Machine Learning (H)
Southern University of Science and Technology
Mengxuan Wu
12212006

Midterm

Mengxuan Wu

Problem 1

a)

Since $Y = AX + V$ and $V \sim N(v|0, Q)$, we have $Y \sim N(y|AX, Q)$. The error function for least squares is given by

$$E(X) = \frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX)$$

To find the optimal X, we take the derivative of $E(X)$ with respect to X and set it to zero.

$$\begin{aligned}
\frac{\partial E(X)}{\partial X} &= \frac{\partial}{\partial X}\left(\frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX)\right) \\
&= \frac{\partial}{\partial X}\left(\frac{1}{2}(Y^T Q^{-1} Y - Y^T Q^{-1} AX - X^T A^T Q^{-1} Y + X^T A^T Q^{-1} AX)\right) \\
&= \frac{\partial}{\partial X}\left(\frac{1}{2}(Y^T Q^{-1} Y - 2A^T Q^{-1} YX + X^T A^T Q^{-1} AX)\right) \\
&= -A^T Q^{-1} Y + A^T Q^{-1} AX \\
&= 0
\end{aligned}$$

Thus, we have

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

b)

We solve the extra constraint $b^T X = c$ by adding a Lagrange multiplier λ to the error function, which becomes

$$E(X, \lambda) = \frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda(b^T X - c)$$

To find the optimal X, we take the derivative of $E(X)$ with respect to X and set it to zero.

$$\begin{aligned}
\frac{\partial E(X, \lambda)}{\partial X} &= \frac{\partial}{\partial X}\left(\frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda(b^T X - c)\right) \\
&= -A^T Q^{-1} Y + A^T Q^{-1} AX + \lambda b \\
&= 0
\end{aligned}$$

Thus, we have

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

To find the optimal λ, we take the derivative of $E(X, \lambda)$ with respect to λ and set it to zero.

$$\begin{aligned}\frac{\partial E(X,\lambda)}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\left(\frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda(b^T X - c)\right) \\ &= b^T X - c \\ &= 0\end{aligned}$$

Thus, we have

$$b^T \hat{X} = c$$

Inserting $\hat{X}$ into the equation, we have

$$b^T (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - b^T (A^T Q^{-1} A)^{-1} \lambda b = c$$

Thus, we can solve λ by

$$\lambda = \frac{b^T (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - c}{b^T (A^T Q^{-1} A)^{-1} b}$$

Substitute λ back into $\hat{X}$, we have

$$\hat{X} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - (A^T Q^{-1} A)^{-1} \frac{b^T (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - c}{b^T (A^T Q^{-1} A)^{-1} b} b$$

c)

We add two Lagrange multipliers λ_1 and λ_2 to handle the two constraints. The error function becomes

$$E(X, \lambda_1, \lambda_2) = \frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda_1(b^T X - c) + \lambda_2(X^T X - d)$$

Taking the derivative of $E(X, \lambda_1, \lambda_2)$ with respect to X and set it to zero, we have

$$\frac{\partial E(X, \lambda_1, \lambda_2)}{\partial X} = -A^T Q^{-1} Y + A^T Q^{-1} A X + \lambda_1 b + 2\lambda_2 X = 0$$

Thus, we have

$$\hat{X} = (A^T Q^{-1} A + 2\lambda_2 I)^{-1} A^T Q^{-1} Y - (A^T Q^{-1} A + 2\lambda_2 I)^{-1} \lambda_1 b$$

where λ_1 and λ_2 can be solved by substitute $\hat{X}$ into the equations

$$\frac{\partial E(X, \lambda_1, \lambda_2)}{\partial \lambda_1} = b^T X - c = 0$$

$$\frac{\partial E(X, \lambda_1, \lambda_2)}{\partial \lambda_2} = X^T X - d = 0$$

d)

If both A and X are unknown, we can solve the problem by alternating optimization.

