



第13章 文本分类与聚类

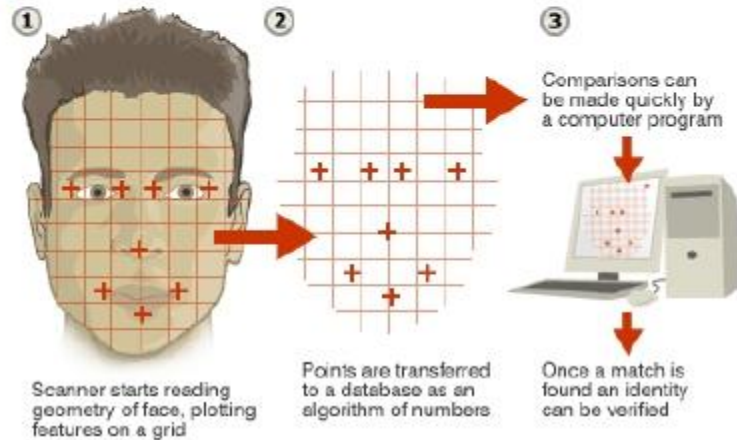


主要内容

- ◆ 文本分类
- ◆ 文本聚类

真实生活中的模式识别问题

HOW 2D FACIAL SCANNERS RECORD IDENTITIES





●●○○○ 中国移动 4G 22:51 29%

科技 体育 财经 军事 文化 旅游 +

俄T-90坦克在叙利亚被导弹打爆 乘员逃生

图片

手机和讯网 276评论

×

许世友因轻敌在越南遭受重创，战后他发誓不再进北京

百代旅行家 341评论 20分钟前

×

中国东风-21D导弹到底有多厉害？外媒一张图把国人惊呆了

热

迷彩先生 84评论 30分钟前

×

为何打仗需要它：解放军开始佩戴新型身份识别牌 全面与美军接轨

