Machine Learning	Kernel FDA,logistic regression,generative method

Lecture Note 12: FDA and Linear Classification

Professor: Zhihua Zhang *Scribes: Kainan Wang, Tianfan Fu*

1 Fisher Discriminant Analysis (Cont'd)

1.1 Kernel FDA

Suppose there exists a kernel $K(\mathbf{x},\mathbf{y}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y})\rangle = \boldsymbol{\phi}(\mathbf{x})^T\boldsymbol{\phi}(\mathbf{y})$ such that $\boldsymbol{\phi}(\cdot)$ projects a vector $\mathbf{x}$ in $\mathbb{R}^p$ into some higher dimensional space $\mathbb{R}^r$. Applying this to some sample data $\mathbf{X}$ (an $n \times p$ matrix), we have the transformation

$$\mathbf{X} \to \boldsymbol{\Phi},$$

where $\boldsymbol{\Phi}$ is an $n \times r$ matrix. We usually have no idea about $\boldsymbol{\Phi}$, but we do know the kernel K, which means we know $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$. If we wish to do FDA with respect to the transformed data $\boldsymbol{\Phi}$, explicitly calculating the data in the higher dimensional space can be avoidable. We first review the results in previous chapters.

$$\begin{gathered}
\widehat{\mathbf{S}}_t = \boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Phi} = \boldsymbol{\Phi}^T\mathbf{H}\mathbf{H}\boldsymbol{\Phi} \\
\widehat{\mathbf{S}}_b = \boldsymbol{\Phi}^T\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\boldsymbol{\Phi} \\
\boldsymbol{\Pi} = \mathbf{diag}(\pi_1, \pi_2, ..., \pi_c) \\
\widehat{\mathbf{S}}_b\widehat{\mathbf{A}} = \widehat{\mathbf{S}}_t\widehat{\mathbf{A}}\boldsymbol{\Lambda}
\end{gathered}$$

Denote $\widehat{\mathbf{A}} = \boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Psi} + \mathbf{N}$ subject to $\mathbf{N}^T\boldsymbol{\Phi}^T\mathbf{H} = \mathbf{0} \Rightarrow \mathbf{H}\boldsymbol{\Phi}\mathbf{N} = \mathbf{0}$ (why we can do this decomposition? hint: try to construct $\boldsymbol{\Psi}$ and $\mathbf{N}$ column by column) and we have

$$\begin{gathered}
(\boldsymbol{\Phi}^T\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\boldsymbol{\Phi})\widehat{\mathbf{A}} = (\boldsymbol{\Phi}^T\mathbf{H}\mathbf{H}\boldsymbol{\Phi})\widehat{\mathbf{A}}\boldsymbol{\Lambda} \\
\mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\boldsymbol{\Phi}\widehat{\mathbf{A}} = \mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{H}\boldsymbol{\Phi}\widehat{\mathbf{A}}\boldsymbol{\Lambda} \\
\mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\boldsymbol{\Phi}(\boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Psi} + \mathbf{N}) = \mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{H}\boldsymbol{\Phi}(\boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Psi} + \mathbf{N})\boldsymbol{\Lambda} \\
\mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Psi} = \mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\mathbf{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{H}\boldsymbol{\Psi}\boldsymbol{\Lambda} \\
\mathbf{H}\mathbf{K}\mathbf{H}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{H}\mathbf{K}\mathbf{H}\boldsymbol{\Psi} = \mathbf{H}\mathbf{K}\mathbf{H}\mathbf{H}\mathbf{K}\mathbf{H}\boldsymbol{\Psi}\boldsymbol{\Lambda}
\end{gathered}$$

Denote $\mathbf{C} = \mathbf{HKH}$, we have

$$\mathbf{C}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{C}\boldsymbol{\Psi} = \mathbf{C}\mathbf{C}\boldsymbol{\Psi}\boldsymbol{\Lambda}$$

We have at least three ways to solve this problem

1. using psudo-inverse: $(\mathbf{C}\mathbf{C})^\dagger\mathbf{C}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{C}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$
2. adding the item $\sigma\mathbf{I}_n$: $(\mathbf{C}\mathbf{C} + \sigma\mathbf{I}_n)^{-1}\mathbf{C}\mathbf{E}\boldsymbol{\Pi}^{-1}\mathbf{E}^T\mathbf{C}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$

