机器学习之监督学习

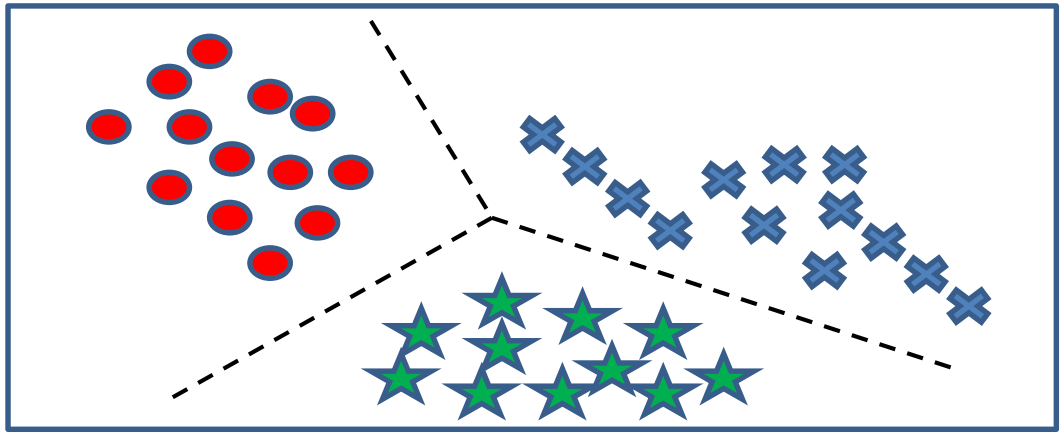
线性判别分析(LDA)

倪冰冰
上海交通大学

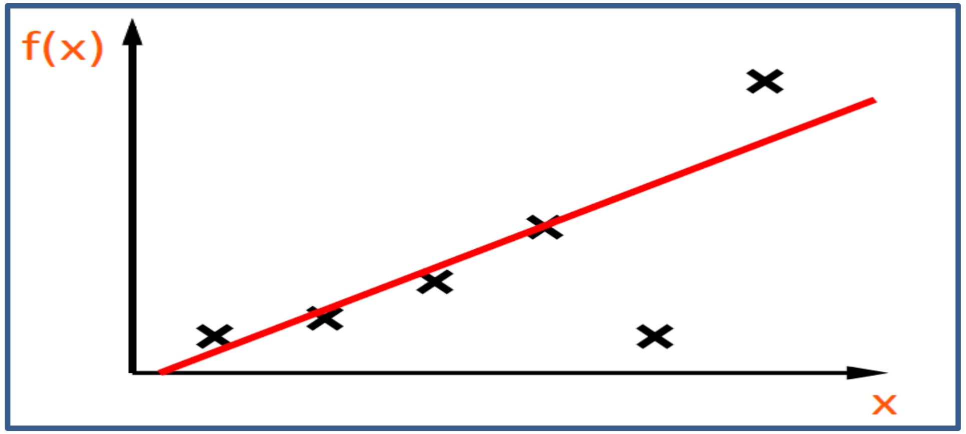
监督学习

- 给定一组数据，我们知道正确的输出结果应该是什么样子，并且知道在输入和输出之间有着一个特定的关系$f(x)$。

- 分类 vs 回归



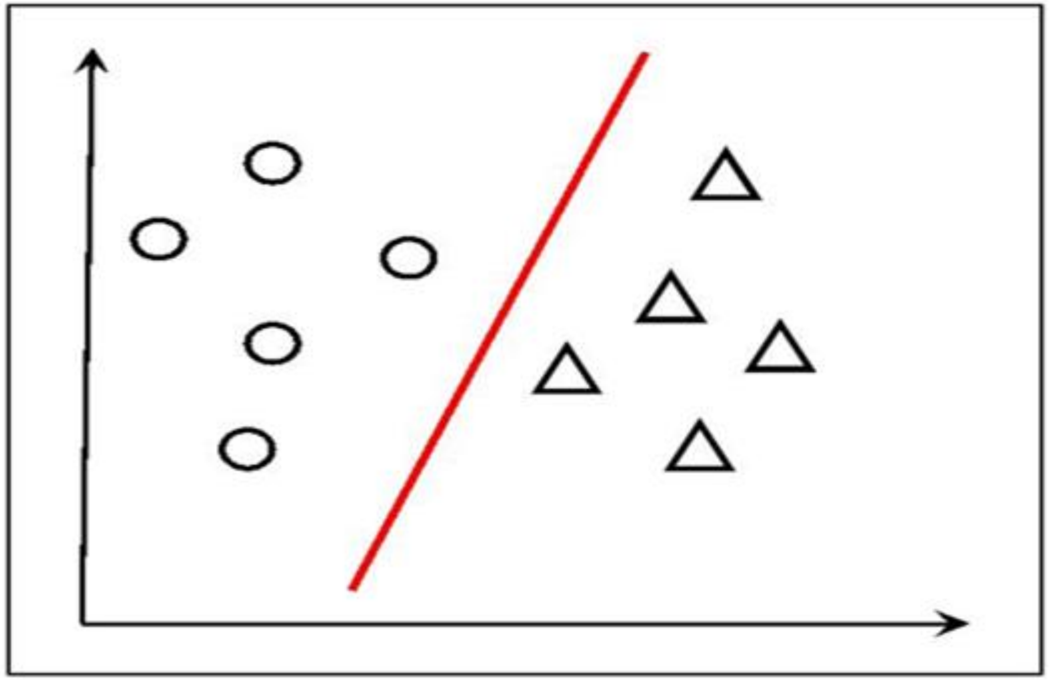
分类(Classification)



回归(Regression)

