根据学校课堂纪律的要求

请同学们坐在前五排

数字媒体技术基础

Meng Yang

www.smartllv.com

SUN YAT-SEN University

机器智能与先进计算教育部重点实验室

智能视觉语言学习研究组

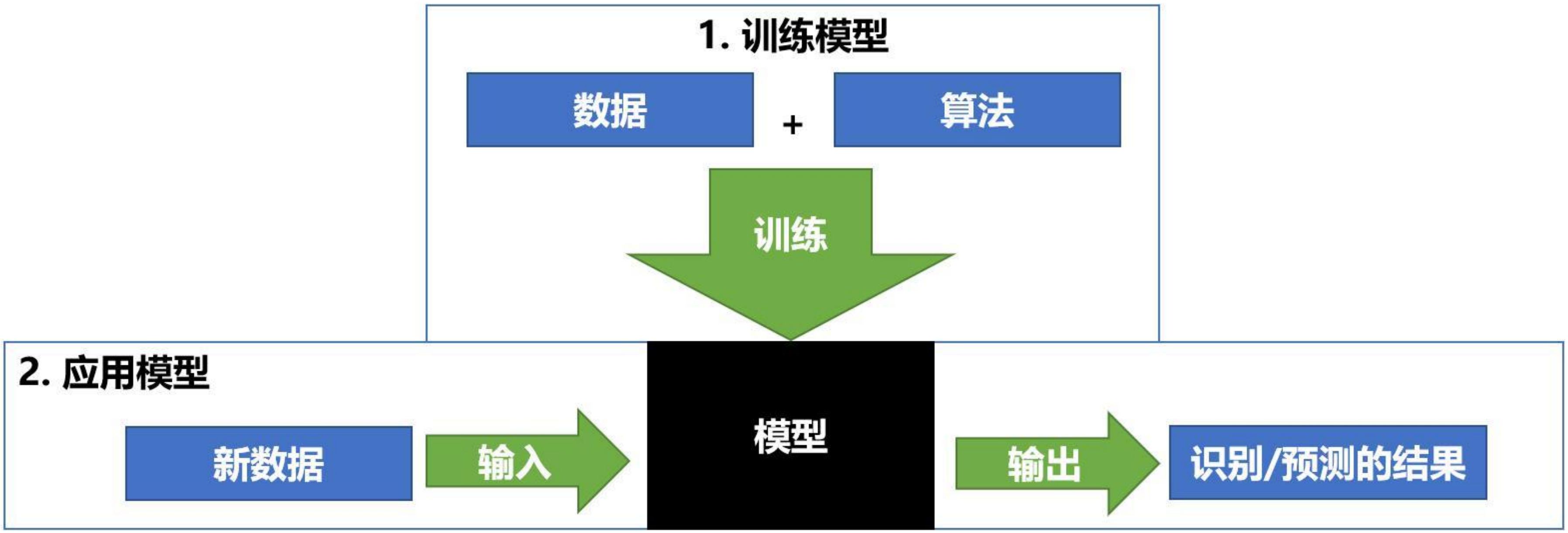
Course Outline

- 8. 数字媒体分类技术
- 8.1 传统分类算法
  - 8.1.1 K-近邻分类器
  - 8.1.2 支持向量机
  - 8.1.3 稀疏协同表示分类器

第一部分

8.1 传统分类算法

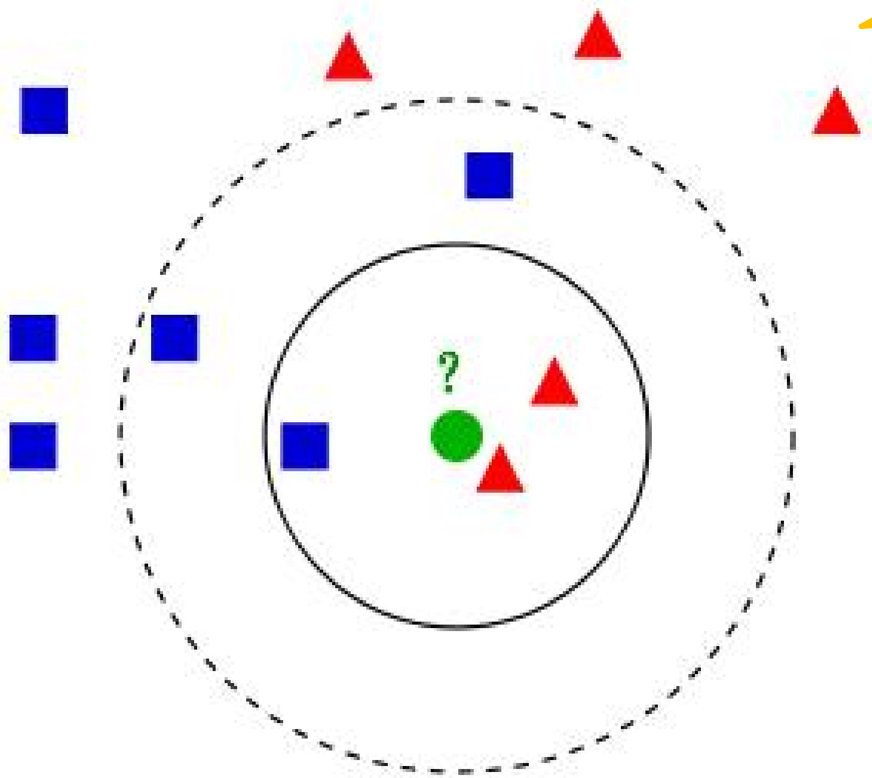
分类器常规训练和测试过程



如何分类

- 中心绿色点属于哪一类？

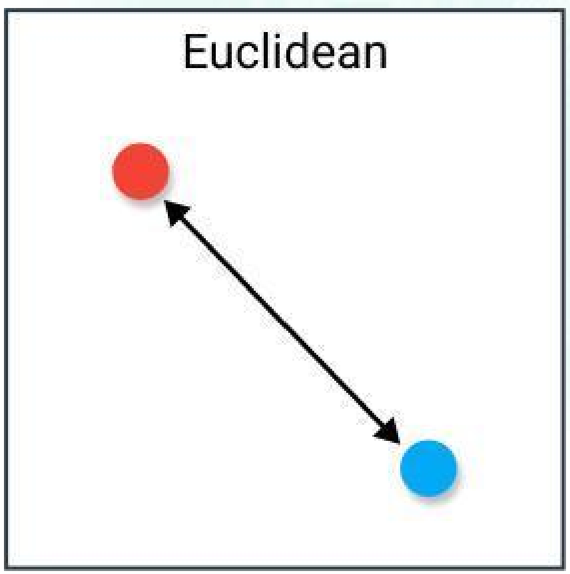
问题？



