PATTERN RECOGNITION, THEORETICAL EXERCISE 1, ECE, NTUA
KONTOGIANNIS ANDREAS, 03115187

1.1 (a) (b) (c) Gaussian Distribution

For independent random variables $X_1 = X$ and $X_2 = Y$, the distribution f_Z of $Z = X_3 = X + Y$ equals the convolution of X and Y:

$$f_Z(z) = \int_{-\infty}^{\infty} f_Y(z-x) f_X(x)\, dx$$

Given that f_X and f_Y are normal densities:

$$f_X(x) = \mathcal{N}(x|\mu_X, \sigma_X^2) = \frac{1}{\sqrt{2\pi}\sigma_X} e^{-(x-\mu_X)^2/(2\sigma_X^2)}$$

$$f_Y(y) = \mathcal{N}(y|\mu_Y, \sigma_Y^2) = \frac{1}{\sqrt{2\pi}\sigma_Y} e^{-(y-\mu_Y)^2/(2\sigma_Y^2)}$$

Substituting into the convolution:

$$\begin{aligned}
f_Z(z) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left[-\frac{(z-x-\mu_Y)^2}{2\sigma_Y^2}\right] \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right] dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_X\sigma_Y} \exp\left[-\frac{\sigma_X^2(z-x-\mu_Y)^2 + \sigma_Y^2(x-\mu_X)^2}{2\sigma_X^2\sigma_Y^2}\right] dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_X\sigma_Y} \exp\left[-\frac{\sigma_X^2(z^2 + x^2 + \mu_Y^2 - 2xz - 2z\mu_Y + 2x\mu_Y) + \sigma_Y^2(x^2 + \mu_X^2 - 2x\mu_X)}{2\sigma_Y^2\sigma_X^2}\right] dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_X\sigma_Y} \exp\left[-\frac{x^2(\sigma_X^2 + \sigma_Y^2) - 2x(\sigma_X^2(z-\mu_Y) + \sigma_Y^2\mu_X) + \sigma_X^2(z^2 + \mu_Y^2 - 2z\mu_Y) + \sigma_Y^2\mu_X^2}{2\sigma_Y^2\sigma_X^2}\right] dx
\end{aligned}$$

Defining $\sigma_Z = \sqrt{\sigma_X^2 + \sigma_Y^2}$:

$$\begin{aligned}
f_Z(z) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_Z} \frac{1}{\sqrt{2\pi}\frac{\sigma_X\sigma_Y}{\sigma_Z}} \exp\left[-\frac{x^2 - 2x\frac{\sigma_X^2(z-\mu_Y)+\sigma_Y^2\mu_X}{\sigma_Z^2} + \frac{\sigma_X^2(z^2+\mu_Y^2-2z\mu_Y)+\sigma_Y^2\mu_X^2}{\sigma_Z^2}}{2\left(\frac{\sigma_X\sigma_Y}{\sigma_Z}\right)^2}\right] dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_Z} \frac{1}{\sqrt{2\pi}\frac{\sigma_X\sigma_Y}{\sigma_Z}} \exp\left[-\frac{\left(x - \frac{\sigma_X^2(z-\mu_Y)+\sigma_Y^2\mu_X}{\sigma_Z^2}\right)^2 - \left(\frac{\sigma_X^2(z-\mu_Y)+\sigma_Y^2\mu_X}{\sigma_Z^2}\right)^2 + \frac{\sigma_X^2(z-\mu_Y)^2+\sigma_Y^2\mu_X^2}{\sigma_Z^2}}{2\left(\frac{\sigma_X\sigma_Y}{\sigma_Z}\right)^2}\right] dx \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left[-\frac{\sigma_Z^2\left(\sigma_X^2(z-\mu_Y)^2 + \sigma_Y^2\mu_X^2\right) - \left(\sigma_X^2(z-\mu_Y) + \sigma_Y^2\mu_X\right)^2}{2\sigma_Z^2(\sigma_X\sigma_Y)^2}\right] \frac{1}{\sqrt{2\pi}\frac{\sigma_X\sigma_Y}{\sigma_Z}} \exp\left[-\frac{\left(x - \frac{\sigma_X^2(z-\mu_Y)+\sigma_Y^2\mu_X}{\sigma_Z^2}\right)^2}{2\left(\frac{\sigma_X\sigma_Y}{\sigma_Z}\right)^2}\right] dx \\
&= \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left[-\frac{(z-(\mu_X+\mu_Y))^2}{2\sigma_Z^2}\right] \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\frac{\sigma_X\sigma_Y}{\sigma_Z}} \exp\left[-\frac{\left(x - \frac{\sigma_X^2(z-\mu_Y)+\sigma_Y^2\mu_X}{\sigma_Z^2}\right)^2}{2\left(\frac{\sigma_X\sigma_Y}{\sigma_Z}\right)^2}\right] dx
\end{aligned}$$

