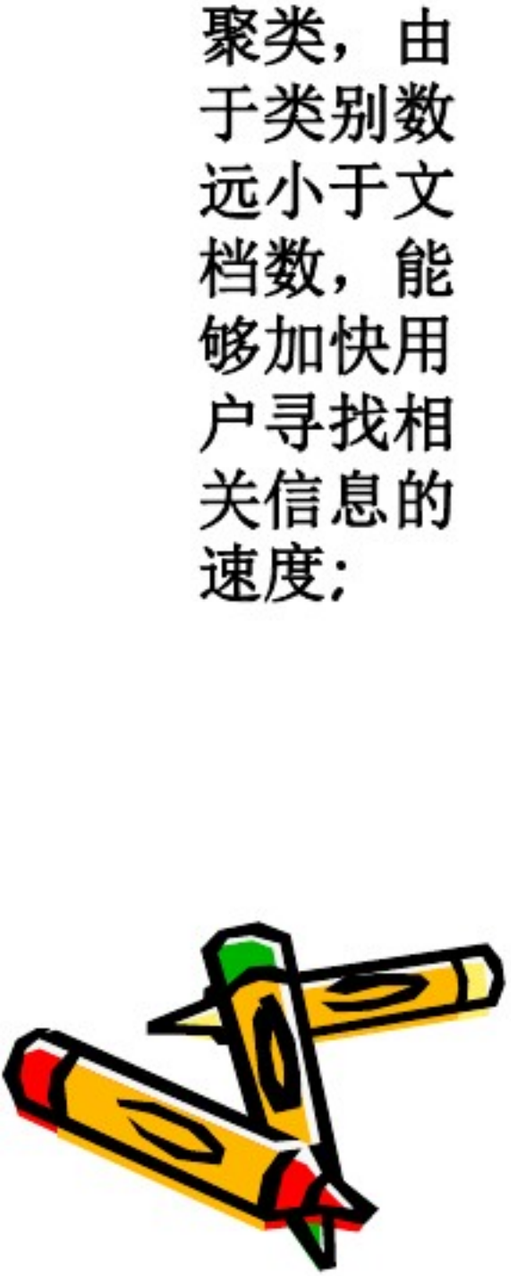
什么是聚类?

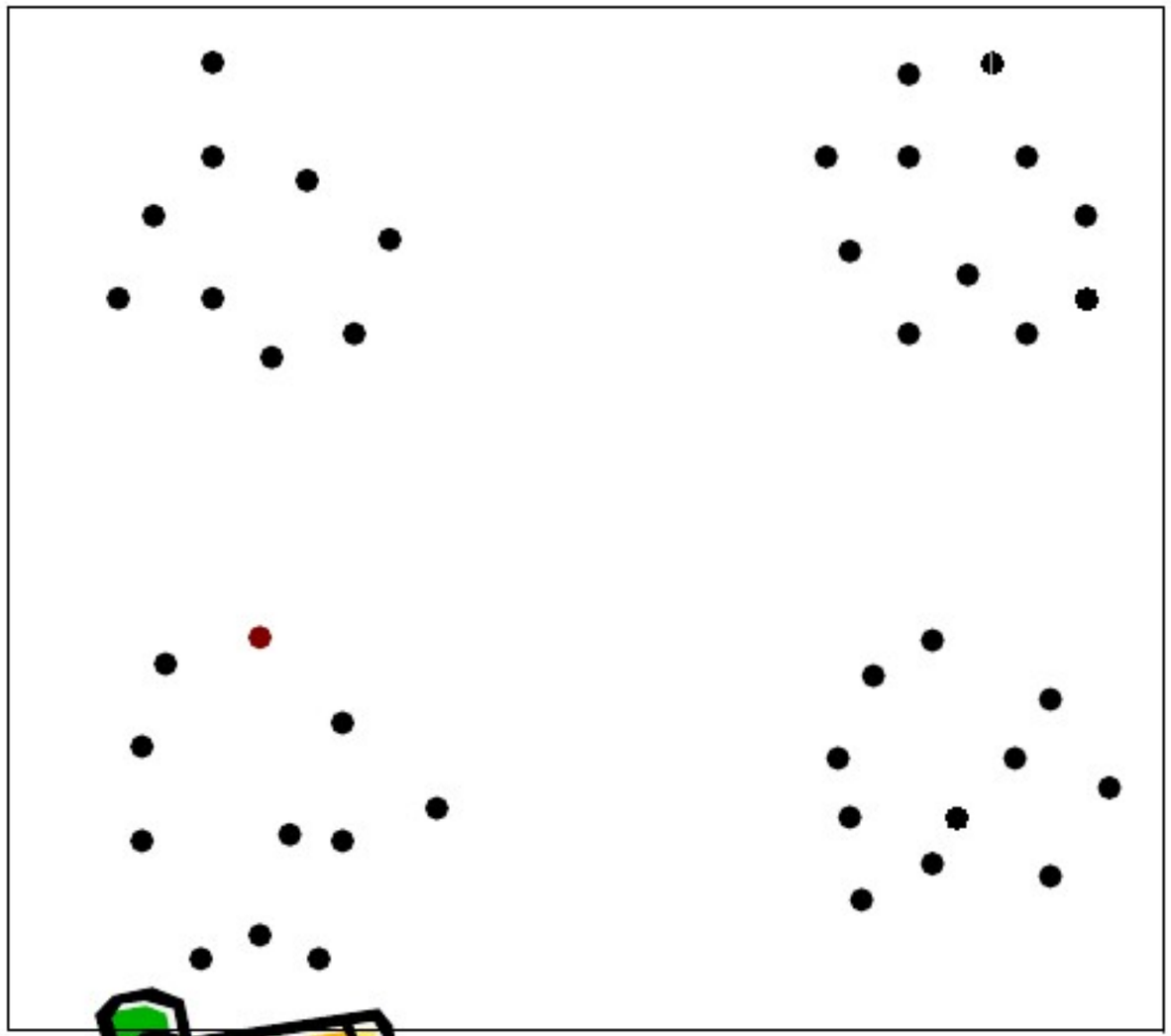
- 聚类就是对大量未知标注的数据集，按数据的内在相似性将数据集划分为多个类别，使类别内的数据相似度较大而类别间的数据相似度较小;