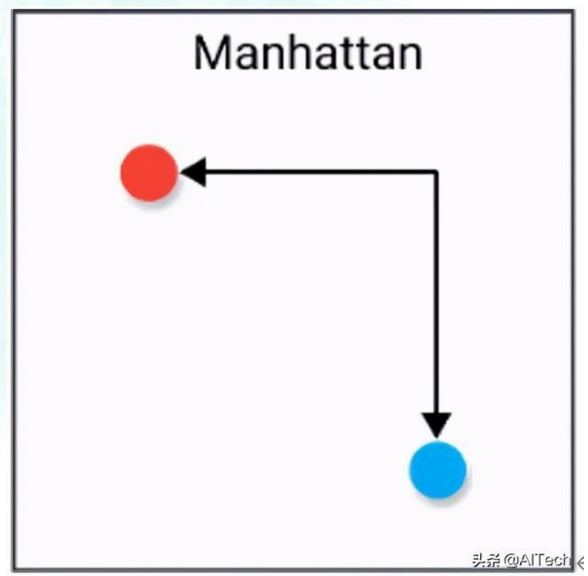
8.1.1 K-近邻分类器

- “物以类聚，人以群分”，K-近邻(K-Nearest Neighbor, KNN)学习就是利用这样一种思想发展起来的分类器。

- K-近邻学习是一种基于距离的分类器。



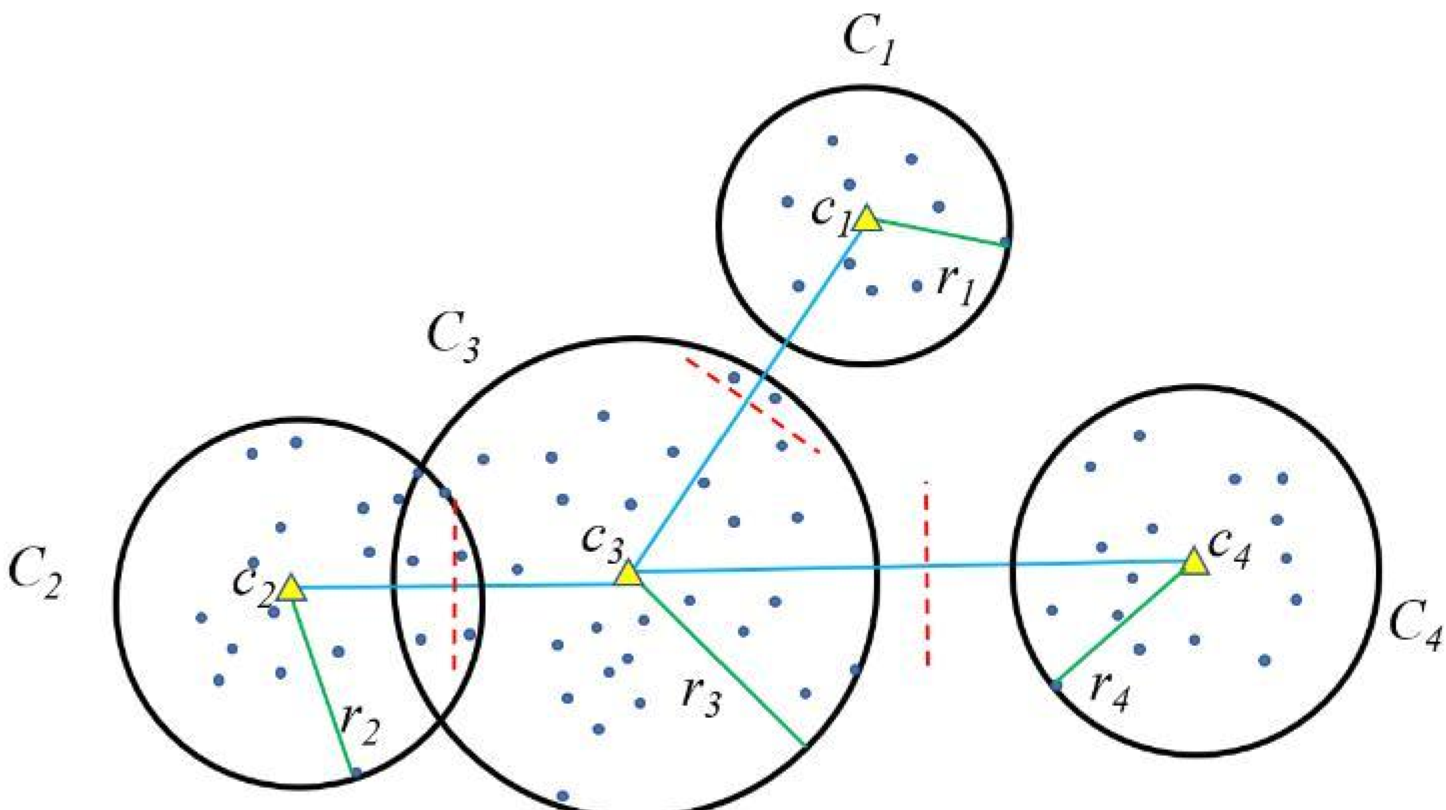
$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$



$$d=|x_1-x_2|+|y_1-y_2|$$

8.1.1 K-近邻分类器

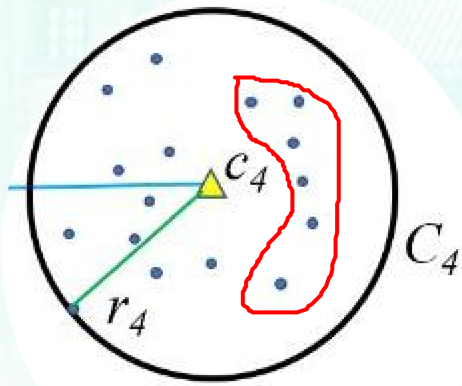
- K-近邻学习是一种常用的监督学习方法，其工作机制非常简单：给定测试样本，基于某种距离度量找出训练集中与其最靠近的K个训练样本，然后基于这K个邻居的信息来进行预测。



