ASSIGNMENT 9   B21140 / SWARNARUP BHUNIA

1) Maximum likelihood Estimation

1.1) i) To maximize the probability of occuring 2-A, 10-B, 60-C, 40-F with their corresponding frequencies:

$$P = \theta^2 \times (3\theta)^{10} \times \left(\frac{1}{2}\right)^{60} \times \left(\frac{1}{2} - 4\theta\right)^{40}.$$

taking - log; $L = 2\ln\theta + 10\ln 3\theta + 60\ln\frac{1}{2} + 40\ln\left(\frac{1}{2} - 4\theta\right)$

$\frac{\partial L}{\partial \theta} = 0$ for maximuml value;

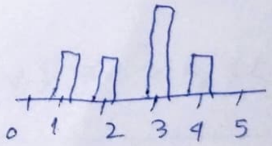
$$\Rightarrow \frac{2}{\theta} + \frac{10}{\theta} + \frac{40}{\frac{1}{2} - 4\theta} \times -4 = 0.$$

$$\frac{12}{\theta} = \frac{4 \times 40}{\frac{1}{2} - 4\theta} \Rightarrow \frac{3}{2} - 12\theta = 40\theta.$$

$$52\theta = \frac{3}{2}$$

$$\theta = \frac{3}{104}.$$

2) The histogram distribution →



$\lambda e^{-\lambda x}$;

$$P = (\lambda e^{-\lambda})^1 (\lambda e^{-2\lambda})^1 (\lambda e^{-\lambda \times 3})^2 (\lambda e^{-\lambda \times 4})$$

$$L = \ln\lambda - \lambda + \ln\lambda - 2\lambda + 2\ln\lambda - 6\lambda + \ln\lambda - 4\lambda$$
$$= 5\ln\lambda - 13\lambda$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow \frac{5}{\lambda} - 13 = 0 \Rightarrow \lambda = \frac{5}{13}.$$

3) $P = (\theta^2)^{K_1} (2\theta(1-\theta))^{K_2} ((1-\theta)^2)^{K_3}$

$L = 2K_1 \ln\theta + K_2 \ln 2\theta + K_2 \ln(1-\theta) + 2K_3 \ln(1-\theta)$.

$$\frac{\partial L}{\partial\theta} = \frac{2K_1}{\theta} + \frac{K_2}{\theta} - \frac{K_2}{1-\theta} - \frac{2K_3}{1-\theta} = 0$$

$(2K_1 + K_2) - 2K_1\theta - 2K_2\theta = K_2\theta + 2K_2\theta + 2K_3\theta$.

$\Rightarrow 2K_1 + K_2 = \theta(2K_1 + 5K_2 + 2K_3)$.

$$\Rightarrow \theta = \frac{2K_1 + K_2}{2K_1 + 5K_2 + 2K_3}.$$

Gaussian Mixture model

2.1)

1) $k = 1, 2 \ldots K$ player playes m_t and wints w_t at day t.

$$P = {}^{m_t}C_{w_t} \, P_k^{w_t} (1 - P_k)^{m_t - w_t}$$

P_k = For each player probability that she wins independent of the day.

$C_t \sim \pi$ which player playes on sat t out of N

$$\gamma_k(n) = P(k \mid n) = \frac{P(k) \, P(n \mid k)}{P(n)}$$

latent variable

$$= \frac{\pi_k \, \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$

where $\pi_k = \frac{N_k}{N}$

hence,

$$\theta_t^{(i)}[k] = P(C_t = k \mid W = w_t) = \frac{P(C_t = t) \, P(W = w_t \mid C_t = t)}{\sum_{k=1}^{K} P(C_t = t_k) \, P(W = W_t \mid C_t = C_{tk})}$$

$$= \frac{\pi[k] \; {}^{m_t}C_{w_t} (P_k)^{w_t} (1 - P_k)^{m_t - w_t}}{\sum_{k=1}^{K} \pi[k] \; {}^{m_t}C_{w_t} \, P_k^{m_t} (1 - P)^{m_t - w_t}}$$

1) E step.
$$Q_t^{(i)}[k] = \frac{\pi^{(i-1)}[k]\, B(w_t, m_t, P_k^{(i-1)})}{\sum_{k=1}^{K} \pi^{(i-1)}[k]\, B(w_t, m_t, P_k^{(i-1)})}$$

