EECE 5644 : Machine Learning **September 28, 2019**

Homework #1

Name: Rohin Arora *NUID:* 001302812

Code relevant to the assignment- Hw1 Github

Problem 1 (10 %)

(1) Variance(x) - $Var(x)$; Expectation(x) - $E(x) - \mu$

$$\begin{aligned}
Var(x) &= E[(x-\mu)^2] \\
&= E[x^2 + \mu^2 - 2 * x * \mu] \\
&= E[x^2] + E[\mu^2] - 2 * E[x * \mu] && \text{linearity of expectation} \\
&= E[x^2] + \mu^2 - 2 * \mu * E[x] && \text{linearity of expectation} \\
&= E[x^2] + \mu^2 - 2 * \mu^2 \\
&= E[x^2] - \mu^2
\end{aligned}$$

■

(2) $E[\vec{x}] = \vec{\mu}$; Covariance($\vec{x}$)=$cov(\vec{x})$

$$\begin{aligned}
cov(\vec{x}) &= E[(\vec{x}-\vec{\mu})((\vec{x}-\vec{\mu})^T] \\
&= E[(\vec{x} * \vec{x}^T - \vec{x} * \vec{\mu}^T - \vec{\mu} * \vec{x}^T + \vec{\mu} * \vec{\mu}^T] \\
&= E[(\vec{x} * \vec{x}^T] - E[\vec{x} * \vec{\mu}^T] - E[\vec{\mu} * \vec{x}^T] + E[\vec{\mu} * \vec{\mu}^T] && \text{linearity of expectation} \\
&= E[(\vec{x} * \vec{x}^T] - E[\vec{x}] * \vec{\mu}^T - \vec{\mu} * E[\vec{x}^T] + E[\vec{\mu} * \vec{\mu}^T] && \text{linearity of expectation} \\
&= E[(\vec{x} * \vec{x}^T] - \vec{\mu} * \vec{\mu}^T - \vec{\mu} * \vec{\mu}^T + \vec{\mu} * \vec{\mu}^T && \text{expectation of constant is constant} \\
&= E[(\vec{x} * \vec{x}^T] - \vec{\mu} * \vec{\mu}^T
\end{aligned}$$

■

Problem 2 (20 %)

(1)

- Class conditional probabilities shall sum to 1
- Let the constant of integration with class l be k_l

$$\int_{-\infty}^{\infty} P(X|L=l)dx = k_l * \int_{-\infty}^{\infty} e^{-\frac{|x-a_l|}{b_l}} dx$$

- This integral should sum to 1.
- Consider a modified integral.

$$f(x) = k_l * \int_{a_l}^{\infty} e^{-\frac{x-a_l}{b_l}} dx$$

- The above integral should sum to 1/2, as the function is symmetric. Also shown in code. Refer Appendix here)
- Continuing with this $f(x)$

$$\begin{aligned}
f(x) &= k_l * \int_{a_l}^{\infty} e^{-\frac{x-a_l}{b_l}} dx \\
&= k_l * e^{\frac{a_l}{b_l}} * \int_{a_l}^{\infty} e^{-\frac{x}{b_l}} dx \\
&= k_l * e^{\frac{a_l}{b_l}} * (-b_l) * e^{-\frac{x}{b_l}} \Big|_{a_l}^{\infty} \\
&= k_l * e^{\frac{a_l}{b_l}} * (-b_l) * (0 - e^{-\frac{a_l}{b_l}}) \\
&= k_l * e^{\frac{a_l}{b_l}} * (b_l) * (e^{-\frac{a_l}{b_l}}) \\
&= k_l * b_l \\
&= \frac{1}{2}
\end{aligned}$$

$$\begin{aligned}
k_l &= \frac{1}{2 * b_l} \\
P(X|L=1) &= \frac{1}{2 * b_1} * e^{-\frac{|x-a_1|}{b_1}} \\
P(X|L=2) &= \frac{1}{2 * b_2} * e^{-\frac{|x-a_2|}{b_2}}
\end{aligned}$$