战略吐槽秀 3100评论 40分钟前

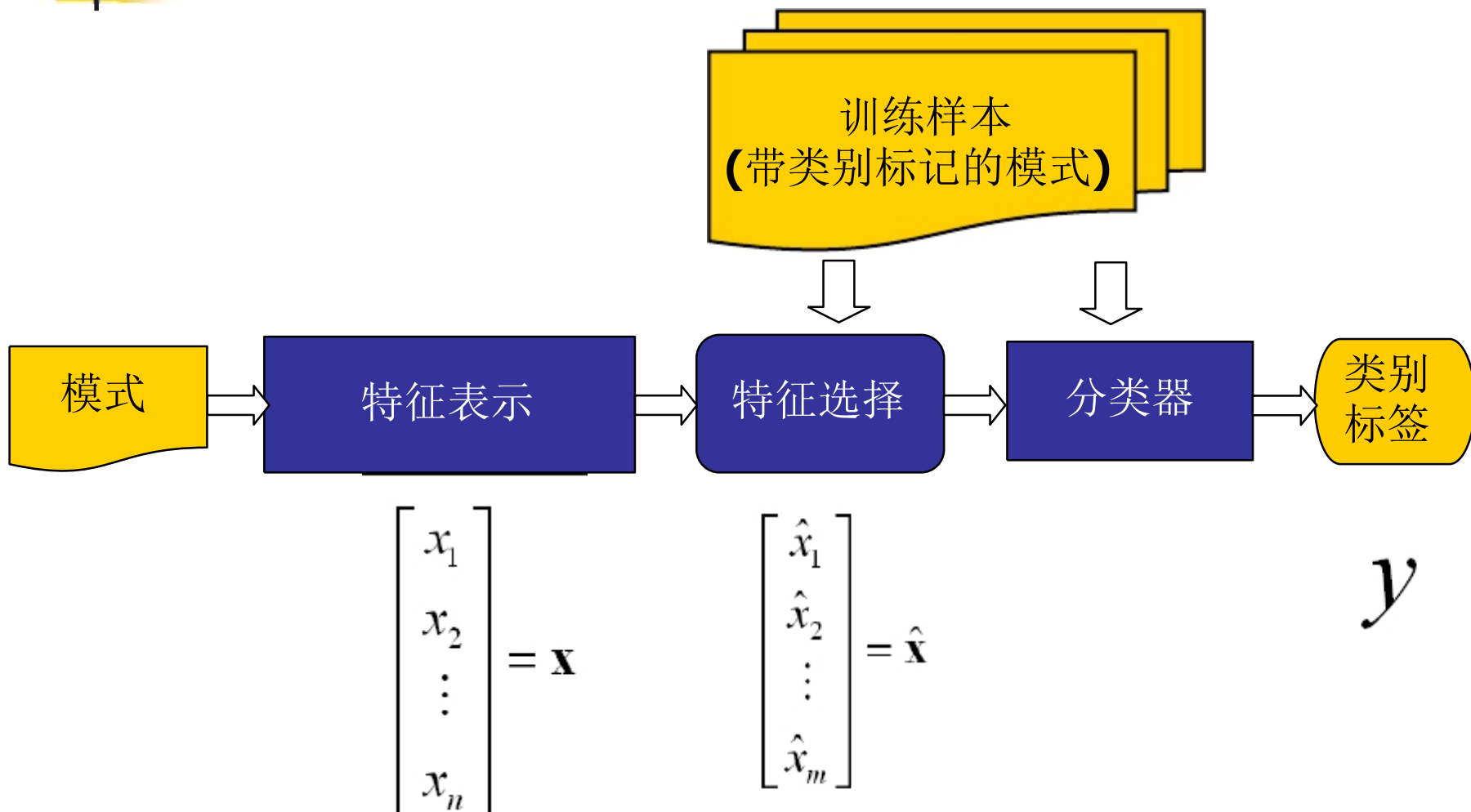
×

中国为何突然曝光绝密战机？俄罗斯空军表示望尘莫及

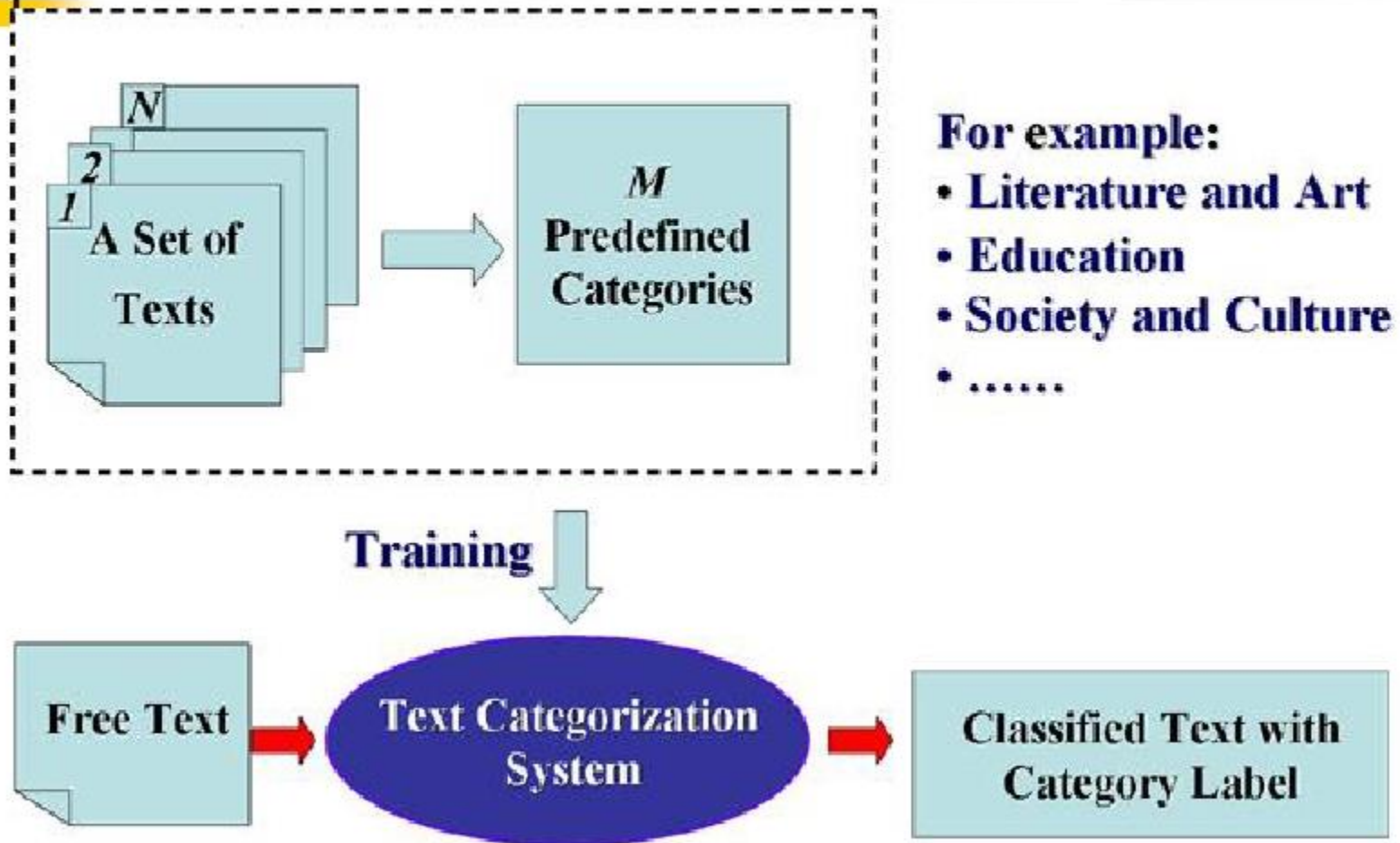
首马

×

模式识别系统的基本框架



文本分类系统的基本框架





主要内容

- ◆ 文本分类
 - 文本表示
 - 特征选择
 - 分类算法

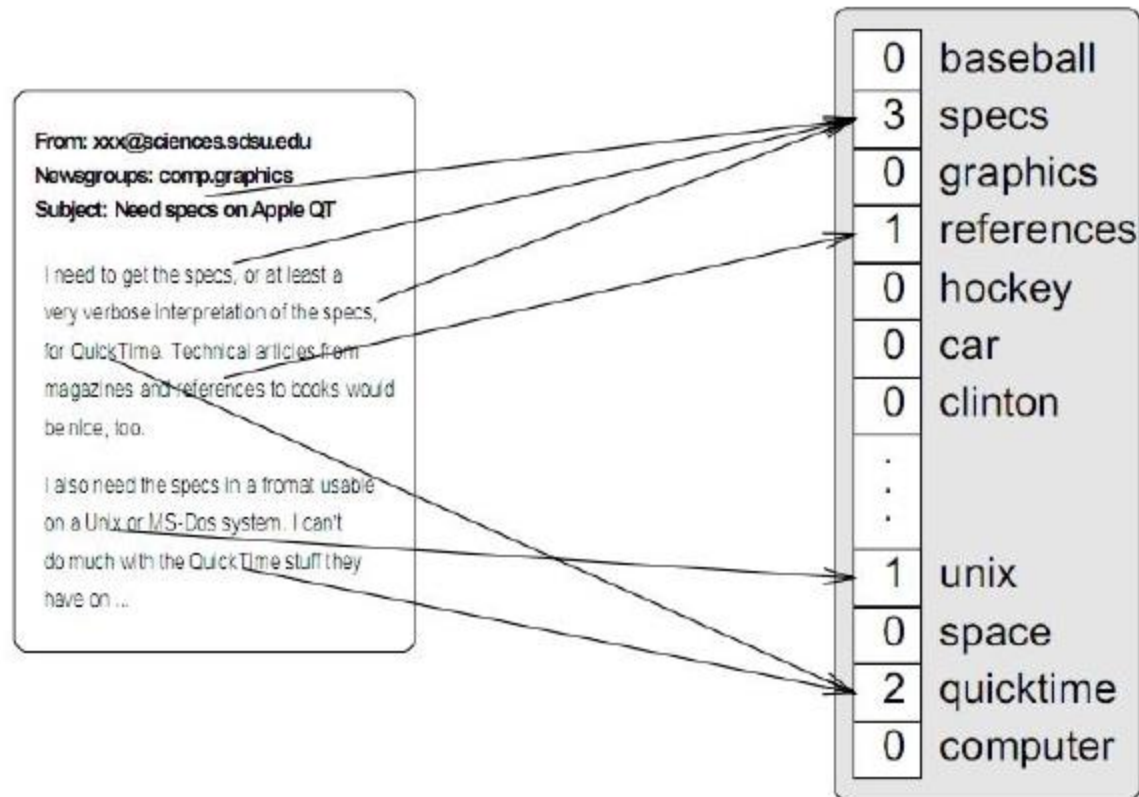
文本表示-离散表示

- ◆ 向量空间模型 (Vector Space Model, VSM)
 - 也称为词袋模型 (Bag-of-Words Model, BOW)



