线性拟合——有监督回归

- 如果把样本数据采用矩阵的形式记为

$$\mathbf{X}=\begin{bmatrix}\hat{\mathbf{x}}_1^T\\ \hat{\mathbf{x}}_2^T\\ \vdots\\ \hat{\mathbf{x}}_n^T\end{bmatrix}=\begin{bmatrix}\hat{\mathbf{x}}_1^T, & 1\\ \hat{\mathbf{x}}_2^T, & 1\\ \vdots & \\ \hat{\mathbf{x}}_n^T, & 1\end{bmatrix}\qquad \mathbf{y}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix}\qquad \mathbf{f}=\begin{bmatrix}f(\mathbf{x}_1)\\ f(\mathbf{x}_2)\\ \vdots\\ f(\mathbf{x}_n)\end{bmatrix}$$

则一组测量数据方程组可整体写为

$$\begin{cases}f(\hat{\mathbf{x}}_1)=\hat{\mathbf{w}}^T\hat{\mathbf{x}}_1\\ f(\hat{\mathbf{x}}_2)=\hat{\mathbf{w}}^T\hat{\mathbf{x}}_2\\ \qquad\vdots\\ f(\hat{\mathbf{x}}_n)=\hat{\mathbf{w}}^T\hat{\mathbf{x}}_n\end{cases}\Longrightarrow\quad \mathbf{f}=\mathbf{X}\hat{\mathbf{w}}$$

- 构建**均方误差**损失函数

$$J(\hat{\mathbf{w}})=\sum_{i=1}^{n}\left(f(\mathbf{x}_i)-y_i\right)^2=(\mathbf{f}-\mathbf{y})^T(\mathbf{f}-\mathbf{y})$$

$$=(\mathbf{X}\hat{\mathbf{w}}-\mathbf{y})^T(\mathbf{X}\hat{\mathbf{w}}-\mathbf{y})$$

$$\hat{\mathbf{w}}^*=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- 岭回归：在线性回归基础上引入正则化项

$$J(\hat{\mathbf{w}})=(\mathbf{X}\hat{\mathbf{w}}-\mathbf{y})^T(\mathbf{X}\hat{\mathbf{w}}-\mathbf{y})+\lambda\|\mathbf{w}\|^2$$

$$\hat{\mathbf{w}}^*=(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

- 更一般地，对于任意单调可逆函数 g $\quad y=g(\mathbf{w}^T\mathbf{x}+b)$
- 只要令 $\quad \tilde{y}=g^{-1}(y)=\mathbf{w}^T\mathbf{x}+b$

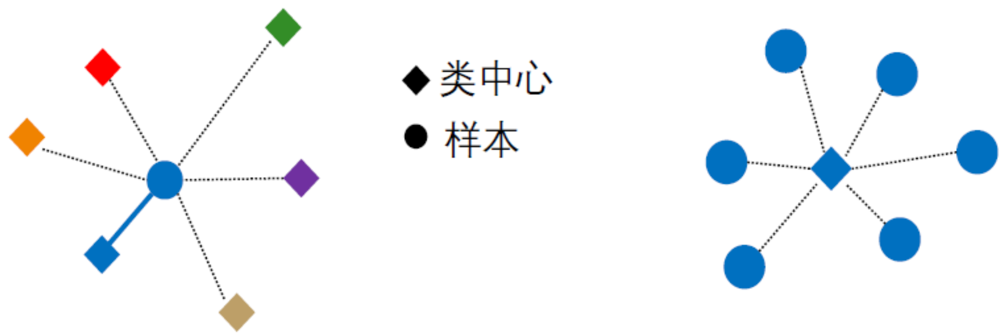
就可以使用标准的线性拟合算法（最小二乘或最大似然）进行求解。

K-means 聚类——无监督聚类

- k均值聚类目标是最小化**类内点到类中心距离**

$$J=\sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{C}_i}\left\|\mathbf{x}-\boldsymbol{\mu}_i\right\|^2=\sum_{\mathbf{x}\in\mathcal{C}_1}\left\|\mathbf{x}-\boldsymbol{\mu}_1\right\|^2+\sum_{\mathbf{x}\in\mathcal{C}_2}\left\|\mathbf{x}-\boldsymbol{\mu}_2\right\|^2+\ldots+\sum_{\mathbf{x}\in\mathcal{C}_k}\left\|\mathbf{x}-\boldsymbol{\mu}_k\right\|^2$$

- 算法分两步交替迭代，直到收敛
  - 类别划分：把每个样本划分到离它最近的中心点类
  - 计算均值：把每类的样本平均值作为新的类别中心



类别划分　　　　　　计算均值

$$j=\arg\min_{i=1,2,\ldots,k}\left\|\mathbf{x}-\boldsymbol{\mu}_i\right\|^2$$

$$\boldsymbol{\mu}_i=\sum_{\mathbf{x}_j\in\mathcal{C}_i}\mathbf{x}_j$$

输入: 样本集$D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
　　聚类簇数k.