(2)

$$\begin{aligned}
l(x) &= \ln P(X|L=1) - \ln P(X|L=2) \\
&= \ln \frac{1}{2 * b_1} - \ln \frac{1}{2 * b_2} - \frac{|x-a_1|}{b_1} + \frac{|x-a_2|}{b_2} \\
&= \frac{|x-a_2|}{b_2} - \frac{|x-a_1|}{b_1} + \ln \frac{b2}{b_1}
\end{aligned}$$

(3) Setting the values $a_1 = 0, b_1 = 1, a_2 = 1, b_2 = 2$

$$l(x) = \frac{|x-1|}{2} - |x| + \ln 2$$

This function is plotted in the following notebook- Q2_Git_Repo

Problem 3 (20 %)

Minimum probability of error classification rule implies 0 - 1 loss. Also, the 2 classes have equal priors. Therefore, the decision rule simplifies to maximum likelihood estimator. Suppose we have 2 labels, 1 and 2. In such a scenario, the decision rule is-

Decide label 1 if $P(x|L=1) > P(x|L=2)$ **Else decide label 2** (contentions resolved arbitrarily)

Rephrasing-
When $a < x < r$ - **class 1**
When $r < x < b$ - **class 1** if $1/(b-a) > 1/(t-r)$ else **class 2**
When $b < x < t$ - **class 2**

$$P(x|L=1) = \begin{cases} \dfrac{1}{b-a} & a \leq x \leq b \quad (0.1) \\ 0 & elsewhere \quad (0.2) \end{cases}$$

$$P(x|L=2) = \begin{cases} \dfrac{1}{t-r} & r \leq x \leq t \quad (0.3) \\ 0 & elsewhere \quad (0.4) \end{cases}$$

$$\frac{P(x|L=1)}{P(x|L=2)} \gtrless^{class1}_{class2} 1$$

$$\frac{1}{b-a} \gtrless^{class1}_{class2} \frac{1}{t-r}$$

The accompanying code for this question with visual example is present here- Q3_Git_Repo

Problem 4 (30 %)

(1)

- We need to find a decision rule that achieves minimum probability of error. This implies we need to do maximum a posteriori estimation (0-1 loss). It's also given the classes have equal priors. This implies its a special case of MAP- maximum likelihood estimation. The decision rule in such a case is -

$$\frac{P(x|L=1)}{P(x|L=2)} \gtrless_{class2}^{class1} 1$$

$$P(x|L=1) \gtrless_{class2}^{class1} P(x|L=2)$$

$$P(x|L=1) \sim \mathcal{N}(0,\, 1)$$
$$P(x|L=2) \sim \mathcal{N}(\mu,\, \sigma^2)$$
$$\frac{1}{\sqrt{2\pi}} e^{-(x)^2/2} \gtrless_{class2}^{class1} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

$$\sigma * e^{-(x)^2/2} \gtrless_{class2}^{class1} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Take log of both sides

$$\ln\sigma - \frac{x^2}{2} \gtrless_{class2}^{class1} -\frac{(x-\mu)^2}{2\sigma^2}$$

$$2\sigma^2 * \ln\sigma - \sigma^2 * x^2 \gtrless_{class2}^{class1} -(x-\mu)^2$$

$$2\sigma^2 * \ln\sigma - \sigma^2 * x^2 + (x-\mu)^2 \gtrless_{class2}^{class1} 0$$

$$x^2 + \mu^2 - 2x\mu - x^2\mu^2 + 2\sigma^2\ln(\sigma) \gtrless_{class2}^{class1} 0$$

$$x^2(1-\sigma^2) - 2x\mu + \mu^2 + 2\sigma^2\ln\sigma \gtrless_{class2}^{class1} 0 \tag{0.5}$$

- Decision boundary is the above parabola
- When this parabola is > 0, we decide class 1. Else we decide class 2

(2)

- If $\mu = 1$ and $\sigma^2 = 2$, the decision boundary becomes-

$$-x^2 - 2x + 1 + 2\ln 2 \gtrless_{class2}^{class1} 0$$

- This is a parabola facing downwards. The zeros of this equation are $x = -2.84019$ and $x = 0.840189$
- **The decision rule is-**

