



# 机器学习面试

## 如何做梯度并行运算

### 试图跳出局部最小的方法

1. 以多组不同参数值初始化多个神经网络，按标准方法（梯度下降）训练后，取其中误差最小的解作为最终参数
2. 模拟退火：在每一步都以一定的概率接受比当前解更差的结果
3. 随机梯度下降
4. 遗传算法

### 关于决策树

最优属性的划分基本思想：希望分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”要越来越高

- 信息增益：

信息熵： $Ent(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$ ，其中 $p_k$ 为当前样本集合D中第k类样本所占比例。

熵越小，D的纯度越高

用属性a对样本集D进行划分的信息增益： $Gain(D, a) = Ent(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} Ent(D^v)$

其中，属性a有V个不同取值 $\{a^1, a^2, \dots, a^v\}$ ，

若使用a划分则会产生V个不同分支点，其中 $D^v$ 第v个分支点包含D中所有在属性a上取值为 $a^v$ 的样本的

- 增益率

$$Gain_{ratio}(D, a) = \frac{Gain(D, a)}{IV(a)}$$

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

- 基尼系数

基尼值

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

$$= 1 - \sum_{k=1}^{|Y|} p_k^2$$

则属性a的基尼指数为  $Gini_{index}(D, a) = \sum_{v=1}^V \frac{|D^v|}{|D|} Gini(D^v)$ ，选择那个使得划分后的基尼指数最小的属性作为最优划分属性

剪枝：

- 预剪枝，在决策树生成中就剪
- 后剪枝，生成一颗完整的决策树后，自底向上对非叶节点进行考虑

一般来说，后剪枝欠拟合风险较小，泛化性能往往优于预剪枝

## 关于SVM

1. 目标函数是啥

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2$$

$$s.t. \quad y_i(\omega^T x_i + b) \geq 1, i = 1, 2, \dots, m.$$

2. 什么是KKT条件

### 3. 关于核函数

若不存在一个超平面将训练样本正确分类（线性可分），可以将样本从原始空间映射到一个更高维的特征空间使之线性可分（一定存在），如  $x \mapsto \phi(x)y$

有

$$\begin{aligned} \min_{\omega, b} \quad & \frac{1}{2} \|\omega\| \\ \text{s.t.} \quad & y_i (\omega^T \phi(x_i) + b) \geq 1, i = 1, 2, \dots, m. \end{aligned}$$

但化为对偶问题，需要求内积  $\phi(x_i)^T \phi(x_j)$ ，由于特征空间维数可能很高，甚至无穷维，因此计算困难，这时希望能找到一个函数： $\kappa(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ ，即向量在特征空间的内积等于它们在原始样本空间通过函数  $\kappa$  计算的结果，这个函数就称为核函数。

只要一个对称函数所对应的核矩阵是半正定的，它就能作为核函数。

常用核函数：

表 6.1 常用核函数

名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d$	$d \geq 1$ 为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^T \mathbf{x}_j + \theta)$	$\tanh$ 为双曲正切函数, $\beta > 0, \theta < 0$

以下这些组合也是核函数：

- 核函数的线性组合  $\gamma_1 \kappa_1 + \gamma_2 \kappa_2$
- 核函数的直积  $\kappa_1 \otimes \kappa_2(x, z) = \kappa_1(x, z) \kappa_2(x, z)$
- 对任意函数  $g(x)$ ，有  $\kappa(x, z) = g(x) \kappa_1(x, z) g(z)$

特征空间的好坏对SVM的性能至关重要，而核函数也隐式定义了这个空间。经验：对文本数据通常采用线性核，情况不明时可先尝试高斯核

#### 4. 软间隔

核函数往往难以确定，可以允许某些样本不满足约束 $y_i(\omega^T x_i + b) \geq 1$

因为在最大化间隔时不满足约束的样本应尽可能少，于是目标函数可写为：

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m l_{0/1}(y_i(\omega^T x_i + b) - 1)$$

$$l_{0/1}(z) = \begin{cases} 1 & , \text{ if } z < 0 \\ 0 & , \text{ otherwise} \end{cases}$$