2) Log likelihood $= \prod_{t=1}^{n} P(W = w_t)$.

$$= \prod_{t=1}^{n} \sum_{k=1}^{K} \pi_k \; {}^{m_t}C_{w_t}\, P_k^{w_t} (1-P_k)^{m_t - w_t}.$$

$$LL(\pi_k, P_k) = \sum_{t=1}^{n} \ln\left(\sum_{k=1}^{K} \pi_k \; {}^{m_t}C_{w_t}\, P_k^{w_t} (1-P_k)^{m_t - w_t}\right)$$

$$\frac{\partial LL}{\partial P_k} = 0 \Rightarrow \sum_{t=1}^{n} \frac{\pi_k \; {}^{m_t}C_{w_t}\, w_t\, P_k^{(w_t - 1)} (1-P_k)^{m_t - w_t - 1} (-(m_t - w_t))}{D} = 0.$$

$$\Rightarrow \sum_{t=1}^{N} \pi_k P_k^{w_t} (1-P_k)^{m_t - w_t} \times \left(\frac{w_t}{P_k} - \frac{m_t - w_t}{1-P_k}\right) = 0.$$

$$\Rightarrow P_k \sum_{t=1}^{N} Q_t^{(i)}[k]\, w_t - \sum_{t=1}^{N} Q_t^{(i)}[k]\, w_t$$
$$= P_k \sum_{t=1}^{N} Q_t^{(i)}[k]\,(m_t - w_t)$$

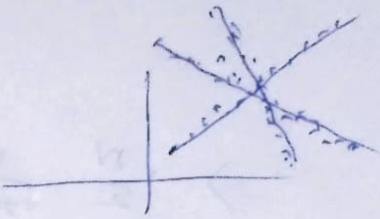
$$P_k = \frac{\sum_{t=1}^{N} Q_t^{(i)}[k]\, w_t}{\sum_{t=1}^{N} Q^{(i)}[k]\, m_t},$$

and $\pi_k = \frac{N_k}{N}$.

$$\frac{\partial LL}{\partial \pi_k} = \sum_{t=1}^{n} \frac{1}{D} \times {}^{m+}C_{n_t} P_k^{w_t} (1-P_k)^{n_t - w_t} = 0.$$

Question - 2

(a) $\gamma_{nk}^{t} = P(z_n = k \mid x_n, y_n, \theta^{(t-1)})$



(x_i, y_i) $x_i \to R^d$

$a_j^T x_i = y_i$ $a_1, a_2 \dots a_m$

$a_i \to R^d.$

$$P(\vec{x}, y \mid \theta) = \sum_{j=1}^{m} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a_j^T x - y)^2}{2\sigma^2}\right).$$

$\theta = (\pi_{1:m}, \vec{a}_{1:m})$ $\sum \pi_j = 1$ $\sigma > 0.$

$$\Rightarrow \gamma_{nk}^{t} = \frac{P(z_n = k)\, P(x_n, y_n, \theta^{(t-1)} \mid z_n = k)}{P(x_n, y_n, \theta^{(t-1)})}$$

$$= \frac{\pi_k^{(t-1)} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{((a_k^{(t-1)})^T x - y)^2}{2\sigma^2}}}{\sum_{k=1}^{m} \pi_k^{(t-1)} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{((a_k^{(t-1)})^T x - y)^2}{2\sigma^2}}}$$

(b) To obtain optimize $\pi_j^{(t)}$ & $a_j^{(t)}$.

