第11章　聚类分析

刘家锋

哈尔滨工业大学

第11章　聚类分析

11.1 无监督学习与聚类

聚类任务

- **无监督学习**
  - 无监督学习中，训练样本的标记信息是未知的；
  - 学习的目标是要揭示训练数据的内在性质和规律；
  - 例如在线性成分分析中，PCA属于无监督学习，LDA属于有监督学习；
- **聚类任务**
  - 聚类是无监督学习中研究最多，应用最广的一类任务；
  - 聚类试图将数据集中的样本划分为若干个不相交的子集，称为簇(cluster)；
  - 每个簇对应一些潜在的概念，而这些概念对聚类算法来说也是未知的，是在聚类过程中自动形成的；

聚类任务

- **形式化描述**
  - 给定样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，均为无监督样本；
  - 将D划分为k个不相交的簇$\{C_l | l = 1, \cdots, k\}$，其中：

$$C_{l'} \bigcap\nolimits_{l' \neq l} C_l = \varnothing, \qquad D = \bigcup\nolimits_{l=1}^{k} C_l$$

  - 聚类结果可以表示为：$y_1, \cdots, y_n$；
  - $y_j \in \{1, \cdots, k\}$，表示样本$\mathbf{x}_j$的簇标记；

聚类与混合密度估计

- **混合密度**
  - 样本$\mathbf{x}$来自于k个聚类$\{\omega_1, \cdots, \omega_k\}$；
  - 每个聚类的先验概率为$P(\omega_j)$，样本的分布为$p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$；
  - 样本$\mathbf{x}$服从混合密度：

$$p(\mathbf{x}) = \sum_{j=1}^{k} P(\omega_j) p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$$

- **聚类与混合密度估计**
  - 给定服从混合密度分布的样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$；
  - 混合密度估计的目标是估计未知参数$P(\omega_j)$和$\boldsymbol{\theta}_j$；
  - 聚类分析的目标是估计样本的聚类标记$\{y_1, \cdots, y_n\}$；

高斯混合聚类

Algorithm 1 高斯混合聚类算法

```
Input: 数据集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，聚类数$k$
Output: 簇划分$\mathcal{C} = \{C_1, \cdots, C_k\}$
用数据集$D$及EM算法学习GMM的参数：$\{(\alpha_j, \boldsymbol{\mu}_j, \Sigma_j)\}_{j=1,\cdots,k}$
$C_j = \varnothing, \quad j = 1, \cdots, k$;
for $i = 1, \cdots, n$ do
    计算样本$\mathbf{x}_i$由各个高斯生成的后验概率：
        $\gamma_{ij} = P(\omega_j|\mathbf{x}_i) = \frac{\alpha_j \cdot p(\mathbf{x}_i|\boldsymbol{\mu}_j, \Sigma_j)}{\sum_{l=1}^{k} \alpha_l \cdot p(\mathbf{x}_i|\boldsymbol{\mu}_l, \Sigma_l)}$
    标记$\mathbf{x}_i$，并划入相应的簇：
        $y_i = \arg\max_{1 \leq j \leq k} \gamma_{ij}, \quad C_{y_i} \leftarrow C_{y_i} \cup \{\mathbf{x}_i\}$
end for
```

混合密度的可辨识性

- **可辨识性**
    - 大多数的混合密度是可以辨识的，根据训练集能够估计出分布参数，但也存在部分混合密度的参数是不可辨识；
    - 不可辨识：无论训练样本的数目有多少，都不存在分布参数的唯一解，则称混合密度是不可辨识的
    - 完全不可辨识：如果分布参数的任何部分都无法估计，则称为完全不可辨识

完全不可辨识的分布

如果样本$x \in \{0,1\}$服从参数为θ的0-1分布，则有：

$$P(x|\theta) = \begin{cases} \theta, & x = 1 \\ 1-\theta, & x = 0 \end{cases}$$

也可以表示为：

$$P(x|\theta) = \theta^x (1-\theta)^{1-x}$$

如果样本x的概率是由2个0-1分布混合而成，参数分别为θ_1, θ_2，并且两个分布的先验概率相等，则有混合分布：

$$P(x|\theta_1,\theta_2) = \frac{1}{2}\theta_1^x (1-\theta_1)^{1-x} + \frac{1}{2}\theta_2^x (1-\theta_2)^{1-x}$$

$$= \begin{cases} \frac{1}{2}(\theta_1+\theta_2), & x = 1 \\ 1-\frac{1}{2}(\theta_1+\theta_2), & x = 0 \end{cases}$$

