第9章 机器学习sklearn库

计算机系：王学军

邮箱：wangxuejun@stdu.edu.cn

2022年4月6日

CONTENT

01 Sklearn库简介

02 分类模型评估

03 几类常见机器学习算法

04 总结

Sklearn库简介

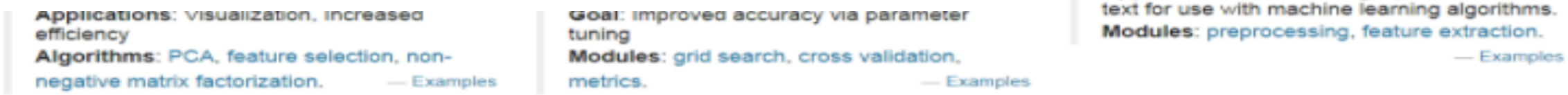
Sklearn库简介

Scikit-learn（以前称为scikits.learn，也称为sklearn）是针对Python 编程语言的免费软件机器学习库，07年由David Cournapeau发起的Google Summer of Code项目。它的名称源于它是“ SciKit”（SciPy工具包）的概念，它是SciPy的独立开发和分布式第三方扩展，建立在Numpy和matplolib库的基础上。利用这几大模块的优势，可以大大的提高机器学习的效率。sklearn拥有着完善的文档，上手容易，具有着丰富的API，封装了大量的**机器学习算法**并内置大量**数据集**，节省获取和整理数据时间。

官方文档：https://scikit-learn.org/stable/　中文文档：

由图中，可以看到库的算法主要有四类：分类，回归，聚类，降维。其中：

- 常用的回归：线性、决策树、SVM、KNN；集成回归：随机森林、Adaboost、GradientBoosting、Bagging、ExtraTrees
- 常用的分类：线性、决策树、SVM、KNN，朴素贝叶斯；集成分类：随机森林、Adaboost、GradientBoosting、Bagging、ExtraTrees
- 常用聚类：k均值（K-means）、层次聚类（Hierarchical clustering）
- 常用降维：LinearDiscriminantAnalysis、PCA



Sklearn库简介

```python
# 引入数据集，sklearn包含众多数据集
from sklearn import datasets
# 将数据分为测试集和训练集
from sklearn.model_selection import train_test_split
# 利用邻近点方式训练数据
from sklearn.neighbors import KNeighborsClassifier

# 引入数据,本次导入鸢尾花数据，iris数据包含4个特征变量
iris = datasets.load_iris()
# 特征变量
iris_X = iris.data
# print(iris_X)
print('特征变量的长度',len(iris_X))
# 目标值
iris_y = iris.target
print('鸢尾花的目标值',iris_y)
# 利用train_test_split进行训练集和测试机进行分开，test_size占30%
X_train,X_test,y_train,y_test=train_test_split(iris_X,iris_y,test_size=0.3)
# 我们看到训练数据的特征值分为3类
# print(y_train)
'''
[1 1 0 2 0 0 0 2 2 2 1 0 2 0 2 1 0 1 0 2 0 1 0 0 2 1 2 0 0 1 0 0 1 0 0 0 0
 2 2 2 1 1 1 2 0 2 0 1 1 1 1 2 2 1 2 2 2 0 2 2 2 0 1 0 1 0 0 1 2 2 2 1 1 1
 2 0 0 1 0 2 1 2 0 1 2 2 2 1 2 1 0 0 1 0 0 1 1 1 0 2 1 1 0 2 2]
 '''
# 训练数据
# 引入训练方法
knn = KNeighborsClassifier()
# 进行填充测试数据进行训练
knn.fit(X_train,y_train)
```

```python
params = knn.get_params()
print(params)
'''
{'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski',
 'metric_params': None, 'n_jobs': None, 'n_neighbors': 5,
 'p': 2, 'weights': 'uniform'}

'''

score = knn.score(X_test,y_test)
print("预测得分为：%s"%score)
'''
预测得分为：0.9555555555555556
[1 2 1 1 2 2 1 0 0 0 0 1 2 0 1 0 2 0 0 0 2 2 0 2 2 2 2 1 2 2 2 1 2 2 1 2 1 2 0
 2 1 2 1 1 0 2 1]
[1 2 1 1 2 2 1 0 0 0 0 1 2 0 1 0 2 0 0 0 1 2 0 2 2 2 2 1 1 2 2 1 2 2 1 2 1 2 0
 2 1 2 1 1 0 2 1]
'''
```

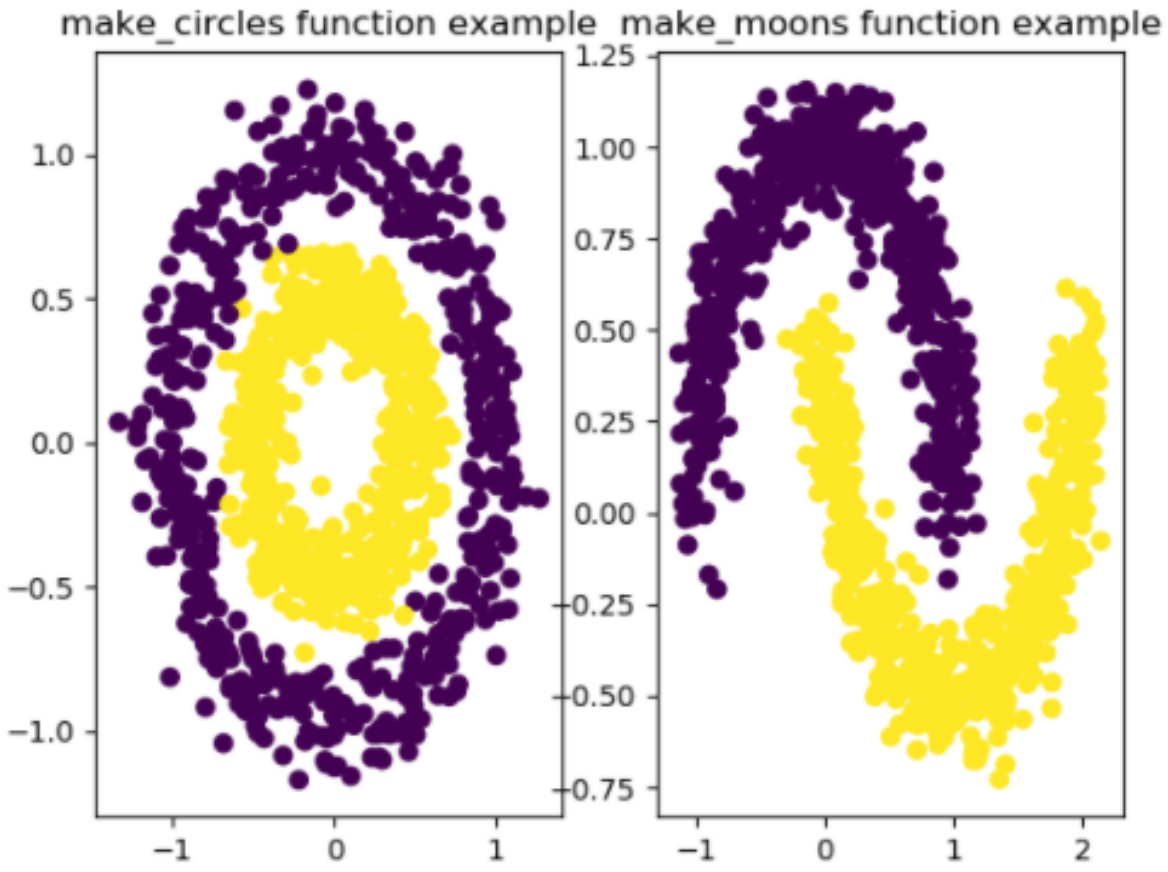
Sklearn库简介

	数据集名称	调用方式	适用算法	数据规模
小数据集	波士顿房价数据集	load_boston()	回归	506*13
	鸢尾花数据集	load_iris()	分类	150*4
	糖尿病数据集	load_diabetes()	回归	442*10
	手写数字数据集	load_digits()	分类	5620*64
大数据集	Olivetti脸部图像数据集	fetch_olivetti_faces()	降维	400*64*64
	新闻分类数据集	fetch_20newsgroups()	分类	-
	带标签的人脸数据集	fetch_lfw_people()	分类；降维	-
	路透社新闻语料数据集	fetch_rcv1()	分类	804414*47236
注：小数据集可以直接使用，大数据集在第一次使用的时会自动下载				

