降维，聚类与分类算法

1 降维算法

1.1 降维算法的概念

降维：在机器学习和统计学领域，降维是指在某些限定条件下，降低随机变量个数，得到一组“不相关”主变量的过程。

换言之，降维其更深层次的意义在于**有效信息的提取综合及无用信息的摈弃。**

数据降维算法是机器学习算法中的大家族，与分类、回归、聚类等算法不同，它的目标是将向量投影到**低维空间**，以达到某种目的如可视化，或是做分类。

当特征选择完成后，可以直接训练模型了，但是可能由于特征矩阵过大，导致计算量大，训练时间长的问题，因此**降低特征矩阵维度**也是必不可少的。

除了便于可视化，数据降维算法还可以提升分类、聚类算的精度，避免维数灾难问题。抽象来看，数据降维就是寻找一个映射函数 f，将高维向量 x 映射成低维向量 y

如何确定这个**映射函数**，是各降维算法核心，它们往往根据不同的准则进行构造。

1.2 降维算法的分类

目前已经存在大量的数据降维算法，可以从不同的维度对它们进行分类。

按照是否有使用样本的标签值，可以将降维算法分为有监督降维和无监督降维；

按照降维算法使用的映射函数，可以将算法分为线性降维与非线性降维；

无监督降维算法不使用样本标签值，因此是一种**无监督学习算法，其典型代表是 PCA；**

有监督的降维算法则使用了样本标签值，是一种**有监督学习算法，其典型代表是 LDA；**

此文档后面的附注链接附有十种常见的降维算法的 python 源代码网址。

1.3 以 PCA 为例：

PCA(Principal Component Analysis)，即主成分分析方法，是一种使用最广泛的数据降维算法。PCA 的主要思想是将 n 维特征映射到 k 维上，这 k 维是全新的正交特征也被称为主成分，是在原有 n 维特征的基础上重新构造出来的 k 维特征。PCA 的工作就是从原始的空间中顺序地找一组相互正交的坐标轴，新的坐标轴的选择与数据本身是密切相关的。其中，第一个新坐标轴选择是原始数据中方差最大的方向，第二个新坐标轴选取是与第一个坐标轴正交的平面中使得方差最大的，第三个轴是与第 1,2 个轴正交的平面中方差最大的。依次类推，可以得到 n 个这样的坐标轴。通过这种方式获得的新的坐标轴，我们发现，大部分方差都包含在前面 k 个坐标轴中，后面的坐标轴所含的方差几乎为 0。于是，我们可以忽略余下的坐标轴，只保留前面 k 个含有绝大部分方差的坐标轴。事实上，这相当于只保留包含绝大部分方差的维度特征，而忽略包含方差几乎为 0 的特征维度，实现对数据特征的降维处理。

思考：我们如何得到这些包含最大差异性的主成分方向呢？

答案：事实上，通过计算数据矩阵的协方差矩阵，然后得到协方差矩阵的特征值特征向量，选择特征值最大(即方差最大)的 k 个特征所对应的特征向量组成的矩阵。这样就可以将数据矩阵转换到新的空间当中，实现数据特征的降维。

由于得到协方差矩阵的特征值特征向量有两种方法：特征值分解协方差矩阵、奇异值分解协方差矩阵，所以 PCA 算法有两种实现方法：基于特征值分解协方差矩阵实现 PCA 算法、基于 SVD 分解协方差矩阵实现 PCA 算法。

简述一下 PCA 的算法步骤:

设有 n 条 d 维数据。

1. 将原始数据按列组成 n 行 d 列矩阵 X
2. 将 X 的每一列（代表一个属性）进行零均值化，即减去这一列的均值
3. 求出协方差矩阵 $C = \frac{1}{m}XX^{\mathsf{T}}$
4. 求出协方差矩阵的特征值及对应的特征向量
5. 将特征向量按对应特征值大小从上到下按行排列成矩阵，取前 k 行组成矩阵 P
6. $Y = PX$ 即为降维到 k 维后的数据

2 聚类算法

聚类(Clustering)是按照某个特定标准(如距离)把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能地大。也即聚类后同一类的数据尽可能聚集到一起，不同类数据尽量分离。

聚类和分类的区别

聚类(Clustering)：是指把相似的数据划分到一起，具体划分的时候并不关心这一类的标签，目标就是把相似的数据聚合到一起，聚类是一种无监督学习(Unsupervised Learning)方法。

分类(Classification)：是把不同的数据划分开，其过程是通过训练数据集获得一个分类器，再通过分类器去预测未知数据，分类是一种监督学(Supervised Learning)方法。