Step 1: Fix A and solve for X. The error function is

$$E(X) = \frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda_1(X^T X - d)$$

Taking the derivative of $E(X)$ with respect to X and set it to zero, we have

$$\frac{\partial E(X)}{\partial X} = -A^T Q^{-1} Y + A^T Q^{-1} A X + 2\lambda_1 X = 0$$

Thus, we have

$$\hat{X} = (A^T Q^{-1} A + 2\lambda_1 I)^{-1} A^T Q^{-1} Y$$

where λ_1 can be solved by substituting $\hat{X}$ into the equation

$$\frac{\partial E(X)}{\partial \lambda_1} = X^T X - d = 0$$

Step 2: Fix X and solve for A. The error function is

$$E(A) = \frac{1}{2}(Y - AX)^T Q^{-1}(Y - AX) + \lambda_2(\mathrm{tr}(A^T A) - e)$$

Taking the derivative of $E(A)$ with respect to A and $lambda_2$ and set them to zero, we have

$$\frac{\partial E(A)}{\partial A} = -Q^{-1} Y X^T + Q^{-1} A X X^T + 2\lambda_2 A = 0$$

$$\frac{\partial E(A)}{\partial \lambda_2} = \mathrm{tr}(A^T A) - e = 0$$

Solving the equations, we can get the optimal A.
By randomly initializing A and X, and then repeating the two steps above, we can get the optimal A and X.

Problem 2

Let $Y = AX + V$, where $X \sim N(x|m_0, \Sigma_0)$ and $V \sim N(v|0, \beta^{-1} I)$.

Conditional distribution:

$$p(Y|X) = N(Y|AX, \beta^{-1} I)$$

Joint distribution: We first define

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix}$$

Then we have

$$Z \sim N(\mu_z, \Sigma_z)$$

where

$$\mu_z = \begin{bmatrix} m_0 \\ A m_0 \end{bmatrix}$$

$$\Sigma_z = \begin{bmatrix} \Sigma_0^{-1} & \Sigma_0^{-1} A^T \\ A\Sigma_0^{-1} & \beta^{-1} I + A \Sigma_0^{-1} A^T \end{bmatrix}$$

Marginal distribution: We can find the marginal distribution of Y by integrating out X from the joint distribution of Z.

$$p(Y) = N(Y|Am_0, \beta^{-1}I + A\Sigma_0 A^T)$$

Posterior distribution: We can find the posterior distribution of X by conditioning on Y.

$$p(X|Y = y, \beta, m_0, \Sigma_0) = N(m_{\text{MAP}}, \Sigma_{\text{MAP}})$$

where

$$m_{\text{MAP}} = (\Sigma_0^{-1} + A^T\beta A)^{-1}(A^T\beta y + \Sigma_0^{-1}m_0)$$
$$\Sigma_{\text{MAP}} = (\Sigma_0^{-1} + A^T\beta I A)^{-1}$$

Predictive distribution:

$$p(\hat{Y}|Y = y, \beta, m_0, \Sigma_0) = \int p(Y|X)p(X|Y = y, \beta, m_0, \Sigma_0)dX$$

Substitute the conditional distribution and the posterior distribution into the equation, we have

$$p(\hat{Y}|Y = y, \beta, m_0, \Sigma_0) = N(\hat{Y}|Am_{\text{MAP}}, \beta^{-1}I + A\Sigma_{\text{MAP}}A^T)$$

Model evaluation: We can evaluate the model by computing the probability of the observed data given the model.

$$p(Y|\beta, m_0, \Sigma_0) = \int p(Y|X)p(X|m_0, \Sigma_0)dX = N(Y|Am_0, \beta^{-1}I + A\Sigma_0 A^T)$$

Problem 3

Given $y = w^T\phi(x) + v$, where $v \sim N(v|0, \beta^{-1})$ and $w \sim N(w|m_0, \alpha^{-1}I)$.

Posterior distribution: We first define

$$\Phi = \begin{bmatrix} \phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$p(w|D, \beta, m_0, \alpha) = N(w|m_N, S_N)$$

where

$$m_N = S_N\left(\alpha m_0 + \beta\Phi^T Y\right)$$
$$S_N^{-1} = \alpha I + \beta\Phi^T\Phi$$

Predictive distribution:

$$p(\hat{y}|\hat{x}, D, \beta, m_0, \alpha) = N(m_N^T \phi(\hat{x}), \beta^{-1} + \phi(\hat{x})^T S_N \phi(\hat{x}))$$

Model evaluation:

$$p(D|\beta, m_0, \alpha) = \int p(D|w, \beta) p(w|m_0, \alpha) dw$$

The exact form can be found by substituting the likelihood with linear Gaussian model and the prior with Gaussian distribution.