完全不可辨识的分布

现有样本集$D = \{x_1, \cdots, x_n\}$来自于上述0-1混合分布，其中60%的样本取值1，40%的样本取值0，即：

$$P(x = 1|\theta_1, \theta_2) \approx 0.6, \quad P(x = 0|\theta_1, \theta_2) \approx 0.4$$

如果希望估计出参数θ_1, θ_2，可以得到方程组：

$$\begin{cases} \frac{1}{2}(\theta_1 + \theta_2) = 0.6 \\ 1 - \frac{1}{2}(\theta_1 + \theta_2) = 0.4 \end{cases}$$

显然无法估计出参数θ_1, θ_2的确切值，只能得到：

$$\theta_1 + \theta_2 = 1.2$$

聚类的准则

- **聚类方法**
  - 以混合密度估计的方式聚类，需要对每个簇的分布密度做出假设，如高斯分布等；
  - 实际应用中往往缺乏样本分布的先验信息，很难进行混合密度估计；
  - 更多的方法是按照某种准则来评价聚类结果的优劣，聚类问题转化为了对准则的优化；
- **聚类准则**
  - 聚类的一般准则是聚在同一簇的样本尽可能相似，不同簇的样本尽可能不同；
  - “簇内相似度”越大越好，“簇间相似度”越小越好；

簇内相似度

- **平方误差准则**
  - 平方误差准则可以用来度量簇内的相似度；
  - 令训练样本被聚成了k个簇$C_1, \cdots, C_k$，簇C_j中的样本数为n_j，平方误差可以表示为：

    $$J_e(C_1, \cdots, C_k) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2$$

    其中，$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$为簇$C_j$的均值；
  - 如果聚类只关心簇内样本相似度的话，可以最小化平方误差准则：

    $$\min_{C_1, \cdots, C_k} J_e(C_1, \cdots, C_k)$$

簇内和簇间相似度

- **散布准则**
  - 散布准则兼顾了簇内和簇间的相似度：

    $$J_f(C_1,\cdots,C_k)=\mathrm{tr}(S_t^{-1}S_w)$$

    其中，tr()表示矩阵的迹，S_w和S_t为样本集的簇内和总体散布矩阵：

    $$S_w=\sum_{j=1}^{k}\sum_{\mathbf{x}\in C_j}(\mathbf{x}-\boldsymbol{\mu}_j)(\mathbf{x}-\boldsymbol{\mu}_j)^t,\quad S_t=\sum_{\mathbf{x}\in D}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t$$

    $\boldsymbol{\mu}$为样本集D的均值；
  - 优化散布准则，可以同时最小化簇内相似度和最大化簇间相似度：

    $$\min_{C_1,\cdots,C_k} J_f(C_1,\cdots,C_k)$$

准则函数的优化

- **最优求解**
  - 聚类准则优化是对样本集划分的优化，属于组合优化问题；
  - 计算准则函数的最优解是NP难问题，将n个样本聚类为k个簇的计算复杂度为：$O(k^n/k!)$
  - 大数据集的聚类问题，寻求最优解往往是不现实的；
- **次优求解**
  - 更多的聚类分析算法寻求的是聚类准则的次优解；
  - 例如k-均值算法就是对平方误差准则的贪心次优求解；

11.2 k-均值聚类

k-means聚类算法

```
Algorithm 2 k-means聚类算法
Input: 数据集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，聚类数$k$
Output: 数据集的聚类标签$\mathcal{Y} = \{y_1, \cdots, y_n\}$
随机初始化聚类中心$\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$;
repeat
    更新聚类标签：
        $y_i = \arg \min_{1 \le j \le k} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$
    更新聚类中心：
        $\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} I(y_i = j)\mathbf{x}_i}{\sum_{i=1}^{n} I(y_i = j)}$
until $\mathcal{Y}$不再改变
```