文本表示-离散表示

- ◆ 向量空间模型 (Vector Space Model, VSM)
 - 也称为词袋模型 (Bag-of-Words Model, BOW)



词的权重

- ◆ 词频 (Term Frequency, TF)

$$\omega_{ki} = tf_{ki}$$

- ◆ 布尔变量 (是否出现)

$$\omega_{ki} = \begin{cases} 1, & \text{if } t_i \text{ exists in } \mathbf{d}_k \\ 0, & \text{otherwise} \end{cases}$$

- ◆ 逆文档频率 (Inverse Document Frequency, IDF)

$$\omega_i = \log \frac{N}{df_i}$$

- ◆ TF-IDF

$$\omega_{ki} = tf_{ki} \cdot \log \frac{N}{df_i}$$

一个文本表示的例子

◆ 训练数据（带类别标签的文档）

教育

重庆 大学 计算机
专业 创建 于 1958 年
是 中国 最早 设立 计算
机 专业 的 高校 之一

体育

第五 届 东亚 运动会 中
国 军团 奖牌 总数 创 新
高 男女 排球 双双 夺冠

词袋表示下的文本

创建 大学 中国 重庆 夺冠 运动会 排球 高校 专业……

1	1	1	1	0	0	0	1	1	...
0	0	1	0	1	1	1	0	0	...



主要内容

- ◆ 文本分类
 - 文本表示
 - 特征选择
 - 分类算法

特征选择（特征过滤）

◆ 文本分类

■ 文本表示

■ 特征选择

- 文档频率（Document Frequency, DF）

- 互信息（Mutual Information, MI）

- 信息增益（Information Gain, IG）

- Chi-Square统计（Chi-Square Statistics, CHI）

■ 分类器设计

特征选择（特征过滤）

◆ 文档频率

根据包含某特征的文档的个数（频率）。对所有特征进行排序，去掉出现频率过低的特征。

◆ 词频

根据训练语料中某特征的频率，对所有特征进行排序

◆ 缺点

基于无监督思想，特征选择缺乏类别信息的指导

相关概率估计

■ 关于特征 t_i 与类别 c_j 的统计表

特征 \ 类别	c_j	$\overline{c_j}$
t_i	A_{ij}	B_{ij}
$\overline{t_i}$	C_{ij}	D_{ij}

$$P(c_j) \approx (A_{ij} + C_{ij}) / N_{all}$$

$$P(t_i) \approx (A_{ij} + B_{ij}) / N_{all}$$

$$P(\overline{t_i}) \approx (C_{ij} + D_{ij}) / N_{all}$$

$$P(c_j | t_i) \approx \frac{A_{ij} + 1}{A_{ij} + B_{ij} + C}$$

$$P(c_j | \overline{t_i}) \approx \frac{C_{ij} + 1}{C_{ij} + D_{ij} + C}$$

特征选择-互信息

■ 互信息 (Mutual Information, MI)

互信息是关于两个随机变量互相依赖程度的一种度量

$$I(X, Y) = H(X) - H(X | Y) = \sum_y \sum_x P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

$$MI(t_i, c_j) = \log \frac{P(t_i, c_j)}{P(t_i)P(c_j)} \approx \log \frac{A_{ij} N_{all}}{(A_{ij} + C_{ij})(A_{ij} + B_{ij})}$$

$$MI_{avg}(t_i) = \sum_{j=1}^C P(c_j) MI(t_i, c_j)$$

特征选择-信息增益

■ 信息增益 (IG)

$$\begin{aligned} & IG(t_i) \\ &= \{-\sum_{j=1}^C P(c_j) \log P(c_j)\} \\ &+ \{P(t_i) [\sum_{j=1}^C P(c_j | t_i) \log P(c_j | t_i)] \\ &+ P(\bar{t}_i) [\sum_{j=1}^C P(c_j | \bar{t}_i) \log P(c_j | \bar{t}_i)]\} \end{aligned}$$

IG 衡量特征能够为分类系统带来多少信息

信息增益实质是**不考虑任何特征时文档的熵**和**考虑特征t后文档的熵**的差值(Entropy(S)-Expected Entropy(S_{t_i}))。

“计算机”的信息增益

特征 \ 类别	教育	体育
计算机	2	0
$\overline{\text{计算机}}$	0	2

$$\begin{aligned}
 P(\text{计算机}) &= 1/2 & \overline{P(\text{计算机})} &= 1/2 \\
 P(\text{教育} | \text{计算机}) &= (2+1)/(2+2) = 3/4 \\
 P(\text{体育} | \text{计算机}) &= 1/(2+2) = 1/4 \\
 P(\text{教育} | \overline{\text{计算机}}) &= 1/(2+2) = 1/4 \\
 P(\text{体育} | \overline{\text{计算机}}) &= (2+1)/(2+2) = 3/4
 \end{aligned}$$

$$IG(\text{计算机}) = -0.5 \log 0.5 - 0.5 \log 0.5$$

$$+ 0.5(0.75 \log 0.75 + 0.25 \log 0.25)$$

$$+ 0.5(0.75 \log 0.75 + 0.25 \log 0.25)$$

$$\begin{aligned}
 IG(t_i) &= \{-\sum_{j=1}^C P(c_j) \log P(c_j)\} \\
 &+ \{P(t_i) [\sum_{j=1}^C P(c_j | t_i) \log P(c_j | t_i)] \\
 &+ P(\overline{t_i}) [\sum_{j=1}^C P(c_j | \overline{t_i}) \log P(c_j | \overline{t_i})]\} \\
 &= -\log 0.5 + 0.75 \log 0.75 + 0.25 \log 0.25 = 0.1308
 \end{aligned}$$