如何给出类标签

- 已知点C4的K的邻居和邻居的标签，
- 如何给出C4的标签？

问题？



8.1.1 K-近邻分类器

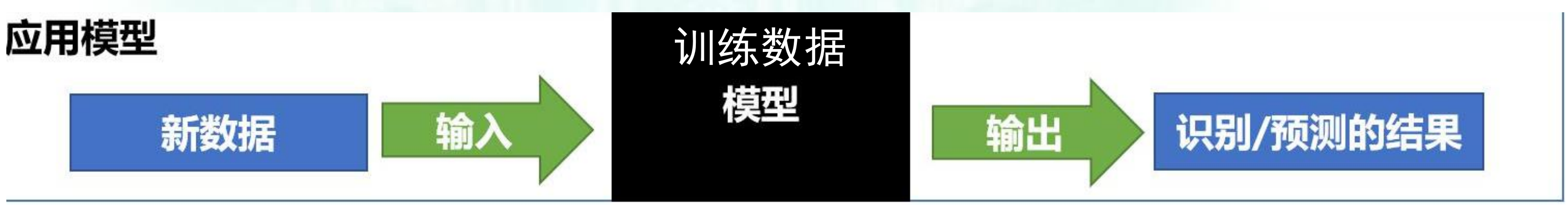
- 通常在分类任务中可使用“投票法”，即选择K个样本中出现最多的类别标记作为预测结果；

- 回归任务中可使用“平均法”，将这K个样本的实值输出标记的平均值作为预测结果；

- 还可基于距离远近进行加权平均或者加权投票。

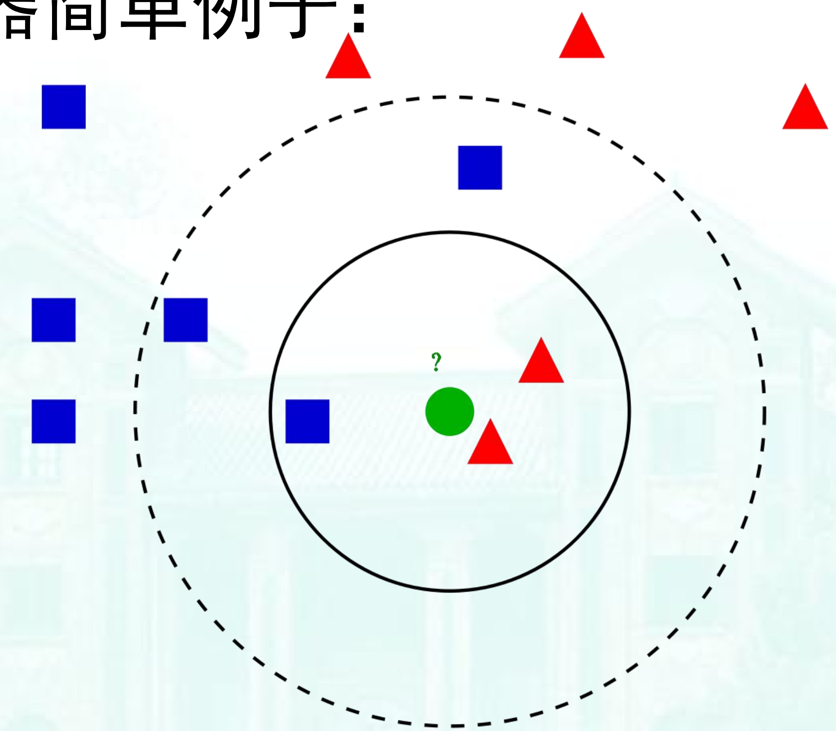
8.1.1 K-近邻分类器

- K-近邻学习没有显示的训练过程，是“懒惰学习”（lazy learning）的著名代表。此类学习技术在训练阶段仅仅是把样本保存起来，训练时间开销为零，收到测试样本时再进行处理。
  - **“急切学习”（eager learning），是在训练阶段就对样本进行学习处理的方法。**



