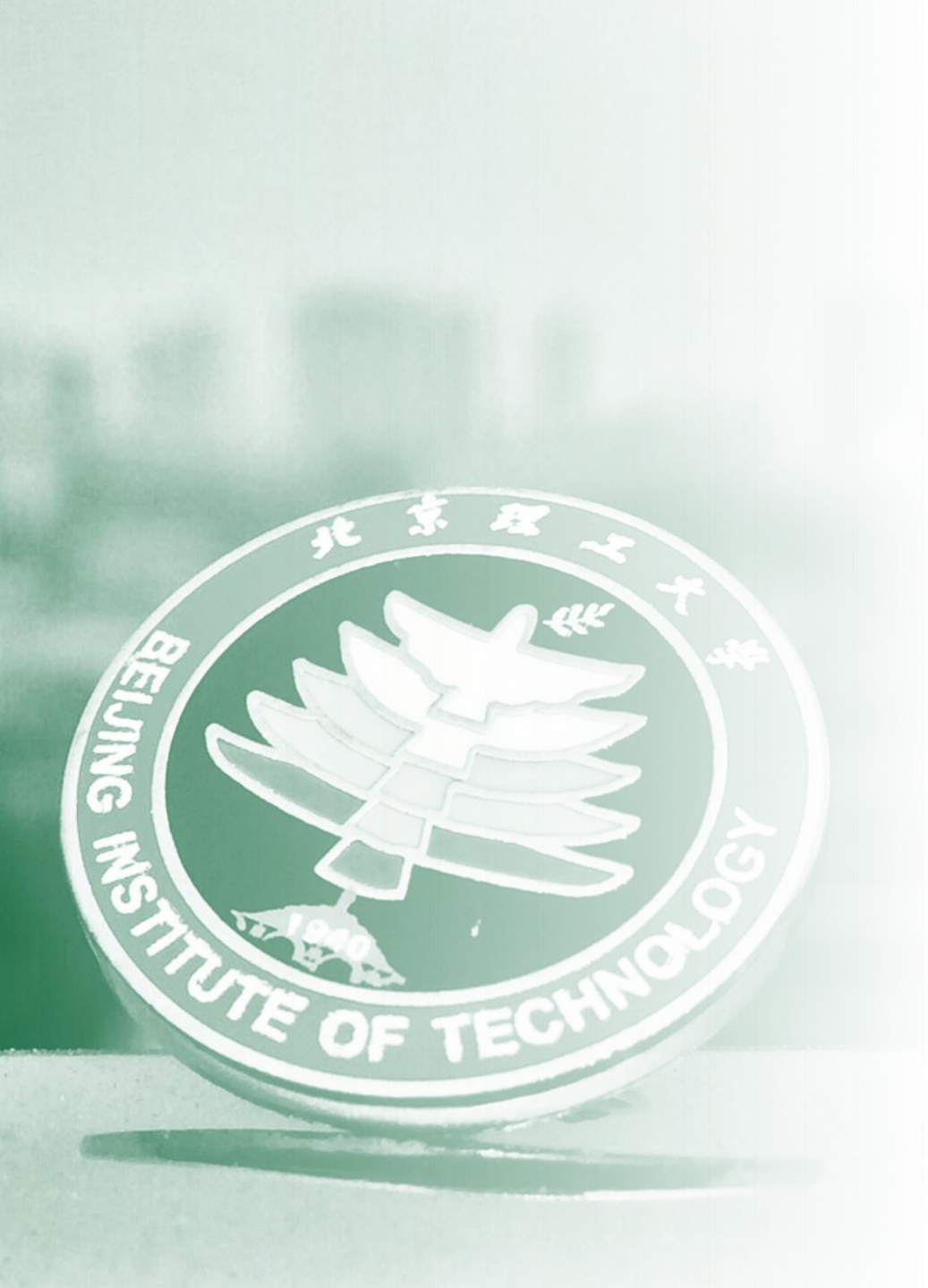
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

文本聚类

答辩人：钟天声，丛颖，王啸天，杨毅辰，戴铭瑞

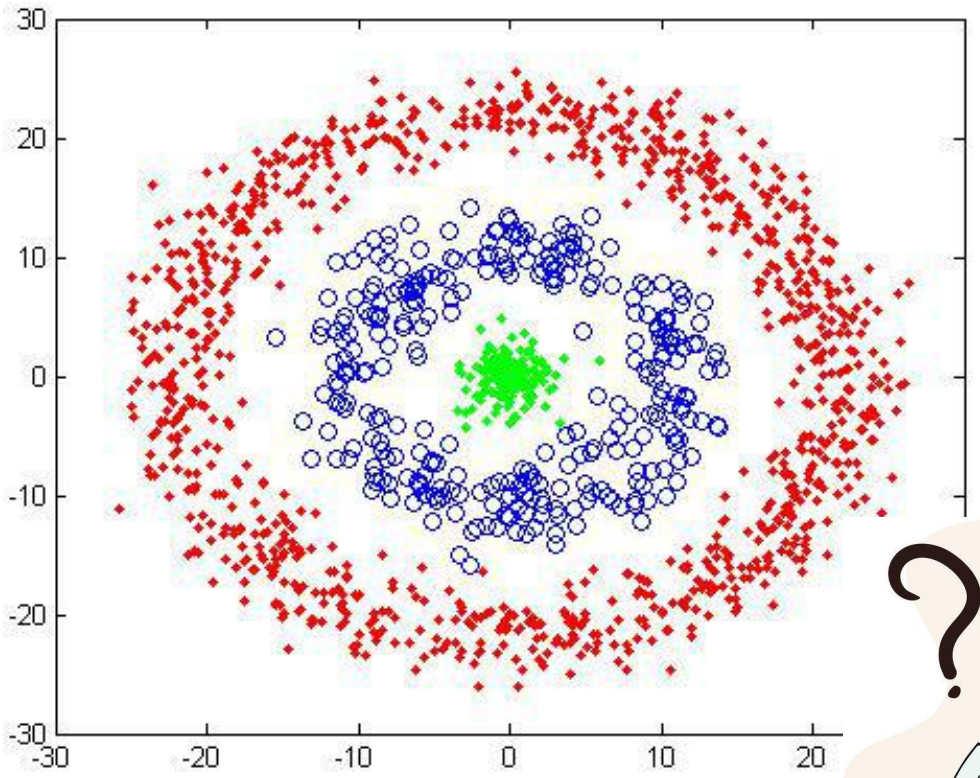


目录 | CONTENTS

1 文本聚类简介

聚类的概念

聚类是一种流行的无监督学习方法，其任务是将一组数据分为若干个类别，使同一个类别的数据之间的差异更小而不同类别间的数据差异更大。如今聚类已经在各个领域受到广泛应用。



文本聚类

文本聚类即将一系列文本作为聚类的数据，把这些文本分为几类，同一类中的文本相似度较大而不同类别的文本间相似度则较小。

例：同为武侠小说的文章相似度较高，而计算机专业书籍和武侠小说的相似度就比较低。

文本聚类

文本聚类是一种无监督学习，机器不知道也不需要每个数据的真实标签，只是凭借机器对“簇”的理解来对数据进行聚类。

VS

文本分类

文本分类是一种监督式学习，机器针对有标注的数据训练一个分类器，对未标注的文本进行分类。

- Newsblaster是哥伦比亚大学开发的一个多文档文摘系统，其功能是将每天发生的重要新闻文本进行聚类，对其中同主题的文本进行冗余消除、信息融合、文本生成等处理，从而生成一个简洁的摘要文档。

Columbia Newsblaster

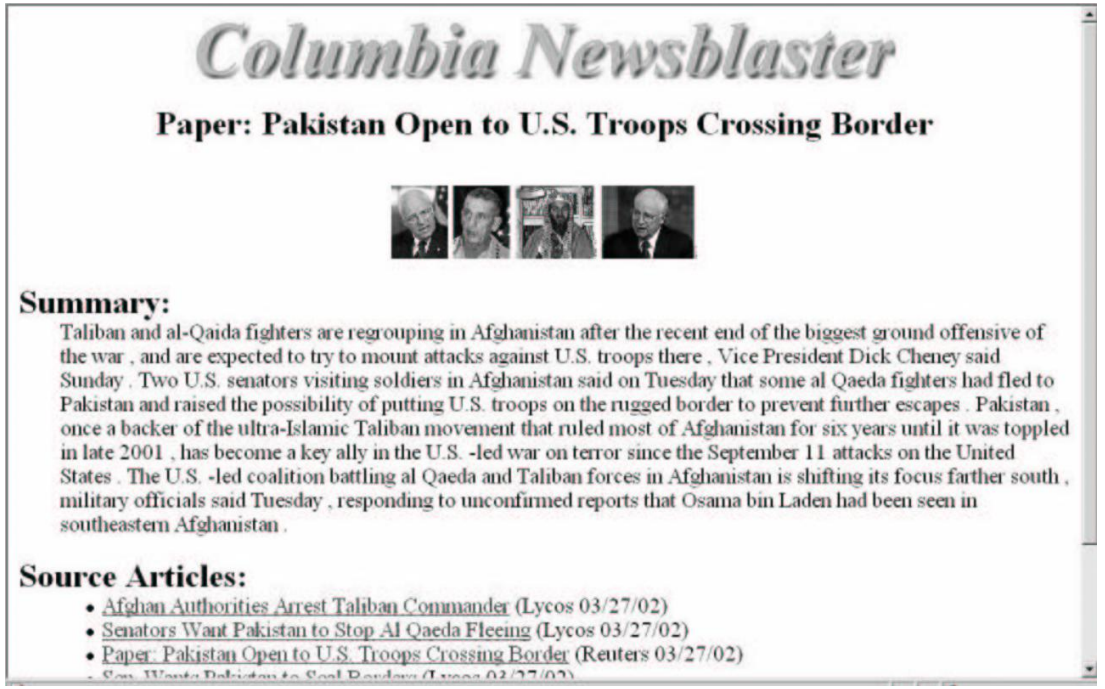
New Mexico Forest Fire Mostly Contained

Summary:

Gov. Gary Johnson said the New Mexico blaze started when a resident dumped fireplace ash in a back yard mistakenly thinking the ashes were cold. The fire forced the evacuation of about 1,300 people. A fire was nearly contained Monday, and residents who had been evacuated will be returning home, a National Park Service spokesman said. No one has been injured. The Lincoln County assessor's office pegged property damage from the fire at $5.2 million in assessed valuation. Last year, 1,649 fires burned 38,890 acres.