$l$ 是0/1损失函数

#### 5. 支持向量回归 (SVR)

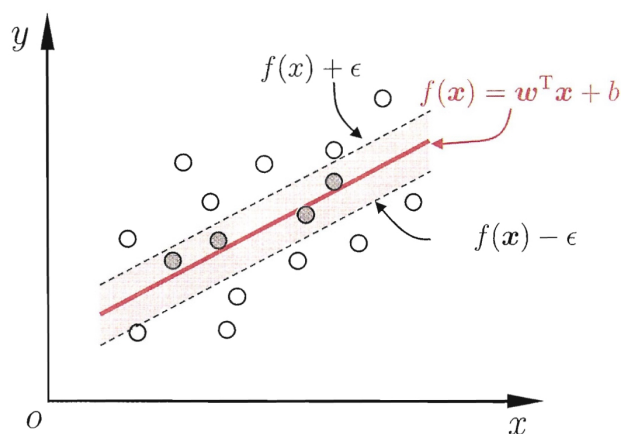


图 6.6 支持向量回归示意图. 红色显示出  $\epsilon$ -间隔带, 落入其中的样本不计算损失.

于是SVR可形式化为：

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m l_{\epsilon}(f(x_i) - y_i)$$

$f(x)$ 是要学的模型

$l_\epsilon$ 是不敏感损失函数： $l_{0/1}(z) = \begin{cases} 1, & \text{if } |z| \leq \epsilon \\ |z| - \epsilon, & \text{otherwise} \end{cases}$

## 贝叶斯分类器

- 贝叶斯决策论

$N$ 种可能的标记 $\{c_1, c_2, \dots, c_N\}$

$\lambda_{ij}$ 为将一个真实标记为 $c_j$ 的样本 $x$ 误分类为 $c_i$ 所产生的损失，则

1. 样本 $x$ 的条件风险（期望损失即为风险）为： $R(c_i|x) = \sum_{j=1}^N \lambda_{ij} P(c_j|x)$
2. 我们的任务是寻找一个判定准则 $h$ 以最小化总体风险： $R(h) = \mathbb{E}_x[R(h(x)|x)]$
3. 贝叶斯判定准则：为了最小化总体风险，只需在每个样本上选择那个能使条件风险 $R(c|x)$ 最小的类别标记，即 $h^*(x) = \arg \min R(c|x)$ ，此时 $h^*$ 称为贝叶斯最优分类器，与之对应的风险为贝叶斯风险
4. 若目标是最小化分类错误率（ $\arg \min R(c|x)$ ），则误判损失可写为：

$$\lambda_{ij} = \begin{cases} 0, & \text{if } i == j \\ 1, & \text{otherwise} \end{cases}$$

5. 此时条件风险为 $R(c_i|x) = 1 - P(c_i|x)$ ，所以贝叶斯最优分类器为：

$$h^*(x) = \arg \min (1 - P(c|x)), \text{ 即 } h^*(x) = \arg \max P(c|x)$$

剩下就是如何计算后验概率 $P(c|x)$ 的问题了

## 判别式模型和生成式模型

- 判别式模型：给定 $x$ ，通过直接建模 $P(c|x)$ 来预测 $c$ ，如决策树，神经网络，支持向量机等
- 生成式模型：先对联合概率分布建模 $P(x, c)$ ，再通过贝叶斯定理获得 $P(c|x)$ ：

$P(c|x) = \frac{P(c)P(x|c)}{P(x)}$ ，其中 $P(c)$ 为先验概率， $P(x|c)$ 可称为似然函数， $P(x)$ 是归一化证据因子（evidence）（对于连续型属性，可将概率质量函数 $P$ 换成概率密度函数 $p$ ，大 $P$ 变小 $p$ ）

## 极大似然估计（MLE）

- 频率主义学派：认为参数虽然未知，但却是客观存在的固定值，因此可通过优化似然函数来确定参数值

做法：假设样本独立同分布，对于第 $c$ 类的样本集 $D_c$ 有 $P(D_c|\theta) = \prod_{x \in D_c} P(x|\theta)$