聚类的一般过程

1. 数据准备：特征标准化和降维
2. 特征选择：从最初的特征中选择最有效的特征，并将其存储在向量中
3. 特征提取：通过对选择的特征进行转换形成新的突出特征
4. 聚类：基于某种距离函数进行相似度度量，获取簇
5. 聚类结果评估：分析聚类结果，如距离误差和(SSE)等

数据对象间的相似度度量

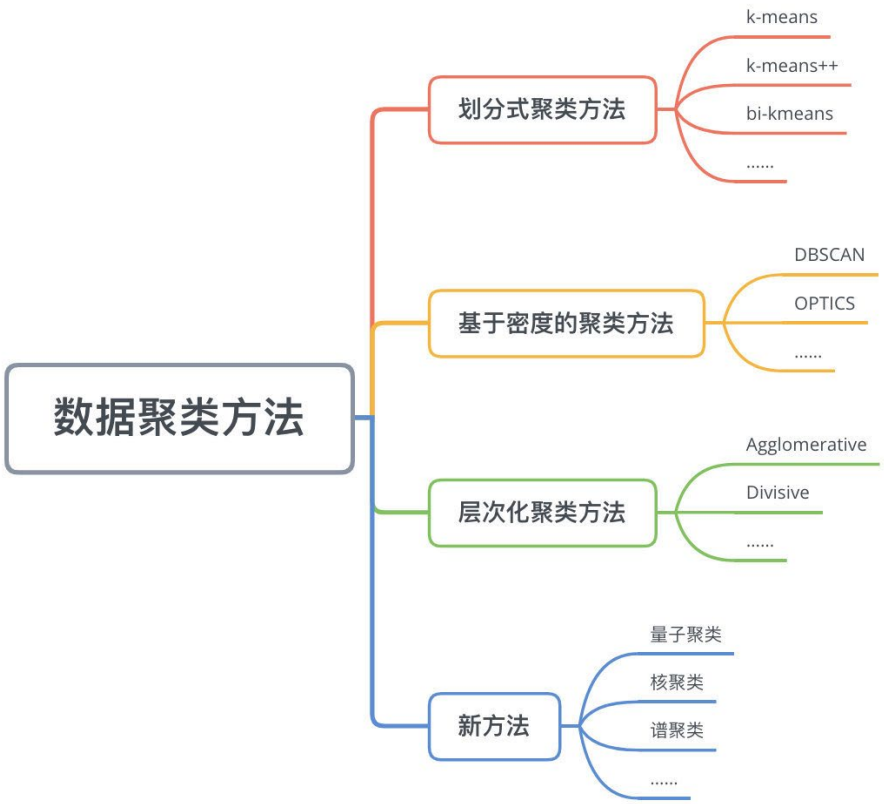
对于数值型数据，可以使用下表中的相似度度量方法：

相似度度量准则	相似度度量函数
Euclidean 距离	$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
Manhattan 距离	$d(x,y)=\sum_{i=1}^{n}\|x_i-y_i\|$
Chebyshev 距离	$d(x,y)=\max_{i=1,2,\ldots,n}^{n}\|x_i-y_i\|$
Minkowski 距离	$d(x,y)=[\sum_{i=1}^{n}(x_i-y_i)^p]^{\frac{1}{p}}$

除了需要衡量对象之间的距离之外，有些聚类算法（如层次聚类）还需要衡量 cluster 之间的距离 。

相似度度量准则	相似度度量函数
Single-link	$D(C_i, C_j) = \min_{x \subseteq C_i, y \subseteq C_j} d(x, y)$
Complete-link	$D(C_i, C_j) = \max_{x \subseteq C_i, y \subseteq C_j} d(x, y)$
UPGMA	$D(C_i, C_j) = \frac{1}{\|C_i\| \|C_j\|} \sum_{x \subseteq C_i, y \subseteq C_j} d(x, y)$
WPGMA	-

