20W-COMSCIM146 ps5

JACOB KAUFMAN

TOTAL POINTS

18 / 18

QUESTION 1

Adaboost 5 pts

1.1 a **2 / 2**

✓ **- 0 pts** Correct

- 1 pts The optimal beta should be 1/2 log(1/epsilon-1)

- 2 pts Wrong answer

1.2 b **3 / 3**

✓ **- 0 pts** Correct

- 1 pts In this case beta_1 should be infinite.

- 2 pts You answer is not correct, you should consider the case that the error achieved by SVM is zero.

- 3 pts Wrong answer.

QUESTION 2

K-means 5 pts

2.1 a **2 / 2**

✓ **- 0 pts** Correct

- 1 pts centers are not written

- 1 pts wrong objective value, true objective is 0.5

- 1.5 pts wrong answer

2.2 b **3 / 3**

✓ **- 0 pts** Correct

- 1 pts partial answer, not sufficent explain why it will not be improved

- 3 pts blank

QUESTION 3

Gaussian Mixture 8 pts

3.1 a **2 / 2**

✓ **- 0 pts** Correct

- 2 pts no answer

3.2 b **3 / 3**

✓ **- 0 pts** Correct

- 3 pts no answer

3.3 c **3 / 3**

✓ **- 0 pts** Correct

- 1 pts wrong \mu_1

- 1 pts wrong w_1, w_2

- 3 pts no answer

CM146, Winter 2020
Problem Set 2: SVM and Kernels
Due March 1, 2020 at 11:59 pm

Jacob Kaufman

03/01/2020

1 AdaBoost [5pts]

Solution:

(a) We aim to minimize the objective function listed in the question. This objective function is

$$J = (e^{\beta_t} - e^{-\beta_t}) \sum_n w_t(n) \mathbb{I}[y_n \neq h_t(x_n)] + e^{-\beta_t} \sum_n w_t(n)$$

In order to minimize this for β_t, we take the derivative with respect to β_t and set it to zero, thus finding β_t^*. We do this below.

$$\begin{aligned} \frac{\partial J}{\partial \beta_t} &= \frac{\partial}{\partial \beta_t}\left((e^{\beta_t} - e^{-\beta_t}) \sum_n w_t(n) \mathbb{I}[y_n \neq h_t(x_n)] + e^{-\beta_t} \sum_n w_t(n) \right) \\ &= (e^{\beta_t} + e^{-\beta_t}) \sum_n w_t(n) \mathbb{I}[y_n \neq h_t(x_n)] - e^{-\beta_t} \sum_n w_t(n) \\ &= (e^{\beta_t} + e^{-\beta_t}) \sum_n w_t(n) \mathbb{I}[y_n \neq h_t(x_n)] - e^{-\beta_t}(1) \end{aligned}$$

because $\sum_n w_t(n) = 1$. Now we set this derivative to zero to find β_t^*.

$$
\begin{aligned}
(e^{\beta_t} + e^{-\beta_t}) \sum_n w_t(n)\mathbb{I}[y_n \neq h_t(x_n)] - e^{-\beta_t} &= 0 \\
(e^{\beta_t} + e^{-\beta_t})\epsilon - e^{-\beta_t} &= 0 \\
e^{-\beta_t} &= \epsilon(e^{\beta_t} + e^{-\beta_t}) \\
(e^{\beta_t})(e^{-\beta t}) &= \epsilon(e^{\beta_t})(e^{\beta_t} + e^{-\beta_t}) \\
1 &= \epsilon(e^{2\beta_t} + 1) \\
1 &= \epsilon(e^{2\beta_t}) + \epsilon \\
1 - \epsilon &= (e^{2\beta_t}) \\
e^{2\beta_t} &= \frac{1-\epsilon}{\epsilon} \\
\implies 2\beta_t^* &= \log \frac{1-\epsilon}{\epsilon} \\
\implies \beta_t^* &= \frac{1}{2} \log \frac{1-\epsilon}{\epsilon}
\end{aligned}
$$

(b) We want to find β_1 for a linearly separable hard-margin SVM. This involves a few assumptions, namely:

i. $w_1 = \frac{1}{N}$ because this is the first iteration.

ii. $\sum_n \mathbb{I}[y_n \neq h_1(x_n)] = m$, where m is the number of misclassifications among the N test features.

