图像处理与分析

— 无监督和监督学习

授课教师：孙剑

jiansun@mail.xjtu.edu.cn

http://jiansun.gr.xjtu.edu.cn

西安交通大学 数学与统计学院

目录

- 无监督学习算法
  - k-means算法
  - 层次化聚类算法
- 监督学习算法
  - K-NN分类算法
  - SVM分类算法

无监督学习和监督学习

- 监督学习：发现数据的属性与目标 / 类别之间的关联性。学习中有带标号的训练数据的指导

$$\underbrace{X=\{x_i,y_i\}}_{\text{已知带类标号的训练数据}} \rightarrow \underbrace{y=f(x)}_{\text{学习到的分类准则}}$$

- 无监督学习:发现数据的模式 / 类别信息，学习过程中没有带标号的训练数据.

例子：

- 监督学习：已知心脏病人和非心脏病人的属性数据（例如血压、年龄等）；学习到判断是否为心脏病人的准则。
- 无监督学习：已知一些测试人的属性数据（例如血压、年龄等），基于这些属性数据对人进行聚类

无监督学习—聚类

- 聚类问题：聚类是对数据集按照某种相似性规则划分为多个组，每个组称为一个类。
  - 同一类内的数据点具有强相似性，不同类的数据点的相似性较低。
- 聚类是最典型、最基本的无监督学习方法。
  - 在聚类中，不知道数据点的类别信息。
- 其他无监督学习问题：
  - 数据降维、可视化等

无监督学习—聚类

- 聚类的例子：下面数据集可划分为3类

无监督学习—聚类

聚类准则：相似类具有相似的特征

- 例子1: 身高、体型相似的人穿相同大小的T-shirts，因此T-shirts被划分为小号、中号和大号
  - 如果为每个人生产不同大小的T-shirts会造成成本的昂贵

- 例子2：在市场中，将相似的客户聚为同类，针对不同类客户制定广告、营销等策略

聚类是一种最常用的数据挖掘技术。在图像处理与分析中的图像分割、图像分类中具有广泛应用

无监督学习—聚类

聚类要素：

- 相似度（距离测度）
  - 针对具体问题设计，合适的数据间的相似度或距离
- 聚类算法
  - 划分式聚类（Partitional Clustering）
  - 层次化聚类（Hierarchical Clustering）
- 聚类质量测度
  - 类间距离极大化
  - 类间距离极小化

无监督学习—k-means聚类

- K-means算法是一种典型的划分式聚类（Partitional clustering）算法

- 设数据点集合为D:　{$\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$},

  每个数据点是一个向量$\mathbf{x}_i$ = (x_{i1}, x_{i2}, …, x_{ir})，其中每个维度对应于数据的一个属性.

- k-means聚类算法将数据点划分为k个类.
  - 每类有一个聚类中心，称为类中心（centroid）.
  - k由用户自己设定

无监督学习—k-means聚类

- k-means算法流程:

Step1: 随机选择*k*个数据点作为初始类中心；

Step2 : 迭代进行如下流程：

1) 基于数据点之间的聚类定义，将每个数据点赋予距离最近的类中心所在的类，

2) 重新计算类中心：即对属于不同类的数据求均值

3) 如果收敛准则不满足，则继续迭代；否则退出迭代，返回聚类结果（包括类中心和数据点的类标号）

无监督学习—k-means聚类

- 数据点之间的距离测度定义

L2距离(欧式距离)：在k-means算法中最常用的距离定义

$$dist(x,y)=\sqrt{\sum_{j=1}^{k}(x_i-y_i)^2}$$

L1距离：

$$dist(x,y)=\sum_{j=1}^{k}|x_i-y_i|$$

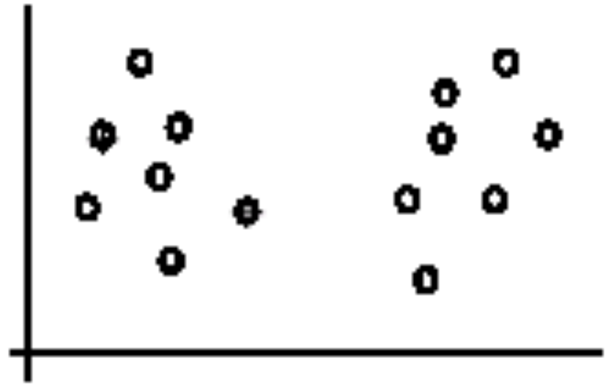
无监督学习—k-means聚类

不同的迭代收敛准则：

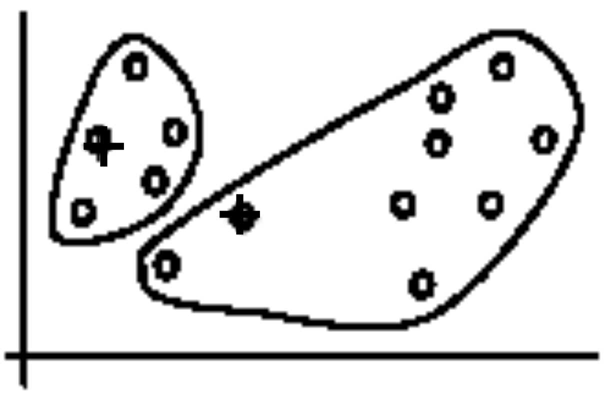
1. 数据点的类标号不在改变，即数据点的聚类结果稳定
2. 类中心没有显著变化（例如相邻两次迭代的类中心差异小于设定的阈值）
3. SSE误差（sum of squared error（SSE））无显著变化