Sklearn库简介

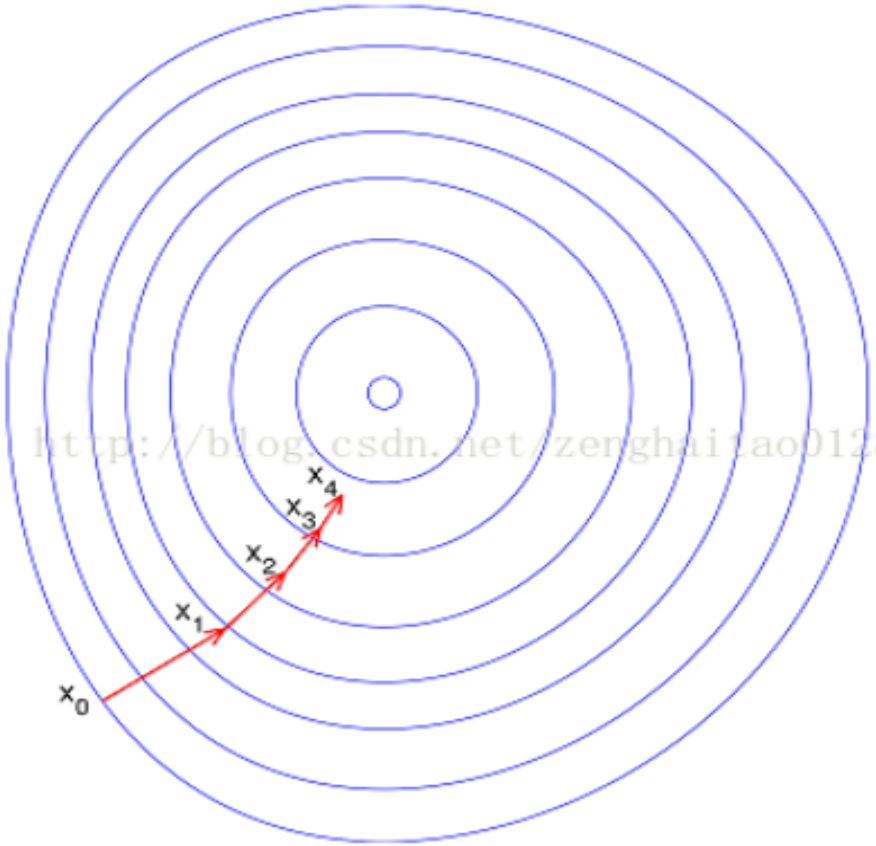
除了可

```
from sklearn.datasets import make_blobs
from matplotlib import pyplot

data,labe

# 绘制样本
pyplot.sc
pyplot.sh
```



Sklearn库简介

--如果进行归一化以后，目标函数会呈现比较"圆"，这样训练速度大大加快，少走很多弯路。



RobustScaler()：$x^{*}=\frac{x-median}{IQR}$ median是样本的中位数，IQR是样本的四分位距(第三四分位数与第一四分位数的差距)

优点：数据包含许多异常值时，可以有效剔除，使数据更为鲁邦。

Normalizer()：$x^{*}=\frac{x}{||x||_p}$ 优点：防止过拟合，加速算法收敛。

Sklearn库简介

数据集拆分：在得到训练数据集时，通常会把训练数据进一步拆分成训练集和测试集，从而验证算法有效性。

```
X_train,X_test, y_train, y_test =train_test_split(train_data,train_target,test_size=0.4,
random_state=0,shuffle=True)
```

- train_data：所要划分的样本特征集
- train_target：所要划分的样本结果
- test_size：样本占比，如果是整数的话就是样本的数量
- random_state：是随机数的种子。
- 随机数种子：其实就是该组随机数的编号，在需要重复试验的时候，保证得到一组一样的随机数。比如你每次都填1，其他参数一样的情况下你得到的随机数组是一样的。但填0或不填，每次都会不一样。
- shuffle ：是否在分割之前对数据进行洗牌

Sklearn库简介

4.4 决策树DT

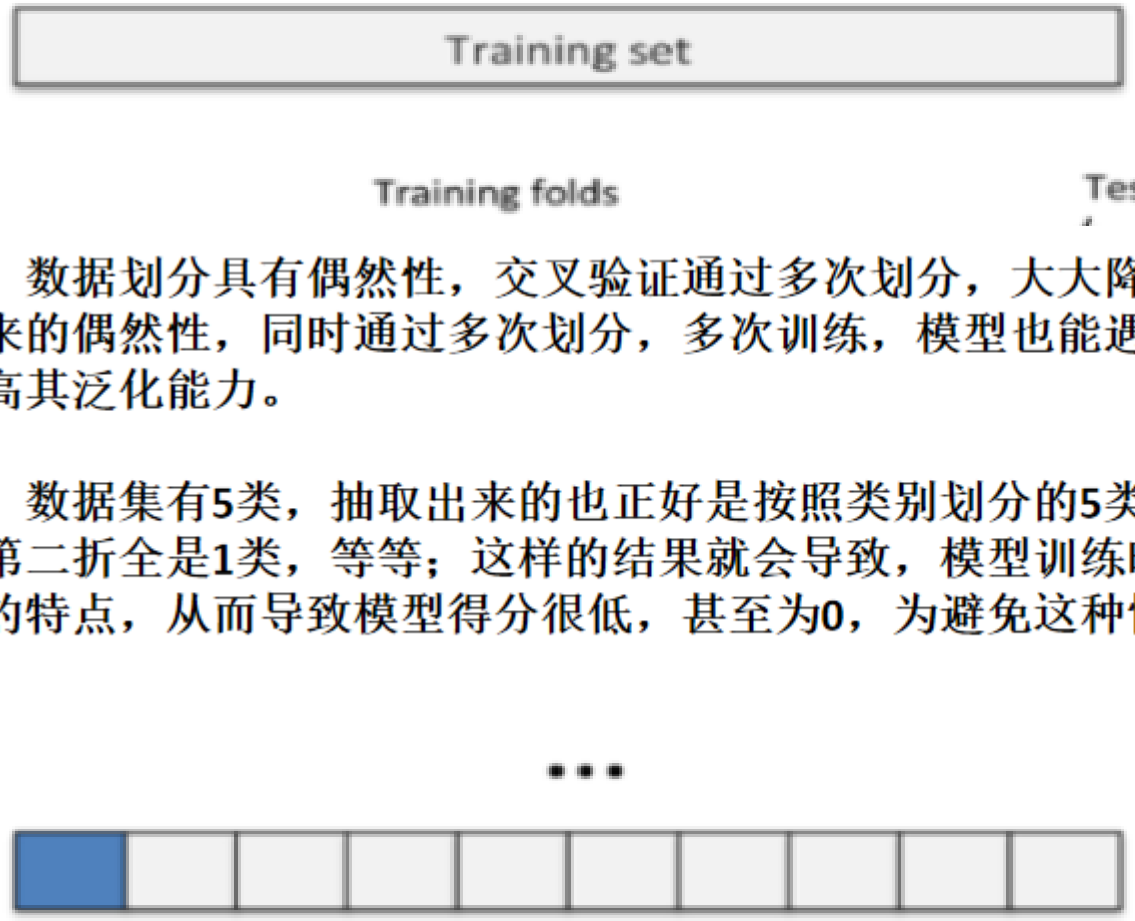
4.7 多层感知器（神经网络）

```
from sklearn.neural_network import MLPClassifier
# 定义多层感知机分类算法
model = MLPClassifier(activation='relu', solver='adam', alpha=0.0001)
"""参数
---
    hidden_layer_sizes: 元祖
    activation: 激活函数
    solver : 优化算法{‘lbfgs’, ‘sgd’, ‘adam’}
    alpha: L2惩罚(正则化项)参数。
"""
```

```
    max_features: 寻找最优分割点时的最大特征数
    max_leaf_nodes: 优先增长到最大叶子节点数
    min_impurity_decrease: 如果这种分离导致杂质的减少大于或等于这个值，则节点将被拆分。
"""
"""
```

Sklearn库简介

模型评估与选择：



优势：数据划分具有偶然性，交叉验证通过多次划分，大大降低了这种由一次随机划分带来的偶然性，同时通过多次划分，多次训练，模型也能遇到各种各样的数据，从而提高其泛化能力。

缺点：数据集有5类，抽取出来的也正好是按照类别划分的5类，也就是说第一折全是0类，第二折全是1类，等等；这样的结果就会导致，模型训练时。没有学习到测试集中数据的特点，从而导致模型得分很低，甚至为0，为避免这种情况，出现改进交叉验证方式。

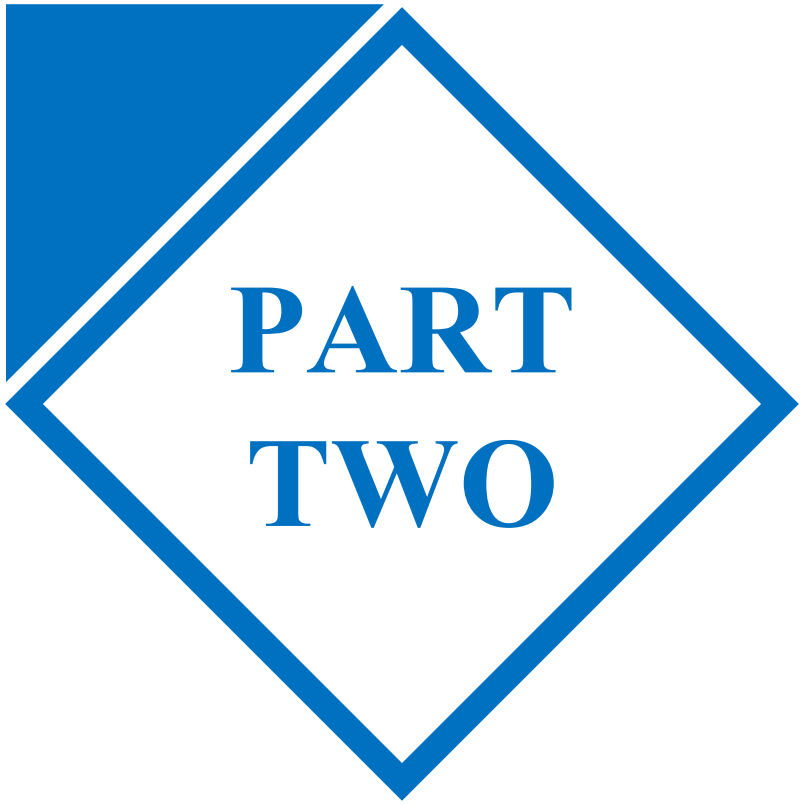
$$\frac{1}{10}\sum_{i=1}^{10} E_i$$

Sklearn库简介

模型保存：

6.2 sklearn自带方法joblib

```
1  from sklearn.externals import joblib
2
3  # 保存模型
4  joblib.dump(model, 'model.pickle')
5
6  #载入模型
7  model = joblib.load('model.pickle')
```