Using these facts we may construct an appropriate value for β_1. We get

$$
\epsilon = \sum_n w_1(n)\mathbb{I}[y_n \neq h_1(x_n)] = \frac{1}{N}\sum_n \mathbb{I}[y_n \neq h_1(x_n)] = \frac{m}{N}
$$

There are zero misclassifications because we are using a linearly separable dataset. Thus $m = 0 \implies \frac{m}{N} = 0 \implies \epsilon = 0$. Using the formula for β_t^* from (a), we get

$$
\begin{aligned}
\beta_1 &= \frac{1}{2} \log \frac{1-\epsilon}{\epsilon} \\
&= \frac{1}{2} \log \frac{1}{0} \\
\implies & \beta_1 \to \infty
\end{aligned}
$$

1.1 a **2 / 2**

- ✓ **- 0 pts** Correct
- **- 1 pts** The optimal beta should be 1/2 log(1/epsilon-1)
- **- 2 pts** Wrong answer

$$
\begin{aligned}
(e^{\beta_t} + e^{-\beta_t}) \sum_n w_t(n)\mathbb{I}[y_n \neq h_t(x_n)] - e^{-\beta_t} &= 0 \\
(e^{\beta_t} + e^{-\beta_t})\epsilon - e^{-\beta_t} &= 0 \\
e^{-\beta_t} &= \epsilon(e^{\beta_t} + e^{-\beta_t}) \\
(e^{\beta_t})(e^{-\beta t}) &= \epsilon(e^{\beta_t})(e^{\beta_t} + e^{-\beta_t}) \\
1 &= \epsilon(e^{2\beta_t} + 1) \\
1 &= \epsilon(e^{2\beta_t}) + \epsilon \\
1 - \epsilon &= (e^{2\beta_t}) \\
e^{2\beta_t} &= \frac{1-\epsilon}{\epsilon} \\
\implies 2\beta_t^* &= \log \frac{1-\epsilon}{\epsilon} \\
\implies \beta_t^* &= \frac{1}{2} \log \frac{1-\epsilon}{\epsilon}
\end{aligned}
$$

(b) We want to find β_1 for a linearly separable hard-margin SVM. This involves a few assumptions, namely:

i. $w_1 = \frac{1}{N}$ because this is the first iteration.

ii. $\sum_n \mathbb{I}[y_n \neq h_1(x_n)] = m$, where m is the number of misclassifications among the N test features.

Using these facts we may construct an appropriate value for β_1. We get

$$
\epsilon = \sum_n w_1(n)\mathbb{I}[y_n \neq h_1(x_n)] = \frac{1}{N} \sum_n \mathbb{I}[y_n \neq h_1(x_n)] = \frac{m}{N}
$$

There are zero misclassifications because we are using a linearly separable dataset. Thus $m = 0 \implies \frac{m}{N} = 0 \implies \epsilon = 0$. Using the formula for β_t^* from (a), we get

$$
\begin{aligned}
\beta_1 &= \frac{1}{2} \log \frac{1-\epsilon}{\epsilon} \\
&= \frac{1}{2} \log \frac{1}{0} \\
\implies \beta_1 &\to \infty
\end{aligned}
$$

1.2 b 3 / 3

✓ **- 0 pts** Correct

- 1 pts In this case beta_1 should be infinite.

- 2 pts You answer is not correct, you should consider the case that the error achieved by SVM is zero.

- 3 pts Wrong answer.

2 K-means for single-dimensional data [5pts]

Solution:

(a) Because $K = 3$ and $N = 4$, we want to assign two means to the exact values of two data points, assigning the other mean to the two points that are closest together. Thus the optimal clustering assigns

$$\begin{aligned}
\{r_{nk}\} = \{&r_{1,1} = 1, r_{2,1} = 1, r_{3,1} = 0, r_{4,1} = 0,\\
&r_{1,2} = 0, r_{2,2} = 0, r_{3,2} = 1, r_{4,2} = 0,\\
&r_{1,3} = 0, r_{2,3} = 0, r_{3,3} = 0, r_{4,3} = 1\}\\
\{\mu_k\} = \{&\mu_1 = 1.5, \mu_2 = 5, \mu_3 = 7\}
\end{aligned}$$

