Assignment1

1. Machine learning problem

(a)

1. BF
2. C
3. C
4. BG
5. AE
6. AD
7. BF
8. AE
9. BF

(b)

Answer: **This is definite False.** If you just choose parameters which performs best on whole dataset, it may cause the **overfitting problem or high variance**, that is to say your model's generalization ability is low. So it's better to split the dataset into **training set、cross-validation set and test set**, we train our model parameters from training set that performs best on cross-validation set, meanwhile use the test set to examine the generalization ability of your model.

---

2. Bayes Decision Rule

(a)

- (i) Since we just have 3 boxes, so:

$$P(B_1 = 1) = \frac{1}{3}$$

- (ii) Since we just have 1 box containes the bonus, so:

$$P(B_2 = 0 \mid B_1 = 1) = 1$$

- (iii) From Bayes formula, we can get that:

$$P(B_1 = 1 \mid B_2 = 0) = \frac{P(B_2 = 0 \mid B_1 = 1) \cdot P(B_1 = 1)}{P(B_2 = 0)} = \frac{1 * (\frac{1}{3})}{(\frac{2}{3})} = \frac{1}{2}$$

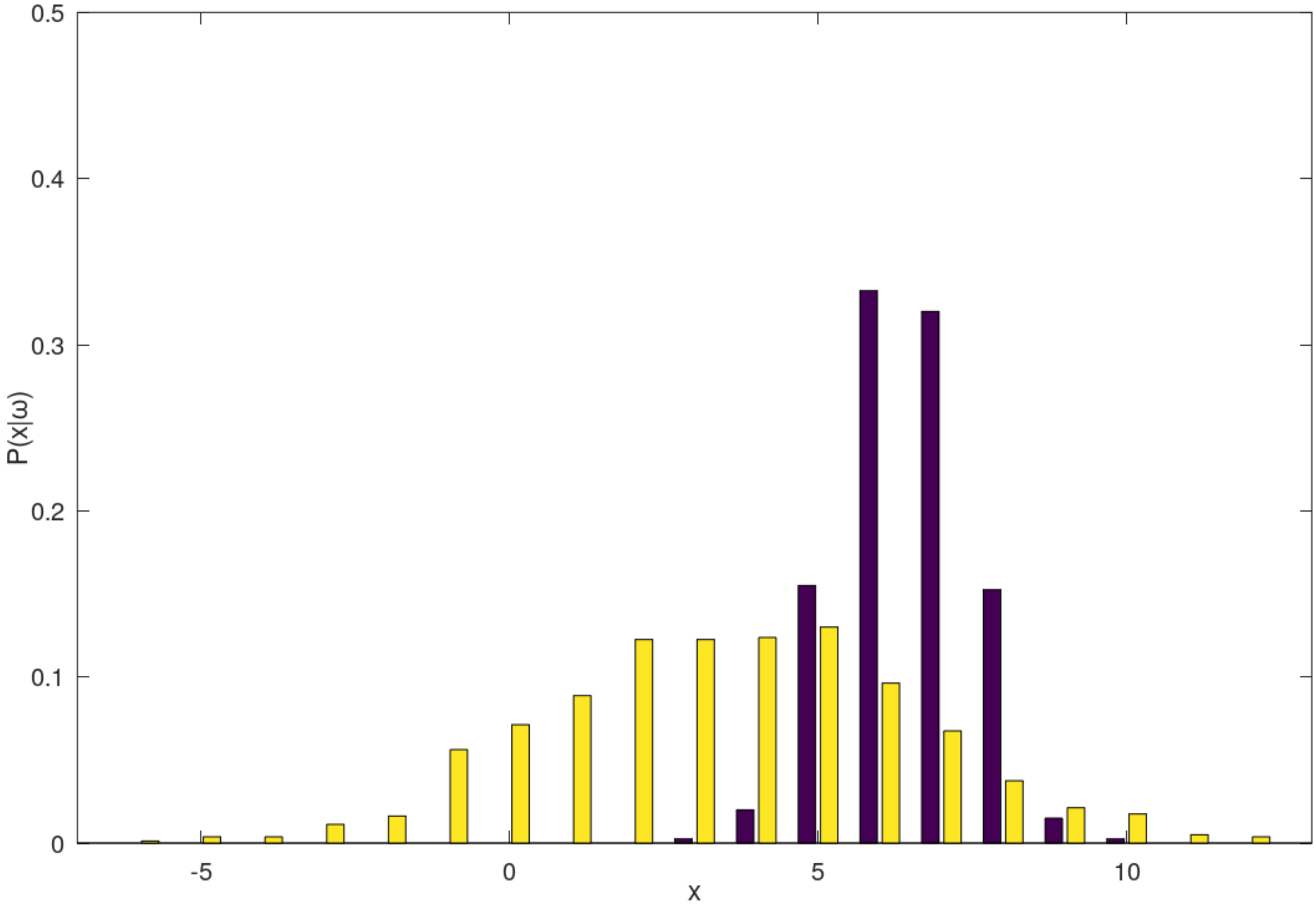
- (iv) Since the B_1 and B_3 has no direct association, so that can get:

$$P(B_1 = 1 \mid B_2 = 0) = P(B_3 = 1 \mid B_2 = 0) = \frac{1}{2}$$

  That's to say, the probability of B_1 or B_3 that contains the bonus is just the same! So you can either choose stick to B_1 or choose the left box B_3, both are optimal choice.

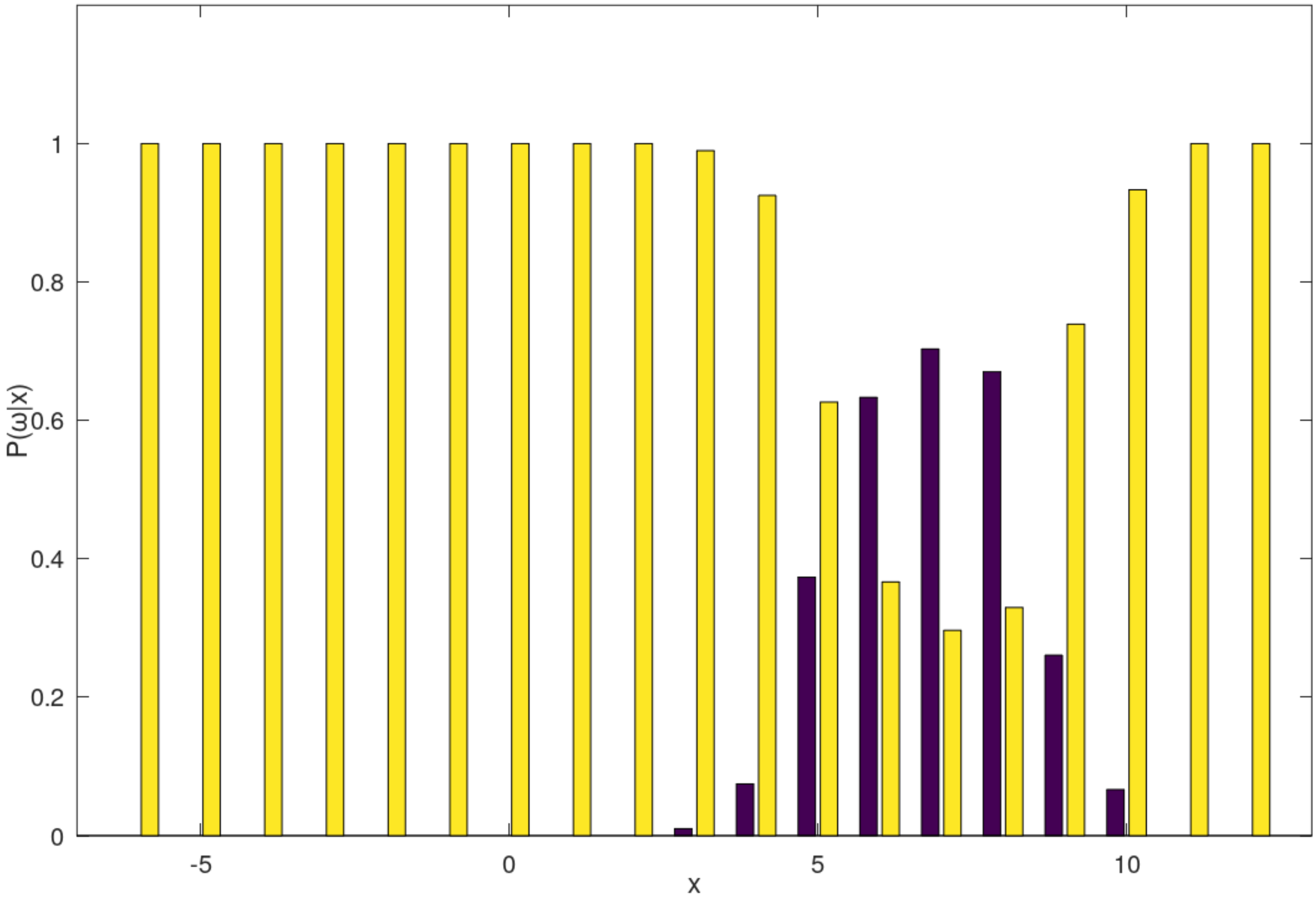
(b)