The expression in the integral is a normal density distribution on X, and so the integral equals 1. Thus:

$$f_Z(z) = \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left[-\frac{(z-(\mu_X+\mu_Y))^2}{2\sigma_Z^2}\right]$$

So, $f_Z(z) = \mathcal{N}(z|\mu_Z, \sigma_Z^2) = \mathcal{N}(z|\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$

1.1 (d) Multivariate Gaussian Distribution

Fact: The sum of independent Gaussian random variables is Gaussian.
Proof
Given independent multivariate Gaussian random variables $\mathbf{Y}, \mathbf{Z}$.
We define

$$\mathbf{X} = \mathbf{Y} + \mathbf{Z}$$

Then, it can be written as a linear combination:

$$\mathbf{L} \cdot \mathbf{X} = \sum_{i=1}^{N} L_i X_i = \sum_{i=1}^{N} (L_i Y_i + L_i Z_i) = \sum_{i=1}^{N} L_i Y_i + \sum_{i=1}^{N} L_i Z_i = A + B$$

Where, $A = N(x|m_a, \sigma_a^2)$ and $B = N(x|m_b, \sigma_b^2)$ and A,B independent. Therefore,

$$A + B = N(x|\mu, \sigma^2)$$

Thus, $\mathbf{L} \cdot \mathbf{X}$ is a Gaussian Normal distribution and as a result $\mathbf{X}$ is a multivariate Gaussian random variable.

Similarly to above, for independent random multivariate variables $X_1 = X$ and $X_2 = Y$, we will show that the distribution f_Z of
$Z = X_3 = X + Y = \mathcal{N}(z|\boldsymbol{\mu}_X + \boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_X + \boldsymbol{\Sigma}_X)$
We know that a Gaussian distribution is fully specified by its mean vector and covariance matrix. If we can determine what these are, then we are done.
For the mean:

$$E[x + y] = E[x] + E[y] = \boldsymbol{\mu}_X + \boldsymbol{\mu}_Y$$

Moreover, the $(i, j)th$ of the covariance matrix is given by:

$$E[(x_i + y_i)(x_j + y_j)] - E[x_i + y_i]E[x_j + y_j] =$$
$$= E[x_i x_j + y_i x_j + x_i y_j + y_i y_j] - (E[x_i] + E[y_i])(E[x_j] + E[y_j]) =$$
$$= E[x_i x_j] + E[y_i x_j] + E[x_i y_j] + E[y_i y_j] - E[x_i]E[x_j] - E[y_i]E[x_j] - E[x_i]E[y_j] - E[y_i][y_j] =$$
$$= (E[x_i x_j] - E[x_i]E[x_j]) + (E[y_i y_j] - E[y_i]E[y_j]) + (E[y_i x_j] - E[y_i]E[x_j]) + (E[x_i y_j] - E[x_i]E[y_j])$$

Using the fact that y and z are independent, we have: $E[y_i x_j] = E[y_i]E[x_j]$and $E[x_i y_j] = E[x_i]E[y_j]$, as well.
Thus, the last two terms drop out, and we have:
$$E[(x_i + y_i)(x_j + y_j)] - E[x_i + y_i]E[x_j + y_j] =$$
$$= (E[x_i x_j] - E[x_i]E[x_j]) + (E[y_i y_j] - E[y_i]E[y_j]) =$$
$$= \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y$$

