ADABOOST

1A | $$\frac{\partial}{\partial \beta_t}\left[(e^{\beta_t}-e^{-\beta_t})\sum_n w_t(n)\mathbb{I}[y_n \neq h_t(\vec{x}_n)] + e^{-\beta_t}\sum_n w_t(n)\right] = 0$$

substituting $\epsilon_t = \sum_n w_t(n)\mathbb{I}[y_n \neq h_t(\vec{x}_n)]$ and $\sum_n w_t(n) = 1$

$$\frac{\partial}{\partial \beta_t}\left[(e^{\beta_t}-e^{-\beta_t})\epsilon_t + e^{-\beta_t}\right] = 0$$

$$(\beta_t \epsilon_t)e^{\beta_t} - (-\beta_t \epsilon_t)e^{-\beta_t} + (-\beta_t)e^{-\beta_t} = 0$$

$$(\beta_t \epsilon_t)e^{\beta_t} + (\beta_t)(\epsilon_t - 1)e^{-\beta_t} = 0$$

$$\log\left[(\beta_t \epsilon_t)e^{\beta_t} = (\beta_t)(1-\epsilon_t)e^{-\beta_t}\right]$$

note: log is natural log

$$\log\beta_t + \log\epsilon_t + \beta_t = \log\beta_t + \log(1-\epsilon_t) - \beta_t$$

$$2\beta_t = \log(1-\epsilon_t) - \log\epsilon_t$$

$$\boxed{\beta_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}}$$

1B | $\beta_1 = \frac{1}{2}\log\frac{1-\epsilon_1}{\epsilon_1}$

For a linearly separable training set, the hard margin SVM will converge and perfectly classify the training set. Since:

$$\epsilon_t = \sum_n w_t(n)\mathbb{I}[y_n \neq h_t(\vec{x}_n)]$$

and $y_n = h_t(\vec{x}_n)$ for all $n \in \{1, \dots, N\}$,

$$\epsilon_1 = 0, \quad \beta_1 = \infty$$

This means our SVM itself is enough to create a strong classifier, which makes sense since its decision boundary linearly separates all the data.

Kernelized K-means

2A | $\vec{\mu}_K = \dfrac{\sum_{n=1}^{N} z_{nK}\vec{x}_n}{\sum_{n'=1}^{N} z_{n'K}} = \sum_{n=1}^{N}\left(\dfrac{z_{nK}}{\sum_{n'=1}^{N} z_{n'K}}\vec{x}_n\right) = \sum_{n=1}^{N}\alpha_{nK}\vec{x}_n$

$$\boxed{\alpha_{nK} = \frac{z_{nK}}{\sum_{n'=1}^{N} z_{n'K}}}$$

2B | $\|\vec{x}_1 - \vec{x}_2\|_2^2$

$= (x_{11}-x_{21})^2 + (x_{12}-x_{22})^2 + \dots + (x_{1n}-x_{2n})^2$

$= \sum_n (x_{1n}-x_{2n})^2$

$(x_{1n}-x_{2n})^2 = x_{1n}^2 - 2x_{1n}x_{2n} + x_{2n}^2$

$= \sum_n x_{1n}^2 + \sum_n x_{2n}^2 - 2\sum_n x_{1n}x_{2n}$

$\vec{x}_1^T\vec{x}_1 = \sum_n x_{1n}^2 \qquad \vec{x}_2^T\vec{x}_2 = \sum_n x_{2n}^2 \qquad \vec{x}_1^T\vec{x}_2 = \sum_n x_{1n}x_{2n}$

$= \boxed{\vec{x}_1^T\vec{x}_1 + \vec{x}_2^T\vec{x}_2 - 2\vec{x}_1^T\vec{x}_2}$

2C | $\vec{\mu}_K = \sum_{n'=1}^{N}\alpha_{n'K}\vec{x}_{n'}$

$\|\vec{x}_n - \vec{\mu}_K\|_2^2 = \vec{x}_n^T\vec{x}_n + \left(\sum_{n'=1}^{N}\alpha_{n'K}\vec{x}_{n'}\right)^T\left(\sum_{n'=1}^{N}\alpha_{n'K}\vec{x}_{n'}\right)$

$- 2\vec{x}_n^T\left(\sum_{n'=1}^{N}\alpha_{n'K}\vec{x}_{n'}\right)$

- $2\vec{x}_n^T\left(\sum_{n'=1}^{N}\alpha_{n'K}\vec{x}_{n'}\right) = \sum_{n'=1}^{N} 2\alpha_{n'K}\vec{x}_n^T\vec{x}_{n'}$

$$\bullet \left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right)^T \left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ik}\alpha_{jk} \vec{x}_i^T \vec{x}_j$$