分类模型评估

分类模型评估

- 在分类任务下，预测结果(Predicted Condition)与正确标记(True Condition)之间存在四种不同的组合，构成混淆矩阵(适用于多分类)，混淆矩阵是针对机器学习分类问题的性能度量。

预测结果

真实结果	正例	假例
正例	真正例TP	伪反例FN
假例	伪正例FP	真反例TN

TP：预测是是肯定的，并且是事实。

FP：预测是肯定的，但它是错误的。

TN：预测是负面的，并且是事实。

FN：预测是负面的，但它是错误的。

分类模型评估

例子：

y	y pred	output for threshold 0.6	Recall	Precision	Accuracy
0	0.5	0	1/2	2/3	4/7
1	0.9	1			
0	0.7	1			
1	0.7	1			
1	0.3	0			
0	0.4	0			
1	0.5	0			

在所有

精准率（Precision）又叫查准率，预测为正样本的结果中，我们有多少把握可以预测正确。

召回率（Recall）又叫查全率，在实际为正的样本中被预测为正样本的概率。

精准率和召回率的分子是相同，都是TP，但分母是不同的，一个是（TP+FP），一个是（TP+FN），两者成反比例关系。

样本不

则为正

样本即

找到二者的平衡点，引入F1值，F1既考虑查准率也考虑查全率，

F1 = （2*P*R）/(P+R)　　$\frac{2}{F_1}=\frac{1}{P}+\frac{1}{R}$

机器学习算法

k近邻算法及实例

电影名称	打斗镜头	接吻镜头	电影类型
California Man	3	104	爱情片

电影名称	与未知电影的距离
California Man	20.5
He's not Really into dues	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9

用“邻居”来推断未知数据类别

定义：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。

k近邻算法及实例

- 优点：
  - 简单，易于理解，易于实现，无需估计参数，无需训练

- 缺点：
  - 贪婪算法，对测试样本分类时的计算量大，内存开销大
  - 必须指定K值，K值选择不当则分类精度不能保证

- 使用场景：小数据场景，几千～几万样本，具体场景具体业务去测试

k近邻算法作业

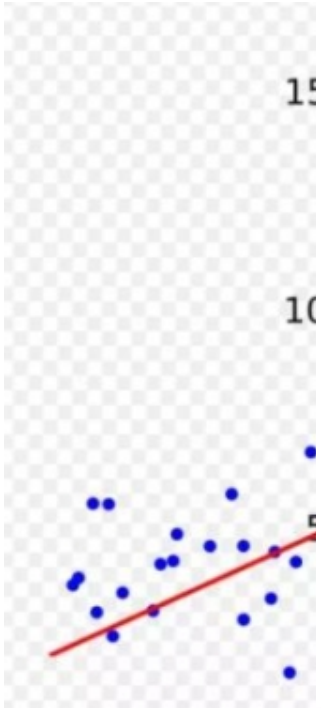
- 通过k-近邻算法对生物物种进行分类——鸢尾花(load_iris)

该虹膜数据集包含150行数据，包括来自每个的三个相关鸢尾种类50个样品： *山鸢尾*，*虹膜锦葵*，和*变色鸢尾*。



从左到右， *Iris setosa*（由 Radomil，CC BY-SA 3.0）， *Iris versicolor*（由 Dlanglois，CC BY-SA 3.0）和*Iris virginica*（由Frank Mayfield，CC BY-SA 2.0））。

logic回归



$$P(y=1|X,w)=\frac{1}{1+e^{-h(X)}}$$

$$P(y=0|X,w)=\frac{e^{-h(X)}}{1+e^{-h(X)}}$$

$$h(X|w_k)=w_{k0}+w_{k1}x_1+w_{k2}x_2\cdots w_{kn}x_n$$

$$P(y=k|X,w)=\frac{e^{h(X|w_k)}}{1+\sum_{i=1}^{K-1}e^{h(X|w_i)}} \quad \text{式1}$$

$$k=1,2,3\cdots K-1$$

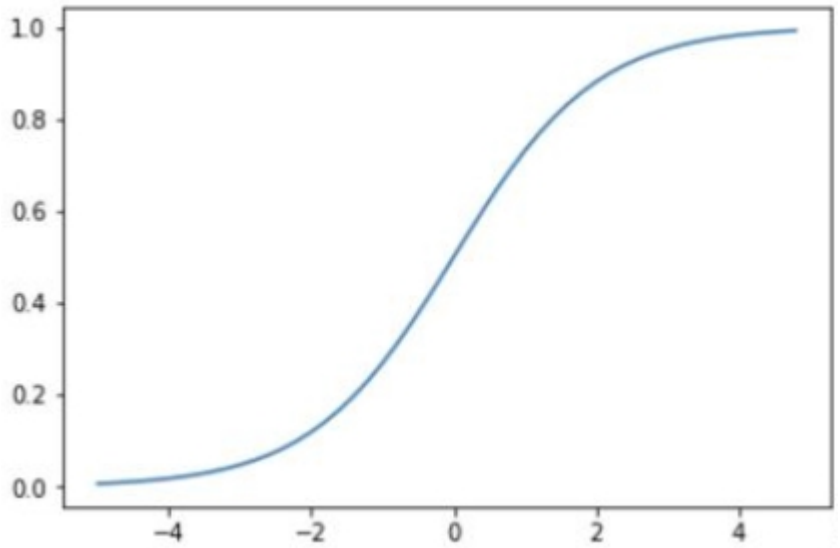
$$P(y=K|X,w)=\frac{1}{1+\sum_{i=1}^{K-1}e^{h(X|w_i)}} \quad \text{式2}$$

类。若想
散值。

于0为负
质不好。

数

视为后验



$$h(x)=\frac{1}{1+e^{-(w^Tx+b)}}$$

p(y=1|x) = h(x)；　p(y=0|x) = 1−h(x)

$$P(X)=h(x)^y\left(1-h(x)\right)^{(1-y)}$$

$$J(\theta)=\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)}))+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

logic回归

1. Logistic回归在线性回归的实数输出范围加上sigmoid函数，将输出值收敛在0~1之间。其目标函数也因此从平方损失变为对数损失函数。

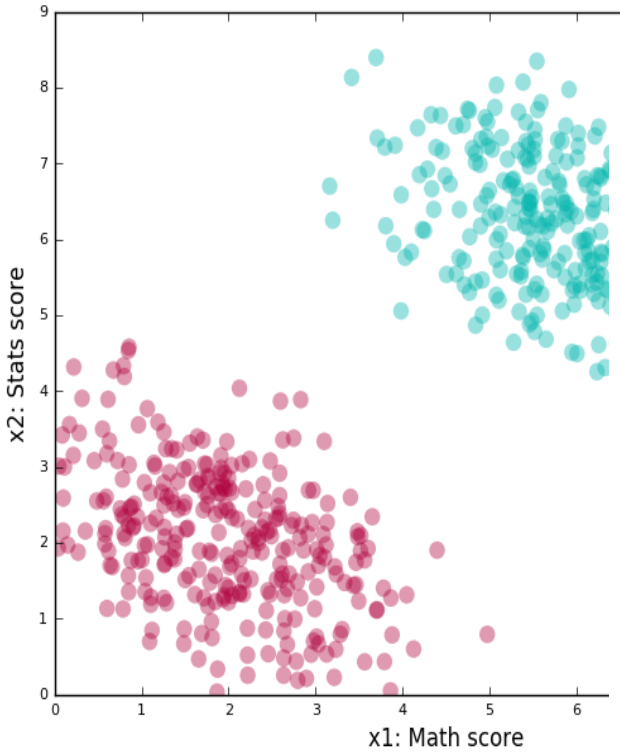
2. 逻辑回归和线性回归都是广义的线性回归，线性回归是使用最小二乘法优化目标函数，而逻辑回归是使用梯度下降或者拟牛顿法。

