \boldsymbol{w}^{(2)} h'(a_1) \boldsymbol{x} \tag{40}$$

Problem VI. Bayesian Neural Network (20 points)

a) Consider a neural network for regression, $t = y(\boldsymbol{w}, \boldsymbol{x}) + v$, where v is Gaussian, i.e., $v \sim \mathcal{N}\left(v \mid 0, \beta^{-1}\right)$, and $\boldsymbol{w}$ has a Gaussian priori, i.e., $\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha^{-1} \boldsymbol{I}\right)$. Assume that $y(\boldsymbol{w}, \boldsymbol{x})$ is the neural network output. Please derive the posterior distribution, $p\left(\boldsymbol{w} \mid D, \beta, \boldsymbol{m}_0, \alpha\right)$, the posterior predictive distribution, $p\left(t \mid x, D, \beta, \boldsymbol{m}_0, \alpha\right)$, and the prior predictive distribution, $p\left(D \mid \beta, \boldsymbol{m}_0, \alpha\right)$, where $D = \{x_n, t_n\}, n = 1, \ldots, N$, is the training data set.

1. Posterior Distribution:

$$\begin{aligned} p(\boldsymbol{w} \mid D, \beta, \boldsymbol{m}_0, \alpha) &= p(D \mid \boldsymbol{w}, \beta) \cdot p(\boldsymbol{w} \mid \boldsymbol{m}_0, \alpha) \\ &= \prod_{n=1}^{N} N(t_n | y(\boldsymbol{w}, x_n), \beta^{-1}) N(\boldsymbol{w} | \boldsymbol{m_0}, \alpha^{-1} I) \end{aligned} \tag{41}$$

$$\begin{gathered} E(\boldsymbol{w}) = -\ln p(\boldsymbol{w} | D, \beta, \boldsymbol{m_0}, \alpha) = \frac{\alpha}{2} \boldsymbol{w^T} \boldsymbol{w} + \frac{\beta}{2} \sum_{n=1}^{N} [y(x_n, \boldsymbol{w}) - t_n]^2 \\ b = \nabla E(\boldsymbol{w}) = \alpha \boldsymbol{w} + \beta \sum_{n=1}^{N} (y_n - t_n) \nabla_w y(x, \boldsymbol{w}) \\ A = \nabla \nabla E(\boldsymbol{w}) = \alpha I + \beta H \\ H: \ Hessian \ matrix \ of \ the \ sum - of - error \ function \\ \boldsymbol{w_{MAP}} = \boldsymbol{w_{old}} - A^{-1} b \end{gathered} \tag{42}$$

So

$$p(\boldsymbol{w}|D,\beta,\boldsymbol{m_0},\alpha)=N(\boldsymbol{w}|\boldsymbol{w_{MAP}},A^{-1}) \tag{43}$$

2. Posterior Predictive Distribution:

$$p(t\mid x,D,\beta,\boldsymbol{m}_0,\alpha)=\int p(t\mid\boldsymbol{w},x,\beta)\cdot p(\boldsymbol{w}\mid D,\beta,\boldsymbol{m}_0,\alpha)\,d\boldsymbol{w} \tag{44}$$

Given:

$$p(t\mid\boldsymbol{w},x,\beta)=\mathcal{N}(t\mid y(\boldsymbol{w},x),\beta^{-1}) \tag{45}$$

and the posterior distribution obtained in step 1,

$$p(\boldsymbol{w}\mid D,\beta,\boldsymbol{m}_0,\alpha) \tag{46}$$

3. Prior Predictive Distribution:

$$p(D\mid\beta,\boldsymbol{m}_0,\alpha)=\int p(t\mid\boldsymbol{w},x,\beta)\cdot p(\boldsymbol{w}\mid\beta,\boldsymbol{m}_0,\alpha)\,d\boldsymbol{w} \tag{47}$$