Class 1	if $-2.84019 < x < 0.840189$
Class 2	otherwise

- Some math demonstrating the posterior calculation -

$$
\begin{aligned}
P(x|L=1) &\sim \mathcal{N}(0,\,1)\\
P(x|L=2) &\sim \mathcal{N}(1,\,2)\\
P(L=1|x) &= \frac{P(x|L=1)P(L=1)}{P(x)}\\
P(L=2|x) &= \frac{P(x|L=2)P(L=2)}{P(x)}\\
\textit{where}\;\; P(x) &= P(x|L=1)P(L=1) + P(x|L=2)P(L=2)\\
\textit{also}\;\; P(L=1) &= P(L=2) \;\text{ as priors are equal}\\
P(L=1|x) &= \frac{P(x|L=1)P(L=1)}{P(x|L=1)P(L=1) + P(x|L=2)P(L=2)}\\
&= \frac{\mathcal{N}(0,\,1)}{\mathcal{N}(0,\,1)\; + \mathcal{N}(1,\,2)}\\
P(L=2|x) &= \frac{P(x|L=2)P(L=2)}{P(x|L=1)P(L=1) + P(x|L=2)P(L=2)}\\
&= \frac{\mathcal{N}(1,\,2)}{\mathcal{N}(0,\,1)\; + \mathcal{N}(1,\,2)}
\end{aligned}
$$

Relevant code- Q4_Git_Repo

(3) * Some Notation.
D == Decide. L==Label. Classes - 1 and 2
D $\in [1, 2]$. L $\in [1, 2]$
class 1 $\sim \mathcal{N}(0,\,1)$
class 2 $\sim \mathcal{N}(1,\,2)$
$F_1(x) = CDF$ of class 1 at x
$F_2(x) = CDF$ of class 2 at x
$\tau 1 = -2.84019$
$\tau 2 = 0.840189$

We decide class 1 between $\tau 1$ and $\tau 2$. And class 2 everywhere else

$$
\begin{aligned}
P(error) &= P(D=1|L=2) * P(L=2) + P(D=2|L=1) * P(L=1)\\
&= 0.5 * [\int_{\tau 1}^{\tau 2} P(x|L=2)dx + \int_{-\infty}^{\tau 1} P(D=2|L=1)dx + \int_{\tau 2}^{\infty} P(D=2|L=1)dx]\\
&= 0.5 * [F_2(\tau 2) - F_2(\tau 1) + F_1(\tau 1) + 1 - F_1(\tau 2)]\\
&= 0.5 * [F_2(0.840189) - F_2(-2.84019) + F_1(-2.84019) + 1 - F_1(0.840189)]\\
&= 0.6433888033787916 * .5\\
&= 0.32169440168
\end{aligned}
$$

Calculation of CDF function in last step can be referred in notebook- Q4_Git_Repo

(4) Putting $\mu = 0$ and *sigma* $>> 1$ in equation 0.5, we get

$$\begin{aligned}
x^2(1-\sigma^2) - 2x\mu + \mu^2 + 2\sigma^2 \ln\sigma &\gtrless_{class2}^{class1} 0 \\
-x^2 * \sigma^2 + 2\sigma^2 \ln\sigma &\gtrless_{class2}^{class1} 0 \\
2\ln\sigma &\gtrless_{class2}^{class1} x^2 \\
\text{class 1 if } x^2 &< 2\ln\sigma \\
\text{class 1 if } -\sqrt{2\ln\sigma} < x &< \sqrt{2\ln\sigma}
\end{aligned}$$

$$\begin{aligned}
\text{Suppose } \sigma &= 25\ (>> 1) \\
\sigma^2 &= 625 \\
\text{class 1 if } -10 < x < 10 \\
\text{else class 2}
\end{aligned}$$

The decision boundary matches with what we might expect intuitively. Suppose 2 classes have same mean. Class 1 has very small variance, and class 2 has very large variance. Far from the mean, class 1 would have very small mass, and chances of class 2 are much higher. Near the mean, class 1 has much high mass than class 2. We should choose class 1 near the mean. The bayes decision rule matches with our intuition.