过程:

```
从D中随机选择k个样本作为初始均值向量{μ1, μ2, ..., μk}        初始类中心
repeat
   令Ci = ∅ (1 ≤ i ≤ k)
   for j = 1, ..., m do
      计算样本xj与各均值向量μi (1 ≤ i ≤ k)的距离: dji = ||xj − μi||2;
      根据距离最近的均值向量确定xj的簇标记: λj = arg min_{i∈{1,2,...,k}} dji;     类别划分
      将样本xj划入相应的簇: Cλj = Cλj ∪ {xj};
   end for
   for i = 1, ..., k do
      计算新均值向量: μ'i = 1/|Ci| Σ_{x∈Ci} x;
      if μ'i ≠ μi then
         将当前均值向量μi更新为μ'i
      else                                                        计算均值
         保持当前均值向量不变
      end if
   end for
until 当前均值向量均未更新
return 簇划分结果
```

输出: 簇划分$\mathcal{C}=\{C_1,C_2,\ldots,C_k\}$

高斯混合模型——无监督聚类

- 高斯混合模型(Guassian Mixture Model, GMM)：假设分布中包括了 k 个高斯分布相叠加

$$p_{\mathcal{M}}(\mathbf{x})=\sum_{i=1}^{k}\alpha_i p\left(\mathbf{x}\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right) \qquad 0\le\alpha_i\le 1,\quad \sum_{i=1}^{k}\alpha_i=1$$

- A1: 极大似然估计

$$\frac{\partial LL(\boldsymbol{\theta})}{\partial\boldsymbol{\mu}_i}=\sum_{j=1}^{n}\frac{\alpha_i p\left(\mathbf{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)}{p_{\mathcal{M}}(\mathbf{x}_j)}=0 \quad\Rightarrow\quad \boldsymbol{\mu}_i=\frac{\sum_{j=1}^{n}\gamma_{ji}\mathbf{x}_j}{\sum_{j=1}^{n}\gamma_{ji}}$$

加权均值

$$\boldsymbol{\Sigma}_i=\frac{\sum_{j=1}^{n}\gamma_{ji}\left(\mathbf{x}_j-\boldsymbol{\mu}_i\right)\left(\mathbf{x}_j-\boldsymbol{\mu}_i\right)^T}{\sum_{j=1}^{n}\gamma_{ji}}$$

加权方差

$$\alpha_i=\sum_{j=1}^{n}\gamma_{ji}$$

该类平均后验概率

- 从而依次可求出每个高斯成分的参数 $\left\{\left(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)\right\},\quad i=1,2,\ldots,k$

- A2: 贝叶斯定理：给定样本 $\mathbf{x}_j\in D$，则它属于第 i 个高斯成分的概率

$$\gamma_{ji}=\frac{\alpha_i\cdot p\left(\mathbf{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)}{\sum_{l=1}^{k}\alpha_l p\left(\mathbf{x}_j\mid\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l\right)}$$

输入： 样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
高斯混合成分个数 k.

过程：

```
初始化高斯混合分布的模型参数{(α_i, μ_i, Σ_i) | 1 ≤ i ≤ k}
repeat
    for j = 1,...,m do
        根据(9.30)计算x_j由各混合成分生成的后验概率，即
        γ_ji = p_M(z_j = i | x_j) (1 ≤ i ≤ k)
    end for
    for i = 1,...,k do
        计算新均值向量：μ'_i = Σ_{j=1}^m γ_ji x_j / Σ_{j=1}^m γ_ji;
        计算新协方差矩阵：Σ'_i = Σ_{j=1}^m γ_ji (x_j − μ'_i)(x_j − μ'_i)^T / Σ_{j=1}^m γ_ji;
        计算新混合系数：α'_i = Σ_{j=1}^m γ_ji / m;
    end for
    将模型参数{(α_i, μ_i, Σ_i) | 1 ≤ i ≤ k}更新为{(α'_i, μ'_i, Σ'_i) | 1 ≤ i ≤ k}
until 满足停止条件
```

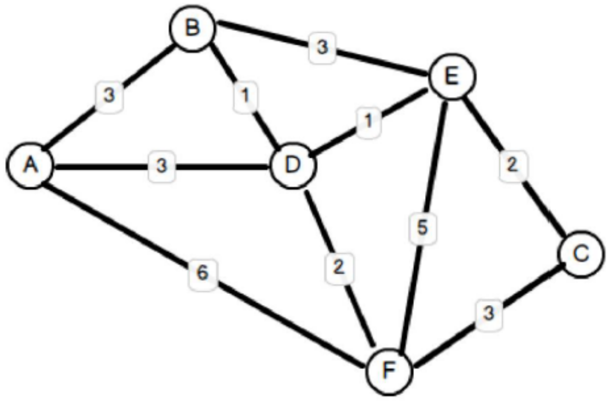
随机初始化

E步：计算后验概率

M步：求解模型参数

谱聚类——无监督聚类

- 假设有数据集 $D=\{\mathbf{x}_i\}_{i=1}^{n}$
- 图包括**节点集**(顺序无关)和对应的**边集**
- 图的邻接矩阵 $\mathbf{W}$，表示节点的**相似性**