1.2 (a) Bayes Decision Theory

We have the following Cauchy distribution:

$$p(x|\omega_i) = \frac{1}{\pi b} \cdot \frac{1}{1 + \frac{(x-a_i)^2}{b^2}} = \frac{1}{\pi} \cdot \frac{b}{b^2 + (x - a_i)^2}, i = 1, 2$$

We must show that its integral equals 1. Let θ represent the angle that a line, with fixed point of rotation, makes with the vertical axis. So:

$$tan\theta = \frac{x - a}{b}$$

$$\theta = tan^{-1}\left(\frac{x - a}{b}\right)$$

$$d\theta = \frac{1}{1 + \frac{(x-a)^2}{b^2}} \frac{dx}{b}$$

$$d\theta = \frac{b \cdot dx}{b^2 + (x - a)^2}$$

Thus, the distribution of angle θ is given by:

$$\frac{d\theta}{\pi} = \frac{1}{\pi} \frac{b \cdot dx}{b^2 + (x - a)^2}$$

However, this is normalized over all angles, since

$$\int_{-\pi/2}^{\pi/2} \frac{d\theta}{\pi} = 1$$

and

$$\int_{-\infty}^{\infty} \frac{1}{\pi} \frac{b \cdot dx}{b^2 + (x-a)^2} = \frac{1}{\pi}\left[tan^{-1}\left(\frac{x-a}{b}\right)\right]_{-\infty}^{\infty} = \frac{1}{\pi}\left[\frac{1}{2}\pi - \left(-\frac{1}{2}\pi\right)\right] = 1$$

1.2 (b)

Given that the a priori probabilities $P(\omega_1) = P(\omega_2)$, we would show that $P(\omega_1|x) = P(\omega_2|x)$, if $x = \frac{a_1+a_2}{2}$, that is the decision boundary, that minimizes the error, is the average of the maximum locations of the two Cauchy distributions.
Let $g_1(x)$ and $g_2(x)$ be the discriminative functions of the two categories, ω_1 and ω_2.
Specifically,

$$g_1(x) = p(x|\omega_1)p(\omega_1)$$
$$g_2(x) = p(x|\omega_2)p(\omega_2)$$

In order to find the decision boundary we should solve the below equation:

$$g_1(x) = g_2(x)$$

$$p(x|\omega_1)p(\omega_1) = p(x|\omega_2)p(\omega_2)$$

$$p(x|\omega_1) = p(x|\omega_2)$$

$$\frac{1}{\pi b} \cdot \frac{b}{b^2 + (x-a_1)^2} = \frac{1}{\pi b} \cdot \frac{b}{b^2 + (x-a_2)^2}$$

$$b^2 + (x-a_1)^2 = b^2 + (x-a_2)^2$$

$$(a_2 - a_1)(x - a_1 - a_2) = 0$$

But, $a_1 \neq a_2$ in order the two categories, ω_1, ω_2, to be discriminative.
So,

$$x - a_1 - a_2 = 0$$

$$x = \frac{a_1 + a_2}{2}$$

1.2 (c) Misclassification rate

In order to compute $P(error)$, we could compute first $P(correct)$. Then,

$$P(error) = 1 - P(correct)$$

Let $a_1 < a_2$, given that $p(\omega_1) = p(\omega_2) = 1/2$

$$P(correct) = \sum_{k=1}^{2} p(x \in R_k, \omega_\kappa) = \sum_{k=1}^{2} \int_{R_k} p(x \in R_k, \omega_\kappa)dx =$$

$$\sum_{k=1}^{2} \int_{R_k} p(x \in R_k|\omega_\kappa)p(\omega_\kappa)dx =$$

$$\frac{1}{2}\int_{-\infty}^{\frac{a_1+a_2}{2}} \frac{1}{\pi b} \cdot \frac{b}{b^2 + (x-a_1)^2}dx + \frac{1}{2}\int_{\frac{a_1+a_2}{2}}^{\infty} \frac{1}{\pi b} \cdot \frac{b}{b^2 + (x-a_2)^2}dx =$$