，为防止数据下溢，通常使用对数似然 $LL(\theta) = \log P(D_c|\theta) =$

$\sum_{x \in D_c} \log P(x|\theta)$ ，然后：

$$\hat{\theta} = \arg \max_{\theta} LL(\theta)$$

- 贝叶斯学派：认为参数是未观察的随机变量，其本身也可有分布EM

## 朴素贝叶斯分类器

假设所有属性相互独立，则， $d$ 为属性的数量

所以此时贝叶斯判定准则为

对于离散属性，可使用高斯分布 $p(x_i|c) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

## EM算法

常用于估计参数隐变量，是一种迭代式方法

基本思想：若模型参数 $\theta$ 已知，则可根据训练数据推断出最优隐变量 $Z$ 的值（E步）；反之，若 $Z$ 的值已知，则可方便地对参数 $\theta$ 做极大似然估计（M步）。具体的以 $\theta^0$ 为起点，迭代以下步骤直至收敛：

- 基于 $\theta^t$ 推断隐变量 $Z$ 的期望，记为 $Z^{t+1}$
- 基于已观测变量 $X$ 和 $Z^{t+1}$ ，对参数 $\theta$ 做极大似然估计，记为 $\theta^{t+1}$

## 关于集成学习

测试例1	测试例2	测试例3	测试例1	测试例2	测试例3	测试例1	测试例2	测试例3
$h_1$	✓	✓	×	$h_1$	✓	✓	×	×
$h_2$	×	✓	✓	$h_2$	✓	✓	×	×
$h_3$	✓	×	✓	$h_3$	✓	✓	×	✓
集成	✓	✓	✓	集成	✓	✓	×	×

(a) 集成提升性能

测试例1	测试例2	测试例3	测试例1	测试例2	测试例3	测试例1	测试例2	测试例3
$h_1$	✓	✓	×	$h_1$	✓	✓	×	×
$h_2$	✓	✓	×	$h_2$	✓	✓	×	×
$h_3$	✓	✓	×	$h_3$	✓	✓	×	✓
集成	✓	✓	×	集成	✓	✓	×	×

(b) 集成不起作用

测试例1	测试例2	测试例3	测试例1	测试例2	测试例3	测试例1	测试例2	测试例3
$h_1$	✓	×	×	$h_1$	✓	×	×	×
$h_2$	×	✓	×	$h_2$	×	✓	×	×
$h_3$	×	×	✓	$h_3$	×	×	✓	✓
集成	×	×	×	集成	×	×	×	×

(c) 集成起负作用

图 8.2 集成个体应“好而不同” ( $h_i$  表示第  $i$  个分类器)

可见，要获得好的集成，个体学习器应“好而不同”：个体学习器要有一定的“准确性”，并且要有“多样性”。

然而，现实任务中，准确性和多样性本身就存在冲突：个体学习器是为解决同一个问题训练出来的，难以多样性。一般的，准确性很高之后，要增加多样性就需要牺牲准确性。

## AdaBoost

- 每一轮如何改变训练数据的权值或概率分布？

提高那些被前一轮弱分类器错误分类样本的权值，降低那些被正确分类样本的权值

- 如何将弱分类器组合成一个强分类器？

加权多数表决，加大分类误差率小的弱分类器的权值，使其在表决中起较大的作用，减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。

算法过程：

给定二分类数据集  $T = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N), y \in \{-1, +1\}$

1. 初始化训练数据权重分布  $D_1 = (w_{11}, w_{1i}, \dots, w_{1N}), w_{1i} = \frac{1}{N}$

2. 对  $m=1, 2, \dots, M$ ，反复学习基本分类器：

2.1 使用具有权值分布  $D_m$  的训练数据集，得到基本分类器  $G_m(x) \rightarrow \{-1, +1\}$

2.2 计算  $G_m(x)$  在训练数据集上的误差率：

$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i)$ ，经计算  $0 < e_m \leq \frac{1}{2}$ ，若误差率大于0.5，则退出迭代

2.3 计算  $G_m(x)$  的系数  $\alpha_m = \frac{1}{2} \log \frac{1-e_m}{e_m}$ ，用于计算分类器  $G_m(x)$  在最终分类器中的重要程度

2.4 更新下一轮训练数据集权重分布

$$D_{m+1} = (w_{m+1,1}, w_{m+1,2}, \dots, w_{m+1,N})$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} e^{-\alpha_m y_i G_m(x_i)}, i = 1, 2, \dots, N$$

其中  $Z_m$  为规范化因子： $Z_m = \sum_{i=1}^N w_{mi} e^{-\alpha_m y_i G_m(x_i)}$

权重更新解释：

$$w_{m+1,i} = \begin{cases} \frac{w_{mi}}{Z_m} e^{-\alpha_m} & \text{if } G_m(x_i) = y_i \\ \frac{w_{mi}}{Z_m} e^{\alpha_m} & \text{if } G_m(x_i) \neq y_i \end{cases}$$