$$\frac{\partial LL}{\partial \pi_j} = 0 \quad \& \quad \frac{\partial LL}{\partial a_j} = 0.$$

$$LL = \sum_{n=1}^{N} \ln \left(\sum_{j=1}^{M} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a_j^T x_n - y_n)^2}{2\sigma^2}} \right)$$

$$\frac{\partial LL}{\partial \pi_j} = \sum_{n=1}^{N} \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a_j^T x_n - y_n)^2}{2\sigma^2}}}{\sum_{j=1}^{M} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a_j^T x_n - y_n)^2}{2\sigma^2}}} = 0$$

$$\Rightarrow \sum_{n=1}^{N} \frac{1}{\pi_j} \gamma_{n,j} - 1 = 0.$$

$$\pi_j' = \frac{\sum_{n=1}^{N} \gamma_{nj}}{N} \Rightarrow \pi_j^{+} = \frac{\sum_{n=1}^{N} \gamma_{nj}^{(t)}}{N}.$$

ii) $$\frac{\partial LL}{\partial a_j} = 0 \Rightarrow \sum_{n=1}^{N} \frac{1}{D} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a_j^T x_n - y_n)^2}{2\sigma^2}} \times \frac{-1}{\sigma^2}(a_j^T x_n - y_n) \times x_n = 0.$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nj} \; -\frac{1}{\sigma^2} (a_j^T x_n - y_n) x_n = 0.$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nj} y_n x_n = \sum_{n=1}^{N} \gamma_{nj} a_j^T x_n x_n.$$

$$\Rightarrow (a_j^T x_n).x_n = x_n \cdot x_n^T \cdot a_j$$

$$a_j^{*} = \left(\sum_{n=1}^{N} \gamma_{nj} x_n x_n^T \right)^{-1} \left(\sum_{n=1}^{N} (\gamma_{nj} y_n) x_n \right)$$

$$a_j^{+} = \left(\sum_{n=1}^{N} \gamma_{nj}^{(t)} x_n x_n^T \right)^{-1} \left(\sum_{n=1}^{N} \gamma_{nj}^{(t)} y_n x_n \right)$$

Complexity of SVM.

general algorithm for SVM;

Iterate for ~~all~~ m no. of times:

iterate over all datapoints n:

update w, b.

Overall The complexity = $O(n \times m)$ given it converges before m iteration.

Interms of learning rate or variable learning rate; Complexity becomes,

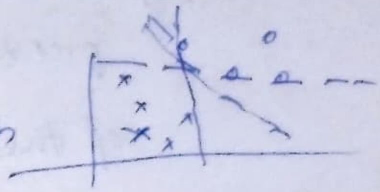
$O(LR \times d(LR) * n \times m)$.

Compare folloning classifires

Logistic Regression:-

Learns a linear boundary.

Objective $f^n = \sum y_i \log\left(\frac{p(x_i)}{1-p(x_i)}\right) + \log(1-p(x_i))$



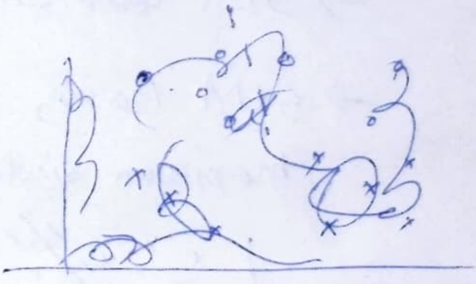
→ learns soft margine.

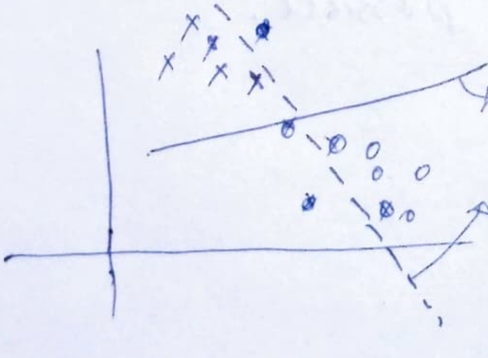
→ learns hyperplanes (need not be optimal hyperplane)

Linear Least Squares Classifier:-

objective $f^n \Rightarrow \underset{w,b}{\arg\min} \sum \sqrt{(y - \sum x_i w - b)^2}$

→ learns the horizontal boundary if there's vertical and horizontal boundary possible





Bayes Classifier

$$\arg\max \prod_{k=1}^{d} p(x_k \mid c_j).$$

→ Bayes classifier does soft clustering.

→ For a given test data It calculates probability of test data being any one of the clusters.

Support Vector Machine

objective f^n $\text{Arg min}\left[\lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(wx_i - b))\right]$

→ SVM can learn Soft or Hard Margine.

→ SVM learns Optimal hyperplane. i.e. maximum distance from all data points. which seperates the clusters as much as possible.