$$\frac{1}{2}\frac{1}{\pi}\left[tan^{-1}\left(\frac{x-a_1}{b}\right)\right]_{-\infty}^{\frac{a_1+a_2}{2}} + \frac{1}{2}\frac{1}{\pi}\left[tan^{-1}\left(\frac{x-a_2}{b}\right)\right]_{\frac{a_1+a_2}{2}}^{\infty} =$$

$$\frac{1}{2}\frac{1}{\pi}\left[tan^{-1}\left(\frac{a_2-a_1}{2b}\right) + \frac{\pi}{2} + \frac{\pi}{2} + tan^{-1}\left(\frac{a_2-a_1}{2b}\right)\right] =$$

$$\frac{1}{2} + \frac{1}{\pi}tan^{-1}\left(\frac{a_2-a_1}{2b}\right)$$

Thus,

$$P(error) = 1 - P(correct) = \frac{1}{2} - \frac{1}{\pi} tan^{-1}\left(\frac{a_2 - a_1}{2b}\right)$$

We can write the general form of the above form, considering $b > 0$,

$$P(error) = 1 - P(correct) = \frac{1}{2} - \frac{1}{\pi} tan^{-1}\left(\frac{|a_2 - a_1|}{2b}\right)$$

1.3 (a) Mahalanobis distance

We define the variables

$$\mathbf{z} = \mathbf{x} - \boldsymbol{\mu}$$

$$\mathbf{B} = \boldsymbol{\Sigma}^{-1}$$

Thus, the Mahalanobis distance can be written as

$${r_i}^2 = \mathbf{z}^T\mathbf{B}\mathbf{z}$$

Thus,

$$\nabla {r_i}^2 = \frac{d(\mathbf{z}^T\mathbf{B}\mathbf{z})}{d\mathbf{z}}$$

Define

$$\mathbf{y} = \mathbf{B}\mathbf{z}$$

So, the gradient of the Mahalanobis distance in terms of $\mathbf{z}$ can be written

$$\nabla {r_i}^2 = \frac{\partial(\mathbf{z}^T\mathbf{y})}{\partial\mathbf{z}} + \frac{d\left(\mathbf{y}(\mathbf{z})^T\right)}{d\mathbf{z}}\frac{\partial(\mathbf{z}^T\mathbf{y})}{\partial\mathbf{y}}$$

$$= \mathbf{y} + \frac{d(\mathbf{z}^T\mathbf{B}^T)}{d\mathbf{z}} = \mathbf{y} + \mathbf{B}^T\mathbf{z}$$

$$= (\mathbf{B} + \mathbf{B}^T)\mathbf{z}$$

$$= 2\mathbf{B}\mathbf{z} = 2\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

1.3 (b) Gradient of the Mahalanobis distance

Let the equation of a line through a given point, $\boldsymbol{\mu}_i$, and parallel to a given vector $\mathbf{a}$

$$\mathbf{x} = \boldsymbol{\mu}_i + \lambda\mathbf{a}$$

Thus,

$$\mathbf{x} - \boldsymbol{\mu}_i = \lambda\mathbf{a}$$

So,

$$\nabla {r_i}^2 = 2\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) = 2\boldsymbol{\Sigma}^{-1}\lambda\mathbf{a} = 2\lambda\boldsymbol{\Sigma}^{-1}\mathbf{a}$$

1.3 (c)

We have the vector line equation through $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$:

$$\mathbf{x} = \boldsymbol{\mu}_1 + l(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Moreover, we can write the above equation as:

$$\mathbf{x} = \boldsymbol{\mu}_2 + k(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Subtracting the above equations and solving in terms of k:

$$\mathbf{0} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + (l - k)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$=> k = l + 1$$

Solving the line equation for point $\boldsymbol{\mu}_1$:

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_1 + l(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$=> l = 0$$

And Solving the line equation for point $\boldsymbol{\mu}_2$:

$$\boldsymbol{\mu}_2 = \boldsymbol{\mu}_1 + l(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$=> l = -1$$

So, $l \in (-1, 0)$. Then, calculating the gradients, we take:

$$\nabla r_1^2 = 2\boldsymbol{\Sigma}^{-1} l(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$
$$\nabla r_2^2 = 2\boldsymbol{\Sigma}^{-1} k(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Then, since $k = l + 1$, we take:

$$\nabla r_1^2 = \frac{l}{l+1} \nabla r_1^2$$

Therefore, since $l \in (-1, 0)$, then $\frac{l}{l+1} < 0$, and as a result ∇r_1^2 and ∇r_1^2 have opposite directions.