模糊k-均值聚类

- **硬聚类与软聚类**
  - k-均值算法每一轮迭代，认为每个样本完全属于某个类别，而不属于其它类别，称为“硬聚类”；
  - “软聚类”认为样本属于任意类别，只是属于的程度不同；
- **隶属度**
  - 元素$\mathbf{x}$与集合A之间的关系可以用示性函数描述：

$$\chi_A(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in A \\ 0, & \mathbf{x} \notin A \end{cases}$$

  - 模糊数学中，元素与模糊集合A之间的关系需要用隶属度函数描述：$\chi_A(\mathbf{x}) \in [0,1]$
  - 隶属度的大小描述了元素$\mathbf{x}$属于集合A的程度；

模糊k-均值聚类

- **聚类的隶属度定义**
  - “软聚类”借鉴模糊数学的概念，聚类过程中将每个簇看作一个模糊集合；
  - 样本$\mathbf{x}$属于第j个簇的隶属度定义为：

$$\chi_j(\mathbf{x}) = \frac{\left(1/\|\mathbf{x}-\boldsymbol{\mu}_j\|^2\right)^{\frac{1}{b-1}}}{\sum_{t=1}^{k}\left(1/\|\mathbf{x}-\boldsymbol{\mu}_t\|^2\right)^{\frac{1}{b-1}}}$$

其中，$\boldsymbol{\mu}_j$为第j个簇的均值，自由参数$b>1$控制簇之间的混合程度；

  - 簇的均值计算：

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n}\chi_j^b(\mathbf{x}_i)\cdot\mathbf{x}_i}{\sum_{i=1}^{n}\chi_j^b(\mathbf{x}_i)}$$

模糊k-均值聚类算法

Algorithm 3 模糊k-均值聚类算法

```
Input: 数据集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，聚类数$k$
Output: 数据集的聚类标签$\mathcal{Y} = \{y_1, \cdots, y_n\}$
随机初始化簇均值$\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$;
repeat
    更新样本的隶属度:
        $$\chi_j(\mathbf{x}_i) = \frac{\left(1/\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2\right)^{\frac{1}{b-1}}}{\sum_{t=1}^{k}\left(1/\|\mathbf{x}_i - \boldsymbol{\mu}_t\|^2\right)^{\frac{1}{b-1}}}$$
    更新簇均值:
        $$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} \chi_j^b(\mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_{i=1}^{n} \chi_j^b(\mathbf{x}_i)}$$
until $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}$改变很小
return $y_i = \arg\max_j \chi_j(\mathbf{x}_i), \quad i = 1, \cdots, n$
```

11.3 层次聚类

层次聚类

- **Hierarchical clustering**
  - 层次聚类在不同层次对数据划分，形成树形的聚类结构；
  - 数据集的划分可以采用“自底向上”的聚合策略，也可以采用“自顶向下”的分拆策略；
- **AGNES(AGglomerative NESting)**
  - AGNES算法采用的是“自底向上”的聚合策略；
  - 初始时，将数据集中的每一个样本作为一个簇；
  - 每一轮迭代，选择距离最近的两个簇合并；
  - 直到达到预设的聚类簇个数为止；

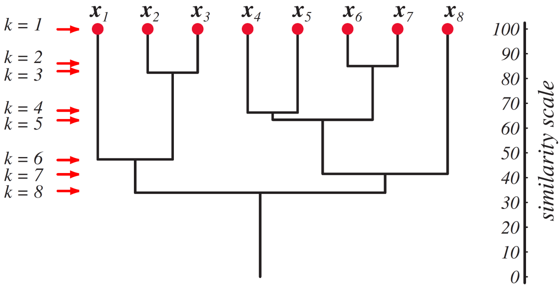
AGNES算法

Algorithm 4 AGNES算法

```
Input: 数据集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，簇距离度量函数$d$，聚类簇数$k$
Output: 簇划分$\mathcal{C} = \{C_1, \cdots, C_k\}$
初始化簇：$C_j = \{\mathbf{x}_j\}$, $j = 1, \cdots, n$；簇个数：$q = n$；
计算簇距离矩阵：$M(i,j) = M(j,i) = d(C_i, C_j)$, $i, j = 1, \cdots, n$
while $q > k$ do
    找出距离最近的两个聚类簇$C_{i^*}$和$C_{j^*}$；
    合并$C_{i^*}$和$C_{j^*}$：$C_{i^*} \leftarrow C_{i^*} \cup C_{j^*}$
    删除$M$的第$j^*$行和列，重编号$j^*$之后的聚类簇；
    重新计算$M$的第$i^*$行和列：
        $$M(i^*, j) = M(j, i^*) = d(C_{i^*}, C_j), \quad j = 1, \cdots, q-1$$
    $q \leftarrow q - 1$
end while
```