Source Articles:

- Ashes Said to Have Caused NM Wildfire (Lycos 03/26/02)
- New Mexico Forest Fire Mostly Contained (FOX News 03/26/02)
- Wired News (Wired 03/26/02)
- Fires Destroy 36 Homes in New Mexico (Lycos 03/26/02)
- New Mexico fire nearly contained (CNN 03/26/02)
- Firefighters tame New Mexico blaze (CNN 03/26/02)
- Winds Swell Blaze Into an Inferno (LA Times 03/26/02)



Columbia Newsblaster

Paper: Pakistan Open to U.S. Troops Crossing Border

Summary:

Taliban and al-Qaida fighters are regrouping in Afghanistan after the recent end of the biggest ground offensive of the war , and are expected to try to mount attacks against U.S. troops there , Vice President Dick Cheney said Sunday . Two U.S. senators visiting soldiers in Afghanistan said on Tuesday that some al Qaeda fighters had fled to Pakistan and raised the possibility of putting U.S. troops on the rugged border to prevent further escapes . Pakistan , once a backer of the ultra-Islamic Taliban movement that ruled most of Afghanistan for six years until it was toppled in late 2001 , has become a key ally in the U.S. -led war on terror since the September 11 attacks on the United States . The U.S. -led coalition battling al Qaeda and Taliban forces in Afghanistan is shifting its focus farther south , military officials said Tuesday , responding to unconfirmed reports that Osama bin Laden had been seen in southeastern Afghanistan .

Source Articles:

- Afghan Authorities Arrest Taliban Commander (Lycos 03/27/02)
- Senators Want Pakistan to Stop Al Qaeda Fleeing (Lycos 03/27/02)
- Paper: Pakistan Open to U.S. Troops Crossing Border (Reuters 03/27/02)

- IBM Watson Explorer和infonetware这两个搜索引擎允许用户输入检索关键词，然后对检索到的文档进行聚类，输出各个类别的简介，这样可以帮助用户缩小检索范围。

- 用户的兴趣模式挖掘：将聚类算法用于对用户感兴趣的文档进行聚类，从而发现用户的兴趣模式，有助于定制化推荐。
- 改善文本分类的结果：Fang Y C, Parthasarathy S等人使用PDDP（Principle Direction Divisive Partitioning，一种基于主成分分析的文本聚类方法）改善了文本分类的结果。

1 文本预处理

文本预处理步骤将对文本进行分词和去停用词等处理。

2 文本表征

文本表征指从文本中提取出特征，通过这些特征来表示相应的文本。

3 聚类算法

将上一步得到的用特征表示的文本送入聚类算法进行聚类，得到最终的文本聚类结果。

2 文本预处理与文本表征

文本预处理

分词（Tokenization）

- 即将 输入的文本（sequence）转换为token 的过程。
- 英文：比较容易实现分词。
- 中文：使用一些分词算法或工具，例如：Jieba分词，百度分词API，阿里云的分词API等。

去停用词

- 停用词在这里通常指文本中出现频率很高，但对语义理解帮助不大的词。
- 语气助词、副词、介词、连词等。如中文常见的“的”、“在”、“和”、“接着”之类。英文常见的“of”，“an”
- 停用词表

词性归一化（tokenization normalization）

- 英文文本处理中特有
- 将相同含义但是不同形式的词转换成同一个词（token）
- 词干提取（stemming）

  beautiful --> beauty
- 词性还原（lemmatization）

例如：good, better 和 best 词形还原的结果是 good, good 和 good.

文本表示对文本进行数学建模，保留语义等需要提取的信息，是后续工作的基础

文本表示模型

- 离散表示

  1.One-hot编码　2.词袋模型　3.N-gram

- 分布式表示

  经典模型是word2vec，还包括后来的Glove、ELMO、GPT和最近很火的BERT

One-hot编码

one-hot编码就时把每个词表示为一个长向量。这个向量的维度是词表大小，向量中只有一个维度的值为1，其余维度为0，这个维度就代表了当前的词。一段文本就应该是多个one-hot组合成的矩阵了。

文本：John likes to watch movies. Mary likes too.

构造词典：
{"John": 1, "likes": 2, "to": 3, "watch": 4, "movies": 5, "also":6, "football": 7, "games": 8, "Mary": 9, "too": 10}

["John", "likes", "to", "watch", "movies", "also", "football","games", "Mary", "too “]

One-hot表示：
John: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
likes: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
too : [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

优点：简洁明了，计算快速。

缺点：无法表述词与词之间的语义关系，同时因为词表太大容易造成维数灾难等。

具体想法是：将每个文档看成一袋子词，忽略每个词出现的顺序，通过统计各个词出现的次数来表述一篇文档，文档的向量表示可以直接将各词的词向量表示加和。

文本：John likes to watch movies. Mary likes too.
==> [1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

优点：这是一个基于统计，能够很快得到文本的表示。

缺点：忽略了文本字词之间的顺序，容易造成信息的丢失or混淆，如“我爱你”和“你爱我”都是同一种表示方法，但他们之间的意思确实不一样的。

2-gram建索引：
"John likes”：1,
"likes to”：2,
"to watch”：3,
"watch movies”：4,
"Mary likes”：5,
"likes too”：6,
"John also”：7,
"also likes”：8,
“watch football”：9,
"football games": 10,

与词袋模型相比，将连续出现的n个词所构成的词组（N-gram）也作为一个单独的特征放到向量表示中去。这个模型认为，第N个词的出现只与前面N-1个词相关，而与其它任何词都不相关。

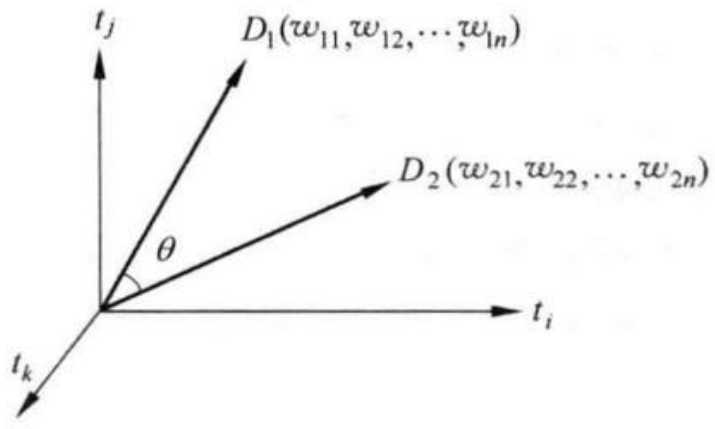
文本：John likes to watch movies. Mary likes too.

[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

优点：考虑了词的顺序。

缺点：n越大 数量级越大。

前面的三种表示都是默认每个字词的重要程度是一样的，而在一段文本当中有的词时关键词而有的词就不是很重要，如‘的’、‘吧’、‘啊’等语气词就不是很重要，因此TF-IDF考虑了某个词在文本当中的重要程度，计算公式如下：



$$TF-IDF(t,d)=TF(t,d)*IDF(t)$$

TF：词频，单词t在文档d中出现的频率，反映了单词t对文档的重要性。

IDF：倒文档频度，衡量单词t区别于其他文档的程度。

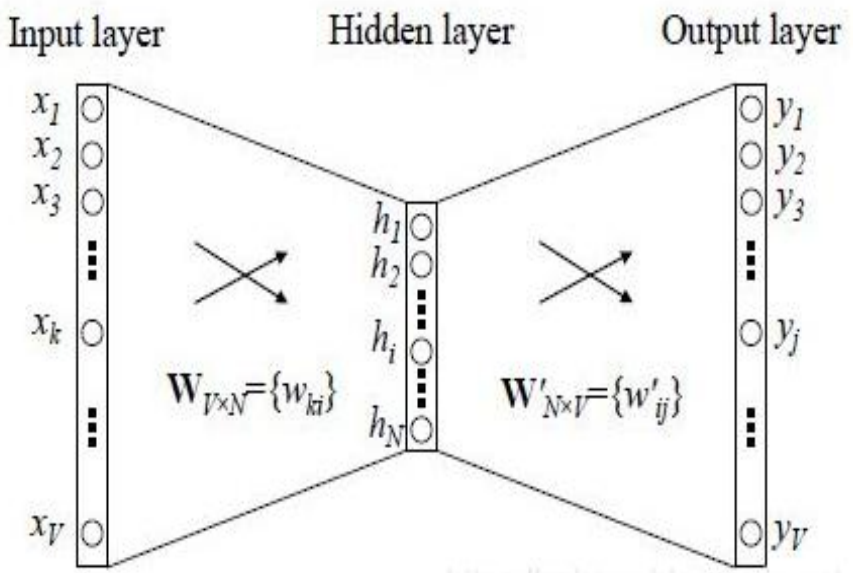
$$IDF(t)=log\frac{\text{文章总数}}{\text{包含单词的文章总数}+1}$$

