DMWIS

吉林大学 JILIN UNIVERSITY

自然语言处理

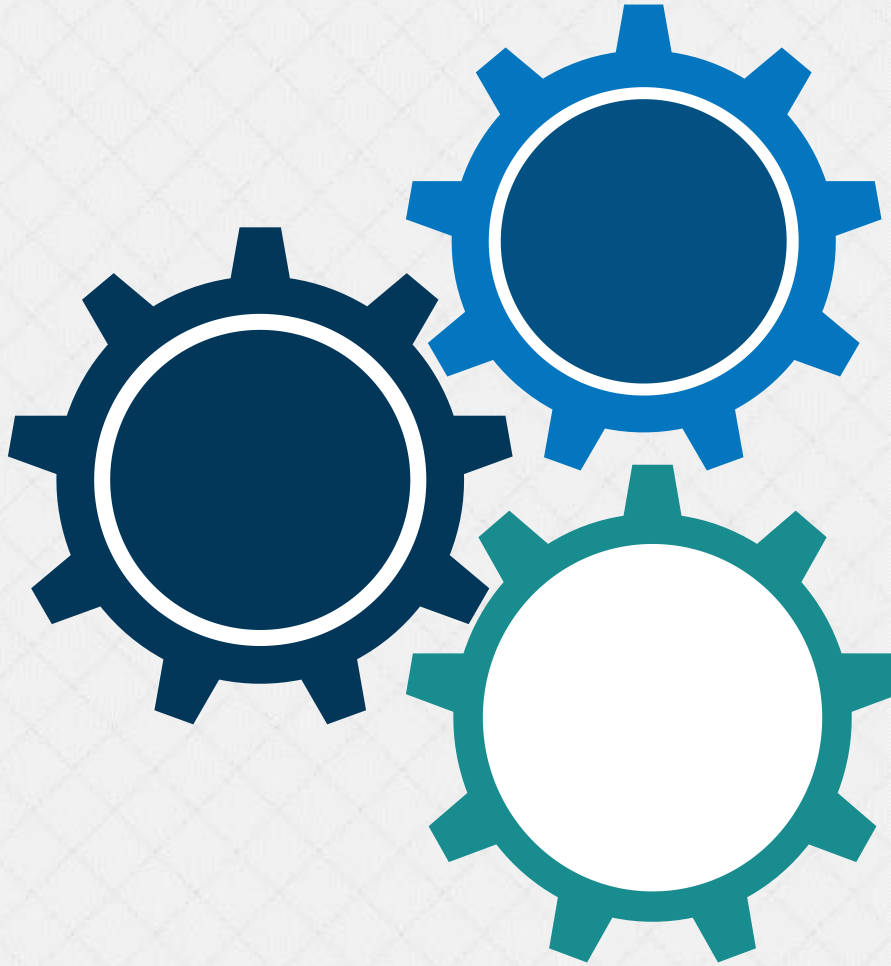
包铁

baotie@jlu.edu.cn
Data Mining and Web Information System Group （DMWIS），
College of Computer Science and Technology, Jilin University

2023年10月14日

主要内容

1. 信息抽取
2. 文本聚类
3. 文本分类
4. 文本表示
5. 文本相似度



信息抽取-宽泛的概念

- 指从非结构化文本中提取结构化信息的技术
  - 实现方法：基于规则的正则匹配、有监督和无监督的机器学习等...
- 可以使用一些简单的无监督学习方法
  - 不需要标注语料库
  - 可以利用现有的海量非结构化文本
  - 按照颗粒的大小包括新词、关键词、关键短语、关键句等...

信息抽取

新词提取

- **词典之外的词语**（也就是未登录词OOV）称作新词
  - 新词是一个相对的概念，暂无统一明确的定义
  - 语料库标注成本较高，需要修订词典（领域），无监督方法较适合
- 新词提取的原理
  - 提取出大量文本（生语料）中的**词语**，无论新旧
  - 用词典过滤掉已有的词语，于是得到新词
  - 词的判别-左右搭配很丰富、内部成分搭配很固定
  - 判别的指标-**信息熵**（搭配丰富）、**互信息**（搭配固定）

信息抽取

信息熵

- **信息熵**（entropy）指某消息所含的信息量，计算方法如下：

$$H(X) = -\sum_{x} p(x)\log p(x)$$

- 单次抛硬币试验结果的信息熵为（对数常以2为底）：

$$H(X) = -\left(p(x=\text{正})\log p(x=\text{正}) + p(x=\text{反})\log p(x=\text{反})\right)$$

$$= -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$$

$$= -\log\frac{1}{2} = 1$$

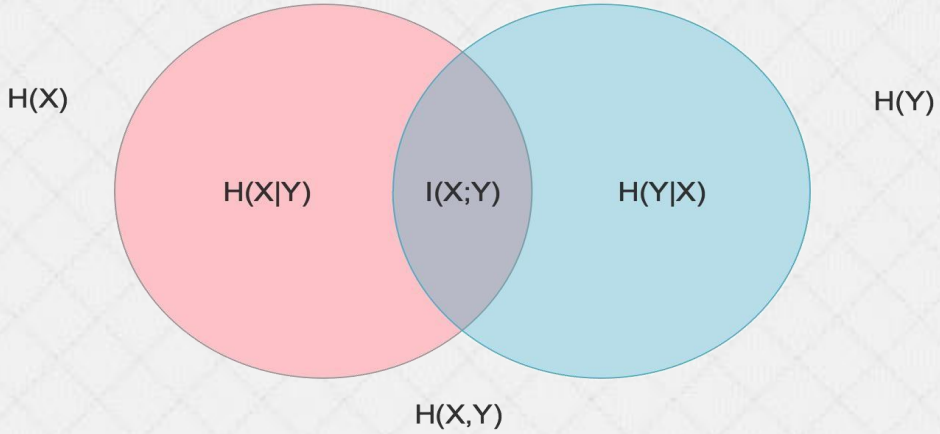
两只蝴蝶飞呀飞
这群蝴蝶飞走了

互信息

- **互信息**（Mutual Information）指的是两个离散型随机变量X与Y相关程度的度量，定义如下：

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) \\
&= H(X) + H(Y) - H(X,Y) \\
&= \sum_x p(x) \log \frac{1}{p(x)} + \sum_y p(y) \log \frac{1}{p(y)} - \sum_{x,y} p(x,y) \log \frac{1}{p(x,y)} \\
&= \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}
\end{aligned}
$$

