STAD68 Assignment 1

Problem 1.

1. Since the distribution of each class is Gaussian, we can calculate the probabilities: $P(Y)$, $P(X|Y)$, $P(X)$, $P(Y|X)$ for naive Bayes

$$g(\tilde{x}|\tilde{\mu},\Sigma) = \frac{1}{\sqrt{2\pi \det(\Sigma)}} e^{(-\frac{1}{2}<(\tilde{x}-\tilde{\mu}), \Sigma^{-1}(\tilde{x}-\tilde{\mu})>)}$$ where $\tilde{\mu}$ is the mean vector, Σ is the covariance matrix

0-1 Loss function: $L^{0-1}(y, f(\tilde{x})) = \begin{cases} 0 & f(\tilde{x}) = y \\ 1 & f(\tilde{x}) \neq y \end{cases}$

$$\begin{aligned}
\hat{y}_{new} = f(\tilde{x}_{new}) &= \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmin}} \sum_{k=1}^{3} L(C_k, y) P(C_k | \tilde{x}_{new}) \\
&= \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmin}} \sum_{k=1}^{3} P(y \neq C_k | \tilde{x}_{new}) \\
&= \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmin}} \sum_{k=1}^{3} (1 - P(y = C_k | \tilde{x}_{new})) \\
&= \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmax}} P(y | \tilde{x}_{new}) \\
&= \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmax}} P(\tilde{x}_{new}|y)P(y) = \underset{y\in\{C_1,C_2,C_3\}}{\operatorname{argmax}} P(y) \prod_{j=1}^{5} P(x_{new}^{(j)} | y)
\end{aligned}$$

2. We use Maximum Likelihood method to estimate parameters

(b) $$P(y=k) = \frac{\sum_{i=1}^{n} I(y_i = k)}{n}$$ where $k = C_1, C_2, C_3$

(a) S'pose the possible values for $x^{(j)}$ is $\{a_{j1}, a_{j2}, \dots a_{jS_j}\}$

$$P(X^{(j)} = a_{jp} | Y = C_k) = \frac{\sum_{i=1}^{n} I(x_i^{(j)} = a_{jp}, y_i = C_k)}{\sum_{i=1}^{n} I(y_i = C_k)}$$

where $j = 1, 2, 3, 4, 5$, $p = 1, 2, \dots S_j$, $k = 1, 2, 3$

3. Naive Bayes would be exact if every dimension of $\tilde{x}$ are conditionally independent given y. i.e.

$$P(\tilde{x}|y) = \prod_{j=1}^{d} P(x_j|y) \quad \text{where } d=5$$

Theoretically when the features of data are conditionally independent given y, the Naive Bayes classifier would ~~perform~~ perform well. Since our data are multivariate Gaussian, when the covariance matrix looks like this:

$$\begin{bmatrix} a & & 0 \\ & b & \\ 0 & & \ddots \; n \end{bmatrix}$$

, namely, the diagonal of the matrix which represents variance has values, and other elements of the matrix $=0$. (the covariance of 2 different random variables are 0)

Problem 2.

1. $f_H(\tilde{x}_1) = \text{sign}(\langle \tilde{x}_1, \tilde{v}_H \rangle - c) = \text{sign}\left(\left\langle \begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle - \frac{1}{2\sqrt{2}}\right) = \text{sign}\left(-\frac{3}{\sqrt{2}} - \frac{1}{2\sqrt{2}}\right) = - < 0$

$f_H(\tilde{x}_2) = \text{sign}(\langle \tilde{x}_2, \tilde{v}_H \rangle - c) = \text{sign}\left(\left\langle \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle - \frac{1}{2\sqrt{2}}\right) = \text{sign}\left(\frac{1}{\sqrt{2}}\right) = + > 0$

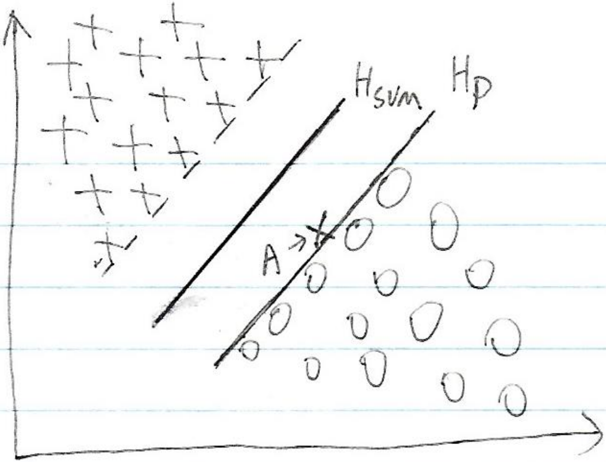
Let the classification be A when the result is < 0

B . . . . . . > 0,

thus $\tilde{x}_1 \in A$, $\tilde{x}_2 \in B$

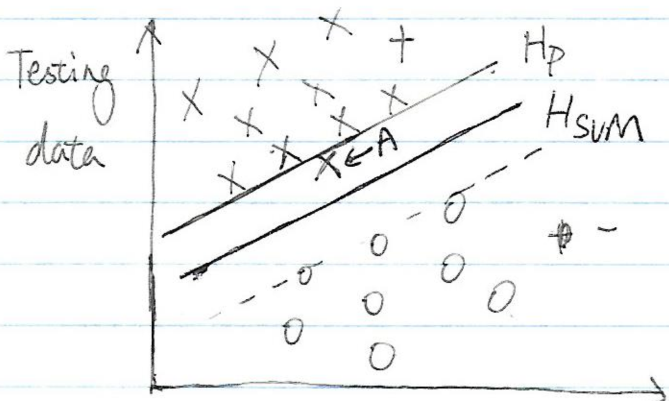
2. Perceptron tries to find a hyperplane that minimizes the total distance from all misclassified data to the hyperplane while SVM tries to find the hyperplane that maximizes the margin and can seperate the data

(1) Perceptron and SVM can disagree on training data when some training data is very close to other classification data



Point A now is classified as "O" by SVM Algorithm and as "X" by ~~Per~~ Perceptron Algorithm as it should be.

(2) Perceptron Algorithm and SVM can disagree on testing data when there are any testing data that is too close to the hyperplane but on the wrong side generated by Perceptron Algorithm



Point A now is classified as "O" by Perceptron Algorithm and as "X" by SVM Algorithm as it should be.