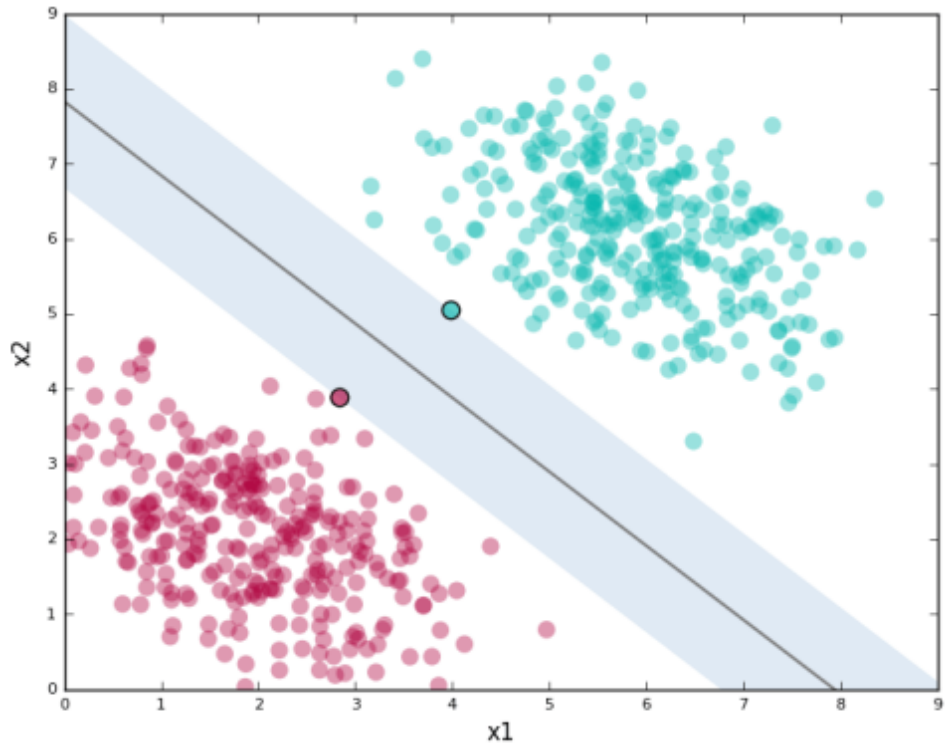
3. 线性回归在整个实数域范围内进行预测，敏感度一致，而分类范围需要在[0,1]。逻辑回归是一种减少预测范围，将预测值限定为[0,1]间的一种回归模型。因而对于二分类问题，逻辑回归的鲁棒性更好。

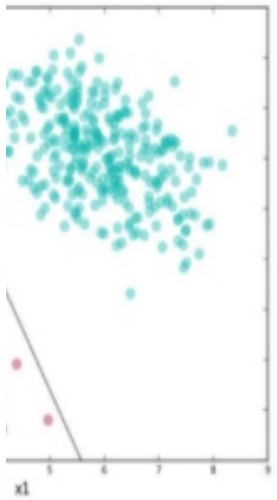
4. 逻辑回归是以线性回归为理论支持的，但线性回归模型无法做到sigmoid的非线性形式。Sigmoid可以轻松处理0/1分类问题。

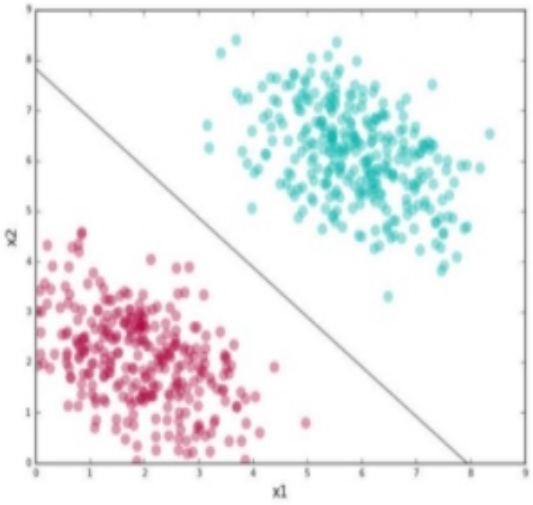
作业：通过逻辑回归算法对生物物种进行分类——鸢尾花(load_iris)

