Machine Learning Midterm Exam

Problem I. Least Square

a)

$$
\begin{aligned}
E(X) &= \frac{1}{2}(y - AX)^T Q^{-1}(y - AX) \\
\frac{\partial E(X)}{\partial X} &= A^T Q^{-1}(y - AX) = 0 \\
\Rightarrow \quad X &= (A^T Q^{-1} A)^{-1} A^T Q^{-1} y
\end{aligned}
$$

b) Using Lagrange multiplier, we have

$$
\begin{aligned}
&L(\lambda, X) = \frac{1}{2}(y - AX)^T Q^{-1}(y - AX) + \lambda(b^T X - c) \\
&\quad \Rightarrow \quad \begin{cases} \frac{\partial L}{\partial \lambda} = b^T X - c = 0 \\ \frac{\partial L}{\partial X} = A^T Q^{-1}(y - AX) + \lambda b^T = 0 \end{cases} \\
&\Rightarrow \quad \hat{X} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - (A^T A) b^T [b(A^T A)^{-1} b^T]^{-1} (b(A^T Q^{-1} A)^{-1} A^T Q^{-1} Y - c)
\end{aligned}
$$

c) Using Lagrange multiplier, we have

$$
\begin{aligned}
L(\lambda_1, \lambda_2, X) &= \frac{1}{2}(y - AX)^T Q^{-1}(y - AX) + \lambda_1(b^T X - c) + \lambda_2(X^T X - d) \\
\Rightarrow \quad &\begin{cases} \frac{\partial L}{\partial \lambda_1} = b^T X - c = 0 \\ \frac{\partial L}{\partial \lambda_2} = X^T X - d = 0 \\ \frac{\partial L}{\partial X} = A^T Q^{-1}(y - AX) + \lambda_1 b^T + 2\lambda_2 X = 0 \end{cases} \\
\Rightarrow \quad &X = 2(A^T Q^{-1} A - dI)^{-1}(2A^T Q^{-1} Y - bI)
\end{aligned}
$$

Problem II. Linear Gaussian System

$$
\begin{aligned}
p(Y|X) &= N(Y|AX, \beta^{-1} I) \\
p(X, Y) &= p(X)\, p(Y|X) = N(X|m_0, \Sigma_0) N(y|AX, \beta^{-1} I) \\
p(Y) &= N(Y|Am_0, \beta^{-1} I + A\Sigma_0 A^T) \\
p(X|Y = y, \beta, m_0, \Sigma_0) &= N(X|(\Sigma_0^{-1} + A^T \beta I A)^{-1}(A^T \beta I Y + \Sigma_0^{-1} m_0), (\Sigma_0^{-1} + A^T \beta I A)^{-1}) \\
p(\hat{Y}|Y = y, \beta, m_0, \Sigma_0) &= N(\hat{Y}|A\Sigma(A^T \beta I Y + \Sigma_0^{-1} m_0), \beta^{-1} I + A\Sigma A^T) \\
p(Y|\beta, m_0, \Sigma_0) &= N(Y|Am_0, \beta^{-1} I + A\Sigma_0 A^T)
\end{aligned}
$$

Problem III. Linear Regression

Because

$$p(w|D,\beta,m_0,\alpha) \propto p(D|w,\beta)\ p(w|m_0,\alpha)$$
$$\text{where} \begin{cases} p(D|w,\beta) = \prod_n p(y_n|w,\beta,\Phi_n) = \prod_n N(y_n|w^T\Phi_n,\beta^{-1}) \\ p(w|m_0,\alpha) = N(w|m_0,\alpha^{-1}I) \end{cases}$$

Then we have

$$p(w|D,\beta,m_0,\alpha) \propto \exp(-\frac{1}{2}\beta\Sigma_n(y_n - w^T\Phi_n)^2 - \frac{1}{2}\alpha(w-m_0)^T(w-m_0)$$
$$\Rightarrow \quad p(w|D,\beta,m_0,\alpha) = N(w|\mu,\Sigma)$$
$$\text{where} \begin{cases} \mu = (\beta\sum_n \Phi_n\Phi_n^T + \alpha I)^{-1}(\beta\sum_n \Phi_n y_n + \alpha m_0) \\ \Sigma = (\beta\sum_n \Phi_n\Phi_n^T + \alpha I)^{-1} \end{cases}$$

$$p(\hat{y}|\hat{x},D,\beta,m_0,\alpha) = \int p(\hat{y}|\hat{x},w,\beta)\ p(w|D,\beta,m_0,\alpha_0)\ dw$$
$$= N(\hat{y}|\hat{\mu},\hat{\sigma}^2)$$
$$\text{where} \begin{cases} \hat{\mu} = \mu^T\Phi \\ \hat{\sigma}^2 = \Phi\Sigma\Phi^T + \beta^{-1} \end{cases}$$

$$p(D|\beta,m_0,\alpha) = N(D|0,\Sigma_p)$$
$$\text{where } \Sigma_p = (\beta^{-1}I + \Sigma_n\Phi_n\Phi_n^T)^{-1}$$