$$p(D|\beta, m_0, \alpha) = \int \left(\prod_{n=1}^{N} N(y_n|\phi(x_n)^T w, \beta^{-1}) \right) N(w|m_0, \alpha^{-1} I) dw$$

However, the integral is intractable. We can use Laplace approximation to approximate the integral. We first find the log posterior

$$\log p(w|D, \beta, m_0, \alpha) = -\frac{\beta}{2} \sum_{n=1}^{N} (y_n - w^T \phi(x_n))^2 - \frac{\alpha}{2} (w - m_0)^T (w - m_0) + \text{const}$$

Then we find the mode of the log posterior by taking the derivative of the log posterior with respect to w and set it to zero.

$$\frac{\partial}{\partial w} \log p(w|D, \beta, m_0, \alpha) = \beta \Phi^T (Y - \Phi w) - \alpha (w - m_0) = 0$$

Thus, we have

$$w_{\text{MAP}} = (\alpha I + \beta \Phi^T \Phi)^{-1} (\beta \Phi^T Y + \alpha m_0)$$

We also find the Hessian of the log posterior at the mode.

$$H = -\frac{\partial^2}{\partial w^2} \log p(w|D, \beta, m_0, \alpha) = \beta \Phi^T \Phi + \alpha I$$

Then we can evaluate the model by computing the probability of the observed data given the model.

$$\begin{aligned} p(D|\beta, m_0, \alpha) &\approx p(D|w_{\text{MAP}}, \beta) p(w_{\text{MAP}}|m_0, \alpha) \frac{(2\pi)^{M/2}}{|H|^{1/2}} \\ &= \left(\prod_{n=1}^{N} N(y_n|\phi(x_n)^T w_{\text{MAP}}, \beta^{-1}) \right) N(w_{\text{MAP}}|m_0, \alpha^{-1} I) \frac{(2\pi)^{M/2}}{|\beta \Phi^T \Phi + \alpha I|^{1/2}} \end{aligned}$$

where M is the dimension of w.

Problem 4

Given $y = \sigma(w^T \phi(x))$, where $w \sim N(w|m_0, \alpha^{-1} I)$.

Posterior distribution:

$$p(w|D, m_0, \alpha) \propto p(D|w)p(w|m_0, \alpha)$$

The likelihood is given by

$$p(D|w) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}$$

The prior is given by

$$p(w|m_0, \alpha) = N(w|m_0, \alpha^{-1}I)$$

Thus, the log posterior is given by

$$\begin{aligned}\log p(w|D, m_0, \alpha) = &\sum_{n=1}^{N}\{t_n \log y_n + (1 - t_n)\log(1 - y_n)\} \\ &- \frac{\alpha}{2}(w - m_0)^T(w - m_0) + \text{const}\end{aligned}$$

The maximum a posteriori estimate w_{MAP} can be found by taking the derivative of the log posterior with respect to w and set it to zero.

$$\frac{\partial}{\partial w}\log p(w|D, m_0, \alpha) = \sum_{n=1}^{N}(y_n - \sigma(w^T\phi(x_n)))\phi(x_n) - \alpha(w - m_0) = 0$$

We then find the Hessian of the log posterior at the mode.

$$H = -\frac{\partial^2}{\partial w^2}\log p(w|D, m_0, \alpha) = \alpha I + \sum_{n=1}^{N} y_n(1 - y_n)\phi(x_n)\phi(x_n)^T$$

The posterior distribution is given by

$$p(w|D, m_0, \alpha) = N(w|w_{\text{MAP}}, H^{-1})$$

Predictive distribution: For a given category t, the predictive distribution is given by

$$p(t|x, D, m_0, \alpha) = \int p(t|x, w)p(w|D, m_0, \alpha)dw$$

We first find the distribution $p(t = 1|x, D, m_0, \alpha)$, and find the distribution $p(t = 0|x, D, m_0, \alpha)$ by $1 - p(t = 1|x, D, m_0, \alpha)$.

$$\begin{aligned}p(t = 1|x, D, m_0, \alpha) &= \int \sigma(w^T\phi(x))p(w|D, m_0, \alpha)dw \\ &= \int \sigma(w^T\phi(x))N(w|w_{\text{MAP}}, H^{-1})dw\end{aligned}$$