- (i) The distribution of $P(x \mid \omega_i)$ is shown below:



```
Using the maximum likelihood decision rule,the number of
misclassfied x also test error is 64.
```

- (ii) The distribution of $P(\omega_i \mid x)$ is shown below:



```
    Using optimal bayes decision rule, the number of
    misclassfied x also the test error is 47.
```

- (iii) The minimum total risk $(R = \sum_x min_i R(\alpha_i \mid x) = \sum_x min_i \lambda(\alpha_i \mid \omega_j) \cdot p(\omega_j \mid x))$, here we know the loss matrix $\lambda(\alpha_i \mid \omega_j)$, and I take the training set posterior into the formula, I calculate the result is **2.444354**, the codes shown below, also can see in the code folder:

```
% p is the posterior
risk = [0, 1; 2, 0];
matrix_omega_1 = sum(p .* risk(1, :)');
matrix_omega_2 = sum(p .* risk(2, :)');
matrix_min = min(matrix_omega_1, matrix_omega_2);
risk_min = sum(matrix_min);
fprintf("The minimum risk is %f\n", risk_min);
```

3. Gaussian Discriminant Analysis and MLE

- **(a)**

First from Beyaes formula, we can get that:

$$p(y=1 \mid \mathbf{x})=\frac{p(\mathbf{x} \mid y=1) \cdot p(y=1)}{p(\mathbf{x})}=\frac{p(\mathbf{x} \mid y=1) \cdot p(y=1)}{\sum_{i=0}^{1} p(\mathbf{x} \mid y=i) \cdot p(y=i)} \tag{1}$$

Since $p(y=0)=p(y=1)=\phi=\frac{1}{2}$, so from (1) we can produce:

$$p(y=1 \mid \mathbf{x})=\frac{1}{1+\frac{p(\mathbf{x} \mid y=0) \cdot p(y=0)}{p(\mathbf{x} \mid y=1) \cdot p(y=1)}}=\frac{1}{1+\frac{p(\mathbf{x} \mid y=0)}{p(\mathbf{x} \mid y=1)}} \tag{2}$$

Until now, our target is to calculate $\frac{p(\mathbf{x} \mid y=0)}{p(\mathbf{x} \mid y=1)}$, we start from calculate $p(\mathbf{x} \mid y=0)$:

$$p(\mathbf{x} \mid y=0)=\frac{1}{2 \pi \sqrt{\left|\Sigma_{0}\right|}} \cdot e^{-\frac{1}{2}\left(\mathbf{x}-\mu_{0}\right)^{T} \Sigma_{0}^{-1}\left(\mathbf{x}-\mu_{0}\right)} \tag{3}$$

we already know that $\Sigma_{0}=\begin{pmatrix}1 & 0 \\ 0 & 1\end{pmatrix}$ and $\mu_{0}=(0,0)^{T}$,so easy to know $\left|\Sigma_{0}\right|=1$ and $\Sigma_{0}^{-1}=\begin{pmatrix}1 & 0 \\ 0 & 1\end{pmatrix}$, take these into (3):

$$p(\mathbf{x} \mid y=0)=\frac{1}{2 \pi} e^{-\frac{1}{2}\left[\begin{array}{ll}x_{1} & x_{2}\end{array}\right] \cdot\left[\begin{array}{ll}1 & 0 \\ 0 & 1\end{array}\right] \cdot\left[\begin{array}{l}x_{1} \\ x_{2}\end{array}\right]}=\frac{1}{2 \pi} e^{-\frac{1}{2}\left(x_{1}^{2}+x_{2}^{2}\right)} \tag{4}$$

The same way for $y=1$, we can get formula below:

$$p(\mathbf{x} \mid y=1)=\frac{1}{2 \pi} e^{-\frac{1}{2}\left[\left(x_{1}-1\right)^{2}+\left(x_{2}-1\right)^{2}\right]} \tag{5}$$

Take $(4)(5)$ back to (2), we can get final result:

$$p\left(y=1 \mid \mathbf{x} ; \phi, \mu_{0}, \mu_{1}, \Sigma_{0}, \Sigma_{1}\right)=\frac{1}{1+\frac{p(\mathbf{x} \mid y=0)}{p(\mathbf{x} \mid y=1)}}=\frac{1}{1+e^{1-x_{1}-x_{2}}} \tag{6}$$

