Pattern Recognition Assignment#3

61821313 Zihao Wang

May 6, 2023

Question1

The final weight in perceptron can be computed through the code in appendix

$$(\text{a}) \longrightarrow \hat{\boldsymbol{a}} = (-4,\ 4,\ -1)^{\mathrm{T}}$$

$$(\text{b}) \longrightarrow \hat{\boldsymbol{a}} = (-4,\ 2,\ 0.5)^{\mathrm{T}}$$

Here is the log of fixed-increment single-sample correction algrithom and batch perceptron algrithom respectively

error	weight	error	weight
$\boldsymbol{y}_1$	$(1,\ 4,\ 1)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-2,\ 4,\ -1)^{\mathrm{T}}$
$\boldsymbol{y}_3$	$(0,\ 4,\ -1)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-3,\ 3,\ -2)^{\mathrm{T}}$
$\boldsymbol{y}_4$	$(-1,\ 3,\ -2)^{\mathrm{T}}$	$\boldsymbol{y}_2$	$(-2,\ 5,\ 0)^{\mathrm{T}}$
$\boldsymbol{y}_4$	$(-2,\ 2,\ -3)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-3,\ 4,\ -1)^{\mathrm{T}}$
$\boldsymbol{y}_2$	$(-1,\ 4,\ -1)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-4,\ 3,\ -2)^{\mathrm{T}}$
$\boldsymbol{y}_4$	$(-2,\ 3,\ -2)^{\mathrm{T}}$	$\boldsymbol{y}_2$	$(-3,\ 5,\ 0)^{\mathrm{T}}$
$\boldsymbol{y}_2$	$(-1,\ 5,\ 0)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-4,\ 4,\ -1)^{\mathrm{T}}$

error	weight	error	weight
$\boldsymbol{y}_1$		$\boldsymbol{y}_2$	$(-3,\ 2,\ 1.5)^{\mathrm{T}}$
$\boldsymbol{y}_2$	$(-2,\ 2,\ 2.5)^{\mathrm{T}}$	$\boldsymbol{y}_3$	
$\boldsymbol{y}_3$		$\boldsymbol{y}_4$	$(-4,\ 1.5,\ 0)^{\mathrm{T}}$
$\boldsymbol{y}_4$	$(-3,\ 1.5,\ 1)^{\mathrm{T}}$	$\boldsymbol{y}_2$	$(-3.5,\ 2.5,\ 1)^{\mathrm{T}}$
$\boldsymbol{y}_4$	$(-3.5,\ 1,\ 0.5)^{\mathrm{T}}$	$\boldsymbol{y}_4$	$(-4,\ 2,\ 0.5)^{\mathrm{T}}$

And the result discriminant function is

$$(\text{a}) \longrightarrow g(\boldsymbol{y}) = \hat{\boldsymbol{a}}^{\mathrm{T}}\boldsymbol{y} = -4y_1 + 4y_2 - y_3$$

$$(\text{b}) \longrightarrow g(\boldsymbol{y}) = \hat{\boldsymbol{a}}^{\mathrm{T}}\boldsymbol{y} = -4y_1 + 2y_2 + 0.5y_3$$

Question2

Let $\ell_1(a,\ b)$ and $\ell_2(a,\ b)$ be the left hand and be the right hand respectively

$$\ell_1(a,\ b) = \frac{1}{m}||a\boldsymbol{v} + b - \boldsymbol{y}||^2$$
$$\ell_2(a,\ b) = \frac{1}{m}||a(\boldsymbol{v} - \bar{v}) + b - (\boldsymbol{y} - \bar{y})||^2$$

Then let $\hat{a}_1$ and $\hat{b}_1$ be the minimizers of ℓ_1

$$\hat{a}_1,\ \hat{b}_1 = arg \min_{a,\ b} \ell_1$$

Plugging them back to ℓ_2

$$\ell_2(\hat{a}_1,\ \hat{b}_1) = \frac{1}{m}||\hat{a}_1(\boldsymbol{v} - \bar{v}) + \hat{b}_1 - (\boldsymbol{y} - \bar{y})||^2 = \frac{1}{m}||\hat{a}_1\boldsymbol{v} + (\hat{b}_1 - \hat{a}_1\bar{v} + \bar{y}) - \boldsymbol{y}||^2$$

Let

$$\hat{a}_2 = \hat{a}_1 \qquad \hat{b}_2 = \hat{b}_1 - \hat{a}_1\bar{v} + \bar{y},$$

Plugging them back to ℓ_2

$$\ell_2(\hat{a}_2,\ \hat{b}_2) = \ell_1(\hat{a}_1,\ \hat{b}_1) = \min_{a,\ b} \ell_1$$