1.3 (e)

Given a problem for two classes ω_1, ω_2 with multivariate Gaussian distributions, where $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ and $P(\omega_1) = P(\omega_2)$. Then, the Bayes decision boundary includes all points that have the same Mahalanobis distances from $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$: Correct or Wrong ?

Wrong *Proof:*

We could find decision boundary by equalising $P(\omega_1|\mathbf{x}) = P(\omega_2|\mathbf{x})$:

$$P(\omega_1|\mathbf{x}) = P(\omega_2|\mathbf{x})$$

$$ln(P(\omega_1|\mathbf{x})) = ln(P(\omega_2|\mathbf{x}))$$

$$ln(P(\omega_1)) + (\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) - ln(\sqrt{(2\pi)^n |\Sigma_1|}) = ln(P(\omega_2)) + (\mathbf{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) - ln(\sqrt{(2\pi)^n |\Sigma_2|})$$

$$(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) - ln(\sqrt{(2\pi)^n |\Sigma_1|}) = (\mathbf{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) - ln(\sqrt{(2\pi)^n |\Sigma_2|})$$

$$r_1{}^2 \neq r_2{}^2$$

1.4 (a) Maximum Likelihood estimation

Let $p(x|\theta) = U(0, \theta) = \frac{1}{\theta}$, for $x \in [0, \theta]$ and 0 elsewhere.

Then,

$$L(\theta, \mathbf{x}) = p(\mathbf{x}|\theta) = \prod_{i=1}^{n} \frac{1}{\theta} = \theta^{-n}$$

since $\mathbf{x}$ is iid.

Taking the derivative of ln likelihood:

$$\frac{d\,(lnL(\theta|\mathbf{x}))}{d\theta} = -\frac{n}{\theta} < 0$$

We want to find the value of θ, so that $L(\theta, \mathbf{x})$ becomes maximum.

Therefore,

$$\theta_{ML} = argmax_{\theta} L(\theta, \mathbf{x})$$

It is clear that $L(\theta, \mathbf{x})$ is a decreasing function for $\theta \geq max_i x_i$. So, the likelihood function is maximized, when θ becomes $max_i x_i$.

Thus,

$$\theta_{ML} = max_i x_i$$

1.4 (b)

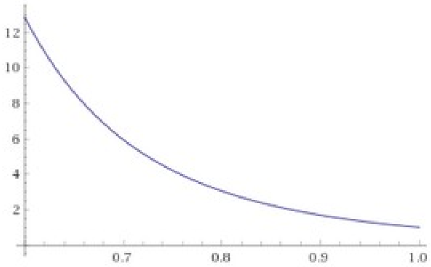
For $n = 5$ and $max_i x_i = 0.6$,

$$L(\theta, \mathbf{x}) = p(\mathbf{x}|\theta) = \prod_{i=1}^{n} \frac{1}{\theta} = \theta^{-n} = \theta^{-5}$$

Thus,

$$\theta_{ML} = max_i x_i = 0.6$$

So, it is not necessary to know any value of the $\{x_i\}$, other than the max value. Therefore, we should plot the likelihood function for $\theta \geq \theta_{ML} = 0.6$:



1.5 (a) k-Nearest Neihbors

Given that for the two categories, ω_1 and ω_2, $p(\mathbf{x}|\omega_i)$ are uniform distributions inside unit hyperspheres at a distance of 10 units and samples belong to either category with probability $p(\omega_i) = \frac{1}{2}$, we note that k-Nearest Neihbors algorithm should perform really well on this classification problem, when there are plenty of samples in both categories, that is inside both hyperspheres, since they are not close to each other.
More specifically, given an odd value of k, in order k-Nearest Neihbors to avoid ties, the algorithm might predict wrong, when there is not a sufficient number of training samples in both classes. This sufficient number depends on the value of k. Since k-Nearest Neihbors needs the aspect of the majority of the k nearest neighbors, for values of number of samples belonging to one category between 0 and $\frac{k-1}{2}$, that is the minority of the nearest k neighbors, the algorithm predicts that a new sample, that belongs to that category, belongs to the other one, and as a result, it fails.
Therefore, the mean error probability for this classification problem is a sum of binomial distributions

$$P_n(e) = \sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j} p(\omega_1)^j (1 - p(\omega_1))^{n-j} = \frac{1}{2^n} \sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j}$$