8.1.1 K-近邻分类器—K的取值

- K-近邻分类器简单例子：

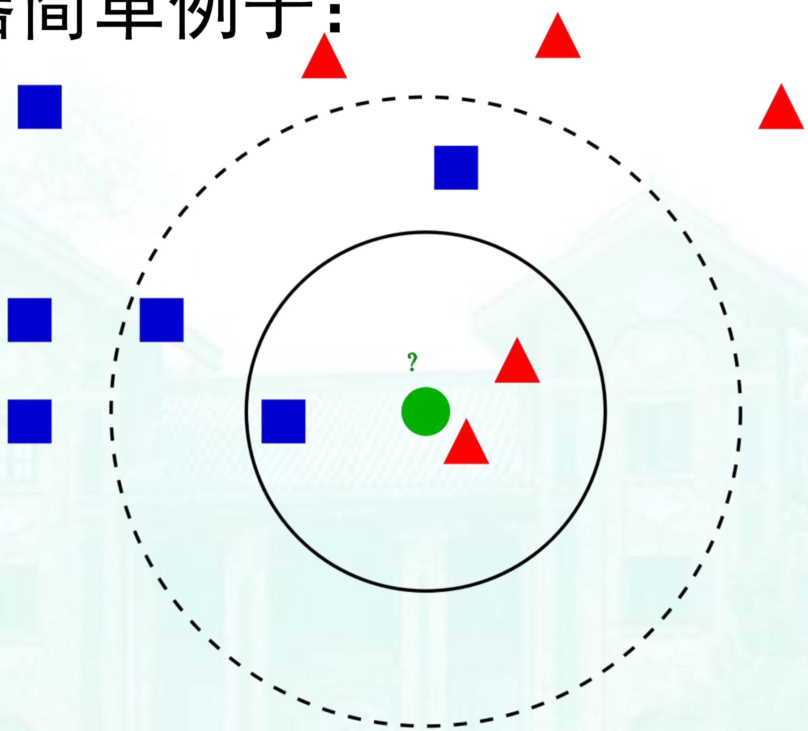


当k=3（等距线实线表示），有两个红色三角形和一个蓝色方形在实线内，被分类为红色三角形；

图源：Wikipedia，https://en.wikipedia.ahmu.cf/wiki/K-nearest_neighbors_algorithm

8.1.1 K-近邻分类器——K的取值

- K-近邻分类器简单例子：



当k=5（虚线表示），有两个红色三角形和三个蓝色方形在虚线内，被分类为蓝色方形。

图源：Wikipedia，https://en.wikipedia.ahmu.cf/wiki/K-nearest_neighbors_algorithm

8.1.1 K-近邻分类器

- K-近邻分类器在设计过程中，有四个要点需要注意：
  - 用于对待分类对象所属类别进行评估的数据集合是什么，不一定需要使用整个数据集；
  - 用于计算对象之间相似度距离或者相似度矩阵如何计算，可以使用马氏距离，欧氏距离等等；
  - K值如何选取；
  - 用于确定待分类对象所属类别的方法是什么。

8.1.1 K-近邻分类器

- K-近邻分类器算法总结：

Algorithm KNN

Input: D is the set of training objects; z is the test object, which is a vector of attribute values;
L is the set of classes used to label the objects.

Output: $c_z \in L$ is the class of z

foreach object $y \in D$ **do**

Compute $d(z, y)$, the distance between z and y;

end

Select $N \subseteq D$, the set (neighborhood) of k closest training objects for z;

$c_z = \underset{v \in L}{\operatorname{argmax}} \sum_{y \in N} I(v = class(c_y))$;

where $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

- 算法复杂度：$O(n)$

8.1.1 K-近邻分类器

- K-近邻分类器总结：
  - KNN算法简单且容易实现。当训练数据集很大时，需要大量的存储空间，而且需要计算待测样本和训练数据集中所有的样本距离，非常耗时。
  - KNN对于对于类内间距小、类间间距大的数据集分类效果好，对于边界不规则的数据集效果好于线性分类器。
  - KNN对于样本不均衡的数据效果不好，可以对k个邻居数据赋予权重。

8.1.1 K-近邻分类器

KNN很耗时，一般用于样本数少的数据集；当数据量大时，可以将数据以树的形式呈现。如KD树

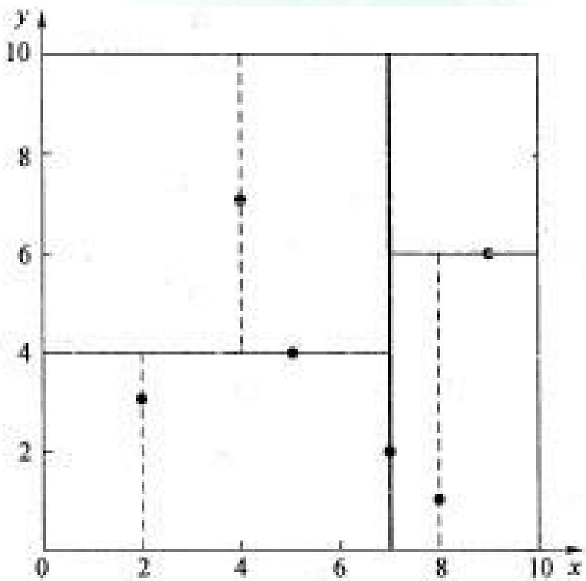


图1 二维数据k-d树空间划分示意图

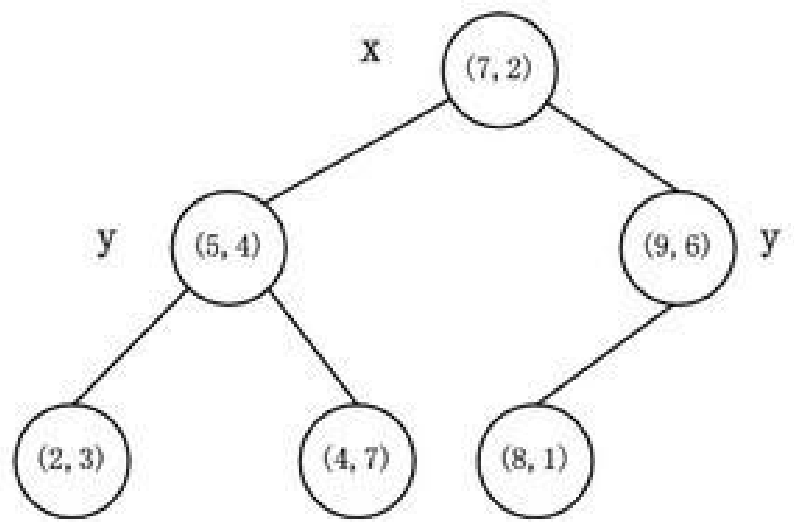
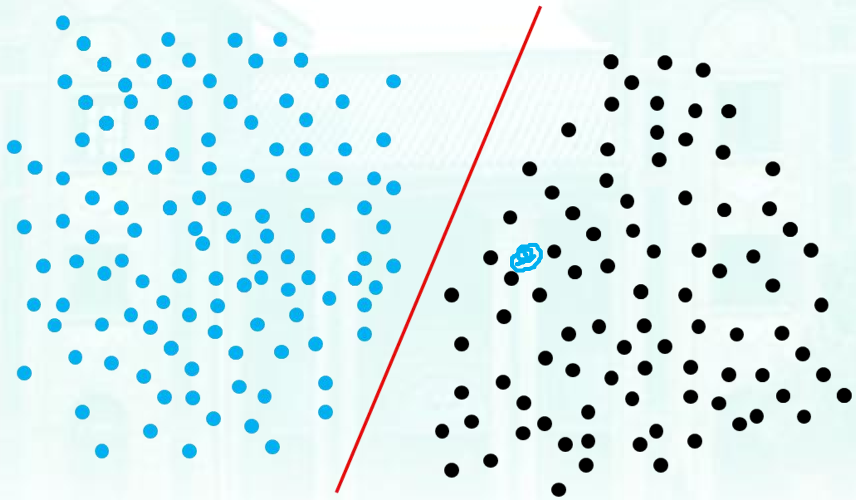


