决策树与随机森林

3月机器学习在线班 邹博

2015年3月29日

目标任务与主要内容

- 复习信息熵
  - 熵、联合熵、条件熵、互信息
- 决策树学习算法
  - 信息增益
  - ID3、C4.5、CART
- Bagging与随机森林的思想
  - 投票机制

先提个额外的问题

- 线性回归的目标函数：

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- Logistic回归的目标函数：

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

- 请问：既然二者都隶属广义线性回归的理论体系，为什么他们的目标函数，一个是最小二乘，一个是似然函数？

复习：条件熵

- H(X,Y) – H(X)
  - (X,Y)发生所包含的熵，减去X单独发生包含的熵：在X发生的前提下，Y发生“新”带来的熵
  - 该式子定义为X发生前提下，Y的熵：
    - 条件熵H(Y|X)

复习：推导条件熵的定义式

$$
\begin{aligned}
& H(X,Y)-H(X) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x} p(x)\log p(x) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x}\left(\sum_{y} p(x,y)\right)\log p(x) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x,y} p(x,y)\log p(x) \\
&= -\sum_{x,y} p(x,y)\log \frac{p(x,y)}{p(x)} \\
&= -\sum_{x,y} p(x,y)\log p(y\,|\,x)
\end{aligned}
$$

复习：根据条件熵的定义式，可以得到

$$H(X,Y)-H(X)=-\sum_{x,y}p(x,y)\log p(y\mid x)$$

$$=-\sum_{x}\sum_{y}p(x,y)\log p(y\mid x)$$

$$=-\sum_{x}\sum_{y}p(x)p(y\mid x)\log p(y\mid x)$$

$$=-\sum_{x}p(x)\sum_{y}p(y\mid x)\log p(y\mid x)$$

$$=\sum_{x}p(x)\left(-\sum_{y}p(y\mid x)\log p(y\mid x)\right)$$

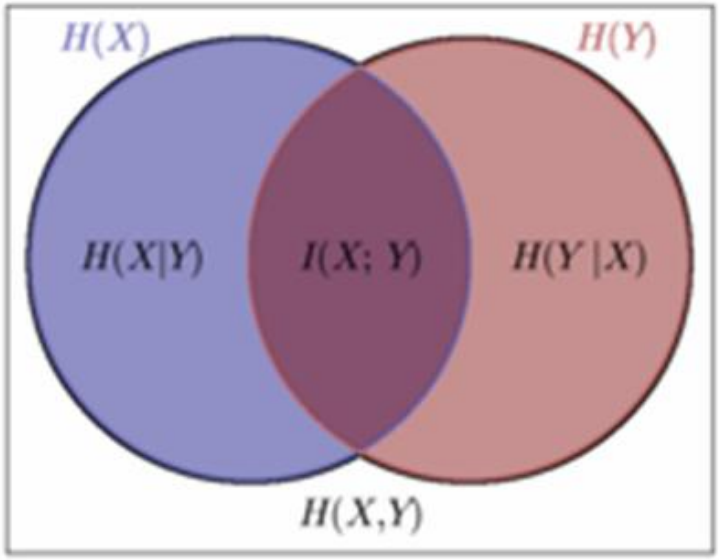
$$=\sum_{x}p(x)H\left(Y\mid X=x\right)$$

复习：熵的等式

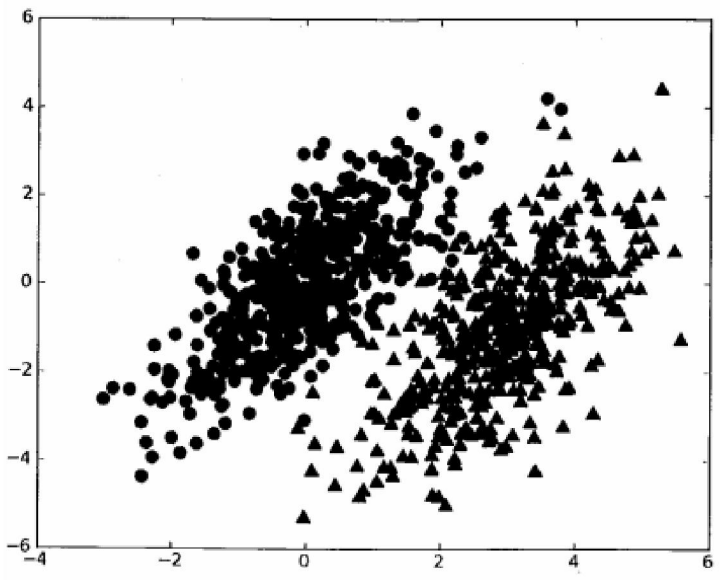
- $H(Y|X)= H(X,Y) - H(X)$
  - 条件熵定义
- $H(Y|X)= H(Y) - I(X,Y)$
  - 互信息定义展开
  - 有些文献将$I(X,Y)=H(Y) – H(Y|X)$作为互信息的定义式
- 对偶式
  - $H(X|Y) = H(X,Y) - H(Y)$
  - $H(X|Y) = H(X) - I(X,Y)$
- $I(X,Y)= H(X) + H(Y) - H(X,Y)$
  - 有些文献将该式作为互信息的定义式
- 试证明：$H(X|Y) \leqslant H(X)$，$H(Y|X) \leqslant H(Y)$

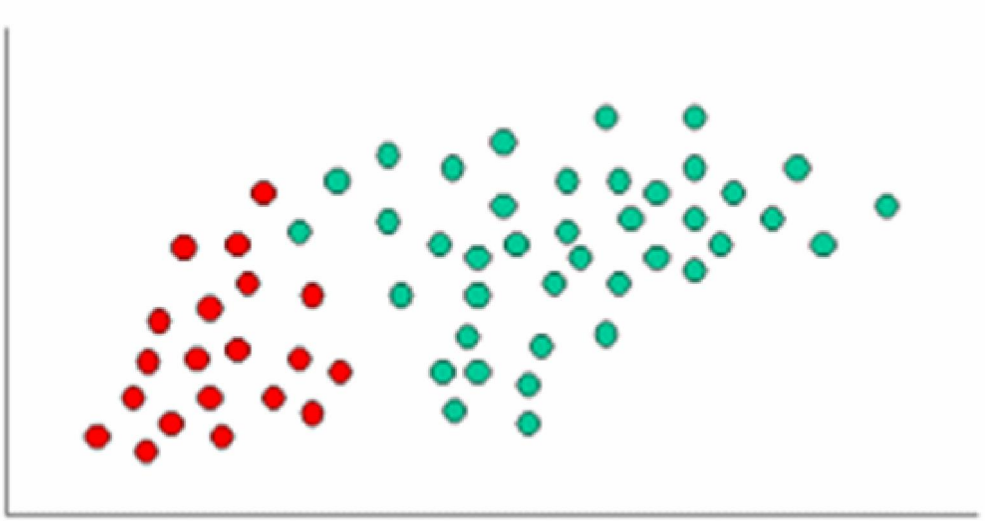
考察互信息

- 根据H(X|Y) = H(X) - I(X,Y)
- 得到I(X,Y) = H(X) - H(X|Y)
- I(X,Y)：在X中包含的关于Y的信息



k近邻分类





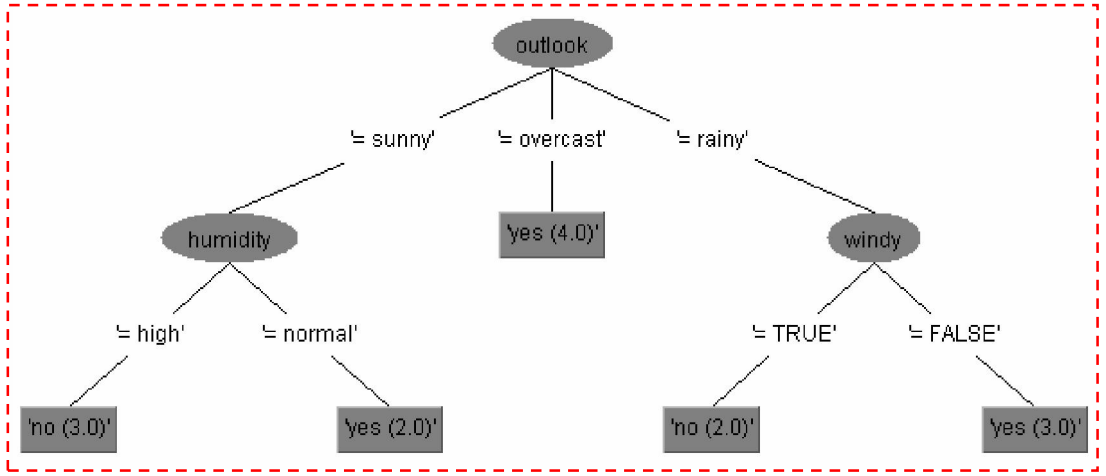
决策树的实例(Weka自带测试数据)

Viewer

Relation: weather.symbolic

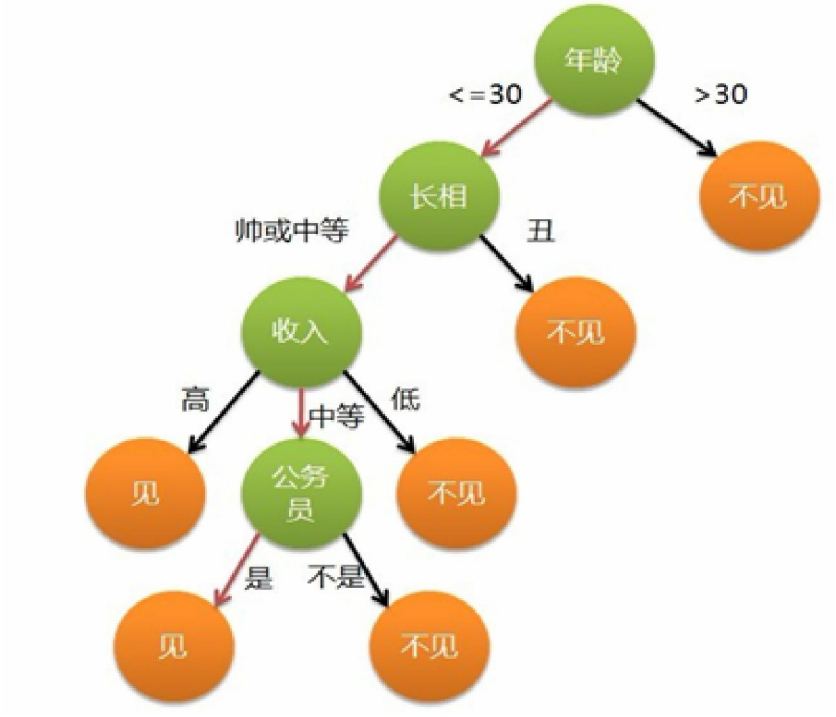
No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