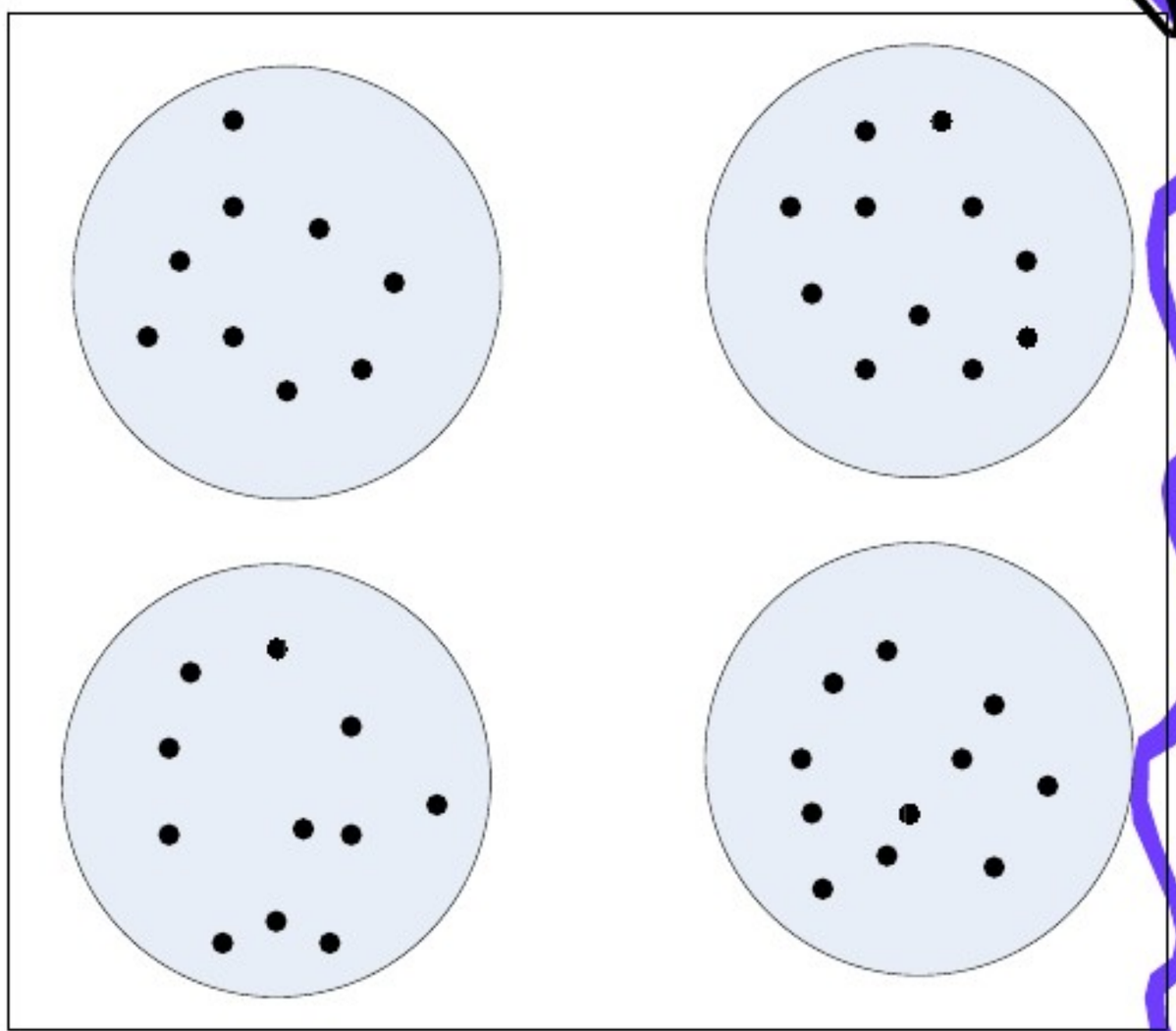
为什么需要聚类?

对相似的文档或超链接进行聚类，由于类别数远小于文档数，能够加快用户寻找相关信息的速度;



聚类图示





聚类中没有任何指导信息，完全按照数据的分布进行类别划分

什么是分类?