当  $G_m(x_i) = y_i$  表示基分类器预测准确，此时两者同号，则  $0 < e^{-\alpha_m} \leq 1$ ，从而降低了正确分类样本的权重；当  $G_m(x_i) \neq y_i$  表示基分类器预测错误，两者异号，则  $e^{\alpha_m} > 1$ ，提高错误分类样本的权重

3. 构建基分类器线性组合  $f(x) = \sum_{m=1}^M \alpha_m G_m(x)$ ，得最终分类器： $G(x) = \text{sign}(f(x))$

可以证明AdaBoost的训练误差呈指数级下降

AdaBoost只适用于二分类

## Bagging

基本思想：先随机取出一个样本放入采样集中，再把该样本放回初始数据集，使得下次采样时该样本仍有可能被选中（互相有交叠）。然后采样出T个含有m个训练样本的采样集后，基于每个采样集训练一个基学习器，再将这些定义



---

输入：训练集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ ;  
基学习算法  $\mathcal{L}$ ;  
训练轮数  $T$ .

过程:

1: for  $t = 1, 2, \dots, T$  do  
2:    $h_t = \mathcal{L}(D, \mathcal{D}_{bs})$   
3: end for

输出:  $H(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(\mathbf{x}) = y)$

---

图 8.5 Bagging 算法

最后对预测输出结合时，Bagging通常对分类任务使用简单的投票法，对回归任务使用简单平均法。

Bagging可以不经修改适用于多分类、回归等任务

## 随机森林

是Bagging一个变体，以基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入随机属性的选择，对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含 $k$ 个属性的子集，然后再从该子集中选择一个最优属性用于划分（传统决策树在选择划分属性时是在当前结点的属性集合中选择一个最优属性）

## 对学习器结合的方法

平均法

投票法

学习法：通过另一个学习器来进行结合。个体学习器称为初级学习器，用于结合的学习器称为次级学习器或元学习器

## 关于聚类（可参考数据挖掘期末论文）

性能指标：中心思想：簇内相似度高，簇间相似度低

设聚类算法将样本划分为若干个互不相交的簇 $\{C_l | l = 1, 2, \dots, k\}$ ，簇的标记为 $\lambda_j \in \{1, 2, \dots, k\}$

- 外部指标度量，将聚类结果与某个“参考模型”进行比较：

定义：

样本对在模型C中划分相同，在模型C\*中划分也相同

$$\left\{ \begin{array}{l} a = |SS|, \quad SS = \{(x_i, x_j) \mid \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}, \quad (9.1) \\ b = |SD|, \quad SD = \{(x_i, x_j) \mid \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}, \quad (9.2) \\ c = |DS|, \quad DS = \{(x_i, x_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}, \quad (9.3) \\ d = |DD|, \quad DD = \{(x_i, x_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}, \quad (9.4) \end{array} \right.$$

这里可能用混淆矩阵更好理解：

	Positive（真实环境）	Negative（真实环境）
Positive（预测环境）	TP=a	FP=b
Negative（预测环境）	FN=c	TN=d

Jaccard系数： $JC = \frac{a}{a+b+c} = \frac{TP}{TP+FP+FN}$ （即Jaccard相似度  
 $JaccardSimilarity = \frac{|A \cap B|}{|A \cup B|}$ ）

FM指数： $FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}$

Rand指数： $RI = \frac{2(a+b)}{m(m-1)}$

以上结果这[0,1]区间，值越大越好

- 内部指标：直接考察聚类结果而不利用任何参考模型

定义：

$$\text{avg}(C) = \frac{2}{|C|(|C| - 1)} \sum_{1 \leq i < j \leq |C|} \text{dist}(\mathbf{x}_i, \mathbf{x}_j), \quad (9.8)$$

$$\text{diam}(C) = \max_{1 \leq i < j \leq |C|} \text{dist}(\mathbf{x}_i, \mathbf{x}_j), \quad (9.9)$$

$$d_{\min}(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} \text{dist}(\mathbf{x}_i, \mathbf{x}_j), \quad (9.10)$$

$$d_{\text{cen}}(C_i, C_j) = \text{dist}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j), \quad (9.11)$$