3. GSVD

$$\begin{pmatrix} \mathbf{\Pi}^{-\frac{1}{2}}\mathbf{E}^{\mathbf{T}}\mathbf{C} \\ \mathbf{C} \end{pmatrix}$$

Finally, for a input data $\mathbf{x}_0$, we transform it to $\boldsymbol{\phi}_0$, decentralize it as $\boldsymbol{\phi}_0 - \frac{1}{n}\mathbf{\Phi}^T\mathbf{1}_n$ and do the FDA:

$$\begin{aligned}\widehat{\mathbf{A}}^T(\boldsymbol{\phi}_0 - \frac{1}{n}\mathbf{\Phi}^T\mathbf{1}_n) &= \mathbf{\Psi}^T\mathbf{H}\mathbf{\Phi}(\boldsymbol{\phi}_0 - \frac{1}{n}\mathbf{\Phi}^T\mathbf{1}_n) \\ &= \mathbf{\Psi}^T\mathbf{H}(\mathbf{\Phi}\boldsymbol{\phi}_0 - \frac{1}{n}\mathbf{K}\mathbf{1}_n)\end{aligned}$$

2 Linear Classification

2.1 Logistic Regression

Given a training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_n, y_n)\} \subseteq \mathbb{R}^p \times \{0, 1\}$, $\mathbf{x}_i$ is the information of the i^{th} training input and y_i is the correspondent label. For now, we will focus on the binary classification problem in which y can take on only two values, 0 and 1.

Definition 2.1. For any training sample $(\mathbf{x}, y)$, it satisfy Bernoulli Distribution

$$\begin{aligned}p(y = 1|\mathbf{x}) &= \mu(\mathbf{x}) \\ \mu(\mathbf{x}) &= \frac{1}{1 + e^{-\eta(\mathbf{x})}} \\ \eta(\mathbf{x}) &= \mathbf{x}^T\boldsymbol{\theta} + b\end{aligned}$$

It is easy to verify that $\mu(\mathbf{x}) > \frac{1}{2} \leftrightarrow x > 0$ and $p(y = 1|\mathbf{x}) = \mu(\mathbf{x})$ can be written more compactly as $p(y|\mathbf{x}) = \mu(\mathbf{x})^y(1 - \mu(\mathbf{x}))^{(1-y)}$. Besides, we can consider $\boldsymbol{\theta}$ and b together as a whole parameter.

2.2 MLE of Logistic Regression

Following the definition, we can obtain formula of the likelihood estimation

$$L = p(y_1, ..., y_n|\mathbf{x}_1, ..., \mathbf{x}_n, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^{n} \mu(\mathbf{x}_i)^{y_i}(1 - \mu(\mathbf{x}_i))^{(1-y_i)}$$

and the log likelihood estimation

$$l(\boldsymbol{\theta}) = -log\ p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = -\sum_{i=1}^{n}\ y_i \log\ \mu(\mathbf{x}_i) + (1 - y_i)\ log\ (1 - \mu(\mathbf{x}_i))$$

Our goal is to maximize the log likelihood

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{n}\ y_i\ log\ \mu(\mathbf{x}_i) + (1 - y_i)\ log\ (1 - \mu(\mathbf{x}_i))$$

Before taking derivative w.r.t $\boldsymbol{\theta}$, we first need to analyse the sigmoid function

$$\begin{aligned}
\mu(\mathbf{x}) &= g(\eta(\mathbf{x})) \\
g(x) &= \frac{1}{1+e^{-x}} \\
g'(x) &= -\frac{-e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}}(1-\frac{1}{1+e^{-x}}) \\
&= g(x)(1-g(x))
\end{aligned}$$

Taking derivative w.r.t $\boldsymbol{\theta}$, (we abbreviate $\mu(\mathbf{x}_i)$ as μ_i and $\eta(\mathbf{x}_i)$ as η_i)