两只蝴蝶飞呀飞
这群蝴蝶飞走了



关键词提取

- **提取文中重要的词**-关键词的判断难以统一
  - 单文档分析：词频、TextRank，可以在每篇文档中独立使用
  - 多文档分析：TF-IDF，可以利用其他文档的信息，但易受噪声干扰
- **词频统计**-主要介绍无监督方法
  - 文中反复出现的词，可能需要去掉标点符号和无意义的助词
  - 需要先分词、去停用词，然后再统计词频

关键词提取

- 相较于词频，**TF-IDF**还综合考虑词语的稀有程度

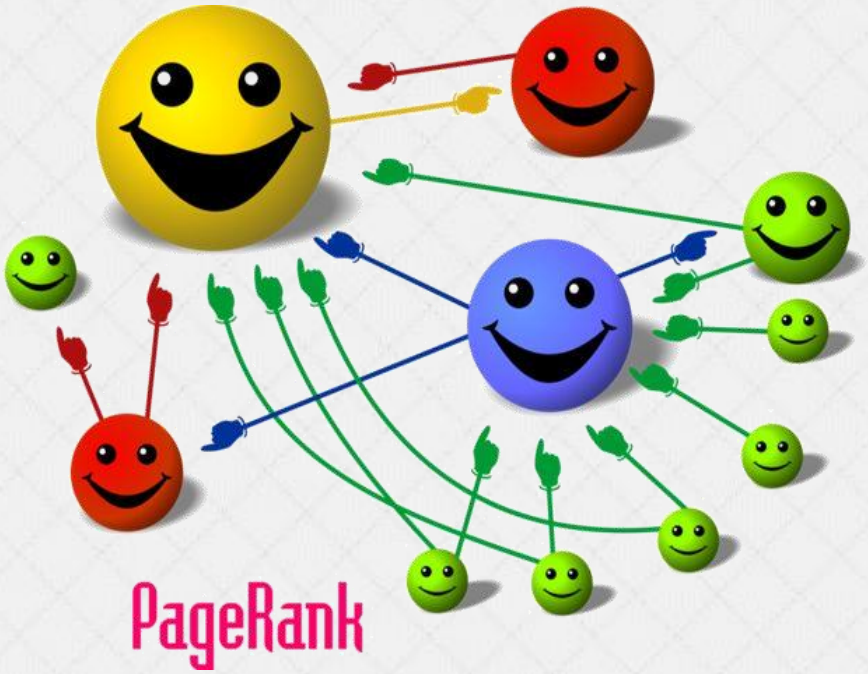
$$\begin{aligned}\text{TF - IDF}(t,d) &= \frac{\text{TF}(t,d)}{\text{DF}(t)} \\ &= \text{TF}(t,d)\cdot\text{IDF}(t)\end{aligned}$$

- 其中，t代表单词，$\text{TF}(t,d)$代表t在d中的出现频次，$\text{DF}(t)$代表有多少篇文档包含t。DF的倒数称为IDF，这也是TF-IDF得名的由来。

关键词提取

- TextRank就是PageRank在文本上的应用
- PageRank将互联网看作有向图，互联网上的网页视作节点，迭代更新权重
  - d 表示阻尼系数，为了解决没有入链网页的得分-一般取值0.85

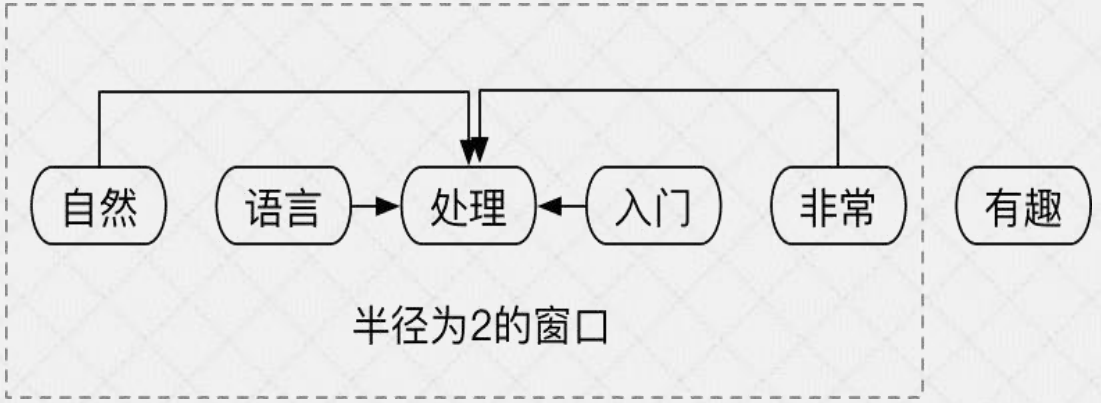
$$S(V_i) = (1-d) + d \times \sum_{Vj \in In(V_i)} \frac{1}{|\ Out(V_j)\ |} S(V_j)$$



关键词提取

- 将PageRank应用到关键词提取，将单词视作节点，每个单词的外链来自自身前后固定大小的窗口内的所有单词
  - 文本分句后分词、词性标注、去停用词
  - 选择词构建无向图-采用共现关系构建两点间的边
  - 迭代计算权重至收敛

$$WS(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in Out(V_j)}w_{jk}}WS(V_j)$$



短语提取

- 新词提取中的“字符”替换为“单词”，“字符串”替换为“单词列表”，可以得到短语提取
- 需要分词，过滤停用词可能得到更好的效果
- 利用信息熵、互信息
- 关键短语-相邻的关键词可以组合为关键短语

关键句提取

- 改进链接的BM25权重计算
  - 窗口的中心句与相邻的句子间的链接有强有弱，相似的句子将得到更高的投票

$$\mathrm{BM25}(D,Q)=\sum_{i=1}^{n}\mathrm{IDF}(q_i)\cdot\frac{\mathrm{TF}(q_i,D)\cdot(k_1+1)}{\mathrm{TF}(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{avgDL})}$$

  - k_1和b是两个常数，$avgDL$是所有文档的平均长度，Q为查询语句
  - k_1越大，TF对文档得分的正面影响越大。b越大，文档长度对得分的负面影响越大。k_1=b=0，则值为所有单词的IDF之和