Let $a = w^T\phi(x)$, we have

$$\sigma(w^T\phi(x)) = \int \delta(a - w^T\phi(x))\sigma(a)da$$

Thus, we have

$$\int \sigma(w^T\phi(x))p(w|D,m_0,\alpha)dw = \int \sigma(a)p(a)da$$

where

$$p(a) = \int \delta(a - w^T\phi(x))N(w|w_{\text{MAP}}, H^{-1})dw$$

The predictive distribution is given by

$$p(t=1|x,D,m_0,\alpha) = \sigma(\kappa(\sigma_a^2)\mu_a)$$

where

$$\mu_a = w_{\text{MAP}}^T\phi(x)$$
$$\sigma_a^2 = \phi(x)^T H^{-1}\phi(x)$$

Model evaluation:

$$p(D|m_0,\alpha) = \int p(D|w)p(w|m_0,\alpha)dw$$

The exact form can be found by substituting the likelihood with Bernoulli distribution and the prior with Gaussian distribution.

$$p(D|m_0,\alpha) = \int \left(\prod_{n=1}^{N} \sigma(w^T\phi(x_n))^{y_n}(1-\sigma(w^T\phi(x_n)))^{1-y_n}\right) N(w|m_0,\alpha^{-1}I)dw$$

Similar to Problem 3, the integral is intractable. We can use Laplace approximation to approximate the integral.

$$\begin{aligned} p(D|m_0,\alpha) &\approx p(D|w_{\text{MAP}})p(w_{\text{MAP}}|m_0,\alpha)\frac{(2\pi)^{M/2}}{|H|^{1/2}} \\ &= \left(\prod_{n=1}^{N} \sigma(w_{\text{MAP}}^T\phi(x_n))^{t_n}(1-\sigma(w_{\text{MAP}}^T\phi(x_n)))^{1-t_n}\right) \\ &\times N(w_{\text{MAP}}|m_0,\alpha^{-1}I)\frac{(2\pi)^{M/2}}{|\alpha I + \sum_{n=1}^{N} y_n(1-y_n)\phi(x_n)\phi(x_n)^T|^{1/2}} \end{aligned}$$

Problem 5

(1)

For the given neural network, we have

$$\begin{aligned} \frac{\partial y}{\partial w^{(1)}} &= \frac{\partial y}{\partial a_2}\frac{\partial a_2}{\partial z}\frac{\partial z}{\partial a_1}\frac{\partial a_1}{\partial w^{(1)}} \\ &= y(1-y)w^{(2)}h'(a_1)x \end{aligned}$$

$$\begin{aligned} \frac{\partial y}{\partial w^{(2)}} &= \frac{\partial y}{\partial a_2}\frac{\partial a_2}{\partial w^{(2)}} \\ &= y(1-y)z \end{aligned}$$

$$\begin{aligned}\frac{\partial y}{\partial a_1} &= \frac{\partial y}{\partial a_2}\frac{\partial a_2}{\partial z}\frac{\partial z}{\partial a_1} \\ &= y(1-y)w^{(2)}h'(a_1)\end{aligned}$$

$$\frac{\partial y}{\partial a_2} = y(1-y)$$

$$\begin{aligned}\frac{\partial y}{\partial x} &= \frac{\partial y}{\partial a_2}\frac{\partial a_2}{\partial z}\frac{\partial z}{\partial a_1}\frac{\partial a_1}{\partial x} \\ &= y(1-y)w^{(2)}h'(a_1)w^{(1)}\end{aligned}$$

(2)

Regression: The loss function is given by

$$E = \frac{1}{2}(y-t)^2$$

We find the derivative of the loss function with respect to y.

$$\frac{\partial E}{\partial y} = y - t$$

Thus, the update rules for $w^{(1)}$ and $w^{(2)}$ are given by

$$w^{(1)} \leftarrow w^{(1)} - \eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial w^{(1)}} = w^{(1)} - \eta(y-t)y(1-y)w^{(2)}h'(a_1)x$$

$$w^{(2)} \leftarrow w^{(2)} - \eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial w^{(2)}} = w^{(2)} - \eta(y-t)y(1-y)z$$