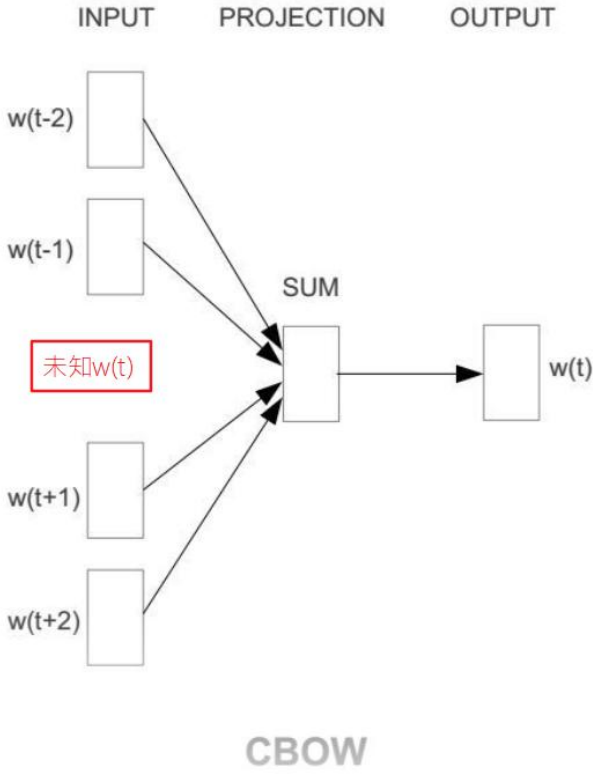
1. CBOW模型

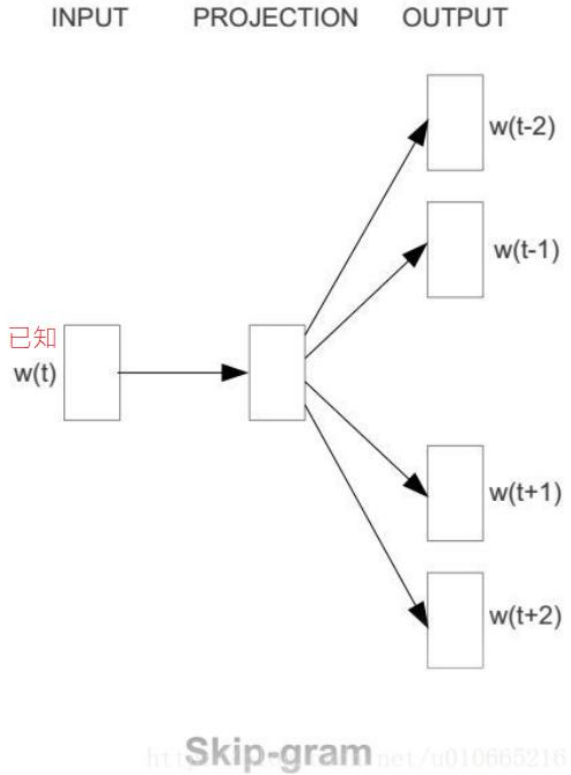
根据中心词W(t)周围的词来预测中心词

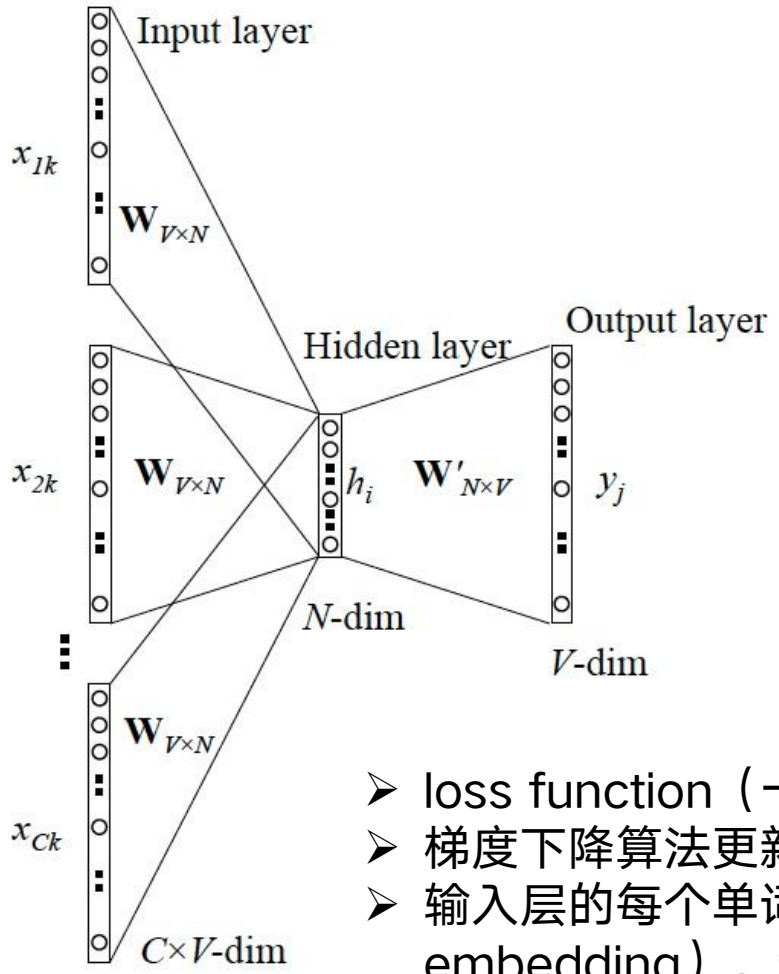
2. Skip-gram模型

则根据中心词W(t)来预测周围词



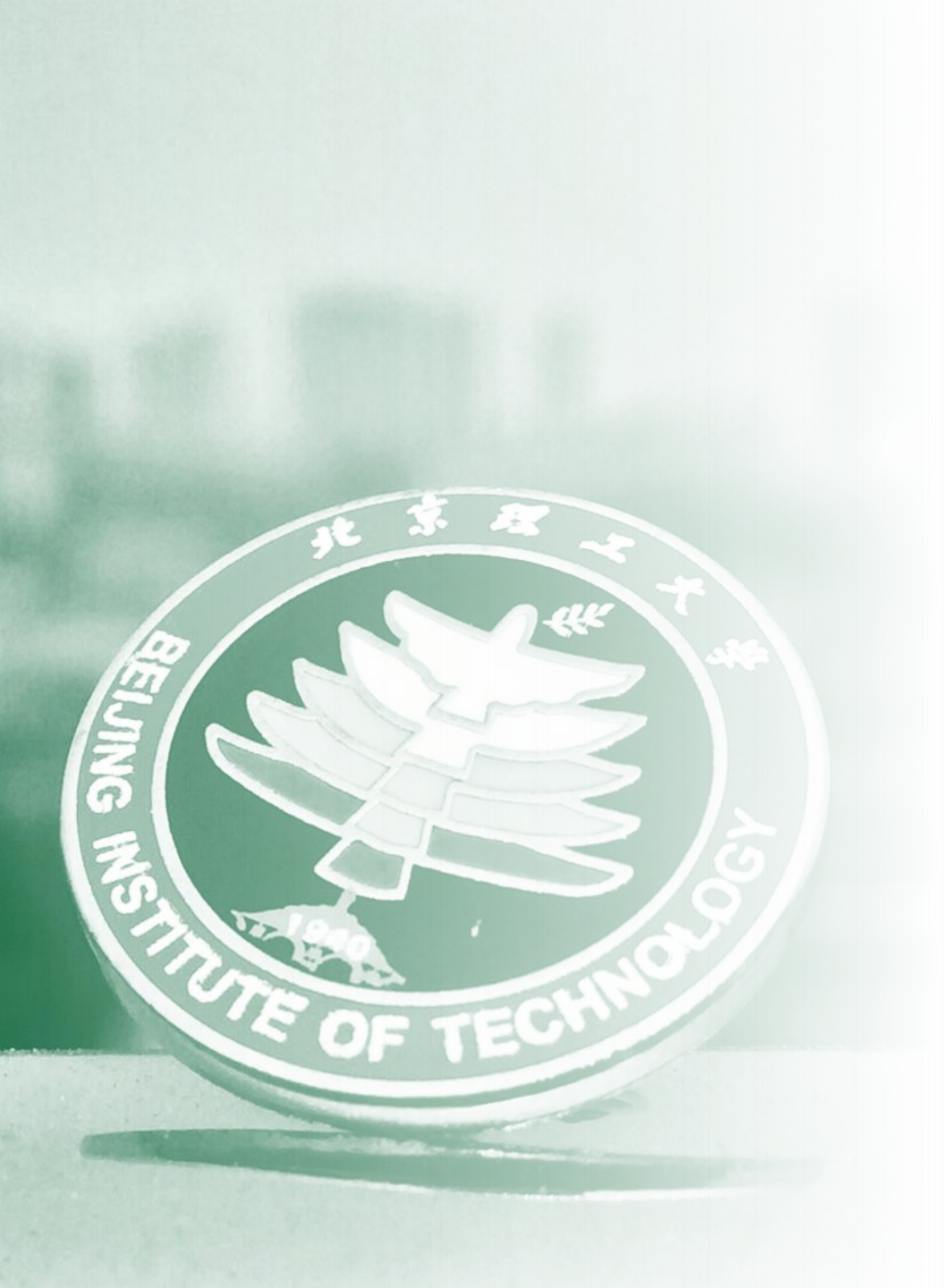






1 .输入层：上下文单词的onehot. {假设单词向量空间dim为V，上下文单词个数为C}
2. 所有onehot分别乘以共享的输入权重矩阵W. {V*N矩阵，N为自己设定的数，初始化权重矩阵W}
3. 所得的向量 相加求平均作为隐层向量, size为1*N.
4. 乘以输出权重矩阵W' {N*V}
5 .得到向量 {1*V}softmax处理得到V-dim概率分布 6 概率最大的index所指示的单词为预测出的中间词（target word）与true label的onehot做比较，误差越小越好
6.根据误差更新权重矩阵

- loss function（一般为交叉熵代价函数）
- 梯度下降算法更新W和W'
- 输入层的每个单词与矩阵W相乘得到的向量的就是我们想要的词向量（word embedding），矩阵W自身也叫做look up table，免去训练过程直接查表得到单词的词向量了。