This clusters x_1 and x_2 together, and x_3 and x_4 individually. The corresponding value of the objective function is

$$\begin{aligned}
J &= \sum_{n=1}^{4}\sum_{k=1}^{3} r_{nk}||x_n - \mu_k||_2^2\\
&= ||1 - 0.5||_2^2 + ||2 - 0.5||_2^2 + ||5 - 5||_2^2 + ||7 - 7||_2^2\\
&= 0.5
\end{aligned}$$

(b) Consider the clustering

$$\{\mu_k\} = \{\mu_1 = 1, \mu_2 = 2, \mu_3 = 6\}$$

and

$$\begin{aligned}
\{r_{nk}\} = \{&r_{1,1} = 1, r_{2,1} = 0, r_{3,1} = 0, r_{4,1} = 0,\\
&r_{1,2} = 0, r_{2,2} = 1, r_{3,2} = 0, r_{4,2} = 0,\\
&r_{1,3} = 0, r_{2,3} = 0, r_{3,3} = 1, r_{4,3} = 1\}
\end{aligned}$$

that clusters x_1 and x_2 individually, and clusters x_3 and x_4 together. This is suboptimal because the objective function has value

2.1 a **2 / 2**

- ✓ **- 0 pts** Correct
- **- 1 pts** centers are not written
- **- 1 pts** wrong objective value, true objective is 0.5
- **- 1.5 pts** wrong answer

2 K-means for single-dimensional data [5pts]

Solution:

(a) Because $K = 3$ and $N = 4$, we want to assign two means to the exact values of two data points, assigning the other mean to the two points that are closest together. Thus the optimal clustering assigns

$$\begin{aligned}\{r_{nk}\} = \{&r_{1,1} = 1, r_{2,1} = 1, r_{3,1} = 0, r_{4,1} = 0,\\ &r_{1,2} = 0, r_{2,2} = 0, r_{3,2} = 1, r_{4,2} = 0,\\ &r_{1,3} = 0, r_{2,3} = 0, r_{3,3} = 0, r_{4,3} = 1\}\\ \{\mu_k\} = \{&\mu_1 = 1.5, \mu_2 = 5, \mu_3 = 7\}\end{aligned}$$

This clusters x_1 and x_2 together, and x_3 and x_4 individually. The corresponding value of the objective function is

$$\begin{aligned}J &= \sum_{n=1}^{4}\sum_{k=1}^{3} r_{nk}||x_n - \mu_k||_2^2\\ &= ||1 - 0.5||_2^2 + ||2 - 0.5||_2^2 + ||5 - 5||_2^2 + ||7 - 7||_2^2\\ &= 0.5\end{aligned}$$

(b) Consider the clustering

$$\{\mu_k\} = \{\mu_1 = 1, \mu_2 = 2, \mu_3 = 6\}$$

and

$$\begin{aligned}\{r_{nk}\} = \{&r_{1,1} = 1, r_{2,1} = 0, r_{3,1} = 0, r_{4,1} = 0,\\ &r_{1,2} = 0, r_{2,2} = 1, r_{3,2} = 0, r_{4,2} = 0,\\ &r_{1,3} = 0, r_{2,3} = 0, r_{3,3} = 1, r_{4,3} = 1\}\end{aligned}$$

that clusters x_1 and x_2 individually, and clusters x_3 and x_4 together. This is suboptimal because the objective function has value

$$
\begin{aligned}
J &= \sum_{n=1}^{4}\sum_{k=1}^{3} r_{nk}||x_n - \mu_k||_2^2 \\
&= ||1-1||_2^2 + ||2-2||_2^2 + ||5-6||_2^2 + ||7-6||_2^2 \\
&= 2 > \frac{1}{2} = J_{\text{optimal}}
\end{aligned}
$$

The algorithm will not improve beyond this initialization. $\{r_{nk}\}$ will not be modified because each data point is closest to its cluster's respective μ, and the k means are already the means of the data points assigned to the means' clusters. This means that $\{\mu_k\}$ will also remain unchanged. Lloyd's algorithm will terminate and return this suboptimal clustering.