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

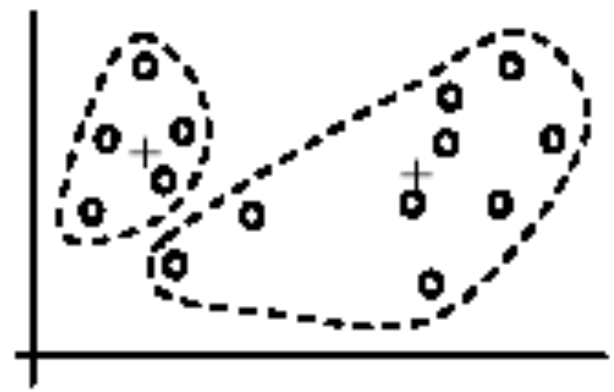
k-means聚类实例



(A). Random selection of *k* centers

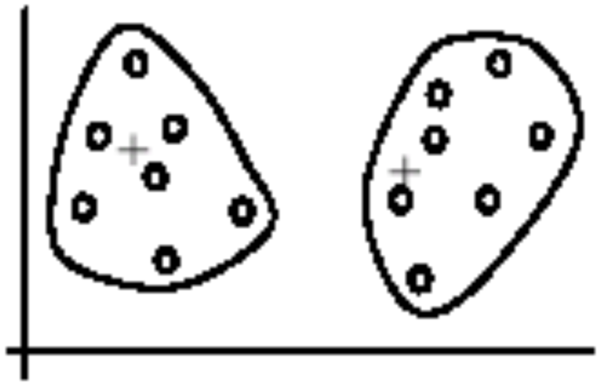


Iteration 1: (B). Cluster assignment

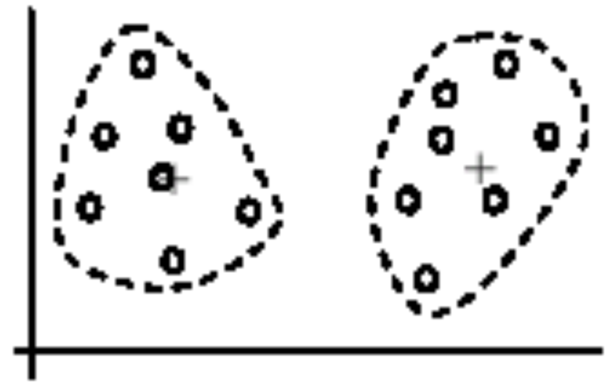


(C). Re-compute centroids

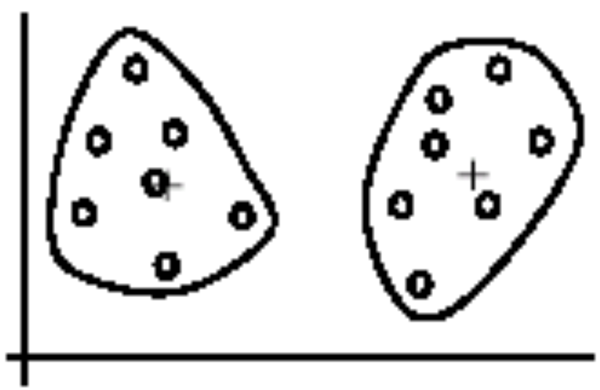
k-means聚类实例



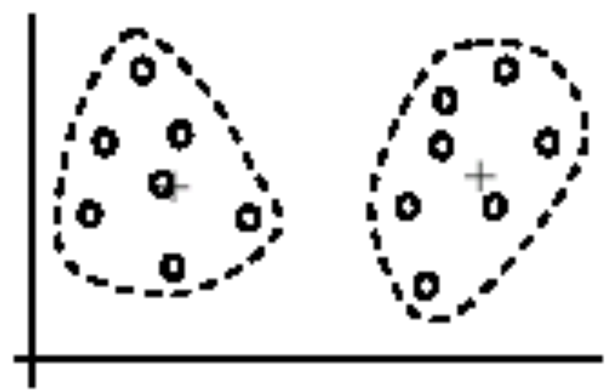
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

无监督学习—k-means聚类

- *k*-means算法的优点:
  - 简单:易于理解和实现
  - 高效：时间复杂度是$O(tkn)$,

    n：*样本点数目*;

    k：*聚类个数*;

    t：*迭代次数*
- K-means是最简单、应用最多的聚类算法之一

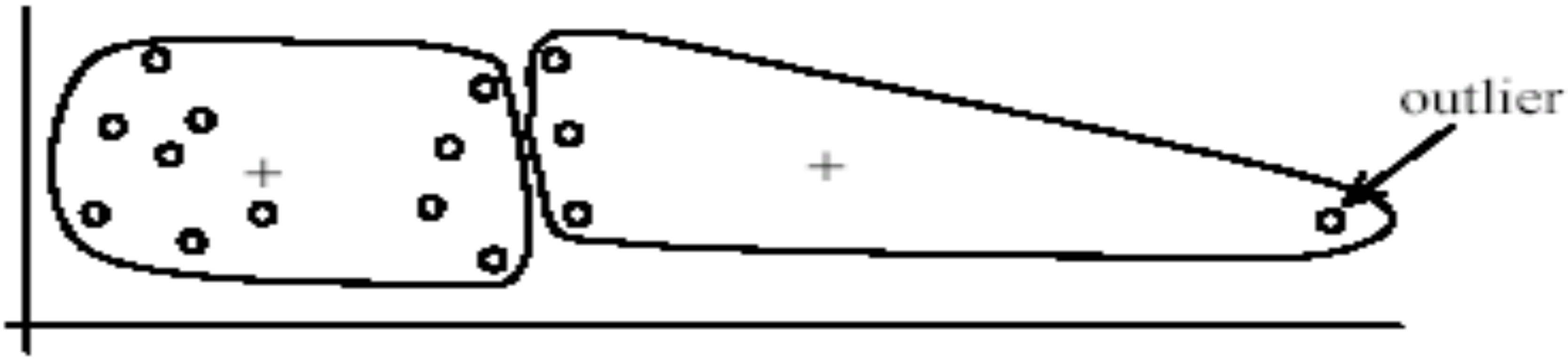
K-means算法非常适合应用于大规模数据集的聚类问题，在大数据计算平台Hadoop上已经实现了该算法的并行版本。

无监督学习—k-means聚类

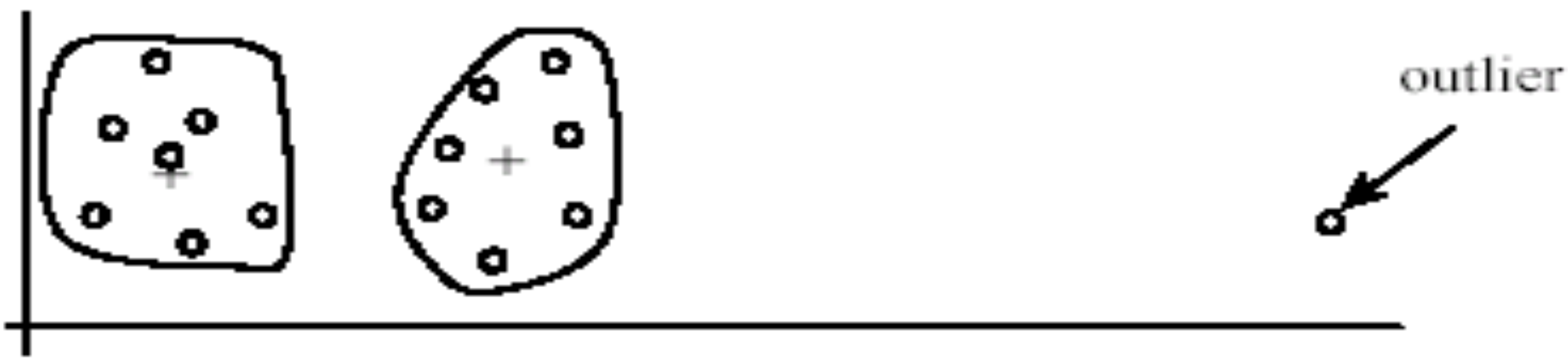
k-means算法的缺点:

- 该算法需要预先设定k.
- 该算法适合应用于同类数据为球形分布的点集合的聚类问题
- 算法对异常点（outliers）不鲁棒
  - 异常点是一些与其他点距离都非常远的孤立点.
  - 异常点产生可能是由于数据采集的误差引起的
  - 如何处理异常点：进行异常点检测

无监督学习—k-means聚类



(A): Undesirable clusters

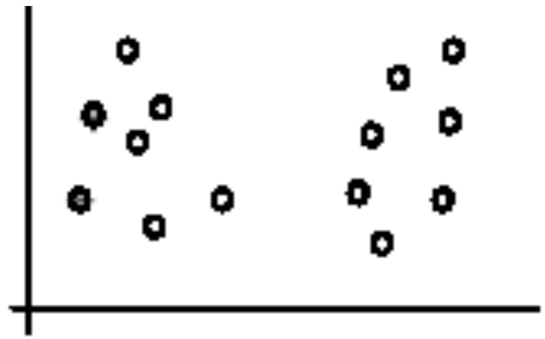


(B): Ideal clusters

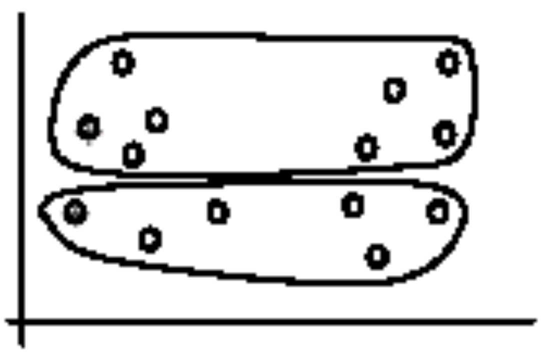
无监督学习—k-means聚类

- K-means算法对初始化中心点比较敏感，不同的初始化会导致不同的聚类结果。

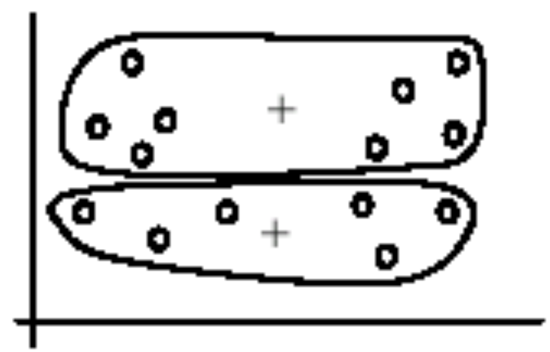
一种初始化产生的聚类结果



(A). Random selection of seeds (centroids)



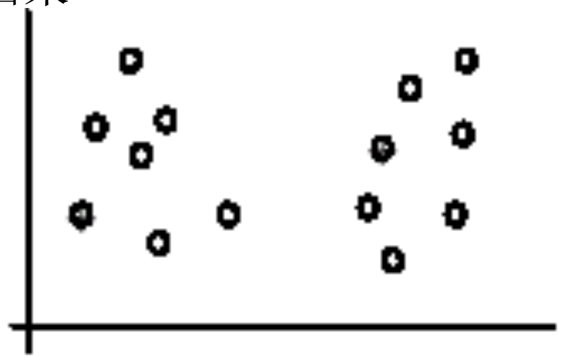
(B). Iteration 1



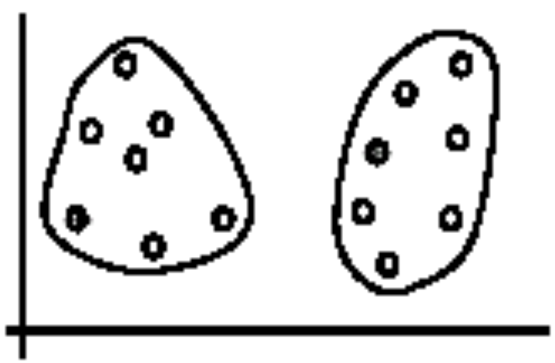
(C). Iteration 2

无监督学习—k-means聚类

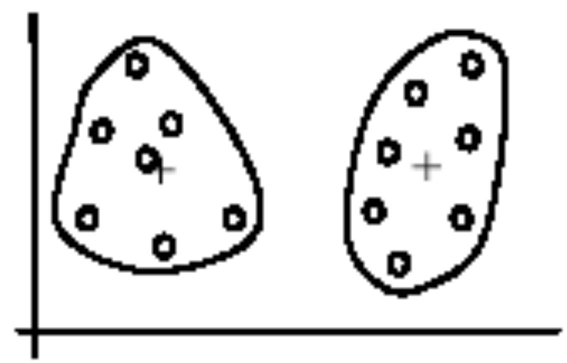
另一种初始化产生的聚类结果



(A). Random selection of *k* seeds (centroids)