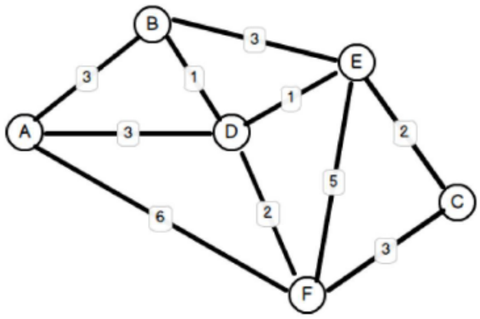
$$\mathbf{W}=\begin{array}{cc} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ & \left[\begin{array}{cccccc} 0 & 3 & 0 & 3 & 0 & 6 \\ 3 & 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 \\ 3 & 1 & 0 & 0 & 1 & 2 \\ 0 & 3 & 2 & 1 & 0 & 5 \\ 6 & 0 & 3 & 2 & 5 & 0 \end{array}\right] \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} \end{array}$$

- 常用全连接图(任意两定点点都相连)或kNN连接图(只连接最近k顶点)
- 边的权重采用 $\mathbf{W}_{ij}=\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\sigma^2\right)$

- 有了图的邻接矩阵 $\mathbf{W}$
  - 图的度矩阵，对角矩阵，每个对角元

$$\mathbf{D}_{ii}=\sum_{j=1}^{n}\mathbf{W}_{ij}$$

节点 i 的度



  - 图的拉普拉斯矩阵，正定矩阵

$$\mathbf{L}=\mathbf{D}-\mathbf{W}$$

  - 归一化拉普拉斯矩阵，正定矩阵

$$\mathcal{L}=\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$$

$$\mathbf{W}=\begin{array}{cc} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ & \left[\begin{array}{cccccc} 0 & 3 & 0 & 3 & 0 & 6 \\ 3 & 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 \\ 3 & 1 & 0 & 0 & 1 & 2 \\ 0 & 3 & 2 & 1 & 0 & 5 \\ 6 & 0 & 3 & 2 & 5 & 0 \end{array}\right] \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} \end{array}$$

- k 类情况，求解 $\mathbf{Lz}=\lambda\mathbf{Dz}$ 最小的 k 个特征值对应的特征向量（$\boldsymbol{v}_1$，$\boldsymbol{v}_2$ …. $\boldsymbol{v}_k$），排成一个 n*k 矩阵
- 每行代表一个样本（**映射样本**），对样本使用聚类算法如k-mean聚类

层次聚类（了解）——无监督聚类

- 层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。
- 数据集划分采用“自底向上”的聚合策略，或采用“自顶向下”的分拆策略。
- 自底向上的层次聚类算法
  a) 首先，将数据集中的**每一个样本**看做一个初始聚类簇，
  b) 然后在每一步中找出距离最近的两个聚类簇**进行合并**
  c) 该过程不断重复，直到达到预设的聚类簇的个数
- 自顶向下的层次聚类算法
  a) 首先，将数据集中的**所有样本**看做一个初始聚类簇，
  b) 然后在每一步中找出最大的聚类**进行拆分**(例如，可先找到该类中最远的两个样本，然后依照距离对该类中剩余样本进行划分)
  c) 该过程不断重复，直到达到预设的聚类簇的个数

朴素贝叶斯——有监督分类

- 朴素贝叶斯分类器

$$c^* = \arg\max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^{d} p(x_i \mid c)$$

- 对于给定的一组样本 $D_c = \{(\mathbf{x}_i, y_i = c)\}_{i=1}^{m}$，如果是离散属性

$$P(c) = \frac{|D_c|}{|D|}, \quad P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|}$$

D_c中第i个属性取值为x_i的样本数

如果第i个属性是连续属性，则利用最大似然估计每个属性的类概率密度函数 $p(x_i \mid c)$ ，然后可利用朴素贝叶斯分类器

线性分类器——有监督分类

- 如何把线性拟合 $z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$ 的输出转换为0-1分类?
  - 阶跃函数

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$

对数回归分类器——有监督分类

$$p(y=1\mid\mathbf{x})=\frac{e^{\mathbf{w}^T\mathbf{x}+b}}{1+e^{\mathbf{w}^T\mathbf{x}+b}}\qquad p(y=0\mid\mathbf{x})=\frac{1}{1+e^{\mathbf{w}^T\mathbf{x}+b}}$$

$$l(\mathbf{w},b)=\sum_{i=1}^{n}\ln p(y_i=j\mid\mathbf{x}_i,\mathbf{w},b)=\sum_{i=1}^{n}\left[y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)-\ln\left(1+e^{\mathbf{w}^T\mathbf{x}_i+b}\right)\right]$$

$$\begin{cases}\mathbf{w}^{(t+1)}=\mathbf{w}^{(t)}-\lambda\Delta\mathbf{w}=\mathbf{w}^{(t)}-\lambda\left.\dfrac{\partial l(\mathbf{w},b)}{\partial\mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}^{(t)},b=b^{(t)}}\\ b^{(t+1)}=b^{(t)}-\lambda\Delta b=b^{(t)}-\lambda\left.\dfrac{\partial l(\mathbf{w},b)}{\partial b}\right|_{\mathbf{w}=\mathbf{w}^{(t)},b=b^{(t)}}\end{cases}$$

$$\text{式中}\begin{cases}\dfrac{\partial l(\mathbf{w},b)}{\partial\mathbf{w}}=-\sum_{i=1}^{n}\left[\mathbf{x}_i y_i-\mathbf{x}_i p(y_i=1\mid\mathbf{x}_i,\mathbf{w},b)\right]\\ \dfrac{\partial l(\mathbf{w},b)}{\partial b}=-\sum_{i=1}^{n}\left[y_i-p(y_i=1\mid\mathbf{x}_i,\mathbf{w},b)\right]\end{cases}$$