$$\begin{aligned}
\frac{\partial l}{\partial \boldsymbol{\theta}} &= -\sum_{i=1}^{n}(y_i\frac{1}{\mu_i} - (1-y_i)\frac{1}{1-\mu_i})\frac{\partial \mu_i}{\partial \boldsymbol{\theta}} \\
&= -\sum_{i=1}^{n}(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i})\frac{\partial g(\eta_i)}{\partial \boldsymbol{\theta}} \\
&= -\sum_{i=1}^{n}(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i})\frac{\partial g(\eta_i)}{\partial \eta_i}\frac{\partial \eta_i}{\partial \boldsymbol{\theta}} \\
&= -\sum_{i=1}^{n}(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i})g(\eta_i)(1-g(\eta_i))\frac{\partial \eta_i}{\partial \boldsymbol{\theta}} \\
&= -\sum_{i=1}^{n}(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i})\mu_i(1-\mu_i)\frac{\partial(\mathbf{x}_i^T\boldsymbol{\theta}+b)}{\partial \boldsymbol{\theta}} \\
&= -\sum_{i=1}^{n}(\frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i})\mu_i(1-\mu_i)\mathbf{x}_i \\
&= \sum_{i=1}^{n}(\mu_i - y_i)\mathbf{x}_i
\end{aligned}$$

This therefore gives us the stochastic gradient descent rule

$$\boldsymbol{\theta}_j^{t+1} = \boldsymbol{\theta}_j^t - \rho(\mu_i - y_i)\mathbf{x}_i$$

ρ is a parameter in [0..1].stochastic gradient descent is an online learning algorithm and can start making progress right away, and continues to make progress with each example it looks at. Often, stochastic gradient descent gets $\boldsymbol{\theta}$ close to the minimum much faster than batch gradient descent. (Note however that it may never converge to the minimum, and the parameters will keep oscillating around the minimum of $l(\boldsymbol{\theta})$ but in practice most of the values near the minimum will be reasonably good approximations to the true minimum.)

2.3 Newton-Raphson method

Taking quadratic term of a local Taylor expansion of $l(\boldsymbol{\theta})$

$$l(\boldsymbol{\theta}) \approx l(\boldsymbol{\theta}^{(t)}) + \nabla_{\boldsymbol{\theta}^{(t)}} l(\boldsymbol{\theta}^{(t)})(\boldsymbol{\theta}-\boldsymbol{\theta}^{(t)}) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^{(t)})^T\mathbf{H}(\boldsymbol{\theta}^{(t)})(\boldsymbol{\theta}-\boldsymbol{\theta}^{(t)})$$

In the above equation, $\mathbf{H}(\boldsymbol{\theta}^{(t)}) = \frac{\partial^2 l}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}$ is called $\boldsymbol{Hessian}$.
When $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}^{(t)}} l(\boldsymbol{\theta}^{(t)}) + \mathbf{H}(\boldsymbol{\theta}^{(t)})(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}) = 0$, which means $l(\boldsymbol{\theta})$ reach its maximum, it follows

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta} = \boldsymbol{\theta}^{(t)} - \mathbf{H}^{-1} \nabla_{\boldsymbol{\theta}^{(t)}} l(\boldsymbol{\theta}^{(t)})$$

Now, we analyse the $\boldsymbol{Hessian\ Matrix}$ $\mathbf{H}(\boldsymbol{\theta})$

$$\begin{aligned}
\frac{\partial^2 l}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} &= \frac{\partial}{\partial \boldsymbol{\theta}} \frac{\partial l}{\partial \boldsymbol{\theta}^{\boldsymbol{T}}} \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} (\sum_{i=1}^{n} (\mu_i - y_i) \mathbf{x}_i^T) \\
&= \sum_{i=1}^{n} \frac{\partial \mu_i}{\partial \boldsymbol{\theta}} \mathbf{x}_i^T \\
&= \sum_{i=1}^{n} \mu_i (1 - \mu_i) \mathbf{x}_i \mathbf{x}_i^T \\
&= \mathbf{X}^T \mathbf{diag}(\mu_1(1-\mu_1), \mu_2(1-\mu_2), ..., \mu_n(1-\mu_n)) \mathbf{X} \\
&= \mathbf{X}^T W \mathbf{X}
\end{aligned}$$

Applying $\nabla_{\boldsymbol{\theta}^{(t)}} l = \mathbf{X}^T(\mu - \mathbf{y})$, we have

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - \mathbf{H}^{-1} \nabla_{\boldsymbol{\theta}^{(t)}} l \\
&= \boldsymbol{\theta}^{(t)} - (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mu - \mathbf{y}) \\
&= \boldsymbol{\theta}^{(t)} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mu)
\end{aligned}$$