Classification: The loss function is given by

$$E = -t\log y - (1-t)\log(1-y)$$

We find the derivative of the loss function with respect to y.

$$\frac{\partial E}{\partial y} = -\frac{t}{y} + \frac{1-t}{1-y} = \frac{y-t}{y(1-y)}$$

Thus, the update rules for $w^{(1)}$ and $w^{(2)}$ are given by

$$w^{(1)} \leftarrow w^{(1)} - \eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial w^{(1)}} = w^{(1)} - \eta(y-t)w^{(2)}h'(a_1)x$$

$$w^{(2)} \leftarrow w^{(2)} - \eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial w^{(2)}} = w^{(2)} - \eta(y-t)z$$

Problem 6

a)

Given a neural network $t = y(w,x) + v$, where $v \sim N(v|0,\beta^{-1})$ and $w \sim N(w|m_0,\alpha^{-1}I)$.

Posterior distribution:

$$p(w|D,\beta,m_0,\alpha) \propto p(D|w,\beta)p(w|m_0,\alpha)$$

The likelihood is given by

$$p(D|w,\beta) = \prod_{n=1}^{N} N(t_n|y(w,x_n),\beta^{-1})$$

The prior is given by

$$p(w|m_0,\alpha) = N(w|m_0,\alpha^{-1}I)$$

Thus, the log posterior is given by

$$\log p(w|D,\beta,m_0,\alpha) = -\frac{\beta}{2}\sum_{n=1}^{N}(t_n - y(w,x_n))^2 - \frac{\alpha}{2}(w-m_0)^T(w-m_0) + \text{const}$$

We find the negative of second derivative of the log posterior at the mode

$$A = -\frac{\partial^2}{\partial w^2}\log p(w|D,\beta,m_0,\alpha) = \alpha I + \beta H$$

where H is the Hessian of the sum of squared error function.
We can find w_{MAP} by repeatedly updating w using the gradient of the log posterior.

$$w_{\text{MAP}} \leftarrow w = w - A^{-1}\nabla \log p(w|D,\beta,m_0,\alpha)$$

The posterior distribution is given by

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

Predictive distribution:

$$p(t|x,D,\beta,m_0,\alpha) = \int p(t|x,w)p(w|D,\beta,m_0,\alpha)dw$$

Let $g = \nabla_w y(w,x)|_{w=w_{\text{MAP}}}$, we have

$$p(t|x,D,\beta,m_0,\alpha) = N(t|y(w_{\text{MAP}},x),\beta^{-1} + g^T A^{-1} g)$$

Model evaluation:

$$p(D|\beta,m_0,\alpha) = \int p(D|w,\beta)p(w|m_0,\alpha)dw$$

With Laplace approximation, we can approximate the integral by

$$\begin{aligned} p(D|\beta,m_0,\alpha) &\approx p(D|w_{\text{MAP}},\beta)p(w_{\text{MAP}}|m_0,\alpha)\frac{(2\pi)^{M/2}}{|A|^{1/2}} \\ &= \left(\prod_{n=1}^{N} N(t_n|y(w_{\text{MAP}},x_n),\beta^{-1})\right) N(w_{\text{MAP}}|m_0,\alpha^{-1}I)\frac{(2\pi)^{M/2}}{|\alpha I + \beta H|^{1/2}} \end{aligned}$$

where M is the dimension of w.

b)

Given a neural network $y = \sigma(a(w, x))$, where $w \sim N(w|0, \alpha^{-1}I)$.

Posterior distribution:

$$p(w|D, \alpha) \propto p(D|w)p(w|\alpha)$$

The likelihood is given by

$$p(D|w) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}$$

The prior is given by

$$p(w|\alpha) = N(w|0, \alpha^{-1}I)$$

Thus, the log posterior is given by

$$\log p(w|D, \alpha) = \sum_{n=1}^{N}\{t_n \log y_n + (1 - t_n)\log(1 - y_n)\} - \frac{\alpha}{2}w^T w + \text{const}$$

We find the negative of second derivative of the log posterior at the mode

$$A = -\frac{\partial^2}{\partial w^2}\log p(w|D, \alpha) = \alpha I + H$$

where H is the Hessian of the cross entropy function.
We can find w_{MAP} by repeatedly updating w using the gradient of the log posterior.