SVM算法









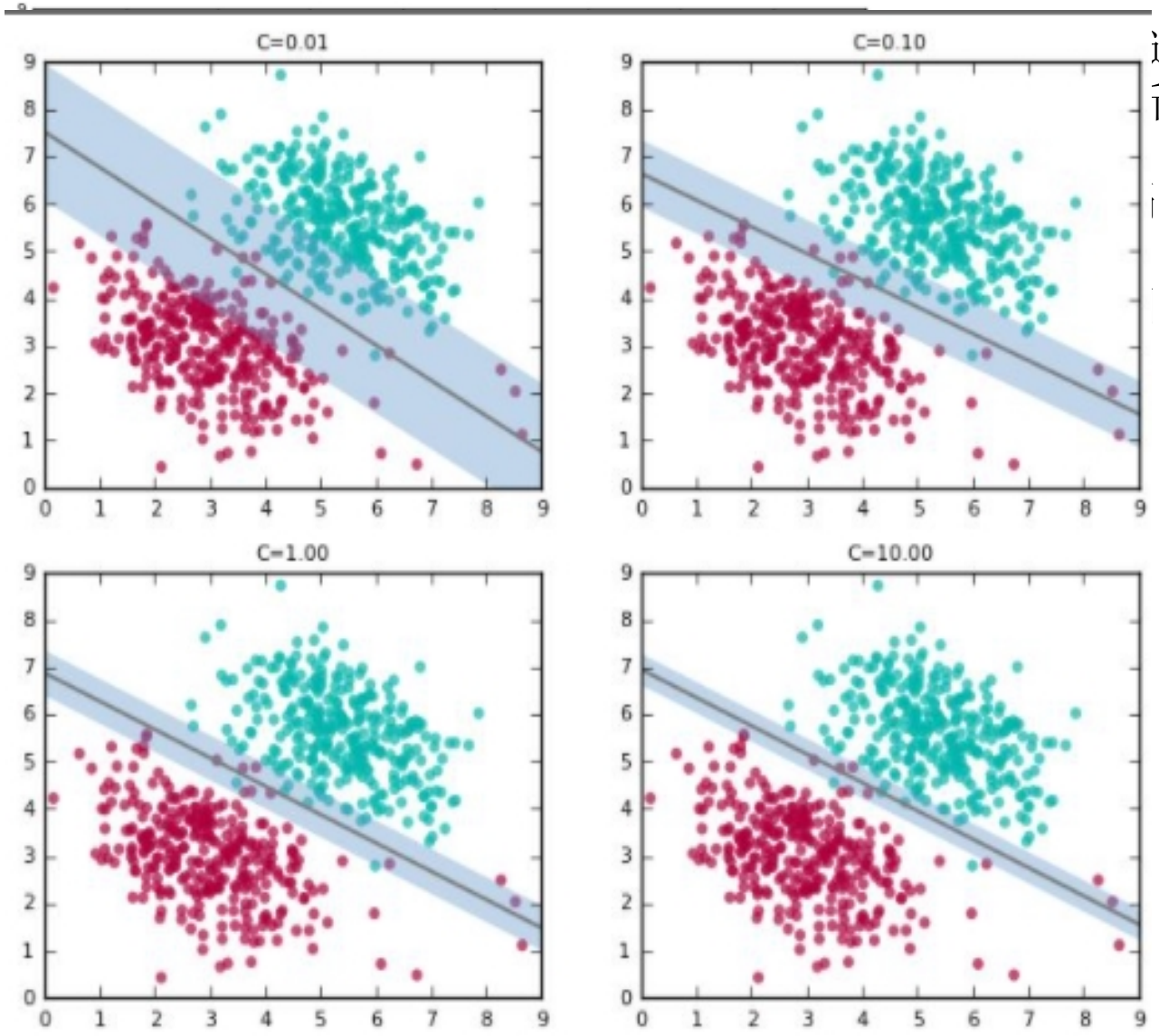
分类器的价值不在于它多么[illegible]能够正确分类。

SVM：1.找到正确分类训练数据的一组直线。

2.在找到的所有直线中，选择那条离最接近的数据点距离最远的直线。

距离最接近的数据点称为支持向量（support vector）。支持向量定义的沿着分隔线的区域称为间隔（margin）。

SVM算法



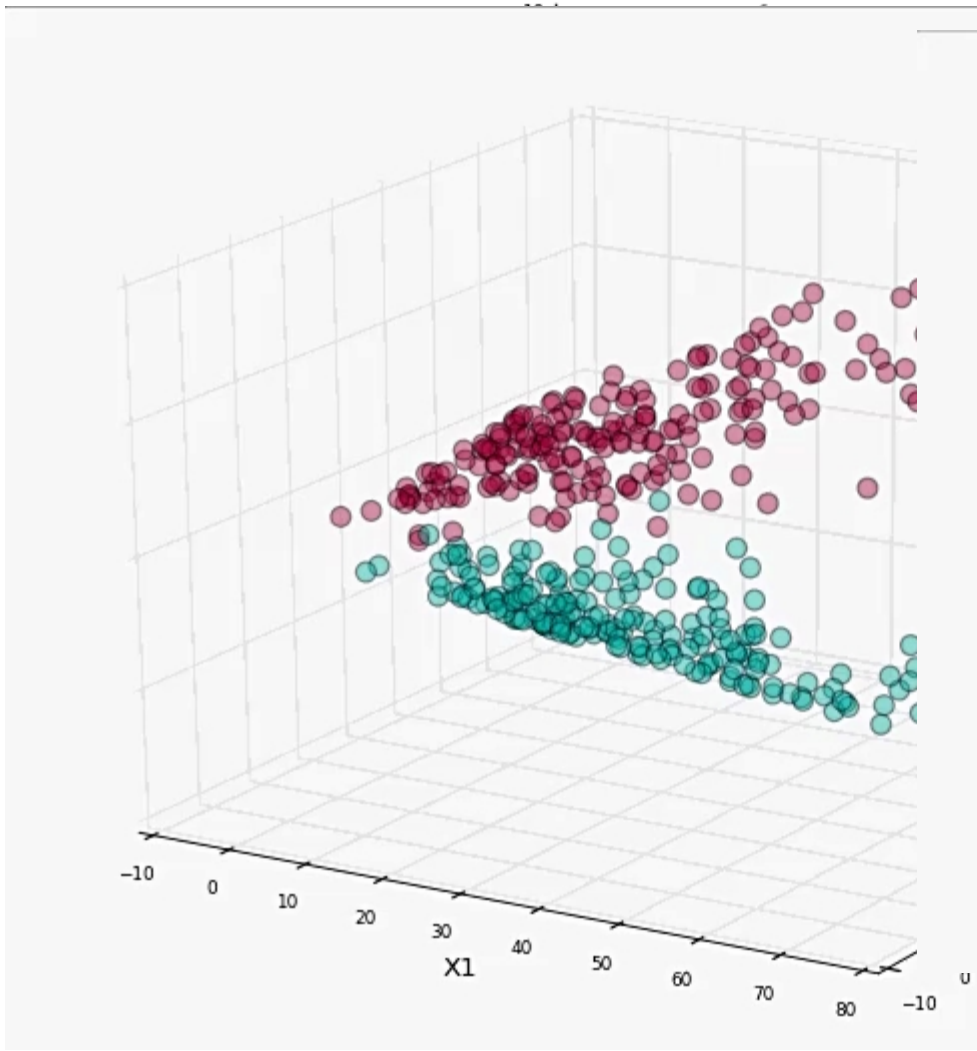
通过参数C指定愿意接受多少误差。C的确定相当于对
衷：

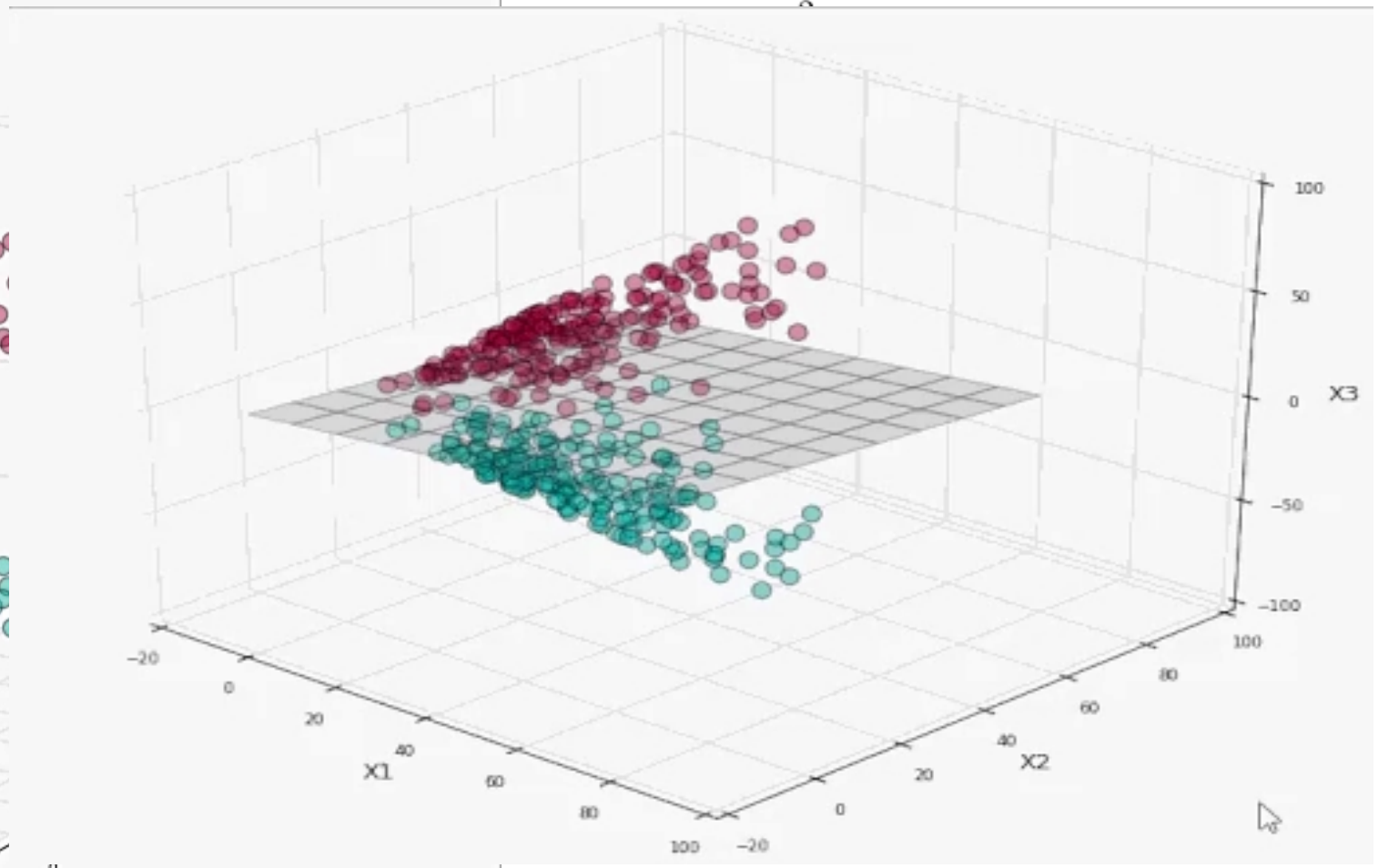
。

练数据。C值较高，意味着训练数据上容许的误差较少。

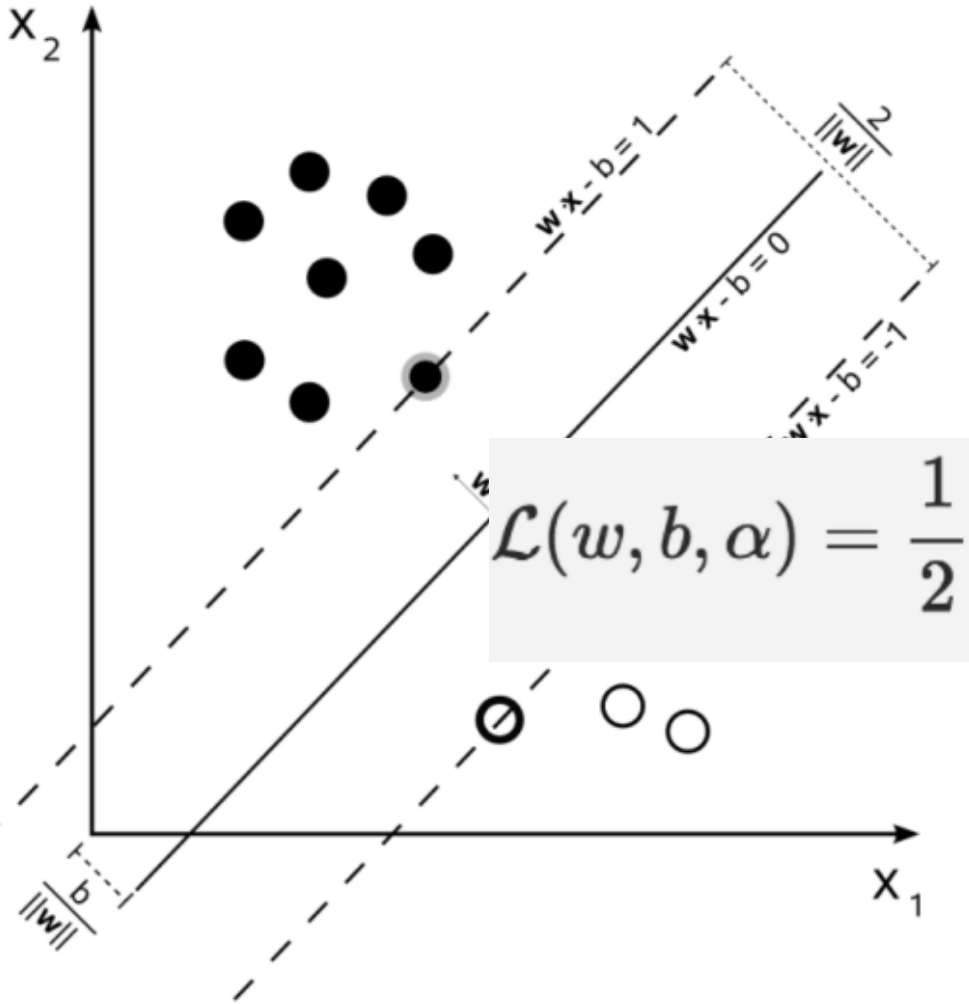
现实世界的数据几乎从来都不是完美可分的，因此决定较优的C值很重要。我们通常使用交叉验证（cross-validation）之类的技术选定较优的C值