“北京” 的信息增益

<i>feature</i> \ <i>class</i>	教育	体育
北京	2	1
$\overline{\text{北京}}$	0	1

$$P(\text{北京}) = (1+2)/4 = 3/4$$

$$P(\overline{\text{北京}}) = 1/4$$

$$P(\text{教育} | \text{北京}) = (2+1)/(3+2) = 3/5$$

$$P(\text{体育} | \text{北京}) = (1+1)/(3+2) = 1/5$$

$$P(\text{教育} | \overline{\text{北京}}) = 1/(1+2) = 1/3$$

$$P(\text{体育} | \overline{\text{北京}}) = (1+1)/(1+2) = 2/3$$

$$IG(\text{北京})$$

$$= -0.5 \log 0.5 - 0.5 \log 0.5$$

$$+ 0.75(0.6 \log 0.6 + 0.4 \log 0.4)$$

$$+ 0.25(0.667 \log 0.667 + 0.333 \log 0.333)$$

$$= 0.0293$$



信息增益的例子

根据信息增益的特征排序

Features	IG
计算机 排球 运动会	0.1308
创建 东亚 高校 奖牌 锦标赛 军团 男女 设立 双双	0.0293
的 夺冠 年 是 中国	0.0000

特征选择-CHI

■ Chi-Square 统计量 (CHI)

CHI统计量衡量的是特征项 t_i 和类别 C_j 之间的相关程度。

特征项 \ 类别	C_j	$\sim C_j$
t_i	A	B
$\sim t_i$	C	D

$$\chi^2(t_i, c_j) = \frac{N_{all} \cdot (A_{ij}D_{ij} - C_{ij}B_{ij})^2}{(A_{ij} + C_{ij}) \cdot (B_{ij} + D_{ij}) \cdot (A_{ij} + B_{ij}) \cdot (C_{ij} + D_{ij})}$$

$$CHI_{avg}(t_i) = \sum_{j=1}^C P(c_j) \chi^2(t_i, c_j) \quad \leftarrow \text{多类别, 求概率平均}$$



主要内容

◆ 文本分类

- 文本表示
- 特征选择
- 分类算法
 - 朴素贝叶斯 (Naïve Bayes)
 - 线性判别函数 (Linear Discriminate Function)
 - KNN算法

分类算法

◆ 监督学习

■ 生成式模型

- 朴素贝叶斯 (Naïve Bayes)

■ 判别式模型

- 线性判别函数 (Linear Discriminate Function)
- 支持向量机 (Support Vector Machine)
- 最大熵模型 (Maximum Entropy)

◆ 无监督、半监督学习

监督学习过程

◆ 我们有什么？

■ 训练数据

■ 我们的任务是什么？

■ 利用参数构建模型（目标函数）

■ 参数需要估计

■ 如何进行参数估计？

■ 根据某个准则从训练数据中学习

■ 学习在训练数据上准则最优的参数

$$y = f(x; \theta)$$

↑
 $\theta?$

↑

$$\theta := \theta + \nabla f$$

贝叶斯决策理论

■ 贝叶斯决策理论

$$P(c_j | \mathbf{x}) = \frac{P(c_j, \mathbf{x})}{P(\mathbf{x})} = \frac{P(c_j)P(\mathbf{x}|c_j)}{P(\mathbf{x})}$$

■ 学习难点

$$P(\mathbf{x} | c_j) = ???$$

■ 朴素贝叶斯假设

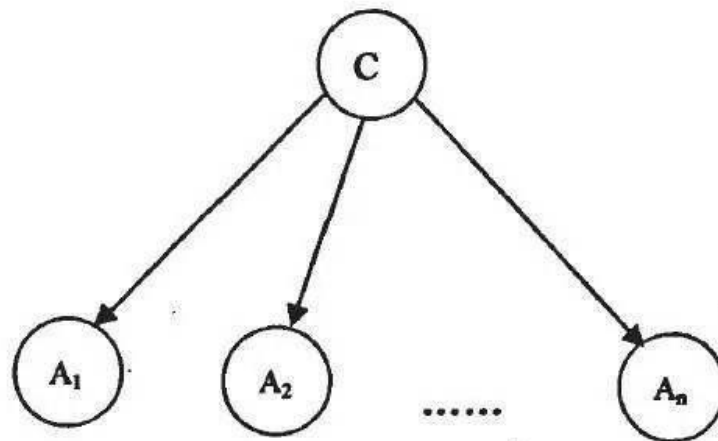
$$P(\mathbf{x} | c_j) \approx P([w_1, \dots, w_N] | c_j) \approx \prod_{k=1}^N P(w_k | c_j) = \prod_{i=1}^M P(w_i | c_j)^{N(w_i)}$$

朴素贝叶斯分类器

■ 朴素贝叶斯模型

$$P(c_j | \mathbf{x}) = \frac{P(\mathbf{x}, c_j)}{P(\mathbf{x})} \propto P(\mathbf{x}, c_j) = P(c_j) \prod_{i=1}^M P(w_i | c_j)^{N(w_i)}$$

$$c^* = \arg \max_{j=1, K, C} P(c_j) \prod_{i=1}^M P(w_i | c_j)^{N(w_i)}$$



NB模型中的参数估计

■ 最大似然估计

$$P(c_j) \approx \frac{1 + N(c_j)}{C + N_{\text{all}}}$$

N_{all} 为文档总数

$$P(w_i | c_j) \approx \frac{1 + N(w_i, c_j)}{M + \sum_{i'=1}^M N(w_{i'}, c_j)}$$

M 为特征词总数

■ NB模型一个例子

$P(c_j)$	$P(\text{教育})=0.5$	$P(\text{体育})=0.5$
	$P(\text{计算机} \text{教育})=0.3$	$P(\text{计算机} \text{体育})=0.1$
	$P(\text{排球} \text{教育})=0.1$	$P(\text{排球} \text{体育})=0.3$
$P(w_i/c_j)$	$P(\text{运动会} \text{教育})=0.1$	$P(\text{运动会} \text{体育})=0.3$
	$P(\text{高校} \text{教育})=0.2$	$P(\text{高校} \text{体育})=0.1$
	$P(\text{大学} \text{教育})=0.3$	$P(\text{大学} \text{体育})=0.2$