The above equation shows that

$$\min_{a,\ b} \ell_2 \leq \ell_2(\hat{a}_2,\ \hat{b}_2) = \min_{a,\ b} \ell_1$$

Similarly, it can be proven that

$$\min_{a,\ b} \ell_1 \leq \min_{a,\ b} \ell_2$$

Therefore

$$\min_{a,\ b} \ell_1 = \min_{a,\ b} \ell_2$$

Question3

Calculate the global scatter matrix $\mathbf{S}_t$

$$\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i = (3,\ 1,\ 1.67)^{\mathrm{T}}$$

$$\mathbf{S}_t = \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{m})(\boldsymbol{x}_i - \boldsymbol{m})^{\mathrm{T}} = \begin{pmatrix} 2 & -2 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 10.67 \end{pmatrix}$$

Calculate eigen values and corresponding eigen vectors of $\mathbf{S}_t$

$$\mathbf{S}_t = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^{-1}$$

$$\boldsymbol{\Lambda} = diag(10.67,\ 4,\ 0) \qquad \mathbf{P} = \begin{pmatrix} 0 & 0.707 & 0.707 \\ 0 & -0.707 & 0.707 \\ 1 & 0 & 0 \end{pmatrix}$$

Therefore, the best direction of projection

$$\boldsymbol{e} = (0,\ 0,\ 1)^{\mathrm{T}}$$

Samples after dimension reduction through PCA

$$\boldsymbol{a}_i = \boldsymbol{e}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{m})$$

$$\boldsymbol{a}_1 = 1.33 \quad \boldsymbol{a}_2 = 1.33 \quad \boldsymbol{a}_3 = -2.67$$

Question4

Calculate the mean value of samples in each categories

$$\boldsymbol{m}_1 = \frac{1}{|D_1|}\sum_{\boldsymbol{x}\in D_1}\boldsymbol{x} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \qquad \boldsymbol{m}_2 = \frac{1}{|D_2|}\sum_{\boldsymbol{x}\in D_2}\boldsymbol{x} = \begin{pmatrix} -2 \\ -5 \end{pmatrix}$$

Calculate the between-class scatter matrix $\mathbf{S}_b$

$$\mathbf{S}_b = (\boldsymbol{m}_1 - \boldsymbol{m}_2)^{\mathrm{T}}(\boldsymbol{m}_1 - \boldsymbol{m}_2) = \begin{pmatrix} 16 & 36 \\ 36 & 81 \end{pmatrix}$$

Calculate the within-class scatter matrix $\mathbf{S}_w$

$$\mathbf{S}_w = \sum_{D_i}\sum_{\boldsymbol{x}\in D_i}(\boldsymbol{x} - \boldsymbol{m}_i)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{m}_i) = \begin{pmatrix} 4 & 8 \\ 8 & 22 \end{pmatrix}$$

Therefore, the LDA criterion function $J(\boldsymbol{w})$ is

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}} = \frac{16w_1^2 + 72w_1w_2 + 81w_2^2}{4w_1^2 + 16w_1w_2 + 22w_2^2}$$

Code Appendix

```
import numpy as np

y1 = np.array([1., 4., 1.])

y2 = np.array([1., 2., 2.])

y3 = np.array([-1., 0., -2.])

y4 = np.array([-1., -1., -1.])


class perceptron:

    def __init__(self) -> None:

        self.a = None

    def single_fit(

            self,

            initial_value,

            train_set,

            eta:float,

            theta:float=0.,

            max_iteration:int=1000

    ) -> None:

        self.a = np.copy(initial_value)

        print('initial weight:', self.a)

        for i in range(max_iteration):

            print('epoch:', i + 1)

            flag = True

            for y in train_set:

                if np.dot(self.a, y) <= theta:

                    print('error in sample:', y, end='\t\t')

                    flag = False

                    self.a += eta * y

                    print('weight after correction:', self.a)

            if flag:

                print('successfully fit the training set')

                print('the final weight is:', self.a)

                return

        print('failed to fit the training set in given iteration
             times')

    def batch_fit(

            self,

            initial_value,

            train_set,

            eta:float,

            theta:float=0.,

            max_iteration:int=1000

    ) -> None:

        self.a = np.copy(initial_value)

        print('initial weight:', self.a)

        for i in range(max_iteration):

            print('epoch:', i + 1)

            error_set = []

            for y in train_set:

                if np.dot(self.a, y) <= theta:

                    error_set.append(y)

            if len(error_set) == 0:

                print('successfully fit the training set')

                print('the final weight is:', self.a)

                return

            for y in error_set:

                print('error in sample:', y)

            self.a += eta * np.sum(error_set, axis=0)

            print('weight after correction', self.a)

        print('failed to fit the training set in given iteration
             times')


single_model = perceptron()

batch_model = perceptron()


single_model.single_fit(

    initial_value=np.array([0., 0., 0.]),

    train_set=[y1, y2, y3, y4],

    eta=1.,

    theta=0.,

    max_iteration=1000

)

batch_model.batch_fit(

    initial_value=np.array([-3., -1., 1.]),

    train_set=[y1, y2, y3, y4],

    eta=0.5,

    theta=0.5,

    max_iteration=1000

)
```