3 聚类算法介绍

1. 聚类算法技术
2. 划分算法
3. 层次算法
4. 密度算法
5. 网格算法

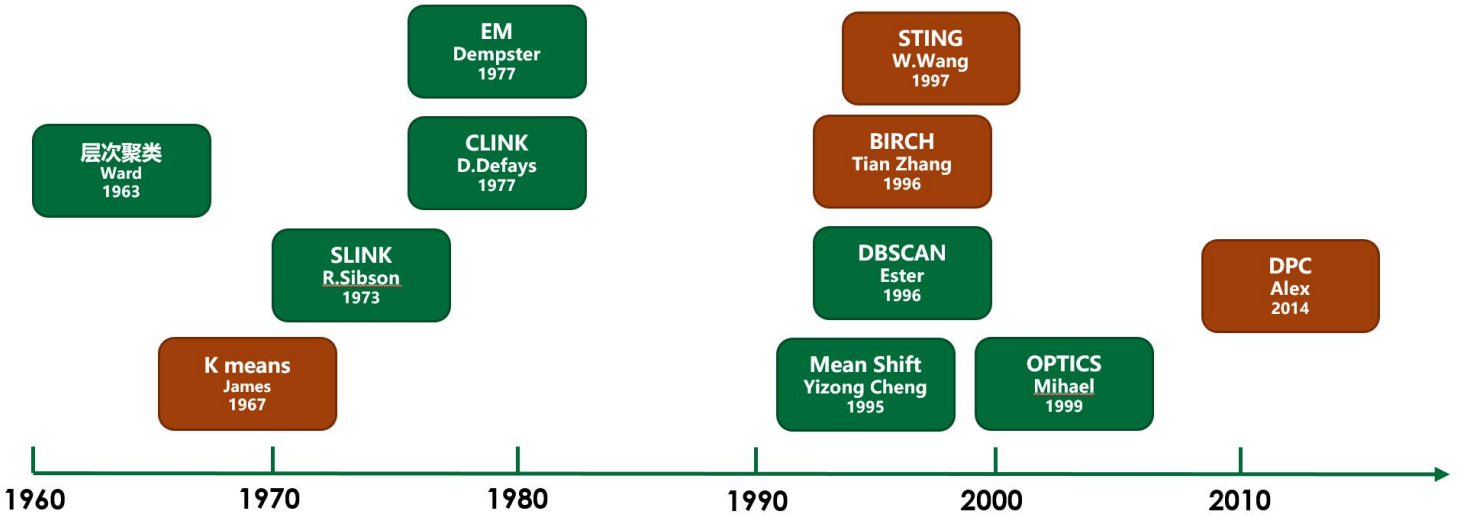
什么是聚类分析？

- **聚类分析**，是一个把数据对象(或观测)划分成**子集**的过程。每个子集是一个**簇**，使得簇中的对象**彼此相似**，但**与其他簇中的对象不相似**。由这个过程产生的一系列簇成为一个聚类

- 可以简单概括为：**同类相同、异类相异**

聚类算法发展

- **聚类算法**的历史与有监督学习一样悠久。层次聚类算法出现于1963年，这是非常符合人的直观思维的算法，现在还在使用。



- **聚类算法**是最早被用于**模式识别**及**数据挖掘任务**的方法之一，并且被用来研究各种应用中的大数据库，因此**如今**用于**大数据**的聚类算法受到越来越多的关注

为什么需要聚类算法技术?

常见的机器学习任务**大部分都是带有标记的**(有监督的)，在真实的世界中，有人工标记的数据仅仅占很少的一部分，对于大数据时代而言甚至可以**忽略不计**，可以说无监督（不带标记)的学习才是机器学习的终极目标。

聚类算法技术目的?

探究那些无法或者还未标记的数据的**内在数据结构或规律**，如文本聚类问题。

如何进行聚类?

聚类分析所有的工作都关注在这两者上, **划分过程**与**对象之间相似与否**。

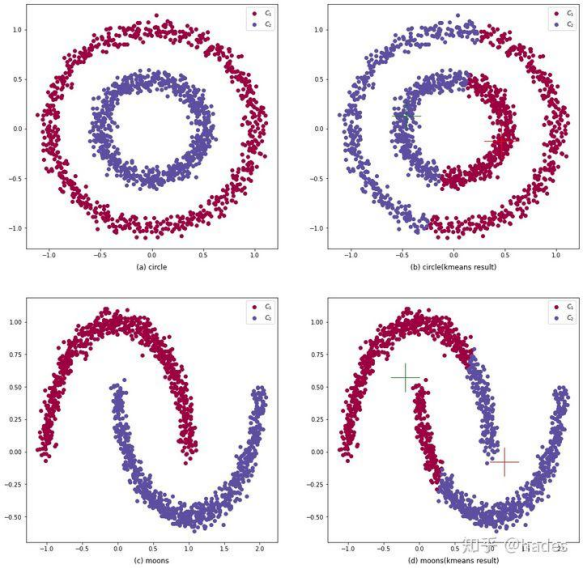
聚类过程一般的有：**基于划分的**、**基于层次的**、**基于密度的**以及**基于网格的**聚类算法

基于划分的聚类算法

给定一个有N个元组或者纪录的数据集，分裂法将构造**K个分组**，每一个分组就代表一个聚类，然后采用一种**迭代**的重定位技术，尝试通过对象在划分间移动来改进划分。

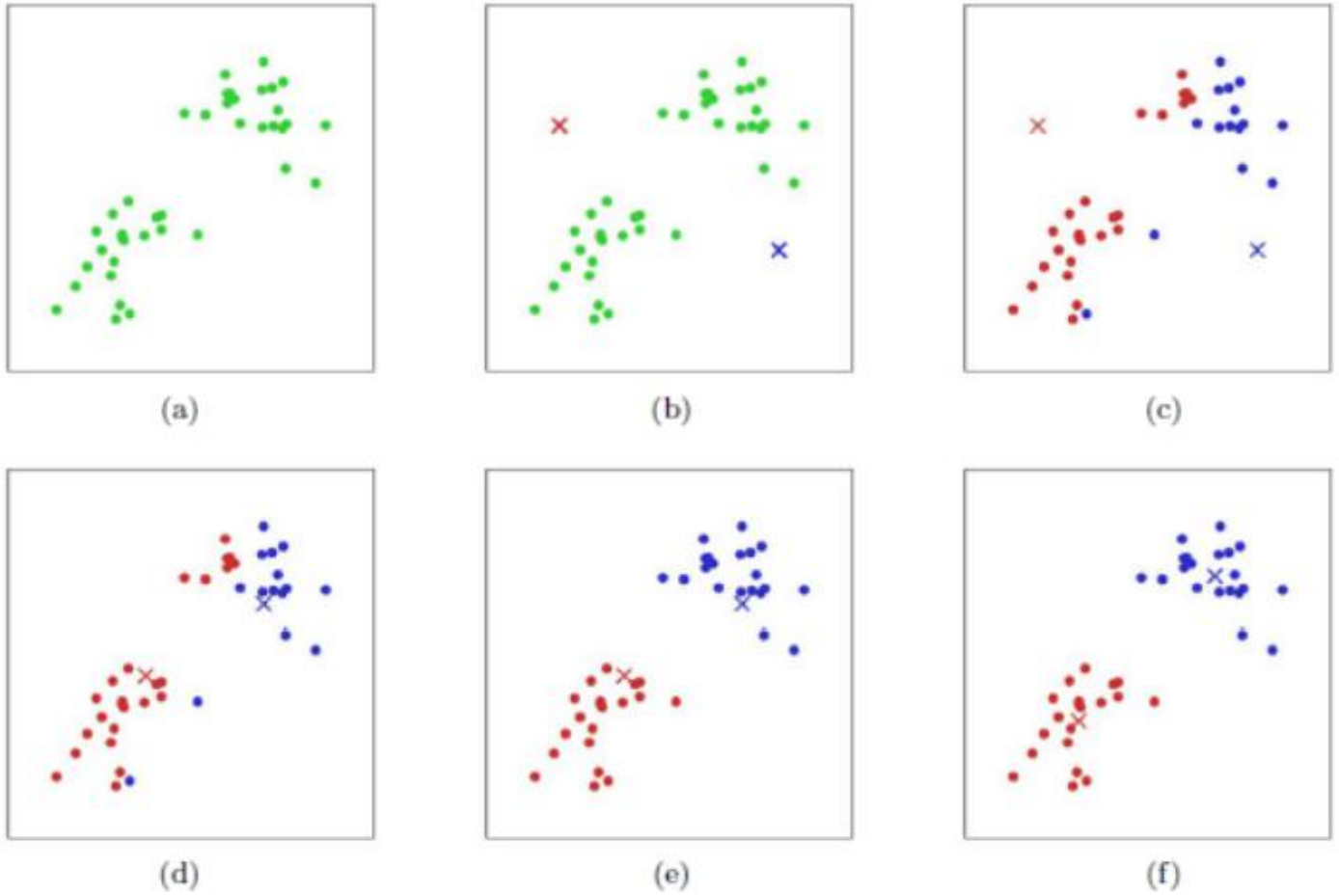
常见算法

- **K-MEANS算法-k均值聚类算法**
- K-MEDOIDS算法
- CLARANS算法



K-Means算法

- 也称为**K-平均**或者**K-均值**，一般作为掌握聚类算法的第一个算法。
- 这里的K为常数，**需事先设定**，通俗地说该算法是将没有标注的 M 个样本通过迭代的方式聚集成K个簇。



K-Means算法

1. K=2，随机两点对应的类别质心
2. 求样本距离第一轮分类
3. 从两类样本点中计算出新聚类中心
4. 重复迭代
5. 标准测度函数收敛

K-Means算法流程

1. 首先从n个 数据对象任意选择 **k 个对象**作为初始聚类中心；
2. 对于所剩下其它对象，则根据它们与这些聚类中心的**相似度**（距离），分别将它们分配给与其最相似的（聚类中心所代表的）聚类；
3. 再计算每个所获新聚类的聚类中心（该聚类中所有对象的均值）
4. 不断重复这一过程直到**标准测度函数开始收敛**为止。

$$J = \sum_{i=1}^{K} J_i = \sum_{i=1}^{K} \sum_{j=1}^{N} w_{ji} \left\| X_j - C_i \right\|^2$$