层次聚类的树型图

- **层次聚类的终止条件**
  - 聚类数达到预设值k；
  - 簇之间的最近距离大于设定的阈值；



簇的距离度量：Hausdorff距离

- **最小距离(single-linkage)**
  - 以两个簇中距离最近的两个样本的距离作为簇之间的距离：

$$d_{min}(C_i, C_j) = \min_{\mathbf{x}\in C_i, \mathbf{z}\in C_j} dist(\mathbf{x}, \mathbf{z})$$

- **最大距离(complet-linkage)**
  - 以两个簇中距离最远的两个样本的距离作为簇之间的距离：

$$d_{min}(C_i, C_j) = \max_{\mathbf{x}\in C_i, \mathbf{z}\in C_j} dist(\mathbf{x}, \mathbf{z})$$

- **平均距离(average-linkage)**
  - 以两个簇之间样本对距离的平均值作为簇之间的距离：

$$d_{min}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\mathbf{x}\in C_i} \sum_{\mathbf{z}\in C_j} dist(\mathbf{x}, \mathbf{z})$$

11.4 竞争学习

Hebb学习规则

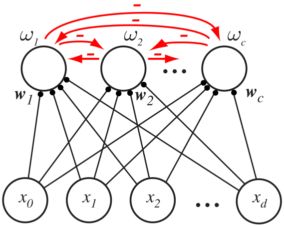
- **Hebb假设**
  - 生物学上的神经元之间由突触连接；
  - 如果一条突触两侧的神经元同时被激活，则该突触的强度将会增大；
- **Hebb学习规则**
  - 第i个神经元与第j个神经元之间的连接w_{ij}；
  - 第i个神经元的输出为p，第j个神经元的输出为a；
  - 连接强度(权值)w_{ij}依照Hebb学习规则，调整为：

$$w_{ij}^{\text{new}} \leftarrow w_{ij}^{\text{old}} + \eta a p$$

其中，η为学习率；

竞争网络

- **网络的结构**
  - 两层神经网络，包括输入层和输出层；
  - 输入层的神经元个数为$d+1$，d为输入样本的特征维数；
  - 输出层也称为竞争层，包含k个神经元，对应聚类数；



竞争层

- **侧向抑制**
  - 竞争层神经元之间有侧向抑制连接，连接权重为−1；
  - 竞争层每个神经元的输出经抑制连接传递给其它神经元，直到稳态为止；
  - 侧向抑制的最终结果是“胜者全得”，只有输出值最大的神经元被激活，输出1，其它神经元被抑制，输出0；
  - 激活的神经元称为“胜元”；

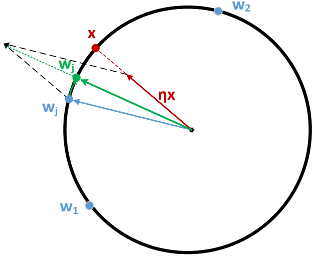
竞争学习

Algorithm 5 竞争网络学习算法

```
Input: 训练集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，聚类数$k$，学习率$\eta$
Output: 竞争层神经元的权值：$\{\mathbf{w}_1, \cdots, \mathbf{w}_k\}$
随机初始化权值$\{\mathbf{w}_1, \cdots, \mathbf{w}_k\}$;
所有训练样本和权值归一化为单位长度矢量;
repeat
    随机选取样本$\mathbf{x} \in D$;
    计算胜元：$j = \arg\max_i \mathbf{w}_i^t\mathbf{x}$
    学习权值：$\mathbf{w}_j \leftarrow \mathbf{w}_j + \eta\mathbf{x}$
    归一化权值：$\mathbf{w}_j \leftarrow \mathbf{w}_j / \|\mathbf{w}_j\|$
until 权值$\{\mathbf{w}_1, \cdots, \mathbf{w}_k\}$无明显改变
```