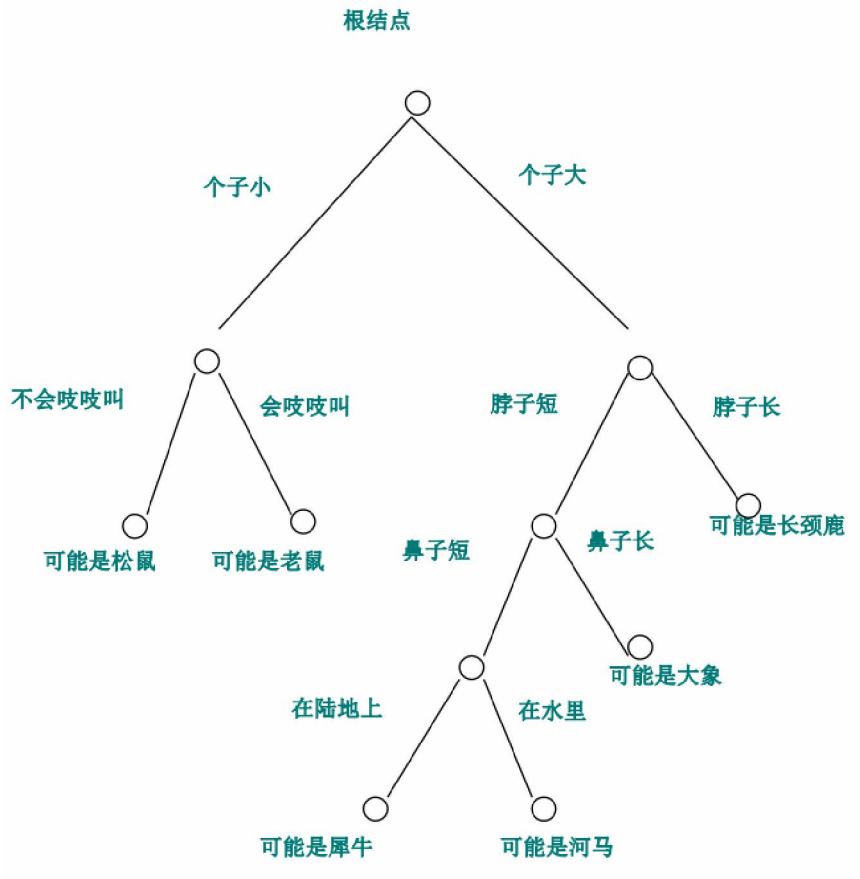
Undo OK Cancel



注：Weka的全名是怀卡托智能分析环境(Waikato Environment for Knowledge Analysis)，是一款免费的，非商业化(与之对应的是SPSS公司商业数据挖掘产品--Clementine)的，基于JAVA环境下开源的机器学习(machine learning)以及数据挖掘(data minining)软件。它和它的源代码可在其官方网站下载。

决策树示意图


年龄
<=30
>30
长相
不见
帅或中等
丑
收入
不见
高
中等
低
见
公务员
不见
是
不是
见
不见


根结点
个子小
个子大
不会吱吱叫
会吱吱叫
脖子短
脖子长
可能是松鼠
可能是老鼠
鼻子短
鼻子长
可能是长颈鹿
可能是大象
在陆地上
在水里
可能是犀牛
可能是河马

决策树 (Decision Tree)

- 决策树是一种树型结构，其中每个内部结点表示在一个属性上的测试，每个分支代表一个测试输出，每个叶结点代表一种类别。
- 决策树学习是以实例为基础的归纳学习。
- 决策树学习采用的是自顶向下的递归方法，其基本思想是以信息熵为度量构造一棵熵值下降最快的树，到叶子节点处的熵值为零，此时每个叶节点中的实例都属于同一类。

决策树学习算法的特点

- 决策树学习算法的最大优点是，它可以自学习。在学习的过程中，不需要使用者了解过多背景知识，只需要对训练实例进行较好的标注，就能够进行学习。
  - 显然，属于有监督学习。
  - 从一类无序、无规则的事物(概念)中推理出决策树表示的分类规则。

决策树学习的生成算法

- 建立决策树的关键，即在当前状态下选择哪个属性作为分类依据。根据不同的目标函数，建立决策树主要有一下三种算法。
  - ID3
  - C4.5
  - CART

信息增益

- 概念：当熵和条件熵中的概率由数据估计(特别是极大似然估计)得到时，所对应的熵和条件熵分别称为经验熵和经验条件熵。
- 信息增益表示得知特征A的信息而使得类X的信息的不确定性减少的程度。
- 定义：特征A对训练数据集D的信息增益g(D,A)，定义为集合D的经验熵H(D)与特征A给定条件下D的经验条件熵H(D|A)之差，即：
  - $g(D,A)=H(D)-H(D|A)$
  - 显然，这即为训练数据集D和特征A的互信息。

基本记号

- 设训练数据集为D，|D|表示其容量，即样本个数。设有K个类C_k，k=1,2,…,K，$|C_k|$为属于类C_k的样本个数。$\Sigma_k|C_k|=|D|$。设特征A有n个不同的取值$\{a_1,a_2\ldots a_n\}$，根据特征A的取值将D划分为n个子集$D_1,D_2,\ldots D_n$,$|D_i|$为Di的样本个数，$\Sigma_i|D_i|=D$。记子集D_i中属于类C_k的样本的集合为D_{ik}，$|D_{ik}|$为D_{ik}的样本个数。

信息增益的计算方法

- 计算数据集D的经验熵 $H(D)=-\sum_{k=1}^{K}\frac{C_k}{D}\log\frac{C_k}{D}$
- 计算特征A对数据集D的经验条件熵H(D|A)
- 计算信息增益：g(D,A)=H(D) – H(D|A)

经验条件熵H(D|A)

$$H(D \mid A) = -\sum_{i,k} p(D_k, A_i) \log p(D_k \mid A_i)$$

$$= -\sum_{i,k} p(A_i) p(D_k \mid A_i) \log p(D_k \mid A_i)$$

$$= -\sum_{i=1}^{n} \sum_{k=1}^{K} p(A_i) p(D_k \mid A_i) \log p(D_k \mid A_i)$$

$$= -\sum_{i=1}^{n} p(A_i) \sum_{k=1}^{K} p(D_k \mid A_i) \log p(D_k \mid A_i)$$

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

其他目标

- 信息增益率：$g_r(D,A) = g(D,A) / H(A)$
- 基尼指数：

$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

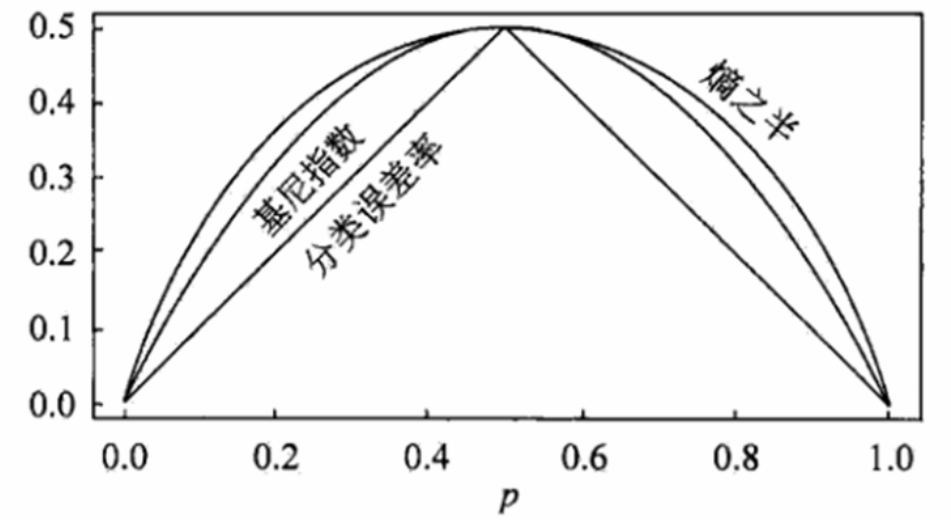
$$= 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{D} \right)^2$$

讨论(一家之言)