NB决策的例子

“复旦 大学 排球 队 获得 本届 大学生 运动会 排球 比赛 冠军”

Feature Set = [计算机, 排球, 运动会, 高校, 大学]

文档特征 $\mathbf{x} = [0, 1, 1, 0, 1]_T$

$$P(\text{教育} | \mathbf{x}) = P(\text{教育})P(\mathbf{x} | \text{教育}) = 0.5 \times 0.1 \times 0.1 \times 0.3 = 0.0015$$

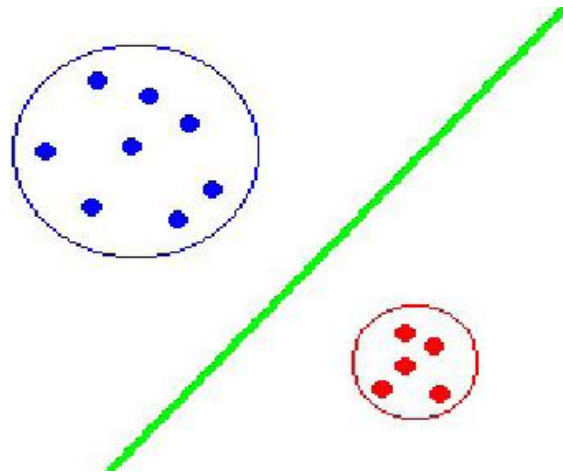
$$P(\text{体育} | \mathbf{x}) = P(\text{体育})P(\mathbf{x} | \text{体育}) = 0.5 \times 0.3 \times 0.3 \times 0.2 = 0.0090$$

$$P(\text{教育} | \mathbf{x}) = \frac{0.0015}{0.0015 + 0.0090} = 0.1429$$

$$P(\text{体育} | \mathbf{x}) = 0.8571$$

线性判别函数

- 线性判别函数是模式识别中一种重要的方法，它使用训练样本集确定一个**最优的线性超平面**。
- 在二维空间中，线性判别函数就是一条直线，在三维空间中就是一个平面。



线性判别函数

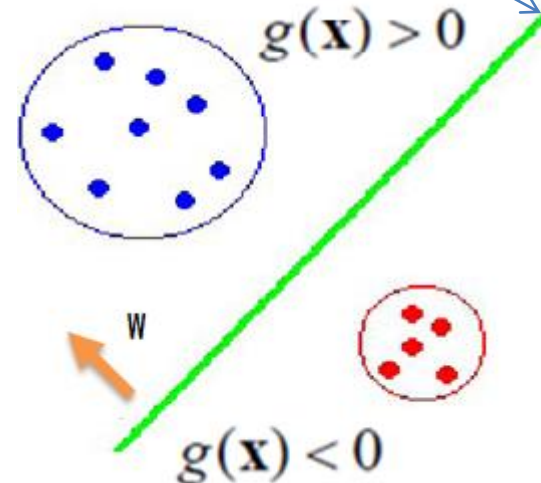
■ 模型表示

$$g(\mathbf{x}) = \underset{\substack{\uparrow \\ \text{权重向量}}}{\mathbf{w}^T} \mathbf{x} + \underset{\substack{\uparrow \\ \text{偏移(阈值)}}}{b} = \sum_{l=1}^M w_l x_l + b$$

\mathbf{x} 是样本特征向量， \mathbf{w} 是法向量，决定了决策面的方向

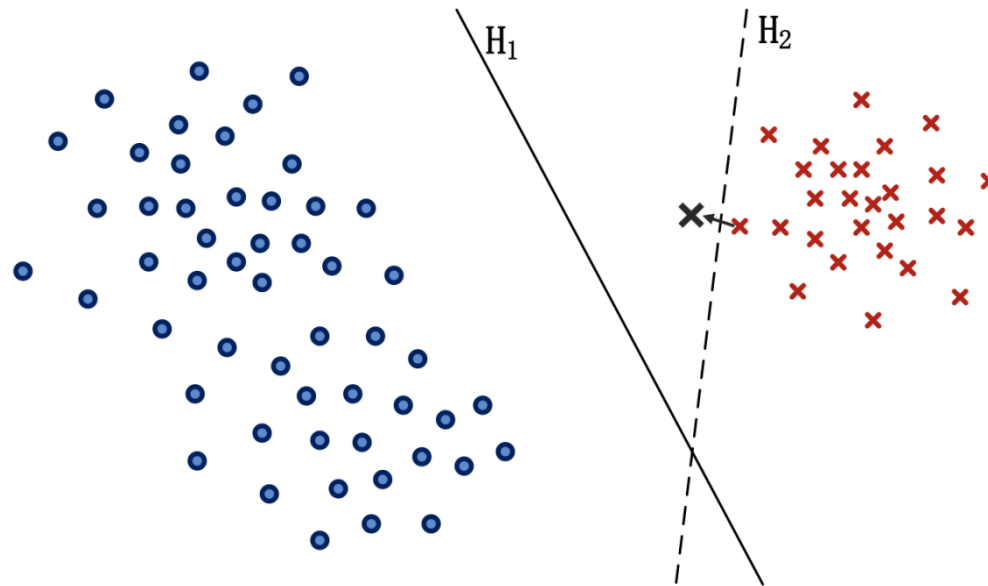
• 两类情形的决策规则：

$$\begin{cases} \mathbf{x} \in \omega_1, & \text{if } g(\mathbf{x}) > 0 \\ \mathbf{x} \in \omega_2, & \text{if } g(\mathbf{x}) < 0 \\ \text{uncertain,} & \text{if } g(\mathbf{x}) = 0 \end{cases}$$