竞争网络

- **竞争学习**
  - 长度归一化之后的样本和权值分布在一个单位球壳上；
  - Hebb学习，相当于胜元的权值$\mathbf{w}_j$向训练样本$\mathbf{x}$方向靠近；



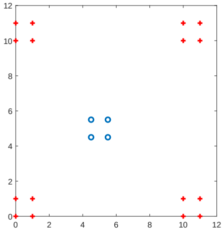
竞争学习过程

训练样本集：

$$D=\left\{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0\\10\end{pmatrix},\begin{pmatrix}0\\11\end{pmatrix},\begin{pmatrix}1\\10\end{pmatrix},\begin{pmatrix}1\\11\end{pmatrix},\right.$$
$$\left.\begin{pmatrix}10\\0\end{pmatrix},\begin{pmatrix}10\\1\end{pmatrix},\begin{pmatrix}11\\0\end{pmatrix},\begin{pmatrix}11\\1\end{pmatrix},\begin{pmatrix}10\\10\end{pmatrix},\begin{pmatrix}10\\11\end{pmatrix},\begin{pmatrix}11\\10\end{pmatrix},\begin{pmatrix}11\\11\end{pmatrix}\right\}$$

权值初始化:

$$\mathbf{w}_1=\begin{pmatrix}4.5\\4.5\end{pmatrix},\mathbf{w}_2=\begin{pmatrix}5.5\\4.5\end{pmatrix},\mathbf{w}_3=\begin{pmatrix}4.5\\5.5\end{pmatrix},\mathbf{w}_4=\begin{pmatrix}5.5\\5.5\end{pmatrix}$$



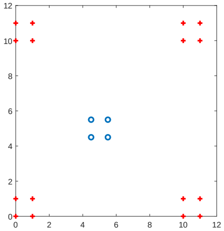
竞争学习过程

训练样本集：

$$D = \left\{ \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0\\10\end{pmatrix}, \begin{pmatrix}0\\11\end{pmatrix}, \begin{pmatrix}1\\10\end{pmatrix}, \begin{pmatrix}1\\11\end{pmatrix}, \right.$$
$$\left. \begin{pmatrix}10\\0\end{pmatrix}, \begin{pmatrix}10\\1\end{pmatrix}, \begin{pmatrix}11\\0\end{pmatrix}, \begin{pmatrix}11\\1\end{pmatrix}, \begin{pmatrix}10\\10\end{pmatrix}, \begin{pmatrix}10\\11\end{pmatrix}, \begin{pmatrix}11\\10\end{pmatrix}, \begin{pmatrix}11\\11\end{pmatrix} \right\}$$

权值初始化:

$$\mathbf{w}_1 = \begin{pmatrix}4.5\\4.5\end{pmatrix}, \mathbf{w}_2 = \begin{pmatrix}5.5\\4.5\end{pmatrix}, \mathbf{w}_3 = \begin{pmatrix}4.5\\5.5\end{pmatrix}, \mathbf{w}_4 = \begin{pmatrix}5.5\\5.5\end{pmatrix}$$



竞争网络与聚类

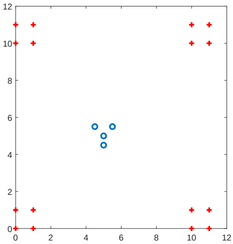
- **竞争网络用于聚类**
  - 竞争网络的学习可以看作是一种在线的k-均值聚类过程；
  - 竞争层神经元的权值代表各个聚类的均值；
  - 网络学习完成，可以再次输入训练集的样本，计算竞争层获胜神经元；
  - 以获胜神经元的标号作为样本的聚类标签：

$$y_i = \arg\max_{1 \le j \le k} \mathbf{w}_j^t \mathbf{x}_i, \quad i = 1, \cdots, n$$

竞争学习过程

学习的过程受权值初始化影响，容易出现“死元”，例如权值初始化:

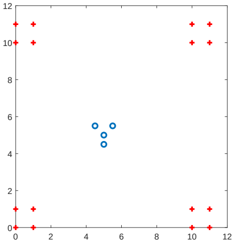
$$\mathbf{w}_1 = \begin{pmatrix} 4.5 \\ 5.5 \end{pmatrix}, \mathbf{w}_2 = \begin{pmatrix} 5.5 \\ 5.5 \end{pmatrix}, \mathbf{w}_3 = \begin{pmatrix} 5 \\ 4.5 \end{pmatrix}, \mathbf{w}_4 = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$



竞争学习过程

学习的过程受权值初始化影响，容易出现“死元”，例如权值初始化:

$$\mathbf{w}_1 = \begin{pmatrix} 4.5 \\ 5.5 \end{pmatrix}, \mathbf{w}_2 = \begin{pmatrix} 5.5 \\ 5.5 \end{pmatrix}, \mathbf{w}_3 = \begin{pmatrix} 5 \\ 4.5 \end{pmatrix}, \mathbf{w}_4 = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$

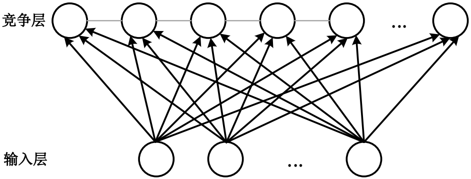


SOFM: Self-Organizing Feature Map

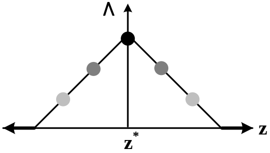
- 自组织特征映射
  - 网络的神经元之间存在一定的拓扑结构，也称为拓扑有序映射，或Kohonen网络；
  - SOFM网络的权值采用竞争的方式学习；
  - 与竞争网络不同的是，不仅获胜神经元的权值需要学习，胜元“拓扑邻域”内的神经元同样需要学习；
  - 学习的结果是使得网络具有空间拓扑有序性，SOFM既可以用于聚类，也可以用于特征降维；

网络的拓扑结构

- **1维SOFM**
  - 竞争层神经元按照1维顺序排列；
  - 学习时，邻域神经元的权值以窗函数加权调整；



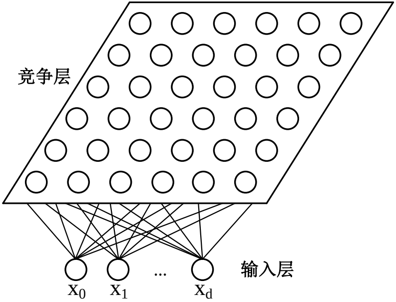
网络拓扑结构



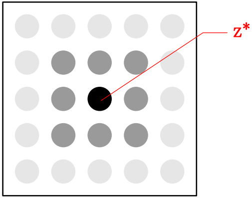
窗函数

网络的拓扑结构

- **2维SOFM**
  - 竞争层神经元按照2维网格顺序排列；
  - 学习时，邻域神经元的权值以窗函数加权调整；



网络拓扑结构



窗函数

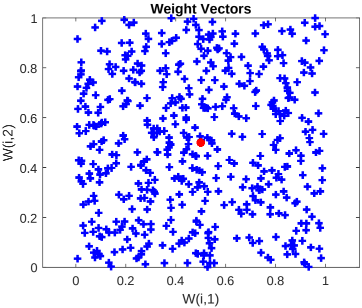
SOFM学习算法

Algorithm 6 SOFM学习算法

```
Input: 训练集$D$，网络拓扑结构，学习率$\eta$，邻域函数$\Lambda$，迭代次数$T$
Output: 竞争层神经元的权值：$\{\mathbf{w}_z\}$
随机初始化权值$\{\mathbf{w}_z\}$，$t \leftarrow 0$;
所有训练样本和权值归一化为单位长度矢量;
repeat
    随机选取样本$\mathbf{x} \in D$;
    计算胜元：$z^* = \arg\max_z \mathbf{w}_z^t \mathbf{x}$
    学习权值：$\mathbf{w}_z \leftarrow \mathbf{w}_z + \eta\Lambda(|z - z^*|)[\mathbf{x} - \mathbf{w}_z]$
    归一化权值：$\mathbf{w}_z \leftarrow \mathbf{w}_z / \|\mathbf{w}_z\|$
    $t \leftarrow t + 1$，缩小邻域$\Lambda$的范围，减小学习率$\eta$;
until $t = T$
```