- 考察基尼指数的图像、熵、分类误差率三者之间的关系
  - 将f(x)=-lnx在x0=1处一阶展开，忽略高阶无穷小，得到f(x)≈1-x

$$H(X) = -\sum_{k=1}^{K} p_k \ln p_k$$

$$\approx \sum_{k=1}^{K} p_k (1 - p_k)$$

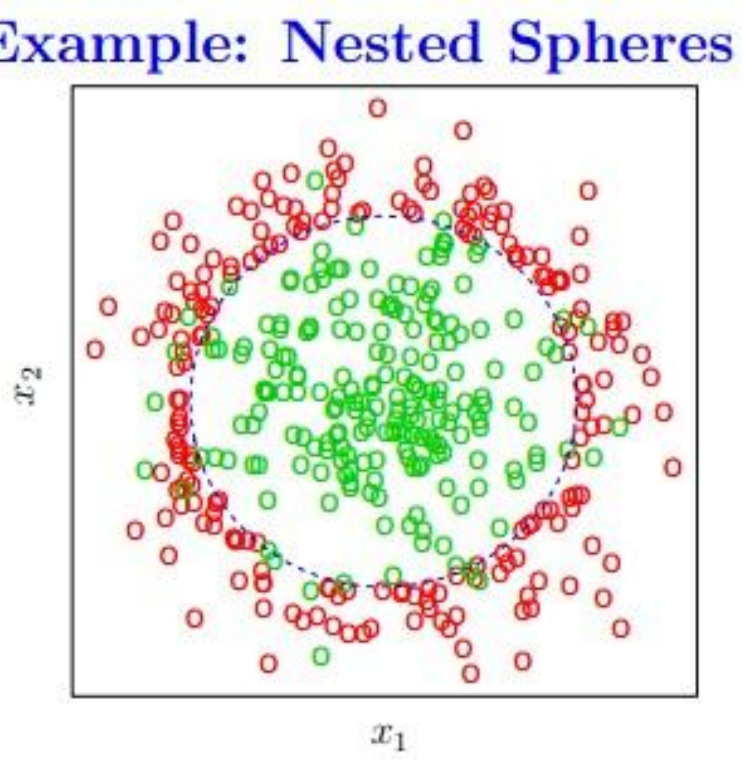


三种决策树学习算法

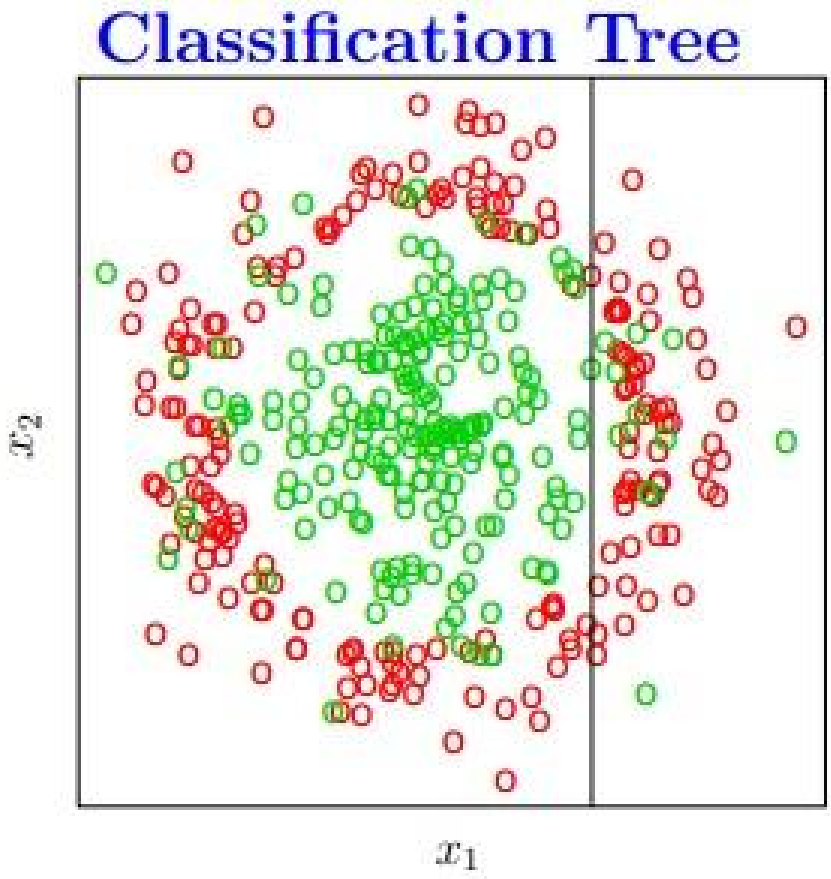
- 适应信息增益来进行特征选择的决策树学习过程，即为ID3决策。
- 所以如果是取值更多的属性，更容易使得数据更“纯”，其信息增益更大，决策树会首先挑选这个属性作为树的顶点。结果训练出来的形状是一棵庞大且深度很浅的树，这样的划分是极为不合理的。
- C4.5：信息增益率 $g_r(D,A) = g(D,A) / H(A)$
- CART：基尼指数
- 总结：一个属性的信息增益越大，表明属性对样本的熵减少的能力更强，这个属性使得数据由不确定性变成确定性的能力越强。