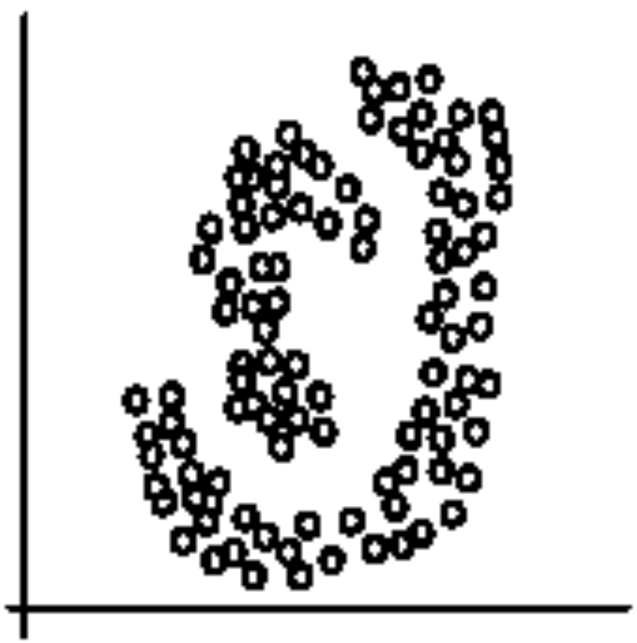
(B). Iteration 1



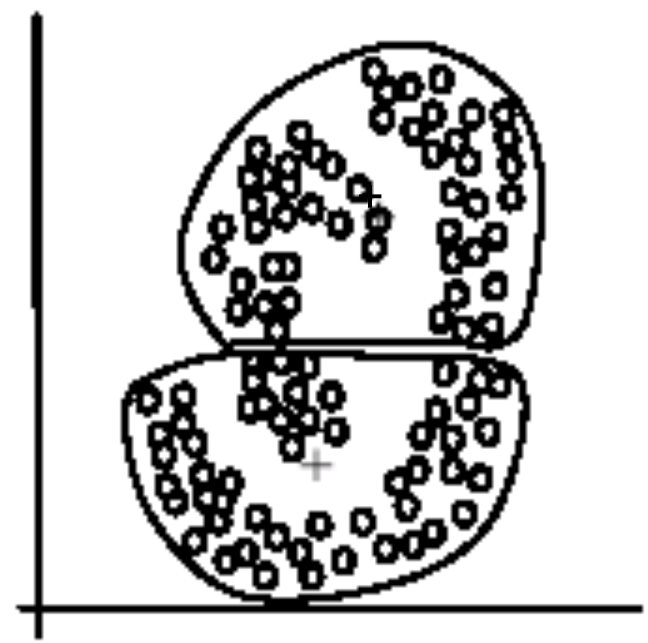
(C). Iteration 2

无监督学习—k-means聚类

- K-means算法不适合于聚类非球形（或类球形分布）的点集的聚类问题



(A): Two natural clusters



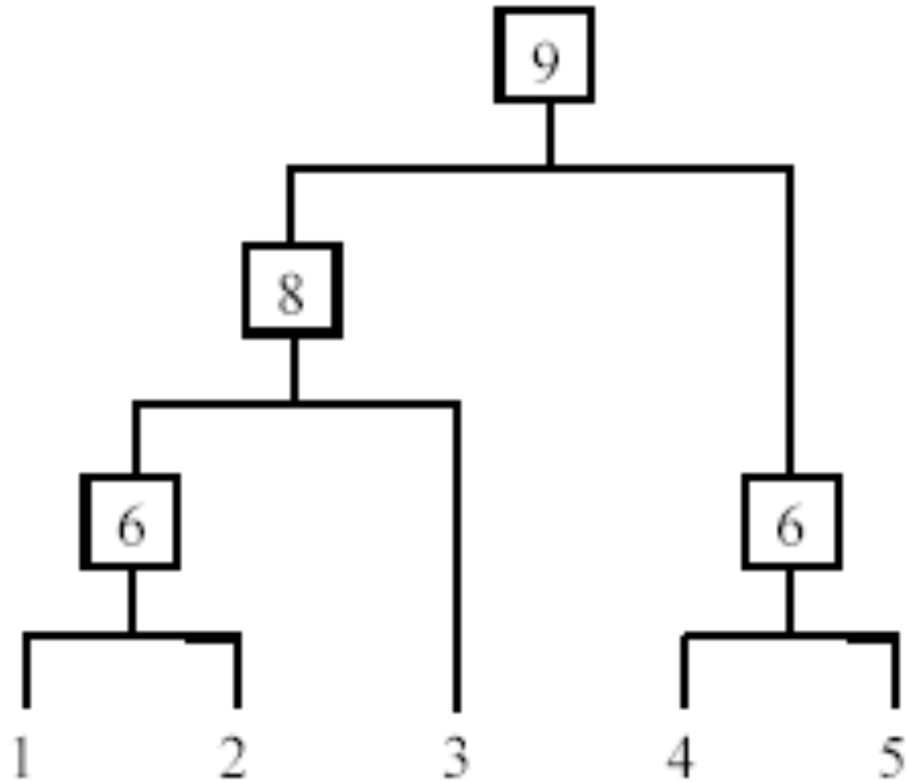
(B): *k*-means clusters

目录

- 无监督学习算法
  - k-means算法
  - 层次化聚类算法
- 监督学习算法
  - K-NN分类算法
  - SVM分类算法

无监督学习—层次化聚类

- 将输入数据集合划分为层次化的聚类结果。例如，这些聚类结果构成一个树结构，树中的每个节点对应于一类。



无监督学习—层次化聚类

- 聚合式层次化聚类：自下而上的层次化聚类

  初始化：将每个点认为是一类，然后迭代执行聚合操作：
  - 将最相近的两类聚合到一起
  - 迭代停止：直到所有的数据点都聚成一类
- 分解式层次化聚类：自上而下的层次化聚类

  初始化：将所有点认为是同一类，然后迭代执行分解操作：
  - 将每个类分解为两类
  - 迭代停止：所有点都被分为单独的一类

聚合式层次化聚类算法比分解式层次化聚类算法应用更广.

