聚类

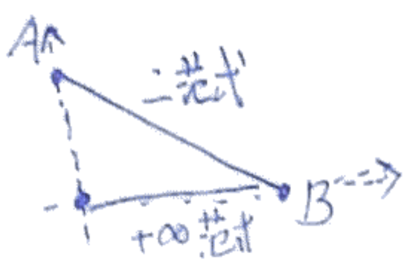
① 相似度度量函数：可用于各种聚类算法

<a> 欧式距离 (Minkowski Distance)： $dist\langle X, Y\rangle = \left(\sum_{i=1}^{N}|x_i-y_i|^p\right)^{\frac{1}{p}}$

- 1范式 ($p=1$)：退化为 $|x_1-y_1|+|x_2-y_2|+\cdots+|x_n-y_n|$
- 2范式 ($p=2$)：
  - 二维：$d=\sqrt[2]{|x_1-y_1|^2+|x_2-y_2|^2}$
  - 三维：$d=\sqrt[2]{|x_1-y_1|^2+|x_2-y_2|^2+|x_3-y_3|^2}$
- ⋮
- k范式 ($p=k$)：$(|x_1-y_1|^k+|x_2-y_2|^k+\cdots+|x_n-y_n|^k)^{\frac{1}{k}}$
- ⋮
- $+\infty$范式 ($p=+\infty$)：$\underset{p\to+\infty}{d} = \lim_{p\to+\infty}|x_1-y_1|\left[1+\left|\frac{x_2-y_2}{x_1-y_1}\right|^k+\cdots+\left|\frac{x_n-y_n}{x_1-y_1}\right|^p\right]^{\frac{1}{p}} \to |x_1-y_1|$

切比雪夫距离

(不妨认为 $|x_1-y_1|$ 值最大，求极限)



* 二范式矩阵化表示

$$d = (\vec{X}-\vec{Y})^T(\vec{X}-\vec{Y})$$

$$= (\vec{X}-\vec{Y})^T\begin{bmatrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{bmatrix}(\vec{X}-\vec{Y}) \qquad \begin{bmatrix}w_1 & & \\ & w_2 & \\ & & \ddots & \\ & & & w_n\end{bmatrix}$$

☆ 当把对角阵换成权重矩阵时，就可以做维度加权.

<b> Jaccard 相似系数

* 基于集合相似性的相似系数(假定元素权重相同)

* 有时会需要降低热门元素权值.

$$J(X,Y) = \frac{|X\cap Y|}{|X\cup Y|}$$

<c> Consine Similarity： $\cos\theta$

$$a^Tb = \vec{a}\cdot\vec{b} = |a||b|\cos\theta$$ 向量点乘的定义，以此可算出 $\cos\theta$

$$\cos(\theta) = \frac{a^T\cdot b}{|a||b|}$$ *用向量夹角来度量相似度.

$X=\{x_1, x_2 \cdots x_n\}$
$Y=\{y_1, y_2, \cdots y_n\}$
随机变量X, Y各有n次的数

<d> Pearson 相关系数.

$$\rho_{XY} = \frac{cov(X,Y)}{\sigma_X\sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y} = \frac{\sum_{i=1}^{n}[(x_i-\mu_x)(y_i-\mu_y)]}{\sqrt{\sum_{i=1}^{n}(x_i-\mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i-\mu_y)^2}}$$

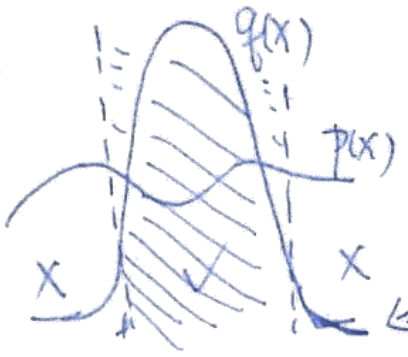
协方差　X方差　Y方差

当 $\mu_x, \mu_y = 0$ 时会退化为 Consine Similarity，也解释了文档相似度为什么使用 Consine Similarity

<e> 相对熵 (K-L 距离)：参见概率

$$D(p\|q)=\sum_x p(x)\log\frac{p(x)}{q(x)} = E_{p(x)}\log\frac{p(x)}{q(x)}$$ ⇐ ★（在 $p(x)\neq 0$ 时）要求 $q(x)$ 够大才有机会被与 p 聚到一个 cluster 中