线性支持向量机

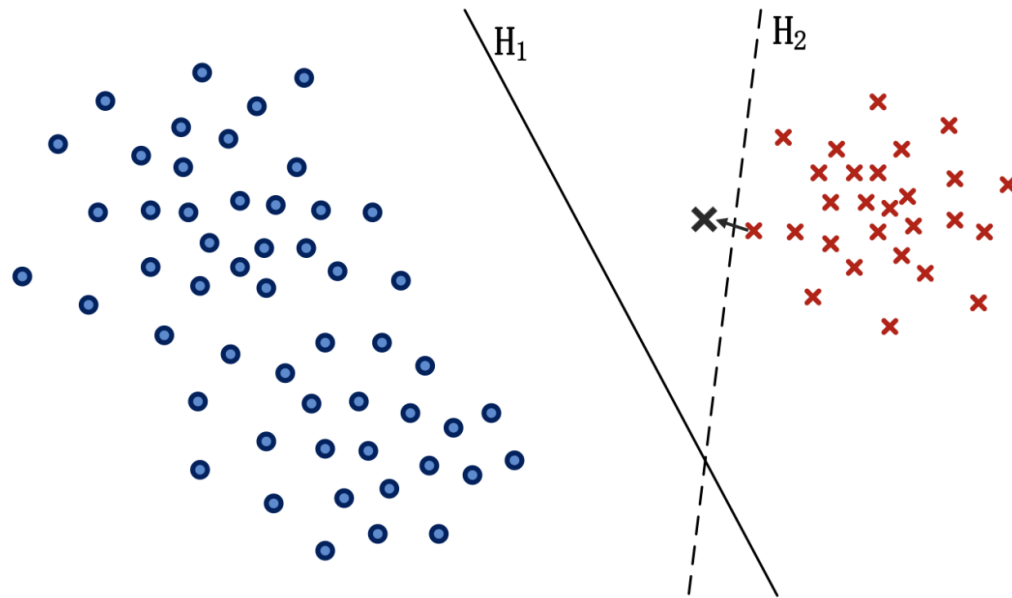
- SVM (Support Vector Machine) 是一种最常用的线性判别函数，主要用于样本线性可分的情况，但加入核函数后，可以实现非线性分类器的效果。



从这张图，可以看见对于现有训练数据， H_1 , H_2 都是分类超平面

线性支持向量机

- 对于未来要预测的数据，显然H1的效果会更好。因为它离各个类别的“距离最远”，鲁棒性更强。
- SVM便是出于这种“距离最远”的想法，希望能够得到H1类型的分类平面。



线性支持向量机

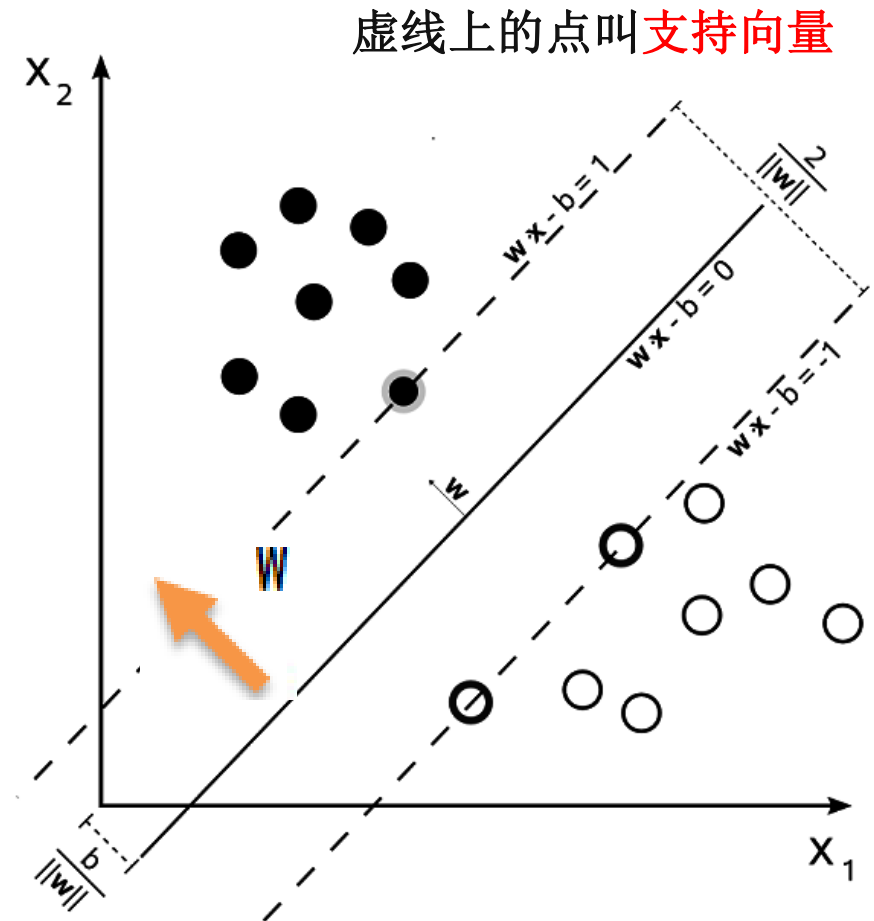
■ 判别函数

$$y = \mathbf{w}^T \mathbf{x} + b$$

■ 最大间隔准则

$$\min \frac{1}{2} \|\mathbf{w}\|^2$$

$$s.t. y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, \dots, n$$



线性支持向量机

$$\begin{aligned} \min & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t. } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, \dots, n \end{aligned}$$

对上述公式，使用拉格朗日乘子法得到其对偶问题（变等式），该问题的拉格朗日函数可以写为

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \quad (8)$$

分别对 \mathbf{w} 和 b 求偏导：

$$\begin{cases} \frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \\ \frac{\partial L}{\partial b} = \sum_{i=1}^m \alpha_i y_i \end{cases} \quad \text{令其分别为0} \quad \begin{cases} \mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i & (9) \\ \sum_{i=1}^m \alpha_i y_i = 0 & (10) \end{cases}$$

线性支持向量机

将 (9, 10) 代入公式 (8), 可得

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, m \end{aligned} \quad (12)$$