支持向量机——有监督分类

- 有约束优化问题（原优化问题）

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{R}^d}\quad & f_0(\mathbf{x}) && \text{目标函数}\\ \text{s. t.}\quad & f_i(\mathbf{x})\le 0,\quad i=1,2,\ldots,p && \text{不等式约束}\\ & g_j(\mathbf{x})=0,\quad j=1,2,\ldots,q && \text{等式约束}\end{aligned}$$

- 拉格朗日对偶函数

$$g(\boldsymbol{\lambda},\boldsymbol{\tau})=\inf_{\mathbf{x}\in D}L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\tau})=\left(f_0(\mathbf{x})+\boldsymbol{\lambda}^T\mathbf{F}(\mathbf{x})+\boldsymbol{\tau}^T\mathbf{G}(\mathbf{x})\right)\Big|_{\mathbf{x}^*\leftarrow\nabla L_{\mathbf{x}}(\mathbf{x}^*,\boldsymbol{\lambda},\boldsymbol{\tau})=0}$$

- 对偶优化问题

$$\begin{aligned}&\max g(\boldsymbol{\lambda},\boldsymbol{\tau})\\ &s.t.\quad \lambda_i\ge 0,\quad i=1,\ldots,p\end{aligned}$$

- KKT条件：最优解的必要条件(**凸优化**时也是**充分必要**条件)
  - 原始问题约束 $f_i(\mathbf{x}) \le 0, \quad i=1,2,\ldots,p; \quad g_j(\mathbf{x})=0, \quad j=1,2,\ldots,q$
  - 对偶问题约束 $\lambda_i \ge 0, \quad i=1,\ldots,p$
  - 互补松弛 $\lambda_i f_i(\mathbf{x})=0, \quad i=1,\ldots,p$
  - 梯度消失 $\nabla L_{\mathbf{x}}\left(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\tau}\right)=0$

- 最大间隔：寻找参数 $\mathbf{w}$ 和 b，使得 γ 最大，同时满足**正确分类**的约束条件.

$$\arg\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i+b) \ge 1, i=1,2,\ldots,n$$

- 第一步：引入拉格朗日乘子 $\alpha_i \ge 0$ 得到拉格朗日函数

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\right)$$

- 第二步：令$L(\mathbf{w},b,\boldsymbol{\alpha})$ 对 $\mathbf{w}$ 和 b 的偏导为零可得

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

- 第三步：回代到 $L(\mathbf{w},b,\boldsymbol{\alpha})$ 中得到拉格朗日对偶函数，从而可得对偶优化问题

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \ge 0, \quad i=1,2,\ldots,n$$

对偶问题

- 基本思路：不断执行如下两个步骤直至收敛.
  - 第一步：选取一对需更新的变量 α_i 和 α_j .
  - 第二步：固定 α_i 和 α_j 以外的参数, 求解对偶问题更新 α_i 和 α_j
- 上面第二步：仅考虑 α_i 和 α_j 时, 对偶问题的约束变为

$$\alpha_i y_i + \alpha_j y_j = -\sum_{k \neq i,j} \alpha_k y_k, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0$$

用一个变量表示另一个变量, 再代入对偶问题中可得一个**单变量**的二次规划, 该问题具有**闭式解**. （舍弃不符合约束条件的负数解）

- 偏移项 b ：通过支持向量满足的方程（蓝线）来确定 $y_i f(\mathbf{x}_i) = 1$
- 最终判决 $y = \mathrm{sign}\left[f(\mathbf{x}_i) \right]$ ， $\mathrm{sign}[x]$表示取 x 符号

- 引入松弛变量

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$s.t. \quad y_i\left(\mathbf{w}^T \mathbf{x}_i + b\right) \geq 1 - \xi_i$$
$$\xi_i \geq 0, \quad i = 1, 2, \ldots, n$$

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{n} \alpha_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \boxed{\leq C,} \quad i = 1, 2, \ldots, n$$

- 设样本 $\mathbf{x}$ 映射后的向量为 $\phi(\mathbf{x})$ ，划分超平面为 $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$

原始问题

$$\begin{cases} \arg\min\limits_{\mathbf{w},b} \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad y_i\left(\mathbf{w}^T \phi\left(\mathbf{x}_i\right) + b\right) \geq 1, \ i = 1, 2, \ldots, n \end{cases}$$