关键句提取

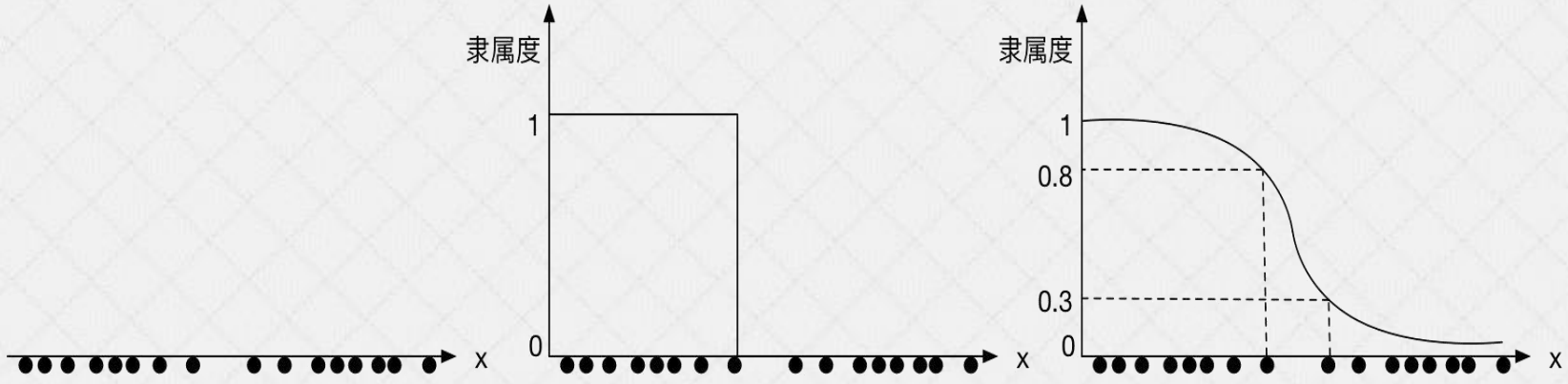
- 以BM25相似度作为PageRank中的链接的权重，于是得到一种改进算法，称为TextRank

$$\mathrm{WS}(V_i) = (1-d) + d \times \sum_{Vj \in In(V_i)} \frac{\mathrm{BM25}(V_i, V_j)}{\sum_{V_k \in Out(V_j)} \mathrm{BM25}(V_k, V_j)} \mathrm{WS}(V_j)$$

- WS迭代后得到该句子最终分数，排序后输出前N个即为关键句，由于句子数量较少且不重复，不再取窗口，认为所有句子都相邻

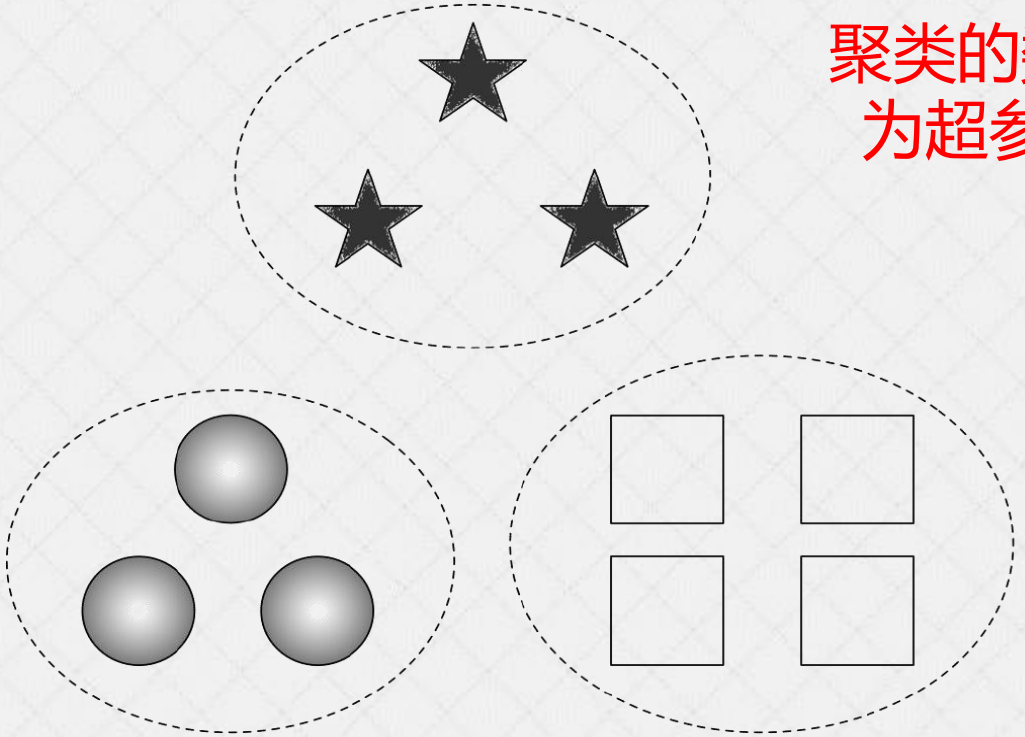
文档层级的聚类任务

- **聚类**（cluster analysis）指的是将给定对象的集合划分为不同子集的过程 - 子集内元素尽量相似、子集间元素尽量不同
  - 这些子集又被称为**簇**（cluster），一般没有交集
- 根据元素从属于集合的确定程度，聚类分为硬聚类和软聚类。
  - 硬聚类
  - 软聚类

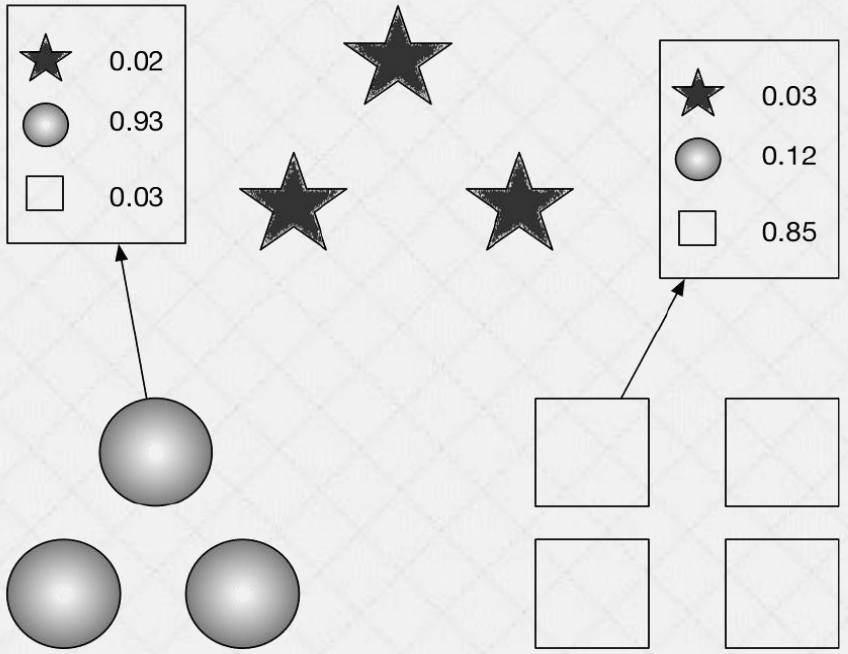


硬聚类与软聚类

聚类的数量
为超参数



硬聚类



软聚类

作用与流程

- **文本聚类**（text clustering，也称文档聚类或document clustering）指的是对文档进行的聚类分析
  - 改善搜索结果，生成同义词；文本预处理时可以在聚类中选代表性样本
- **文本聚类的基本流程**
  - 特征提取-文档表示为向量
  - 向量聚类-多种聚类算法

