Linear Discriminant Analysis (LDA)

https://github.com/Huafeng-XU

1.Significance of LDA

LDA is a **supervised learning** method.
Seek projections that best separate the data in a least-squares sense.(Find component that are efficient for distinguish from different classes).

2.Formula description

In this part, I will show you the Formula description of LDA from my teacher showed for me. Actually, it is uneasy to understand its principle.

Feeling it difficult? Me too. If you just want to know how LDA work on matlab, you can **skip this part** and read part 3.

- The objective function of LDA:

$$\boldsymbol{w}_{\text{opt}} = \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}}$$

$$\mathbf{S}_B = \sum_{i=1}^{l} n_i \left(\mathbf{m}^{(i)} - \mathbf{m}\right)\left(\mathbf{m}^{(i)} - \mathbf{m}\right)^T, \quad \mathbf{S}_W = \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\boldsymbol{x}_j^{(i)} - \mathbf{m}^{(i)}\right)\left(\boldsymbol{x}_j^{(i)} - \mathbf{m}^{(i)}\right)^T\right)$$

where l = number of classes, n_i = no. of samples in class i

- Derive projection vectors for discriminating data from different classes
- Suppose a set of n d-dimensional samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$

  n_1 in the subset X_1 labelled ω_1

  n_2 in the subset X_2 labelled ω_2
- A linear projection of $\boldsymbol{x}$ onto $\boldsymbol{w} = [w_1\ w_2\ \ldots\ w_d]^T$

$$\begin{aligned} y &= w_1 x_1 + \ldots + w_d x_d \\ &= \boldsymbol{w}^T \boldsymbol{x} \end{aligned}$$

$\Rightarrow$ Projection of these n samples $y_1, \ldots, y_n$ divided into the subsets Y_1 and Y_2

- A measure of the separation between the projected points is the difference of the sample means

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in X_i} \mathbf{x}$$

then the sample mean for the projected points

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y$$

$$= \frac{1}{n_i} \sum_{\mathbf{x} \in X_i} \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{m}_i$$

$$\Rightarrow |\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)|$$

- The difference between the means to be large relative to some measure of the standard deviations for each class
- The total *within-class scatter*: $\tilde{s}_1^2 + \tilde{s}_2^2$

where $\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2$ the *scatter* for projected samples labelled ω_i

- The *Fisher linear discriminant* is defined as that linear function $\mathbf{w}^T\mathbf{x}$ for which the *criterion function*

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$ is maximum

- Define the **scatter matrices** $\mathbf{S}_i$ and $\mathbf{S}_w$ by

$$\mathbf{S}_i = \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

and $$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

then $$\tilde{s}_i^2 = \sum_{\mathbf{x} \in X_i} (\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_i)^2$$

$$= \sum_{\mathbf{x} \in X_i} \mathbf{w}^T (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \mathbf{w}$$

$$= \mathbf{w}^T \mathbf{S}_i \mathbf{w}$$

so that $$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T \mathbf{S}_W \mathbf{w}$$

Here is the most import: **SW** and **SB**. If you read this part, I hope that you can understand the concept.

Similarly
$$(\tilde{m}_1 - \tilde{m}_2)^2 = (\mathbf{w}^T \mathbf{m}_1 - \mathbf{w}^T \mathbf{m}_2)^2$$
$$= \mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w}$$
$$= \mathbf{w}^T \mathbf{S}_B \mathbf{w}$$

where
$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$

$\mathbf{S}_W$ is called the *within-class scatter matrix*

$\mathbf{S}_B$ is called the *between-class scatter matrix*

- both $\mathbf{S}_W$ and $\mathbf{S}_B$ are symmetric and positive semi-definite
- for any $\mathbf{w}$, $\mathbf{S}_B\mathbf{w}$ is in the direction of $\mathbf{m}_1 - \mathbf{m}_2$

$$\mathbf{S}_B\mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w}$$

- The criterion function J can be written as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

a vector $\mathbf{w}$ that maximizes J must satisfy
$$\mathbf{S}_B\mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$$

or
$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda \mathbf{w}$$

As $S_B\mathbf{w}$ is always in the direction of $\mathbf{m}_1 - \mathbf{m}_2$,
$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$
this converts a d-dimensional problem to a hopefully more manageable one-dimensional problem

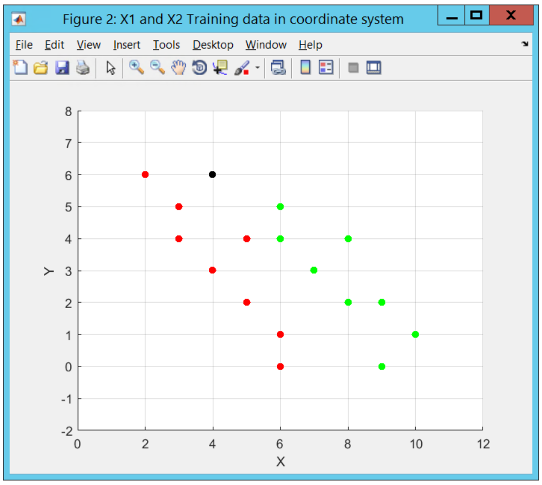
- ## 3.Sample and code on matlab

Consider the following two sets of training data, where the samples are drawn randomly:
$X_1 = \{(2,6),(3,4),(3,5),(4,3),(5,2),(5,4),(6,0),(6,1)\}$
$X_2 = \{(6,4),(6,5),(7,3),(8,2),(8,4),(9,0),(9,2),(10,1)\}$
Assume the $q = (4,6)$ is a query input. You need to find out q belongs to which class.