Given:

$$p(t\mid\boldsymbol{w},x,\beta)=\mathcal{N}(t\mid y(\boldsymbol{w},x),\beta^{-1}) \tag{48}$$

and the prior distribution on $\boldsymbol{w}$:

$$p(\boldsymbol{w}\mid\beta,\boldsymbol{m}_0,\alpha)=\mathcal{N}(\boldsymbol{w}\mid\boldsymbol{m}_0,\alpha^{-1}\boldsymbol{I}) \tag{49}$$

b) Consider a neural network for two-class classification, $y=\sigma(a(\boldsymbol{w},\boldsymbol{x}))$ and a data set $D=\{x_n,t_n\}$, where $t_n\in\{0,1\},\boldsymbol{w}$ has a Gaussian priori, i.e., $\boldsymbol{w}\sim,\mathcal{N}\left(\boldsymbol{w}\mid\mathbf{0},\alpha^{-1}\mathrm{I}\right)$, and $a(\boldsymbol{w},\boldsymbol{x})$ is the neural network model. Please derive the posterior distribution, $p(\boldsymbol{w}\mid D,\alpha)$, posterior predictive distribution, $p(t\mid x,D,\alpha)$, and the prior predictive distribution, $p(D\mid\alpha)$, respectively.

1. Posterior Distribution:

$$p(\boldsymbol{w}\mid D,\alpha)\propto p(D\mid\boldsymbol{w})\cdot p(\boldsymbol{w}\mid\alpha) \tag{51}$$

Given

$$p(D\mid\boldsymbol{w})=\prod_{n=1}^{N}y_n^{t_n}(1-y_n)^{1-t_n} \tag{52}$$

and

$$p(\boldsymbol{w}\mid\alpha)=\mathcal{N}(\boldsymbol{w}\mid\mathbf{0},\alpha^{-1}\mathrm{I}) \tag{53}$$

The posterior distribution is then proportional to the product of the likelihood and the prior:

$$p(\boldsymbol{w}\mid D,\alpha)\propto\prod_{n=1}^{N}y_n^{t_n}(1-y_n)^{1-t_n}\cdot\mathcal{N}(\boldsymbol{w}\mid\mathbf{0},\alpha^{-1}\mathrm{I}) \tag{54}$$

$$\begin{gathered}
E(\boldsymbol{w}) = -\ln p(D|\boldsymbol{w}) = \frac{\alpha}{2}\boldsymbol{w^T w} - \sum_{n=1}^{N}[t_n \ln y_n + (1-t_n)\ln(1-y_n)] \\
b = \nabla E(\boldsymbol{w}) = \alpha \boldsymbol{w} + \sum_{n=1}^{N}(y_n - t_n)\nabla_{\boldsymbol{w}} y(x, \boldsymbol{w}) \\
A = \nabla\nabla E(\boldsymbol{w}) = \alpha I + H \\
H:\ Hessian\ matrix\ of\ the\ sum-of-error\ function \\
\boldsymbol{w_{MAP}} = \boldsymbol{w_{old}} - A^{-1}b \\
p(\boldsymbol{w}|D, \alpha) = N(\boldsymbol{w}|\boldsymbol{w_{MAP}}, A^{-1})
\end{gathered} \tag{55}$$

2. Posterior Predictive Distribution:

$$p(t \mid x, D, \alpha) = \int p(t \mid \boldsymbol{w}, x) \cdot p(\boldsymbol{w} \mid D, \alpha)\, d\boldsymbol{w} \tag{56}$$

Where

$$p(t \mid \boldsymbol{w}, x) = y^t(1-y)^{1-t} \tag{57}$$

3. Prior Predictive Distribution:

$$p(D \mid \alpha) = \int p(t \mid \boldsymbol{w}, x) \cdot p(\boldsymbol{w} \mid \alpha)\, d\boldsymbol{w} \tag{58}$$

Given:

$$p(t \mid \boldsymbol{w}, x) = y^t(1-y)^{1-t} \tag{59}$$

and the prior distribution on $\boldsymbol{w}$:

$$p(\boldsymbol{w} \mid \alpha) = \mathscr{N}(\boldsymbol{w} \mid \mathbf{0}, \alpha^{-1}\mathrm{I}) \tag{60}$$

Problem VII. Critical Analyses (20 Points)

a) Please explain why the dual problem formulation is used to solve the SVM machine learning problem.