文档的特征提取

- **如何将一篇文档表示为一个向量？ - 文档不定长、单词种类无穷大**
- **采用词袋模型-有损**
  - 词袋（bag-of-words）是信息检索与自然语言处理中最常用的文档表示模型，它将文档想象为一个装有词语的袋子，通过袋子中每种词语的计数等统计量将文档表示为向量

词袋模型

- **文档：** 人 吃 鱼。

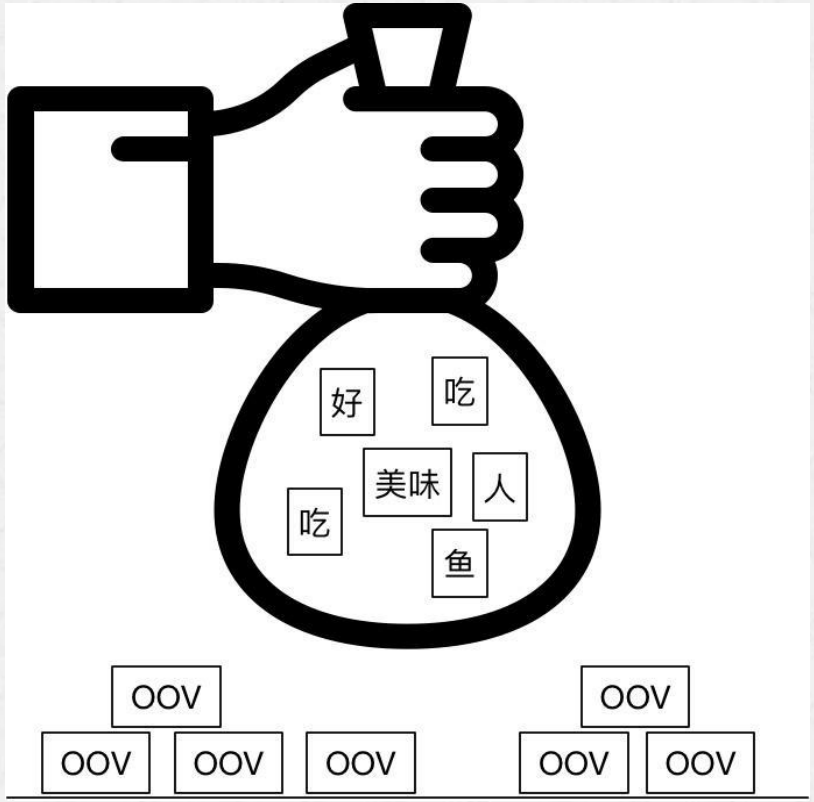
  美味 好 吃！

- **词频统计**（文档表示）=[1, 2, 1, 1, 1]

  人=1 吃=2 鱼=1 美味=1 好=1

- **新文档：** “人 吃 大 鱼”

  词袋向量表示是[1, 1, 1, 0, 0]



词袋中的统计指标

- **词频、布尔词频**（词频非零取为1，否则为0）
- **TF-IDF**-将每个词语的倒排频次也纳入考虑
- **词向量**-词语本身也是向量，可以将词向量求和作为文档向量

- 由n个文档组成的集合S，其中第i个文档d_i的特征向量为$\boldsymbol{d}_i$，**文档词频表示**计算方式如下：
- $\boldsymbol{d}_i = \left(\mathrm{TF}(t_1, d_i), \mathrm{TF}(t_2, d_i), \cdots, \mathrm{TF}\left(t_j, d_i\right), \cdots, \mathrm{TF}(t_m, d_i)\right)$
- 其中t_j表示词表中第j种单词，**m为词表大小**。$\mathrm{TF}(t_j, d_i)$表示单词t_j在文档d_i中的出现次数，缩放向量使得$\| \boldsymbol{d} \| = 1$。

k 均值算法-简单易用的聚类算法

- 定义 k 均值算法所解决的问题，给定n个向量$\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n \in \mathbb{R}^l$以及一个整数$k$，要求找出$k$个簇$S_1, \cdots, S_k$以及各自的质心$\boldsymbol{c}_1, \cdots, \boldsymbol{c}_k \in \mathbb{R}^l$，使得下式最小：

$$\text{minimize } \mathcal{I}_{\text{Euclidean}} = \sum_{r=1}^{k} \sum_{d_i \in S_r} \|\mathbf{d}_i - \mathbf{c}_r\|^2$$

- 其中$\| \boldsymbol{d}_i - \boldsymbol{c}_r \|$是向量与质心的欧拉距离，$\mathcal{I}_{\text{Euclidean}}$称作聚类的**准则函数**（criterion function）