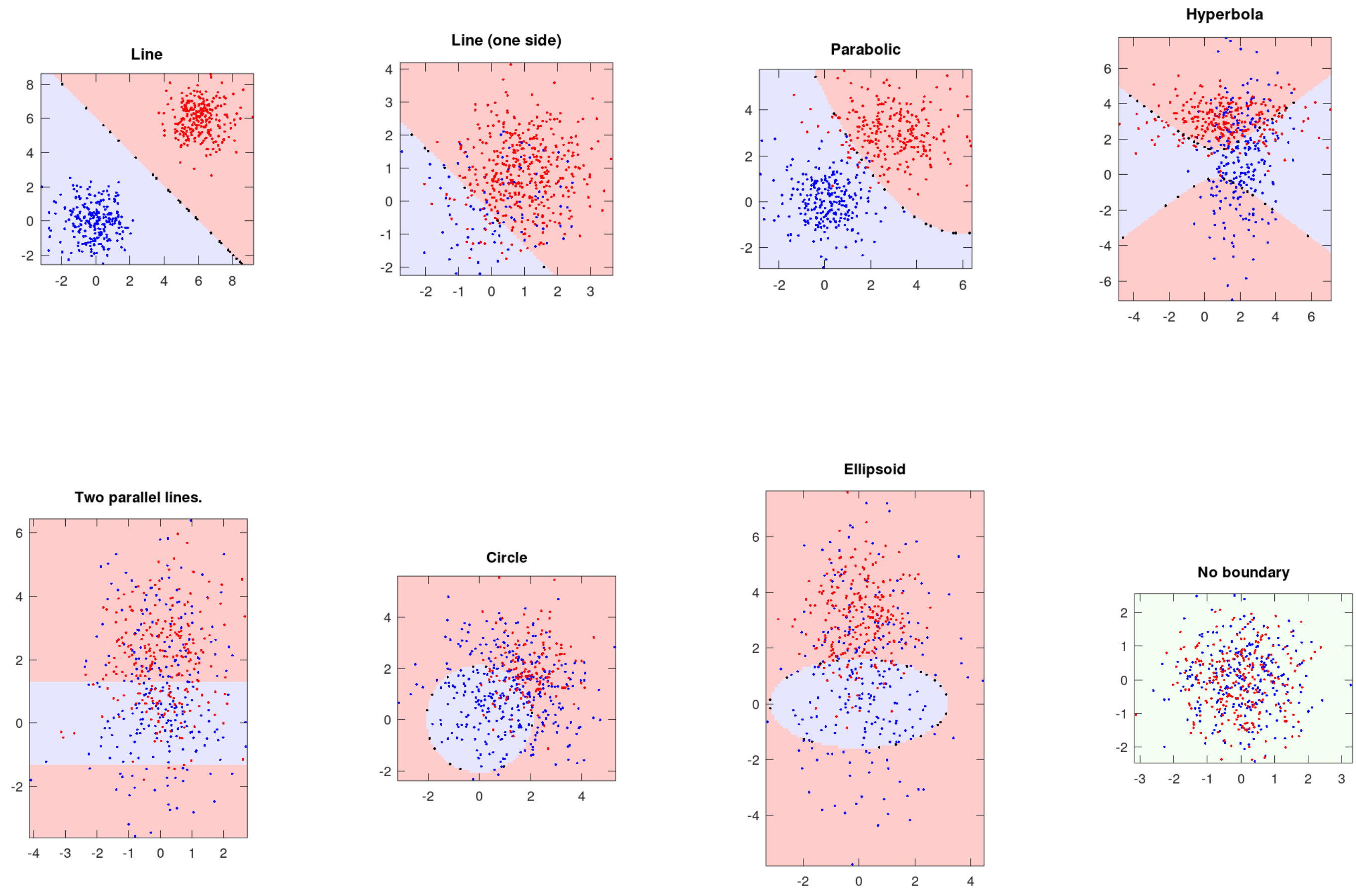
So far, we continue to go on to get the **decision boundary:** using the same way calculate that $P(y=0 \mid \mathbf{x})=\frac{1}{1+e^{x_{1}+x_{2}-1}}$, we let $p(y=0 \mid \mathbf{x})=p(y=1 \mid \mathbf{x})$, then we get the **decision boundary also discrimant plane**: $x_{1}+x_{2}=1$. When $x_{1}+x_{2}<1 \Longrightarrow p(y=0 \mid \mathbf{x})>p(y=1 \mid \mathbf{x})$. When $x_{1}+x_{2}>1 \Longrightarrow p(y=1 \mid \mathbf{x})>p(y=0 \mid \mathbf{x})$.