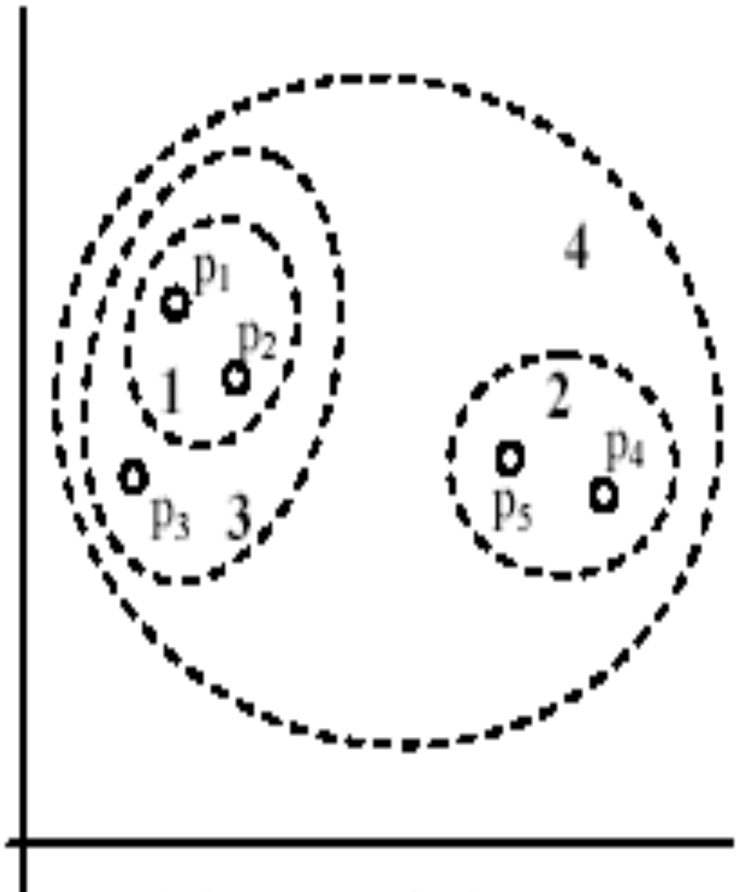
无监督学习—聚合式层次化聚类

聚合式层次化聚类算法描述：

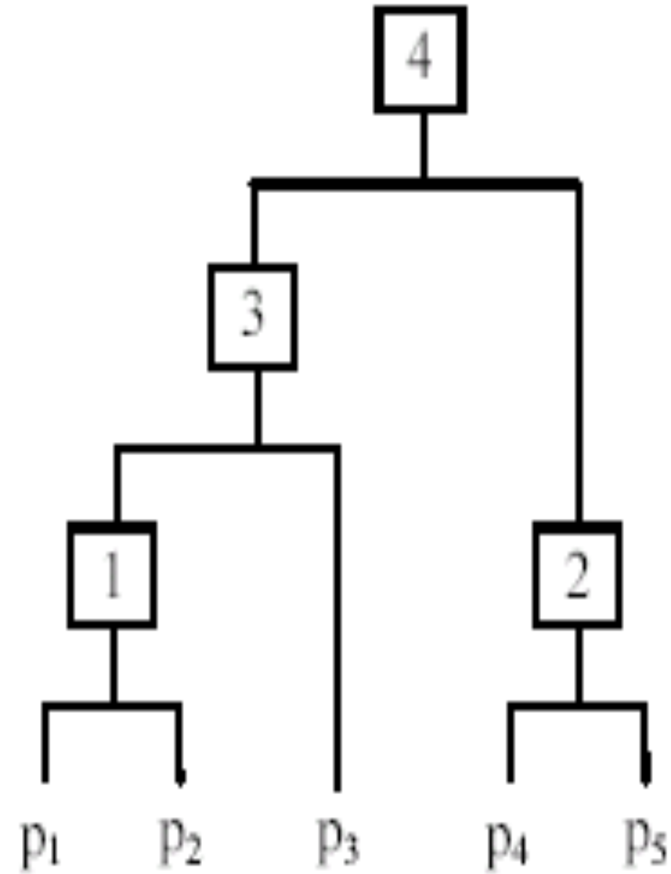
Algorithm Agglomerative(D)

```
Make each data point in the data set D a cluster,
Compute all pair-wise distances of x_1, x_2, ..., x_n ∈ D;
repeat
    find two clusters that are nearest to each other;
    merge the two clusters form a new cluster c;
    compute the distance from c to all other clusters;
until there is only one cluster left
```

无监督学习—聚合式层次化聚类



(A). Nested clusters



(B) Dendrogram

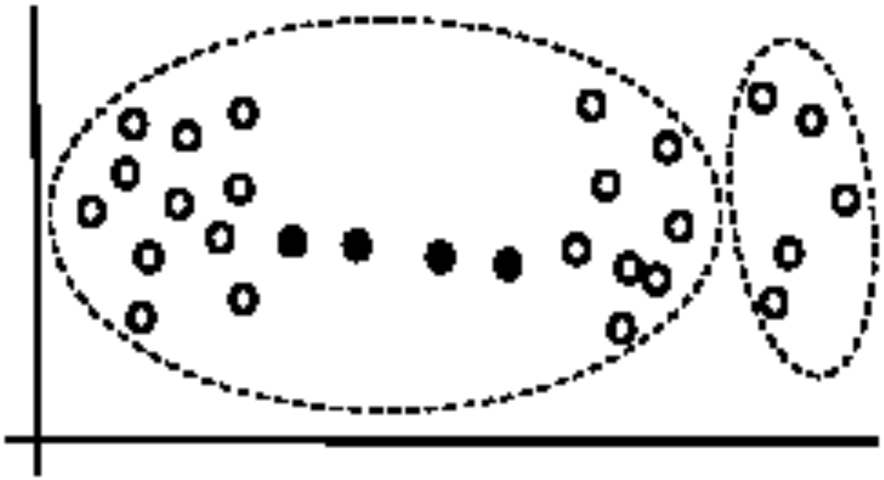
无监督学习—聚合式层次化聚类

- 如何测度类之间的距离？
  - Single link
  - Complete link
  - Average link
  - Centroids
  - …

无监督学习—聚合式层次化聚类

Single link方法：

- 两类距离用两类之间的最近点距离表示：



$$dist(c_1, c_2) = \min_{(x,y):x \in c_1, y \in c_2} \| x - y \|_2$$

优点：可以找到任意形状的类，但对噪声点不鲁棒

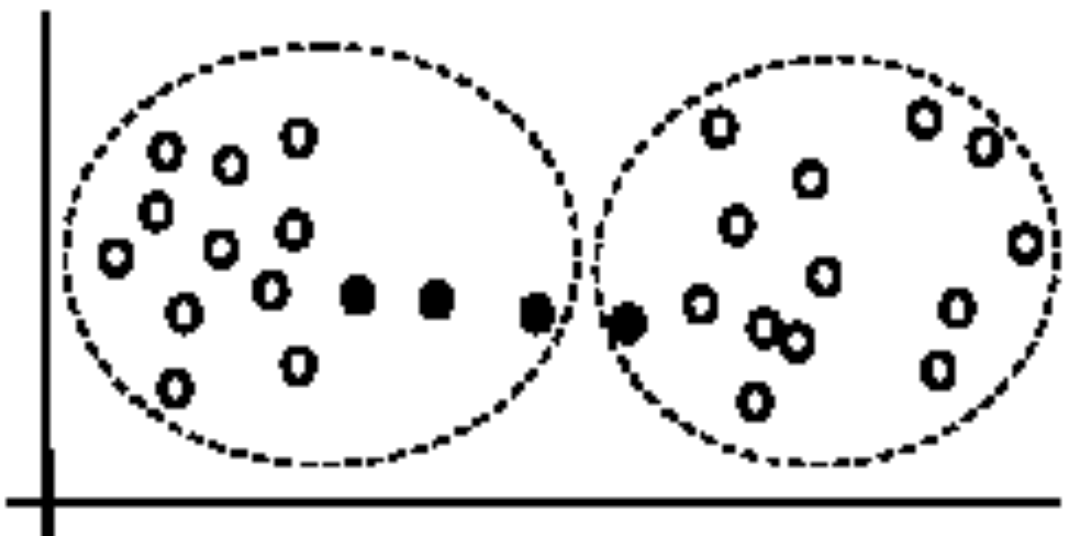
无监督学习—聚合式层次化聚类

Complete link方法：

- 两类之间的距离用两类中最远点的距离进行测度.