$$w_{\text{MAP}} \leftarrow w = w - A^{-1}\nabla \log p(w|D, \alpha)$$

The posterior distribution is given by

$$p(w|D, \alpha) = N(w|w_{\text{MAP}}, A^{-1})$$

Predictive distribution:

$$p(t|x, D, \alpha) = \int p(t|x, w)p(w|D, \alpha)dw$$

Let $a(x, w) = a_{\text{MAP}}(x) + b^T(w - w_{\text{MAP}})$, where $a_{\text{MAP}}(x) = a(w_{\text{MAP}}, x)$ and $b = \nabla a(x, w_{\text{MAP}})$, we have

$$p(a|x, D, \alpha) = \int \delta(a - a_{\text{MAP}}(x) - b^T(w - w_{\text{MAP}}))p(w|D, \alpha)dw$$

Thus, the predictive distribution is given by

$$p(t = 1|x, D, \alpha) = \sigma(\kappa(\sigma_a^2)a_{\text{MAP}}(x))$$

where

$$\sigma_a^2 = b^T A^{-1} b$$

Model evaluation:

$$p(D|\alpha) = \int p(D|w)p(w|\alpha)dw$$

With Laplace approximation, we can approximate the integral by

$$\begin{aligned} p(D|\alpha) &\approx p(D|w_{\text{MAP}})p(w_{\text{MAP}}|\alpha)\frac{(2\pi)^{M/2}}{|A|^{1/2}} \\ &= \left(\prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}\right) N(w_{\text{MAP}}|0,\alpha^{-1}I)\frac{(2\pi)^{M/2}}{|\alpha I + H|^{1/2}} \end{aligned}$$

where M is the dimension of w.

Problem 7

a)

1. The primal problem of SVM is computationally expensive to solve, especially when the number of data points is large. The dual problem is more efficient to solve.
2. The dual problem form allows us to use the kernel trick to solve the data points which are not linearly separable in the original space.
3. The dual problem form also allows us to use the KKT conditions to solve the problem more efficiently.

b)

i)

For SVM classification:

1. Cost function: $E(w,b) = \frac{1}{2}||w||^2 + C\sum_{n=1}^{N}\xi_n$
2. Constraints: $t_n(w^T\phi(x_n)+b) \geq 1-\xi_n$, $\xi_n \geq 0$
3. Predictions: $y(x) = \text{sign}(w^T\phi(x)+b)$

For logistic regression:

1. Cost function: $E(w) = -\sum_{n=1}^{N}\{t_n \log y_n + (1-t_n)\log(1-y_n)\}$
2. Constraints: None
3. Predictions: $y(x) = \sigma(w^T\phi(x))$

ii)

For ν-SVM regression:

1. Cost function: $E(w,b) = \frac{1}{2}||w||^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*)$
2. Constraints: $|t_n - w^T\phi(x_n) - b| \leq \epsilon + \xi_n$, $\xi_n, \xi_n^* \geq 0$, $\sum \xi_n + \xi_n^* \leq \nu N$
3. Predictions: $y(x) = \sum_{n=1}^{N}(a_n - a_n^*)K(x, x_n) + b$

For least squares regression:

1. Cost function: $E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n) - b)^2$
2. Constraints: None
3. Predictions: $y(x) = w^T\phi(x)$ + b

For both questions, SVM can handle non-linearity better with kernel trick. SVM can also use hyperparameters to control the trade-off between the complexity of the model and the error of the model.

c)

1. Logistic activation function can be viewed as binary classification, and its output can be interpreted as the probability of the data point belonging to the positive class.
2. Logistic activation function introduces non-linearity to the model, which allows the model to learn complex patterns in the data.
3. Logistic activation function is always differentiable, and the gradient can be easily computed, which allows back-propagation to be used efficiently.
4. Logistic activation function smooth the data and make the model less sensitive to outliers.

d)

i)