- 数据集合 $Data$，类别标记集合 C

$$\forall x \in Data,\ \ Class(x) \in C$$

- 数据集合： 训练数据 $TrainData$

待分类数据 $ClassData$

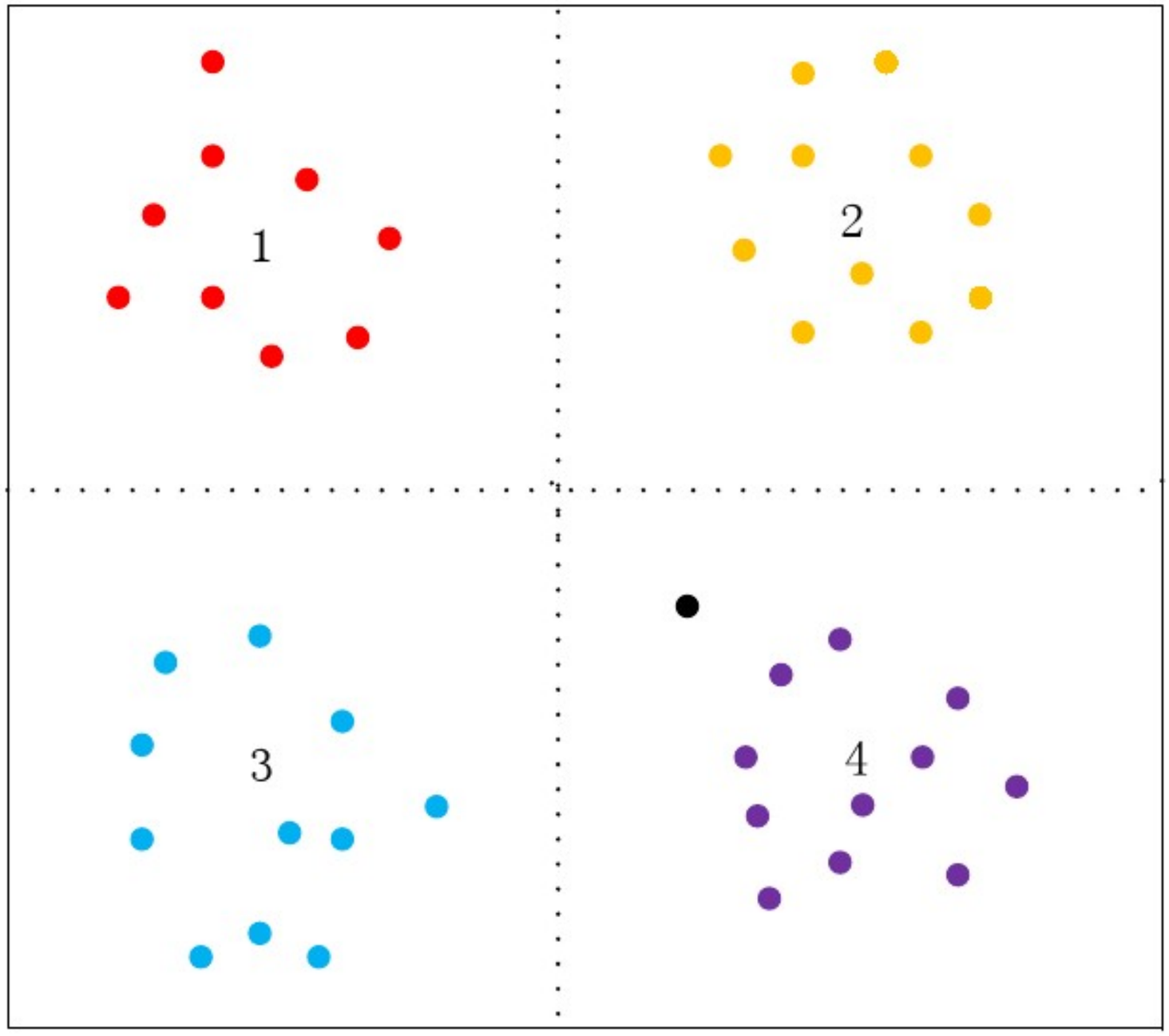
- 已知 $\forall x \in TrainData;\ konwClass(x) \&\& Class(x) \in C$
- 问题： $\forall t \in ClassData;\ \ Class(t)?$
- 方法： 根据训练数据获得类别划分标准 $f(x)$

$$\forall t \in ClassData;\ \ Class(t) = f(t)$$

分类图示

训练数据

待分类数据



聚类与分类的区别

- 有类别标记和无类别标记;
- 有监督与无监督;
  （有训练语料与无训练语料）
  - Train And Classification （分类）;
  - No Train（聚类）;

聚类的基本要素

- 定义数据之间的相似度;
- 聚类有效性函数（停止判别条件）；

1. 在聚类算法的不同阶段会得到不同的类别划分结果，可以通过聚类有效性函数来判断多个划分结果中哪个是有效的;

2. 使用有效性函数作为算法停止的判别条件，当类别划分结果达到聚类有效性函数时即可停止算法运行;

- 类别划分策略（算法）；

通过何种类别划分方式使类别划分结果达到有效性函数;

相似度

- Euclidean Distance

$$Euclidean(\boldsymbol{A}_i, \boldsymbol{A}_j) = \sum_{m=1}^{r} (A_{im} - A_{jm})$$

数据表示为向量，向量中某一维对应数据某一特征或属性

仅计算了数据向量中属于同一维度特征的权值差距；

聚类有效性函数

- 最小误差（J_e）：

c个类别，待聚类数据x，m_i为类别C_i的中心，

$$m_i = \frac{\sum_{x \in C_i} x}{|C_i|} \qquad J_e = \sum_{i=1}^{c} \sum_{x \in C_i} \| x - m_i \|^2$$

J_e越小聚类结果越好

J_e 衡量属于不同类别的数据与类别中心的的误差和；

- 最小方差:

$$\overline{S_i} = \frac{1}{n^2} \sum_{x \in C_i} \sum_{x' \in C_i} \| x - x' \|^2$$