Original problem:

$$\begin{array}{ll}
\min_{\boldsymbol{w}, b} & \frac{1}{2}\|\boldsymbol{w}\|^2 \\
\text{s.t.} & y_i\left(\boldsymbol{w} \cdot \boldsymbol{x_i} + b\right) \geq 1
\end{array} \tag{61}$$

Dual problem:

$$\begin{array}{ll}
\min_{\lambda} & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j - \sum_{i=1}^{n}\lambda_i \\
\text{s.t.} & \lambda_i \geq 0, \quad \sum_{i=1}^{n}\lambda_i y_i = 0
\end{array} \tag{62}$$

- Dual problem does not involve $\boldsymbol{w}$ and b, which is simpler to solve;
- The constraint $\sum_{i=1}^{n}\lambda_i y_i = 0$ in dual problem is easy to eliminate;
- The constraint of the original algorithm is a complex linear inequality $y_i\left(\boldsymbol{w} \cdot \boldsymbol{x_i} + b\right) \geq 1$, while the constraint of dual problem is simply $\lambda_i \geq 0$;

- The dual formulation allows for the straightforward application of the kernel trick;

b) Please explain, in terms of cost functions, constraints and predictions, i) what are the differences between SVM classification and logistic regression; ii) what are the differences between v -SVM regression and least square regression.

SVM Classification vs. Logistic Regression:

	SVM Classification	Logistic Regression
Cost Functions	Aims to maximize the margin between different classes. The cost function involves minimizing the hinge loss, which penalizes misclassifications and encourages the correct classification of data points. The objective is to find a hyperplane that separates classes with the maximum margin. Eg. linear kernel and regularization parameter C: $\min_{\mathbf{w},b} \frac{1}{2}\|\|\mathbf{w}\|\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$	Uses the logistic (sigmoid) function to model the probability that a given input belongs to a particular class. The cost function is the negative log-likelihood, which penalizes deviations from the true class probabilities. Eg. $\min_{\mathbf{w},b} C\sum_{i=1}^{N} \left[y_i \log\left(\sigma(\mathbf{w}^T\mathbf{x}_i + b)\right) + (1 - y_i)\log\left(1 - \sigma(\mathbf{w}^T\mathbf{x}_i + b)\right)\right] + \frac{1}{2}$
Constraints	$\forall i : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$.	Generally soft constraints that penalize deviations from the correct class probabilities.
Predictions	Predict the class label based on the sign of $\mathbf{w}^T\mathbf{x} + b$.	Predicts the probability that a data point belongs to a certain class. A threshold is then applied to these probabilities to obtain binary class predictions. Eg. $\sigma(\mathbf{w}^T\mathbf{x} + b) > 0.5$

v-SVM Regression vs. Least Squares Regression:

	v-SVM Regression	Least Squares Regression
Cost Functions	The cost function involves minimizing the ε-insensitive loss, which allows for a certain amount of error (ε) in the predictions. Eg. Linear kernel $\min_{\mathbf{w},b,\epsilon} \frac{1}{2}\|\|\mathbf{w}\|\|^2 + \frac{1}{N\nu}\sum_{i=1}^{N} \epsilon_i + C\sum_{i=1}^{N} [\max(0, \epsilon_i - \nu) + \max(0, \nu - \epsilon_i)]$	Minimizes the sum of squared differences between the predicted and actual values. Eg. linear least squares regression $\min_{\mathbf{w},b} \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i - b)^2$
Constraints	$\forall i : \epsilon_i \geq 0$ and $\sum_{i=1}^{N} \epsilon_i \leq N\nu$.	No specific constraints; the model minimizes the sum of squared errors directly.
Predictions	Predicts the target value for a new data point based on its position relative to the hyperplane.	Predicts the target value based on a linear combination of features without introducing a specific margin concept.

c) Please explain why neural network (NN) based machine learning algorithms use logistic activation functions?