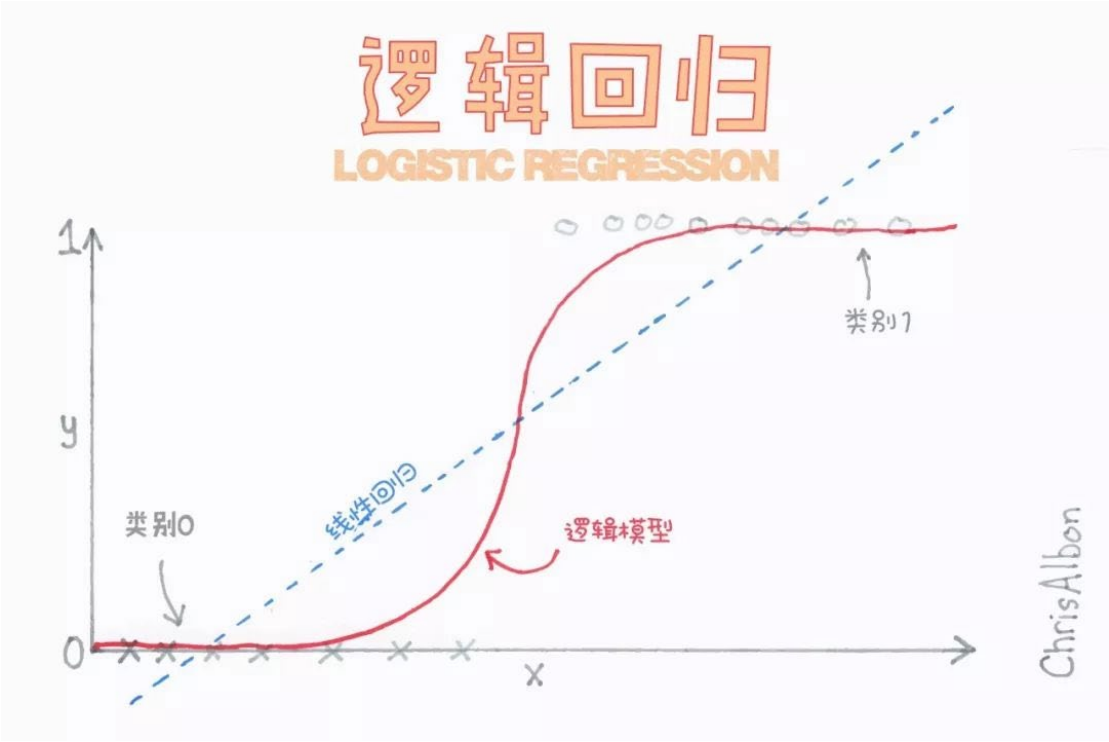
常见的聚类算法：



3 分类算法

逻辑回归

逻辑回归类似于线性回归，适用于因变量不是一个数值字的情况（例如，一个“是/否”的响应）。它虽然被称为回归，但却是基于根据回归的分类，将因变量分为两类。



K-NN 算法是一种最简单的分类算法，通过识别被分成若干类的数据点，以预测新样本点的分类。K-NN 是一种非参数的算法，是“懒惰学习”的著名代表，它根据相似性（如，距离函数）对新数据进行分类。