线性分类模型

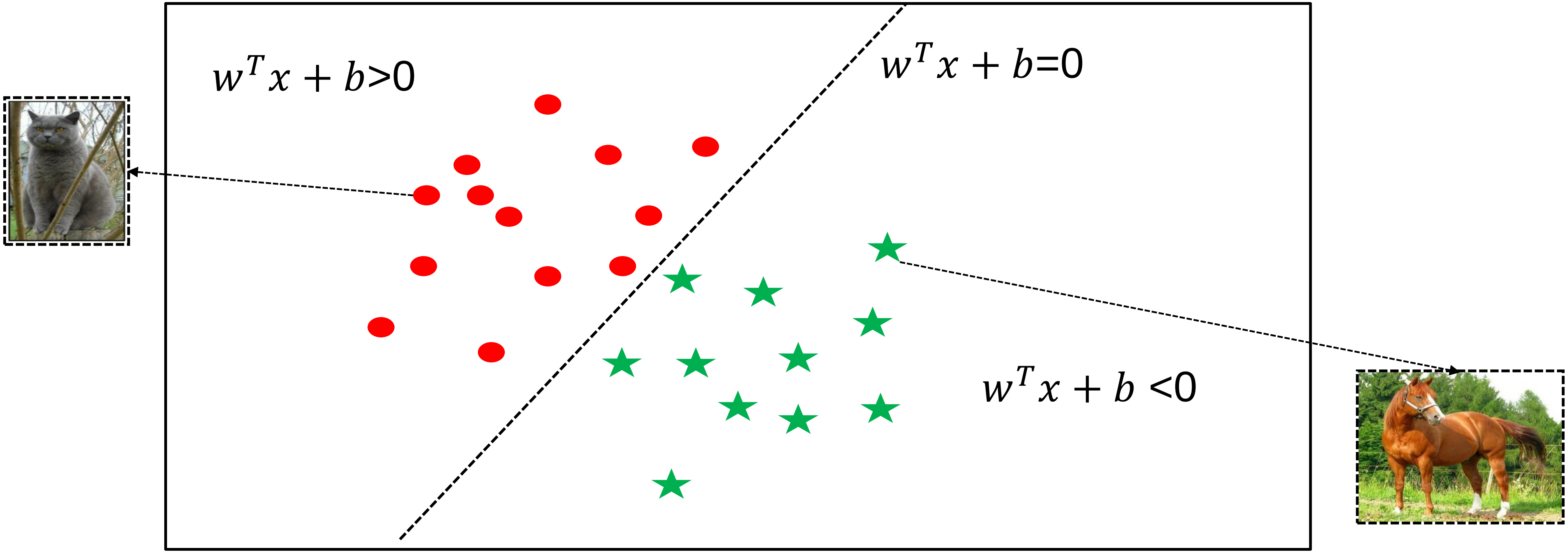
线性分类器



思路一：用超平面（线）将不同类样本分开

线性分类器

- 线性分类器



分类器参数(w, b)

$$y = f(x) = w^T x + b$$

线性分类器实例

输入图像特征$\boldsymbol{x}$

32×32×3

图像分类器

$$f(x) = w^T x + b$$

3072×1　　3072×1　　1×1

输出预测值

$$f(x) = 0.6$$

x　　w　　b

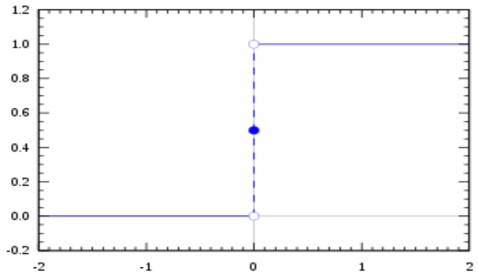
正例

$$\left(\begin{bmatrix}0.1\\0.2\\0.1\end{bmatrix} \cdot \begin{bmatrix}1.0\\3.0\\1.0\end{bmatrix} \right) + \begin{bmatrix}0.1\end{bmatrix} = 0.9$$

反例

$$\left(\begin{bmatrix}0.1\\-0.2\\0.2\end{bmatrix} \cdot \begin{bmatrix}1.0\\3.0\\1.0\end{bmatrix} \right) + \begin{bmatrix}0.1\end{bmatrix} = -0.2$$

阈值操作



"1"

"0"

如何训练模型

- 模型学习：如何从数据中训练得到最佳参数w, b

1.训练数据准备

$\boldsymbol{x}_1$ $y_1 = 1$

$\boldsymbol{x}_2$ $y_2 = 1$

$\boldsymbol{x}_3$ $y_3 = 1$

$\boldsymbol{x}_4$ $y_4 = -1$

$\boldsymbol{x}_5$ $y_5 = -1$

$\boldsymbol{x}_6$ $y_6 = -1$

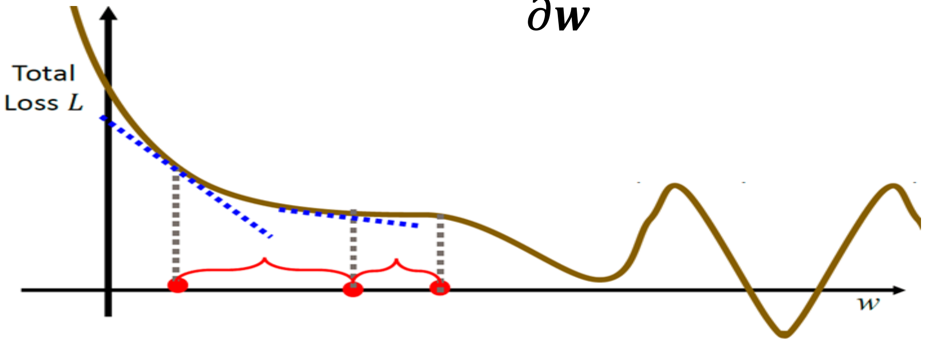
2.定义损失函数

$$-y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\begin{cases} f(x) < 0 & \text{若标签为正，预测为正} \\ f(x) > 0 & \text{若标签为正，预测为负} \end{cases}$$

损失函数：$L(\{\boldsymbol{x}_i\}; w, b) = -\sum_i y_i\,(\boldsymbol{w}^T\boldsymbol{x}_i + b)$