优点

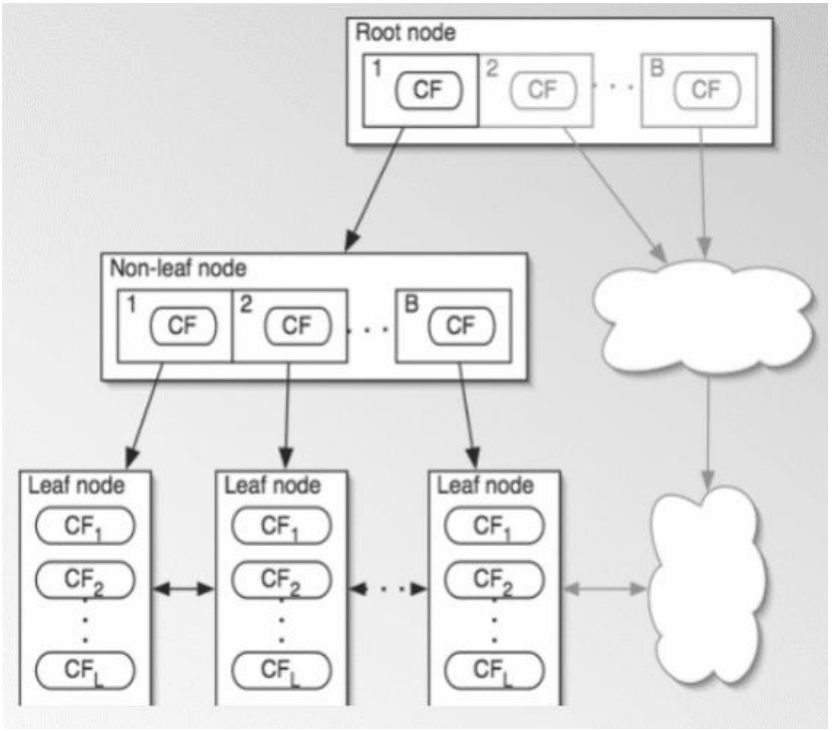
1. 原理比较简单，实现也是很容易，收敛速度快。
2. 聚类效果较优。
3. 算法的可解释度比较强。
4. 主要需要调参的参数仅仅是簇数k

缺点

1. K值的选取不好把握
2. 如果各类别的数据不平衡，则可能聚类效果不佳。
3. 采用迭代方法，得到的结果只是局部最优。
4. 对噪音和异常点比较的敏感。

基于层次的聚类算法

基于层次的聚类方法是指对给定的数据进行**层次分解**，直到满足某种条件为止。该算法根据层次分解的顺序分为自底向上法和自顶向下法，即**凝聚式层次聚类算法**和**分裂式层次聚类算法**。



常见算法

- BIRCH算法➡利用层次方法的平衡迭代规约和聚类
- CURE算法
- CHAMELEON算法

BIRCH算法

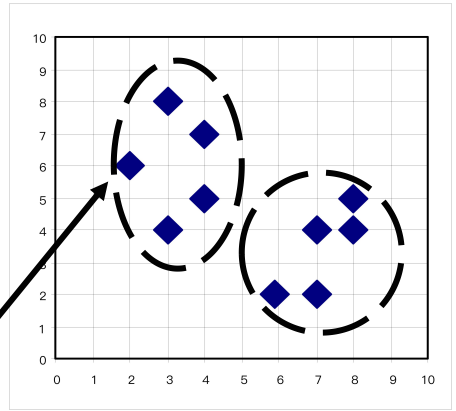
- **核心是聚类特征树**Clustering Feature Tree简称CF Tree
- 每个节点包括叶子节点都有若干个CF
- 叶子结点由双向链表连接

BIRCH算法-CF

- 聚类特征:

  CF = (N , LS , SS)

- N: CF中拥有的样本点的数量
- LS: CF中拥有的样本点各特征维度的和向量 $\sum_{i=1}^{N} X_i$
- SS: CF中拥有的样本点各特征维度的平方和 $\sum_{i=1}^{N} X_i^2$

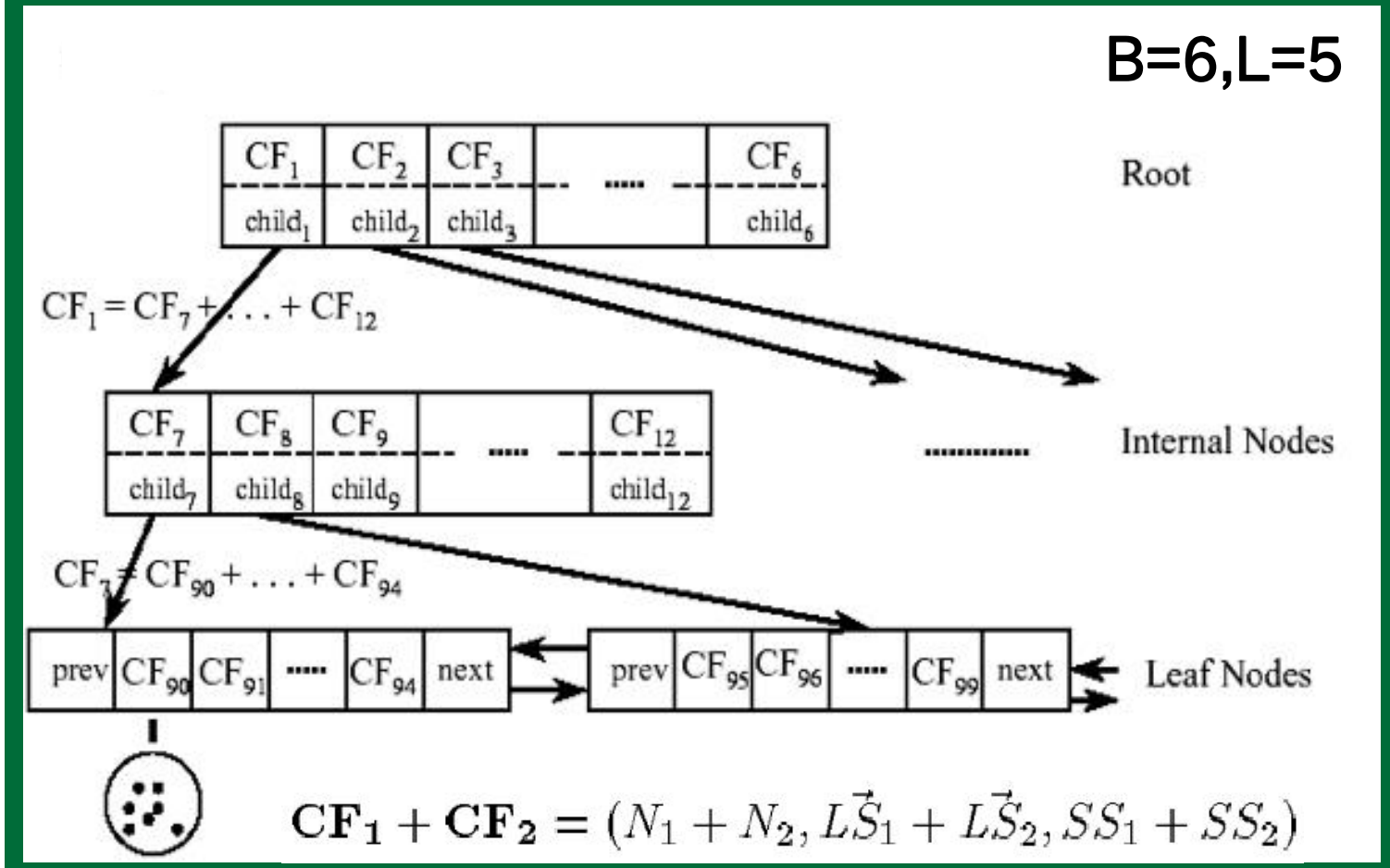


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF = (5, (16,30),(54,190))

LS=(3+2+4+4+3,4+6+5+7+8)=(16,30)

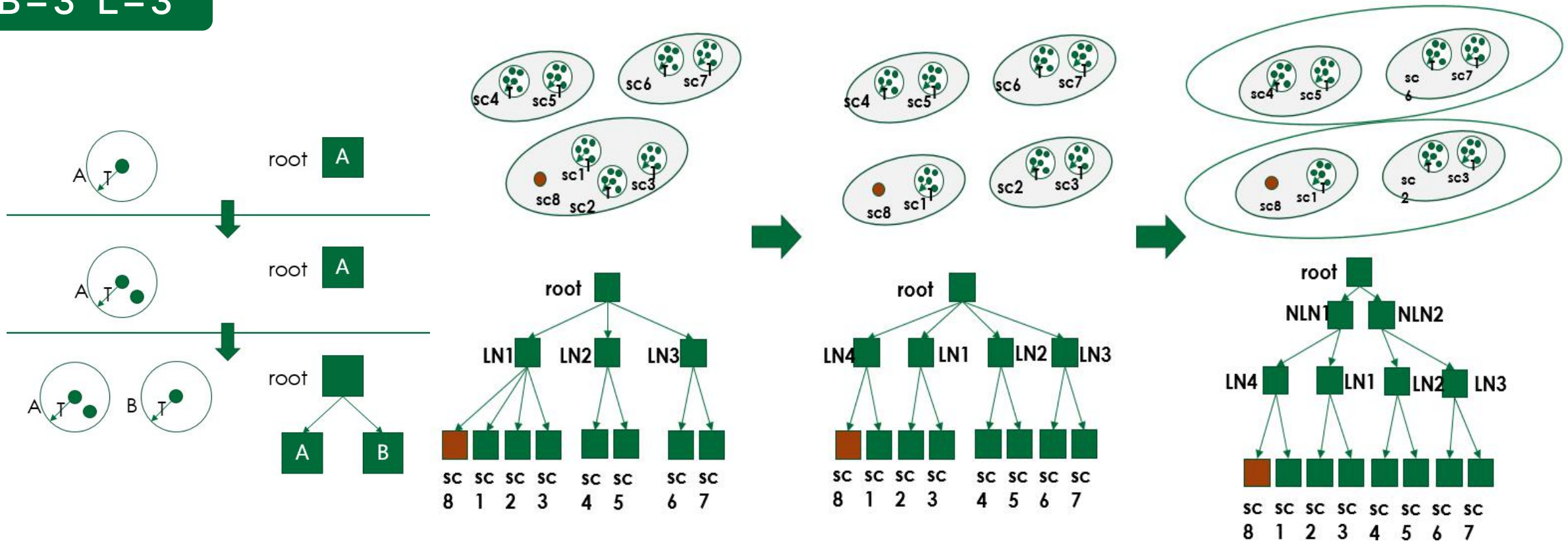
SS =(9+4+16+16+9+16+36+25+49+64)
=(54+190)
=244