1.5 (b)

For $k = 1$, the error probability equals the probability for n number of samples, all to belong to the same category. Therefore

$$P_n(e) = \frac{1}{2^n} < \frac{1}{2^n} \sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j}$$

1.5 (c)

$$\frac{1}{2^n} \sum_{j=0}^{a\sqrt{n}} \binom{n}{j} \leq \frac{a\sqrt{n} \cdot n^{a\sqrt{n}}}{2^n} = a\sqrt{n} \frac{2^{a\sqrt{n} logn}}{2^n}$$

$$= \frac{1}{2^{n - a\sqrt{n} logn - \frac{1}{2} logn - loga}}$$

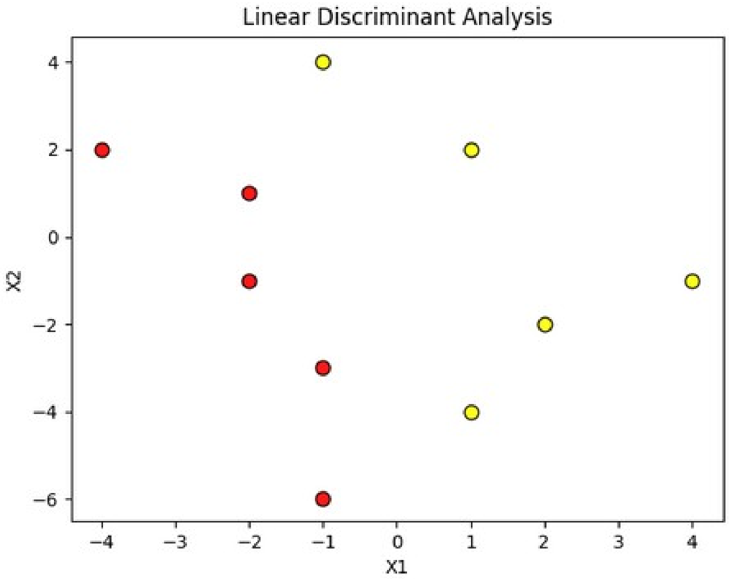
which lim equals 0 when $n \to \infty$

1.6 Perceptrons

At first we define the perceptron model:

- $\mathbf{w}(t+1) = \mathbf{w}(t) + p\mathbf{x}_t$, if $\mathbf{x}_t \in \omega_1$ and $\mathbf{w}(t)^T \mathbf{x}_t \leq 0$
- $\mathbf{w}(t+1) = \mathbf{w}(t) - p\mathbf{x}_t$, if $\mathbf{x}_t \in \omega_2$ and $\mathbf{w}(t)^T \mathbf{x}_t \geq 0$
- $\mathbf{w}(t+1) = \mathbf{w}(t)$, otherwise

Given 10 samples; 5 of ω_1 and 5 of ω_2, and $p = 1$, we should note if they are linearly discriminant.



From the above plot, we note that they are linearly discriminant.
However, with initial $\mathbf{w}(0) = [0, 0]^T$, since the samples given are unbiased, perceptron algorithm will never converge, and will never return a result $\mathbf{w}$, because it seeks for a line equation: $ax + by = 0$.
Thus, we should add the bias term as follows:

$\omega_1 : [1, -1, 4]^T, [1, 1, 2]^T, [1, 2, -2]^T, [1, 1, -4]^T, [1, 4, -1]^T$

$\omega_2 : [1, -4, 2]^T, [1, -2, 1]^T, [1, -2, -1]^T, [1, -1, -3]^T, [1, -1, -6]^T$.