- **(b):** $Posterior=p\left(y_{i} \mid \mathbf{x}\right)=\frac{p\left(\mathbf{x} \mid y_{i}\right) p\left(y_{i}\right)}{p(\mathbf{x})}$, in this task already know the likelihood formula form and $prior=p\left(y_{i}\right)=\phi_{i}, p(\mathbf{x})=\sum_{i=1}^{m} p\left(\mathbf{x} \mid y_{i}\right) p\left(y_{i}\right)$, so according to the information we know above, the implementation below, you can also find these in code folder and some parameter description:

```
%   Inputs:
%       'X'     - M-by-N matrix, N data points of dimension M.
%       'Mu'    - M-by-K matrix, mean of K Gaussian distributions.
%       'Sigma' - M-by-M-by-K matrix (yes, a 3D matrix), variance matrix of
%                   K Gaussian distributions.
%       'Phi'   - 1-by-K matrix, prior of K Gaussian distributions.
%
%   Outputs:
%       'p'     - N-by-K matrix, posterior probability of N data points
%                   with in K Gaussian distributions.

for i=1:K
    phi_i = Phi(i);
    Sigma_i = Sigma(:, :, i);
    mu_i = Mu(:, i);
    deteminant = det(Sigma_i);
    Sigma_i_inv = pinv(Sigma_i);
    post_i = phi_i / (2 * pi * sqrt(deteminant)) * ...
      exp(-0.5 * sum((X - mu_i)' * Sigma_i .* (X - mu_i)', 2));
    P(:, i) = post_i;
end
evidence_matrix = sum(P, 2);
P = P ./ evidence_matrix;
```

- **(c):** The result plots shown below:



- **(d):**
  Here I start from K-class gaussian model to calculate the MLE of all parameters. The procedure is below:
  Since we have K-class, so we can divide into K-class dataset, denote it as $D_1, D_2, \ldots, D_K$, we take the k_{th} dataset to calculate the MLE of parameters, the others is just the same.

  Suppose the dataset D_k have m iid samples, and denote θ to the params $\theta = [\mu, \Sigma]$, then the likelihood is:

$$p(D_k \mid \theta) = \prod_{i=1}^{m} p(x_i \mid \theta) \tag{1}$$

  we change it to log-likelihood for calculation convenience:

$$\ln p(D_k \mid \theta) = \sum_{i=1}^{m} \ln p(x_i \mid \theta) \tag{2}$$

  Since $p(x_i \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}[(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)]}$, take into (2),

$$l(\theta) = \ln p(D_k \mid \theta) = \sum_{i=1}^{m} -\frac{1}{2}[d \ln 2\pi + \ln|\Sigma| + (x_i-\mu)^T\Sigma^{-1}(x_i-\mu)] \tag{3}$$

  Our target is to maximum this function value, we denote gradient operator $\nabla_\theta = [\frac{\partial}{\partial\theta_1}, \ldots, \frac{\partial}{\partial\theta_n}]$, so that the neccessary conditions to the optimal is $\nabla_\theta l(\theta) = 0$. First we gonna to calculate the MLE for μ_k.

$$\begin{aligned}
\nabla_\mu \ln p(x_i \mid \theta) &= -\frac{1}{2}\frac{\partial[d\ln 2\pi + \ln|\Sigma| + (x_i-\mu)^T\Sigma^{-1}(x_i-\mu)]}{\partial\mu} \\
&= -\frac{1}{2}\frac{\partial[(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)]}{\partial\mu} \\
&= -\frac{1}{2}\frac{\partial(x_i^T\Sigma^{-1}x_i - x_i^T\Sigma^{-1}\mu - \mu^T\Sigma^{-1}x_i + \mu^T\Sigma^{-1}\mu)}{\partial\mu} \\
&= -\frac{1}{2}[0 - (x_i^T\Sigma^{-1})^T - \Sigma^{-1}x_i + (\Sigma^{-1} + (\Sigma^{-1})^T)\mu] \\
&= -\frac{1}{2}(\Sigma^{-1} + (\Sigma^{-1})^T)(\mu - x_i)
\end{aligned}$$

$$
\begin{aligned}
\nabla_{\mu} l(\theta) &= \nabla_{\mu} \sum_{i=1}^{m} \ln p(x_i \mid \theta) \\
&= \sum_{i=1}^{m} -\frac{1}{2}(\Sigma^{-1} + (\Sigma^{-1})^T)(\mu - x_i) = 0 \implies \hat{\mu_k} = \frac{1}{m}\sum_{i=1}^{m} x_i
\end{aligned}
$$