- The input range is $-\infty \to +\infty$ and the output range is (0, 1), which is the same as the range of probability distribution;
- The derivative of the logistic function has a simple and computationally efficient form, making it well-suited for gradient descent optimization algorithms and backpropagation;
- The logistic function has a bounded output, which helps in mitigating the vanishing and exploding gradient problems during backpropagation.

d) Please explain i) what are the differences between the logistic activation function and other activation functions (e.g., relu, tanh); and ii) when these activation functions should be used.

	Logistic (Sigmoid)	ReLU	Tanh	Softmax
Output Range	(0, 1)	Outputs the input for positive values, and 0 for negative values	(-1, 1)	(0, 1), with a sum of 1 across all classes
Shape	S-shaped curve, smooth and differentiable	Piecewise linear, not differentiable at 0	S-shaped curve similar to the sigmoid, but with a range from -1 to 1	Converts raw scores into probability distribution across multiple classes
Usage	Commonly used in the output layer for binary classification tasks. Also used in some cases in hidden layers, particularly in networks where the goal is to model probabilities.	Popular in hidden layers due to its simplicity and efficiency. Addresses the vanishing gradient problem and accelerates training. Not suitable for output layers in classification tasks, as it doesn't squash values into a probability range.	Similar to sigmoid, but with a broader range. Commonly used in hidden layers to model non-linear relationships and address the vanishing gradient problem	Typically used in the output layer for multi-class classification problems. Ensures that the output represents a valid probability distribution over all classes.

e) Please explain why Jacobian and Hessian matrices are useful for machine learning algorithms.

Jacobian Matrix:

The Jacobian matrix represents the first-order partial derivatives of a vector-valued function with respect to its input variables.

1. Gradient Descent Optimization. It provides information about the rate of change of the output with respect to each input, guiding the optimization process to find the minimum of a cost function.
2. Backpropagation in Neural Networks. It helps update the model parameters to minimize the difference between predicted and actual values.
3. Sensitivity Analysis. The Jacobian matrix provides insight into how small changes in input features affect the model's predictions.

Hessian Matrix:

The Hessian matrix represents the second-order partial derivatives of a scalar-valued function with respect to its input variables.

1. Newton's Method for Optimization. Newton's method uses the Hessian matrix to refine the search for the minimum or maximum of a cost function. The Hessian provides information about the curvature of the cost function, allowing the algorithm to adjust the step size during optimization.
2. Second-Order Optimization Algorithms. Algorithms like L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) use approximations of the Hessian to perform more sophisticated updates during optimization.
3. Stability Analysis. Eigenvalues of the Hessian matrix can provide insights into the stability of critical points (minima, maxima, or saddle points) of the cost function. A positive definite Hessian indicates a local minimum, while a negative definite Hessian indicates a local maximum.

f) Please explain why exponential family distributions are so common in engineering practice. Please give some examples which are NOT exponential family distributions.

Exponential family distributions are common in engineering practice for several reasons:

1. Conjugate Priors. Many exponential family distributions have conjugate prior distributions. This property simplifies Bayesian analysis by ensuring that the posterior distribution belongs to the same family as the prior, leading to analytical solutions and computational efficiency.
2. Computational Efficiency. Exponential family distributions often lead to closed-form solutions for maximum likelihood estimation, making them computationally efficient.

Examples of Distributions NOT in the Exponential Family:

1. Cauchy Distribution:

   PDF: $f(x|x_0, \gamma) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}$

2. Pareto Distribution:

   PDF: $f(x|x_{\min}, \alpha) = \frac{\alpha x_{\min}^{\alpha}}{x^{\alpha+1}}$, for $x \geq x_{\min}$

3. Multinomial Distribution:

   PMF: $P(X_1 = k_1, \ldots, X_k = k_k) = \frac{n!}{k_1! \ldots k_k!} p_1^{k_1} \ldots p_k^{k_k}$, where X_i represents the count of category i, n is the total number of trials, and p_i is the probability of category i.

4. Student's t-Distribution:

   PDF: $f(x|\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$, where ν is the degrees of freedom.

5. Logistic Distribution:

   PDF: $f(x|\mu, s) = \frac{e^{-(x-\mu)/s}}{s\left(1+e^{-(x-\mu)/s}\right)^2}$, where μ is the location parameter, and s is the scale parameter.

g) Please explain why KL divergence is useful for machine learning? Please provide two examples of using KL divergence in machine learning.

