第十一章 自组织映射网络SOM和DBSCAN聚类

一、自组织特征映射SOM

有两层结构：输入和输出
输入层：输入的属性数量是对应的节点数
输出层：簇的数量K是输出的节点数
w_{jk}是输入层节点i到输出层节点k的权重

二、聚类步骤

第一阶段：准备工作 (Initialization)

1. **数据预处理**
   - **操作**：将所有输入数据的属性值归一化到 $[0, 1]$ 区间。
   - **原因**：算法核心依赖**欧几里得距离**。如果特征尺度不一致（比如一个是身高 cm，一个是工资 k），大数值特征会主导距离计算，导致模型失效。
2. **网络初始化**
   - 设定聚类数量 K（即输出层神经元的数量）。
   - 每个神经元 j 都被初始化为一个权重向量 $W_j = [w_{1j}, w_{2j}, \ldots, w_{pj}]$。
   - p 代表输入数据的维度（特征数）。可以将 W_j 理解为该神经元在 p 维空间中的“坐标位置”。

第二阶段：迭代训练 (Iterative Training)

假设总迭代次数为 c，当前时刻为 t（从 0 开始）。

步骤 1：竞争 (Competition)

- 随机从训练集中抽取一个输入样本向量 $X(t)$。
- 计算 $X(t)$ 与所有神经元权重向量 $W_j(t)$ 之间的**欧几里得距离**。
- **选出获胜者 (Winner)**：距离最小的那个神经元被标记为优胜节点（Best Matching Unit, BMU），记为 $W_i(t)$。

步骤 2：协同 (Cooperation & Adaptation)

- **确定邻域**：以获胜节点 $W_i(t)$ 为中心，划定一个半径 $R(t)$。落在该半径内的所有神经元被视为“邻居”。
- **权重更新**：对获胜节点 **及其所有邻居节点** 进行更新。它们都会向输入样本 $X(t)$ 靠拢。
  - **通用更新公式**：

$$W_j(t+1) = W_j(t) + \eta(t) \cdot [X(t) - W_j(t)]$$

- *注：这里 j 代表获胜者自己以及它的邻居们。*
- **物理含义**：新位置 = 旧位置 + (学习率 $\times$ 差距向量)。即向目标迈进一步。

步骤 3：参数衰减 (Decay)
在进入下一次循环前，需要降低“学习率”和“邻域半径”：

1. **学习率衰减**（步子越迈越小）：

$$\eta(t) = \eta(0) \times (1 - \frac{t}{c})$$

   - $\eta(0)$：初始学习率。
   - c：总迭代周期数。
   - 随着 $t \to c$，学习率趋近于 0。
2. **邻域半径收缩**（影响范围越缩越小）：
   - 随着时间推移，半径 $R(t)$ 逐渐减小。
   - **初期**：半径大，大片区域跟着动（全局调整，类似“拓扑排序”）。
   - **末期**：半径减为 0，只更新获胜者自己（局部微调）。

步骤 4：循环判断

- 检查是否达到停止条件（如：达到最大循环次数 c，或权重变化量极小）。
- 若未满足，令 $t = t + 1$，返回**步骤 1** 继续下一轮。

核心机制总结

突出了 SOM 的两个核心特点：

1. **竞争学习 (Competitive Learning)**：
   只有“像”输入数据的神经元（获胜者）才有资格调整权重，这使得不同的神经元逐渐学会识别不同的模式。
2. **拓扑保持 (Topology Preservation)**：
   这是 SOM 区别于 K-Means 的最大不同点。
   - **机制**：因为“邻居”也跟着一起更新。
   - **结果**：在输出层（网格）上物理位置相邻的神经元，其代表的权重向量在输入空间中也是相似的。这不仅完成了聚类，还完成了数据的可视化降维。