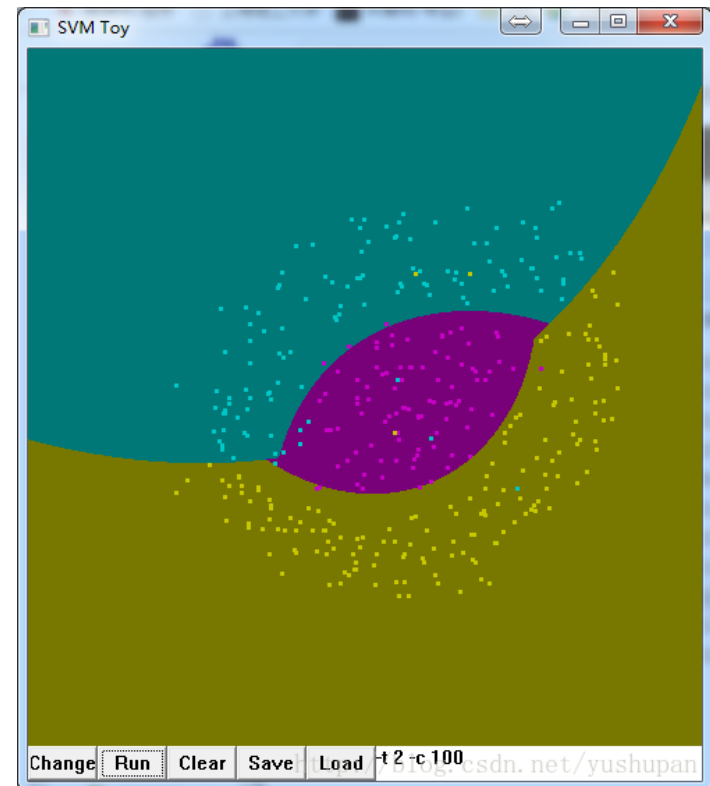
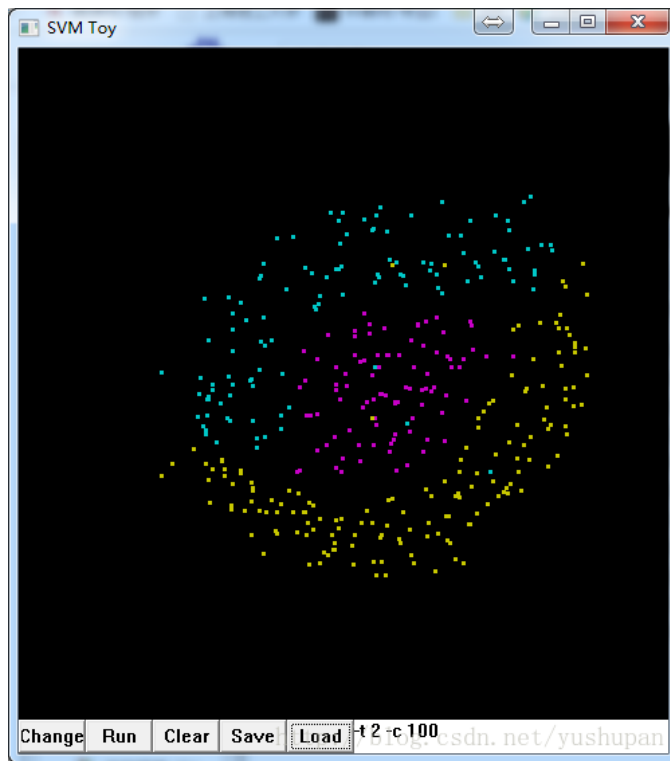
解出 α 后, 根据公式 (9) 可以求得 w , 进而求得 b , 最后可以得到模型

$$f(x) = w^T x + b = \sum_{i=1}^m \alpha_i y_i x_i^T x + b \quad (13)$$

线性支持向量机

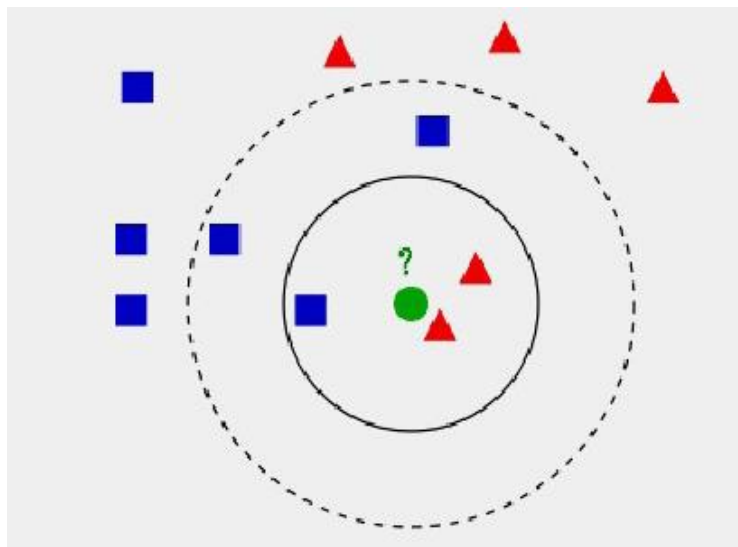
常用的SVM工具包，支持java，c和python

LIBSVM——<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>



KNN算法

- 最近邻 (k-Nearest Neighbors, KNN) 算法是一种分类算法。
- 该算法的思想是： 如果一个样本与数据集中的k个样本相似，且这k个样本中的大多数属于某个类别i，则该样本也属于类别i。

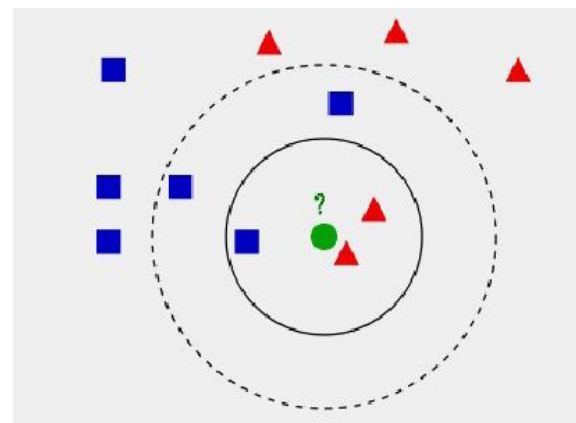


KNN算法

- KNN算法是一种“懒惰”的算法，不需要明显的学习过程。
- KNN算法受到K值和所选距离度量标准的影响
- K值的确定

如果选择较小的K值，预测结果会对邻近的实例点非常敏感，如果邻近的实例点恰巧是噪声，预测就会出错。

如果k值选择较大，算法对噪声比较鲁棒，但不确定性也会随之增加。



KNN算法

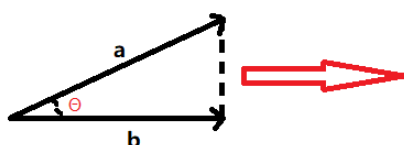
■ 距离量度

样本空间中，两个点之间的距离表示两个样本点的相似程度：距离越短，相似程度越高。

常用距离包括欧氏距离或曼哈顿距离。

欧式距离: $d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$, 曼哈顿距离: $d(x, y) = \sqrt{\sum_{k=1}^n |x_k - y_k|}$