Useful Aspects in Machine Learning:

1. Model Training. Minimizing $D_{\mathrm{KL}}(P \parallel Q)$ encourages the model to approach the true distribution in machine learning.
2. Regularization. KL divergence serves as a regularization term in models like VAEs, preventing the learned distribution from deviating too far from a chosen prior.
3. Bayesian Inference. KL divergence measures the discrepancy between prior and posterior distributions in Bayesian inference.
4. Information Theory. KL divergence is related to information gain and minimizing surprise during transitions between distributions.

Examples of KL Divergence in Machine Learning:

1. Variational Autoencoders (VAEs). In VAEs, the model is trained to minimize the KL divergence between the learned latent distribution (approximate posterior) and a chosen prior distribution (usually a simple distribution like a standard normal). This ensures that the learned latent space follows the desired properties, making the model suitable for generating new samples and interpolating between existing ones.

2. Probabilistic Graphical Models. KL divergence is used in probabilistic graphical models to quantify the difference between the true distribution over variables and the distribution inferred from observed data. This is often part of the training process in Bayesian networks or other probabilistic models.

h) Please explain why data augmentation techniques are a kind of regularization skills for NNs.

Data augmentation can be seen as a form of regularization because it helps to prevent overfitting by increasing the diversity of the training data and introducing more variability.

Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function. The goal is to make the model more robust and better at generalizing from the training data to unseen data.

Data augmentation works by creating variations of the training data. For example, in image classification, this could involve rotating, scaling, cropping, or flipping the images. This increases the size of the training set and introduces more variability, which can help the model to learn more robust features and improve its ability to generalize.

By increasing the diversity of the training data, data augmentation can help to reduce overfitting. Overfitting occurs when a model learns the training data too well, to the point where it performs poorly on unseen data because it's too focused on the specific details and noise in the training set. By introducing more variability into the training data, data augmentation can help to ensure that the model learns more general features that are applicable to a wider range of data.

i) Please explain why Gaussian distributions are preferred over other distributions for many machine learning models?

1. Central Limit Theorem: The Central Limit Theorem states that when a large number of independent and identically distributed random variables are added together, their sum tends towards a Gaussian distribution, regardless of the original distribution of these variables.
2. Mathematical Simplicity: The Gaussian distribution is defined by two parameters, the mean and the variance. This simplicity makes the math involved in working with Gaussian distributions straightforward and computationally efficient.
3. Connection to Exponential Families: The Gaussian distribution is part of the exponential family of distributions, which has several useful properties. For instance, all members of the exponential family have conjugate priors, which is especially desirable for Bayesian methods. Furthermore, it's simple to derive the moments of the distribution, as these are just derivatives of the log partition function.
4. Marginals of Gaussians are Gaussians.

j) Please explain why Laplacian approximation can be used for many cases?

1. Simplification of Complex Problems: Laplace approximation is used to approximate the posterior distribution in Bayesian statistics. It provides an analytical expression for a posterior probability distribution by fitting a Gaussian distribution with a mean equal to the Maximum a Posteriori (MAP) solution and precision equal to the observed Fisher information.
2. Convergence to Gaussian: According to the Bernstein–von Mises theorem, under regularity conditions, the posterior distribution converges to a Gaussian distribution in large samples.

3. Applicability to Well-behaved Functions: Laplace approximation works for functions that are in the class of L^2, meaning that the integral of the square of the function is finite. Such functions generally have very rapidly decreasing tails so that in the far reaches of the domain we would not expect to see large spikes.
4. Replacement of Integration with Maximization: Laplace approximation replaces the problem of integrating a function with the problem of maximizing it. In order to compute the Laplace approximation, we have to compute the location of the mode, which is an optimization problem.

k) What are the fundamental principles for model selection (degree of complexity) in machine learning?

Bias-Variance Tradeoff, Complexity and Interpretability, Computational Resources, Domain Knowledge, Cross-Validation.

l) How to choose a new data sample (feature) for regression and classification model training, respectively? How to choose it for testing? Please provide some examples.