决策树的例子

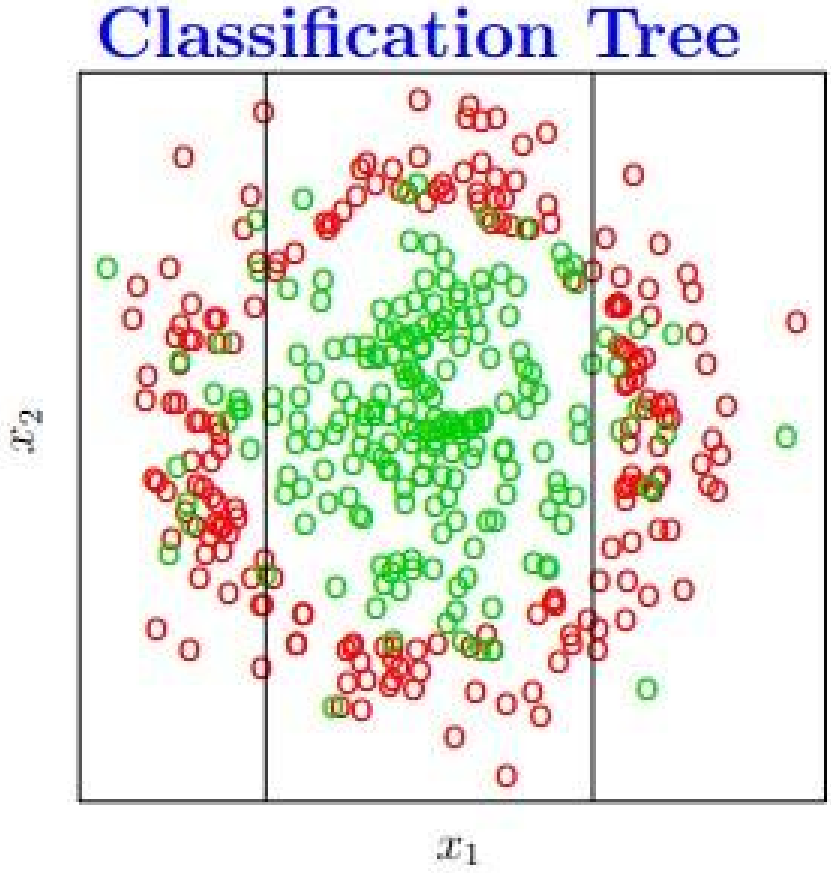
- 对于下面的数据，希望分割成红色和绿色两个类



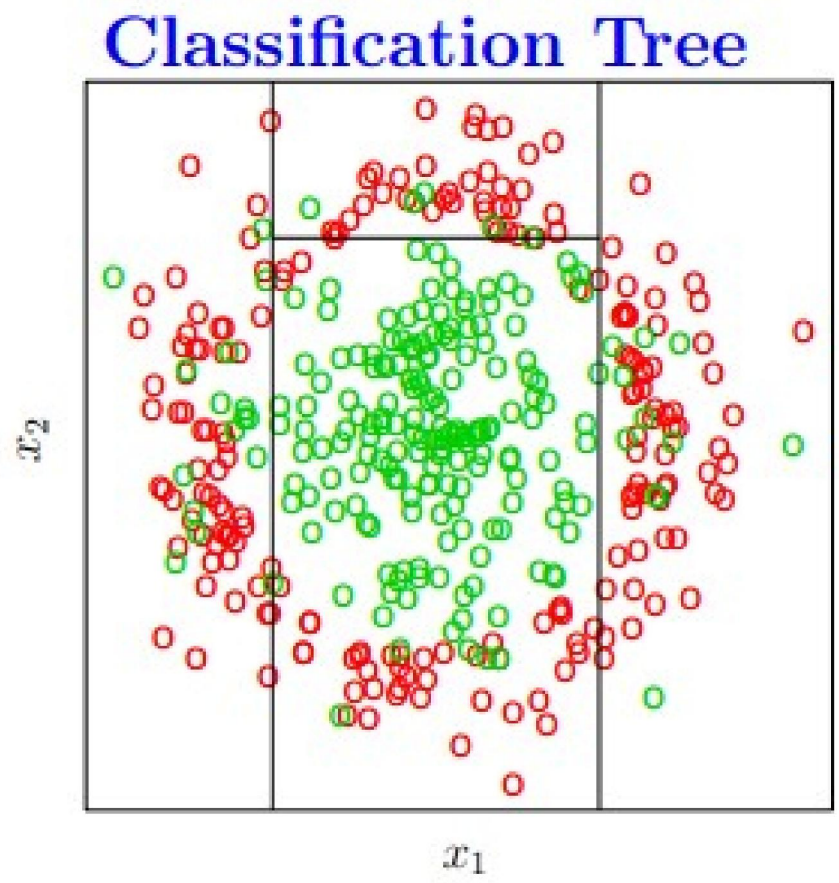
决策树的生成过程



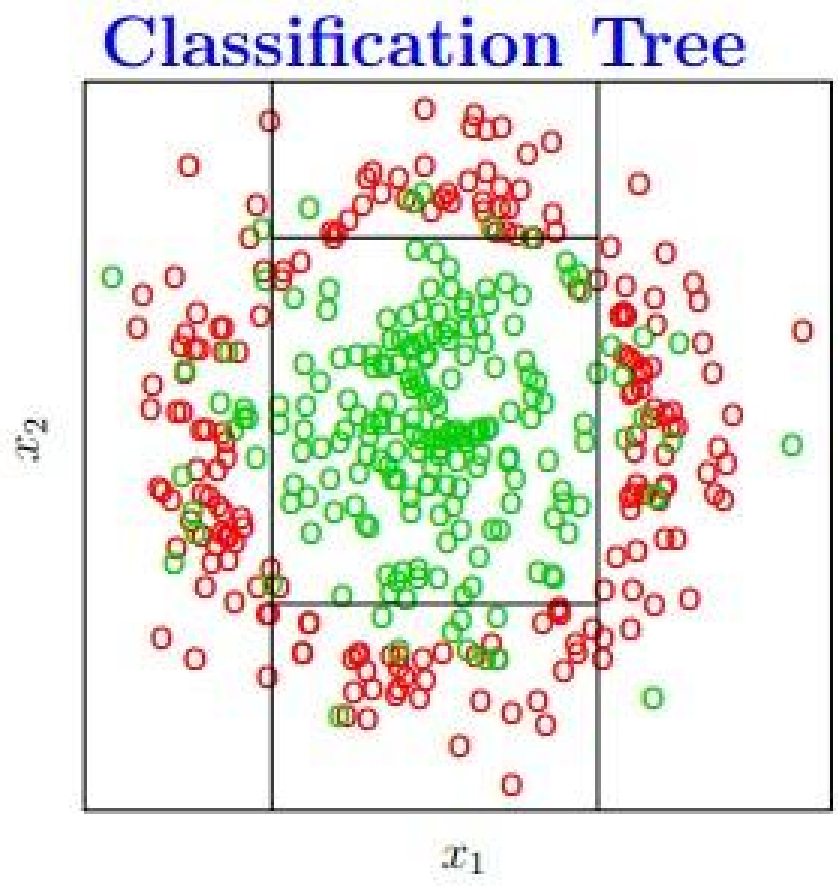
决策树的生成过程



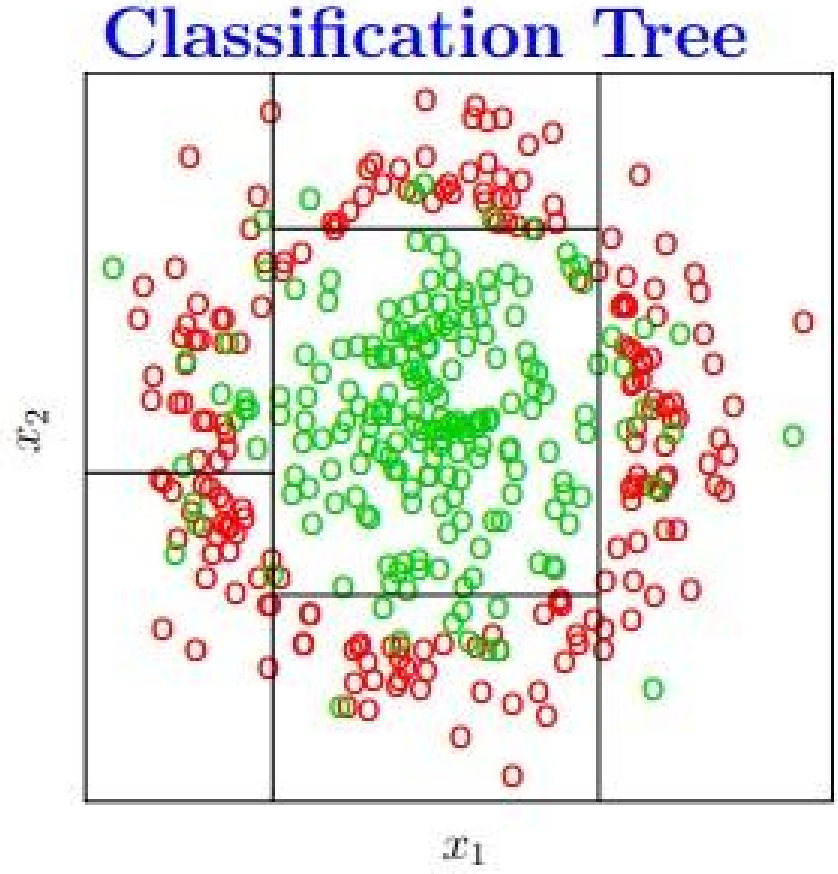
决策树的生成过程



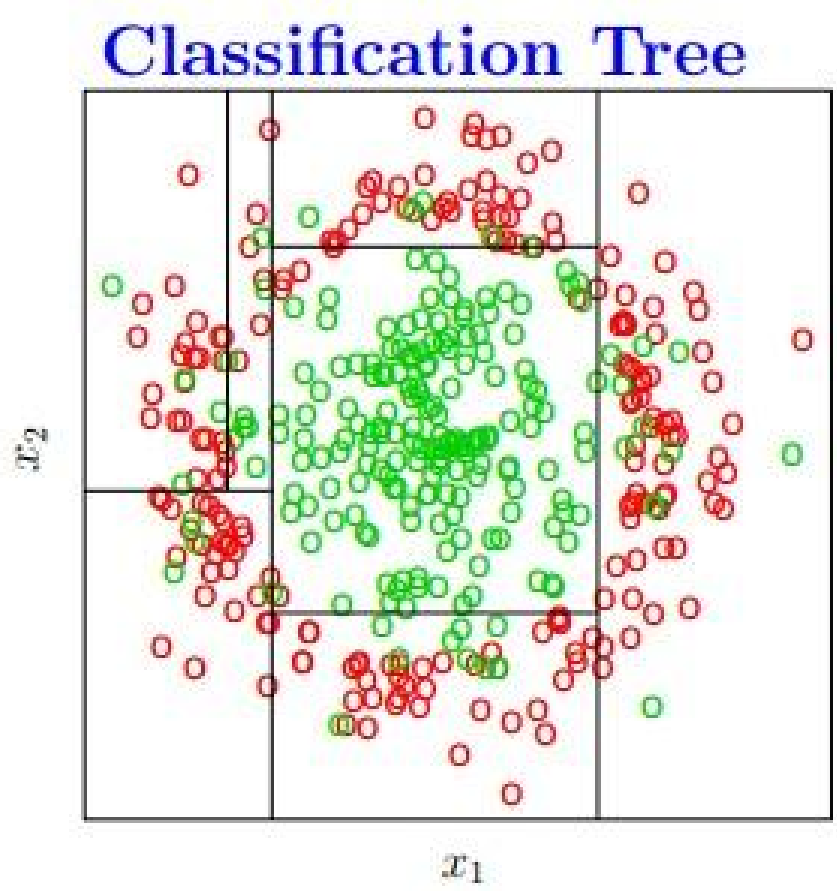
决策树的生成过程



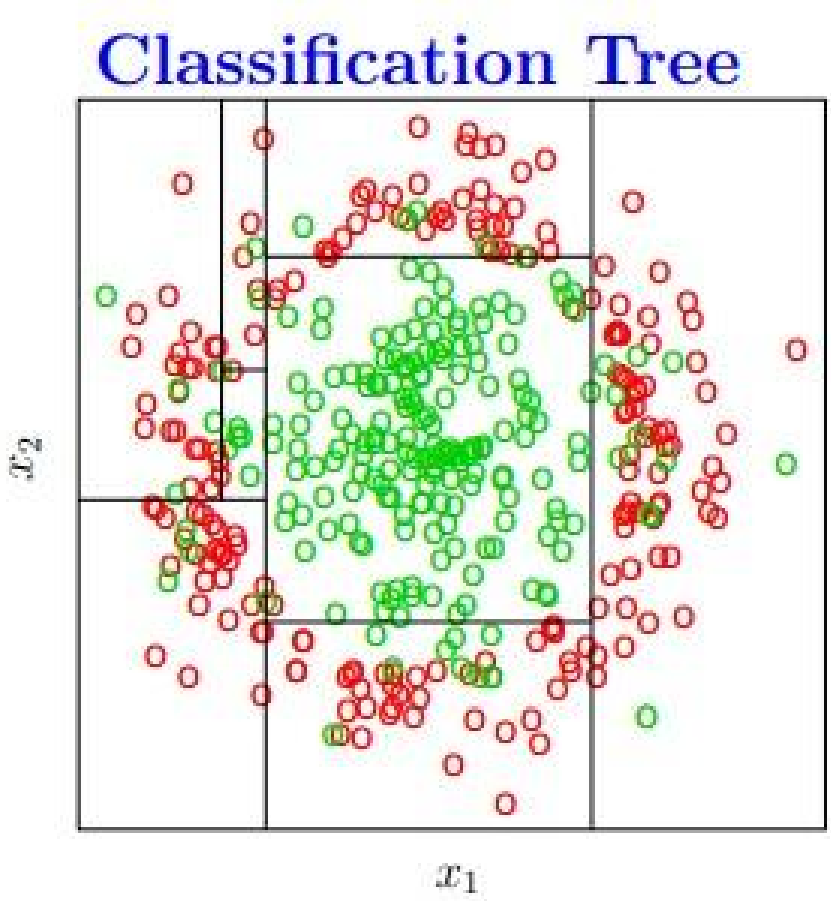
决策树的生成过程



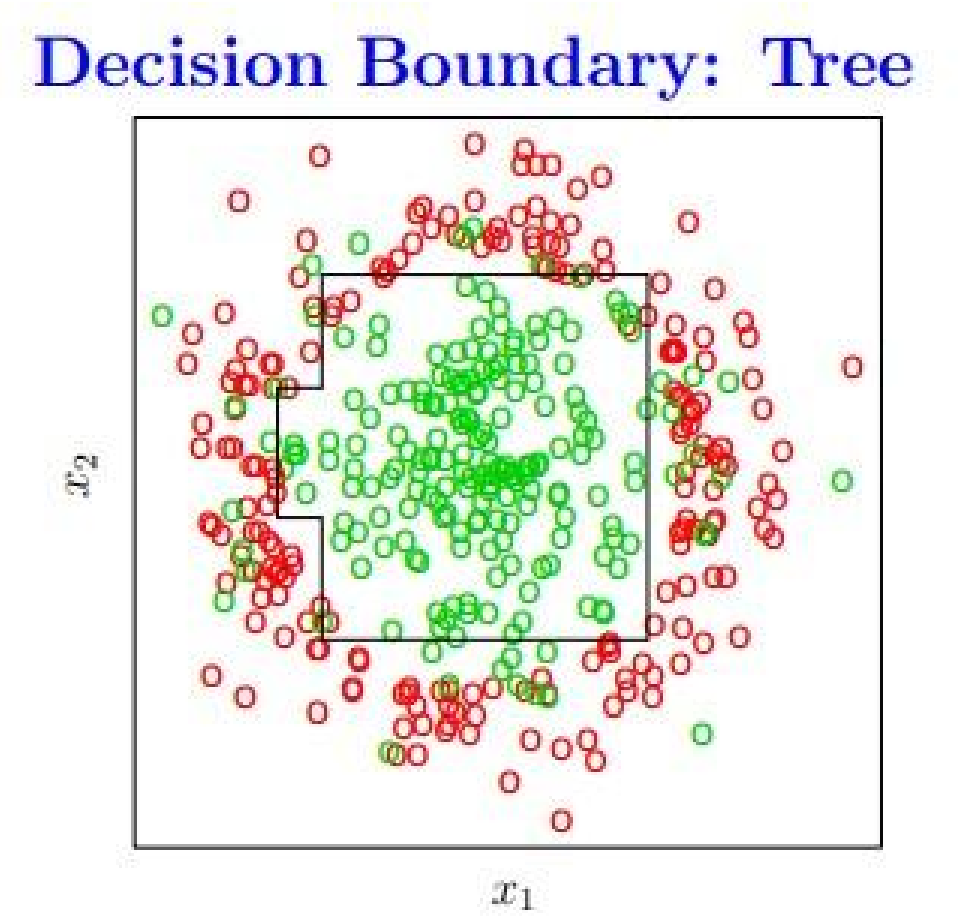
决策树的生成过程



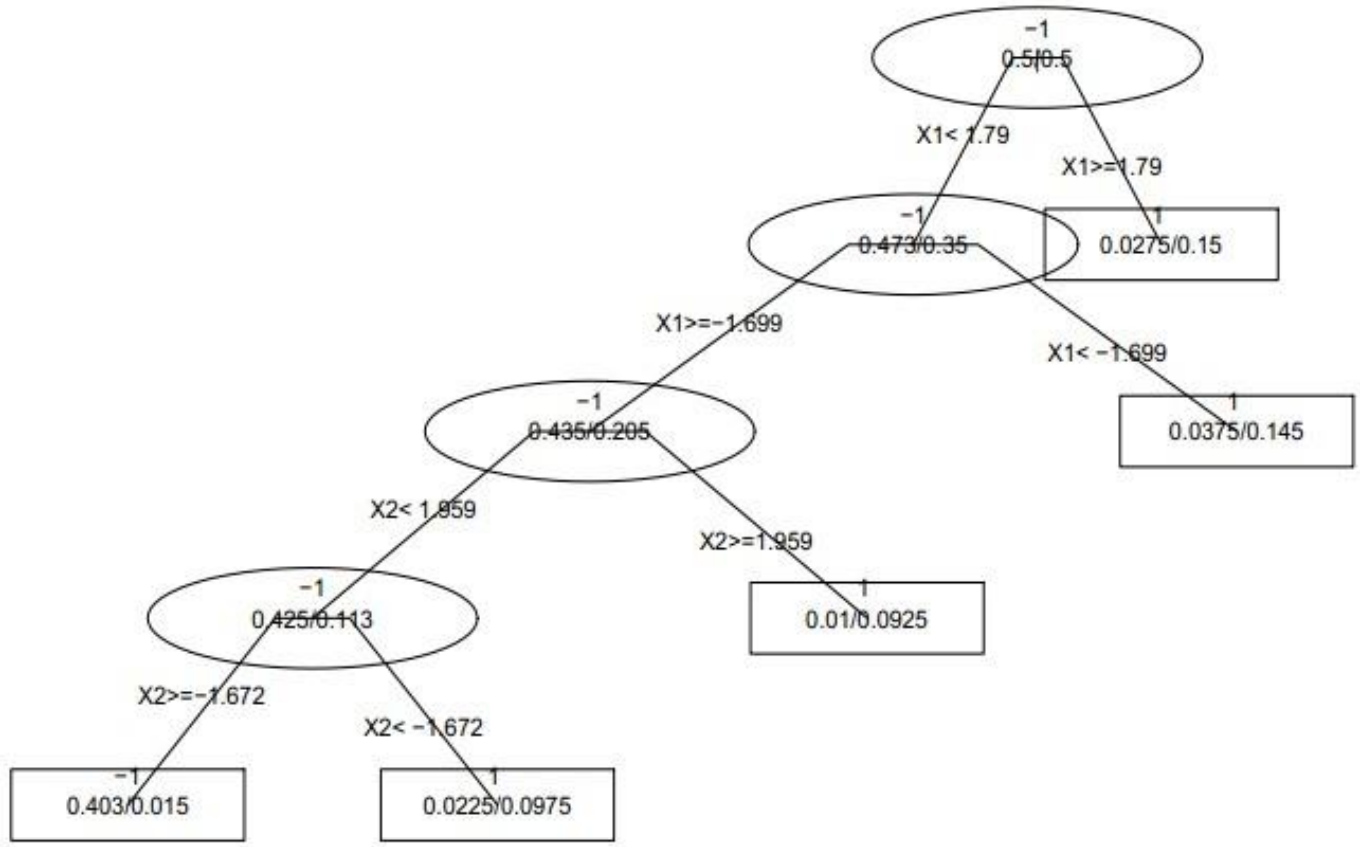
决策树的生成过程



决策树的生成过程



决策树的生成过程


-1
0.5/0.5
X1< 1.79
X1>=1.79
-1
0.473/0.35
1
0.0275/0.15
X1>=-1.699
X1< -1.699
-1
0.435/0.205
1
0.0375/0.145
X2< 1.959
X2>=1.959
-1
0.425/0.113