$$dist(c_1, c_2) = \max_{(x,y):x\in c_1, y\in c_2} \| x - y \|_2$$

  - 缺点:对异常点比较不鲁棒



无监督学习—聚合式层次化聚类

- Average link方法:两类之间的距离用两类中的所有点对距离的平均值进行测度

$$dist(c_1,c_2)=\frac{1}{N}\sum_{(x,y):x\in c_1,y\in c_2}\|x-y\|_2$$

N为点对的个数

- Centroid 方法:两类之间的距离用两类的中心之间的距离来测度

目录

- 无监督学习算法
  - k-means算法
  - 层次化聚类算法
- 监督学习算法
  - K-NN分类算法
  - SVM分类算法

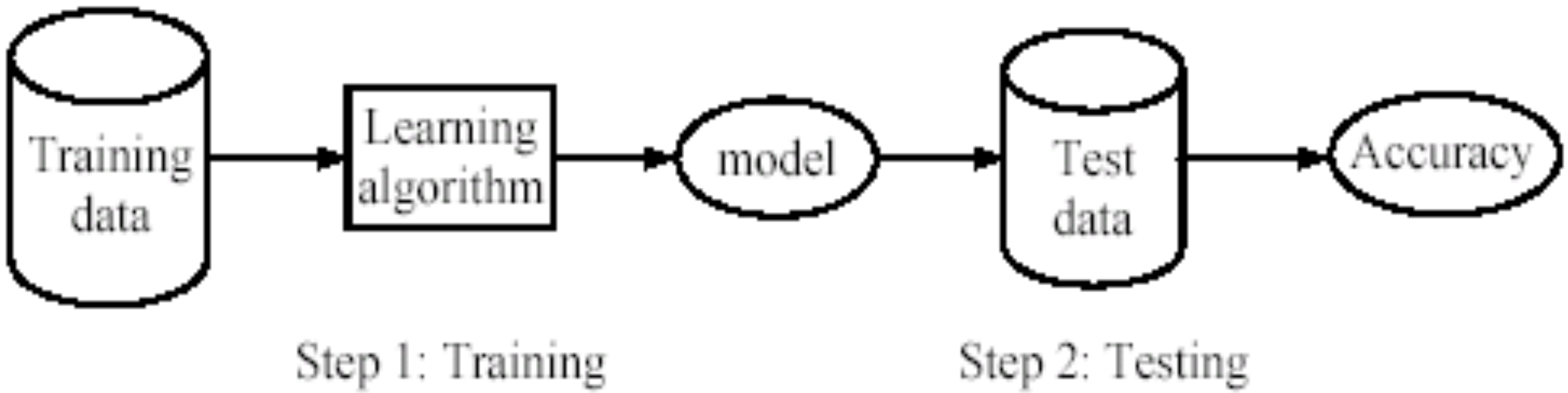
有监督学习

- 训练数据集: 由训练数据点构成的集合，每个数据点的构成为：
  - 数据属性向量：$\mathrm{x} = (x_1, x_2, \cdots, x_k)$ ，对应于*k*个不同的属性: A_1, A_2, … A_k.
  - 类别信息: $y \in \{c_1, c_2, \cdots, c_N\}$，对应于数据所属的类别

- 训练目标：从训练数据中学习到一个分类模型，用于预测新数据的类别信息

训练数据集合：$T = \{\mathrm{x}_i, y_i\}$

有监督学习



训练流程

测试精度的计算：

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$

有监督学习算法—kNN

k-近邻分类算法(k-Nearest Neighbor)

- 给定测试样本a, 在训练样本集合T中找到测试样本的k个最近邻样本集合P
- 计算P中c_j类样本的个数n_j
- 估计测试样本a属于c_j 类的可能性：$P(c_j|d) = n_j/k$

测试样本a的类标号估计为：

$$class(a) = \arg\min_{c \in \{c_1, \cdots, c_k\}} P(c \mid d)$$

监督学习方法—kNN

Algorithm kNN(D, d, k)

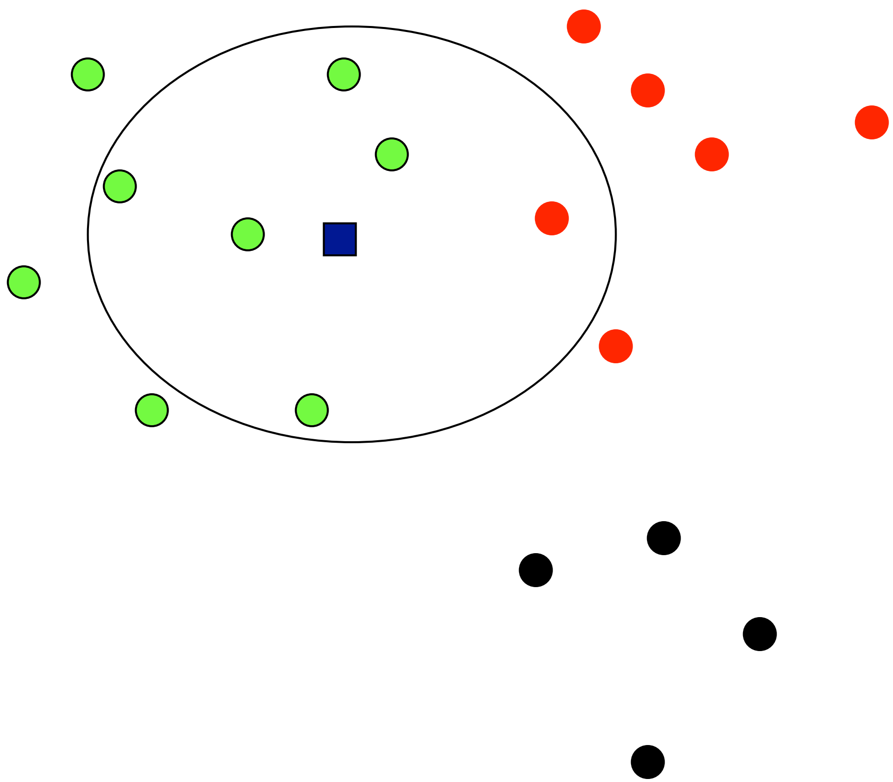
1 Compute the distance between d and every example in D;

2 Choose the k examples in D that are nearest to d, denote the set by P ($\subseteq D$);

3 Assign d the class that is the most frequent class in P (or the majority class);

- *k*的选择通常采用交叉验证的方法.
- 聚类函数的选择非常重要，依赖于具体问题

监督学习方法—kNN



- Government
- Science
- Arts

A new point ■
Pr(science|■)?