Problem IV. Logistic Regression

Since

$$p(w|D,m_0,\alpha) \propto p(D|w)\ p(w|m_0,\alpha)$$
$$\text{where} \begin{cases} p(D|w) = \prod_n p(t_n|w) = \prod_n y_n^{t_n}(1-y_n)^{1-t_n} \\ p(w|m_0,\alpha) = N(w|m_0,\alpha^{-1}I) \end{cases}$$

Using Laplace's method, we get

$$\log p(w|D,m_0,\alpha) = \log p(D|w) + \log p(w|m_0,\alpha)$$
$$\nabla_w(-\log p(w|D,m_0,\alpha)) = -\sum_n(t_u - y_u)\Phi_n + \alpha(w-m_0) = 0$$
$$\Rightarrow \ w = \frac{\sum_n(t_n - y_n)\Phi_n}{\alpha} + m_0$$

which gives us the mode of the posterior distribution, then

$$p(t|x,D,m_0,\alpha) \propto p(t|\hat{w},x)\ p(\hat{w}|D,m_0,\alpha)$$
$$\text{where } \ \hat{w} = \frac{\sum_n(t_n - y_n)\Phi_n}{\alpha} + m_0$$
$$p(D|m_0,\alpha) = \int \prod_n y_n^{t_n}(1-y_n)^{1-t_n} N(w|m_0,\alpha^{-1}I)dw$$

Problem V. Neural Network

(1)

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

(2)

$$
\begin{aligned}
E &= \frac{1}{2}(y-t)^2 \\
\frac{\partial E}{\partial y} &= y-t \\
\frac{\partial E}{\partial w^{(1)}} &= \frac{\partial E}{\partial y}\frac{\partial y}{\partial w^n} = (y-t)y\,(1-y)z \\
\Rightarrow \quad \nabla w^{(1)} &= -\alpha\,(y-t)y\,(1-y)z \\
&\text{where } \alpha \text{ is the learning rate} \\
\frac{\partial E}{\partial w^{(2)}} &= \frac{\partial E}{\partial y}\frac{\partial y}{\partial w^{(2)}} = (y-t)y\,(1-y)w^{(2)}h'(a_1)x \\
\Rightarrow \quad \nabla w^{(2)} &= -\alpha\,(y-t)y\,(1-y)w^{(2)}h'(a_1)x \\
&\text{where } \alpha \text{ is the learning rate}
\end{aligned}
$$

Problem VI. Bayesian Neural Network

a) By Bayes' theorem, we have

$$
\begin{aligned}
p(w|D,\beta,m_0,\alpha) &= \frac{p(D|w,\beta,\alpha,m_0)\;p(w|\beta,\alpha,m_0)}{p(D|\beta,\alpha,m_0)} \\
p(D|w,\beta,\alpha,m_0) &= \prod_n p(t_n|w_n,\beta) \qquad (1)
\end{aligned}
$$

Since $v \sim N(0,\beta^{-1})$, we have

$$
p(t_n|w_n,\beta) = \sqrt{\frac{\beta}{2\pi}}e^{-\frac{\beta}{2}(t_n-y(w_n,x_n))^2} \qquad (2)
$$

Since $w \sim N(m_0,\alpha^{-1}I)$, we have

$$
p(w|\beta,\alpha,m_0) = \sqrt{\frac{\alpha}{2\pi}}e^{-\frac{\alpha}{2}(w-m_0)^T(w-m_0)} \qquad (3)
$$

By $(1)(2)(3)$ we can get posterior distribution $p(w|D,\beta,m_0,\alpha)$, then

$$
\begin{aligned}
p(D|\beta, m_0, \alpha) &= \int p(|w, \beta)\ p(w|\beta, \alpha, m_0)\ dw \\
p(t|x, D, \beta, m_0, \alpha) &= \int p(t|w, x, \beta)\ p(w|D, \beta, m_0, \alpha)\ dw
\end{aligned}
$$

b) By Bayes' theorem, we have

$$
\begin{aligned}
p(w|D, \alpha) &= \frac{p(D|w, \alpha)p(w|\alpha)}{p(D|\alpha)} \\
p(D|w, \alpha) &= \prod_n p(t_n|w_n, \alpha) \\
&= \prod_n y(w_n, x_n)^{t_n}(1 - y(w_n, x_n))^{1-t_n} \qquad (1)
\end{aligned}
$$