对偶问题

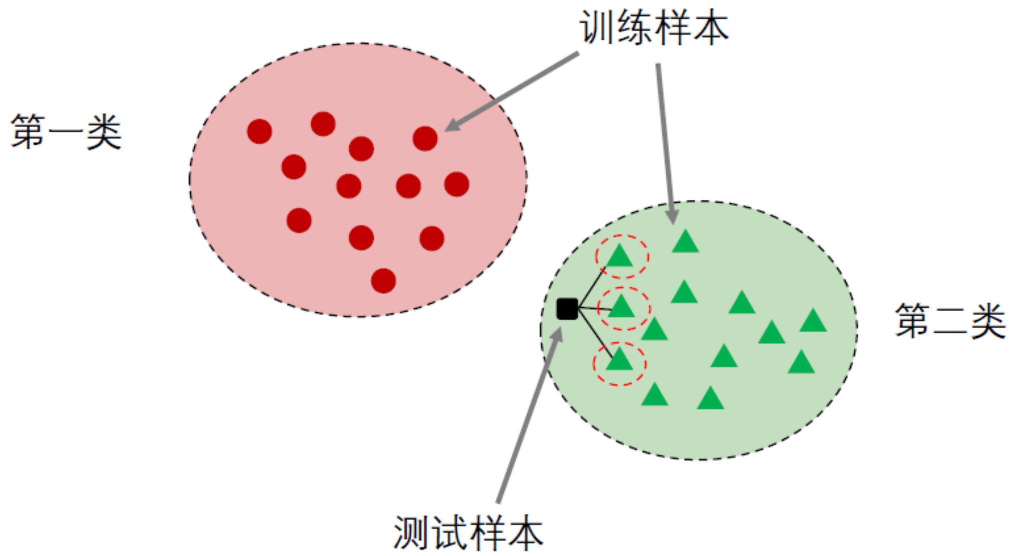
$$\begin{cases} \min\limits_{\boldsymbol{\alpha}} \dfrac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\phi\left(\mathbf{x}_i\right)^T \phi\left(\mathbf{x}_j\right)} - \sum\limits_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad \sum\limits_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, n \end{cases}$$

预测 $$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{j=1}^{n} \alpha_i y_i \boxed{\phi\left(\mathbf{x}_i\right)^T \phi\left(\mathbf{x}\right)} + b$$

只以**内积**的形式出现，不需计算出映射，只要设计**核函数** $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$

kNN——有监督分类

- k>1时，投票决定，k个样本中得票最多的类作为最终类



- Given:
  - training examples $\{x_i, y_i\}$
    - x_i ... attribute-value representation of examples
    - y_i ... class label: {ham,spam}, digit {0,1,...9} etc.
  - testing point x that we want to classify
- Algorithm:
    - compute distance $D(x,x_i)$ to every training example x_i
    - select k closest instances $x_{i1}...x_{ik}$ and their labels $y_{i1}...y_{ik}$
    - output the class y^* which is most frequent in $y_{i1}...y_{ik}$

- 用 $\mathcal{S}=\{(\mathbf{x}_i,\mathbf{x}_j) \mid y_i=y_j\}$ 表示所有同类样本对，$\mathcal{D}=\{(\mathbf{x}_i,\mathbf{x}_j) \mid y_i \neq y_j\}$ 表示所有异类样本对，则度量学习目标如下

$$\max \quad g(\mathbf{A})=\sum_{(\mathbf{x}_l,\mathbf{x}_m)\in\mathcal{D}} \|\mathbf{x}_l-\mathbf{x}_m\|_{\mathbf{A}}$$

$$\text{s. t.} \quad f(\mathbf{A})=\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2 \leq 1 \quad \leftarrow \quad C_1$$

Iterate
  Iterate
    $A := \arg\min_{A'}\{\|A'-A\|_F : A' \in C_1\}$
    $A := \arg\min_{A'}\{\|A'-A\|_F : A' \in C_2\}$
  until A converges
  $A := A + \alpha(\nabla_A g(A))_{\perp \nabla_A f}$
until convergence

决策树——有监督分类

Function *TreeGenerate* (输入数据集 $D=\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ ， 属性集 A)

- *Step 1*: 从 A 中选择一个**最优属性** a^*, 创建节点 a^*
- *Step 2*: 按 a^* 相同取值 (离散型)或区间(连续型)把 D 划分为若干互斥子集 $D_1, D_2, \cdots, D_k$.
- *Step 3*: 从 A 中移除 a^* : $A_a = A- \{a^*\}$， 对每个子集D_k递归调用自身
  TreeGenerate (输入数据集 D_k ， 属性集 A_a)

输出： 以为 a^* 根节点的一棵决策树

Algorithm 1 决策树学习基本算法

输入:
- 训练集 $D=\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A=\{a_1, \ldots, a_d\}$.

过程: 函数 TreeGenerate(D, A)

```
生成结点 node;
if D 中样本全属于同一类别 C then
    将 node 标记为 C 类叶结点; return                    (1)
end if
if A = ∅  OR  D 中样本在 A 上取值相同 then
    将 node 标记叶结点，其类别标记为 D 中样本数最多的类; return   (2)
end if
从 A 中选择最优划分属性 a*;
for a* 的每一个值 a*^v do
    为 node 生成每一个分枝; 令 D_v 表示 D 中在 a* 上取值为 a*^v 的样本子集;
    if D_v 为空 then
        将分枝结点标记为叶结点, 其类别标记为 D 中样本最多的类; return   (3)
    else
        以 TreeGenerate(D_v, A − {a*}) 为分枝结点
    end if
end for
```

输出: 以 node 为根结点的一棵决策树

三种情况下递归回终止

（1）当前结点包含的样本全部属于同一类别

（2）当前属性集为空，或所有样本在所有属性上取值相同

（3）当前结点包含的样本集合为空

- 数据集 $D=\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ 包含 k 个类样本，每类样本所占比例p_k， 则数据集 D 的信息熵定义为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

- 数据集越纯，Ent(D)越小

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \boxed{\frac{|D^v|}{|D|}} \mathrm{Ent}(D^v)$$