$\overline{S_i}$ 衡量同一类别内数据的平均误差和；

聚类算法的简单分类

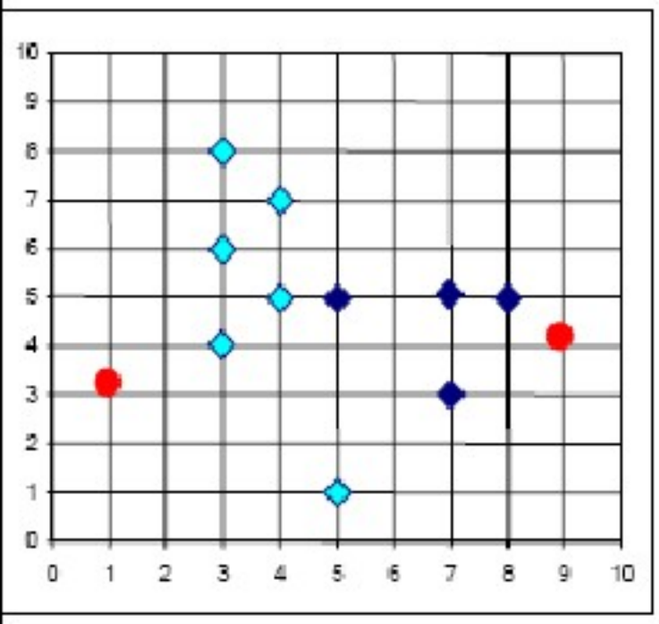
- 基于划分: K-means, K-medoids
- 基于层次: HFC
- 基于密度: DBSCAN
- 基于网格: CLIQUE，STING

K-means

- 初始参数-类别数&初始类别中心;
- 聚类有效性函数-最小误差;
- 优点：
  聚类时间快;
- 缺点：
  对初始参数敏感;
  容易陷入局部最优;

K-means步骤

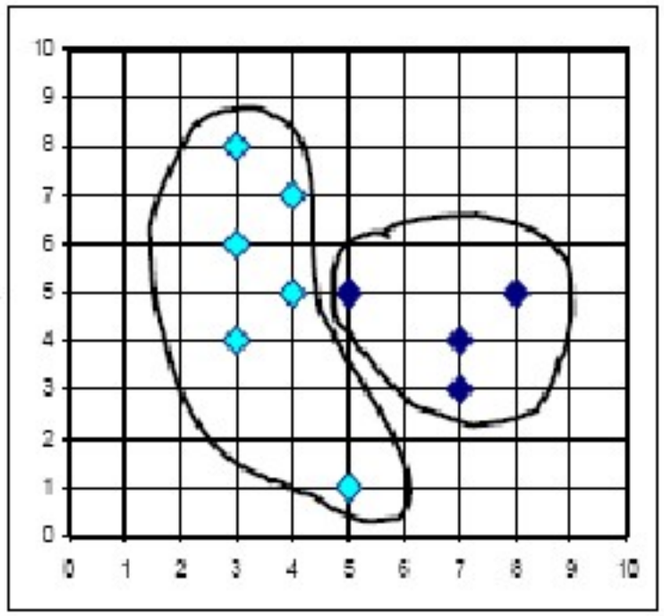
- 1 设置初始类别中心和类别数;
- 2 根据类别中心对数据进行类别划分;
- 3 重新计算当前类别划分下每类的中心;
- 4 在得到类别中心下继续进行类别划分;
- 5 如果连续两次的类别划分结果不变则停止算法;否则循环2～5；



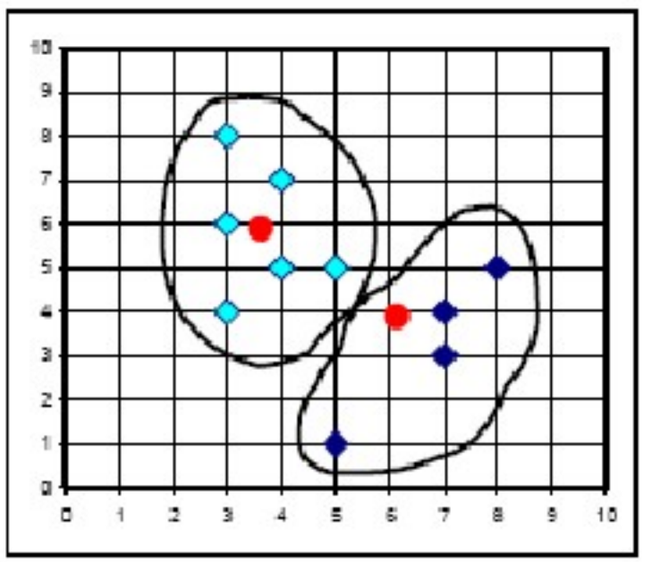
K=2

Arbitrarily choose K object as initial cluster center

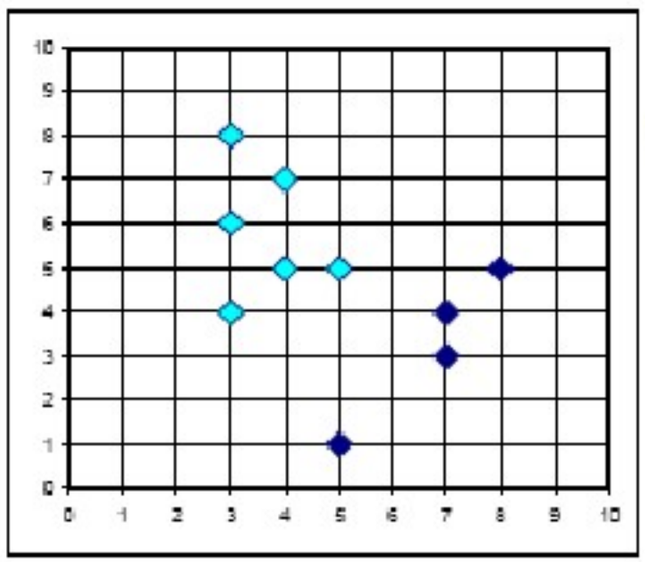
Assign each objects to most similar center