Question: By using Linear Discriminant Analysis (LDA) method to classify the query input q belongs to which class.

Method:

➢ **Step 1**: Plot X1 and X2 Training data in coordinate system using Matlab. X_1 marked at red, X_2 marked at green and q marked at black.



You can easily use the code below at Matlab to achieve this:

```
clear all;
%row 1 represent x in coordinate system.
%row 2 represent y in coordinate system.
X1=[2 3 3 4 5 5 6 6
   6 4 5 3 2 4 0 1];
X1_x = X1(1,:);
X1_y = X1(2,:);

X2=[6 6 7 8 8 9 9 10
   4 5 3 2 4 0 2 1];
X2_x = X2(1,:);
X2_y = X2(2,:);

%Plot X1 and X2 Training data in coordinate system.
figure('Name','X1 and X2 Training data in coordinate system')
scatter(X1_x,X1_y,'filled','red')
hold on
scatter(X2_x,X2_y,'filled','green')
scatter(4,6,'filled','black')
grid on
axis([0 12 -2 8])
xlabel('X');
ylabel('Y');
```

Hoping that you can type the code yourself. But if you think this may waste your time. You can just download all the code from my **GitHub**: https://github.com/Huafeng-XU. It is in the ***Linear-Discriminant-Analysis-LDA-/****Code*. Actually, step 1 is not necessary. However, it is good for us to observe the training data firstly.

- **Step 2**: LDA is supervised learning method. Thus, we assume that X1 belongs to class1 and X2 belong to class2.

  Calculate the mean:

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

The code of Matlab is below:

```
%Calculate the mean value of X1,X2
meanX1 = (1/8 * sum(X1'))';
meanX2 = (1/8 * sum(X2'))';
```

- **Step 3**: Calculate the **scatter matrices** Si :

$$Si = \sum_{i=1}^{n} (xi - mi) \cdot (xi - mi)^T$$

The code of Matlab is below:

```
SX1 = (X1 - meanX1) * (X1 - meanX1)';
SX2 = (X2 - meanX2) * (X2 - meanX2)';
```

- **Step 4**: Calculate the **within-class scattering matrix** Sw:

$$Sw = \sum_{i}^{n} Si$$

The code of Matlab is below:

```
Sw = SX1 + SX2;
```

- **Step 5**: Calculate the **Matrix inverse** of Sw :

$$SwP = Sw^{-1}$$

The code of Matlab is below:

```
SwP = inv(Sw);
```

- **Step 6**: Calculate the **projection vector**:

$$w = Sw^{-1} \cdot (meanX1 - meanX2)$$

The code of Matlab is below:

```
w = SwP * (meanX1 - meanX2);
```

(Note that ***-w*** is also correct if you choose it to use)

- **Step 7**: Calculate the **projection value**:

$$Yi = w' \cdot Xi$$

The code of Matlab is below:

```
Y1 = w' * X1;
Y2 = w' * X2;
```

- **Step 8**: Using the minimum distance classifier, calculate the mean of the projected samples of each class:

$$meanY1 = \frac{1}{8} \cdot Y1$$

$$meanY2 = \frac{1}{8} \cdot Y2$$

The code of Matlab is below:

```
meanY1 = 1/8 * sum(Y1);
meanY2 = 1/8 * sum(Y2);
```

- **Step 9**: Calculate the accuracy or the recognition rate of the minimum distance classifier, based on LDA.

$$Di = |Yi - m|$$

For class1 ($w1$, data X1 or Y1):

$$X1Di_1 = |Y1i - meanY1|$$

$$X1Di_2 = |Y1i - meanY2|$$

IF $X1Di_1 < X1Di_2$, Then $Y1i$ belongs to class1, X1count ++. Otherwise, it belongs to class2.

For class2($w2$, data X2 or Y2) is just the same.

The code of Matlab is below:

```
%Find the accuracy
%For X1
X1D1 = abs(Y1-meanY1);
X1D2 = abs(Y1-meanY2);
X1count = size(find((X1D1-X1D2)<0),2);
%For X2
X2D1 = abs(Y2-meanY1);
X2D2 = abs(Y2-meanY2);
X2count = size(find((X2D1-X2D2)>0),2);
%The LDA accuracy.
LDA_accuracy = (X1count + X2count)/(size(X1,2)+size(X2,2))
```

- **Step 10**: Find out the query input $q = (4,6)$ belongs to which class.

$$yq = w' \cdot q$$

The distance to Class1

$$qD1 = |yq - meanY1|$$

The distance to Class2

$$qD2 = |yq - meanY2|$$

IF $qD1 < qD2$, Then q belongs to class1;
Otherwise, q belongs to class2.

The code of Matlab is below:

```
%Find out the query input q=(4,6) belongs to
which class
q = [4 6]';
yq = w' * q;
%the distance to class1
qD1 = abs(yq-meanY1);
%the distance to class2
qD2 = abs(yq-meanY2);
if qD1 < qD2
   class =  1;
else
   class = 2;
end
class
```

- The ***classs* = *1*** means that q belongs to class1 and ***classs* = *2*** means that q belongs to class2.

  I hope that my work can give you a hand to help you understand LDA better. Thank you for read.

Reference:

[1]. Pattern Recognition:Theory & Application Prof.Kennetth K.M.Lam.
[2]. S. Balakrishnama, A. J. I. f. S. Ganapathiraju, and i. Processing, "Linear discriminant analysis-a brief tutorial," vol. 18, pp. 1-8, 1998.