图3 上述实例生成的k-d树

8.1.2 支持向量机

- 线性可分：在二维空间上，两类点被一条直线完全分开叫做线性可分。

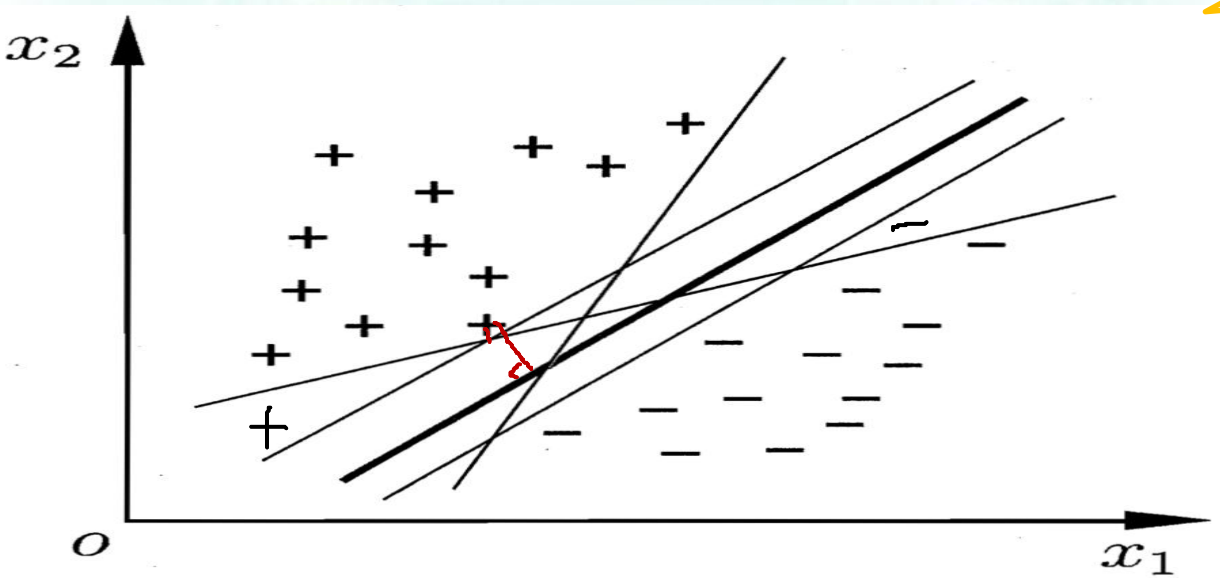


8.1.2 线性可分数学表示

- 严格数学定义：D_0和D_1是n维欧氏空间中的两个点集。如果存在n维向量$\boldsymbol{w}$和实数b，使得所有属于D_0的点x_i都有$wx_i + b > 0$, 而对于所有属于D_1的点x_j都有$wx_j + b < 0$，则称D_0和D_1线性可分。

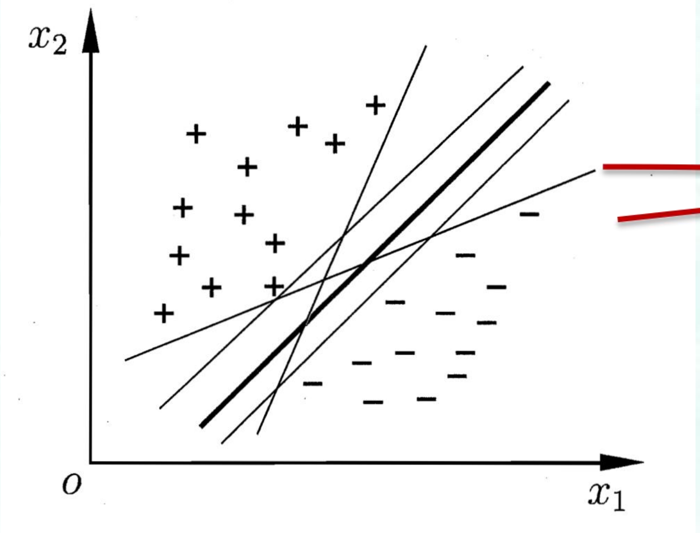
8.1.2 支持向量机

- 间隔超平面：从二维拓展到多维中，上述直线即拓展维一个超平面。
- 给定训练样本集$D=\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，其中$y_i\in\{-1,+1\}$，能把不同类别样本划分开的超平面不止一个：