2.2 b **3 / 3**

✓ **- 0 pts** Correct

- 1 pts partial answer, not sufficent explain why it will not be improved

- 3 pts blank

3 Gaussian Mixture Models [8 pts]

Solution:

(a) We want to find $\nabla_{\mu_j} l(\theta)$. We do this below.

$$
\begin{aligned}
l(\theta) &= \sum_k \sum_n \gamma_{nk} \log w_k + \sum_k \Big\{ \sum_n \gamma_{nk} \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \Big\} \\
\nabla_{\mu_j} l(\theta) &= \nabla_{\mu_j} \sum_k \Big\{ \sum_n \gamma_{nk} \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \Big\} \\
&= \nabla_{\mu_j} \sum_k \Big\{ \sum_n \gamma_{nk} \log \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\big(-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\big) \Big\} \\
&= \nabla_{\mu_j} \Big\{ \sum_n \gamma_{nj} \log \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\big(-\frac{1}{2}(x_n - \mu_j)^T \Sigma_j^{-1} (x_n - \mu_j)\big) \Big\} \\
&= 0 + \nabla_{\mu_j} \Big\{ \sum_n \gamma_{nj} \big(-\frac{1}{2}(x_n - \mu_j)^T \Sigma_j^{-1} (x_n - \mu_j)\big) \Big\} \\
&= \sum_n \gamma_{nj} \Big(-\Sigma_j^{-1}(x_n - \mu_j) \Big) \\
&= \sum_n \Big(\gamma_{nj} \Sigma_j^{-1} \mu_j - \gamma_{nj} \Sigma_j^{-1} x_n \Big) \\
&= \sum_n \gamma_{nj} \Sigma_j^{-1} \mu_j - \sum_n \gamma_{nj} \Sigma_j^{-1} x_n \\
&= \Sigma_j^{-1} \mu_j \sum_n \gamma_{nj} - \Sigma_j^{-1} \sum_n \gamma_{nj} x_n
\end{aligned}
$$

(b) We set the result from (a) to 0.

3.1 a **2 / 2**

✓ **- 0 pts** Correct

- 2 pts no answer

3 Gaussian Mixture Models [8 pts]

Solution:

(a) We want to find $\nabla_{\mu_j} l(\theta)$. We do this below.

$$
\begin{aligned}
l(\theta) &= \sum_k \sum_n \gamma_{nk} \log w_k + \sum_k \Big\{ \sum_n \gamma_{nk} \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \Big\} \\
\nabla_{\mu_j} l(\theta) &= \nabla_{\mu_j} \sum_k \Big\{ \sum_n \gamma_{nk} \log \mathcal{N}(x_n|\mu_k, \Sigma_k) \Big\} \\
&= \nabla_{\mu_j} \sum_k \Big\{ \sum_n \gamma_{nk} \log \frac{1}{\sqrt{2\pi|\Sigma_k|}} \exp\big(-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\big) \Big\} \\
&= \nabla_{\mu_j} \Big\{ \sum_n \gamma_{nj} \log \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\big(-\frac{1}{2}(x_n - \mu_j)^T \Sigma_j^{-1} (x_n - \mu_j)\big) \Big\} \\
&= 0 + \nabla_{\mu_j} \Big\{ \sum_n \gamma_{nj} \big(-\frac{1}{2}(x_n - \mu_j)^T \Sigma_j^{-1} (x_n - \mu_j)\big) \Big\} \\
&= \sum_n \gamma_{nj} \Big(-\Sigma_j^{-1}(x_n - \mu_j) \Big) \\
&= \sum_n \Big(\gamma_{nj} \Sigma_j^{-1} \mu_j - \gamma_{nj} \Sigma_j^{-1} x_n \Big) \\
&= \sum_n \gamma_{nj} \Sigma_j^{-1} \mu_j - \sum_n \gamma_{nj} \Sigma_j^{-1} x_n \\
&= \Sigma_j^{-1} \mu_j \sum_n \gamma_{nj} - \Sigma_j^{-1} \sum_n \gamma_{nj} x_n
\end{aligned}
$$

(b) We set the result from (a) to 0.