3.最小化损失函数

$$w^{(t+1)} = w^{(t)} - \alpha\frac{\partial L}{\partial \boldsymbol{w}}$$

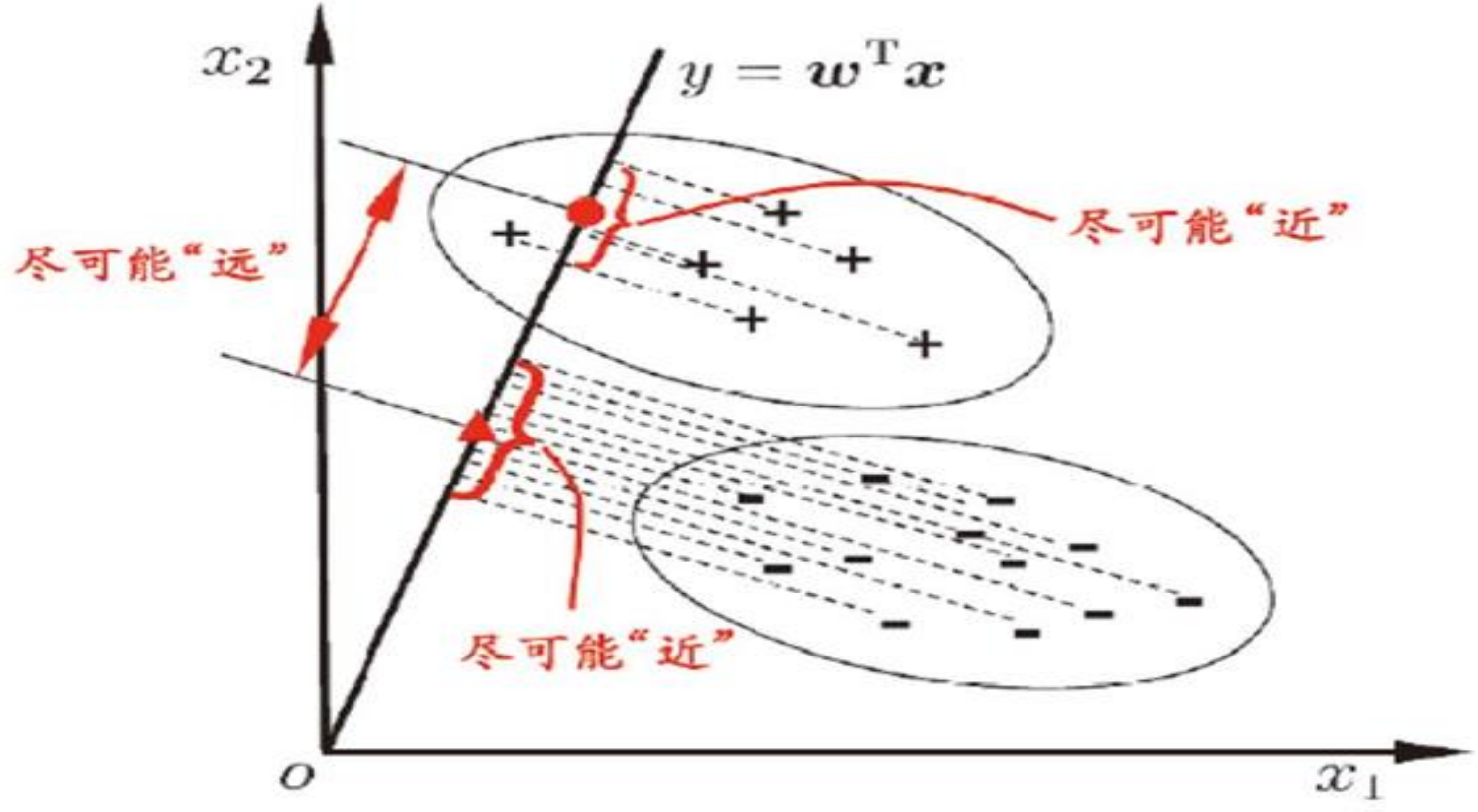


$$\frac{\partial L}{\partial b} = -\sum_i y_i$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = -\sum_i y_i\boldsymbol{x}_i^T$$

梯度下降

线性分类模型



思路二：映射到新的空间，新的空间里“同类近，异类远”

Linear Discriminant Analysis 线性判别分析

Linear Discriminant Analysis

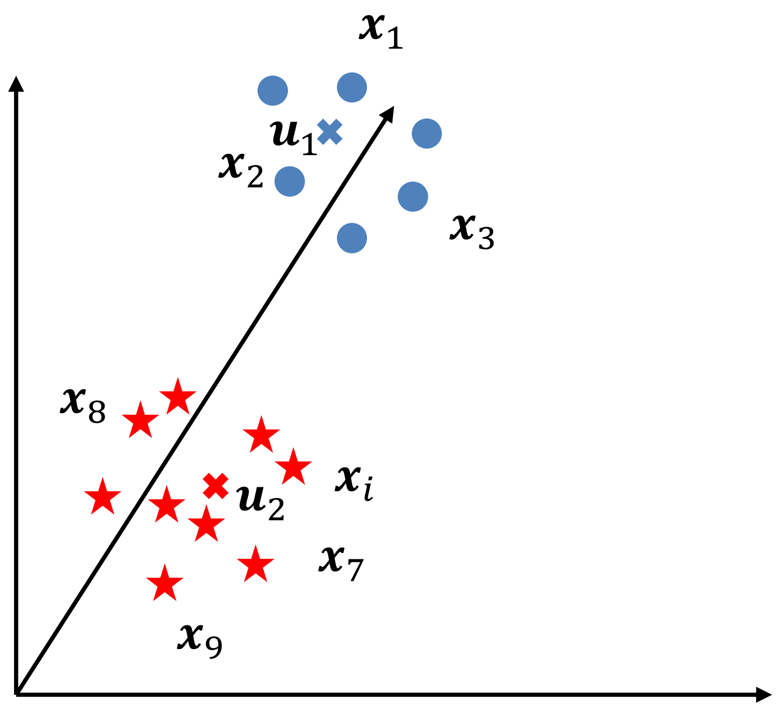
- 假设两类问题(Binary Classification)