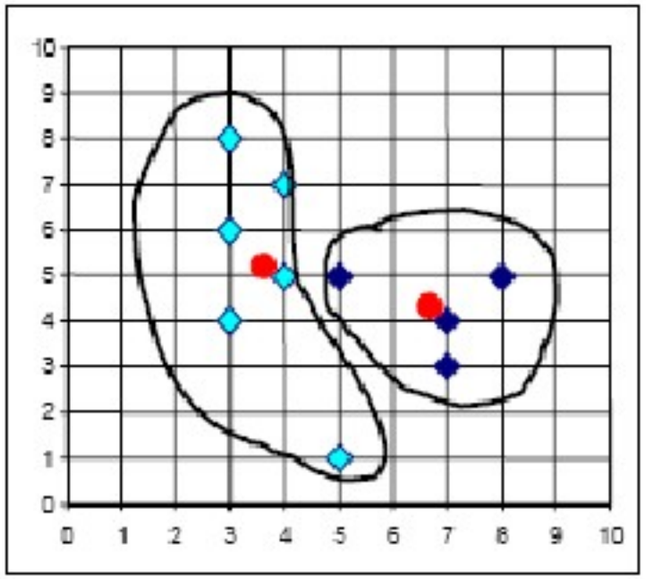
reassign



Update the cluster means

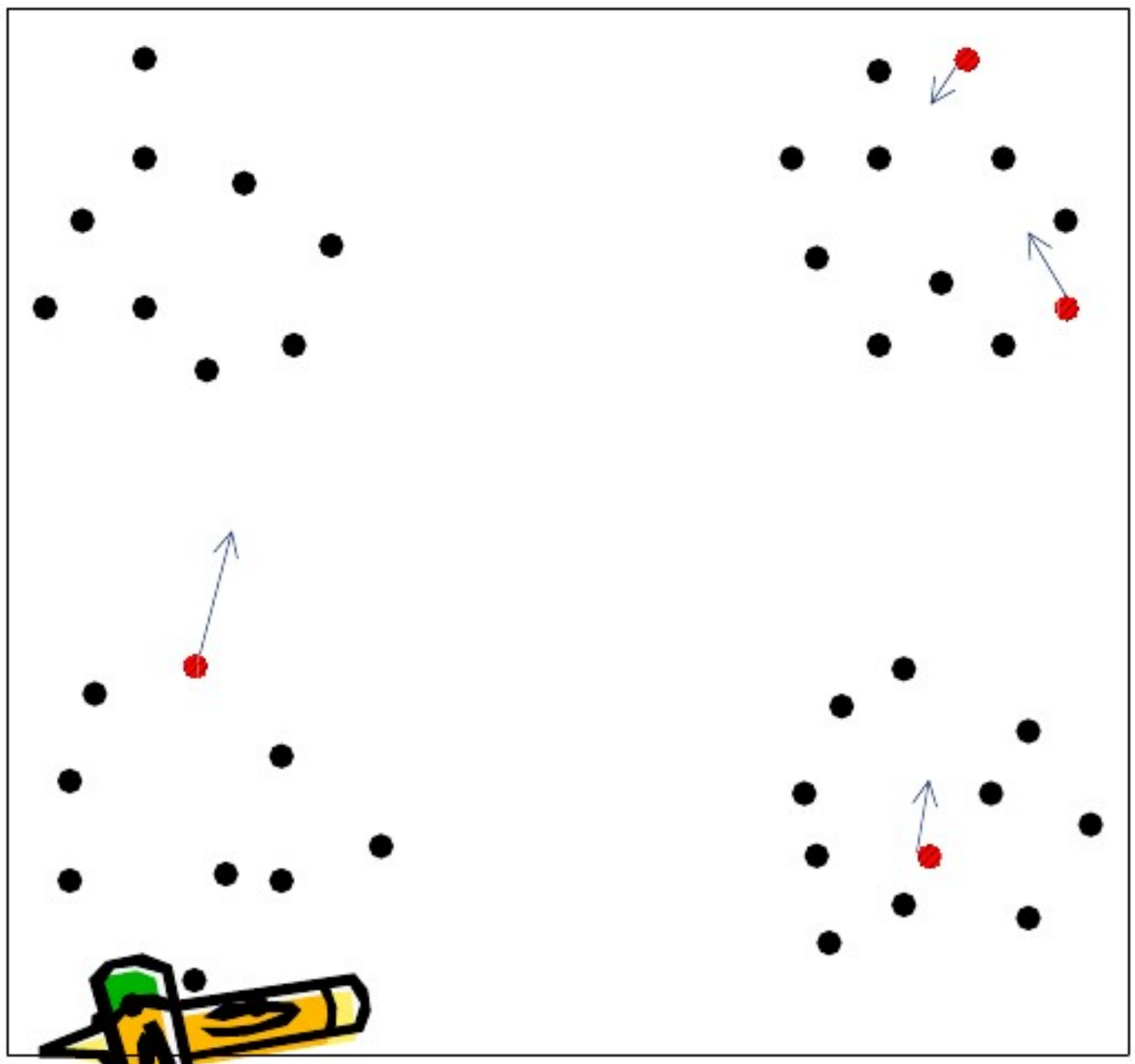


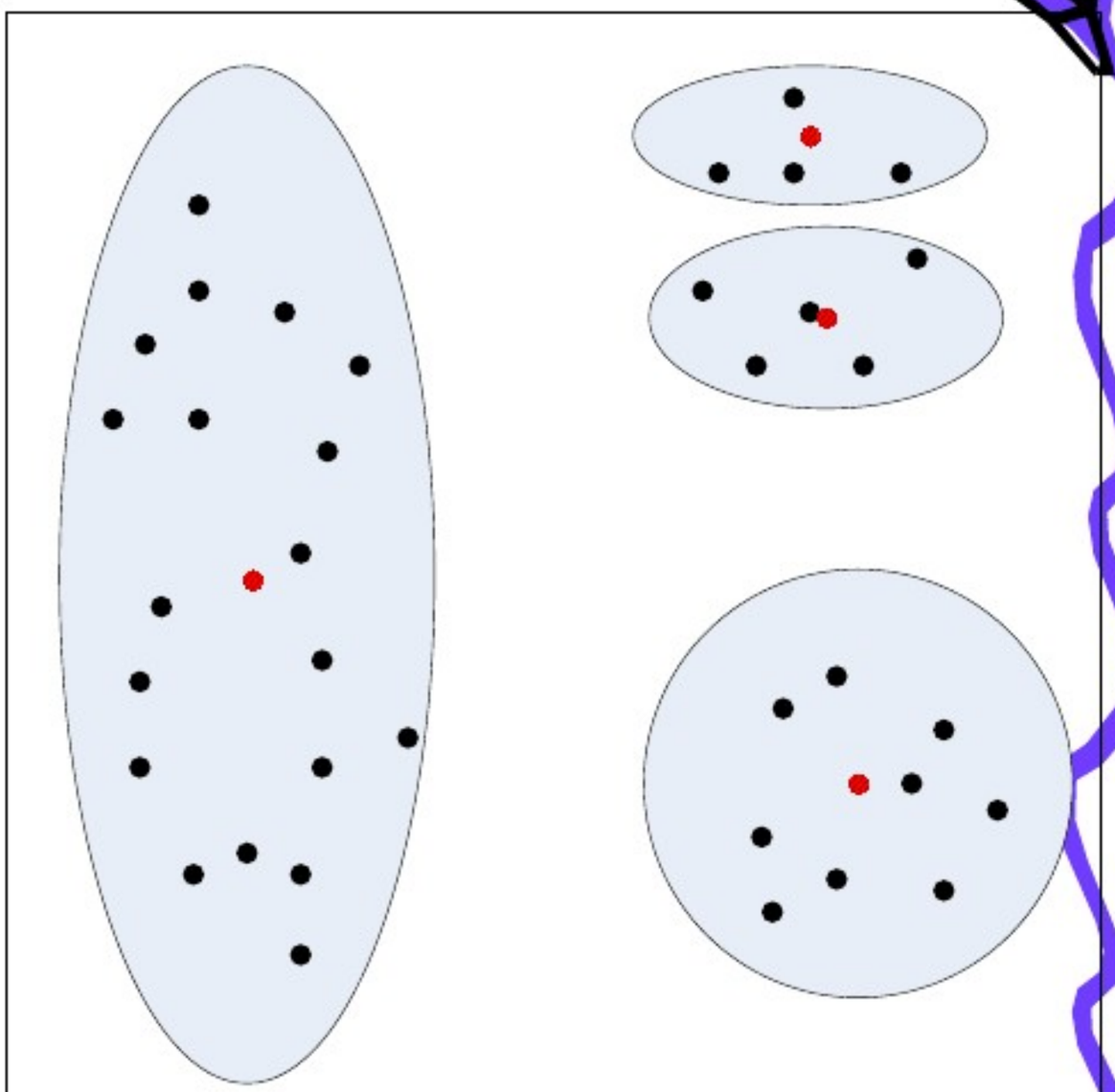
reassign



Update the cluster means

初始值敏感



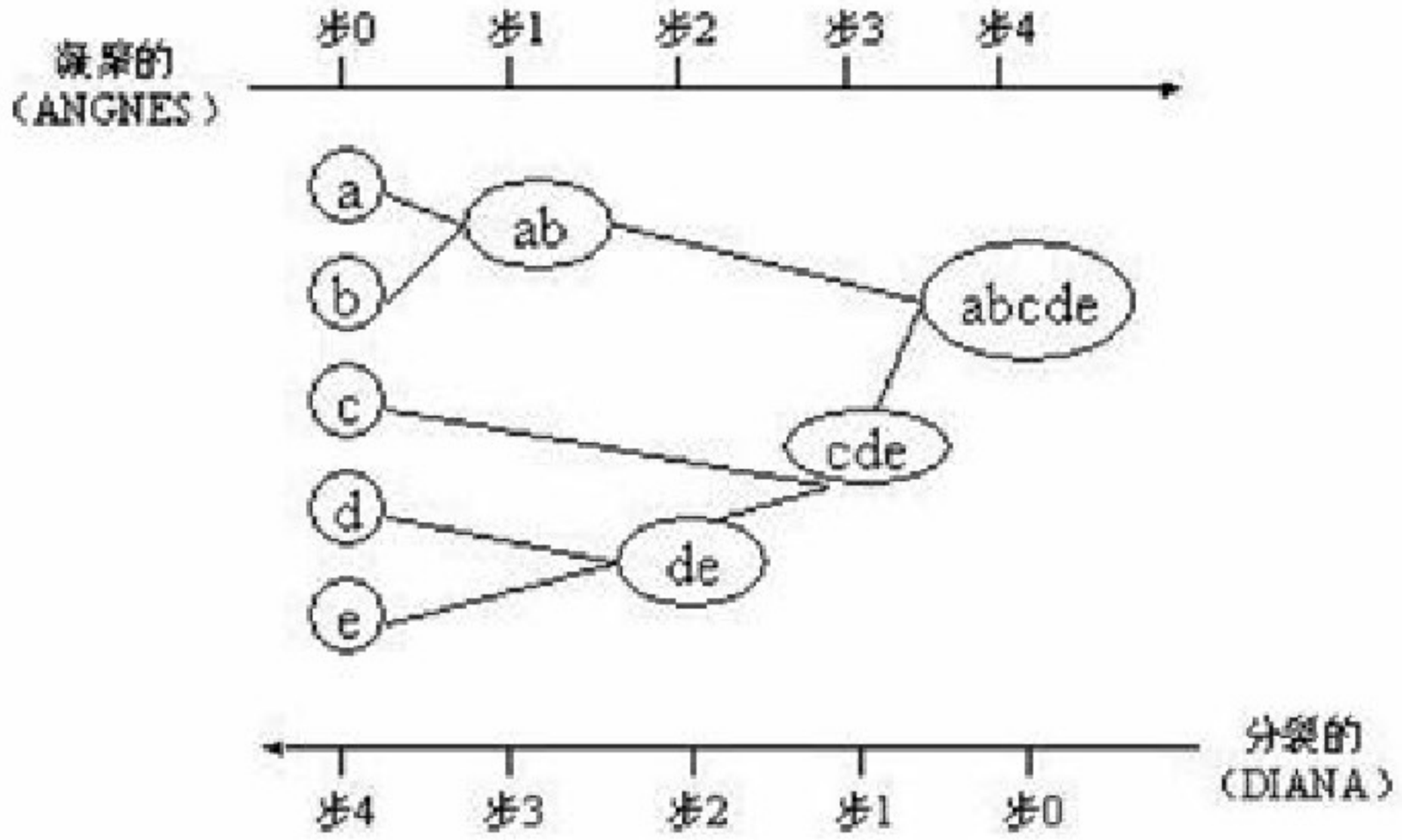


初始化4个类别中心；

左侧的全体数据仅与第一个类别中心相似；

层次聚类

- 分裂或凝聚