k 均值算法-原理

- 而质心的计算就是簇内数据点的几何平均：

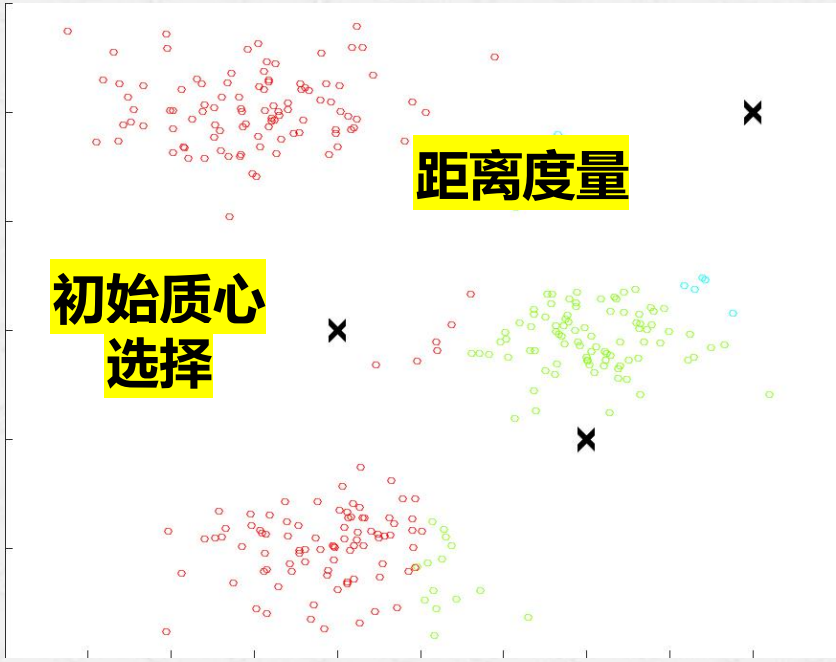
$$\boldsymbol{s}_i = \sum_{\mathbf{d}_j \in S_i} \boldsymbol{d}_j$$

$$\boldsymbol{c}_i = \frac{\boldsymbol{s}_i}{|S_i|}$$

- 其中，$\boldsymbol{s}_i$是簇S_i内所有向量之和，称作**合成向量。**

k 均值算法-原理

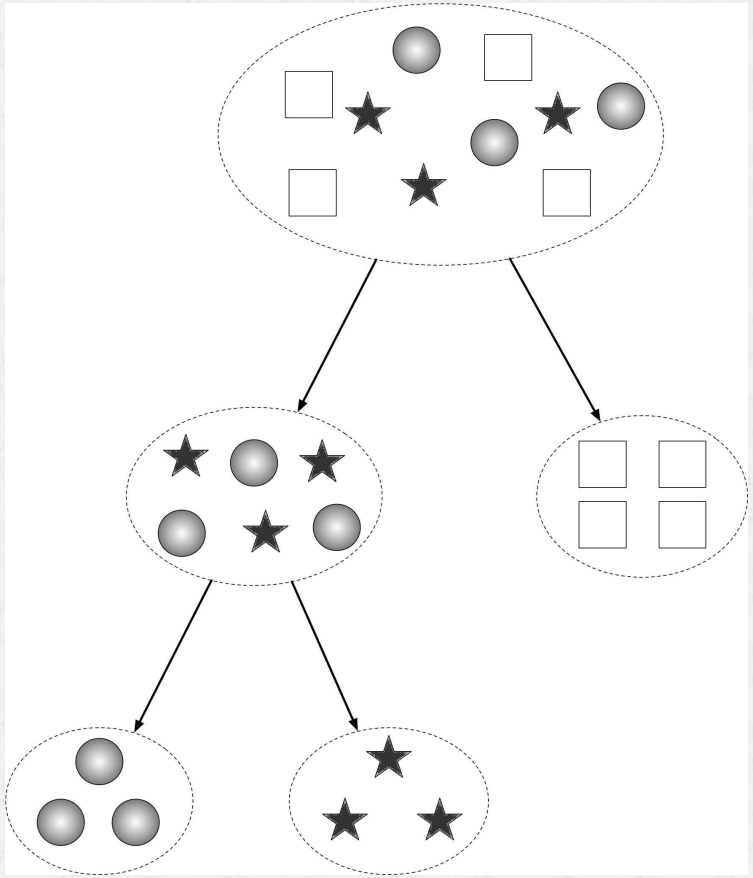
- 一种迭代式算法，每次迭代会优化上一次聚类结果，步骤如下：
  - 选取k个点作为k个簇的初始质心；
  - 将所有点分别分配给最近的质心所在的簇；
  - 重新计算每个簇的质心；
  - 重复步骤2和3直到质心不再发生变化。



- 无法保证收敛到全局最优，但能够有效地收敛到一个局部最优点

重复二分聚类算法

- k 均值算法的加强版，其中的“二分”是反复对子集进行二分：
  - 挑选一个簇进行划分-挑选簇划分的标准；
  - 利用k均值算法将该簇划分为2个子集；
  - 重复上述2个步骤至产生足够数量的簇。