$\boldsymbol{u}_1$： 表示第一类样本的均值向量

$$\boldsymbol{u}_1 = \frac{1}{N_1}\sum_{i \in C_1} \boldsymbol{x}_i$$

$\boldsymbol{u}_2$： 表示第二类样本的均值向量

$$\boldsymbol{u}_2 = \frac{1}{N_2}\sum_{i \in C_2} \boldsymbol{x}_i$$



数据集

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$$

- 表示第一类 C_1
- 表示第二类 C_2
- 第一类中心 $\boldsymbol{u}_1$
- 第二类中心 $\boldsymbol{u}_2$

Linear Discriminant Analysis

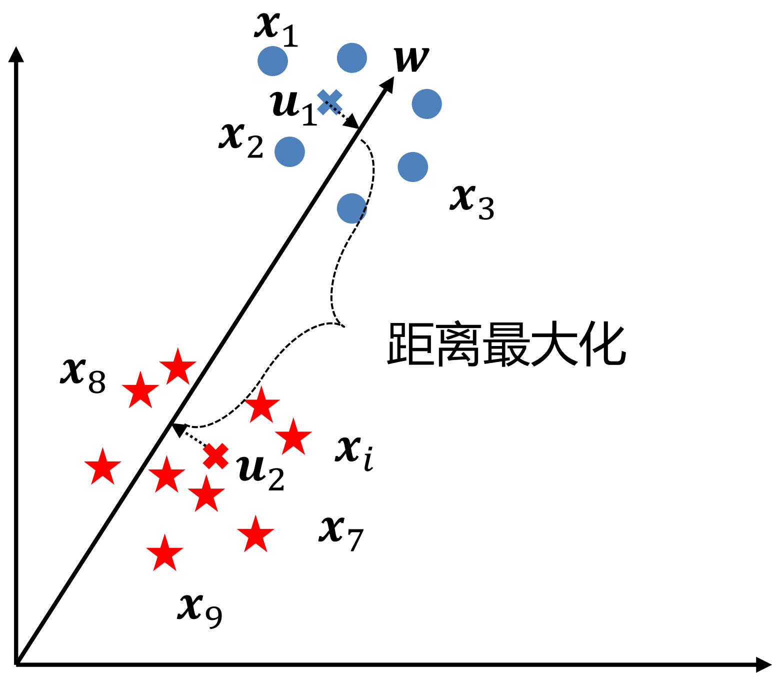
- 假设两类问题(Binary Classification)

类间距离 $d = |\boldsymbol{w}^T\boldsymbol{u}_1 - \boldsymbol{w}^T\boldsymbol{u}_2|$

距离平方

$$d^2 = (\boldsymbol{w}^T\boldsymbol{u}_1 - \boldsymbol{w}^T\boldsymbol{u}_2)(\boldsymbol{u}_1^T\boldsymbol{w} - \boldsymbol{u}_2^T\boldsymbol{w})$$

$$= \boldsymbol{w}^T(\boldsymbol{u}_1 - \boldsymbol{u}_2)(\boldsymbol{u}_1 - \boldsymbol{u}_2)^T\boldsymbol{w}$$



数据集

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$

- ● 表示第一类 C_1
- ★ 表示第二类 C_2
- ✖ 第一类中心 $\boldsymbol{u}_1$
- ✖ 第二类中心 $\boldsymbol{u}_2$

目标1：通过一个映射$\boldsymbol{w}$, 将映射到$\boldsymbol{w}$轴上的两中心点$\boldsymbol{w}^T\boldsymbol{u}_1$与$\boldsymbol{w}^T\boldsymbol{u}_2$之间距离最大

AI300学院

Linear Discriminant Analysis

- 假设两类问题(Binary Classification)

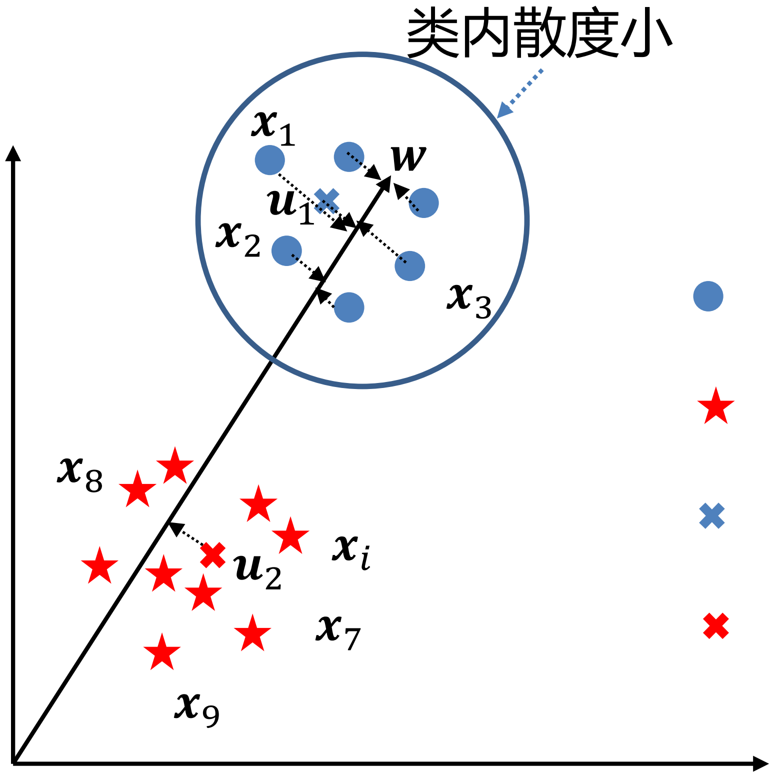
类内距离 $d_1 = \sum_{i\in C_1} |\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{u}_1|$

距离“平方”

$$d_1^2 = \sum_{i\in C_1} (\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T u_1)(\boldsymbol{x}_i^T\boldsymbol{w} - \boldsymbol{u}_1^T\boldsymbol{w})^T$$

$$= \boldsymbol{w}^T \underbrace{\sum_{i\in C_1} (\boldsymbol{x}_i - \boldsymbol{u}_1)(\boldsymbol{x}_i - \boldsymbol{u}_1)^T}_{\Sigma_1} \boldsymbol{w}$$