聚类特征树性质

- 满足线性关系
- 叶节点是双向链表
- B:每个内部节点的最大CF数
- L:每个叶子节点的最大CF数
- T:叶节点每个CF的最大样本半径阈值

B=3 L=3


A
root
A
root
A
B
root
A
B
sc4
sc5
sc6
sc7
sc1
sc8
sc2
sc3
root
LN1
LN2
LN3
sc 8
sc 1
sc 2
sc 3
sc 4
sc 5
sc 6
sc 7
LN4
NLN1
NLN2

BIRCH算法

BIRCH算法优点：

- 节约内存，所有的样本都在磁盘上，CF Tree**仅仅存了CF节点和对应的指针**。
- **聚类速度快**，只需要一遍扫描训练集就可以建立CF Tree，CF Tree的增删改都很快。
- 可以**识别噪音点**，还可以对数据集进行初步分类的预处理

BIRCH算法缺点：

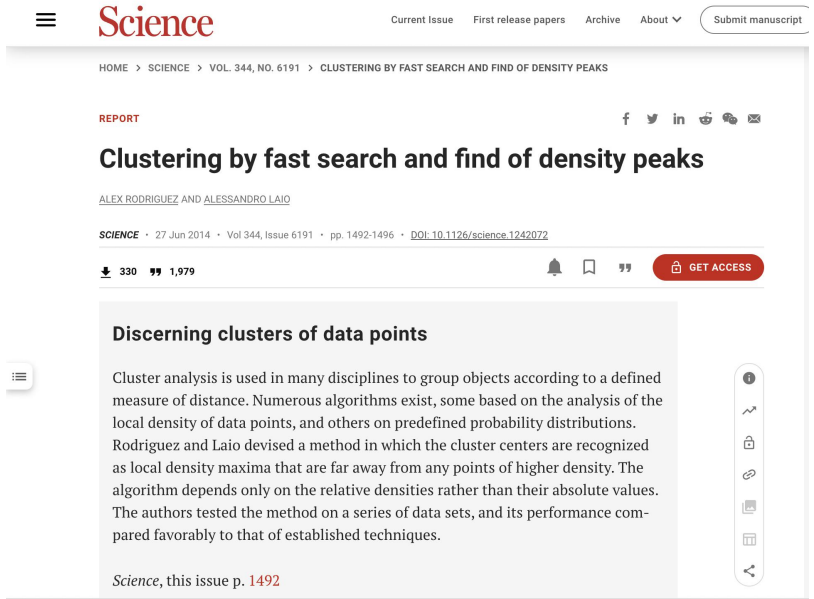
- 由于CF Tree对每个节点的**CF个数有限制**，导致聚类的结果可能和真实的类别分布不同.
- 对**高维特征的数据聚类效果不好**。
- 如果数据集的分布簇**不是类似于超球体**，或者说不是凸的，则聚类效果不好。

基于密度的聚类算法

它不是基于各种各样的距离的，而是**基于密度**的。这样就能克服基于距离的算法只能发现“类圆形”的缺点。**只要一个区域中的点的密度大过某个阈值，就把它加到与之相近的聚类中去**

常见算法

- DBSCAN算法
- **DPC算法-基于快速搜索和发现密度峰值的聚类算法**
- OPTICS算法



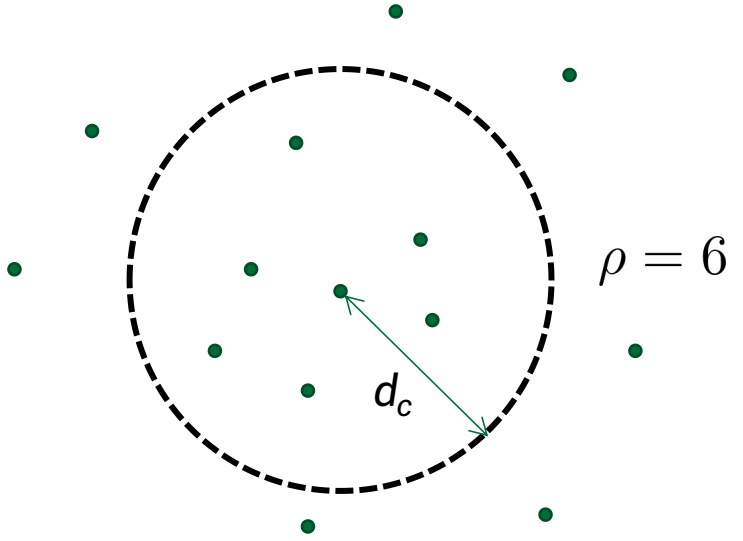
DPC算法

- 能够**自动地发现簇中心**，实现**任意形状**数据的高效聚类
- 算法分为两步：**寻找聚类中心**和**分配簇标签**

DPC算法-参数

- **密度ρ**：DPC将每个点的密度定义为，距离该点d_c以内的其它点的个数，用公式表示为

$$\rho_i = \sum_j \chi(d_{ij} - d_c)$$



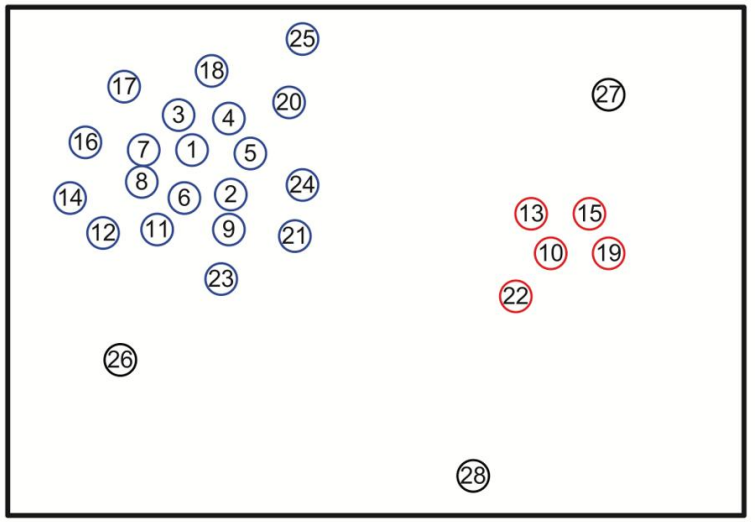
- **d_c**：称为**截断距离**，是该算法的唯一一个参数，X(x)在x<0时等于1，在x>0时等于0.

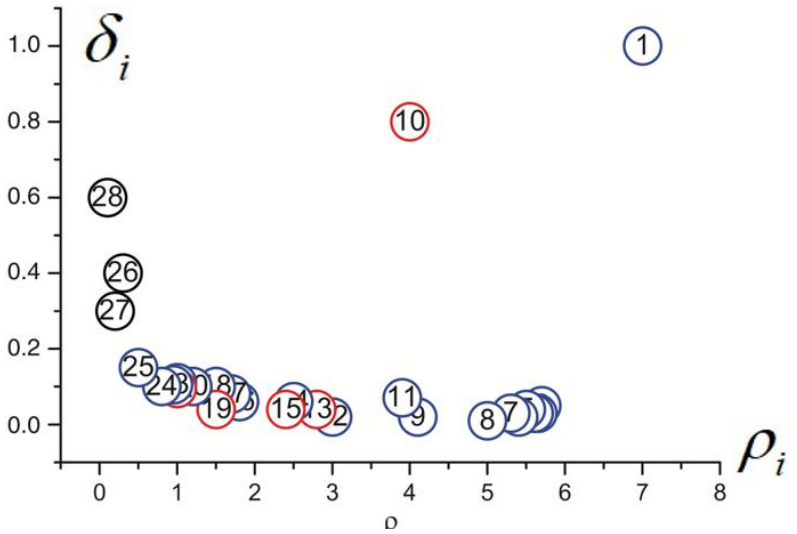
- **δ：每个点与最近的密度更高的点的距离**，在所有密度比P点大的点中，假设离P点最近的那个点是点Q，则δ_P为PQ的距离。计算完密度后，DPC将计算每个点的δ值。　用公式表示为：