8.1.2 支持向量机

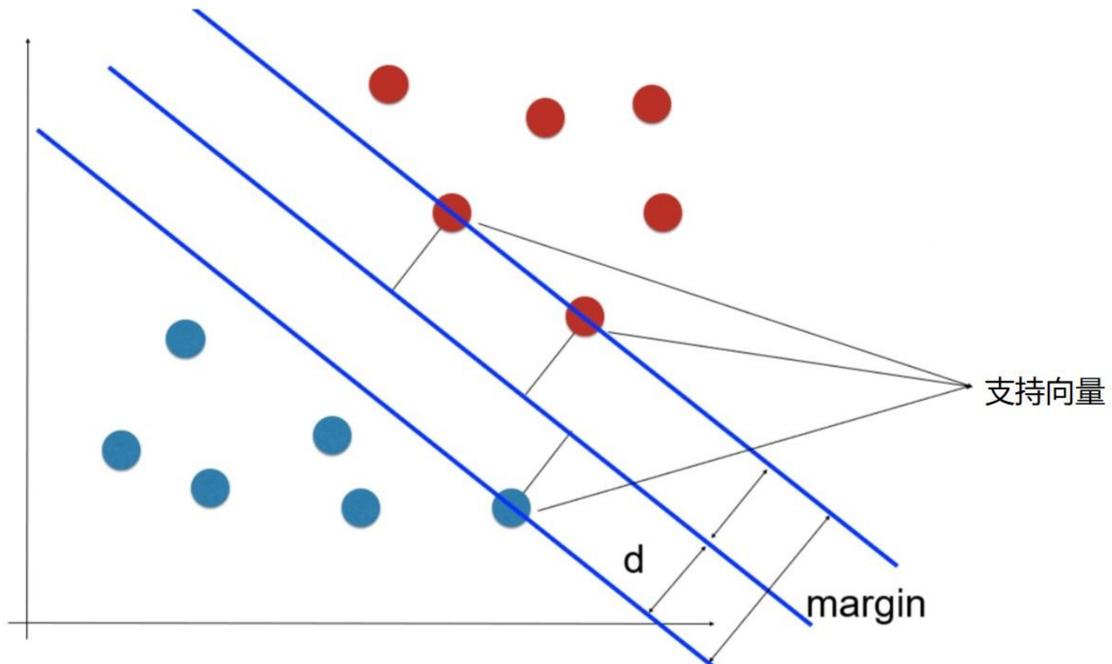
- 最大间隔超平面：两侧距离超平面最近的样本点到超平面的距离被最大化了。



多个划分超平面

8.1.2 支持向量机

- 支持向量(support vector)：样本中距离超平面最近的样本点。



- 间隔(margin)：两个异类支持向量到超平面的距离和。

8.1.2 支持向量机

- 支持向量机最优化问题：即找到最大间隔超平面。
  - 任意超平面用以下线性方程描述：

$$w^T x + b = 0$$

其中$w = (w_1; w_2; \ldots; w_d)$为法向量决定超平面的方向；$b$为位移项，决定超平面与原点之间的距离。

  - 样本空间中任意点x到超平面的距离可写为：

$$r = \frac{|w^T x + b|}{||w||}$$

  - 超平面需要将样本正确分类，满足以下公式：

$$\begin{cases} w^T x_i + b \geq +1, y_i = +1; \\ w^T x_i + b \leq -1, y_i = -1. \end{cases}$$

从公式看，支持向量就是使上述公式等号成立的几个样本点。

8.1.2 支持向量机

- 给定训练样本集$D=\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，其中$y_i\in\{-1,+1\}$，在这个前提下：
  - 间隔(margin)：两个异类支持向量到超平面的距离：

$$\gamma=\frac{2}{||w||}$$

  - 找到最大间隔超平面的约束条件，即让距离γ最大：

$$\max_{w,b}\frac{2}{||w||}$$
$$s.t.\ y_i(w^Tx_i+b)\geq 1, i=1,2,\dots,m.$$

  - 最大化间隔，即最大化$||w||^{-1}$，即最小化$||w||^2$：

$$\boxed{\begin{array}{c}\min_{w,b}\frac{1}{2}||w||^2\\ s.t.\ y_i(w^Tx_i+b)\geq 1, i=1,2,\dots,m.\end{array}}$$

这就是支持向量机(Support Vector Machine, SVM)的基本型。

8.1.2 支持向量机

- 求解SVM问题的基本思路：

- 不等式约束优化
- 引入松弛变量（也看作优化变量）
- 等式约束优化
- 引入Lagrange乘子（看作优化变量）
- 无约束优化
- 将Lagrange函数对所有优化变量求偏导，令其值为零

- 引入拉格朗日乘子α_i将SVM问题得到其对偶问题(dual problem)：

拉格朗日函数：$L(w,b,\alpha)=\frac{1}{2}||w||^2+\sum_{i=1}^{m}\alpha_i(1-y_i(w^Tx_i+b))$

对w和b求偏导并令值为0得：$w=\sum_{i=1}^{m}\alpha_i y_i x_i$； $\mathbf{0}=\sum_{i=1}^{m}\alpha_i y_i$

将偏导带入拉格朗日函数，将w和b消去，得到对偶问题：

$$\boxed{\max_{\alpha}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j\,;\, s.t.\sum_{i=1}^{m}\alpha_i y_i=\mathbf{0};\ \alpha_i\geq 0}$$

8.1.2 支持向量机

- 如何求解：

$$\max_{\alpha}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j\,;s.t.\sum_{i=1}^{m}\alpha_i y_i=\mathbf{0};\ \alpha_i\geq 0$$

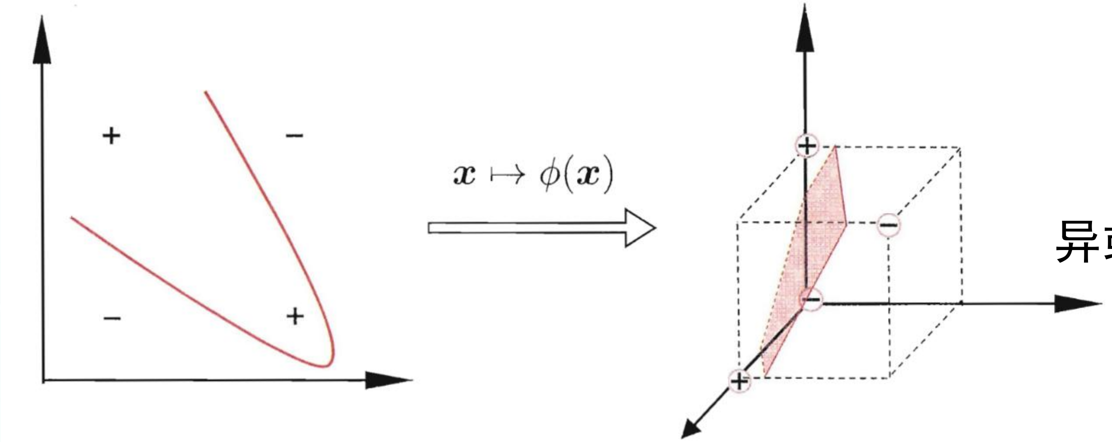
- 对于α_i和α_j来说，提出过许多高效算法，以顺序最小优化(Sequential Minimal Optimization, SMO)为例，不断执行如下两步直至收敛：
  - 选取一对需要更新得变量α_i和α_j；
  - 固定α_i和α_j以外的参数，求解上述公式获得更新后的α_i和α_j。
- 对于偏移量b，对于任意的支持向量(x_s, y_s)，都有$y_s f(x_s)=1$，即：