特点：适用于这样的场景

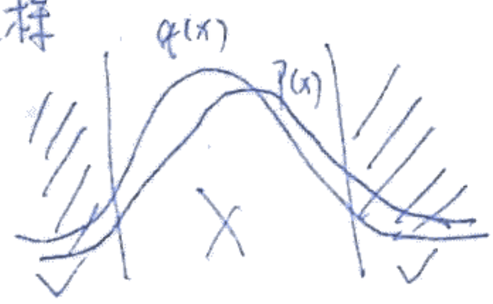


$q(x)$ 接近 0 时，$\frac{p(x)}{q(x)}$ 会非常大，$D(p\|q)$ 也非常

此时要考虑的问题是：在 $p(x)\neq 0$ 的位置，尽量让 $q(x)$ 不接近 0

$$D(q\|p)=\sum_x q(x)\log\frac{q(x)}{p(x)} = E_{q(x)}\log\frac{q(x)}{p(x)}$$ ⇐ 要求 $q(x)$ 在 $p(x)$ 很小时也很小，才有机会与 p 聚到一个 cluster 中

特点：适用于这样的场景.



$p(x)$ 接近 0 时，只有让 $q(x)$ 也很小，$\frac{q(x)}{p(x)}$ 才不会很大，

<f> Hellinger Distance

$$D_\alpha(p\|q)=\frac{2}{1-\alpha^2}\left[1-\int p(x)^{\frac{1+\alpha}{2}}\,q(x)^{\frac{1-\alpha}{2}}\,dx\right]$$

该距离满足三角不等式，是对称，非负距离（证明见 PPT）

α 接近 ±1 时，$D_\alpha(p\|q)$ 接近于 $D(p\|q)$ 或 $D(q\|p)$

α 接近 0 时，$D_\alpha(p\|q)$ 接近于 $2\left[1-\int\sqrt{p(x)q(x)}\,dx\right]$

$$= 2-2\int\sqrt{p(x)q(x)}\,dx$$

$$= \int\sqrt{p(x)}\,dx + \int q(x)\,dx - 2\int\sqrt{p(x)q(x)}\,dx$$

$$= \int\left(p(x)-2\sqrt{p(x)q(x)}+q(x)\right)dx$$

$$= \int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2 dx$$

② K-Means 聚类

<a> 方法步骤：

初始：指定K个cluster中心点（☆初始值选择会影响聚类效果）

迭代：对每个样本，距离哪个中心点最近，就将计它属于哪个中心点

重新计算cluster中心点

终止：不断迭代，直到满足中止条件，如

*迭代次数 *簇中心变化率 *MSE达到阈值

<b> 算法变种：k-Medoids（K中位聚类）

噪声影响太大，如样本 1, 2, 3, 4, 100 的均值是22，距离大多数样本太远，此时用K中位来更新cluster会更合适

<c> 如何选初值：选K个样本，作为K个cluster中心点，但这K个样本尽量不要太近

选择方法：样本1随机

(KMeans++) 样本2按到样本1距离加权随机来尽量选远的

代码见PPT 样本3按到样本1,2距离 : : : : : : : : : : :

: : 4 : : : : 1,2,3 : : : : : : : : : : : :

:

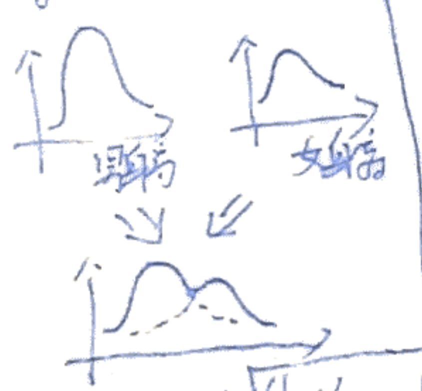
: : k : : : : 1,2~k-1 : : : : : : : : : : : :

<d> k-Means公式化解释

但KMeans不仅假定是GMM，还假定多个高斯分布方差相同

混合高斯分布 (GMM)

e.g.