For logistic activation function and tanh activation function, we have

$$\tanh(x) = 2\sigma(2x) - 1$$

Thus, those two activation functions are interchangeable in some sense: if we apply extra linear transformation to the input and output of the logistic activation function, we can get the tanh activation function.
The most important difference between the three activation functions is their output range. The output of the logistic activation function is in the range of $(0, 1)$, the output of the tanh activation function is in the range of $(-1, 1)$, and the output of the ReLU activation function is in the range of $[0, \infty)$.
Another difference is the vanishing gradient problem. The gradient of the logistic activation function and the tanh activation function is always less than 1, which can cause the

vanishing gradient problem in deep neural networks. However, the gradient of the ReLU activation function is either 0 or 1, which can avoid the vanishing gradient problem (since neurons activated by ReLU receive full gradient).
Speed is another difference. The ReLU activation function is faster to compute than the logistic and tanh activation functions.

ii)

Logistic activation function is used in the output layer of a binary classification problem, and other layers where we want to model a gating mechanism. tanh activation function is used when the output should be centered around zero. ReLU activation function is used when efficiency is important, or we want to avoid the vanishing gradient problem.

e)

Jacobi matrix: Jacobi matrix is the matrix of all first-order partial derivatives. It is the key component in the back-propagation algorithm, which is used to compute the gradient of the loss function with respect to the weights with proper chain rule.

Hessian matrix: Hessian matrix is the matrix of all second-order partial derivatives. It is used to compute the curvature of the loss function, which can be used to determine the convergence of the optimization algorithm. Also, the inverse of the Hessian matrix can be used to determine the relative importance of the weights, thus can be used as step size in the optimization algorithm. In Bayesian models, the Hessian matrix is involved in the Laplace approximation for model evaluation.

f)

The exponential family is a class of probability distributions that can be written in the form

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - A(\eta))$$

It is important in machine learning because it includes many common distributions, such as Gaussian, Bernoulli, Poisson distributions. The exponential family has an important property that is called conjugate prior, which means that the posterior distribution is in the same family as the prior distribution. This property makes the exponential family distributions easy to work with in Bayesian models.
Some distributions that are not in the exponential family include the Cauchy distribution and the Student's t-distribution.

g)

KL divergence is a measure of how one probability distribution is different from a second, reference probability distribution. In machine learning, KL divergence is used to measure the difference between the true distribution (real data) and the predicted distribution (model output). Thus, one important application of KL divergence is in model evaluation, where we can use KL divergence as part of the loss function to measure how close the model output is to the real data (it is proved that minimizing the KL divergence is equivalent to maximizing the likelihood, in PRML section 1.6.1).

KL divergence can also be used to calculated mutual information, which is a measure of how much information one random variable contains about another random variable. Thus, another important application of KL divergence is in feature selection, where we can use KL divergence to measure how much information a feature contains about the target variable.

h)

Data augmentation can be viewed as adding noise to the data. This encourages the model to learn the invariant features of the data, which can improve the generalization of the model. Thus, we say that data augmentation can be used to regularize the model.

i)

1. The central limit theorem states that the sum of numerous independent and identically distributed random variables will be approximately normally distributed. Thus, using Gaussian distribution to model the data is a good choice.

2. The Gaussian distribution is easy to work with in Bayesian models. Since the Gaussian distribution is in the exponential family, it has a conjugate prior, which makes the posterior distribution easy to compute.

3. With fixed mean and variance, the Gaussian distribution has the maximum entropy among all distributions. This means that the Gaussian distribution is the most uncertain distribution given the mean and variance, which makes it a good choice for modeling uncertainty.

4. Applying linear transformation to the Gaussian distribution will still result in a Gaussian distribution. This property makes the Gaussian distribution a good choice for linear models.

5. The marginal distribution and conditional distribution of the Gaussian distribution are also Gaussian distributions. This property makes the Gaussian distribution easy to work with.

6. The Gaussian distribution handles the outliers well, since the Gaussian distribution gives extremely low probability to the data points that are far from the mean.

j)

The Laplacian approximation eliminates the need to compute the integrals. And since in many cases the integrals are intractable (such as in the posterior distribution of the Bayesian models), the Laplacian approximation is a good choice to approximate them.
Also, rather than observing all the data points, the Laplacian approximation only needs to observe the mode of the posterior distribution. This makes the Laplacian approximation more efficient in terms of computation.

k)

1. Bias-variance trade-off: The bias-variance trade-off is the trade-off between the bias of the model and the variance of the model. A model with high bias will underfit the data, while a model with high variance will overfit the data. The goal is to find a model that has a good balance between bias and variance.