- In general, such a case can come when we are measuring any random variable normally distributed, and there is additive Gaussian white noise in the channel/measurement process. If we assume the white noise has high variance, it exactly resembles this case. For eg. In astronomy. Consider the experiment setup to distinguish the background noise from actual light of the star. The background noise of the universe can be take all frequencies. Stars we are interested in will emit some wavelengths following a Gaussian with small variance.

Further the probability of mis-classifying class 1 goes down dramatically as sigma increases. This is demonstrated in the following notebook- Q4_Git_Repo

Problem 5 (20 %)

(1)

$$\vec{x} = \boldsymbol{A}\vec{z} + \vec{b} \;\text{ where } z \in \mathcal{N}(0,\, \boldsymbol{I}) \;\text{ and } \vec{x} \;\in R^n$$

By linearity of expectation

$$\begin{aligned}
E[\vec{x}] &= E[\boldsymbol{A}\vec{z} + \vec{b}] \\
&= E[\boldsymbol{A}\vec{z}] + E[\vec{b}] \qquad \boldsymbol{A} \text{ and } \vec{b} \text{ are constants} \\
&= \boldsymbol{A}E[\vec{z}] + \vec{b} \\
E[\vec{z}] &= 0 \\
\text{Therefore, } E[\vec{x}] &= \vec{b}
\end{aligned}$$

$$\begin{aligned}
CoVar(\vec{x}) &= Covar[\boldsymbol{A}\vec{z} + \vec{b}] \\
&= E[(\boldsymbol{A}\vec{z} + \vec{b} - \mu_{A\vec{z}+\vec{b}})(\boldsymbol{A}\vec{z} + \vec{b} - \mu_{A\vec{z}+\vec{b}})^T] \\
&= E[(\boldsymbol{A}\vec{z} + \vec{b} - \vec{b})(\boldsymbol{A}\vec{z} + \vec{b} - \vec{b})^T] \\
&= E[(\boldsymbol{A}\vec{z})(\boldsymbol{A}\vec{z})^T] \\
&= E[(\boldsymbol{A}\vec{z})(\vec{z}^T\boldsymbol{A}^T)] \\
&= \boldsymbol{A}E[(\vec{z})(\vec{z}^T)]\boldsymbol{A}^T \\
E[(\vec{z})(\vec{z}^T)] &= CoVar(\vec{z}) \qquad \text{as } \vec{z} \text{ has zero mean} \\
CoVar(\vec{z}) &= \boldsymbol{I} \qquad \text{given in the question} \\
\text{Therefore, } CoVar(\vec{x}) &= \boldsymbol{A} * \boldsymbol{I} * \boldsymbol{A}^T \\
&= \boldsymbol{A}\boldsymbol{A}^T
\end{aligned}$$

$$\vec{x} \sim \mathcal{N}(\vec{b},\, \boldsymbol{A}\boldsymbol{A}^T)$$

■

(2)
Let the random vector be $\vec{y}$

$$\begin{aligned}
\vec{y} &= \boldsymbol{A}\vec{z} + \vec{b} \\
\vec{y} &\sim \mathcal{N}(\vec{\mu},\, \Sigma)
\end{aligned}$$

From the previous subproblem,

$$\begin{aligned}
\vec{b} &= \vec{\mu} \\
\boldsymbol{A}\boldsymbol{A}^T &= \Sigma
\end{aligned}$$

$\boldsymbol{A}$ can be found by Cholesky decomposition of Σ. As Σ is the covariance matrix of a Gaussian distribution, and it is positive semidefinite. Therefore cholesky decomposition can be applied here. In the code, implementation of Cholesky has been taken from Python's numpy library

$$\boldsymbol{A} = \text{numpy.linalg.cholesky}(\Sigma)$$

(3)

Code for this question- Q5_Git_Repo

References and Acknowledgements

1. Python
2. Scientific python stack
3. Variance
4. Latex
5. Lecture Notes
6. Iridescent
7. DON'T PANIC