D×D维矩阵 Σ_1 第一类样本协方差矩阵



数据集

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$

● 表示第一类 C_1

★ 表示第二类 C_2

✖ 第一类中心 $\boldsymbol{u}_1$

✖ 第二类中心 $\boldsymbol{u}_2$

$$d_1^2 + d_2^2 = \boldsymbol{w}^T\Sigma_1\boldsymbol{w} + \boldsymbol{w}^T\Sigma_2\boldsymbol{w}$$

目标2：通过一个映射$\boldsymbol{w}$，每个类内数据$\boldsymbol{w}^T\boldsymbol{x}_i$与其均值$\boldsymbol{w}^T\boldsymbol{u}_1$之间距离最小

Linear Discriminant Analysis

- 假设两类问题(Binary Classification)

类内距离越小，同时类间聚类越大

$$\underbrace{d_w = \boldsymbol{w}^T\Sigma_1\boldsymbol{w} + \boldsymbol{w}^T\Sigma_2\boldsymbol{w} \downarrow \qquad d_b = \boldsymbol{w}^T(\boldsymbol{u}_1 - \boldsymbol{u}_2)}(\boldsymbol{u}_1 - \boldsymbol{u}_2)^T\boldsymbol{w} \uparrow$$

$$max\,\frac{d_b}{d_w}$$

终极目标：如何同时满足目标1和目标2

Linear Discriminant Analysis

- 假设两类问题(Binary Classification)

记： $d_w = \boldsymbol{w}^T \Sigma_1 \boldsymbol{w} + \boldsymbol{w}^T \Sigma_2 \boldsymbol{w} = \boldsymbol{w}^T (\Sigma_1 + \Sigma_2) \mathbf{w} = \mathbf{w}^{\mathrm{T}} \mathrm{S}_{\mathrm{w}} \mathbf{w}$

其中： $S_w = \Sigma_1 + \Sigma_2$ 称为类内散射矩阵

$$d_b = \boldsymbol{w}^T (\boldsymbol{u}_1 - \boldsymbol{u}_2)(\boldsymbol{u}_1 - \boldsymbol{u}_2)^T \boldsymbol{w} = \boldsymbol{w}^T S_b \boldsymbol{w}$$

其中： $S_b = (\boldsymbol{u}_1 - \boldsymbol{u}_2)(\boldsymbol{u}_1 - \boldsymbol{u}_2)^T$ 称为类间散射矩阵

$$\max \frac{d_b}{d_w} \Longrightarrow \max \frac{\boldsymbol{w}^T S_b \boldsymbol{w}}{\boldsymbol{w}^T S_w \boldsymbol{w}}$$

Linear Discriminant Analysis

- LDA的目标：最大化广义瑞利商 J(generalized Rayleigh quotient)

单值 $J(\boldsymbol{w}) = \dfrac{\boldsymbol{w}^T S_b \boldsymbol{w}}{\boldsymbol{w}^T S_w \boldsymbol{w}}$ $\boldsymbol{w}$成倍缩放不影响 J值
仅考虑方向

求偏导数，设为0，求解$\boldsymbol{w}$

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{(w^T S_b w)' w^T S_w w - w^T S_b w (w^T S_w w)'}{(w^T S_w w)^2}$$

$$= \frac{2(w^T S_w w) S_b w - 2(w^T S_b w) S_w w}{(w^T S_w w)^2}$$

$$\Rightarrow \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \Rightarrow (w^T S_w w) S_b w - (w^T S_b w) S_w w = 0$$

$$\Rightarrow S_b w - J S_w w = 0$$

$$\Rightarrow S_b w = J S_w w$$

$$\Rightarrow \boxed{S_w^{-1} S_b w = J w}$$

变成特征值分解问题！

Linear Discriminant Analysis

- LDA的解

$$\boxed{S_w^{-1}S_B\boldsymbol{w} = J\boldsymbol{w}}$$

特征值分解矩阵$S_w^{-1}S_B$　　D×D 方阵

最优的$\boldsymbol{w}$即为最大的特征值所对应的那个特征向量!

$$A\boldsymbol{e} = \lambda\boldsymbol{e}$$

Why?　　$S_w^{-1}S_B\boldsymbol{w}_m = J_m\boldsymbol{w}_m$

$$S_w^{-1}S_B = [e_1 \quad \cdots \quad e_D]\begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & \lambda_D \end{bmatrix}\begin{bmatrix} e_1^T \\ \vdots \\ e_D^T \end{bmatrix}$$

$$J(\boldsymbol{w}_m) = \frac{\boldsymbol{w}_m^T S_B \boldsymbol{w}_m}{\boldsymbol{w}_m^T S_w \boldsymbol{w}_m} = \frac{\boldsymbol{w}_m^T(\boldsymbol{S}_w J_m \boldsymbol{w}_m)}{\boldsymbol{w}_m^T S_w \boldsymbol{w}_m} = \frac{J_m \boldsymbol{w}_m^T \boldsymbol{S}_w \boldsymbol{w}_m}{\boldsymbol{w}_m^T S_w \boldsymbol{w}_m} = J_m \dashrightarrow \textbf{最大化!}$$

$$\text{单值} \dashleftarrow J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_w \boldsymbol{w}}$$