2. Regularization: Regularization is a technique used to prevent overfitting. It adds a penalty term to the loss function, which discourages the model from learning complex patterns in the data. Regularization can be used to control the complexity of the model.

3. Number of parameters: The number of parameters in the model is an important factor in determining the complexity of the model. A model with too many parameters will likely overfit the data, while a model with too few parameters will likely underfit the data.

l)

Training Regression Model:

1. The features should be relevant to the target variable. This can be measured by the correlation between the features and the target variable.

2. The features should be mostly independent of each other. This can be measured by the correlation between the features.

3. The features should not have high dimensionality, otherwise, the model will likely overfit the data.

Training Classification Model:

1. The features should be relevant to the target variable. This can be measured by the correlation between the features and the target variable.

2. The features should be mostly independent of each other. This can be measured by the correlation between the features.

3. The features should be mostly balanced, otherwise, the model will likely be biased towards the majority class.

Testing Model:

1. The data should be split into training set, validation set, and test set. The training set is used to train the model, the validation set is used to tune the hyperparameters, and the test set is used to evaluate the model.

2. If possible, we can use k-fold cross-validation to evaluate the model. This can give us a more reliable estimate of the model performance.

3. We should randomly sample the data points and balance the classes in the test set. This can give us a more reliable estimate of the model performance.

m)

1. When prior knowledge is available: MAP models can leverage the prior knowledge to improve the model performance.
2. When the data is limited: MAP models can prevent overfitting small datasets by incorporating the prior knowledge.
3. When the model is complex: MAP models is equivalent to adding a regularization term to the loss function, which can prevent overfitting in complex models.

Problem 8

(1)

Generative approach: In the generative approach, we model the conditional distribution for each class $p(x|C_k)$ and the prior distribution $p(C_k)$. Then we can use the Bayes' theorem to compute the posterior distribution $p(C_k|x)$

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

Then, we can use decision theory with the posterior distribution to make predictions.
The advantage of the generative approach is that, since we know $p(x) = \sum_k p(x|C_k)p(C_k)$, we can detect the outliers in the data (e.g., the data points that have low $p(x)$). Also, we can generate new data points with $p(x|C_k)$.
The disadvantages of the generative approach is that it is computationally expensive to model the conditional distribution for each class, especially when the number of classes is large.
One example of the generative approach is the Naive Bayes classifier. For example, we model the probability of getting sick given the symptoms $p(\text{sick}|\text{symptoms})$. Firstly, we find the prior distribution $p(\text{sick}) = \frac{\text{number of sick people}}{\text{total number of people}}$. Then we find the conditional distribution $p(\text{symptoms}|\text{sick}) = \frac{\text{number of sick people with the symptoms}}{\text{number of sick people}}$ and $p(\text{symptoms}|\text{healthy}) = \frac{\text{number of healthy people with the symptoms}}{\text{number of healthy people}}$. Then we can use the Bayes' theorem to compute the posterior distribution $p(\text{sick}|\text{symptoms})$.

Discriminative approach: In the discriminative approach, we model the posterior distribution directly $p(C_k|x)$. Then we can use decision theory with the posterior distribution to make predictions.
The advantage of the discriminative approach is that it is computationally efficient, since we only need to model the posterior distribution.
The disadvantages of the discriminative approach is that the model cannot handle missing data, since we do not have the prior distribution.
One example of the discriminative approach is the logistic regression. For example, we model the probability of getting sick given the symptoms $p(\text{sick}|\text{symptoms})$. We model the posterior distribution directly $p(\text{sick}|\text{symptoms}) = \sigma(w^T\text{symptoms})$, where σ is the logistic activation function.

(2)

GAN model uses both generative and discriminative approaches. In the GAN model, there is a generator that generates the data points, and a discriminator that discriminates the generated data points from the real data points.
The generator is a generative model. It learns to generate data point that mimics real data distribution $p(x)$ (similar to the likelihood in the generative approach, but with only one class). It takes a random noise as input and generates the data points.
The discriminator is a discriminative model. It learns to discriminate the generated data points from the real data points. It takes the data points as input and outputs the probability of the data points being real, which is same as the discriminative approach where we model the posterior distribution directly as $p(C_k|x) = \text{discriminator}(x)$.