监督学习方法—kNN

kNN算法总结：

- kNN可以处理复杂的分类边界.
- 尽管kNN算法很简单，但是对于特定问题，该算法的精度与更好的分类算法的精度可比.
- kNN算法的测试过程比较慢，因为需要在训练数据中找到测试数据的k个最近邻点

有监督学习算法—SVM

- 支撑向量机（SVM: Support vector machines ）是最有效的有监督学习算法之一
  - 最早由俄国科学家V. Vapnik和合作者在1970s提出
  - SVM算法具有很好的理论基础，在不同具体应用中具有很高的分类精度
- 支撑向量机的基本思想: 寻找两类数据点之间的分割平面

有监督学习算法—SVM

- 训练数据集：D

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_r, y_r)\},$$

其中$\mathbf{x}_i = (x_1, x_2, \cdots, x_n)$ 是数据向量；y_i 是类标号

$$y_i = 1 \text{ 或 } -1.$$

1：表示正类数据； -1：表示负类数据.

- SVM的目标是找到一个分类面：

$$f(\mathrm{x}_i) = w^T \mathrm{x}_i + b$$

$$y_i = \begin{cases} 1 & if \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 0 \\ -1 & if \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$

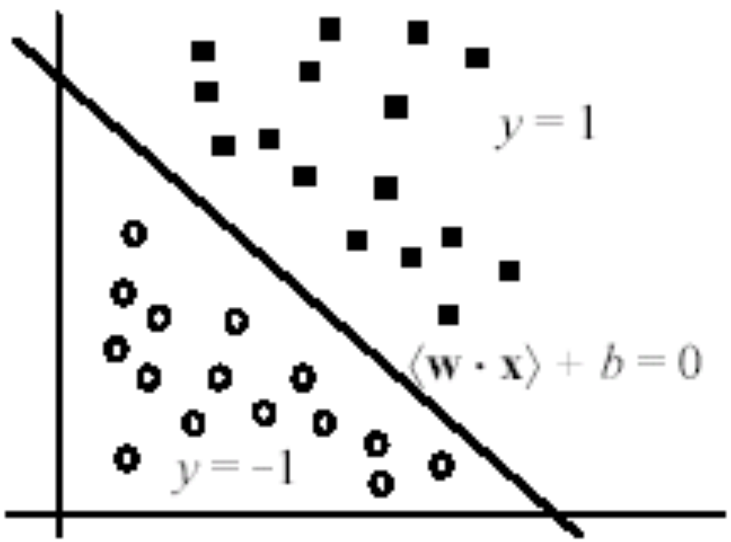
有监督学习算法—线性SVM

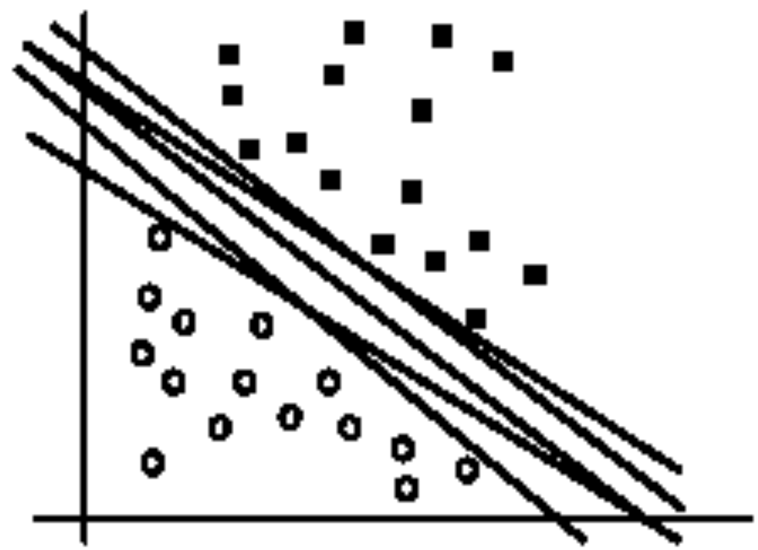
- 正、负类样本之间的分类面：

$$f(\mathrm{x}_i) = w^T \mathrm{x}_i + b = 0$$

也称为决策面。

- 在两类样本之间有很多决策面，如何选择最优的决策面?