Linear Discriminant Analysis

- 使用LDA进行分类
  - -获得最优的映射方向w
  - -将所有原始数据均映射至一维（1D）

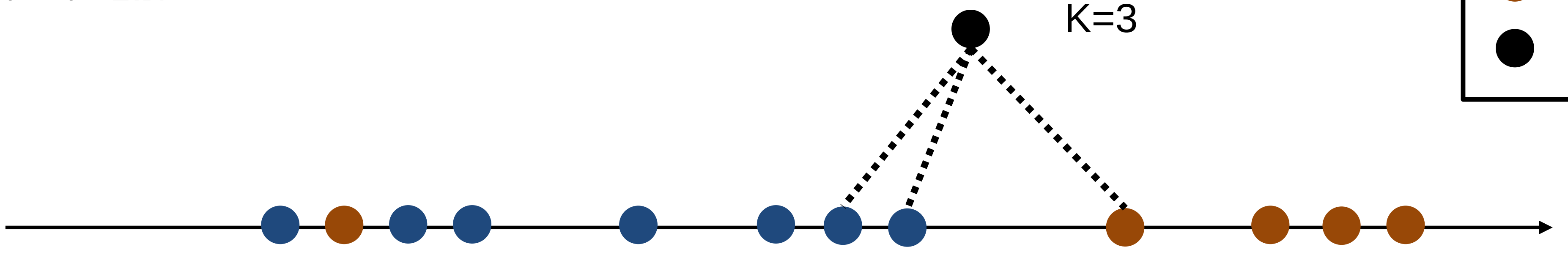
数据集 $\{(\boldsymbol{x}_i, l_i)\}_{i=1:N}$ $y_1 = \boldsymbol{w}^T\boldsymbol{x}_1$ $y_2 = \boldsymbol{w}^T\boldsymbol{x}_2$ $y_3 = \boldsymbol{w}^T\boldsymbol{x}_3$ ………… $y_N = \boldsymbol{w}^T\boldsymbol{x}_N$

$$\downarrow$$

$\{(y_i, l_i)\}_{i=1:N}$

使用K近邻分类器

A类样本
B类样本
待分类样本



A类：2次；B类：1次 → 判为A类

Linear Discriminant Analysis

- 若原数据$\boldsymbol{x}$维度较高，映射到一维（1D）恐怕信息损失较多，不利于分类
- 那就保留更多的维度！

$$\boxed{S_W^{-1} S_B \boldsymbol{w} = J\boldsymbol{w}}$$

不仅保留最大特征值对应的特征向量$\boldsymbol{w}_m$，还可以保留“次大”的向量$\boldsymbol{w}_{m-1}$,$\boldsymbol{w}_{m-2}$等

假设保留2D维度

数据集 $\{(\boldsymbol{x}_i, l_i)\}_{i=1:N}$

$$y_1^1 = \boldsymbol{w}_m^T \boldsymbol{x}_1 \quad y_2^1 = \boldsymbol{w}_m^T \boldsymbol{x}_2 \quad \cdots\cdots\cdots\cdots \quad y_N^1 = \boldsymbol{w}_m^T \boldsymbol{x}_2$$

$$y_1^2 = \boldsymbol{w}_{m-1}^T \boldsymbol{x}_1 \quad y_2^2 = \boldsymbol{w}_{m-1}^T \boldsymbol{x}_2 \quad y_N^2 = \boldsymbol{w}_{m-1}^T \boldsymbol{x}_2$$

$$\Downarrow$$

$\{(\boldsymbol{y}_i, l_i)\}_{i=1:N}$

现在$\boldsymbol{y}_i$是一个低维度向量

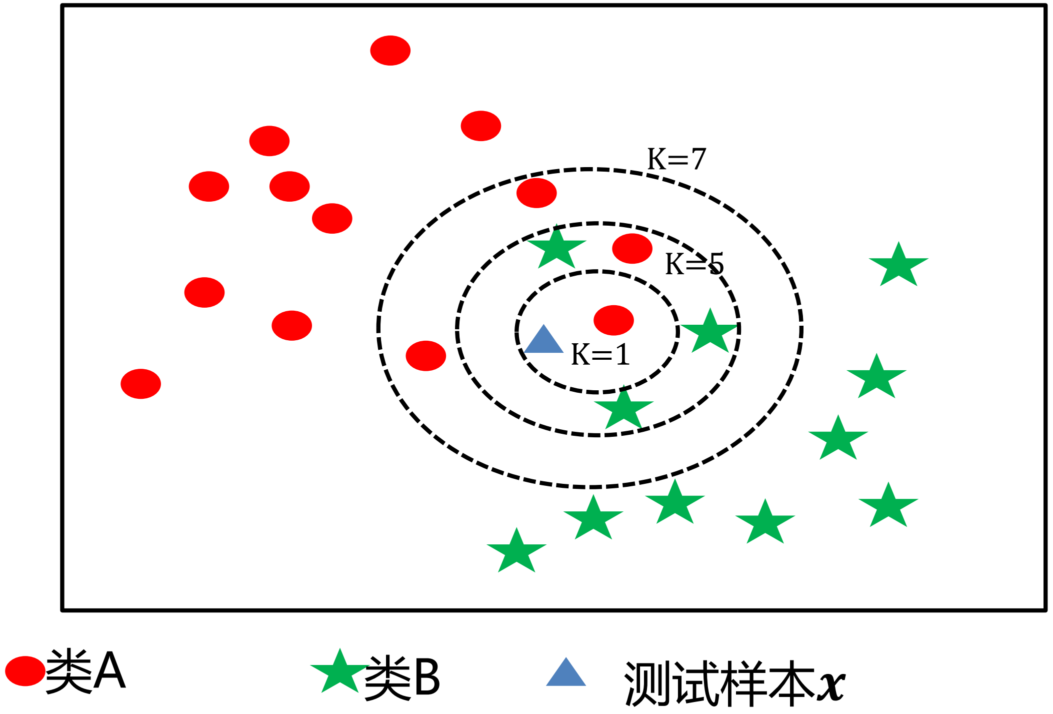
貌似：最多可以保留到原来的维度！

Linear Discriminant Analysis

- 假设降到2维后的K近邻分类
  - 根据距离度量公式（如欧式距离），寻找它的k个近邻
  - 根据它的k个近邻中每个类别出现的频率，确定它的类别

本例中

1. *当K=1时，x的近邻样本中类A出现1次，类B出现0次，因此预测为类A*
2. *当K=5时，x的近邻样本中类A出现2次，类B出现3次，因此预测为类B*
3. *当K=7呢?*