推导：K个cluster中心记为 $\mu_1, \mu_2, \cdots \mu_k$，样本数为 $N_1, N_2, \cdots N_k$

用MSE作为目标函数

$$J(\mu_1, \mu_2, \cdots \mu_k) = \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{N_j}(x_i - \mu_j)^2$$

关于 $\mu_1, \mu_2, \cdots \mu_k$ 求偏导，令偏导为0求驻点

$$\frac{\partial J}{\partial \mu_j} = -\sum_{i=1}^{N_j}(x_i - \mu_j) \xrightarrow{令} 0 \Rightarrow \boxed{\mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i}$$

正是迭代时重新计算的cluster中心

结论：KMeans本质是一个方差相同的混合高斯分布 (GMM)

建立在每个cluster都服从高斯分布且方差相同的前提假设下

何以求均值，本质都是残差梯度下降的一步操作

<e> Mini-batch k-Means：根据<d>既然KMeans本质是梯度下降，那对其用批量梯度下降也是可以的吗？→ 可以的，代码见PPT

<e> k-Means总结

*优点：简单，快，伸缩性好，效率高，簇近似高斯分布时效果好

*缺点：a. 簇平均值可被定义时才能用　b. K必须事先给出
c. 对cluster中心点敏感　d. 不适合于发现非凸形状的簇或大小差别很大的簇　e. 对噪声和孤立点数据敏感

*可作为其它聚类方法的基础算法如谱聚类

<f> Canopy算法（可能类似kMeans）：在给定先验 near_distance 和 far_distance 条件下，用空间索引做预处理的方法，也能用于聚类

③ 聚类的衡量指标

<a> 概括：*均一、完整性及两者加权　*ARI　*AMI　*轮廓系数

<b> 均一性（Homogeneity）：一个簇只含一个类别的样本

$$h = \begin{cases} 1 & \leftarrow H(C)=0\text{时} \\ 1 - \dfrac{H(C|K)}{H(C)} = 1 - \left(-P(c,k)\ln P(c|k)\right) / \left(-p(c)\ln(p(c))\right) = 1 - \dfrac{P(C,k)\ln p(c|k)}{p(c)\ln(p(c))} \end{cases}$$

条件熵

信息熵

其实就是 $\boxed{\dfrac{I(C,K)}{H(C)}}$

完整性（Completeness）：同类别样本属于同一个簇

$$c = \begin{cases} 1 & \leftarrow H(K)=0\text{时} \\ 1 - \dfrac{H(K|C)}{H(K)} = 1 - \dfrac{P(c,k)\ln p(k|c)}{p(k)\ln p(k)} \end{cases}$$

条件熵

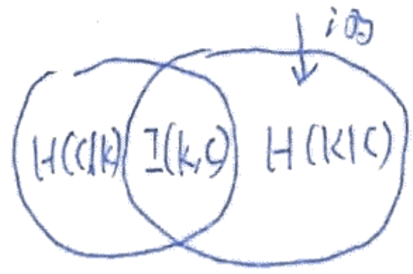
信息熵

其实就是 $\boxed{\dfrac{I(C,k)}{H(K)}}$

V-measure：加权平均

$$v_\beta = \frac{(1+\beta)\cdot h\cdot c}{\beta\cdot h + c}$$

2个事件同时发生新增的



H(C)　H(K)

互比

条件熵越小，即互信息占比越大，聚类效果越好

<c> ARI (RI: Rand(人名) Index; A: Adjusted，因为ARI把值域从类似[0.8,1]变到[0,1]方便使用)

C	Y_1	Y_2	⋯	Y_s	sum
X_1	n_{11}	n_{12}	⋯	n_{1s}	a_1
X_2	n_{21}	n_{22} ←交集样本数	⋯	n_{2s}	a_2
⋮	⋮	⋮		⋮	⋮
X_r	n_{r1}	n_{r2}	⋯	n_{rs}	a_r
sum	b_1	b_2	⋯	b_s	N

数据集：S

两个聚类结果：

$X=\{X_1, X_2, \cdots, X_r\}$

$Y=\{Y_1, Y_2, \cdots, Y_s\}$

X和Y的元素个数为：

$a=\{a_1, a_2, \cdots, a_r\}$

$b=\{b_1, b_2, \cdots, b_s\}$

$n_{ij}=|X_i \cap Y_j|$ 交集元素个数

$$ARI=\frac{Index - EIndex}{MaxIndex - EIndex}$$

$$=\frac{\sum_{ij} C_{n_{ij}}^2 - \boxed{\left[\left(\sum_{i=1}^{r} C_{a_i}^2\right)\left(\sum_{j=1}^{s} C_{b_j}^2\right)\right] / C_n^2}}{\frac{1}{2}\left[\left(\sum_i C_{a_i}^2\right)+\left[\sum_j C_{b_j}^2\right)\right] - \boxed{\left[\left(\sum_i C_{a_i}^2\right)\cdot\left(\sum_j C_{b_j}^2\right)\right] / C_n^2}}$$