$y_s\left(\sum_{i\in S}\alpha_i y_i x_i^T x_s+b\right)=1$，其中$S$是支持向量的下标集。
  - 理论上，选择任意一个支持向量求解可以得到b；
  - 实际使用更鲁棒的方法：使用所有支持向量求解的平均值：

$$b=\frac{1}{|b|}\sum_{s\in S}\left(\frac{1}{y_s}-\sum_{i\in S}\alpha_i y_i x_i^T x_s\right)$$

8.1.2 支持向量机

- 若原始样本空间内不存在划分两类样本的超平面时，可以将原始空间映射到更高维的特征空间，使得其在更高维特征空间内线性可分：



异或问题的非线性映射

- 使用非线性映射后的$\phi(x)$代替x，之前最优化问题变为：

$$\boxed{\max_{\alpha}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\phi(x_i)^T\phi(x_j)\,;\,s.t.\sum_{i=1}^{m}\alpha_i y_i=\mathbf{0};\ \alpha_i\geq 0}$$

- 令：$\mathcal{K}(x_i,x_j)=\langle\phi(x_i),\phi(x_j)\rangle=\phi(x_i)^T\phi(x_j)$，即为“核函数”（Kernel function）。

8.1.2 支持向量机

- 常用核函数：

名称	表达式	参数
线性核	$\mathcal{K}(x_i, x_j) = x_i^T x_j$	
多项式核	$\mathcal{K}(x_i, x_j) = (x_i^T x_j)^d$	$d \geq 1$为多项式次数
高斯核	$\mathcal{K}(x_i, x_j) = \exp(-\frac{\|\|x_i - x_j\|\|^2}{2\sigma^2})$	$\sigma > 0$为高斯核的带宽
拉普拉斯核	$\mathcal{K}(x_i, x_j) = \exp(-\frac{\|\|x_i - x_j\|\|}{\sigma})$	$\sigma > 0$
Sigmoid 核	$\mathcal{K}(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$	$tanh$为双曲正切函数，$\beta > 0, \theta < 0$

8.1.2 支持向量机

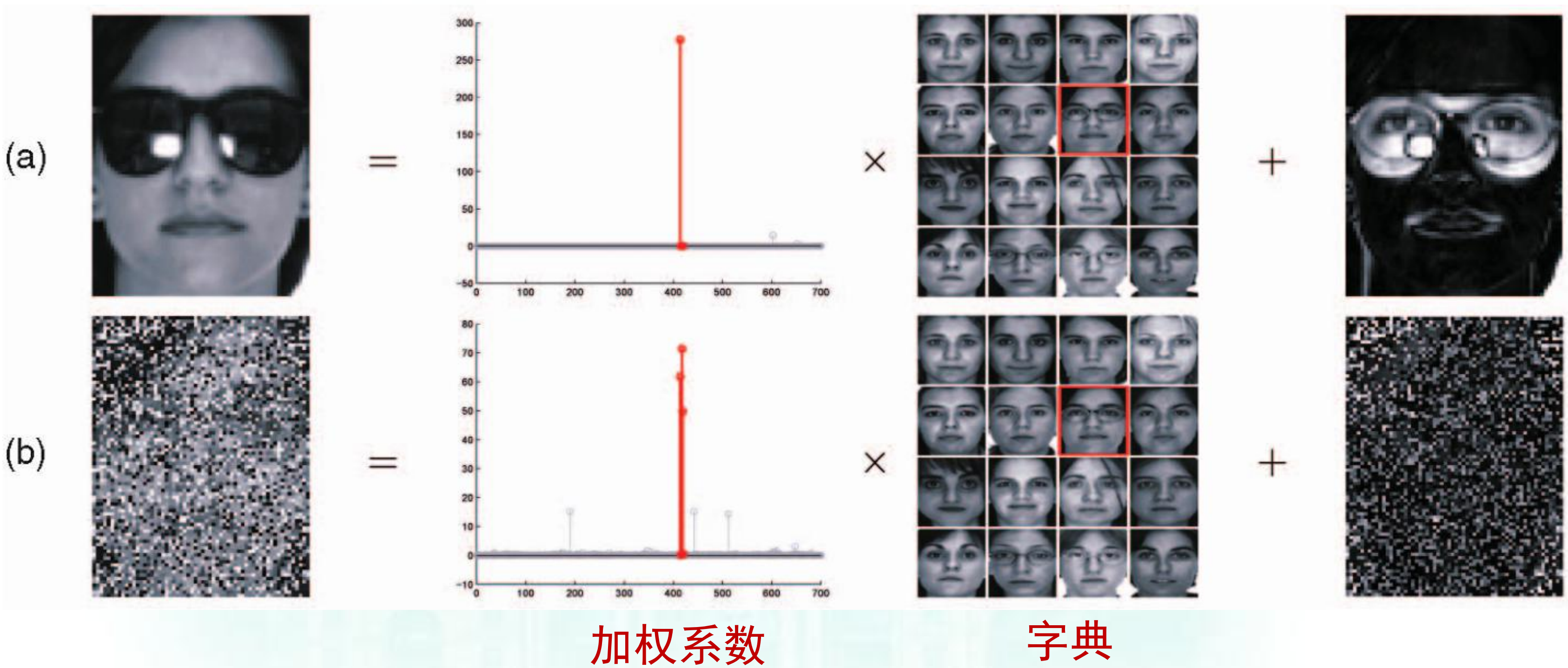
- 支持向量机总结：
  - 支持向量机的求解通常时借助凸优化技术。如何提高效率，使SVM能适应大规模数据是研究重点之一。
  - 支持向量机是针对二分类任务设计的，对多分类任务要进行专门的推广。
  - 核函数直接决定了支持向量机的最终性能，但核函数的选择是一个未决问题。