多类问题LDA

- 与两类问题LDA的异同

	两类问题	多类问题
类内中心	$\boldsymbol{u}_1 \quad \boldsymbol{u}_2$	$\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \boldsymbol{u}_3 \quad \boldsymbol{u}_4$ ……….
类间散射矩阵	$S_b = (\boldsymbol{u}_1 - \boldsymbol{u}_2)(\boldsymbol{u}_1 - \boldsymbol{u}_2)^T$	?
类内散射矩阵	$S_w = \sum_{i \in C_1} (\boldsymbol{x}_i - \boldsymbol{u}_1)(\boldsymbol{x}_i - \boldsymbol{u}_1)^T + \sum_{i \in C_2} (\boldsymbol{x}_i - \boldsymbol{u}_2)(\boldsymbol{x}_i - \boldsymbol{u}_2)^T$	$S_w = \sum_{j=1:M} \sum_{i \in C_j} (\boldsymbol{x}_i - \boldsymbol{u}_j)(\boldsymbol{x}_i - \boldsymbol{u}_j)^T$

多类问题LDA

- 多类问题LDA

定义全局散度矩阵 $$S_T = \sum_{i=1}^{N}(\boldsymbol{x}_i-\boldsymbol{u})(\boldsymbol{x}_i-\boldsymbol{u})^T$$ $\boldsymbol{u}$ 为所有数据样本的中心

S_T是什么?

$$\begin{aligned}
S_T &= \sum_{i=1}^{N}(\boldsymbol{x}_i-\boldsymbol{u})(\boldsymbol{x}_i-\boldsymbol{u})^T = \sum_{j=1}^{M}\sum_{i\in C_j}^{N_j}\left(\boldsymbol{x}_i-\boldsymbol{u}_j+\boldsymbol{u}_j-\boldsymbol{u}\right)\left(\boldsymbol{x}_i-\boldsymbol{u}_j+\boldsymbol{u}_j-\boldsymbol{u}\right)^T \\
&= \sum_{j=1}^{M}\sum_{i\in C_j}^{N_j}(\boldsymbol{x}_i-\boldsymbol{u}_j)(\boldsymbol{x}_i-\boldsymbol{u}_j)^T + \sum_{j=1}^{M}\sum_{i\in C_j}^{N_j}(\boldsymbol{u}_j-\boldsymbol{u})(\boldsymbol{u}_j-\boldsymbol{u})^T + 2\sum_{j=1}^{M}\sum_{i\in C_j}^{N_j}(\boldsymbol{x}_i-\boldsymbol{u}_j)(\boldsymbol{u}_j-\boldsymbol{u})^T \\
&= \underbrace{\sum_{j=1}^{M} S_w^j}_{S_w} + \underbrace{\sum_{j=1}^{M} N_j(\boldsymbol{u}_j-\boldsymbol{u})(\boldsymbol{u}_j-\boldsymbol{u})^T}_{S_b} + 0
\end{aligned}$$

$$\Longrightarrow \boxed{S_T = S_w + S_b}$$

多类问题LDA

- 与两类问题同样的解法

$$\boxed{S_w^{-1}S_b\boldsymbol{w} = J\boldsymbol{w}}$$

特征值分解！

若类的个数是M，$\boldsymbol{w}$顶多有$M-1$个有意义!

因为 $S_b = \sum_{j=1}^{M} N_j(\boldsymbol{u}_j - \boldsymbol{u})(\boldsymbol{u}_j - \boldsymbol{u})^T$ 秩顶多$M-1$，所以S_wS_b顶多有$M-1$个大于0的特征值！

所以顶多能降到$M-1$维!

所以对于两类问题，LDA顶多可以降到1维！

LDA降维实战

用scikit-learn进行LDA降维

【对scikit-learn中LDA类概述】

在scikit-learn中， LDA类是sklearn.discriminant_analysis.LinearDiscriminantAnalysis。

参数总结:

- solver： 求LDA超平面特征矩阵使用的方法;
- shrinkage：正则化参数，默认是None;
- priors： 类别权重，一般用于分类，降维时不考虑;
- n_components：进行LDA降维时降到的维数。