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij})$$

DPC算法-聚类流程

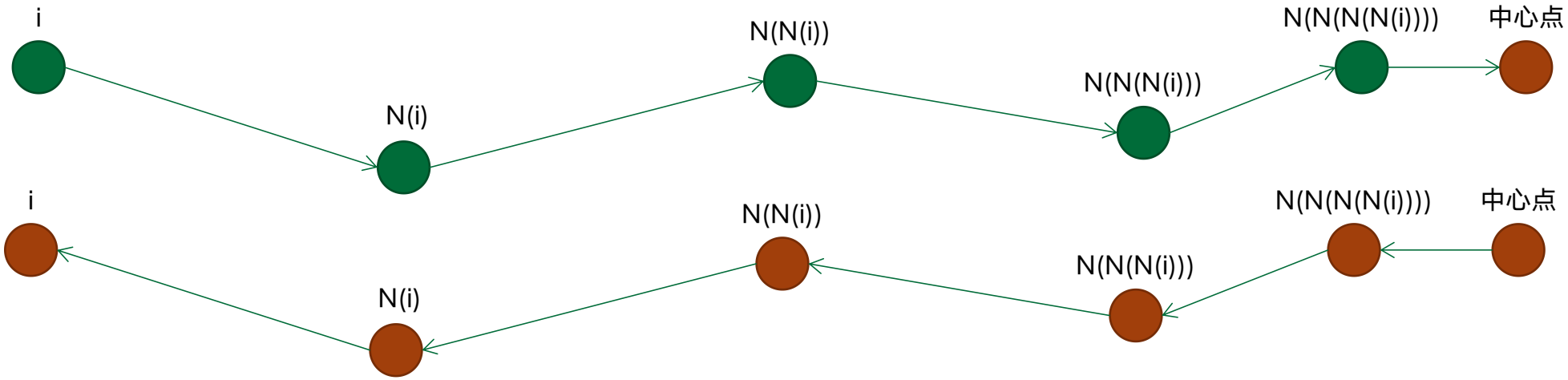
- 1. 根据**截断距离**算出任意数据点 x_i 的**局部密度** ρ_i
- 2. 对于任意数据点 x_i，计算出δ_i
- 3. 以 ρ_i为横轴，以 δ_i 为纵轴，画出**决策图**
- 4. 利用决策图，将ρ_i 和δ_i **都相对较高的点标记为簇中心**；将ρ_i 相对较低但是 δ_i 相对较高的点**标记为噪声点**
- 5. **将剩余点进行分配**，分配时，将每个剩余点分配到它的最近邻且密度比其大的数据点所在簇





DPC算法-聚类流程

- 在选择完中心点后，对数据集中剩余的各点进行**簇的分配工作**。
- 分配的原则十分简单，在之前计算δ那一步，我们已经知道了任意一点i，所有密度大于i的点中距离i最近的点是N_i。所以只需将每个点i分配到与N_i相同的簇中即可，最后找到密度值最大的中心点，这是一个**递归**的过程，如下图所示。



优点

1.参数非常少，只有一个。

3.在服从高斯分布的数据集上非常准确。

3.实现简单，时间复杂度仅O(n^2)

缺点

1.在不符合高斯分布的数据集上结果不好，且很难通过调整参数改善。

基于网格的聚类算法

用不同的网格划分方法，将数据空间划分成为**有限个单元的网格结构**，并对网格数据结构的统计信息进行压缩表达，基于这些统计信息**判断高密度网格单元**，最后将相连的高密度网格单元识别为**簇**

常见算法

- **STING算法-统计信息网格**
- CLIQUE算法
- WAVE-CLUSTER算法

STING算法

Statistical Information Grid-based method

是一种基于网格的多分辨率聚类技术，将空间区域划分为矩形单元。

STING算法

- **核心思想：**根据属性的相关统计信息进行划分网格，而且网格是**分层次的**，在一个网格内的数据点**即为一个簇**。
- 针对不同级别的分辨率，通常存在**多个级别**的矩形单元，这些单元形成了一个层次结构：**高层**的每个单元被划分为**多个低一层**的单元。

STING算法的两个参数：

- 网格的**步长**——确定空间网格划分
- **密度阈值**——网格中对象数量大于等于该阈值表示该网格为稠密网格

STING算法-聚类流程

1.首先我们先划分一些层次，按**层次划分网格**；

2.计算最底层单位网格的**统计信息**（如均值，最大值和最小值）；

3.从最底层**逐层计算**上一层每个父单元格的统计信息，直到最顶层

4.同时根据**密度阈值**标记**稠密网格**。

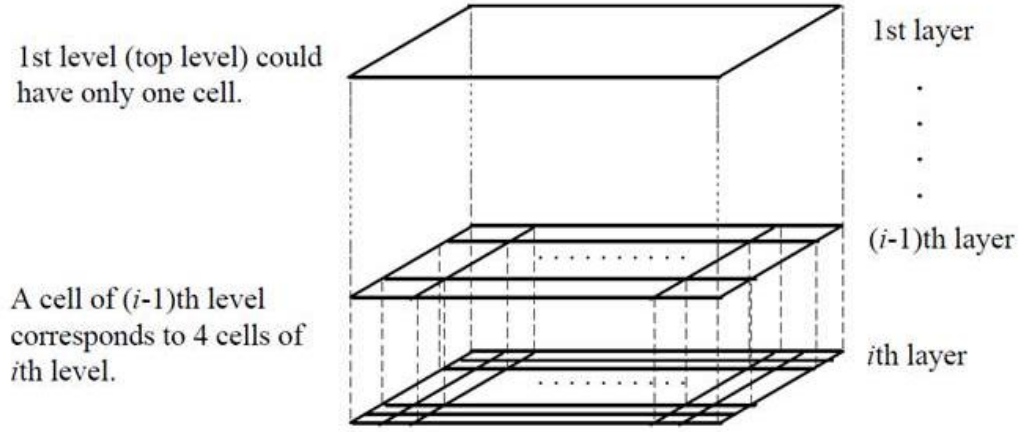
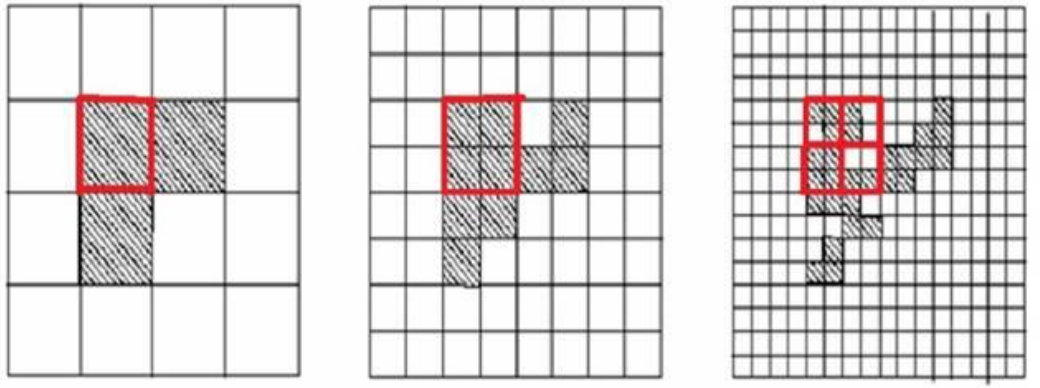


Figure 1. Hierarchical Structure



STING

STING优点：

- 网格结构有利于并行处理和增量更新;
- 效率很高。

STING缺点：

- STING的聚类质量取决于网格结构的**最底层的粒度**;
- STING在构建一个父亲单元时没有考虑到子女单元和其他相邻单元之间的联系。所有的簇边界不是水平的，就是竖直的，**没有斜的分界线**。降低了聚类质量。