三、基于密度的空间应用噪声聚类DBSCAN

1. 特点
   生成任意形状的聚类簇
   对噪声具有鲁棒性

无需提前设定K值
与人类的视觉有点相似

2. **DBSCAN需要的两个参数**
   eps——用于定义数据点周围的邻域。即，如果两个点之间的距离小于或等于eps值，则将它们视为邻点。
   min_samples——在eps半径范围内所需的最少邻居（数据点）数量。一般来说，最小值（MinPts）可以通过数据集的维度数D来计算得出，即MinPts>=D+1。MinPts的最小值至少应选择为3。
3. 基本概念
   核心点：如果一个点在邻域中有超过最小邻居点的其他点，则可以认为它是核心点
   边界点：如果一个点在邻域中有少于最小邻居点数量的其他点，并且它在核心点的邻域内，可以认为它是边界点
   噪声或者异常值：不是核心店也不是边界点的叫做噪声或者异常值
4. DBSCAN的可达性和连接性
   直接密度可达：如果对象（或样本）q 位于对象 p 的 **ε 邻域** 内，并且 **p 是核心对象**，那么 q 就称为从 p **直接密度可达**。直接密度可达并不是对称的，我们不能认为p到q是直接密度可达的，因为q不是核心点
   密度可达的传递性
   若在给定参数 ε 和 MinPts 条件下，存在一条对象序列 $q_1, q_2, \ldots, q_n$，且 $q_1 = p$、$q_n = q$，并且对于所有 $1 \le i \le n-1$，都有 q_{i+1} **对** q_i **直接密度可达**，则称对象 **q 从对象 p 出发是密度可达的**。
   密度相连（Density Connectivity）： 在参数 ε 和 MinPts 下，如果存在某个对象 **o**，使得 **p 和 q 都能从 o 密度可达**，那么称对象 **p 与 q 是密度相连的**。**密度相连是对称的**： 如果 **q 与 p 密度相连**，则 **p 也一定与 q 密度相连**。
5. **离群点（Outliers）：**
   所有**无法从任何其他点密度可达（或密度相连）**的点，都被视为离群点。
   换句话说，这类点既**不是核心点**，也不是**边界点（Border）**，并且在距离 n 的范围内包含的邻域点数 **少于 m**。
6. **一个簇（cluster）需要满足以下两个性质：**
   簇内所有点必须至少互相密度相连，即任意两点之间都可以通过密度连接路径关联起来。
   如果某个点能够从簇中任意一点密度可达，那么该点也属于该簇。

四、DBSCAN算法

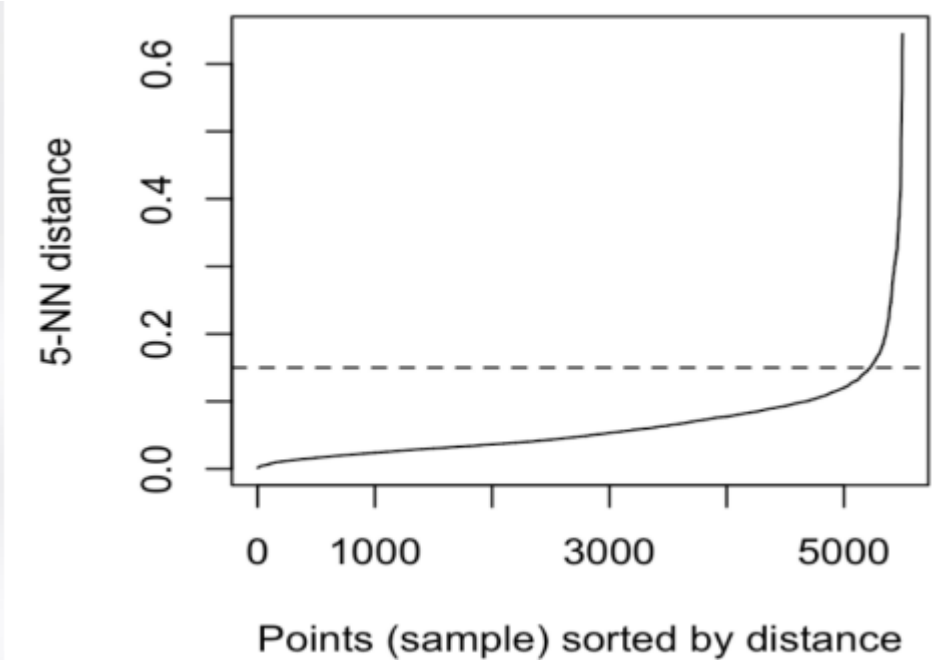
1. **算法步骤**
   - **初始化**
     - 将所有点的状态标记为**未访问**；
     - 输入参数：
       - `ε`（邻域半径，用前面提到的 k-距离方法确定）
       - `MinPts`（最小领域内点数）
   - **遍历数据集**