算法运行到某一阶段，类别划分结果达到聚类标准时即可停止分裂或凝聚；

基于聚类的入侵检测方法

- 由于IDS需要处理的数据量非常大，对建模和检测的准确性、时效性要求高，因此在研究基于聚类的入侵检测方法时重点考虑三个方面的要求：
  - 聚类算法时间复杂度低；
  - 聚类精度高，能将不同类型的数据聚集在分离的簇中；
  - 给簇准确做标记，能得到较准确的分类模型。

基于聚类的检测方法

- 主要由两大模块构成：
  - 模型建立
    - 第一步：对训练集进行聚类；
    - 第二步：利用聚类结果得到分类模型；
  - 模型评估
    - 检测率：被正确检测的攻击记录数占整个攻击记录数的比例。
    - 误报率：表示正常记录被检测为攻击的记录数占整个正常记录数的比例。
    - 未见攻击类型的检测率：表示测试集中出现而训练集中没有出现的新类型攻击记录被正确检测的比例。

基于聚类的入侵检测方法
分类

- 有指导的入侵检测方法
  - 通过在已标记为正常和入侵的数据集上进行训练，建立分类模型，通过检测数据偏离各分类模型的偏差来检测非正常的、潜在的入侵行为。
  - 方法的有效性取决于训练数据集的质量。
  - 要求训练数据被正确地标记为正常或攻击，如果标记不正确，则算法可能会将某种入侵行为及其变种看成正常而不能检测，从而使检测率降低，或者将正常行为看成入侵，使误报率提高。