SVM算法


−10
0
10
20
30
40
50
60
70
80
X1
−10


100
50
0
−50
−100
X3
−20
0
20
40
60
80
100
X1
−20
0
20
40
60
80
100
X2

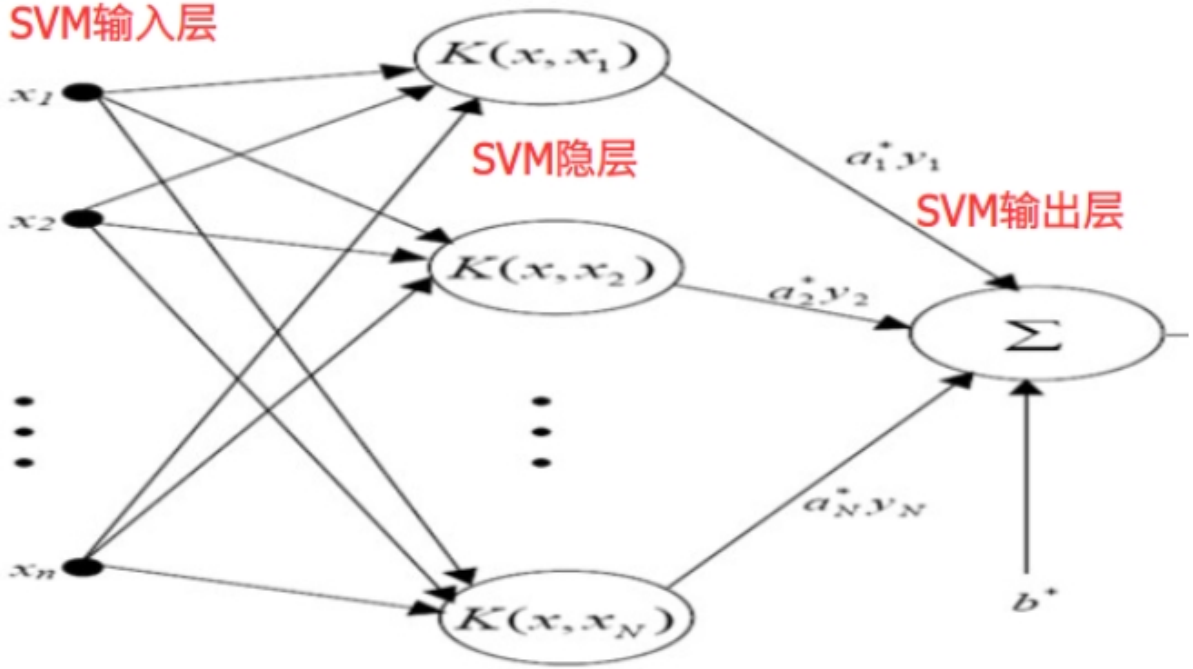
SVM算法



$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}\|w$$

令y的取值为-1或1

SVM输入层



$$\frac{\partial \mathcal{L}}{\partial w}=0\Rightarrow w=\sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b}=0\Rightarrow \sum_{i=1}^{n}\alpha_i y_i=0$$

SVM算法

SVM的优势：

SVM本质就是个MLP，但是这个MLP真的很赞，它智能地解决了MLP中隐层神经元个数问题。

隐层神经元无须很多，只要覆盖支持向量即可。

线性SVM ⇔ 单层感知器；
非线性核SVM ⇔ 多层感知器；

SVM的缺点：

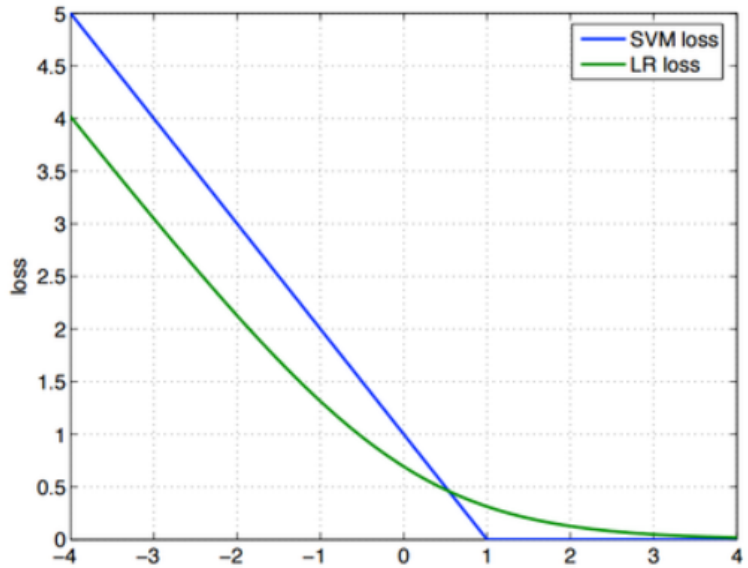
对缺失数据敏感。

对非线性问题没有通用解决方案，必须谨慎选择Kernelfunction来处理。

SVM算法

- SVM 损失函数：$Loss(z) = (1-z)^{+}$

- LR 损失函数: $Loss(z) = \log(1+\exp(-z))$



$\|\mathbf{w}_1\|/2$

$\lambda\|\mathbf{w}_1\|/2$

$\mathbf{w}_1\|/2$

作业：利用SVM算法对生物物种进行分类——鸢尾花(load_iris)

两者之间的关系：

1. LR的目标函数是对数似然函数，SVM的目标函数是hinge损失函数。这两个函数都是增加对分类结果影响较大的数据点的权重，减少影响较小的数据点的权重。

2.□□回□相□来□模型更□□，好理解，特□是大□模□性分□□比□方便。而SVM的理解和□化相□来□复□一些，SVM□化□□偶□□后,分□只需要□算与少数几个支持向量的距离,□个在□行复□核函数□算□□□很明□,能□大大□化模型和□算。

朴素贝叶斯算法

联合概率和条件概率

联合概率：包含多个条件，且所有条件**同时**成立的概率

记作：$P(A, B)$

条件概率：就是事件*A*在另外一个事件*B*已经发生条件下的发生概率

记作：$P(A|B)$

特性：P(A1,A2|B) = P(A1|B)*P(A2|B)
注意：此条件概率的成立，是由于A1,A2相互独立的结果

有P(A|B) = P(B|A) P(A) / P(B)

P(A|B)P(B)=P(B|A)P(A)

朴素贝叶斯算法

症状	职业	疾病
打喷嚏	护士	感冒
打喷嚏	农夫	过敏
头痛	建筑工人	脑震荡
头痛	建筑工人	感冒
打喷嚏	教师	感冒
头痛	教师	脑震荡

一个打喷嚏的建筑工人。请问他患上感冒的概率有多大？

$$P(A|B) = P(B|A)\,P(A) / P(B)$$

$$P(C|F1, F2, \ldots) = \frac{P(F1, F2, \ldots|C)P(C)}{P(F1, F2, \ldots)}$$

P(打口嚏|感冒) =2/3
P(建筑工人|感冒)=1/3
P(感冒)=1/2
P(打口嚏)=1/2
P(建筑工人)=1/3

P(感冒|打口嚏,建筑工人)
= P(打口嚏,建筑工人|感冒) x P(感冒)
/ P(打口嚏,建筑工人)

P(感冒|打口嚏x建筑工人)
= P(打口嚏|感冒) x P(建筑工人|感冒) x P(感冒)
/ P(打口嚏) x P(建筑工人)

=(2/3 * 1/3 * 1/2)/(1/2 *1/3)
= 2/3

朴素贝叶斯算法

- 优点：
  - 朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。
  - 对缺失数据不太敏感，算法也比较简单，常用于文本分类。
  - 分类准确度高，速度快
- 缺点：
  - 需要知道先验概率P(F1,F2,...|C)，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。
  - **朴素□叶斯模型假□属性之□相互独立，□个假□在□□□用中往往是不成立的，在属性个数比□多或者属性之□相关性□大□，分□效果不好**