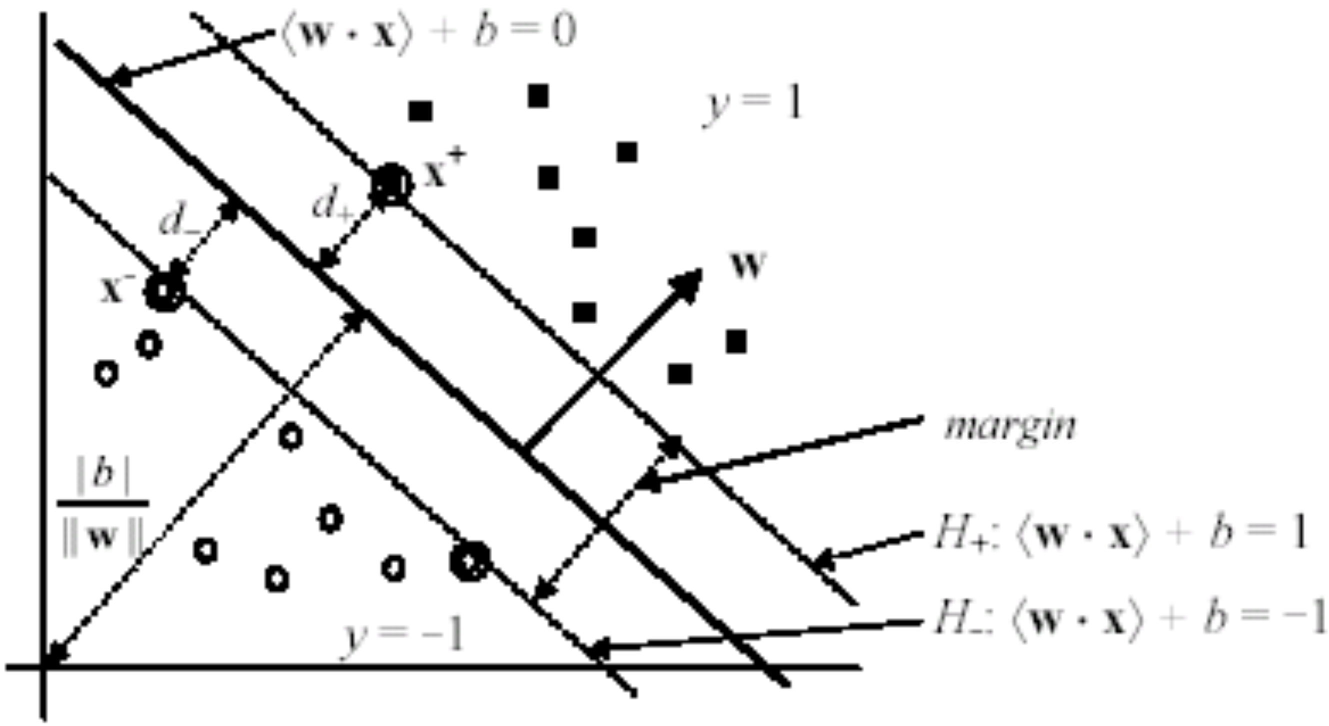




有监督学习算法—线性SVM

- SVM算法寻找具有最大边界(max-margin)的分类面

Maximal margin hyperplane



有监督学习算法—线性SVM

- 假定训练数据是线性可分的
- 考虑最接近分类面的正数据点（$\mathbf{x}^+$，1）和负类数据点（$\mathbf{x}^-$，-1）。分类面为：

$$f(\mathrm{x}_i)=w^T\mathrm{x}_i+b=0$$

- 定义两个平行的超平面：H_+ 和 H_-，分别通过$\mathbf{x}^+$ 和 $\mathbf{x}^-$. H_+ 和H_- 均平行于分类面：$f(\mathrm{x}_i)=w^T\mathrm{x}_i+b=0$

$$H_+: \quad \langle \mathbf{w}\cdot\mathbf{x}^+\rangle+b=1$$

$$H_-: \quad \langle \mathbf{w}\cdot\mathbf{x}^-\rangle+b=-1$$

$$\text{such that} \quad \begin{array}{ll} \langle \mathbf{w}\cdot\mathbf{x}_i\rangle+b\geq 1 & \text{if } y_i=1 \\ \langle \mathbf{w}\cdot\mathbf{x}_i\rangle+b\leq -1 & \text{if } y_i=-1, \end{array}$$

有监督学习算法—线性SVM

- 边际(margin)： 超平面H_+ 和 H_-之间距离。($d_+ + d_-$).
- 点$\mathbf{x}_i$ 到平面 $f(\mathrm{x}_i) = w^T \mathrm{x}_i + b = 0$的垂直距离为：

$$\frac{|\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b|}{\|\mathbf{w}\|}$$

其中||w||是w的范数：

$$\|\mathbf{w}\| = \sqrt{<\mathbf{w} \cdot \mathbf{w}>} = \sqrt{w_1{}^2 + w_2{}^2 + \ldots + w_n{}^2}$$

有监督学习算法—线性SVM

- **假设$\mathbf{x}_s$**是分类面上的一个点，满足：

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b = 0$$

则$\mathbf{x}_s$到 $H_+ : \langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 1$ 的距离是

$$d_+ = \frac{|\langle \mathbf{w} \cdot \mathbf{x_s} \rangle + b - 1|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

$$margin = d_+ + d_- = \frac{2}{\|\mathbf{w}\|}$$

有监督学习算法—线性SVM

给定训练数据：$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_r, y_r)\}$

线性SVM学习极大边际(max-margin)分类面，即优化如下问题：

$$\text{Minimize}: \quad \frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$$

$$\text{Subject to}: \quad y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, 2, \ldots, r$$

有监督学习算法—线性SVM

优化算法：

- 约束优化问题：

$$L_P = \frac{1}{2}\langle \mathbf{w}\cdot\mathbf{w}\rangle - \sum_{i=1}^{r}\alpha_i[y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) - 1]$$

α_i　是Lagrange乘子.

- 上述约束优化问题必须满足Kuhn-Tucker条件

（必要条件但不是充分条件）

有监督学习算法—线性SVM

Kuhn-Tucker条件

$$\frac{\partial L_P}{\partial w_j} = w_j - \sum_{i=1}^{r} y_i \alpha_i \mathbf{x}_i = 0, \quad j = 1, 2, \ldots, m \tag{48}$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{r} y_i \alpha_i = 0 \tag{49}$$

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \geq 0, \quad i = 1, 2, \ldots, r \tag{50}$$

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, r \tag{51}$$

$$\alpha_i(y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1) = 0, \quad i = 1, 2, \ldots, r \tag{52}$$

- 支撑向量（support vectors）：对应的乘子 α_i 非零的数据向量称为支撑向量

有监督学习算法—线性SVM

优化问题的探讨：

- 一般来说Kuhn-Tucker条件是最优解的必要条件，不是充分条件.
- 针对上述凸目标函数、线性约束优化问题，Kuhn-Tucker条件是最优解的充分必要条件.
- Lagrangian方法可以导出原优化问题的对偶优化问题，优化对偶问题相对于原问题更容易优化求解。

有监督学习算法—线性SVM

对偶问题

- 如何导出对偶问题：将Lagrangian乘子目标函数关于原变量（i.e.，**w**和b）的导数设为0,求出**w**和b，并反带入 Lagrangian乘子目标函数中，得到关于Lagrangian乘子的优化目标：

$$L_D = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle,$$

有监督学习算法—线性SVM

对偶问题

$$\text{Maximize: } L_D = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} y_i y_i \alpha_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle. \tag{56}$$

$$\text{Subject to: } \sum_{i=1}^{r} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, r.$$

有监督学习算法—线性SVM

对偶问题的分类面：

- **分类面：** $\mathbf{w}=\sum_i \alpha_i \mathbf{x}_i$

$$\langle \mathbf{w}\cdot\mathbf{x}\rangle + b = \sum_{i\in sv} y_i\alpha_i\langle \mathbf{x}_i\cdot\mathbf{x}\rangle + b = 0$$

- **给定测试样本z,**

$$sign(\langle \mathbf{w}\cdot\mathbf{z}\rangle + b) = sign\left(\sum_{i\in sv}\alpha_i y_i\langle \mathbf{x}_i\cdot\mathbf{z}\rangle + b\right)$$

- **如果**值为1, 则样本**z**分为正类; 否则, 样本**z**分为负类

有监督学习算法—线性SVM

线性SVM的扩展：

- 上述SVM算法假定数据线性可分，对于非线性可分的数据集，可以采用非线性SVM算法。
- 上述SVM算法假定训练数据仅有两类。对于多类数据，可以采用训练多个二分类SVM来实现多分类问题。常见的策略为：one-vs-all和one-vs-one