Let $\vec{X} = \begin{bmatrix} | & | & & | \\ \alpha_{1k}\vec{x}_1 & \alpha_{2k}\vec{x}_2 & \cdots & \alpha_{nk}\vec{x}_n \\ | & | & & | \end{bmatrix}$

thus, $\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'} = \vec{I}_n \vec{X}$

$$\left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right)^T \left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right) = (\vec{I}_n \vec{X})^T (\vec{I}_n \vec{X})$$

$$= \vec{X}^T \vec{I}_n^T \vec{I}_n \vec{X}$$

$$= \vec{X}^T \vec{X}$$

$$\vec{X}^T\vec{X} = \begin{bmatrix} - \alpha_{1k}\vec{x}_1^T - \\ - \alpha_{2k}\vec{x}_2^T - \\ \vdots \\ - \alpha_{nk}\vec{x}_n^T - \end{bmatrix} \begin{bmatrix} | & | & & | \\ \alpha_{1k}\vec{x}_1 & \alpha_{2k}\vec{x}_2 & \cdots & \alpha_{nk}\vec{x}_n \\ | & | & & | \end{bmatrix}$$

$$= \alpha_{1k}\alpha_{1k}\vec{x}_1^T\vec{x}_1 + \alpha_{1k}\alpha_{2k}\vec{x}_1^T\vec{x}_2 + \ldots + \alpha_{2k}\alpha_{1k}\vec{x}_2^T\vec{x}_1$$
$$+ \ldots + \alpha_{ik}\alpha_{jk}\vec{x}_i^T\vec{x}_j + \ldots + \alpha_{nk}\alpha_{nk}\vec{x}_n^T\vec{x}_n$$

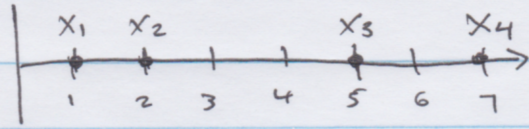
$$= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ik}\alpha_{jk} \vec{x}_i^T \vec{x}_j$$

$$\left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right)^T \left(\sum_{n'=1}^{N} \alpha_{n'k} \vec{x}_{n'}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ik}\alpha_{jk} \vec{x}_i^T \vec{x}_j$$

Therefore,

$$\|\vec{x}_n - \vec{\mu}_k\|_2^2 = \vec{x}_n^T\vec{x} + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ik}\alpha_{jk} \vec{x}_i^T \vec{x}_j - \sum_{n'=1}^{N} 2\alpha_{n'k} \vec{x}_n^T \vec{x}_{n'}$$

K-means for single dimensional data



3a

$r_{11} = 1 \quad r_{32} = 1 \quad r_{43} = 1$

$r_{21} = 1$

$\mu_1 = 1.5 \quad \mu_2 = 5 \quad \mu_3 = 7$

$$J(\{r_{nk}\}, \{\mu_k\}) = (1-1.5)^2 + (2-1.5)^2 + (5-5)^2 + (7-7)^2$$
$$= 0.5$$

3b

suboptimal cluster assignment: $\mu_1 = 1 \quad \mu_2 = 2 \quad \mu_3 = 6$

$r_{11} = 1 \quad r_{22} = 1 \quad r_{33} = 1$

$r_{43} = 1$

$$J(\{r_{nk}\}, \{\mu_k\}) = (1-1)^2 + (2-2)^2 + (5-6)^2 + (7-6)^2$$
$$= 2$$

We can arrive at this suboptimal assignment with the following initialization:

$$\mu_1 = 0, \quad \mu_2 = 3, \quad \mu_3 = 6$$

Run Lloyd's algorithm:

$r_{11} = 1, \quad r_{22} = 1, \quad r_{33} = 1, \quad r_{43} = 1$

$\mu_1 = \frac{1}{1}, \quad \mu_2 = \frac{2}{1}, \quad \mu_3 = \frac{5+7}{2} = 6$

$J(\{r_{nk}\}, \{\mu_k\}) = (1-1)^2 + (2-2)^2 + (5-6)^2 + (7-6)^2 = 2$

$r_{11} = 1, \quad r_{22} = 1, \quad r_{33} = 1, \quad r_{43} = 1$

$\mu_1 = 1, \quad \mu_2 = 2, \quad \mu_3 = 6$

$J = 2$, didn't change from prev. iteration, <u>stop</u>.

This cluster assignment is suboptimal as the final cost = 2, when the optimal cost = 0.5.
With this initialization, the cost function will reach 2 and stop being reduced with Lloyd's algorithm (leaving it trapped at 2, a local minimum). ~~[illegible]~~ In other words, ~~[illegible]~~ the values of r_{nk} and μ_k stop changing with Lloyd's algorithm. This is an example of how Lloyd's algorithm can't guarantee convergence on a global minimum, even when $d=1$.

Hidden Markov Models

4a| $\sum_i q_{ij} = 1$, $q_{11} + q_{21} = 1$

$q_{12} + q_{22} = 1$

$q_{11} = 1$, $q_{12} = 1$, therefore $\boxed{q_{21} = 0 \\ q_{22} = 0}$

$\sum_b e_k(b) = 1$, $e_1(A) + e_1(B) = 1$

$e_2(A) + e_2(B) = 1$

$e_1(A) = 0.99$, $e_2(B) = 0.51$, therefore $\boxed{e_1(B) = 0.01 \\ e_2(A) = 0.49}$

4b| $P(O_1 = A) = P(O_1 = A | q_1 = 1) + P(O_1 = A | q_1 = 2)$

$P(O_1 = B) = P(O_1 = B | q_1 = 1) + P(O_1 = B | q_1 = 2)$

$P(O_1 = A | q_1 = 1) = (0.49)(0.99) = 0.4851$

$P(O_1 = A | q_1 = 2) = (0.51)(0.49) = 0.2499$

$P(O_1 = A) = 0.4851 + 0.2499 = 0.735$

$P(O_1 = B | q_1 = 1) = (0.49)(0.01) = 0.0049$

$P(O_1 = B | q_1 = 2) = (0.51)(0.51) = 0.2601$

$P(O_1 = A) = 0.0049 + 0.2601 = 0.265$

$P(O_1 = A) > P(O_1 = B)$, therefore A is the more frequent output symbol to appear in the first position of being generated from this HMM