作业：sklearn20类新闻分类数据集 fetch_20newsgroups

20个主题的18000个新闻组帖子

利用朴素贝叶斯进行预测

决策树

决策树思想的来源非常朴素，程序设计中的条件分支结构就是if-then结构，最早的决策树就是利用这类结构分割数据的一种分类学习方法。

比如：你母亲要给你介绍男朋友，是这么来对话的：

女儿：多大年纪了？

母亲：26。

女儿：长的帅不帅？

母亲：挺帅的。

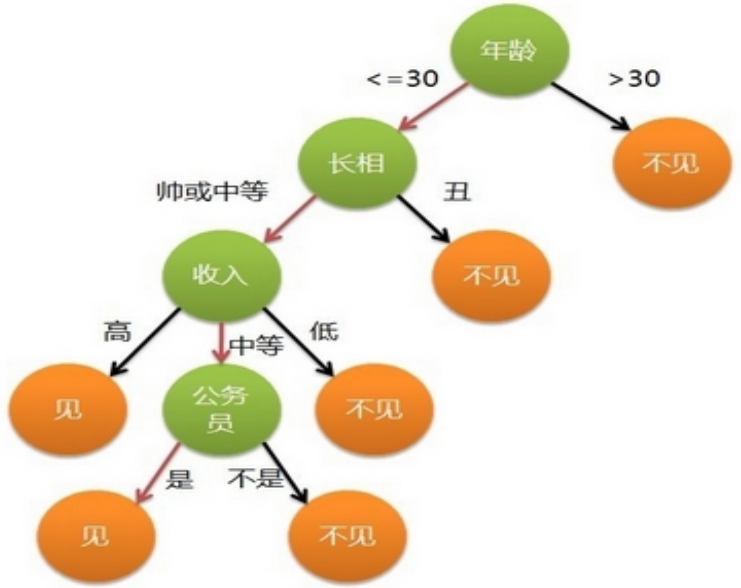
女儿：收入高不？

母亲：不算很高，中等情况。

女儿：是公务员不？

母亲：是，在税务局上班呢。

女儿：那好，我去见见。



决策树

每猜一次给一块钱，告诉我是否猜对了，那么我需要掏多少钱才能知道谁是冠军？我可以把球编上号，从1到32，然后提问：冠军在1-16号吗？依次询问，只需要五次，就可以知道结果。

5元（消除信息的不确定性）

决策树

“谁是世界杯冠军”的信息量应该为5比特。香农指出，它的准确信息量应该是：

H = -(p1*logp1* + *p2*logp2 + ... + p32log32)

- **H的专业术语称之为信息熵，单位为比特。**
- **公式：**

$$-H(X) = \sum_{x \in X} P(x) \log P(x)$$

当这32支球队夺冠的几率相同时，对应的信息熵等于5比特

决策树的划分依据之一-信息增益

特征A对训练数据集D的信息增益g(D,A),定义为集合D的信息熵H(D)与特征A给定条件下D的信息条件熵H(D|A)之差，即公式为：

$$g(D,A)=H(D)-H(D|A)$$

注：信息增益表示得知特征X的信息而使得类Y的信息的不确定性减少的程度

决策树

银行贷款数据

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

你如何去划分是否能得到贷款？

按照工作划分、有房子划分、年龄划分？

决策树

结合贷款数据来看我们的公式：

信息熵的计算：

$$H(D)=-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log\frac{|C_k|}{|D|}$$

条件熵的计算：

$$H(D|A)=\sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D_i|}\log\frac{|D_{ik}|}{|D_i|}$$

注：C_k表示属于某个类别的样本数，

决策树

既然我们有了这两个公式，我们可以根据前面的是否通过贷款申请的例子来通过计算得出我们的决策特征顺序。那么我们首先计算总的经验熵为：

$$H(D)=-(9/15)\log(9/15)-(6/15)\log(6/15) = 0.971$$

然后我们让A_1, A_2, A_3, A_4分别表示年龄、有工作、有自己的房子和信贷情况4个特征，则计算出年龄的信息增益为：

$$\text{年龄：}\ g(D,A1)=H(D) - [(5/15)H(D1)+(5/15)H(D2)+(5/15)H(D3)]$$
$$H(D1) = -(2/5)\log(2/5)-(3/5)\log(3/5)$$
$$H(D2)=-(3/5)\log(3/5)-(2/5)\log(2/5)$$
$$H(D3)=-(4/5)\log(4/5)-(1/5)\log(1/5)$$

同理其他的也可以计算出来，g(D,A2)=0.324,g(D,A3)=0.420,g(D,A4)=0.363，相比较来说其中特征A3（有自己的房子）的信息增益最大，所以我们选择特征A3为最有特征

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|}$$

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

决策树

- ID3

信息增益 最大的准则

- C4.5

信息增益比 最大的准则

- CART

回归树: 平方误差 最小

分类树: 基尼系数 最小的准则 在sklearn中可以选择划分的原则

作业：决策树（泰坦尼克号乘客生存预测）

1、pd读取数据

2、选择有影响的特征，处理缺失值

3、熟悉决策树估计器流程

决策树

- 优点：
  - 简单的理解和解释，树木可视化。
  - 需要很少的数据准备，其他技术通常需要数据归一化，

- 缺点：
  - 当特征较多时，往往生成过于复杂的树，从而过拟合
  - 决策树可能不稳定，因为数据的小变化可能会导致完全不同的树被生成

- 改进：
  - 减枝cart算法
  - 随机森林

随机森林

什么是随机森林？

集成学习通过建立几个模型组合的来解决单一预测问题。它的工作原理是生成多个分类器/模型，各自独立地学习和作出预测。这些预测最后结合成单预测，因此优于任何一个单分类的做出预测。

定义：在机器学习中，**随机森林**是一个包含多个决策树的分类器，并且其输出的类别是由树输出的类别的众数而定。

例如, 如果你训练了5个树, 其中有4个树的结果是True, 1个数的结果是False, 那么最终结果会是True.

随机森林

学习算法

根据下列算法而建造每棵树：

•用N来表示训练用例（样本）的个数，M表示特征数目。

•输入特征数目m，用于确定决策树上一个节点的决策结果；其中m应远小于M。

•从N个训练用例（样本）中以有放回抽样的方式，取样N次，形成一个训练集（即bootstrap取样），并用未抽到的用例（样本）作预测，评估其误差。

- **为什么要随机抽样训练集？**

如果不进行随机抽样，每棵树的训练集都一样，那么最终训练出的树分类结果也是完全一样的

- **为什么要有放回地抽样？**

如果不是有放回的抽样，那么每棵树的训练样本都是不同的，都是没有交集的，这样每棵树都是“有偏的”，也就是说每棵树训练出来都是有很大的差异的；而随机森林最后分类取决于多棵树（弱分类器）的投票表决。

随机森林

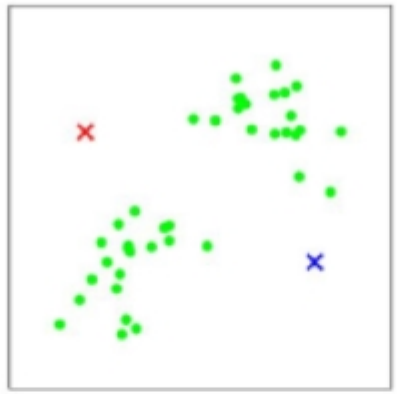
- *class* sklearn.ensemble.RandomForestClassifier(*n_estimators=10*, *criterion='gini'*, *max_depth=None*, *bootstrap=True*, *random_state=None*)
  - 随机森林分类器
  - n_estimators：integer，optional（default = 10） 森林里的树木数量
  - criteria：string，可选（default ="gini"）分割特征的测量方法
  - max_depth：integer或None，可选（默认=无）树的最大深度
  - bootstrap：boolean，optional（default = True）是否在构建树时使用放回抽样

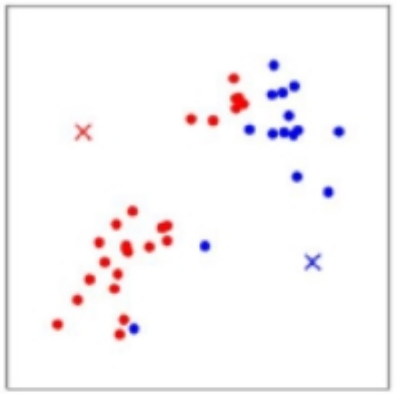
作业：随机森林进行泰坦尼克号乘客生存预测

随机森林的优点

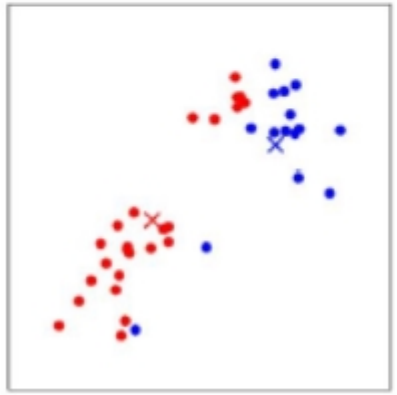
- 在当前所有算法中，具有极好的准确率
- 能够有效地运行在大数据集上
- 能够处理具有高维特征的输入样本，而且不需要降维
- 能够评估各个特征在分类问题上的重要性
- 对于缺省值问题也能够获得很好得结果

K-Means





(a)



(d)

1、 随机选取k个聚类质心点（cluster centroids）为$\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$。