Next step we continue to the MLE for Σ_k, the same way as previous, and I denote $x_i - \mu = A$ and $\Sigma^{-T} = (\Sigma^{-1})^T = (\Sigma^T)^{-1}$:

$$
\begin{aligned}
\nabla_{\Sigma} \ln p(x_i \mid \theta) &= -\frac{1}{2}\frac{\partial[d \ln 2\pi + \ln|\Sigma| + (x_i - \mu)^T \Sigma^{-1}(x_i - \mu)]}{\partial \Sigma} \\
&= -\frac{1}{2}\frac{\partial(\ln|\Sigma| + A^T \Sigma^{-1} A)}{\partial \Sigma} \\
&= -\frac{1}{2}(\Sigma^{-T} - \Sigma^{-T} A A^T \Sigma^{-T}) \\
&= -\frac{1}{2}\Sigma^{-T}(\mathrm{I} - A A^T \Sigma^{-T})
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\Sigma} l(\theta) &= \nabla_{\Sigma} \sum_{i=1}^{m} \ln p(x_i \mid \theta) \\
&= -\frac{1}{2}\sum_{i=1}^{m}[\Sigma^{-T}(\mathrm{I} - A A^T \Sigma^{-T})] \\
&= -\frac{1}{2}\Sigma^{-T}[m\mathrm{I} - \sum_{i=1}^{m}(A A^T \Sigma^{-T})] = 0 \implies m\mathrm{I} = \sum_{i=1}^{m}(A A^T \Sigma^{-T}) \\
&\implies \mathrm{I} = \frac{1}{m}(\sum_{i=1}^{m} A A^T)\Sigma^{-T} \\
&\implies \frac{1}{m}(\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T) = (\Sigma^{-T})^{-1} = \Sigma^T \\
&\implies \Sigma = \frac{1}{m}(\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T)^T \\
&\implies \hat{\Sigma_k} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu_k})(x_i - \hat{\mu_k})^T
\end{aligned}
$$

For the last MLE of ϕ_k, from the law of large numbers, we can simply get that $\hat{\phi_k} = \frac{N_k}{N}$, where N_k is number of samples in class k, and N is total number of samples.

So far, the MLE of all parameters for class k is shown below:

$$\begin{cases} \hat{\mu_k} = \frac{1}{m}\sum_{i=1}^{m} x_i \\ \hat{\Sigma_k} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu_k})(x_i - \hat{\mu_k})^T \\ \hat{\phi_k} = \frac{N_k}{N} \end{cases}$$

4. Text Classification with Naive Bayes

- **(a):**

  1. **nbsp** index:30033
  2. **viagra** index:75526
  3. **pills** index:38176
  4. **cialis** index:45153
  5. **voip** index:9494
  6. **php** index:65398
  7. **meds** index:37568
  8. **computron** index:13613
  9. **sex** index:56930
  10. **ooking** index:9453

- **(b):**

  The accuracy rate of my classifier on test set is **98.573156%.**, you can check this simply by running the `run.m` in the `text_classification` code folder.

- **(c):**

  That is **not correct**. A model with 99% accuracy is not always a perfect model. Just like the description in the hint. Consider in this situation that our spam filter works **when the ratio of spam and ham email is 1:99.** If we get one model that **simply classify all the emails to ham emails**, then our model's accuracy is exactly 99%, but **of course this is not a good model.**

- **(d):**

  According to my learnt model, I get this result: $f_p: 28,\ f_n: 31,\ \ t_p: 1093,\ t_n: 2983,$ and from these data get the $precision = \frac{t_p}{t_p+f_p} = 0.975022$, and the $recall = \frac{t_p}{t_p+f_n} = 0.972420.$

- **(e):**

  In my opinion, in **spam filter** the **recall is more important than the precision**, because we just focus on filter the spam emails, small number of ham emails classify to spam does not so large effect on this, we only need it can filter the spam emails out. But a classifier to **identify drugs and bombs at airport we should more focus on precision**, because in this situation we must be sure that the classify precision is high, we can't let one bomb or drugs to go through the airport right, and save things we just them go. So for this classifier, **precision is more inportant than recall.**