$$
\begin{aligned}
\Sigma_j^{-1}\mu_j \sum_n \gamma_{nj} - \Sigma_j^{-1} \sum_n \gamma_{nj} x_n = 0 \\
\implies \Sigma_j^{-1}\mu_j \sum_n \gamma_{nj} = \Sigma_j^{-1} \sum_n \gamma_{nj} x_n \\
\implies \mu_j \sum_n \gamma_{nj} = \sum_n \gamma_{nj} x_n \\
\implies \mu_j = \frac{\sum_n \gamma_{nj} x_n}{\sum_n \gamma_{nj}}
\end{aligned}
$$

(c) We use the formulae for the EM algorithm to obtain values for w_1, w_2, μ_1 and μ_2.

$$
\begin{aligned}
w_1 = \frac{\sum_n \gamma_{n1}}{\sum_k \sum_n \gamma_{nk}} \\
= \frac{0.2 + 0.2 + 0.8 + 0.9 + 0.9}{0.2 + 0.2 + 0.8 + 0.9 + 0.9 + 0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{0.6} \\
w_2 = \frac{\sum_n \gamma_{n2}}{\sum_k \sum_n \gamma_{nk}} \\
= \frac{0.8 + 0.8 + 0.2 + 0.1 + 0.1}{0.2 + 0.2 + 0.8 + 0.9 + 0.9 + 0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{0.4} \\
\mu_1 = \frac{\sum_n \gamma_{n1} x_n}{\sum_n \gamma_{n1}} \\
= \frac{(0.2)(5) + (0.2)(15) + (0.8)(25) + (0.9)(30) + (0.9)(40)}{0.2 + 0.2 + 0.8 + 0.9 + 0.9} \\
= \boxed{29} \\
\mu_2 = \frac{\sum_n \gamma_{n2} x_n}{\sum_n \gamma_{n2}} \\
= \frac{(0.8)(5) + (0.8)(15) + (0.2)(25) + (0.1)(30) + (0.1)(40)}{0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{14}
\end{aligned}
$$

3.2 b **3 / 3**

✓ **- 0 pts** Correct

- 3 pts no answer

$$
\begin{aligned}
\Sigma_j^{-1}\mu_j \sum_n \gamma_{nj} - \Sigma_j^{-1} \sum_n \gamma_{nj} x_n = 0 \\
\implies \Sigma_j^{-1}\mu_j \sum_n \gamma_{nj} = \Sigma_j^{-1} \sum_n \gamma_{nj} x_n \\
\implies \mu_j \sum_n \gamma_{nj} = \sum_n \gamma_{nj} x_n \\
\implies \mu_j = \frac{\sum_n \gamma_{nj} x_n}{\sum_n \gamma_{nj}}
\end{aligned}
$$

(c) We use the formulae for the EM algorithm to obtain values for w_1, w_2, μ_1 and μ_2.

$$
\begin{aligned}
w_1 = \frac{\sum_n \gamma_{n1}}{\sum_k \sum_n \gamma_{nk}} \\
= \frac{0.2 + 0.2 + 0.8 + 0.9 + 0.9}{0.2 + 0.2 + 0.8 + 0.9 + 0.9 + 0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{0.6} \\
w_2 = \frac{\sum_n \gamma_{n2}}{\sum_k \sum_n \gamma_{nk}} \\
= \frac{0.8 + 0.8 + 0.2 + 0.1 + 0.1}{0.2 + 0.2 + 0.8 + 0.9 + 0.9 + 0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{0.4} \\
\mu_1 = \frac{\sum_n \gamma_{n1} x_n}{\sum_n \gamma_{n1}} \\
= \frac{(0.2)(5) + (0.2)(15) + (0.8)(25) + (0.9)(30) + (0.9)(40)}{0.2 + 0.2 + 0.8 + 0.9 + 0.9} \\
= \boxed{29} \\
\mu_2 = \frac{\sum_n \gamma_{n2} x_n}{\sum_n \gamma_{n2}} \\
= \frac{(0.8)(5) + (0.8)(15) + (0.2)(25) + (0.1)(30) + (0.1)(40)}{0.8 + 0.8 + 0.2 + 0.1 + 0.1} \\
= \boxed{14}
\end{aligned}
$$

3.3 C 3 / 3

- ✓ **- 0 pts** Correct
- **- 1 pts** wrong \mu_1
- **- 1 pts** wrong w_1, w_2
- **- 3 pts** no answer