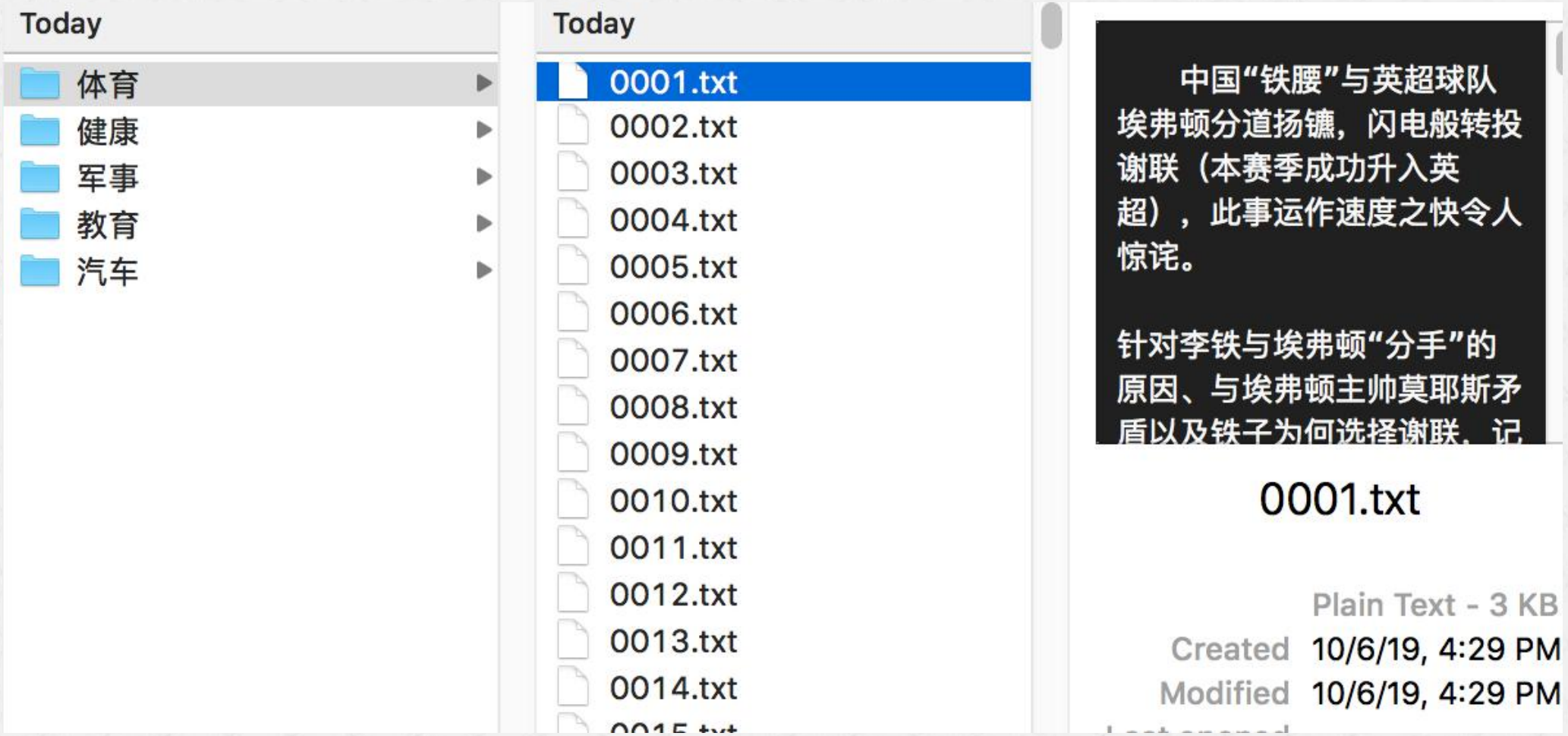


概念及任务

- **文本分类**（text classification），又称**文档分类**（document classification），指的是将一个文档归类到一个或多个类别中的自然语言处理任务，监督学习任务
  - 文本的**类别**（category或class）有时又称**标签**（label），所有类别组成一个标注集
  - 垃圾邮件过滤、垃圾评论过滤、社交媒体自动标签推荐 、情感分析

文本分类

语料库-标注相对简单



特征提取

- **与文本聚类相同**
  - 词袋向量-词频、TF-IDF
  - 一般以词语为基本单位，需要先进行分词
  - 有时将相邻两个字符构成的所有二元语法作为“词”，分类的准确率会更高。[2016 清华大学-THUCTC : An Efficient Chinese Text Classifier]
  - 需要对特征进行选择过滤

卡方特征选择

- 许多常用单词对分类决策的帮助不大
  - 比如汉语的虚词“的”和标点符号等
  - 也可能有一些单词在所有类别的文档中均匀出现
- 为了消除这些单词的影响
  - 可以采用停用词表
  - 可以用**卡方非参数检验**（χ^2）来过滤掉与类别相关程度不高的词语

卡方特征选择

- 统计学中，χ^2检验常用于检验两个事件的独立性（两个事件独立则两者同时发生的概率为$P(AB)=P(A)P(B)$），
- 计算词语“高兴”的χ^2值 - 事件：词的出现、类别的出现

	在“正面”文档中	在“负面”文档中
词语“高兴”	频次= 49	频次= 27 652
其他词语	频次= 141	频次= 774 106

卡方特征选择

- 随机取一个词语，将该词语是否为t记作e_t；
- 随机取一个文档，其类别是否为c记作e_c

	$e_c = e_{\text{正面}} = 1$	$e_c = e_{\text{正面}} = 0$
$e_t = e_{\text{高兴}} = 1$	$N_{11} = 49$	$N_{10} = 27\,652$
$e_t = e_{\text{高兴}} = 0$	$N_{01} = 141$	$N_{00} = 774\,106$

卡方特征选择

- 计算e_t与e_c两个事件同时成立与否的4种组合（即$E_{11}, E_{10}, E_{01}, E_{00}$）的期望，以$E_{11}$（词语为“高兴”且文档为“正面”的期望）为例，其计算方法如下

$$
\begin{aligned}
E_{11} &= N \times P(t) \times P(c) \\
&= N \times \frac{N_{11} + N_{10}}{N} \times \frac{N_{11} + N_{01}}{N} \\
&= N \times \frac{49 + 27652}{N} \times \frac{49 + 141}{N} \\
&\approx 6.6
\end{aligned}
$$

N 为所有词语词频之和

$P(t)$为词出现的概率

$P(c)$为文档类别出现的概率

卡方特征选择

- 同理，四个事件的期望

	$e_c = e_{正面} = 1$	$e_c = e_{正面} = 0$
$e_t = e_{高兴} = 1$	$E_{11} = 6.6$	$E_{10} = 27\,694.4$
$e_t = e_{高兴} = 0$	$E_{01} = 183.4$	$E_{00} = 774\,063.6$

- 带入卡方的计算公示

$$\chi^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

卡方特征选择

- 得到词语“高兴”与类别“正面”的卡方检验值为$\chi^2(\mathbb{D}, t = \text{高兴}, c = \text{正面}) \approx 284$，代表的含义可以查表进行判别

p	χ^2临界值
0.1	2.71
0.05	3.84
0.01	6.63
0.005	7.88
0.001	10.83

χ^2值大于6.63时，
两者独立的置信度小于0.01

当C有多个类别时，取最大的卡方值作为特征的最终卡方值

词袋向量表示

- 将词袋向量记作$\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$，向量的第$i$维记作$x_i$。将类别记作$y \in \mathcal{Y} = \{c_1, c_2, \cdots, c_K\}$，其中$K$为类别总数。则语料库（训练数据集）$T$可以表示为词袋向量$\boldsymbol{x}$和类别$y$所构成的二元组的集合：

$$T = \{(\boldsymbol{x}^{(1)}, y_1), (\boldsymbol{x}^{(2)}, y_2), \cdots, (\boldsymbol{x}^{(N)}, y_N)\}$$

- 如以词语作为特征则n约在10万量级，以字符二元语法作为特征则n约在50万量级。**利用卡方特征选择可以减少到10%~20%。**

朴素贝叶斯分类器-最简单的生成式模型

- 通过训练集学习联合概率分布$p(X,Y)$
- 由贝叶斯定理，将联合概率转换为先验概率分布与条件概率分布之积，利用特征条件独立简化计算：

$$p(X=\boldsymbol{x},Y=c_k)=p(Y=c_k)p(X=\boldsymbol{x}|Y=c_k)$$

- 类别的先验概率分布$p(Y=c_k)$很容易估计，通过统计每个类别下有多少样本即可（极大似然），即：

$$p(Y=c_k)=\frac{\text{count}(Y=c_k)}{N}$$

朴素贝叶斯分类器

- 而 $p(X = \boldsymbol{x}|Y = c_k)$则难以估计，因为$\boldsymbol{x}$的量级非常大

$$p(X = \boldsymbol{x}|Y = c_k) = p(X_1 = x_1, \cdots, X_n = x_n|Y = c_k), k = 1,2, \cdots, K$$