- 从数据集中随机选择一个未访问点 `p`
  - 将 `p` 标记为**已访问**；
  - 计算 `p` 的邻域，即找出所有到 `p` 距离 ≤ `ε` 的点集合 `N(p)`；
- **判断核心点**
  - 若 `N(p)` 中的点数 ≥ `MinPts`，则 `p` 为**核心点**，以 `p` 为起点创建一个新簇 `C`；
  - 若 `N(p)` 中点数 < `MinPts`，则 `p` 暂标记为**噪声/离群点**（注意：后面有可能被别的簇吸收）
- **扩展簇（核心点扩张阶段）**
  - 对邻域集合 `N(p)` 中每个点 `q` 进行检查：
  1. 若 `q` 未访问 → 标记为已访问，并计算其邻域 `N(q)`；
  2. 若 `q` 也是核心点（即邻域 ≥ MinPts）→ 将 `N(q)` 合并进当前簇（不断扩张）；
  3. 若 `q` 未被分配簇 → 将 `q` 加入当前簇 `C`；
  - 持续执行直到当前簇无法再扩张为止
- **继续处理剩余未访问点**
  - 回到第 2 步，寻找下一个未访问点；
  - 直到所有点均被访问处理完成；
- **输出**
  - 所有被聚类的点形成若干簇；
  - 未归入任何簇的点即为**噪声点/离群点**。

五、DBSCAN参数的确定

1. 半径距离ϵ选择
   k-距离是点$p(i)$到所有点(除了$p(i)$点)之间距离第k近的距离。对待聚类集合中每个点p(i)都计算k-距离，最后得到所有点的k-距离集合$E=\{e(1),e(2),\ldots,e(n)\}$。 k-距离变化趋势确定K值从而确定邻域半径（距离值）
   计算每个点到K个近邻点的距离的平均值，然后将这些平均值按照从小到大顺序排序，以此来决定肘部点，肘部点对应最优的ϵ参数

肘部点代表k距离曲线中出现剧烈变化的点



2. 确定最小点数MinPts的通用规则
   MinPts 的取值最好结合领域知识以及对数据的理解来设定。下面是一些选择 MinPts 的经验准则：
   - 数据集越大，MinPts 的取值应越大
   - 数据噪声越多，应选择更大的 MinPts
   - 通常情况下，MinPts 应 **大于或等于数据的维度（dimension）**
   - 经验公式：**MinPts = 2 × dimension − 1**，也可写作 **MinPts = k + 1**

六、DBSCAN与K-means的比较

对比维度	**K-means**	**DBSCAN**
聚类形状	聚类结果大多呈球形或凸形	可以发现任意形状的簇
参数需求	需要预先指定 **K**（簇数）	无需指定簇数，只需设置 **半径 ε 与 MinPts**
对 K 的敏感程度	对 K 选择非常敏感，设错影响很大	不需要指定 K，因此不存在此问题
适用场景	更适合大规模数据，效率高	高维数据处理效率不佳
噪声与离群点	对噪声与离群点非常敏感，容易被拉偏	能有效识别与处理离群点与噪声
数据密度变化的适应性	数据点密度变化不会明显影响聚类结果	对稀疏或密度变化较大的数据表现不理想