Since $w \sim N(0, \alpha^{-1}I)$, we have

$$
\begin{aligned}
p(w|\alpha) &= \sqrt{\frac{\alpha}{2\pi}} e^{-\frac{\alpha}{2}w^T w} \qquad (2) \\
p(D|\alpha) &= \int p(D|w, \alpha)p(w|\alpha)dw \qquad (3)
\end{aligned}
$$

Using $(1)(2)(3)$, we can get posterior distribution $p(w|D, \alpha)$, then

$$
\begin{aligned}
p(D|\alpha) &= \int p(D|w, \alpha)p(w|\alpha)dw \\
p(t|x, D, \alpha) &= \int p(t|w, x, \alpha)p(w|D, \alpha)dw
\end{aligned}
$$

Problem VII. Critical Analyses

a)

1)

- **Cost function**
  - SVM: hinge loss
  - LR: cross-entropy loss
- **Constrain**
  - SVM: use the concept of margin to find the optimal decision boundary, which aims at maximizing the margin while still classifying the data
  - LR: model the probabilities and maximize the likelihood
- **Prediction**
  - SVM: it predicts based on the position of the data points relative to the decision boundary

    LR: it predicts the probabilities (sigmoid), and then applies a threshold (0.5)

2)

- **Cost function**
  - v-SVM: epsilon-insensitive loss
  - LSR: MSE

- **Constrain**
  - v-SVM: introduces a parameter called v, which coutrols the trade-off between the margin size and the outlier.
  - LSR: finds the line that minimizes the loss function.
- **Prediction**:
  - v-SVM: predicts based on the position of the data points relative to the decision hyperplane.
  - LSR: predicts the tanget value based on fitted line.

b)

1. Non-linearity
2. Output range (0,1) or (-1,1)
3. Smoothness and differentiability

c)

- Difference:
  - logistic: map value to (0.1). Smooth but suffers from vanishing gradient problem.
  - ReLU: max(0,x). It is computational efficient but it can lead to dead neurons.
  - tanh: map valve to (-1,1). Similow to logistic. but has a steeper slope.
- When:
  - logistic: predict probabilities
  - ReLU: in hidden layers of deep network
  - tanh: in hidden layers when the data needs to be centered around 0

d) Gradient-based optimization: Jacobian and Hessian matrix provide information about first-order and second-order gradient

e)

1. **Mathematical tractability**: Exponentid family dstributron have properties which facilitate caloulation.
2. **Flexibility**: It encompasses a wide range of distribution like Gaussian, Poisson, which is helpful to model data.
3. **Cauchy distrbution**: Student-t distribution.

f)

- **Model comparison**: KL divergence can be used to measure difference between two distributions
- **Regularization**: KL divergence can be incorporated as a regularization term to prevent overfitting
- **VAE**: information retrival

g)

- **Increase generalization**: increase the size of the training set and add more romdomness to the data.
- **Robustness to variation**: model are more robust to the noise in the input data
- **Regularization**: it can add some noise to the original dataset, preventing the model from overfitting

h)

- **Center Limit Theorem**: the sum of a large number of independent and identically distributed random variables tends to follow Gaussian distribution
- **Simplicity and tractability**: most of the calculation over Gaussian distribution is still Gaussian distribution
- **Robustness to noise and outlier**: Gaussian has thin tails, meaning that extreme values have low probability

i)

- **Incorporation of prior knowledge**: by using bayes inference, the MAP combines the prior disttibution of the parameters with the likelihood function, resulting in a posterior distribution
- **Regularization**: The MAP model acts as a form of regularization by adding a term to the likelihood function, which helps prevent overfitting

Problem VIII. Discussion

- **Generative**: use Bayes' theorem

  $$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}$$

  advantage/disadvantage: it can handle missing data, but it might be computational expensive

  eg. Naive Bayes

- **Discriminative**: assume some functional form for $P(Y|X)$ and then estimate the parameters of $P(Y|X)$ with the help of data

  advantage/disadvantage: it focus on decision boundary and its compution is more efficient. However, it is sensitive to data distribution and noise data

  eg. Logistic Regression Classifier