1
0.01/0.0925
X2>=-1.672
X2< -1.672
-1
0.403/0.015
1
0.0225/0.0975

决策树的过拟合

- 决策树对训练属于有很好的分类能力，但对未知的测试数据未必有好的分类能力，泛化能力弱，即可能发生过拟合现象。
  - 剪枝
  - 随机森林

Bootstraping

- Bootstraping的名称来自成语“pull up by your own bootstraps”，意思是依靠你自己的资源，称为自助法，它是一种有放回的抽样方法。

  - 注：Bootstrap本义是指高靴子口后面的悬挂物、小环、带子，是穿靴子时用手向上拉的工具。“pull up by your own bootstraps”即“通过拉靴子让自己上升”，意思是“不可能发生的事情”。后来意思发生了转变，隐喻“不需要外界帮助，仅依靠自身力量让自己变得更好”。

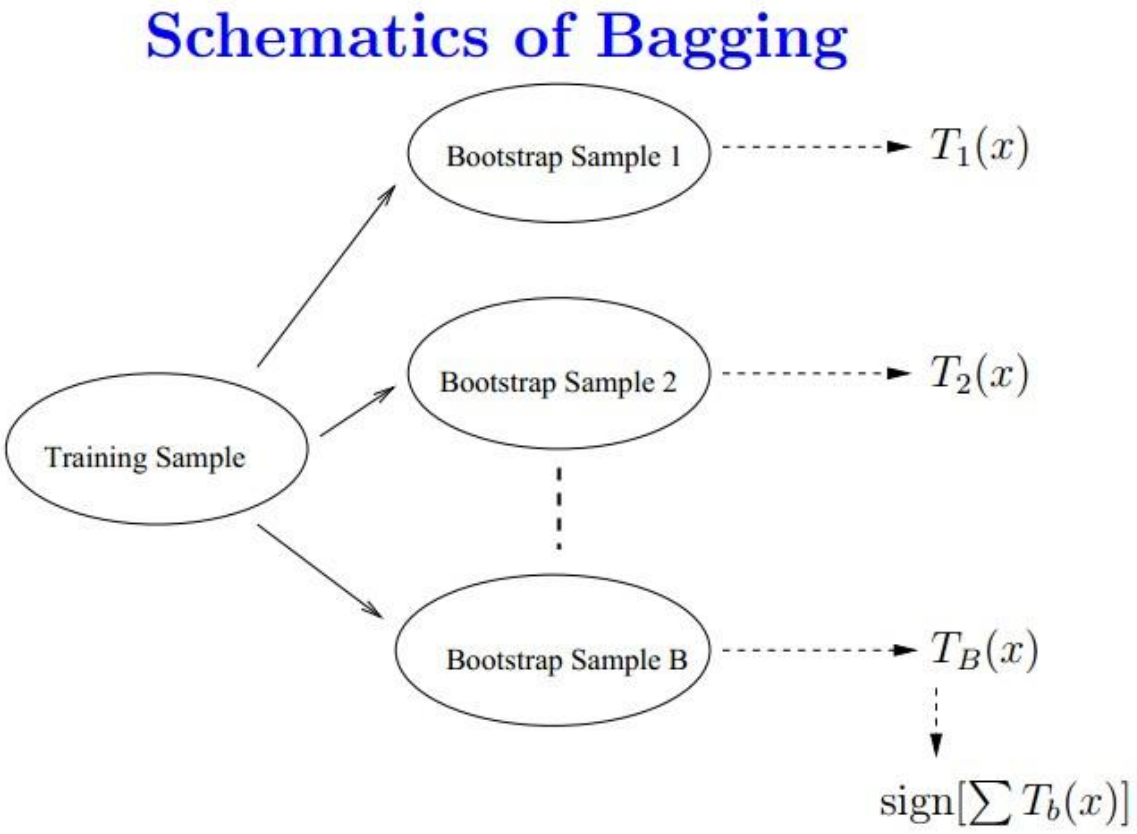
Bagging的策略

- bootstrap aggregation
- 从样本集中重采样(有重复的)选出n个样本
- 在所有属性上，对这n个样本建立分类器(ID3、C4.5、CART、SVM、Logistic回归等)
- 重复以上两步m次，即获得了m个分类器
- 将数据放在这m个分类器上，最后根据这m个分类器的投票结果，决定数据属于哪一类