--→ RI☆

--→ RI☆

如果X，Y是2个聚类结果，ARI度量的是这两个聚类结果的相似度

如果X是聚类结果，Y是真实的类别标注，ARI度量的是X的相似程度

* 理解上面的公式：

分子：聚类结果中任取2个，在同一个类别中的组合数

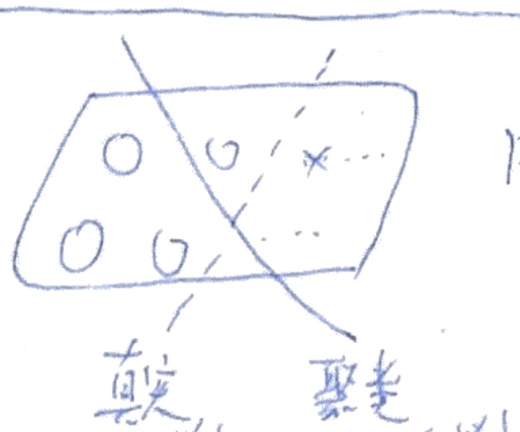
分母：原始样本中，任取2个，在同一个类别中的组合数

--→ RI☆

m_{c_1}个聚到C_1，其中$n_{c_1}-m_{c_1}$个聚错，则

$$RI_{C_1}^{☆}=\frac{C_{m_{c_1}}^2 C_{(n_{c_1}-m_{c_1})}^2}{C_{n_{c_1}}^2}$$

原理



用 C_3^2 / C_4^2 作为度量依据

这个值越高，说明聚类效果越好

<d> AMI：把两次聚类(X, Y)的互信息正则化，求数学期望，再写成类似ARI的形式

X, Y { 是两个聚类结果时，度量的是这两次聚类的相似程度
X是聚类结果，Y是样本聚类标注，度量的是X的聚类效果

本质上是以互信息 $I(X,Y)$ 作为度量依据，值越高代表效果越好.

* 互信息：$MI(X,Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} P(i,j)\ln\frac{P(i,j)}{P(i)P(j)}$

概率

* 正则化之后的互信息：$NMI(X,Y) = \frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}$ 值域正则化到[0,1]

* X服从几何分布(→ 概率)

正则化之后互信息的数学期望

$$E(MI) = \sum_{x} P(X=x)\,MI(X,Y) = \sum_{x=\max(1,\,a_i+b_j-N)}^{\min(a_i,b_j)}\left[MI\cdot\frac{a_i!\,b_j!\,(N-a_i)!\,(N-b_j)!}{N!\,x!\,(a_i-x)!\,(b_j-x)!\,(N-a_i-b_j+x)!}\right]$$

* 写成ARI的形式

$$AMI(X,Y) = \frac{MI(X,Y) - E[MI(X,Y)]}{\max\{H(X),H(Y)\} - E[MI(X,Y)]}$$

<e> 轮廓系数(Sillouette)：以样本的簇内不相似度，簇间不相似度作为度量依据，优点是不需要样本标注就可以度量聚类效果

* { 样本i的簇内不相似度(a_i)：X_i到簇内其它样本的平均距离，越大越不该聚到这个簇.
样本i的簇间不相似度(b_i)：$b_i = \min\{b_{i1}, b_{i2}, \dots, b_{ik}, \dots b_{iK}\}$

样本X_i到簇C_k中样本的平均距离

越大说明X_i越不该属于其它簇

* 公式

不该分到其它簇？ 不该分到当前簇？

$$S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} = \begin{cases} 1-\frac{a(i)}{b(i)} & a(i)<b(i) \quad \text{越接近1，说明聚类越合理} \\ 0 & a(i)=b(i) \quad \text{接近0，说明样本在簇边界上} \\ \frac{b(i)}{a(i)}-1 & a(i)>b(i) \quad \text{接近-1，说明应该分到其它簇} \end{cases}$$

样本i的轮廓系数

正则化

③轮廓系数不准的时候，例如：

同理，也会出现与样本指派相违背的时候

← 这部分样本：

根据轮廓系数该分到 ~~B~~ A

∴ 指派属于B

④ AGNES (AGglomerative NESting)：自底向上的层次聚类

<a> 初始状态：每个对象作为一个簇

<b> 迭代：距离最近的两个簇被合并

簇间距离 {
- X：两个簇最近样本距离 → 容易形成链状结构
- X：两个簇最远样本距离 → 容易受噪声影响
- V：平均距离，两个簇样本间两两距离的均值 (avg) 或 平方和 (ward)