Then, the algorithm would seek for a line equation: $c + ax + by = 0$, where c is the bias term, and will converge, with initial $\mathbf{w}(0) = [0, 0, 0]^T$.
More specifically,

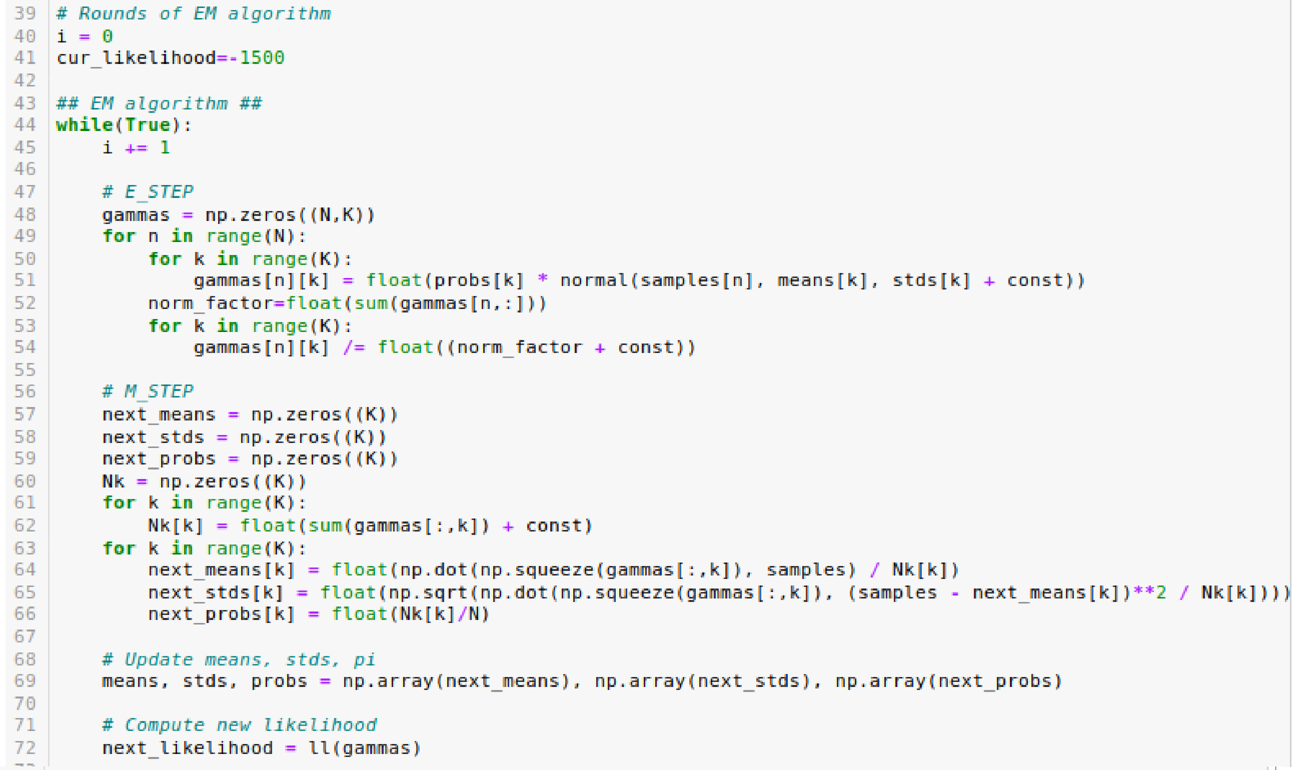
for $t = 0$: $\mathbf{w}(0) = [0, 0, 0]^T$, $\mathbf{x}_0 = [1, -1, 4]$, $\mathbf{w}(1) = [1, -1, 4]$,
for $t = 1$: $\mathbf{w}(1) = [1, -1, 4]^T$, $\mathbf{x}_1 = [1, 1, 2]$, $\mathbf{w}(2) = [1, -1, 4]$, etc.

And, we end up with $\mathbf{w} = [5, 7, 1]^T$. Therefore, the line equation is $5 + 7x + y = 0$.