- 连续属性时，可选取时 Gain(D, a) 最大化属性值作为划分点

$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

- 其中

$$\text{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

- 数据集D的纯度除了用信息熵外，还可以用基尼值衡量

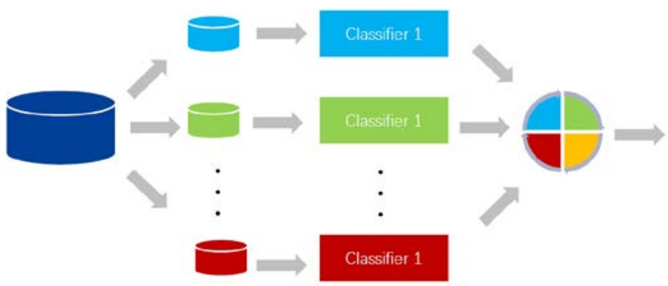
$$\text{Gini}(D)=\sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k}p_k p_{k'}=1-\sum_{k=1}^{|\mathcal{Y}|}p_k^2$$

- 基尼值越小，数据集纯度越高。
- 属性 a 的基尼指数定义为：

$$\text{Gini\_index}(D,a)=\sum_{v=1}^{V}\frac{|D^v|}{|D|}\text{Gini}(D^v)$$

- **最优属性是**应选择那个使划分后**基尼指数最小**的属性作为最优划分属性
- 预剪枝：构建树的过程中，计算划分前和划分后验证精度，如有提升，则保留该分支；否则不进行划分
- 后剪枝：构建树后，从底往上逐个考察中间节点，如果删除后验证精度有下降，则保留；否则就删除该中间节点

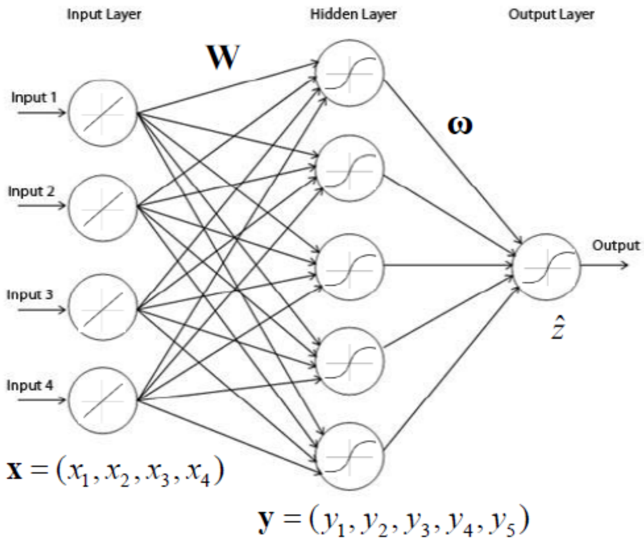
随机森林——有监督分类



- 给定数据 $D=\{(\mathbf{x}_i,y_i)\}_{i=1}^{n}$，随机森林包括三步：
  - Bagging：随机可重复地从D中选取n个样板组成一个训练集，此过程重复T次获得T个训练集D_1, D_2, ⋯, D_T
  - **训练：**每个训练集上学习一个“随机决策树”，即每个节点从随机选的k个属性中选取一个最优的属性用于划分
  - **预测：**对新样本，用T个决策树预测的结果投票(分类)或平均(回归)决定最终结果

神经网络——有监督分类

$$\begin{cases} \hat{z} = f\left(\sum_{i=1}^{5} \omega_i y_i + b_o\right) \\ y_i = f\left(\sum_{j=1}^{4} w_{ji} x_j + b_i\right), i = 1..5 \end{cases}$$



$$\boldsymbol{\omega}^{\mathbf{T}} = (\omega_1, \omega_2, ..., \omega_4)$$
$$\mathbf{y}^{\mathbf{T}} = (y_1, y_2, ..., y_5)$$
$$\mathbf{b}^{\mathbf{T}} = (b_1, b_2, ..., b_5)$$
$$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_5)$$

$$\begin{cases} \hat{z} = f\left(\boldsymbol{\omega}^T \mathbf{y} + b_o\right) & \text{网络输出} \\ \mathbf{y} = f\left(\mathbf{W}^T \mathbf{x} + \mathbf{b}\right) & \text{隐含层输出 (函数} f \text{元素运算函数)} \end{cases} \quad \begin{cases} \mathbf{y} = f(\mathbf{x}) \\ y_i = f(x_i) \end{cases}$$

	输入：训练数据集 $(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), ..., (\mathbf{x}_n, z_n)$，学习步长 λ
1	随机初始化 $\boldsymbol{\omega}, b_o, \mathbf{W}, \mathbf{b}$，设置计数器 $t = 0$
2	随机选择训练数据 $(\mathbf{x}_i, z_i)$
3	**前向计算** $\mathbf{y} = f\left(\mathbf{W}^T \mathbf{x}_i + \mathbf{b}\right)$，$\hat{z}_i = f\left(\boldsymbol{\omega}^T \mathbf{y} + b_o\right)$，$J(\mathbf{x}_i; z_i) = \frac{1}{2}(\hat{z}_i - z_i)^2$
4	**反向计算** $\Delta\boldsymbol{\omega}, \Delta b_o, \Delta\mathbf{W}, \Delta\mathbf{b}$
5	变量更新 $\begin{cases} \boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - \lambda \Delta\boldsymbol{\omega} \\ b_o^{(t+1)} = b_o - \lambda \Delta b_o \end{cases}$, $\begin{cases} \mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \lambda \Delta\mathbf{W} \\ \mathbf{b}^{(t+1)} = \mathbf{b} - \lambda \Delta\mathbf{b} \end{cases}$
6	如未满足终止条件，$t = t+1$, 重复第2步