DB指数： $DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{\text{avg}(C_i) + \text{avg}(C_j)}{d_{\text{cen}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)} \right)$ ，显然值越小越好

Dunn指数： $DI = \min_{i \leq i \leq k} \left\{ \min_{j \neq i} \left( \frac{d_{\min}(C_i, C_j)}{\max_{1 \leq l \leq k} \text{diam}(C_l)} \right) \right\}$ ，显然值越大越好

#### • 距离计算

闵可夫斯基距离（即 $L_p$ 范数 $\|\mathbf{x}_i - \mathbf{x}_j\|_p$ ）

$$\text{dist}_{mk}(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{u=1}^n |\mathbf{x}_{iu} - \mathbf{x}_{ju}|^p \right)^{\frac{1}{p}}$$

$p = 1$  为曼哈顿距离

$p = 2$  为欧氏距离

对于有序属性如{1,2,3}可采用闵可夫斯基距离

对于无序属性如{飞机，火车，轮船}可采用VDM(value difference metric)：

$$VDM_p(a, b) = \sum_{i=1}^k \left| \frac{m_{u,a,i}}{m_{u,a}} - \frac{m_{u,b,i}}{m_{u,b}} \right|^p, \text{ 其中 } m_{u,a} \text{ 表示属性 } u \text{ 上取值为 } a \text{ 的样本数,}$$

$m_{u,a,i}$  为第  $i$  个簇中这属性  $u$  上取值为  $a$  的样本数

$$\text{余弦相似度: } \cos(\alpha) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$$

$$\text{皮尔逊相关系数 } p = \frac{\text{cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{D(\mathbf{x}_i)} \sqrt{D(\mathbf{x}_j)}}$$

$$\text{马氏距离 } m = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T S^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$$

- K-means

目标：最小化平方误差  $E = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|_2^2$ ，但这是一个NP难问题

使用贪心策略，不断寻找均值向量的过程

需要假定聚类数

---

```

输入：样本集  $D = \{x_1, x_2, \dots, x_m\}$ ；
      聚类簇数  $k$ .
过程：
1: 从  $D$  中随机选择  $k$  个样本作为初始均值向量  $\{\mu_1, \mu_2, \dots, \mu_k\}$ 
2: repeat
3:   令  $C_i = \emptyset$  ( $1 \leq i \leq k$ )
4:   for  $j = 1, 2, \dots, m$  do
5:     计算样本  $x_j$  与各均值向量  $\mu_i$  ( $1 \leq i \leq k$ ) 的距离:  $d_{ji} = \|x_j - \mu_i\|_2$ ;
6:     根据距离最近的均值向量确定  $x_j$  的簇标记:  $\lambda_j = \arg \min_{i \in \{1, 2, \dots, k\}} d_{ji}$ ;
7:     将样本  $x_j$  划入相应的簇:  $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$ ;
8:   end for
9:   for  $i = 1, 2, \dots, k$  do
10:    计算新均值向量:  $\mu'_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ ;
11:    if  $\mu'_i \neq \mu_i$  then
12:      将当前均值向量  $\mu_i$  更新为  $\mu'_i$ 
13:    else
14:      保持当前均值向量不变
15:    end if
16:  end for
17: until 当前均值向量均未更新
输出：簇划分  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ 