- 假设第i维x_i有m_i种取值，那么组合起来$\boldsymbol{x}$一共有$\prod_i^n m_i$种

朴素贝叶斯分类器

- 朴素贝叶斯法“朴素”地假设了所有特征是条件独立的，该条件独立性假设为：

$$p(X=\boldsymbol{x}|Y=c_k) = p(X_1=x_1,\cdots,X_n=x_n|Y=c_k)$$
$$= \prod_{i=1}^{n} p(X_i=x_i|Y=c_k)$$

- 于是，又可以利用极大似然来进行估计：

$$p(X_i=x_i|Y=c_k) = \frac{\text{count}(X_i=x_i, y_i=c_k)}{\text{count}(y_i=c_k)}$$

朴素贝叶斯分类器

- 在预测时，朴素贝叶斯法依然利用贝叶斯公式找出后验概率$p(Y = c_k|X = \boldsymbol{x})$最大的类别$c_k$作为输出$y$

$$y = \arg\max_{c_k} p(Y = c_k|X = \boldsymbol{x})$$

- 将贝叶斯公式代入上式得到：

$$y = \arg\max_{c_k} \frac{p(X = \boldsymbol{x}|Y = c_k)p(Y = c_k)}{p(X = \boldsymbol{x})}$$

朴素贝叶斯分类器

- 由于分母 $p(X = \boldsymbol{x})$与c_k无关，求最大后验概率时可以省略，亦即：

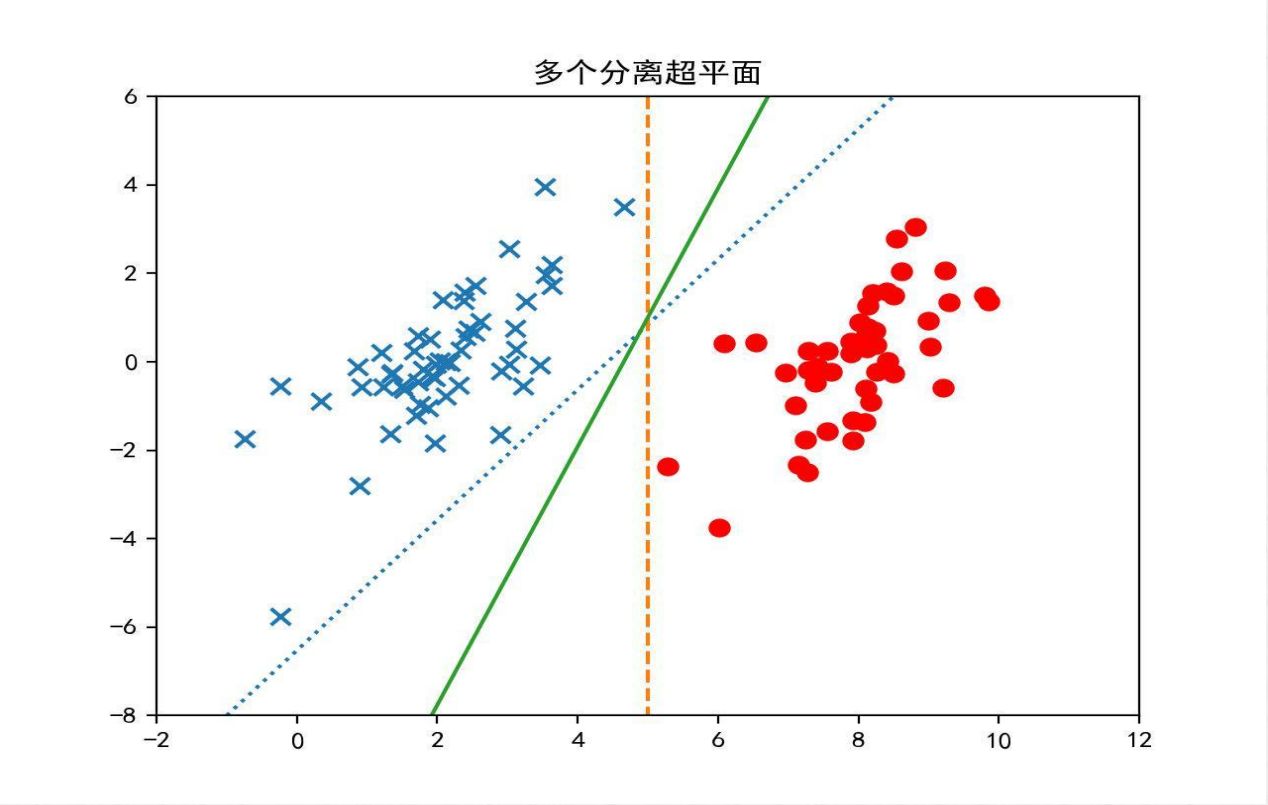
$$y = \arg\max_{c_k} p(X = \boldsymbol{x}|Y = c_k)\, p(Y = c_k)$$

- 然后将独立性假设公式代入上式，得到最终的分类预测函数：

$$y = \arg\max_{c_k} p(Y = c_k) \prod_{i=0}^{n} p(X_i = x_i|Y = c_k)$$

支持向量机分类器

- **支持向量机**（Support Vector Machine，SVM）是一种二分类模型，其学习策略在于如何找出一个决策边界，使得边界到正负样本的最小距离都最远。



传统文本表示

- **语料**
  - 我们都生活在阴沟里，但仍有人仰望星空。
  - 每个圣人都有过去，每个罪人都有未来。
- **分词**
  - [我们，都，生活，在，阴沟，里，但，仍有，人，仰望，星空]
  - [每个，圣人，都有，过去，每个，罪人，都有，未来]

ont-hot表示

- **索引编码**
  - {我们:0, 都:1, 生活:2, 在:3, 阴沟:4, 里:5, 但:6, 仍有:7,人:8,仰望:9, 星空:10, 每 个:11, 圣人:12, 都有:13, 过去:14, 罪人:15, 未来:16}
- **分词**
  - 我们: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  - 过去: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
  - 未来: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

文本表示

ont-hot表示-词袋

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
每个	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
圣人	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
都有	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
过去	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
每个	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
罪人	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
都有	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
未来	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
count	0	0	0	0	0	0	0	0	0	0	0	2	1	2	1	1	1