- 交叉熵损失 (Cross Entropy Loss): $J\left(z, \hat{z}\right) = \sum_{i=1}^{n} H(\mathbf{z}_i, \hat{\mathbf{z}}_i)$

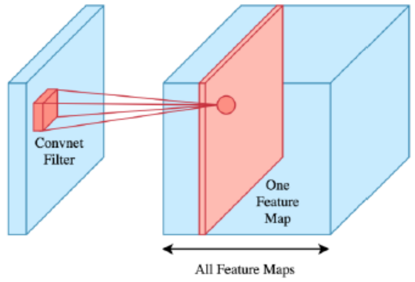
  其中交叉熵 $H(q, p)$ 衡量概率分布 q, p 的差异性

$$H(\mathbf{q}, \mathbf{p}) = -\sum_{j=1}^{C} q_j \log p_j \qquad H(q, p) = \int q(x) \log p(x) dx$$

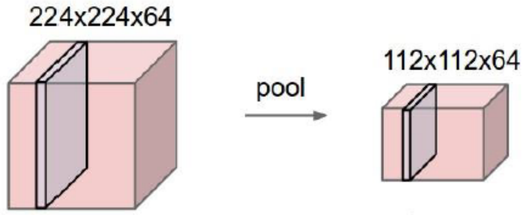
- 如何把网络输出输出 $\hat{\mathbf{z}}_i$ 转化为类别概率分布？

$$P(j \mid \hat{\mathbf{z}}) = \frac{e^{\hat{z}_j}}{\sum_{k=1}^{C} e^{\hat{z}_k}} \qquad \sum_{j=1}^{C} P(j \mid \hat{\mathbf{z}}) = \sum_{j=1}^{C} \frac{e^{\hat{z}_j}}{\sum_{k=1}^{C} e^{\hat{z}_k}} = \frac{1}{\sum_{k=1}^{C} e^{\hat{z}_k}} \sum_{j=1}^{C} e^{\hat{z}_j} = 1$$

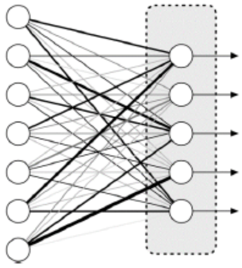
- 只要给定输入输出和上游导数，就可以求下游导数，即**梯度反向传播**



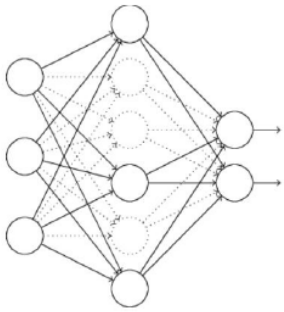
卷积层(convolutional layer)



聚集层(Pooling layer)



全连接层(fully connected layer)



丢弃层 (dropout layer)

线性判别分析 LDA——无监督降维

- 数据集 $D=\{(\mathbf{x}_i,y_i)\}_{i=1}^{n}, y\in\{0,1\}$，二分类问题
- 投影前，每类的均值和协方差矩阵

$$\begin{cases}\mathbf{u}_0=\dfrac{1}{n_0}\sum\limits_{y_i=0}\mathbf{x}_i\,, & \mathbf{\Sigma}_0=\dfrac{1}{n_0-1}\sum\limits_{y_i=0}\mathbf{x}_i\mathbf{x}_i^T\\ \mathbf{u}_1=\dfrac{1}{n_1}\sum\limits_{y_i=1}\mathbf{x}_i\,, & \mathbf{\Sigma}_1=\dfrac{1}{n_1-1}\sum\limits_{y_i=1}\mathbf{x}_i\mathbf{x}_i^T\end{cases}$$

- 投影后，每类的均值和协方差（投影到**直线**，因此均值协方差都是实数）

$$\begin{cases}\hat{u}_0=\mathbf{w}^T\mathbf{u}_0\,, & \hat{\Sigma}_0=\mathbf{w}^T\mathbf{\Sigma}_0\mathbf{w}\\ \hat{u}_1=\mathbf{w}^T\mathbf{u}_1\,, & \hat{\Sigma}_1=\mathbf{w}^T\mathbf{\Sigma}_1\mathbf{w}\end{cases}$$

- 定义类内离散度矩阵和类间离散度矩阵

$$\mathbf{S}_w=\mathbf{\Sigma}_0+\mathbf{\Sigma}_1\quad \mathbf{S}_b=(\mathbf{u}_0-\mathbf{u}_1)(\mathbf{u}_0-\mathbf{u}_1)^T$$