KNN算法

Does K-NN Learn

KNN算法
样本A最接近的k个样本本身不训练，他是懒惰的，仅仅记住所有数据

*译者注：

KNN可理解为一种死记硬背式的分类器，记住所有的训练数据，对于新的数据则直接和训练数据匹配，如果存在相同属性的训练数据，则直接用它的分类来作为新数据的分类。

Chris Albon

KNN算法的小技巧

k-nearest neighbors tips and tricks

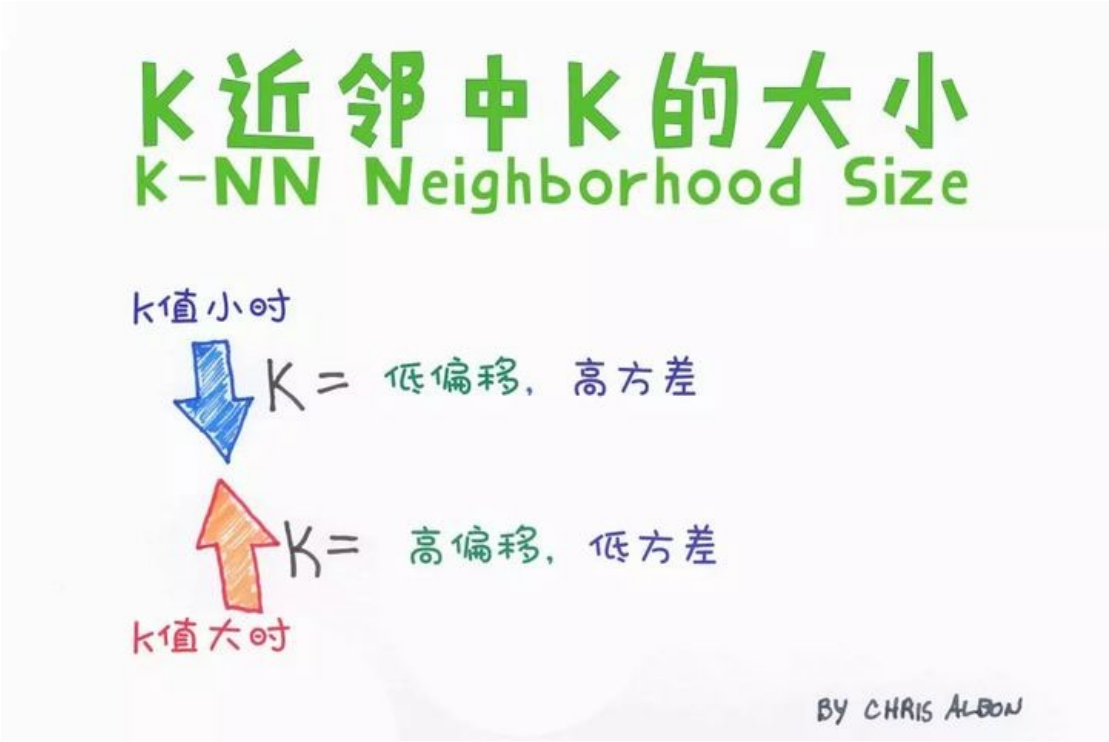
1. 所有的特征应该被放缩到相同的量级

2. 为了避免平票的出现，K应该选择奇数

3. 投票的结果会被到近邻样本的距离归一化，这样更近的样本的投票价值更大

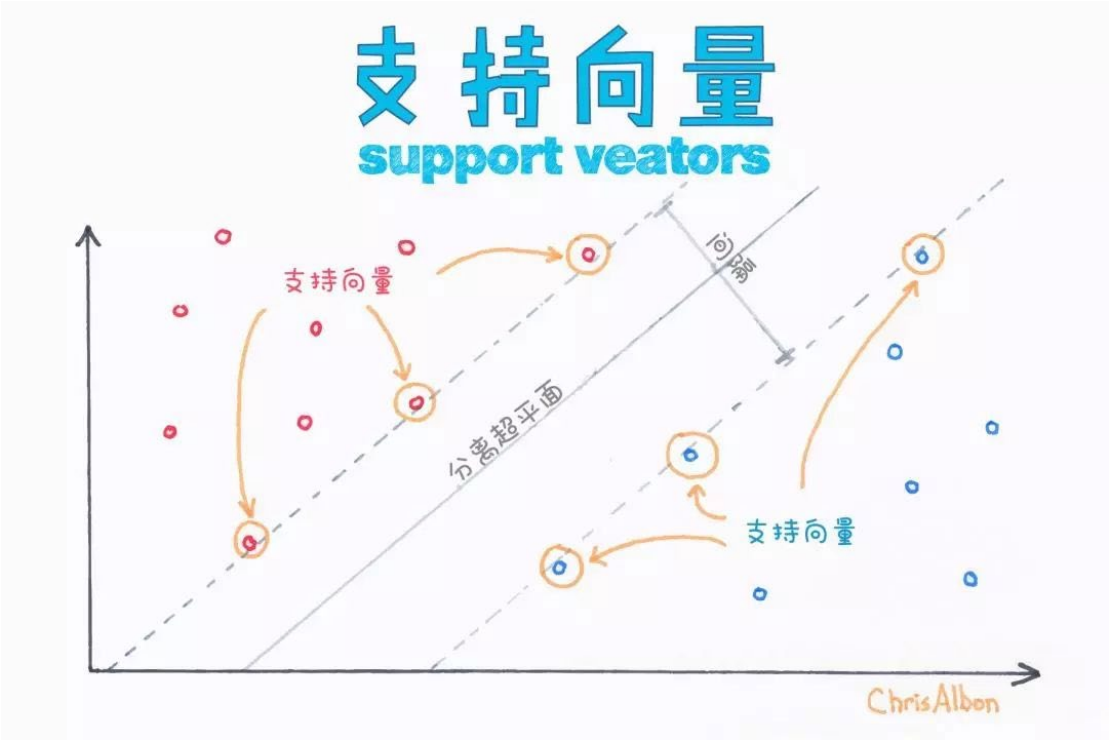
4. 尝试各种不同的距离度量方法

Chris Albon

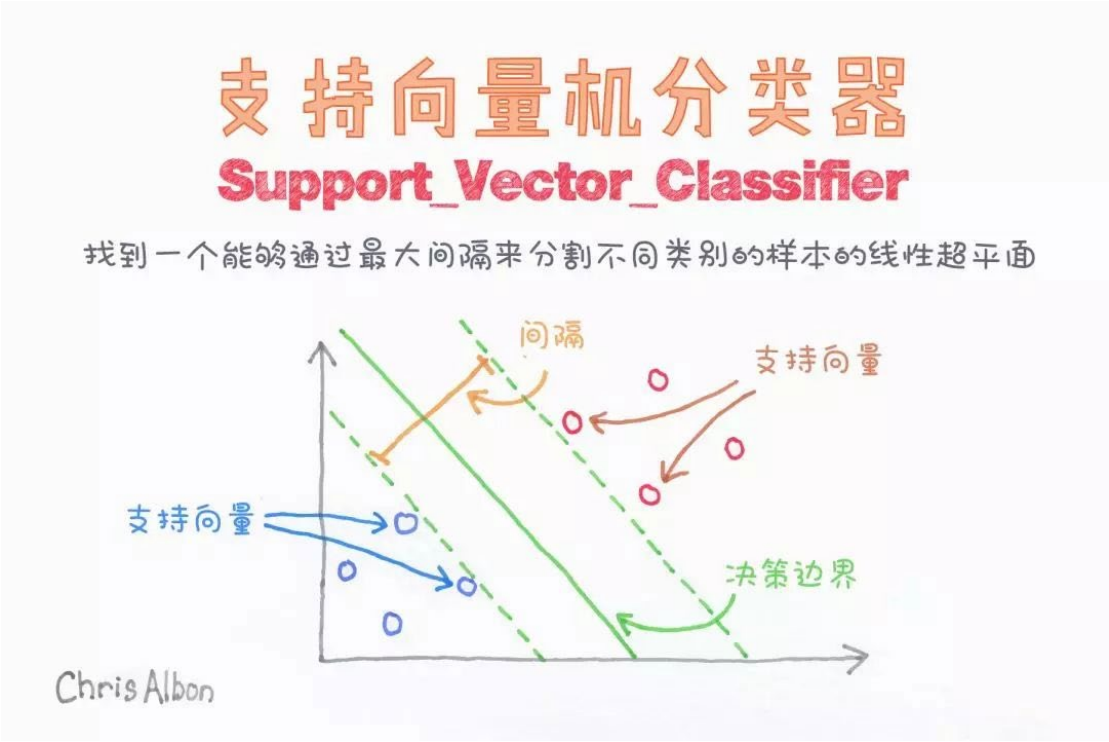