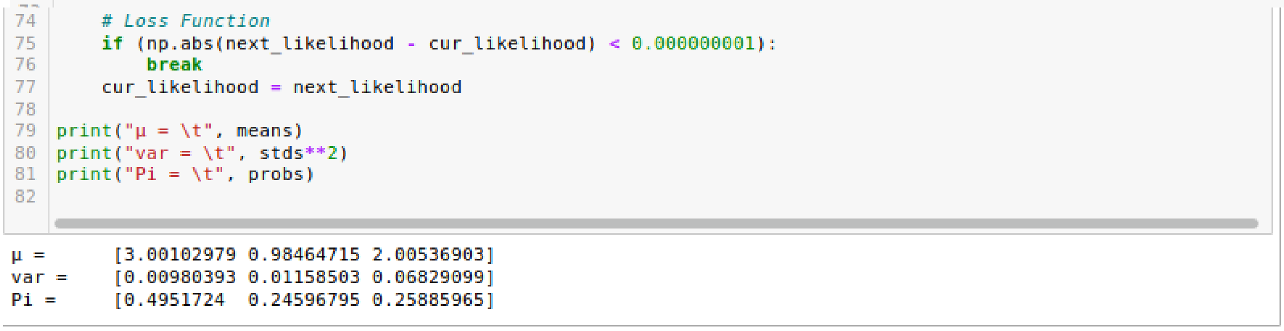
1.7 EM on GMMs

We use Christopher M.Bishop EM algorithm for GMMs in order to maximize the likelihood function with respect to the parameters, comprising the means and covariances of the components and the mixing coefficients.

```
import numpy as np
import scipy.stats

# Dataset
samples = []
for i in range(500):
    mod=i%4
    if mod in [0,1]:
        samples.append(np.random.normal(3, 0.1))
    elif (mod == 2):
        samples.append(np.random.normal(1, 0.1))
    else:
        samples.append(np.random.normal(2, 0.2))

# n_samples
N = len(samples)

# n_classes
K = 3

# Initialization of parameters
stds = [0.2, 0.3, 1]
means = [3.1, 0.7, 2.5]
probs = [0.5, 0.15, 0.15]
const = 0.0000000001


def normal(x, mean, std):
    # Normal Distribution
    return scipy.stats.norm(mean, std).pdf(x)


def ll(gammas):
    temp = 0
    for n in range(N):
        temp += np.log((sum(gammas[n,:]) + const))
    return temp
```



```
# Rounds of EM algorithm
i = 0
cur_likelihood=-1500

## EM algorithm ##
while(True):
    i += 1

    # E_STEP
    gammas = np.zeros((N,K))
    for n in range(N):
        for k in range(K):
            gammas[n][k] = float(probs[k] * normal(samples[n], means[k], stds[k] + const))
        norm_factor=float(sum(gammas[n,:]))
        for k in range(K):
            gammas[n][k] /= float((norm_factor + const))

    # M_STEP
    next_means = np.zeros((K))
    next_stds = np.zeros((K))
    next_probs = np.zeros((K))
    Nk = np.zeros((K))
    for k in range(K):
        Nk[k] = float(sum(gammas[:,k]) + const)
    for k in range(K):
        next_means[k] = float(np.dot(np.squeeze(gammas[:,k]), samples) / Nk[k])
        next_stds[k] = float(np.sqrt(np.dot(np.squeeze(gammas[:,k]), (samples - next_means[k])**2 / Nk[k])))
        next_probs[k] = float(Nk[k]/N)

    # Update means, stds, pi
    means, stds, probs = np.array(next_means), np.array(next_stds), np.array(next_probs)

    # Compute new likelihood
    next_likelihood = ll(gammas)
```



```
    # Loss Function
    if (np.abs(next_likelihood - cur_likelihood) < 0.000000001):
        break
    cur_likelihood = next_likelihood

print("μ = \t", means)
print("var = \t", stds**2)
print("Pi = \t", probs)
```

```
μ =      [3.00102979 0.98464715 2.00536903]
var =    [0.00980393 0.01158503 0.06829099]
Pi =     [0.4951724  0.24596795 0.25885965]
```

We note that EM's Pi output is incredibly close to the expected values of the dataset, having better results than k-means, which result in this problem for Pi is approximately [0.5, 0.27, 0.23]. Moreover, we could have set k-means results as EM inputs for a more efficient result.