Since $\mu_i(1 - \mu_i) \leq \frac{1}{4}$, we have

$$\begin{aligned}
\mathbf{W} &= \mathbf{diag}(\mu_1(1-\mu_1), \mu_2(1-\mu_2), ..., \mu_n(1-\mu_n)) \preceq \frac{1}{4} \mathbf{I}_n \\
\mathbf{H} &= \mathbf{X}^T \mathbf{W} \mathbf{X} \preceq \mathbf{X}^T \mathbf{X}
\end{aligned}$$

and deduce a new iteration rule

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + 4(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mu)$$

2.4 Softmax Regression

Consider a classification problem in which the response variable y can take on any one of k values, so $\mathbf{y} \in \{0, 1\}^c$. For example, rather than classifying email into the two classes spam or not-spamwhich would have been a binary classification problem we might want to classify it into three classes, such as spam, personal mail, and work-related mail. The response variable is still discrete, but can now take on more than two values. We will thus model it as distributed according to a multinomial distribution.

Now, we define the training data as $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$ and for each sample $(\mathbf{x}, \mathbf{y})$, we have $\mathbf{y} \in \{0, 1\}^c$ and $\sum_j \mathbf{y}_j = 1$. Thus,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{c} \mu_j^{(\mathbf{y}_j)} \quad s.t. \sum_{j=1}^{c} \mu_j = 1$$

$$p(\mathbf{y}_j = 1|\mathbf{x}, \boldsymbol{\theta}) = \mu_j = \frac{e^{\boldsymbol{\theta}_j^T \mathbf{x}}}{\sum_{k=1}^{c} e^{\boldsymbol{\theta}_k^T \mathbf{x}}}$$

This model, which applies to classification problems where $\mathbf{y} \in \{0, 1\}^c$, is called **softmax regression**. It is a generalization of logistic regression.
Lastly, let's discuss parameter fitting. Similar to our original derivation of ordinary least squares and logistic regression, if we have a training set of n examples $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$ and would like to learn the parameters $\boldsymbol{\theta}$ of this model, we would begin by writing down the log-likelihood

$$\begin{aligned} l(\boldsymbol{\theta}) &= -log \prod_{i=1}^{n} \prod_{j=1}^{c} \mu_{ij}^{\mathbf{y}_{ij}} \\ &= -\sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{y}_{ij} log(\mu_{ij}) \end{aligned}$$

Attention that for each kind of y, there exist a parameter $\boldsymbol{\theta_j}$ and a correspondent μ_j and in order to normalize them, we set $\sum_{j=1}^{c} \boldsymbol{\theta}_j = \mathbf{0}$ and take the Lagrangian

$$\mathcal{L}(\mu, \boldsymbol{\theta}) = -\sum_{i=1}^{n} \sum_{j=1}^{c} \mathbf{y}_{ij} log(\mu_{ij}) + \lambda \sum_{j=1}^{c} \boldsymbol{\theta}_j$$

Taking derivative w.r.t $\boldsymbol{\theta_j}$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta_j}} &= \sum_{i=1}^{n} (\mu_{ij} - \mathbf{y}_{ij}) \mathbf{x}_i + \lambda \\ \sum_{j=1}^{c} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta_j}} &= \sum_{j=1}^{c} \sum_{i=1}^{n} (\mu_{ij} - \mathbf{y}_{ij}) \mathbf{x}_i + c\lambda \\ &= \sum_{i=1}^{n} \mathbf{x}_i \sum_{j=1}^{c} (\mu_{ij} - \mathbf{y}_{ij}) + c\lambda \\ &= c\lambda \end{aligned}$$

and $\lambda = 0$ would follow naturally which means the Lagrangian with or without the limit of $\sum_{j=1}^{c} \boldsymbol{\theta}_j = \mathbf{0}$ have the same result.

Note: if you find any error or mistake above, contact Kainan to point it out, for the rest part, go for Tianfan.

3 Generative Model and Discriminant Model

There are two main types of model used in Machine Learning. One is generative model and the other is discriminant model. Generally speaking, in generative model, the joint probability $P(X, Y)$ of model $\mathbf{X}$ and data sample $\mathbf{Y}$ are estimated. In contrast, in discriminant model, the conditional probability $P(Y|X)$ is estimated.

Definition 3.1. Generative Model is a model for randomly generating observable data. It specifies a joint probability distribution over observation and label sequences.

Definition 3.2. Discriminative models are a class of models for modeling the dependence of an unobserved variable y on an observed variable x. Within a probabilistic framework, this is done by modeling the conditional probability distribution $P(y|x)$, which can be used for predicting y from x.