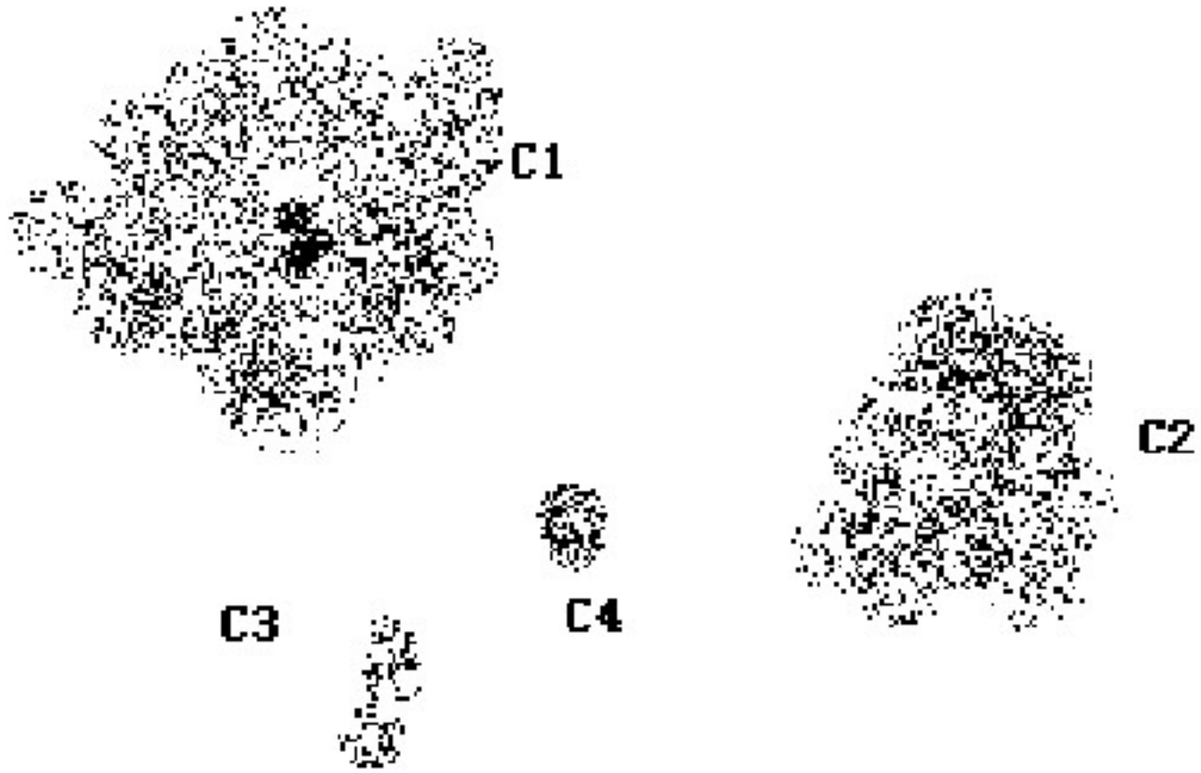
有指导的聚类检测过程

- 1.初始时，簇集合为空，读入一个新的对象；
- 2.以这个对象构建一个新的簇，该记录的类别标记作为新簇类别的标志；
- 3.若已到数据库末尾，则转6，否则读入新对象，利用给定的距离定义，计算它与每个簇间距离，并选择最小的距离；
- 4.若最小距离超过阈值 r，或对象的类别与其最近簇的类别不同，转2；
- 5.否则将该对象并入具有最小距离的簇中并更新该簇的各类属性值的统计频度及数值属性的簇中心，转3；
- 6.结束。

基于聚类的入侵检测方法
分类

- 无指导的入侵检测方法
  - 是在未标记的数据上训练模型并检测入侵，不需要任何先验知识，可能检测新的、未知的入侵。
- 基于基本的假定：
  - 正常行为较入侵行为占绝对的比例；
  - 入侵行为偏离正常行为是可以区别的。

聚类簇



无指导的聚类检测过程

- 1.模型建立
  - 第一步：对训练集T1进行聚类，得到聚类结果T1={C1,C2,…,Ck};
  - 第二步：给簇做标记：统计每个簇Ci（1≤i ≤k）的异常因子或数据量的大小。
- 2.确定模型：确定每个簇的类中心和半径阈值r。
- 3.利用最近邻分类方法对测试集中的每个对象进行分类;

实验数据集KDD CUP

- KDD Cup1999入侵数据包是真正的网络数据，它是在军事网络坏境中运用非常广泛的模拟入侵攻击所得到的数据集。包含大约490万条数据纪录。通过检测记录中是否包含有攻击行为以及攻击行为的类别，把记录标记成为正常记录或是某种攻击的记录。并且认为这些标记都是正确可信的。

实验数据集KDD CUP

- KDD Cup1999中总共包括了41个特征，其中9个是离散的特征值，而32个是连续的特征值。这些特征是从连接中抽取出来专门为了区分正常连接和异常连接的特征。

单个TCP连接的基本属性

特征名称	特征描述	数据类型
duration	连接时间的长短	连续型
protocol_type	协议类型，比如tcp，udp等	离散型
service	目的端的网络服务，比如http，telnet等	离散型
src_bytes	从源端到目的端传输的字节数	连续型
dst_bytes	从目的端到源端传输的字节数	连续型
flag	连接的状态为normal还是error	离散型
land	源和目的主机/端口是否相同，相同为1，不同为0	离散型
wrong_fragment	错误分片的数目	连续型
urgent	紧急包的数目	连续型

实验数据集KDD CUP

- 实验数据集采用KDD Cup1999网络数据集。该数据集中包含的攻击类型可以分成是四大类:
  - DOS——拒绝服务攻击类型(比如，Syn flood);
  - U2R——非授权得到超级用户权限或运行超级用户函数(比如，缓冲溢出攻击);
  - R2L——从远程计算机进行非授权的访问(比如，密码的猜测及用户权限级别的提升);
  - Probing——扫描或者对其它系统漏洞的探测(比如，端口扫描)。

实验数据集KDD CUP

攻击所属类别	攻击名称
DOS	Back, land, Neptune, pod, smurf, teardrop
U2R	Buffer_overflow, loadmodule, perl, rootkit
R2L	ftp_write, guess_passwd,imap,multihop, phf,spy, warezclient,warezmaster
Probing	Ipsweep,nmap,portsweep, satan

Thank You!