```

---

图 9.2  $k$  均值算法

时间复杂度  $O(t \cdot K \cdot N)$ ， $t$  为迭代次数， $K$  为簇数量， $N$  为样本数量

- LVQ，学习向量量化

需要假定样本有类别标记，即给定样本集为  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$

算法随机选取一个有标记的训练其中  $z$

需要假定原型向量数（也即聚类数）

	输入: 样本集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ; 原型向量个数 $q$ , 各原型向量预设的类别标记 $\{t_1, t_2, \dots, t_q\}$ ; 学习率 $\eta \in (0, 1)$ .
	过程:
	1: 初始化一组原型向量 $\{p_1, p_2, \dots, p_q\}$
	2: <b>repeat</b>
	3: 从样本集 $D$ 随机选取样本 $(x_j, y_j)$ ;
	4: 计算样本 $x_j$ 与 $p_i$ ( $1 \leq i \leq q$ ) 的距离: $d_{ji} = \ x_j - p_i\ _2$ ;
	5: 找出与 $x_j$ 距离最近的原型向量 $p_{i^*}$ , $i^* = \arg \min_{i \in \{1, 2, \dots, q\}} d_{ji}$ ;
$x_j$ 与 $p_{i^*}$ 的类别相同.	6: <b>if</b> $y_j = t_{i^*}$ <b>then</b>
	7: $p' = p_{i^*} + \eta \cdot (x_j - p_{i^*})$
$x_j$ 与 $p_{i^*}$ 的类别不同.	8: <b>else</b>
	9: $p' = p_{i^*} - \eta \cdot (x_j - p_{i^*})$
	10: <b>end if</b>
如达到最大迭代轮数.	11: 将原型向量 $p_{i^*}$ 更新为 $p'$
	12: <b>until</b> 满足停止条件
	输出: 原型向量 $\{p_1, p_2, \dots, p_q\}$

图 9.4 学习向量量化算法

算法第7行, 若最近的原型向量 $p_{i^*}$ 与 $x_j$ 的类标记相同, 则令 $p_{i^*}$ 向 $x_j$ 方向靠拢, 可以看到此时 $p'$ 与 $x_j$ 的距离为 $\|p' - x_j\| = \|p_{i^*} + \eta(x_j - p_{i^*}) - x_j\|_2 = (1 - \eta)\|p_{i^*} - x_j\|_2$

算法第9行, 若类别标记不同, 则将原型向量与 $x_j$ 的距离增大为 $(1 + \eta)\|p_{i^*} - x_j\|_2$

- K-mediods

K-mediods每次选取的中心点必须是样本点, 而K-means选取的中心点可以是样本点以外的点。

K-mediods选取中心点时, 需要遍历簇中所有的点, 在前簇中选择所有其他点到中心点距离之和最小的点为中心点。通过这样的方式K-mediods可以大大减少异常点对聚类结果影响。

时间复杂度 $O(k(n - k)^2)$

- DBSCAN

一种基于密度的聚类算法, 考察样本之间的紧密程度譬如可连接性, 并基于可连接性将样本不断扩展聚类簇以获得最终的聚类效果。

- 层次聚类算法 (Hierarchical Clustering)

在不同层次上对数据集进行划分，形式树形结构。数据集的划分可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是一种自底向上的层次聚类算法。它先将数据集中的每个样本看作是一个个初始的簇，然后在算法运行过程中的每一步找出距离最近的两个聚类簇进行合并，不断重复直到达到预设的聚类簇个数。

关于簇之间距离的计算可用最小距离（两个簇最近的样本决定），最大距离（两个簇最远的样本决定），平均距离等衡量（两个簇所有样本决定）

## 降维

一般来说，欲获得低维子空间，最简单的是对原始高维空间进行线性变换： $Z = W^T X$ ,

其中 $W$ 是变换矩阵， $Z$ 是新空间中的表示（ $d'$ 维）， $X$ 是原空间中的表示（ $d$ 维）， $W$ 可视为 $d'$ 个 $d$ 维的基向量， $z_i = W^T x_i$ 是第 $i$ 个样本与这 $d'$ 个基向量分别做内积而得到的 $d'$ 维属性向量。

## PCA——主成分分析

考虑：对于正交属性空间中，如何用一个超平面对所有的样本进行恰当的表达？

这样的超平面大概具有这样的性质：

- 最近重构性：样本点到这个超平面的距离都足够近
- 最大可分性：样本到这个超平面的投影能尽可能分开

从最大可分性出发，样本点 $x_i$ 在新空间中超平面上的投影 $z_i = W^T x_i$ 要尽可能分开，则应该使投影后的样本点的方差 $\sum_i W^T x_i x_i^T W$ 最大化，即优化目标为（注意约束是 $w_j$ 是标准正交基， $\|w_j\|_2 = 1, w_i^T w_j = 0$ ）：

$$\begin{aligned} \max_W & \text{tr}(W^T X X^T W) \\ \text{s.t.} & W^T W = I \end{aligned}$$

使用拉格朗日乘子法可得： $X X^T W = \alpha W$

对协方差矩阵  $XX^T$  进行特征值分解（或者对  $X$  进行奇异值分解），将求得的特征值由大到小排序，再取前  $d'$  个特征值对应的特征向量构成  $W = \{w_1, w_2, \dots, w_{d'}\}$  获得投影矩阵