文本之间的距离，可以用余弦相似度，即两个向量夹角的余弦值



$\cos(\theta) = b/a$

$$\text{cosine}(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\|\mathbf{d}_j\| \times \|\mathbf{q}\|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$



KNN算法

■ KNN算法描述

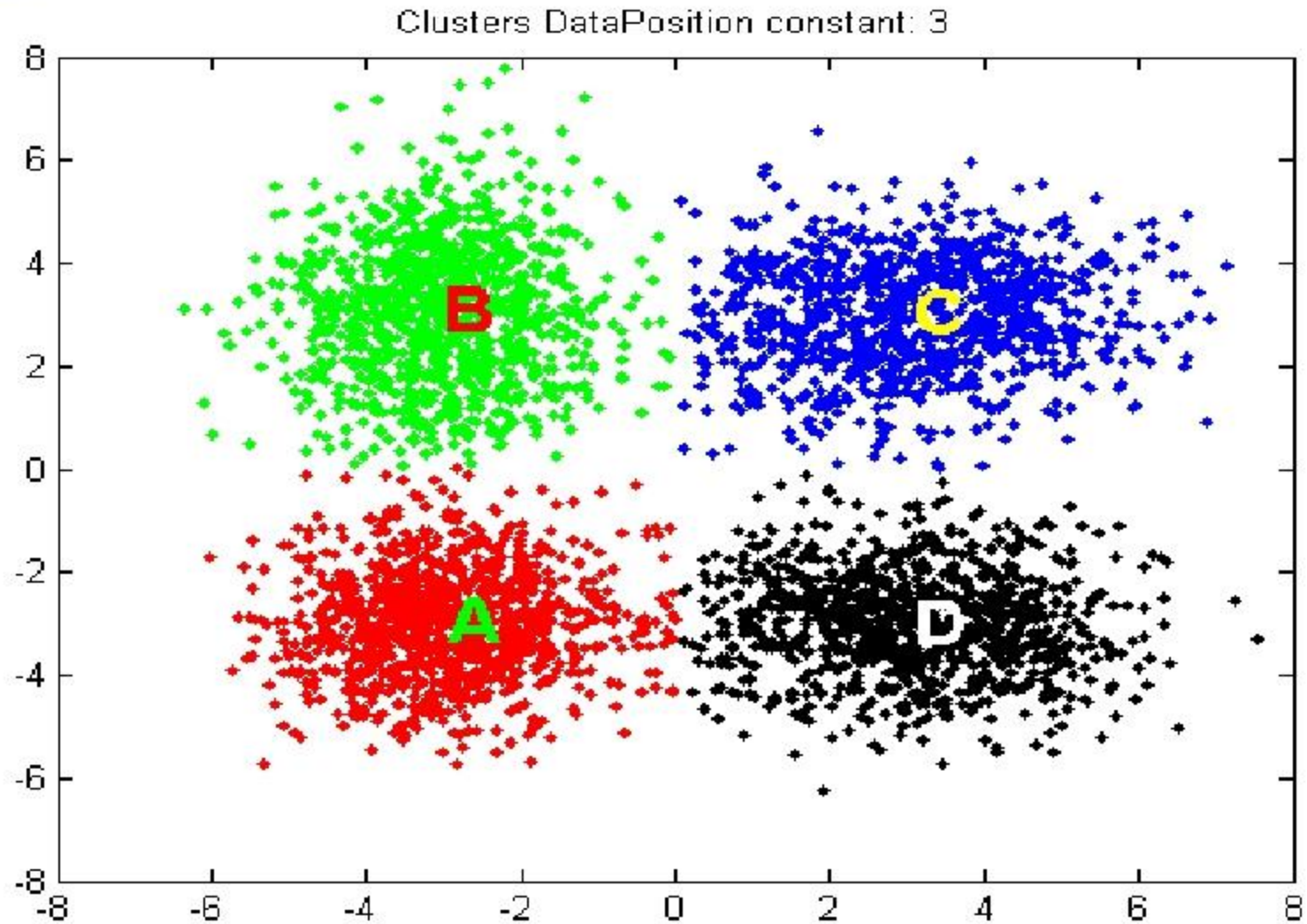
- 1) 计算测试样本与各个训练样本之间的距离;
- 2) 按照距离的递增关系进行排序;
- 3) 选取距离最小的K个样本点;
- 4) 计算前K个点分属不同类别的次数;
- 5) 返回前K个点中出现频率最高的类别作为测试数据的预测分类。



主要内容

- ◆ 文本分类
 - 文本表示
 - 特征选择
 - 分类算法
- ◆ 文本聚类

文本聚类



文本聚类

◆ 假设

- 同类的文本相似度较大
- 不同类的文本相似度较小

◆ 与文本分类的区别

- 没有带标签的训练数据
- 不采用生成式或判别式模型的方法

文本聚类算法

◆ 分割法

- K-means算法
- K-medoids算法
- CLARANS算法

◆ 层次法

- BIRCH算法
- CURE算法

◆ 基于密度的方法

◆ 基于网格的方法

K-means算法

◆ 流程

- 1, 随机选取k个文本作为初始的聚类中心;
- 2, 根据与聚类中心的距离, 将每个文本重新赋给最相似的簇;
- 3, 如果所有文本均分配了, 重新计算每个簇中所有文本的平均值, 用此平均值作为新的聚类中心;
- 4, 重复执行2、3步, 直到各个簇不再发生变化

K-medoids 算法

◆ 流程

- 1, 随机选取 k 个文本作为初始的聚类种子;
- 2, 根据聚类种子的值, 将每个文本重新赋给最相似的簇;
- 3, 重新选择每个簇的**中心文本 (点)**, 要求该文本到簇中其他所有文本的距离之和最小, **用此文本作为新的聚类种子**;
- 4, 重复执行2、3步, 直到各个簇不再发生变化



Thanks

谢谢!