<c> 终止：所有对象最终满足簇数目

好处：* 无需事先知道该聚多少个簇 * 可给出不同聚类粒度的多个模型

⑤ DIANA (DIvisive ANAlysis)：自顶向下的层次聚类

与AGNES过程相反，初始将所有样本放到一个cluster中不断分裂

因为解空间太大，一般使用AGNES

⑥ 基于距离的聚类方法：

KMeans, KMeans++, K-Medoids, Canopy, AGNES, DIANA

局限：只能发现"类圆形"(凸)聚类

⑦ 基于密度的聚类方法：

DBSCAN，密度最大值算法

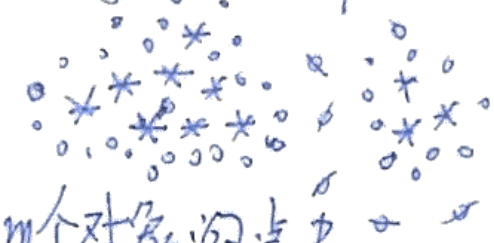
可以发现任意形状的聚类，且对噪声不敏感

点的密度 => 样本周围有多少样本.

⑧ DBSCAN算法：把簇重新定义为密度相连点的最大集合。　　不包含在任何簇中

该算法把具有足够高密度的区域划分为簇，并在有"噪声"的数据中发现任意形状的聚类 (Density-Based Spatial Clusting of Application with Noise)

<a> 计算过程：构建一个簇



*启动：为ε邻域①内包含多于m个对象的点p创建一个新簇，新簇以p为核心对象

*迭代：A. 在簇内寻找核心对象②

B. 该对象直接密度可达③的对象也加入到簇中

*终止：没有新对象可以更新该簇

类似地，可以用剩余样本继续构建新的簇直到整个样本集合聚类完成

① ε邻域：对象半径ε内的区域

② 核心对象：ε邻域内至少有m个其它对象的点

③ 从对象a直接密度可达

满足 { a是核心对象；在a的ε邻域内

那么邻域内这些对象就是从a直接密度可达

<b> 调参：半径ε：增大→宽松；减小→收紧（有可能把一个簇拆成两个）

m：增大→严格

减小→宽松：噪声点→簇内点，小簇合入大簇；一些噪点

⑨ 密度最大值聚类：简洁、优美，可以识别各种形状的类簇且参数很容易确定

<a> 步骤原理

step1：找出各个簇的中心

怎样的点是簇中心：① 局部密度(ρ_i)大，周围点多

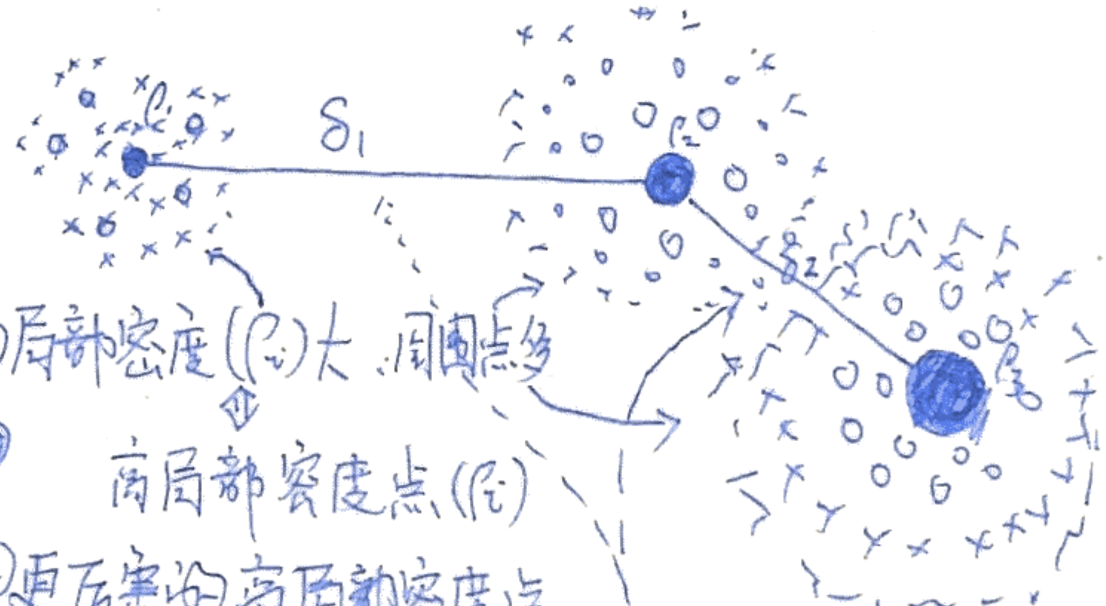
⇕

高局部密度点(ρ_i)