3.1 Generative Model

An example will be given in the following to describe the generative model.
x is p-dimensional vector, which belongs to Gaussian component y at the distribution of $p(y) = \pi^y(1-\pi)^{1-y}$ Where $y \in \{0, 1\}$, 0,1 stands for the Gaussian component respectively. The joint probability can be expressed as

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y)$$

Where Y satisfys the Bernoulli Distribution

$$p(y) = \pi^y(1-\pi)^{1-y}$$

and satisfys $\pi \in (0, 1)$

The conditional probability given Y is as following:

$$p(x|y=1) = \frac{exp(-\frac{1}{2}(x-\mu_1)^T\Sigma(x-\mu_1))}{|2\pi\Sigma|^{\frac{p}{2}}}$$
$$p(x|y=0) = \frac{exp(-\frac{1}{2}(x-\mu_0)^T\Sigma(x-\mu_0))}{|2\pi\Sigma|^{\frac{p}{2}}}$$

The model's parameters θ contains the following components: $\theta = (\pi, \mu_1, \mu_0, \Sigma)$ Then Likelihood Function are represent as $\prod_{i=1}^{n} p(x_i, y_i|\theta)$ and we need use the MLE(Maximum Likelihood Estimation) to optimize it.

The conditional probability can be expressed and simplified as:

$$
\begin{aligned}
& p(y=1|x,\theta) \\
&= \frac{p(x|y=1,\theta)p(y=1)}{p(x|y=1,\theta)p(y=1)+p(x|y=0,\theta)p(y=0)} \\
&= \frac{exp(\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))}{exp(\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1))\pi + exp(\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0))(1-\pi)} \\
&= \frac{1}{1+\frac{\pi}{1-\pi}exp(-\frac{1}{2}[(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)-(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)])} \\
&= \frac{1}{1+\frac{\pi}{1-\pi}exp(-\frac{1}{2}[(\mu_1-\mu_0)^T\Sigma^{-1}x+\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)])} \\
&= \frac{1}{1+\frac{\pi}{1-\pi}exp(-\frac{1}{2}[(\mu_1-\mu_0)^T\Sigma^{-1}x+\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1-\mu_0)]+log\ \frac{\pi}{1-\pi})} \\
&= \frac{1}{1+exp(-\theta^Tx-b)} \\
&= \frac{1}{1+exp(-\eta(x))}
\end{aligned}
$$

where

$$
\begin{aligned}
\theta &= (\mu_1-\mu_0)^T\Sigma^{-1} \\
b &= -\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^T(\mu_0-\mu_1)-\log\ \frac{1-\pi}{\pi}
\end{aligned}
$$

The Log-Likelihood about θ can be expressed as

$$
L(\theta|D) = \log\ (\prod_{i=1}^{n} p(x_i|y_i\theta)) + log(\prod_{i=1}^{n} p(y_i|\pi))
$$

Then unfold the first term:

$$
\begin{aligned}
& p(x_i|y_i,\theta) \\
&= \Sigma_{i=1}^{n}[\log\ p(y_i|\pi) + \log\ p(x_i|y_i,\theta)] \\
&= \Sigma_{i=1}^{n} \log\ p(y_i|\pi) + \Sigma_{i=1}^{n} \log\ p(x_i|y_i,\theta)
\end{aligned}
$$

Estimating π is irrelavent to the second terms. So we just need to get the partial derviation to π and obtain that

$$
\Sigma_{i=1}^{n}[y_i \log\ \pi + (1-y_i)\log\ (1-\pi)] = 0
$$

And get the solution.

$$
\widehat{\pi} = \frac{\Sigma_{i=1}^{n} y_i}{n}
$$

Since we get the representation of π, we estimate other parameters.

$$
\begin{aligned}
&\Sigma_{i=1}^{n} \log\ p(x_i|y_i,\theta) \\
&= \Sigma_{i=1}^{n} \log \prod_{j=1}^{p} p(x_{ij}|y_i,\theta) \\
&= \Sigma_{i=1}^{n}\Sigma_{j=1}^{p} \log\ p(x_{ij}|y_i,\theta) \\
&= \Sigma_{i=1}^{n}\Sigma_{j=1}^{p} \log\ p(x_{ij}|y_i,\mu_{ij},\gamma_j)^{y_i} p(x_{ij}|y_i=0,\mu_{ij},\sigma_j)^{1-y_i} \\
&= \Sigma_{i=1}^{n}\Sigma_{j=1}^{p} y_i \log\ p(x_{ij}|y_i=1,\mu_{ij},\sigma_j) + (1-y_i)\log\ p(x_{ij}|y_i=0,\mu_{0j},\sigma_j)
\end{aligned}
$$