- 则目标函数简化为 $J=\dfrac{\mathbf{w}^T\mathbf{S}_b\mathbf{w}}{\mathbf{w}^T\mathbf{S}_w\mathbf{w}}$

$$\mathbf{w}^*=\mathbf{S}_w^{-1}(\mathbf{u}_0-\mathbf{u}_1)$$

多维缩放 MDS——无监督降维

- 给定 d 维空间中样本距离矩阵 $\mathbf{D} \in \mathbb{R}^{n \times n}$， $\mathbf{D}_{ij}$ 表示样本 $\mathbf{x}_i, \mathbf{x}_j$ 之间的距离，通常假设是欧氏距离 $\mathbf{D}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$
- 假设降维后对应的样本为 $\{\mathbf{z}_i\}_{i=1}^n \in \mathbb{R}^{d'} \quad (d' \le d)$
- 求解方法二：不直接求解低维嵌入坐标 $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$， 转而求它们的内积矩阵 $\mathbf{B} = \mathbf{Z}^T \mathbf{Z}$，其中 $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ ，并且 $\mathbf{B}_{ij} = \mathbf{z}_i^T \mathbf{z}_j$
- 设低维嵌入坐标 $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$ 均值为0: $\frac{1}{n}\sum_{i=1}^n \mathbf{z}_i = 0$

$$\mathbf{B}_{ij} = \frac{1}{2n}\left(\mathbf{D}_{i\cdot} + \mathbf{D}_{\cdot j} - n\mathbf{D}_{ij}^2 - \frac{1}{n}\mathbf{D}_{\cdot\cdot}\right)$$

- 因此给定 $\mathbf{D} \in \mathbb{R}^{n \times n}$ 后，矩阵 $\mathbf{B}$ 可求出。在利用特征分解可求出 $\mathbf{Z}$

输入: 距离矩阵 $\mathbf{D} \in \mathbb{R}^{m \times m}$, 其元素 $dist_{ij}$ 为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离;
低维空间维数 d'.

过程:

1: 根据式(10.7)–(10.9)计算 $dist_{i\cdot}^2$, $dist_{\cdot j}^2$, $dist_{\cdot\cdot}^2$;
2: 根据式(10.10)计算矩阵 $\mathbf{B}$;
3: 对矩阵 $\mathbf{B}$ 做特征值分解;
4: 取 $\tilde{\mathbf{\Lambda}}$ 为 d' 个最大特征值所构成的对角矩阵, $\tilde{\mathbf{V}}$ 为相应的特征向量矩阵.

输出: 矩阵 $\tilde{\mathbf{V}}\tilde{\mathbf{\Lambda}}^{1/2} \in \mathbb{R}^{m \times d'}$, 每行是一个样本的低维坐标

主成分分析 PCA——无监督降维

Function	PCA(**数据集D，主成分数** k)
1	去中心化 $\mathbf{m} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i; \quad \hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{m}, \quad i = 1, \ldots, n$
2	求协方差矩阵 $\mathbf{C} = \frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^T$
3	特征分解 $\mathbf{C}\mathbf{w} = \lambda \mathbf{w}$ 求出前 k 个特征对 $(\lambda_i, \mathbf{w}_i), i = 1 \ldots k$
4	计算前k个主成分 $\mathbf{Y} = \mathbf{W}^T \mathbf{X}, \quad \mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k)$
5	输出 $\mathbf{Y}, \mathbf{W}$

核化主成分分析 KPCA（了解）——无监督降维

- 映射变换 $\mathbf{z}=\phi(\mathbf{x})$ 后，在特征空间中寻找主成分

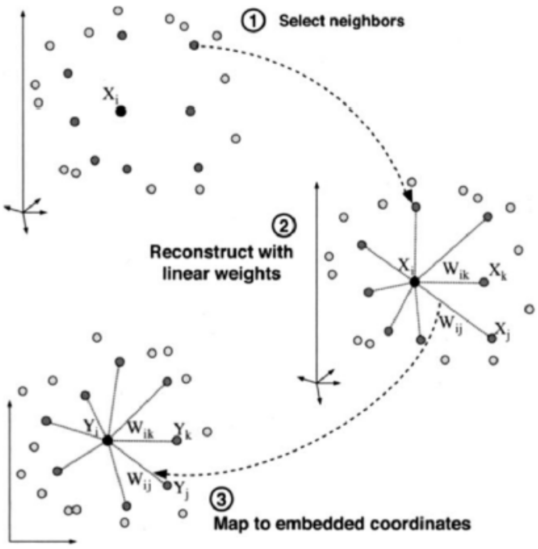
$$K_{ij} := (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) \qquad \mathbf{K}\boldsymbol{\alpha}=\lambda\boldsymbol{\alpha}$$

等度量映射 Isomap（了解）——无监督降维

- 计算图上任意两点间的最小图距离 $g_{ij}(\mathbf{x}_i,\mathbf{x}_j)$
- 对图距离应用经典的多维度缩放MDS算法

$$\min E(Y)=\min\sum_{i,j}(\|y_i-y_j\|-d_g(x_i,x_j))^2$$

局部线性嵌入 LLE（了解）——无监督降维



Step 2:

$$\phi(W)=\sum_i\left|\vec{X}_i-\sum_j W_{ij}\vec{X}_j\right|^2$$

$$s.t.\quad \sum_j W_{ij}=1$$

Step 3:

$$\Phi(Y)=\sum_i\left|\vec{Y}_i-\sum_j W_{ij}\vec{Y}_j\right|^2$$

$$s.t.\ \sum_i\vec{Y}_i=0;\ \frac{1}{N}\sum_i\vec{Y}_i\otimes\vec{Y}_i=I$$