支持向量机（SVM）

支持向量机既可用于回归也可用于分类。它基于定义决策边界的决策平面。决策平面（超平面）可将一组属于不同类的对象分离开。

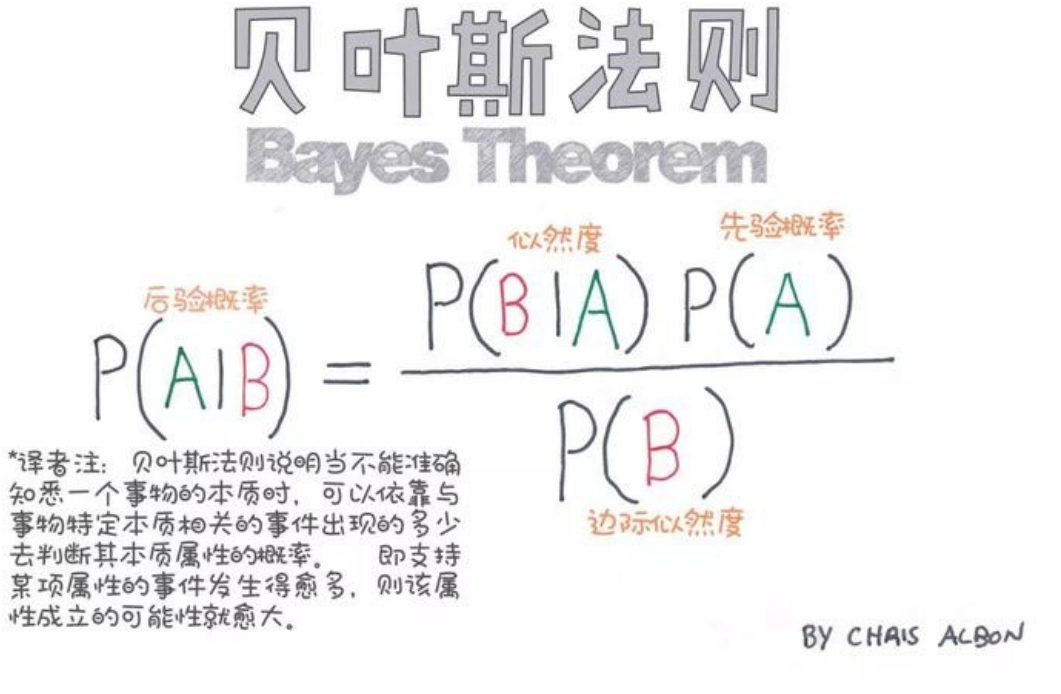


在支持向量的帮助下，SVM 通过寻找超平面进行分类，并使两个类之间的边界距离最大化。



SVM 中超平面的学习是通过将问题转化为使用一些某种线性代数转换问题来完成的。（上图的例子是一个线性核，它在每个变量之间具有线性可分性）。

对于高维数据，使用可使用其他核函数，但高维数据不容易进行分类。



高斯朴素贝叶斯分类器
Gaussian Naive Bayes Classifier

称之为"高朴"是因为这是一个正态分布

这是我们的先验信念

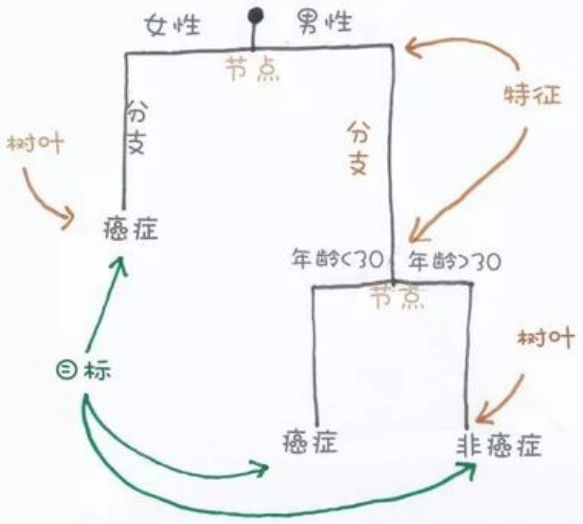
$$P(class \mid data) = \frac{P(data \mid class) \times P(class)}{P(data)}$$