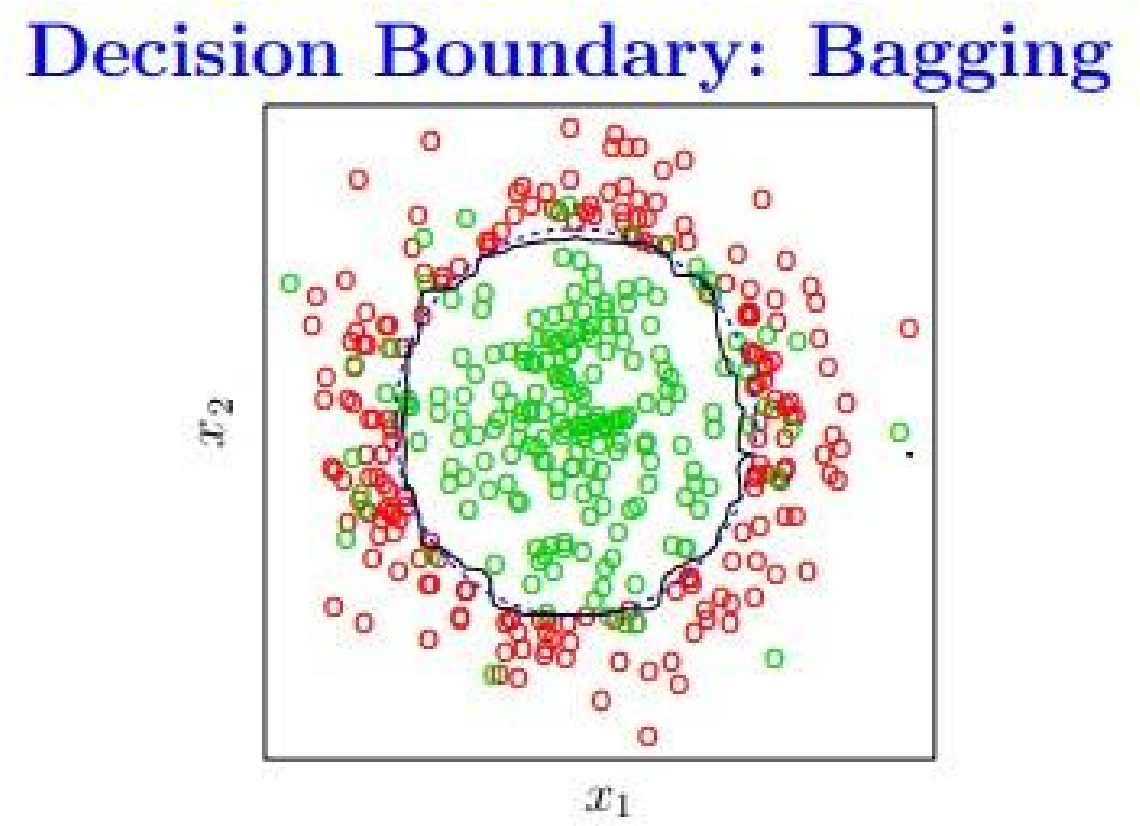
Another description of Bagging

Given a standard training set D of size n, bagging generates m new training sets D_i, each of size n' n, by sampling examples from D uniformly and with replacement. By sampling with replacement, it is likely that some examples will be repeated in each D_i. If $n'=n$, then for large n the set D_i is expected to have the fraction (1 - 1/e) (≈63.2%) of the unique examples of D, the rest being duplicates.[1] This kind of sample is known as a bootstrap sample. The m models are fitted using the above m bootstrap samples and combined by averaging the output (for regression) or voting (for classification).

Bagging


Schematics of Bagging
Bootstrap Sample 1
$T_1(x)$
Bootstrap Sample 2
$T_2(x)$
Training Sample
Bootstrap Sample B
$T_B(x)$
$\text{sign}[\sum T_b(x)]$

Bagging的结果



随机森林

- 随机森林在bagging基础上做了修改。
  - 从样本集中用Bootstrap采样选出n个样本；
  - 从所有属性中随机选择k个属性，选择最佳分割属性作为节点建立CART决策树；
  - 重复以上两步m次，即建立了m棵CART决策树
  - 这m个CART形成随机森林，通过投票表决结果，决定数据属于哪一类

思考

- 下图是实际B超拍摄的胎儿影像。设计算法完成头骨的自动检测，从而进一步估算胎儿头骨直径。
- 假定现在有已经标记的几千张的不同胎儿的图像，对于新的一张图像，如何做自动检测和计算？

一种可行的解决方案

- 通过Haar特征提取等对每幅图片分别处理，得到M个特征。N个图片形成N*M的矩阵。
- 随机选择若干特征和样本，得到a*b的小矩阵，建立决策树；
- 重复K次得到随机森林。
- 投票方法选择少数服从多数。

应用实例：Kinect

3.3. Randomized decision forests

Decision forests are considered effective multi-class classifiers. Forest is a group of T decision trees, where each split node is represented by a feature and a threshold τ. The procedure starts at the root node of a tree, evaluating equation 1 at each split node and branching left or right according to the comparison to threshold τ. The leaf node consists of a learned distribution $P_t(c|I,x)$ over body part label c, in the tree t. Figure IV illustrates the forest approach.

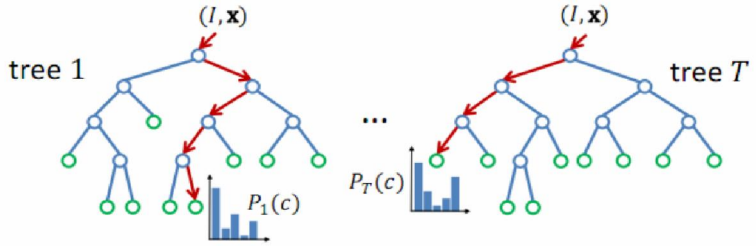
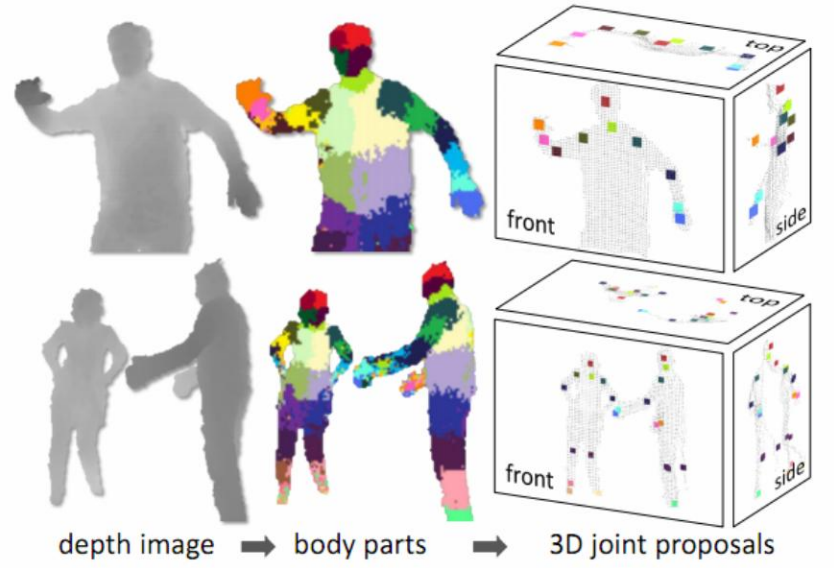


Figure IV. Decision forest.

The final classification is given by averaging all the distributions together in the forest. Equation 2 represents this classification.

$$P(c|I,x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,x) \quad (2)$$

The final classification is given by averaging all the distributions together in the forest. Equation 2 represents this classification.



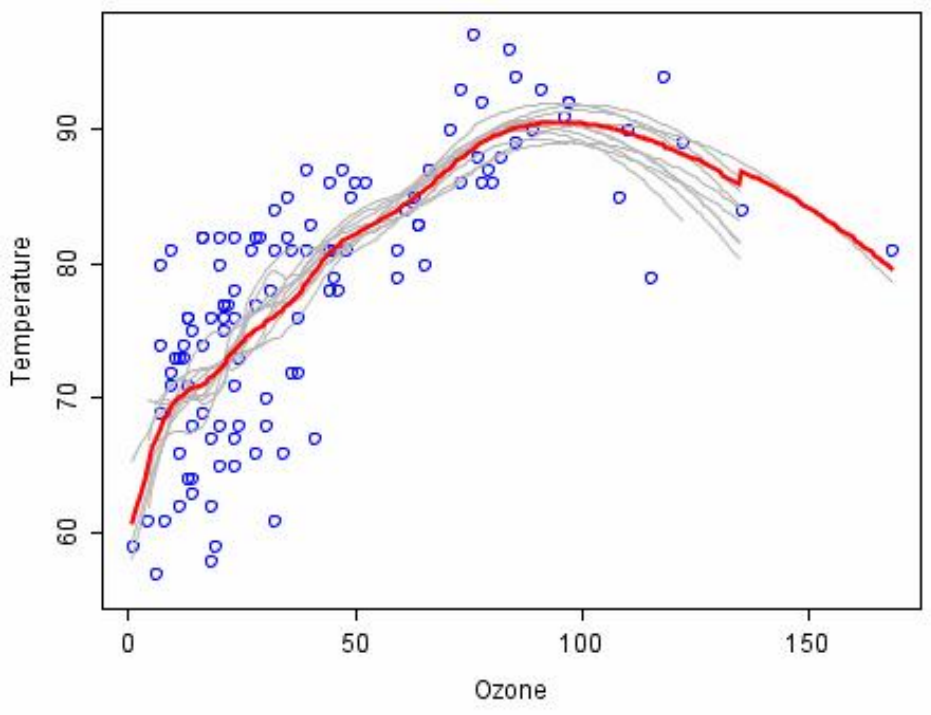
Real-Time Human Pose Recognition in Parts from Single Depth Images, Jamie Shotton etc,2001,

随机森林/Bagging和决策树的关系

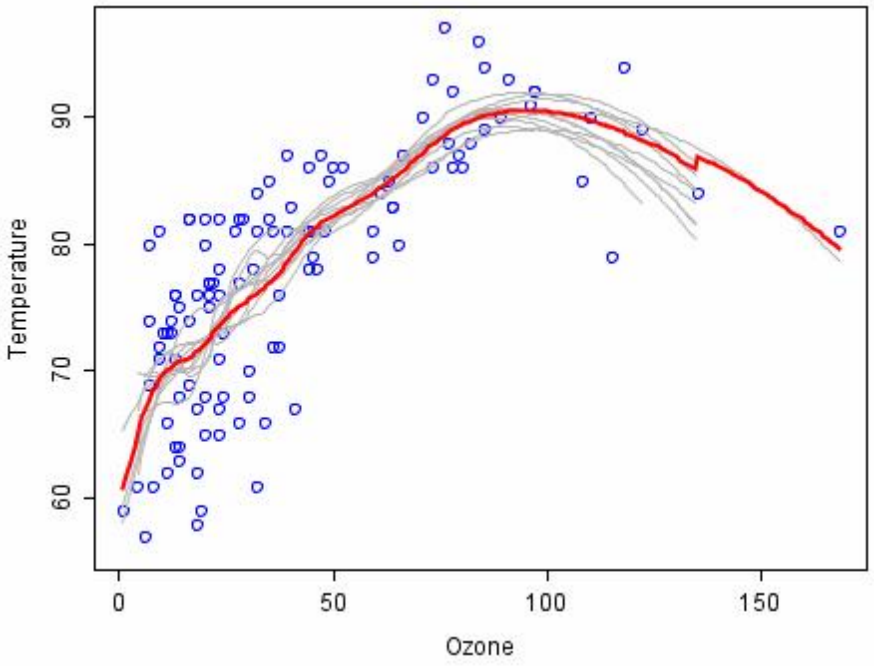
- 当然可以使用决策树作为基本分类器
- 但也可以使用SVM、Logistic回归等其他分类器，习惯上，这些分类器组成的“总分类器”，仍然叫做随机森林。
- 举例
  - 回归问题