8.1.3 稀疏协同表示分类器

- 稀疏表示(Sparse Representation):
  - 将给定的数据集D用矩阵的形式表示，矩阵中存在许多零元素，具有稀疏性。
  - 如文档分类学习中，将《现代汉语字典》中的汉字视为数据集D，用矩阵表示；给定一个文档，字典中大多数字不出现在这个文档中，则矩阵具有大量零元素，即为样本具有稀疏性。
- 字典学习(Dictionary Learning):
  - 在一般的学习任务中，没有给定的《现代汉语字典》，需要为稠密表达的样本找到合适的字典，转化为稀疏表示形式，让模型复杂度得以降低，这类方法为字典学习。

8.1.3 稀疏协同表示分类器

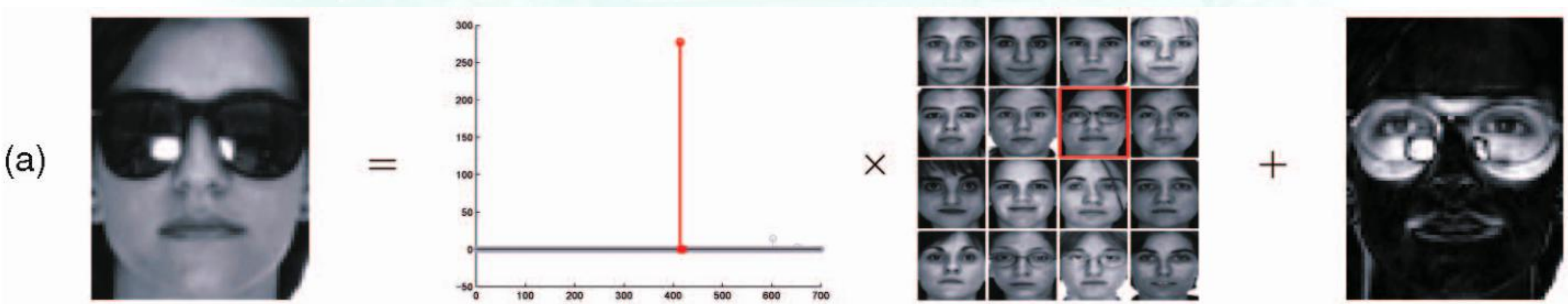
- 稀疏表示在人脸识别中的应用：



- 被遮挡(a)和损坏(b)图像，是所有训练图像的稀疏线性组合(中间)加上由于遮挡或损坏而引起的稀疏错误(右)。

稀疏表示如何分类？

问题？



加权系数　　字典

8.1.3 稀疏协同表示分类器

- 稀疏表示分类(Sparse Representation Classification, SRC)算法：从训练样本中抽取一定的样本构建词典，然后用词典表示测试样本，选取对应最小残差值的类作为预测结果。

8.1.3 稀疏协同表示分类器

- 若X_i表示第i类的训练样本，大小为$d \times l_i$， d为特征维数，l_i为样本数量。将所有类别的训练样本合成一个矩阵 $D = [X_1 \ldots X_i \ldots X_K]$，大小为$d \times l$，对于任意一个测试样本可以表示为如下线性方式：

$$y = D\alpha$$

由于l通常会远大于d，因此需要添加一个约束条件，得到α唯一解

$$y = D\alpha \quad \mathrm{s.t.} \min||\alpha||_1$$

求解最优系数α^*后，利用残差确定相应类别：

$$y^* = argmin_i e_i(y), e_i(y) = ||y - X_i\alpha_i||_2$$

8.1.3 稀疏协同表示分类器

- 给定数据集$\{x_1, x_2, \dots, x_m\}$，**字典学习最简单形式：**

$$\min_{\boldsymbol{B},\boldsymbol{\alpha}_i} \sum_{i=1}^{m} ||\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i||_2^2 + \lambda \sum_{i=1}^{m} ||\boldsymbol{\alpha}_i||_1$$

其中$\boldsymbol{B} \in \mathbb{R}^{d\times k}$为字典矩阵，$k$为字典词汇量，$\boldsymbol{\alpha}_i \in \mathbb{R}^k$是样本$\boldsymbol{x}_i$的稀疏表示。公式第一项让$\boldsymbol{\alpha}_i$能很好重构$\boldsymbol{x}_i$，第二项让$\boldsymbol{\alpha}_i$更加稀疏。

  - 求解第一步，固定字典$\boldsymbol{B}$，将公式按分量展开：

$$\min_{\alpha_i} ||x_i - B\alpha_i||_2^2 + \lambda||\alpha_i||_1$$

  - 求解第二步，以$\boldsymbol{\alpha}_i$的初值来更新字典$\boldsymbol{B}$：

$$\min_{B} ||X - BA||_F^2$$

8.1.3 稀疏协同表示分类器

- 协同表示分类(Collaborative Representation Classification, CRC)算法：当样本高度相关时，样本y在各个类的投影可能大致相同，SRC的结果就不稳定，于是CRC提出：

$$y = D\alpha \quad \mathrm{s.t.} \min||\alpha||_2$$

或者表示为：

$$\min||\alpha||_2 \ \ \mathrm{s.}t.\,||y - D\alpha||_2 \leq \varepsilon$$