2、 重复下面过程直到收敛 {

对于每一个样例i，计算其应该属于的类

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2. \quad (1)$$

对于每一个类j，重新计算该类的质心

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}. \quad (2)$$

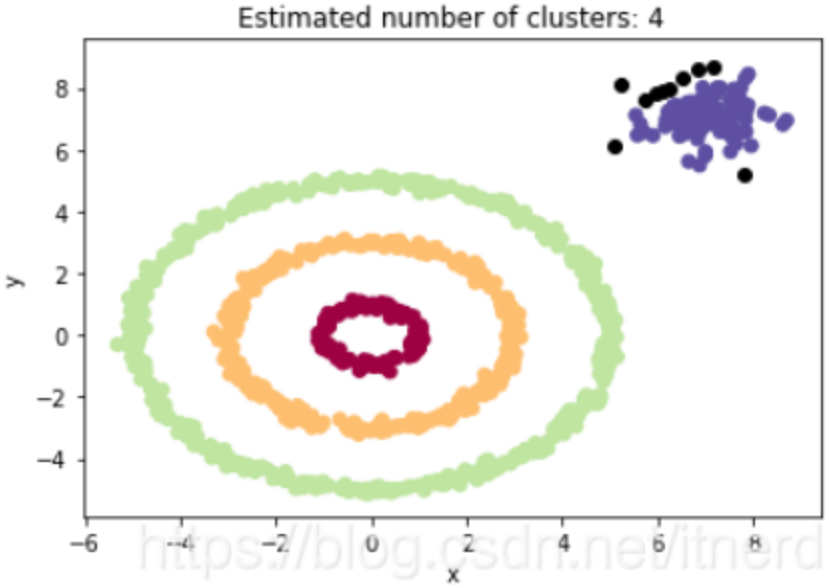
}

Kmeans优缺点

优点

容易理解，聚类效果不错，虽然是局部最优， 但往往局部最优就够了；

处理大数据集的时候，该算法可以保证较好的伸缩性；



当簇近似高斯分布

算法复杂度低。

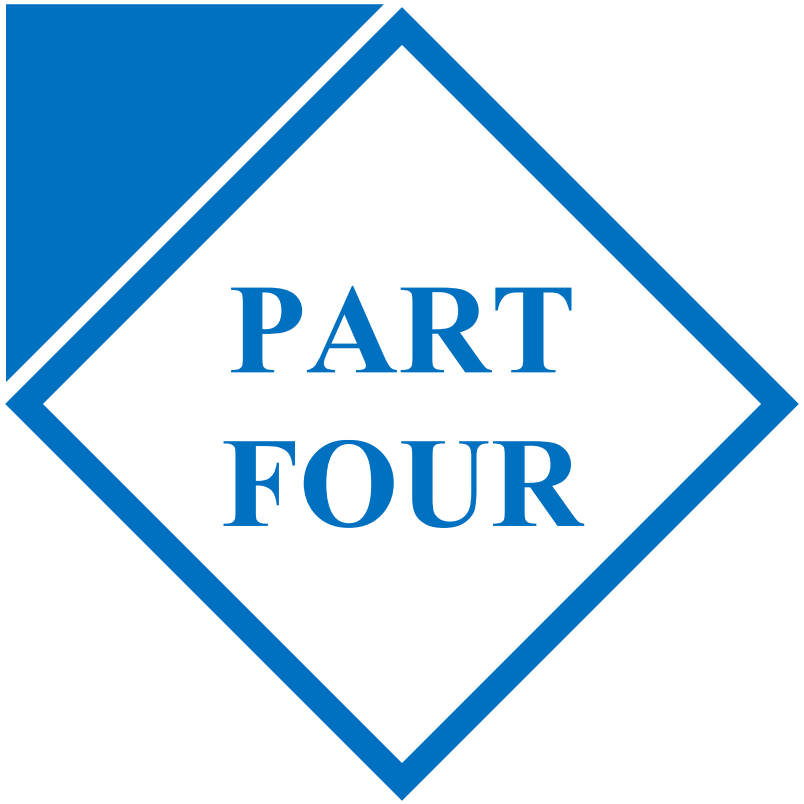
缺点

K 值需要人为设定，

对初始的簇中心敏感

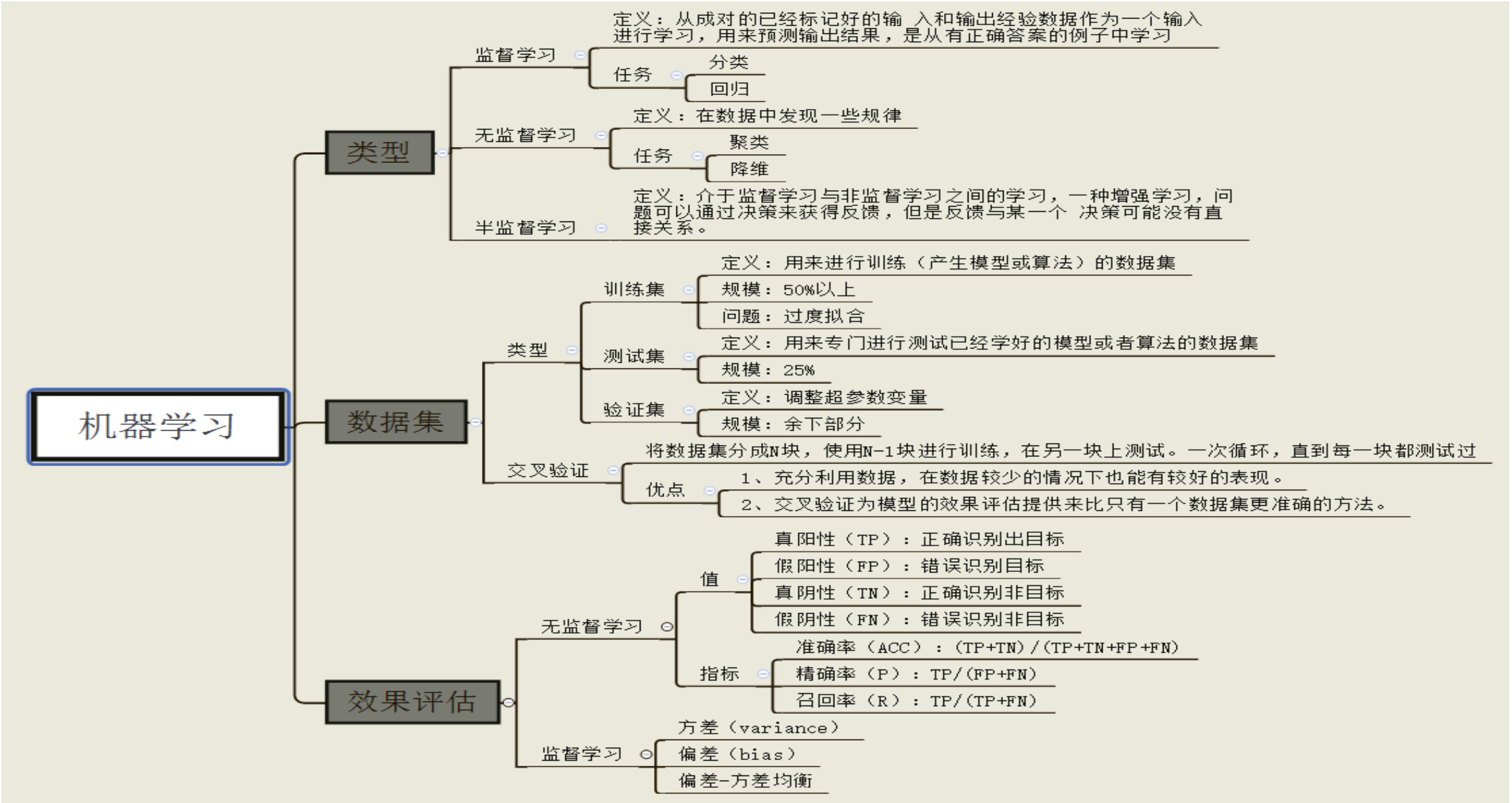
对异常值敏感；

不适合太离散的分类、样本类别不平衡的分类、非凸形状的分类。



总结

总结


机器学习
类型
监督学习
定义：从成对的已经标记好的输 入和输出经验数据作为一个输入进行学习，用来预测输出结果，是从有正确答案的例子中学习
任务
分类
回归
无监督学习
定义：在数据中发现一些规律
任务
聚类
降维
半监督学习
定义：介于监督学习与非监督学习之间的学习，一种增强学习，问题可以通过决策来获得反馈，但是反馈与某一个 决策可能没有直接关系。
数据集
类型
训练集
定义：用来进行训练（产生模型或算法）的数据集
规模：50%以上
问题：过度拟合
测试集
定义：用来专门进行测试已经学好的模型或者算法的数据集
规模：25%
验证集
定义：调整超参数变量
规模：余下部分
交叉验证
将数据集分成N块，使用N-1块进行训练，在另一块上测试。一次循环，直到每一块都测试过
优点
1、充分利用数据，在数据较少的情况下也能有较好的表现。
2、交叉验证为模型的效果评估提供来比只有一个数据集更准确的方法。
效果评估
无监督学习
值
真阳性（TP）：正确识别出目标
假阳性（FP）：错误识别目标
真阴性（TN）：正确识别非目标
假阴性（FN）：错误识别非目标
指标
准确率（ACC）：(TP+TN)/(TP+TN+FP+FN)
精确率（P）：TP/(FP+FN)
召回率（R）：TP/(TP+FN)
监督学习
方差（variance）
偏差（bias）
偏差-方差均衡

第9章 机器学习sklearn库

Thank you!

Q & A