回归问题

- 离散点是样本集合，描述了臭氧(横轴)和温度(纵轴)的关系
- 试拟合二者的变化曲线



使用Bagging



记原始数据为D，长度为N(即图中有N个离散点)

- 算法过程
  - 做100次bootstrap，每次得到的数据Di，Di的长度为N
  - 对于每一个Di，使用局部回归(LOESS)拟合一条曲线(图中灰色线是其中的10条曲线)
  - 将这些曲线取平均，即得到红色的最终拟合曲线
  - 显然，红色的曲线更加稳定，并且没有过拟合明显减弱

附：局部加权线性回归

- LWR：Locally Weighted linear Regression
- LOESS : LOcal regrESSion

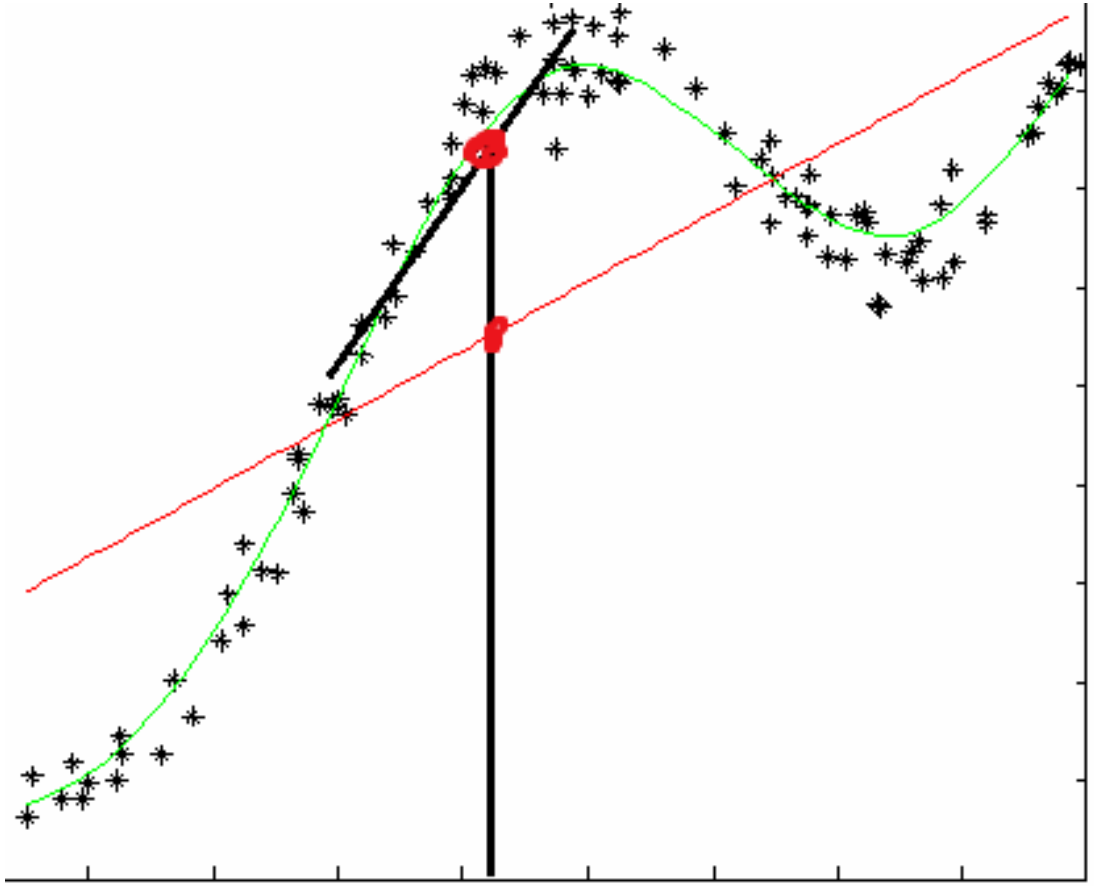
1. Fit θ to minimize $\sum_i (y^{(i)} - \theta^T x^{(i)})^2$
2. Output $\theta^T x$.

1. Fit θ to minimize $\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$
2. Output $\theta^T x$.

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

附：线性回归与局部加权回归

- 黑色是样本点
- 红色是线性回归曲线
- 绿色是局部加权回归曲线



投票机制

- 简单投票机制
  - 一票否决(一致表决)
  - 少数服从多数
    - 有效多数(加权)
  - 阈值表决
- 贝叶斯投票机制

贝叶斯投票机制

- 简单投票法假设每个分类器都是平等的。
- 在实际生活中，我们听取一个人的意见，会考虑这个人过去的意见是否有用，从而加大或者减少权值。
- 贝叶斯投票机制基于每个基本分类器在过去的分类表现设定一个权值，然后按照这个权值进行投票。

投票机制举例

- 假定有N个用户可以为X个电影投票(假定投票者不能给同一电影重复投票)，投票有1、2、3、4、5星共5档。
- 如何根据用户投票，对电影排序？
  - 本质仍然是分类问题：对于某个电影，有N个决策树，每个决策树对该电影有1个分类(1、2、3、4、5类)，求这个电影应该属于哪一类(可以是小数：分类问题变成了回归问题)

一种可能的方案

$$WR = \frac{v}{v+m}R + \frac{m}{v+m}C$$

- WR：加权得分(weighted rating)
- R：该电影的用户投票的平均得分(Rating)
- C：所有电影的平均得分
- v：该电影的投票人数(votes)
- m：排名前250名的电影的最低投票数
  - 根据总投票人数，250可能有所调整
  - 按照v=0和m=0分别分析

朝花夕拾

- 谱聚类

拉普拉斯矩阵的定义

- 计算点之间的邻接相似度矩阵W
  - 若两个点的相似度值越大，表示这两个点越相似；
  - 同时，定义$w_{ij}=0$表示v_i,v_j两个点没有任何相似性(无穷远)
- W的第i行元素的和为v_i的度。形成顶点度对角阵D
  - d_{ii}表示第i个点的度
  - 除主对角线元素，D其他位置为0
- 未正则的拉普拉斯矩阵：L=D-W
- 正则拉普拉斯矩阵
  - 对称拉普拉斯矩阵 $L_{\text{sym}} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$
  - 随机游走拉普拉斯矩阵 $L_{\text{rw}} := D^{-1}L = I - D^{-1}W$
    - Random walk

谱聚类算法：未正则拉普拉斯矩阵

- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算n×n的相似度矩阵W和度矩阵D；
  - 计算拉普拉斯矩阵L=D-W；
  - 计算L的前k个特征向量$u_1,u_2,...,u_k$；
  - 将k个列向量$u_1,u_2,...,u_k$组成矩阵U，$U \in R^{n \times k}$；
  - 对于i=1,2,...,n,令$y_i \in R^k$是U的第i行的向量；
  - 使用k-means算法将点$(y_i)_{i=1,2,...,n}$聚类成簇$C_1,C_2,...C_k$；
  - 输出簇$A_1,A_2,...A_k$，其中，$Ai=\{j|y_j \in Ci\}$

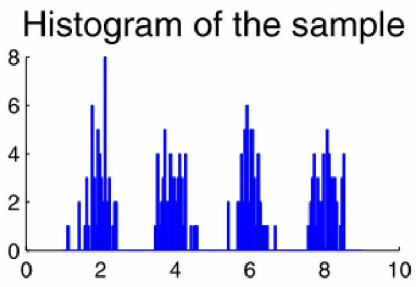
谱聚类算法：随机游走拉普拉斯矩阵

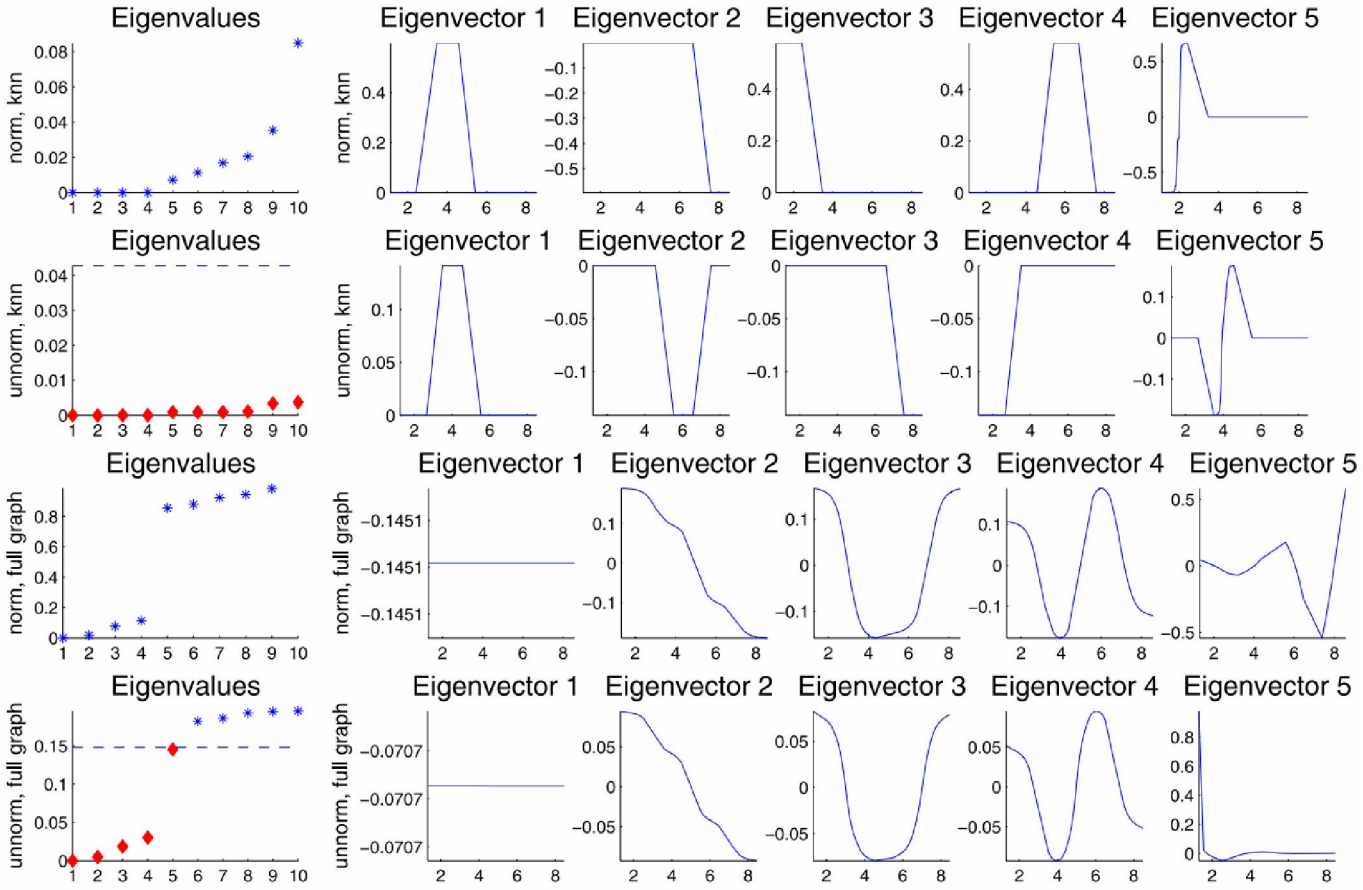
- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算n×n的相似度矩阵W和度矩阵D；
  - 计算正则拉普拉斯矩阵$L_{rw}=D^{-1}(D-W)$；
  - 计算L_{rw}的前k个特征向量$u_1,u_2,...,u_k$；
  - 将k个列向量$u_1,u_2,...,u_k$组成矩阵U，$U \in R^{n\times k}$；
  - 对于i=1,2,...,n,令$y_i \in R^k$是U的第i行的向量；
  - 使用k-means算法将点$(y_i)_{i=1,2,...,n}$聚类成簇$C_1,C_2,...C_k$；
  - 输出簇$A_1,A_2,...A_k$，其中，$Ai=\{j|y_j \in Ci\}$