② 更高密度的高局部密度点距离远，即

高局部密度点距离(δ_i)大

step2：其它点按已知簇中心以最近进行分类



<b> 概念

* 局部密度(多种定义方法)

点数量截断值: $\rho_i=\sum_j \chi(d_{ij}-d_c)$ 其中 $\chi(x)=\begin{cases}1 & x<0\\0 & \text{otherwise.}\end{cases}$

邻近

d_c 距离阈值: 因为算法只看 ρ_i 相对值, 所以 d_c 选择
通常选 d_c 值使得平均每个节点邻居数为所有点的 1%~2

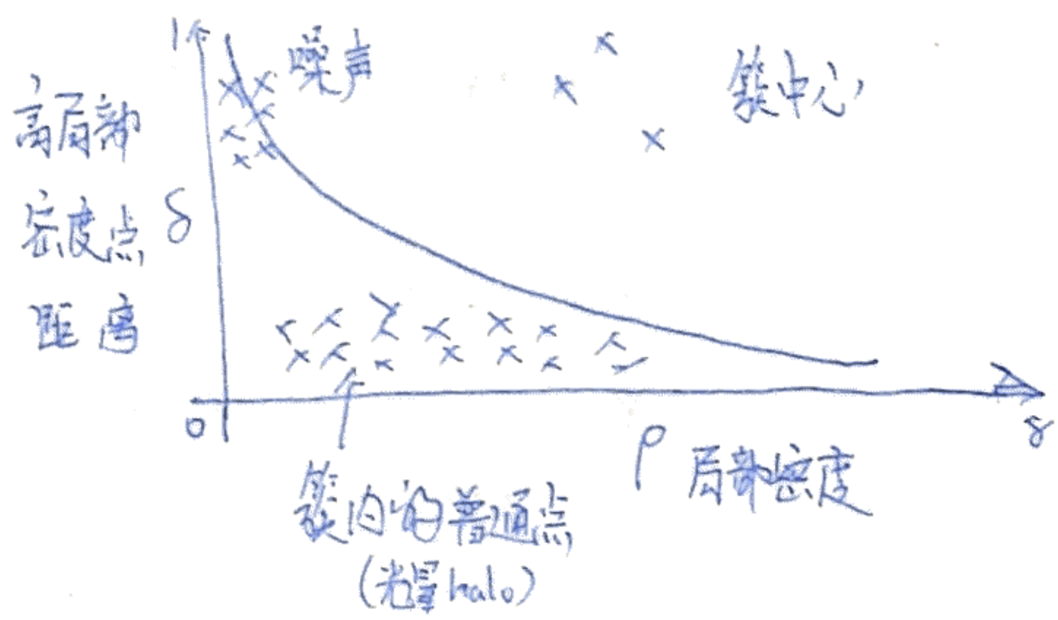
高斯核相似度: 给距离近的点加权

$$\rho_i=\sum_{j\in I_s\setminus\{i\}} e^{-\left(\frac{d_{ij}}{d_c}\right)^2}$$

K近邻均值: 直接用距离, 当然下面公式还要修正下, 使其从距离仅转为局部密度(距离大→局部密度小; 距离小→局部密度大)

$$d_i=\frac{1}{K}\sum_{j=1}^{K} d_{ij}$$

<c> Density Peak 与对象类型



上面这两个聚类算法(DBSCAN, 密度最大值聚类)都只能找到cluster边界, 不能计算后验概率 (后验概率要用EM)

⑩ Affinity Propagation (略): 用于中小规模数据量, 算法不错, 主要问题是计算慢且调参"微妙" (公式,代码见PPT)

⑪ 谱聚类: 基于图论的聚类方法, 通过对样本数据拉普拉斯矩阵⑥的特征向量进行聚类, 来做到对样本聚类 本节介绍

<基于密度, 不是基于距离>

谱: 方阵特征值的全体

谱半径(方阵): 最大的特征值

谱半径(普通矩阵A): (A^TA)最大的特征值

用到的线性代数的定理: 线代⑮, 证明见PPT.