我们在朴素贝叶斯分类器中不用计算这个概率值

*译者注：
不需要计算P(data)是因为它与类标记无关。
高斯朴素贝叶斯是针对连续值特征的，离散特征的话直接计算出现的频率就ok。

ChrisAlbon

决策树

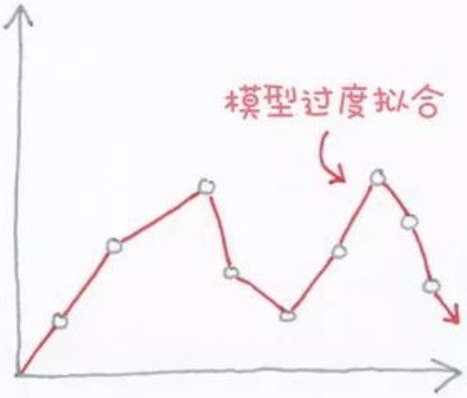


决策树易于解释。训练之后，你还可以原原本本的把它们画出来。
在提供最高信息增益的特征处将数据划分开。

BY CHRIS ALBON

过度拟合 Overfitting

过度拟合是指模型过于彻底地学习训练数据，以致无法一般化推广的现象。



Chris Albon

集成学习方法
Ensemble Methods

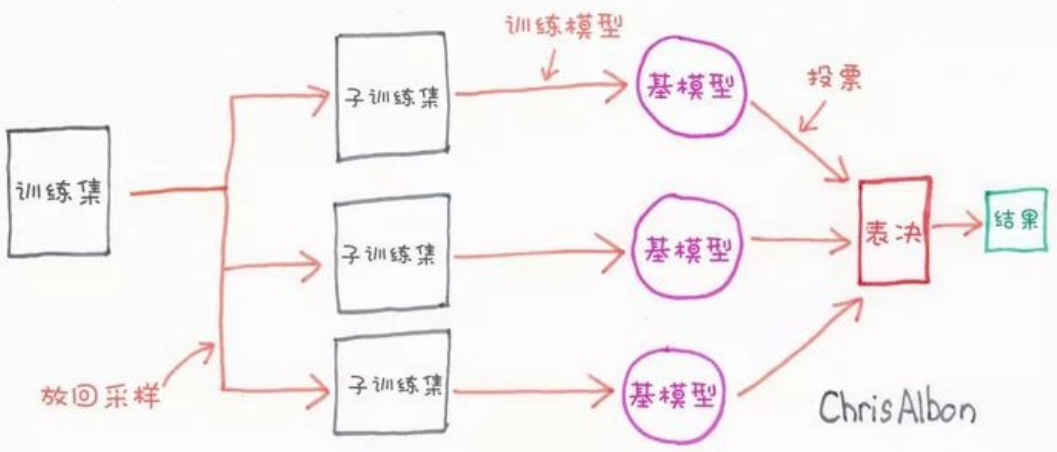
当几个模型分别训练完毕后，我们通过投票或平均结果的方式来得到预测值。例如，随机森林模型。