Regression Models:

For regression models, you need to consider the relationship between the dependent and independent variables. The new data sample should ideally contain a variety of values for the independent variables to capture the full range of possible relationships.

For example, if you're predicting house prices based on the size of the house, you might want to include data samples where the house size varies widely to capture the different price ranges.

Classification Models:

For classification models, the new data sample should ideally represent the different classes that the model is trying to predict.

For example, if you're predicting whether an email is spam or not, you might want to include data samples that include both spam and non-spam emails.

Testing:

When choosing a new data sample for testing, it's important to ensure that the data is representative of the data that the model will encounter in the real world.

For example, if your model is being used to predict house prices based on location, you might want to include data samples from different locations in your test set. This will give you a better idea of how well your model will perform in different scenarios.

m) Please explain why the MAP model is usually more preferred than the ML model?

1. MAP incorporates prior knowledge into the model through the use of a prior distribution.
2. MAP can be seen as a form of regularization. By incorporating a prior distribution, MAP can effectively limit the complexity of the model by pushing the parameters towards more conservative values.

3. Finding the MAP estimate is often easier than finding the ML estimate. This is because MAP involves optimizing a single function, while ML involves maximizing a function over an entire parameter space.
4. MAP provides a flexible framework that can be adapted to different types of models and data. It can be used with a wide range of likelihood functions and prior distributions, making it a versatile tool for many different types of statistical and machine learning problems.

Problem VIII. Discussions (10 Points)

(1) What are the generative and discriminative approaches to machine learning, respectively? Can you explain the advantages and disadvantages of these two approaches and provide a detailed example to illustrate your points?

	Generative	**Discriminative**
Definition	Predict the joint probability distribution $P(X,Y)$ using Bayes Theorem	Learn the conditional probability $P(Y \mid X)$
Advantage	Useful for unsupervised machine learning tasks	Computationally cheap, useful for supervised machine learning tasks, being more robust to outliers
Disadvantage	Computationally expensive	Dependence on training data, difficulty with imbalanced datasets
Example	Linear Discriminant Analysis, Hidden Markov models, Bayesian networks	SVM, logistic regression, decision trees, random forests

Generative Model Example:

Generative Adversarial Network (GAN). GANs consist of two parts: a generator network that creates new data instances, and a discriminator network that evaluates them for authenticity. The generator network tries to create images that look as real as possible, while the discriminator network tries to distinguish between real and generated images. Through this adversarial process, the generator learns to create more realistic images.

In this case, the advantage of a generative model is its ability to generate new data instances that are similar to the training data. The disadvantage is that it can be computationally expensive and sensitive to outliers in the data.

Discriminative Model Example:

Use SVM to classify spam emails. SVMs learn a hyperplane in an N-dimensional space (N being the number of features) that distinctly classifies the data points. In this case, the data points are emails, and the hyperplane is learned based on the features of the emails, such as the presence of certain words or phrases.

In this case, the advantage of a discriminative model is its robustness to outliers and its ability to distinguish between classes, especially in large datasets where the classes are well-separated. The disadvantage is that it cannot generate new data instances and is dependent on the training data.

(2) How do you analyze the GAN model from the generative and discriminative perspectives?

Generative Perspective:

From a generative perspective, GANs are used to generate new data instances that resemble the training data. GANs consist of two parts: a generator network and a discriminator network. The generator network creates new data instances, and the discriminator network evaluates them for authenticity. The generator network tries to create images that look as real as possible, while the discriminator network tries to distinguish between real and generated images. Through this adversarial process, the generator learns to create more realistic images.

The advantage of this generative approach is that it can generate new data instances that are similar to the training data. However, it can be computationally expensive and sensitive to outliers in the data.

Discriminative Perspective:

From a discriminative perspective, GANs are used to distinguish between real data instances and generated ones. The discriminator network in a GAN is trained to classify images as real or generated. It learns to distinguish between real and generated images based on the features of the images.

The advantage of this discriminative approach is that it can distinguish between real and generated data instances. However, it does not have the ability to generate new data instances.