谱聚类算法：对称拉普拉斯矩阵

- 输入：n个点$\{p_i\}$，簇的数目k
  - 计算n×n的相似度矩阵W和度矩阵D；
  - 计算正则拉普拉斯矩阵$L_{sym}=D^{-1/2}(D-W)\ D^{-1/2}$；
  - 计算L_{sym}的前k个特征向量$u_1,u_2,...,u_k$；
  - 将k个列向量$u_1,u_2,...,u_k$组成矩阵U，$U \in R^{n\times k}$；
  - 对于i=1,2,...,n,令$y_i \in R^k$是U的第i行的向量；
  - 对于i=1,2,...,n,将$y_i \in R^k$依次单位化，使得$|y_i|=1$；
  - 使用k-means算法将点$(y_i)_{i=1,2,...,n}$聚类成簇$C_1,C_2,...C_k$；
  - 输出簇$A_1,A_2,...A_k$，其中，$Ai=\{j|y_j \in Ci\}$

一个实例


Histogram of the sample


Eigenvalues
Eigenvector 1
Eigenvector 2
Eigenvector 3
Eigenvector 4
Eigenvector 5
norm, knn
unnorm, knn
norm, full graph
unnorm, full graph

切割图

- 聚类问题的本质：
- 对于定值k和图G，选择一组划分：$A_1,A_2,...,Ak$，最小化下面的式子：

$$\mathrm{cut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k} W(A_i,\bar{A}_i)$$

修正目标函数

- 上述的目标函数存在问题：在很多情况下，minCut的解，将图分成了一个点和其余的n-1个点。为了避免这个问题，目标函数应该显示的要求$A_1,A_2,...,Ak$足够大。

$$\text{RatioCut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{|A_i|} = \sum_{i=1}^{k}\frac{\text{cut}(A_i,\bar{A}_i)}{|A_i|}$$

$$\text{Ncut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k}\frac{\text{cut}(A_i,\bar{A}_i)}{\text{vol}(A_i)}$$

分析分母对目标函数的影响

$$\mathrm{RatioCut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{|A_i|} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i,\bar{A}_i)}{|A_i|}$$

$$\mathrm{Ncut}(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)} = \sum_{i=1}^{k}\frac{\mathrm{cut}(A_i,\bar{A}_i)}{\mathrm{vol}(A_i)}$$

- 上述目标函数以Ai的点数或者权值作为被除数，使得函数 $\sum_{i=1}^{k}\frac{1}{|A_i|}$ 的最小值在$|A_i|$相等的时候达到；函数 $\sum_{i=1}^{k}\frac{1}{vol(A_i)}$ 的最小值在$\mathrm{vol}(A_i)$相等的时候达到。从而，目标函数能够试图得到“平衡”的簇。
  - 带等式约束的极值问题，约束条件：

$$\sum_{i=1}^{k}|A_i| = n \qquad \sum_{i=1}^{k}vol(A_i) = sum(D)$$

当k=2时的RatioCut

- 目标函数：$\min_{A \subset V} \text{RatioCut}(A, \bar{A})$
- 定义向量f=(f1,f2,...fn)^T，
  - 它其实是分割子图的指示向量

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } v_i \in \bar{A} \end{cases}$$

RatioCut与拉普拉斯矩阵的关系

$$
\begin{aligned}
f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\sum_{i\in A, j\in \bar{A}} w_{ij}\left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 \\
&\quad + \frac{1}{2}\sum_{i\in \bar{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 \\
&= \mathrm{cut}(A, \bar{A})\left(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2\right) \\
&= \mathrm{cut}(A, \bar{A})\left(\frac{|A| + |\bar{A}|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|}\right) \\
&= |V| \cdot \mathrm{RatioCut}(A, \bar{A}).
\end{aligned}
$$

目标函数

$$\min_{A \subset V} f' L f, \quad f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & \text{if } v_i \in \bar{A} \end{cases}$$

- 该目标函数的自变量部分，是不同的子图划分；从而得到不同的f指示向量。优化的目标，是使得该目标函数取值最小。
  - 由于f只能取2个值，离散化的定义域导致问题是NP的。

考察f的性质

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i\in \bar{A}}\sqrt{\frac{|A|}{|\bar{A}|}} = |A|\sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}|\sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

- 上式可以看做f和全1向量的点乘，从而 $f \perp 1$

$$\|f\|^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\bar{A}|}{|A|} + |\bar{A}|\frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n$$

- 上式说明，f的模是定值。

目标函数约束条件的放松relaxation

- 将f的严格定义用f的性质代替，向量f各个分量的取值从离散若干个值延拓到整个实数域，从而得到：

$$\min_{f \in \mathbb{R}^n} f'Lf, \text{ s.t. } f \perp \mathbb{1}, \|f\| = \sqrt{n}$$

进一步

$$\min_{f \in \mathbb{R}^n} f'Lf, \ \text{s.t.}\ f \perp \mathbb{1}, \ \|f\| = \sqrt{n}$$

- 根据Rayleigh-Ritz定理，该目标函数的解即为L的次小特征向量。

推广上述结论：将子图划分从2扩展到k

The relaxation of the RatioCut minimization problem in the case of a general value k follows a similar principle as the one above. Given a partition of V into k sets $A_1, \ldots, A_k$, we define k indicator vectors $h_j = (h_{1,j}, \ldots, h_{n,j})'$ by

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j, \\ 0 & \text{otherwise} \end{cases}$$

$$(i = 1, \ldots, n;\ j = 1, \ldots, k)$$

推广上述结论：考察指示向量组成的矩阵

Then we set the matrix $H \in \mathbb{R}^{n \times k}$ as the matrix containing those k indicator vectors as columns. Observe that the columns in H are orthonormal to each other, that is $H'H = I$. Similar we can see that

$$h_i' L h_i = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

$$h_i' L h_i = (H' L H)_{ii}$$

推广上述结论：目标函数

$$\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} h_i' L h_i = \sum_{i=1}^{k} (H' L H)_{ii} = \text{Tr}(H' L H)$$

$$\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' L H), \text{ s.t. } H' H = I$$

- 根据Rayleigh-Ritz定理，L的前k个特征向量即为$h_1, h_2, \ldots, h_k$。

进一步思考

- 谱聚类中的K，如何确定？
  - 考察计算得到的各个特征值λ：选择k，使得$\lambda_1, \lambda_2, \ldots, \lambda_k$很小，而$\lambda_{k+1}$相对比较大。
- 最后一步的K-Means，作用是什么？
  - 事实上，目标函数是关于子图划分指示向量的函数，该向量的值根据子图划分确定，是离散的若干值。但由于问题是NP的，转换成求连续实数域上的解，最后再用K-Means的办法离散化。
  - 可以用其他方法代替。如有人使用超平面，或者使用k个特征向量张成的子空间达到同样的目的。
- 未正则拉普拉斯矩阵、对称拉普拉斯矩阵、随机游走拉普拉斯矩阵，首选哪一个？
  - 随机游走拉普拉斯矩阵
- 拉普拉斯矩阵除了通过切割图的方法，可以通过随机游走、扰动论等理论来解释。

参考文献

- Elements of Information Theory (Cover & Thomas)
- Pattern Recognition and Machine Learning, Bishop M, Springer-Verlag, 2006
- 统计学习方法，李航著，清华大学出版社，2012年
- Jamie Shotton, Andrew Fitzgibbon, etc, Real-Time Human Pose Recognition in Parts from Single Depth Images, 2011
- Clustering by fast search and find of density peak. Alex Rodriguez, Alessandro Laio
- A tutorial on spectral clustering，Ulrike von Luxburg, 2007
- Lütkepohl H, Handbook of Matrices. Wiley, Chichester,1997
- Lang K, Fixing two weaknesses of the spectral method. In: Weiss Y, Schölkopf B, Platt J (eds.) Advances in Neural Information Processing Systems 18, pp. 715–722. MIT Press, Cambridge, 2006
- Bach F, Jordan M， Learning spectral clustering. In: Thrun S, Saul L, Schölkopf B(eds.), Advances in Neural Information Processing Systems 16 (NIPS), pp. 305–312. MIT Press, Cambridge,2004

谢谢大家！

恳请大家批评指正！