LDA降维实战

用scikit-learn进行LDA降维

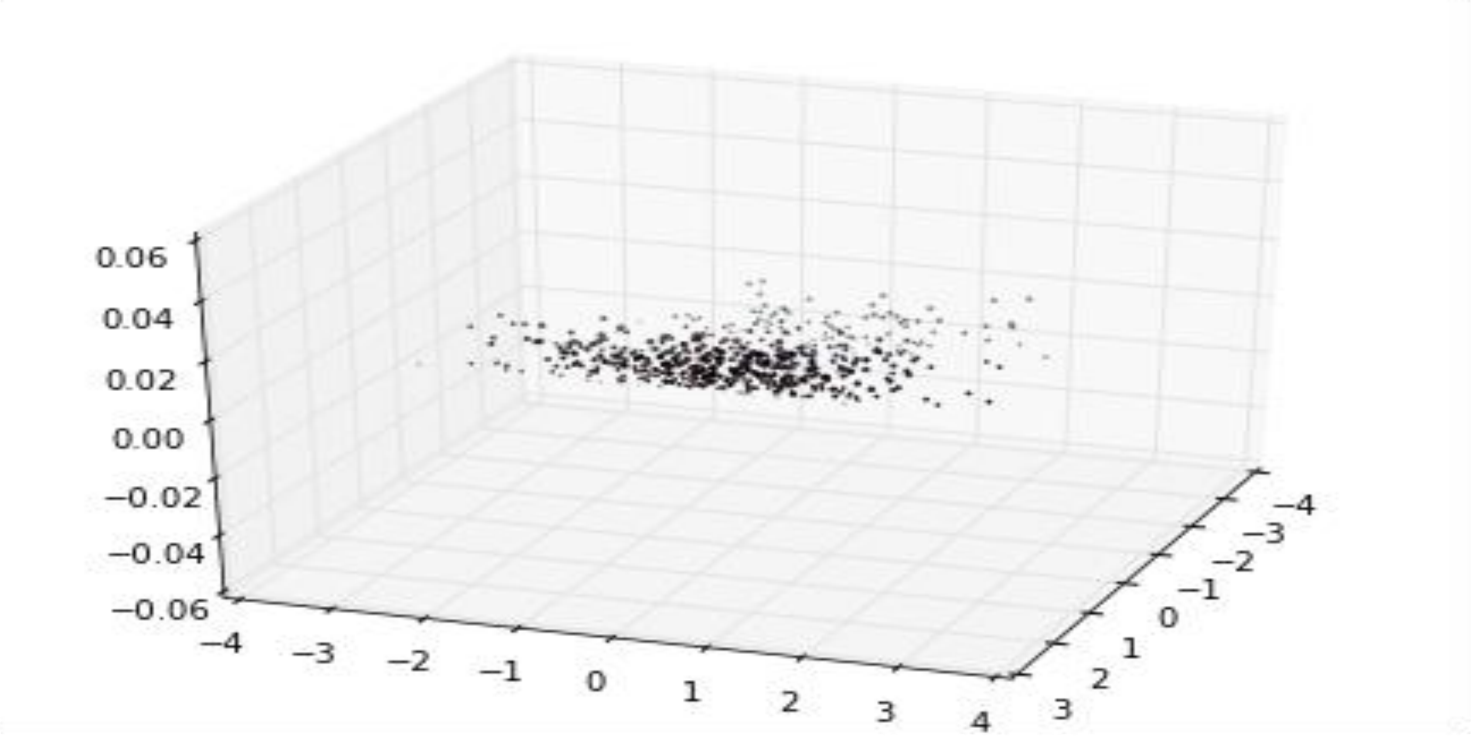
【生成三维数据特征】

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline
from sklearn.datasets.samples_generator import make_classification
X, y = make_classification(n_samples=1000, n_features=3, n_redundant=0, n_classes=3,
        n_informative=2,n_clusters_per_class=1,class_sep =0.5, random_state =10)
fig = plt.figure()
ax = Axes3D(fig, rect=[0, 0, 1, 1], elev=30, azim=20)
plt.scatter(X[:, 0], X[:, 1], X[:, 2],marker='o',c=y)
```

LDA降维实战

用scikit-learn进行LDA降维

【三维数据分布】



LDA降维实战

用scikit-learn进行LDA降维

【PCA降维到二维】

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(X)
print pca.explained_variance_ratio_
print pca.explained_variance_
X_new = pca.transform(X)
plt.scatter(X_new[:, 0], X_new[:, 1],marker='o',c=y)
plt.show()
```

PCA找到的两个主成分方差比和方差
[0.43377069 0.3716351]
[1.20962365 1.03635081]

样本特征和类别的信息关联几乎完全丢失!

AI300学院

LDA降维实战

用scikit-learn进行LDA降维

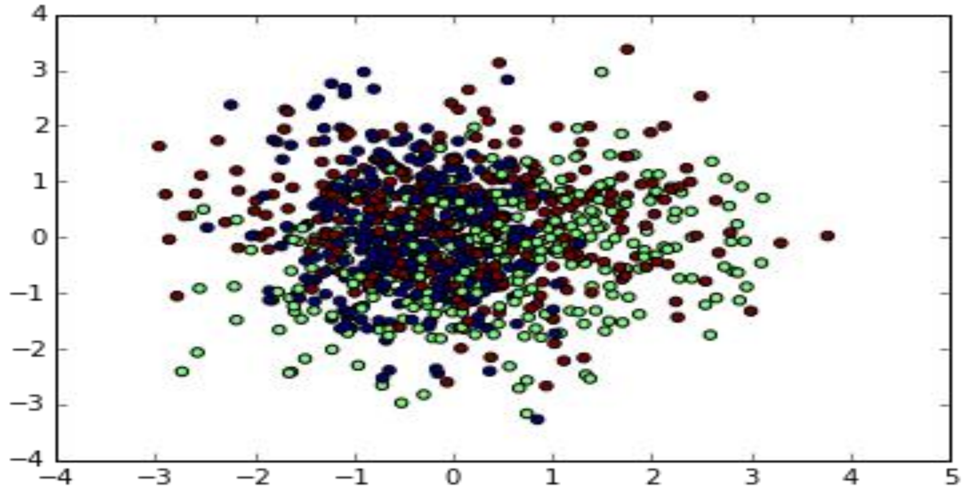
【LDA降维到二维】

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis(n_components=2)
lda.fit(X,y)
X_new = lda.transform(X)
plt.scatter(X_new[:, 0], X_new[:, 1],marker='o',c=y)
plt.show()
```

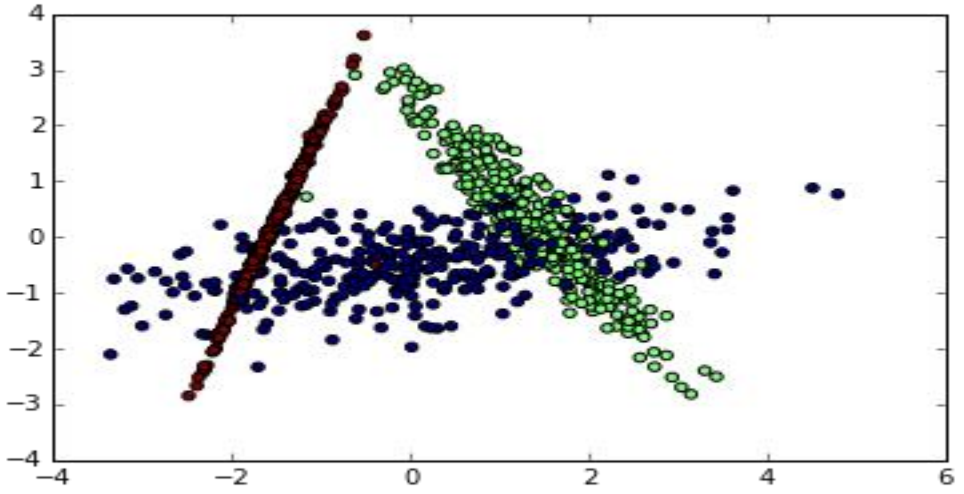
降维后样本特征和类别信息之间的关系得以保留。

LDA降维实战

用scikit-learn进行LDA降维



(a) PCA降维结果



(b) LDA降维结果

Thank You

AI300学院