译者注：
集成学习应该还包括提升（典型的有GBDT模型），而不仅仅只是bagging这一种形式。

Chris Albon

Bagging算法

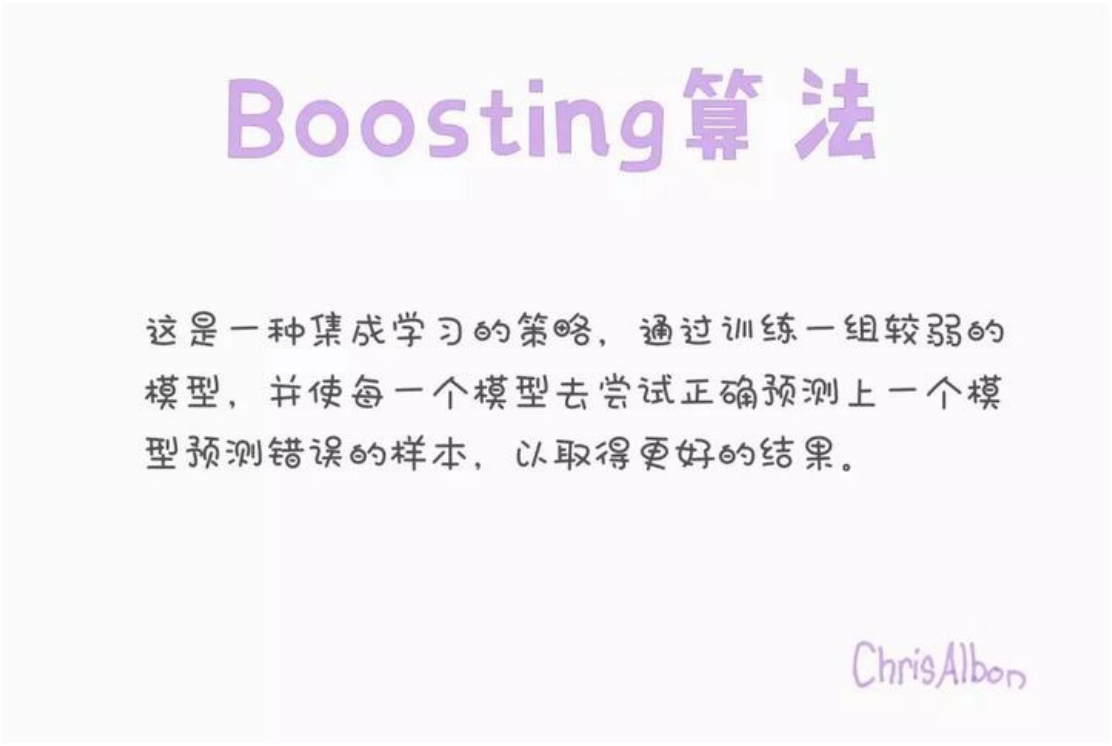
Bootstrap aggregation



随机森林

random for forest

1.很多树是用特征变量和引导数据的随机子集创建的。


Boosting算法
这是一种集成学习的策略，通过训练一组较弱的模型，并使每一个模型去尝试正确预测上一个模型预测错误的样本，以取得更好的结果。
ChrisAlbon

分类模型的评价标准：


混淆矩阵
confusion matrix
混淆矩阵通过比较分类结果与真实值，形象反映分类模型的准确度，即好坏。在左图，不在正方形对角线上的小正方形是错误的分类结果。
+注释在对角线上的数值是预测值与实际值相符的数值。
例如图所示第一行A预测正确，B、C预测错误。
ChrisAlbon
预测类
真实类
A
B
C

误报率
False Positive Rate

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

准确率
ACCURACY

$$Acc = \frac{1}{n}\sum 1(\hat{y}_i = y_i)$$

$\hat{y}_i$：预测值 $\hat{y}$帽

y_i：真实值 y

n：观察体数量（样本数量）
统计学中针对目标群体的
一个定向观察指标

$1(\hat{y}_i = y_i)$：指示函数，定义在某集合X上
的函数，表示其中有哪些元
素属于某一子集A

对于给定的一组数据的一系列测量，当y_i的预测值与真实值y相等时，指示函数取1，不相等时，指示函数取0。在类别严重不均衡时，F1 值更符合。

译者注：
准确率accuracy其定义：对于给定的测试数据集，分类器正确分类的样本数与总样本数之比，反映了分类系统对整个样本的判定能力——能将正的判定为正，负的判定为负。