Since covariance of the Gaussian Component is same in this example, we only need to estimate two mean vector and one covariance matrix. $(\mu_0, \mu_1, \sigma_1^2, ..., \sigma_p^2)$, where $\mu_0 = (\mu_{01}, \mu_{02}...\mu_{0p})^T$ and $\mu_{\mathbf{1}} = (\mu_{\mathbf{11}}, \mu_{\mathbf{12}}...\mu_{\mathbf{1p}})^{\mathbf{T}}$.

Since in p-dimensional vector, there is no relationship between different dimensions. So we redefine the optimization problem as follow.

$$L(\theta) = \Sigma_{j=1}^{p}\Sigma_{i=1}^{n}(p(x_{ij}|y_i,\theta))$$

$$L(\theta) = \Sigma_{j=1}^{p}\Sigma_{i=1}^{n}(p(x_{ij}|y_i,\theta))$$

The j-th component can be represent as

$$L_j = \Sigma_{i=1}^{n} y_i \log\ p(x_{ij}|y_i=1,\mu_{1j},\sigma_j^2) + (1-y_i)\log\ p(x_{ij}|y_i=0,\mu_{0j},\Sigma_j^2)$$

To obtain the optimal, let the partial derivation equal to 0, get the following three equation.

$$
\begin{aligned}
\frac{\partial l_i}{\partial \mu_{1j}} &= 0 \\
\frac{\partial l_i}{\partial \mu_{0j}} &= 0 \\
\frac{\partial l_i}{\partial \sigma_j^2} &= 0
\end{aligned}
$$

Then unfold the L_j

$$
\begin{aligned}
&\log\ p(x_{ij}|y_i=1,\mu_{1j},\sigma_j^2) \\
&= \log\ \frac{1}{|2\pi|^{\frac{1}{2}}\sigma_j} exp(-\frac{(x_{ij}-\mu_{ij})^2}{2\sigma_j^2}) \\
&= -\frac{(x_{ij}-\mu_{ij})^2}{2\sigma_j^2} - \log\ \sigma_j
\end{aligned}
$$

Similarly,

$$\begin{aligned}
&\log\ p(x_{ij}|y_i=0,\mu_{0j},\sigma_j^2)\\
&=\log\ \frac{1}{|2\pi|^{\frac{1}{2}}\sigma_j}exp(-\frac{(x_{ij}-\mu_{ij})^2}{2\sigma_{(x_{ij}-\mu_{0j})^2}2\sigma_j^2})\\
&=-\frac{(x_{ij}-\mu 0j)^2}{2\sigma_j^2}-\log\ \sigma_j
\end{aligned}$$

$$l_j=-[\Sigma_{i=1}^n y_i\frac{(x_{ij}-\mu_{1j})^2}{2\sigma_j^2}+\frac{1}{2}y_i\log\ \sigma_j+(1-y_i)\frac{(x_{ij}-\mu_{0j})^2}{2\sigma_j^2}+\frac{1}{2}(1-y_i)\log\ \sigma_j]$$

$$\frac{\partial l_j}{\partial\mu_{0j}}=-\Sigma_{i=1}^n(1-y_i)\frac{(\mu_{0j}-x_{ij})}{\sigma_j^2}$$

Then firstly get the estimation of mean component and then process the variance.

$$\begin{aligned}
\widehat{\mu_{0j}}&=\frac{\Sigma_{i=1}^n(1-y_i)x_{ij}}{\Sigma_{i=1}^n(1-y_i)}\\
\widehat{\mu_{1j}}&=\frac{\Sigma_{i=1}^n y_i x_{ij}}{\Sigma_{i=1}^n y_i}
\end{aligned}$$

To estimate the variance, we need the estimated mean value.

$$\widehat{\sigma_j^2}=\frac{\Sigma_{i=1}^n[y_i(x_{ij}-\widehat{\mu_{ij}})^2+(1-y_i)(x_{ij}-\mu_{0j})]}{n}$$