N-gram表示

- **Bi-gram 索引编码**
  - {我们 都:0, 都 生活:1, 生活 在:2, 在 阴沟:3, 阴沟 里:4, 里 但:5, 但 仍有:6, 仍有 人:7, 人 仰望:8, 仰望 星空:9, 每个 圣人:10, 圣人 都有:11, 都有 过去:12, 过去 每 个:13, 每个 罪人:14, 罪人 都有:15, 都有 未来:16}
  - 每个圣人都有过去，每个罪人都有未来。
  - [0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1]

TF-IDF表示

- 既考虑词频，也考虑词语的稀有程度

$$\begin{aligned}\text{TF - IDF}(t,d) &= \frac{\text{TF}(t,d)}{\text{DF}(t)} \\ &= \text{TF}(t,d)\cdot\text{IDF}(t)\end{aligned}$$

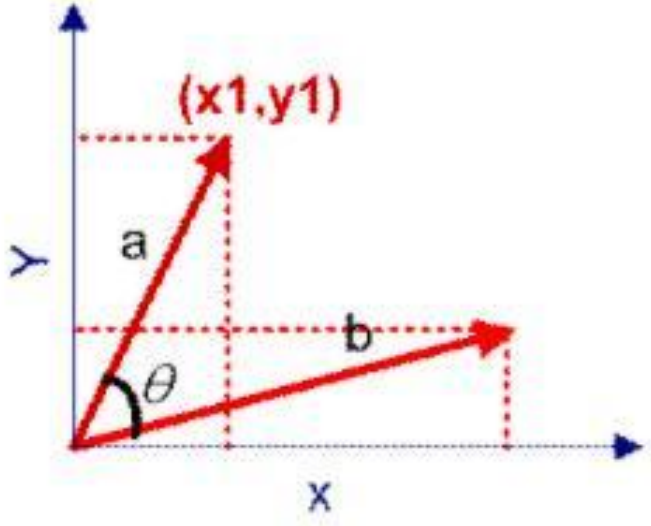
- 词袋模型+TF-IDF表示，词频替换为TF-IDF值

传统文本表示问题

- 传统文本表示方法简单易理解
- 传统文本表示方法表示效果差
  - 数据稀疏
  - 相似词语的表示向量可能相似度较低

余弦相似度-文本向量化表示

- 余弦相似度，又称为余弦相似性，是通过计算两个向量的夹角余弦值来评估他们的相似度。余弦相似度将向量根据坐标值，绘制到向量空间中，如最常见的二维空间。



$c^2 = a^2 + b^2 - 2ab\cos\theta$

$a^2 = x1^2 + y1^2$

$b^2 = x2^2 + y2^2$

$c^2 = (x1 - x2)^2 + (y1 - y2)^2$

那么cosθ就可以算出来了

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

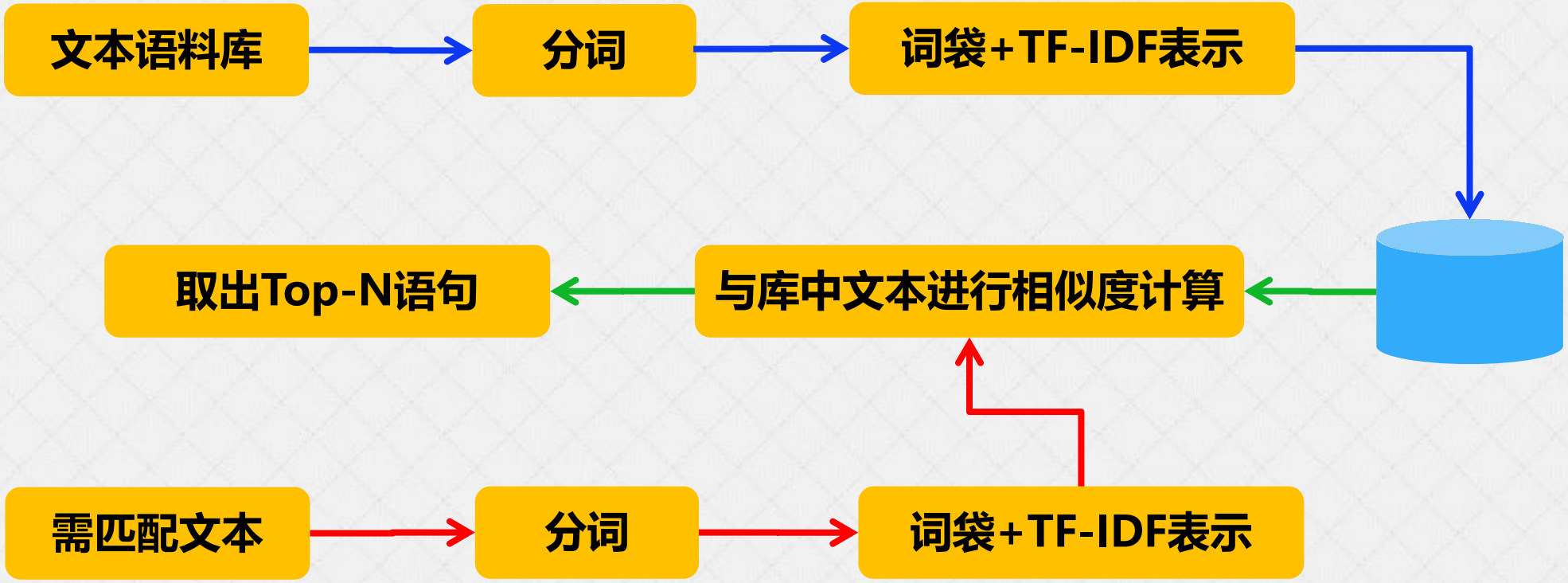
余弦相似度

- 把这个概念推广到多维-还有许多**其他相似度计算方法**

$$\cos\theta = \frac{\sum_1^n (A_i \times B_i)}{\sqrt{\sum_1^n A_i^2} \times \sqrt{\sum_1^n B_i^2}}$$

- 余弦值的范围在[-1,1]之间，值越趋近于1，代表两个向量的方向越接近；越趋近于-1，他们的方向越相反；接近于0，表示两个向量近乎于正交。

文本相似度匹配流程


文本语料库
分词
词袋+TF-IDF表示
取出Top-N语句
与库中文本进行相似度计算
需匹配文本
分词
词袋+TF-IDF表示