精确度

确度是指分类器不会把真阴性分类为阳性的能力

真阳性/ （ 真阳性 + 假阳性 ）

正确标记正值

正确标记正值 + 误标记正值

ChrisAlbon

召回率

Recall

召回率是关于正类的。

True Positives

True Positives + False Negatives

TP/(TP+FN)

召回率是关于（预测为正确的实例中）真正正确（的实例比例）的

。

召回率是用于度量分类器发现真正正确实例的能力。如果我们想要确保发现所有真正正确的实例，我们需要最大化召回率。

Chris Albon

F1值
F1 score

$$F_1 = 2 \times \frac{准确率 * 召回率}{准确率 + 召回率}$$

F1的值是精确率与召回率的调和平均数，F1的取值范围从0到1，数量越大，表明实验结果越理想。

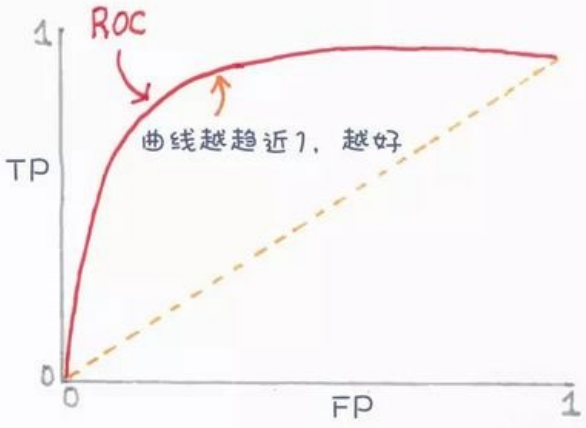
*译者注：

Precision 精确率 TP/(TP+FP) 预测为真，实际也为真/预测为真的总数

Recall 召回率 TP/(TP+FN) 预测为真，实际也为真/实际为真的总数

Chris Albon

ROC曲线
receiver operating characteristic



给出了一个二元分类器的每个概率阈值的真阳率和假阳率

Chris Albon

4、一些题外话

很多同学都不知道的是，在场景比较简单的情况下，用 matlab 做机器学习真的真的非常简单!!!比 python 还要简单很多。由于 matlab 拥有自带的神经网络工具箱以及分类工具箱等，我们只要点点鼠标，就可以训练出很多个由各种算法训练出来的模型，而且 matlab 还会自动地帮我们把图给画好!!!参数我们可以手动调节。此外，matlab 有个非常友好的地方：一经下载，就所有的工具箱和库都是全的，不需要额外地花心思去导库，而各种操作都有详细的官方 api 文档说明，所以哪怕不知道机器学习的算法或者内容，借助 matlab，我们只需要了解一个算法的数据格式、输入、输出即可，其它的都让 matlab 来帮我们完成。

因此，强烈建议下载一个 matlab!!!强烈建议稍微花点时间学习一下 matlab 的神经网络工具箱的使用，因为能够用到的场景真的非常非常多，不管你是要做分类、预测还是降维，或者需要用到一些现成的优化算法，matlab 都是真香之选！对于刚刚入门的小白来说，更加重要的是可以快速构建和部署模型，而不是被复杂的数学公式杀死深入学习的欲望。

当然，如果想做算法或者需要在更加复杂的场景下进行定制化、个性化的运用的话，那么还是得回到 python 上来，但是作为入门的选择，matlab 足矣！推荐一本书《大数据挖掘-系统方法与实例分析》卓金武著，稍微花一两个礼拜的时间把这本书读读，学会的知识足够应付多数场景下的需求。

5、附录

1、10 种常用降维算法源代码(python) - 超爱学习的文章 - 知乎
https://zhuanlan.zhihu.com/p/68754729

2、MATLAB BP 神经网络工具箱使用步骤 - 大维的文章 - 知乎
https://zhuanlan.zhihu.com/p/429221020

3、优雅的使用 Matlab 做机器学习 - 了凡春秋的文章 - 知乎
https://zhuanlan.zhihu.com/p/87554660

4、人工神经网络学习笔记 2——MATLAB 神经网络工具箱 - 在逃水星下落不明的文章 - 知乎
https://zhuanlan.zhihu.com/p/66280431

5、一文读懂机器学习分类算法（附图文详解） - CDA 数据分析师的文章 - 知乎
https://zhuanlan.zhihu.com/p/82114104

6、scikt-learn 官方社区 https://scikit-learn.org.cn/