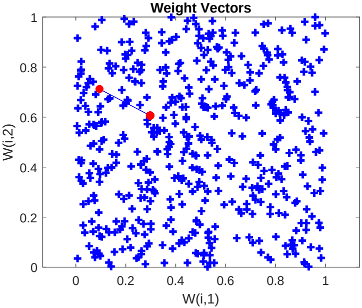
SOFM学习过程

500个训练样本均匀分布在0-1之间，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



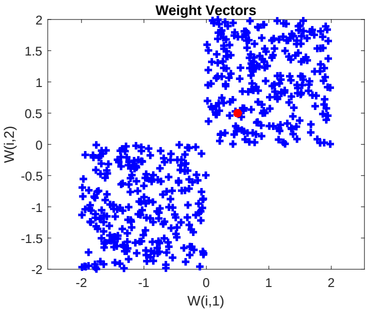
SOFM学习过程

500个训练样本均匀分布在0-1之间，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



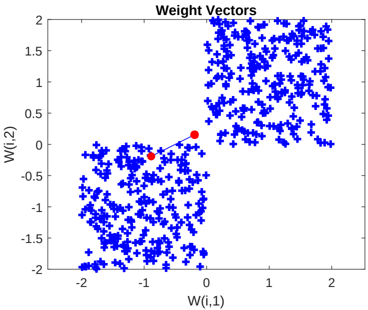
SOFM学习过程

500个训练均匀分布在-2-0和0-2两个区域，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



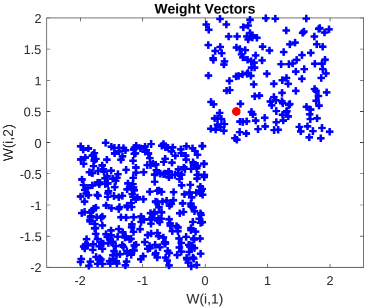
SOFM学习过程

500个训练均匀分布在-2-0和0-2两个区域，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



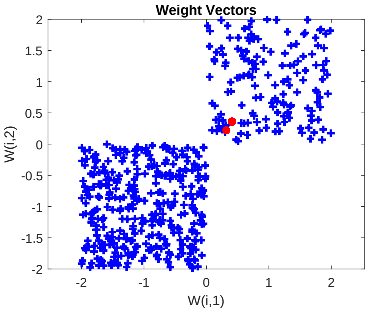
SOFM学习过程

500个训练非均匀分布在-2-0和0-2两个区域，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



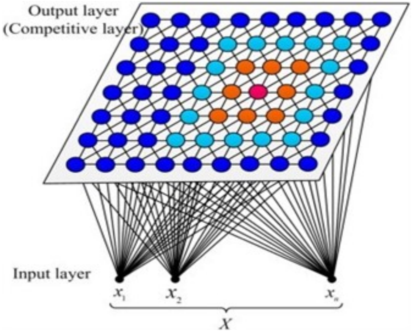
SOFM学习过程

500个训练非均匀分布在-2-0和0-2两个区域，SOFM网络包含64个神经元，排列成8×8的网格，所有神经元的权值初始化为$(0.5, 0.5)^t$；



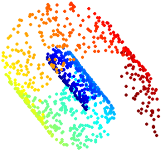
SOFM聚类

- **标记网络**
  - 网络学习完成后，需要标记网络神经元的类别；
  - 方法一：可以输入部分有监督样本，根据监督信息标注神经元的类别；
  - 方法二：输入无监督样本集，将以同一个神经元作为胜元的样本标记为一个簇；

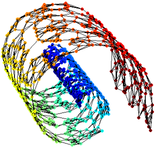


SOFM降维

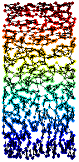
- **学习流形**
  - 样本分布在嵌入于高维空间的低维流形(超曲面)；
  - 学习一个低维结构的SOFM；
  - 计算每个样本在SOFM上的获胜神经元，投影到网络的低维空间；



样本分布流形



SOFM



低维空间映射