实对称阵的特征值是实数

实对称阵不同特征值的特征向量正交

特点: 计算过程清晰, 理论依据不容易理解

(a) 谱聚类过程

样本 $(x_1, x_2, \cdots x_n)$ → 样本两两相似度 $S_{ij} = \langle x_i, x_j \rangle$ 相似度图 $G(V,E)$ → { 相似度矩阵 W；度矩阵 D ↓ 拉普拉斯矩阵 L } → 对 L 前 k 个特征向量描述的样本 $(u_1, u_2, \cdots u_n)$ 聚类（可能需要单位化） ↓ 映射到原始样本聚类

算法得到的cluster，样本在cluster内有较高权值，cluster间有较低权值。

⊛ 相似度图：有3种选择

全连接图（任意2点都有相似度，越近相似度越高）：$S(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$ 高斯相似度

但对角线置零，方便计算度矩阵。

ε近邻图（互相在ε邻域内的点才会相连）

↳ 阈值取值：G权值均值；G最小生成树的最大边。

推荐先使用 ↓

K近邻图（有向图，B属于A的K近邻，则有一条边从A到B）

※ 优缺点：全连接图权值完整但不够稀疏；ε近邻图有的点连接太密有的连接不到

K近邻图解决ε近邻图的问题，互K近邻图是无向图 性质介于K近邻图和ε近邻图之间

⊛ 从相似度图得到相似度矩阵 W

⊛ 从相似度矩阵 W 得到度矩阵 D：（W 的行加和，放到对角线上）

$$W = \begin{bmatrix} 0 & w_{12} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \xrightarrow[d_{ii} = \sum_{j=1}^{n} w_{ij}]{} D = \begin{bmatrix} d_{11} & & & 0 \\ & d_{22} & & \\ & & \ddots & \\ 0 & & & d_{nn} \end{bmatrix}$$

无自连，自己不和自己聚类

⊛ 从 W, D 得到拉普拉斯矩阵 L，有3种选择（差别不大，除了最后一种要单位化度1）

- 未正则拉普拉斯矩阵：$L = D - W$
- 对称拉普拉斯矩阵：$L_{sym} = D^{-\frac{1}{2}} \cdot L \cdot D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} \cdot W \cdot D^{-\frac{1}{2}}$
- 随机游走拉普拉斯矩阵：$L_{rw} = D^{-1} \cdot L = I - D^{-1} W$（首选）

其实是转移矩阵：从i走到j的概率正比于 w_{ij}

$P_{ij} = w_{ij} / d_i$

$P = D^{-1} W$

特点：① 是对称半正定阵 ② 最小特征值是0 ③ 相应特征向量是全1向量。

⊗ 对L(或L_{sym}, L_{rw})求特征值和特征向量(列向量)，按特征值由大到小排列

L(或L_{sym}, L_{rw})

$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$

$$L_{n\times n} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix} \longrightarrow U_{n\times n} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{bmatrix} \longrightarrow U'_{n\times k}$$

只保留前k列

$k^* = \arg\max_k |\lambda_{k+1} - \lambda_k|$

一种确定k的方法

删掉

⊗ $U'_{n\times k}$ 的n个行向量对应于n个样本

对这n个行向量用k-means聚类

(其中如果使用的是随机游走拉普拉斯矩阵，还需多做一步归一化，使行向量$|y_i|$为1)

⑪ 谱聚类可用切割图/随机游走/扰动论来解释：即能够找到图的一个划分，使得随机游走在相同的簇中停留而几乎不会游走到其它簇.

代码见PPT

⑫ 标签传递算法：用于部分样本有标记的情况(半监督学习)

Label Propagation Algorithm，将标记样本的标记通过一定概率传递给未标记样本，直到最终收敛.

代码见PPT

⑬ 应用 (Vector Quantization)

kmeans用于提取图像特征：矢量量化 (CNN出现之前常用的方法)

例：

SIFT + Vector Quantization + Pyramid Pooling + SVM

SIFT + Local Sparse Coding Macrofeatures + Pyramid Pooling + SVM

SIFT + Fish Vectors + Deformable Parts Pooling + SVM