4 demo展示

基于豆瓣读书书籍简介的聚类

数据来源

来源：豆瓣网

内容：书籍简介

数量：85本

聚类方法

- K-means
- Birch
- DPC

调用库

python：

glob、jieba、

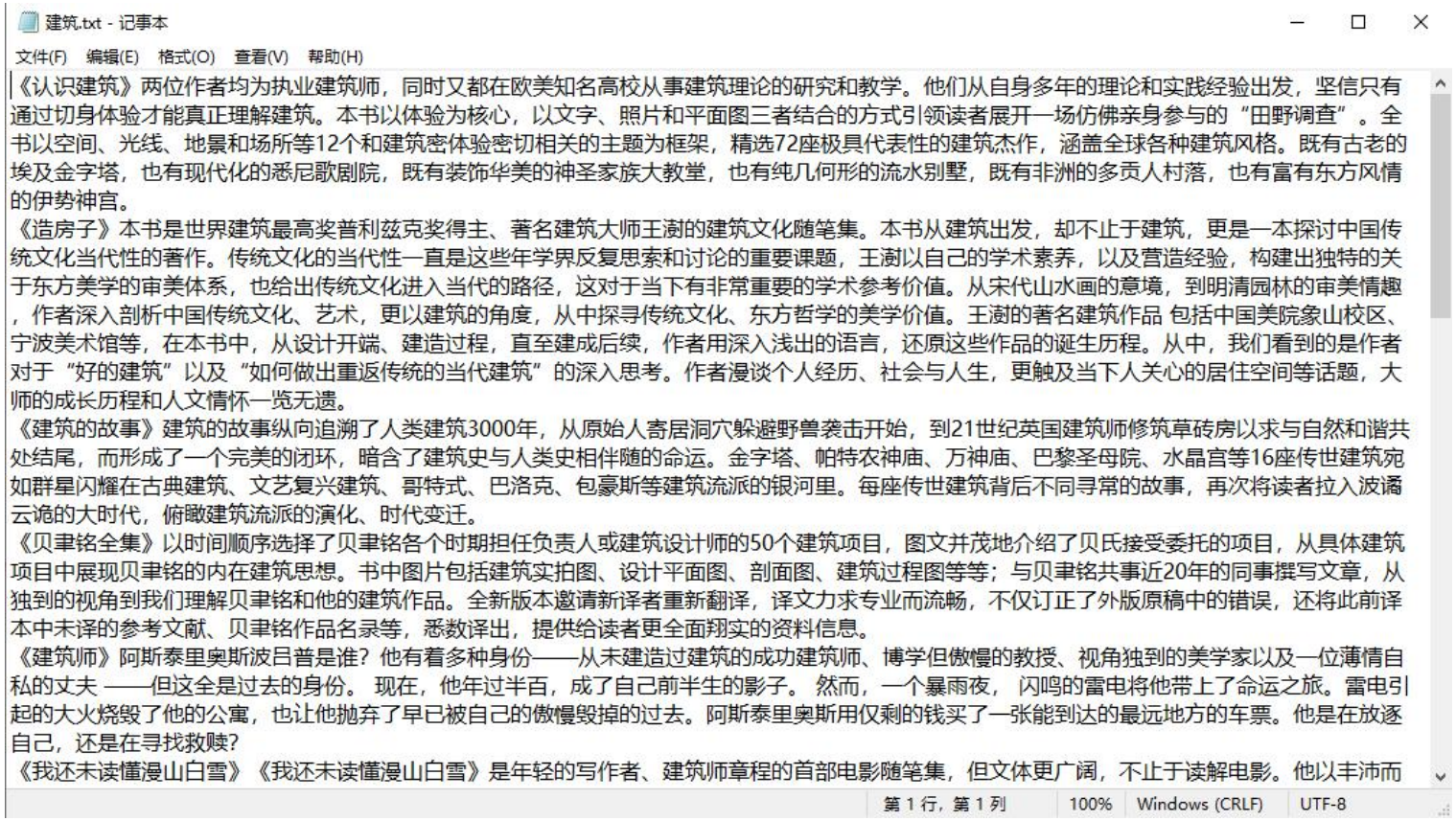
sklearn、os等

方法步

```
if __name__ == "__main__":
    root = './Txtc'
    stopWords = open('Txtc1.txt', 'r', encoding='utf-8').read().split('\n')
    docPath = 'doc.txt'
    k = num
    SaveDoc(['', '/Txtc'], docPath, stopWords)
    bktitle = [[] for i in range(k)]
    weight = TFIDF(docPath)
    X = PCA(weight, dimension=85)  # 将原始权重数据降维
    # plt.scatter(X[:,0],X[:, 1])
    # plt.show()
    # y = kmeans(X, k) # y=聚类后的类标签
    y = birch(X, k)
    # dpc = DPC(density_measure='gauss')
    # y = dpc.fit_predict(X)
    # centers = dpc.get_cluster_center()
    # c_num = len(centers)
    # bktitle = [[] for i in range(c_num)]
    # print("开始聚类:        DPC")
    # print(y)
    plt.scatter(X[:, 0], X[:, 1], c=y)
    # for i in range(len(bookTitle)):
    #     bktitle[y[i] - 1].append(bookTitle[i])
    # for i in range(c_num):
    #     print(bktitle[i],len(bktitle[i]))
    plt.show()
    silhouette_avg, sample_silhouette_values = Silhouette(X, y)  # 轮廓系数
    Draw(silhouette_avg, sample_silhouette_values, y, k)
```

基于豆瓣读书书籍简介的聚类

语料库



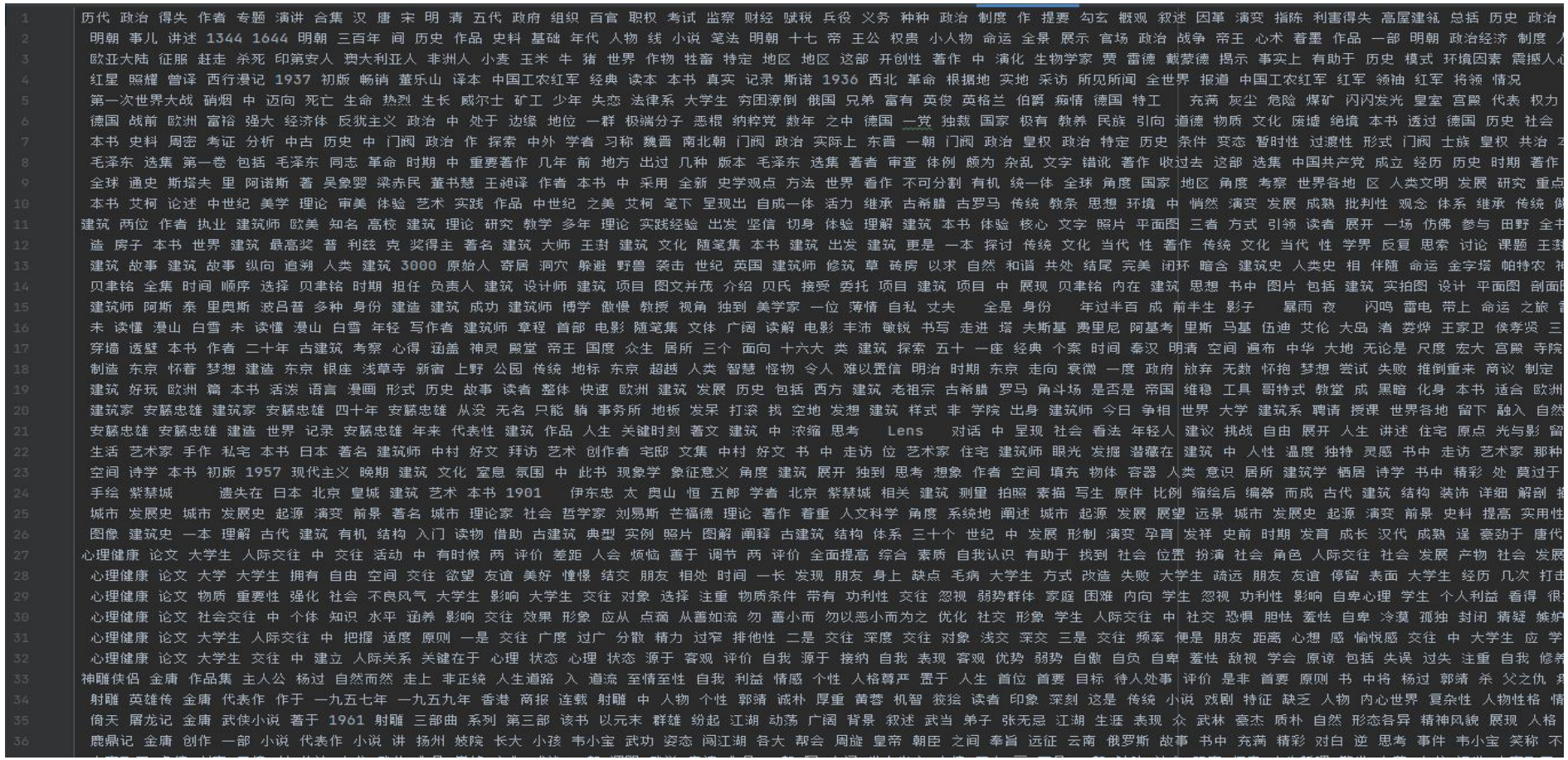
基于豆瓣读书书籍简介的聚类

停词表

Txtc1.txt - 记事本

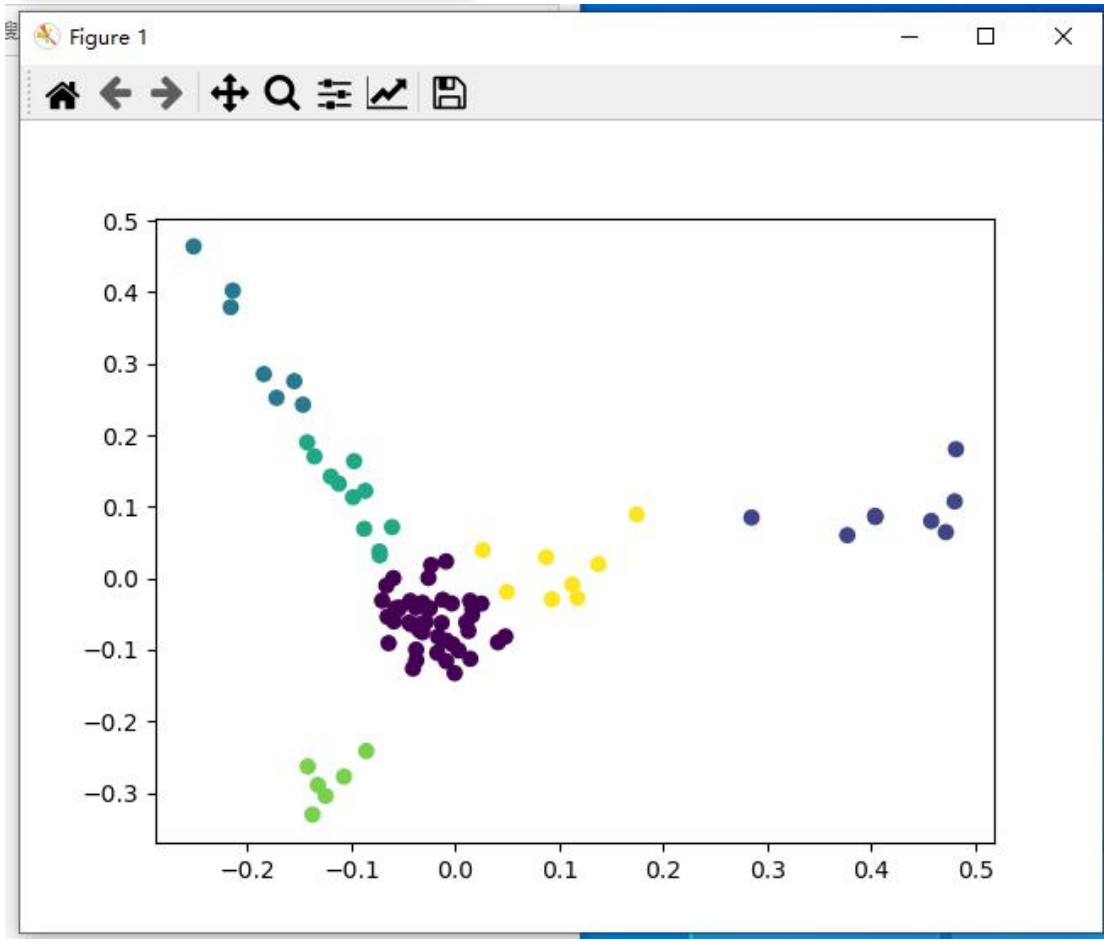
文件(F) 编辑(E) 格式(O) 查看(V) 帮

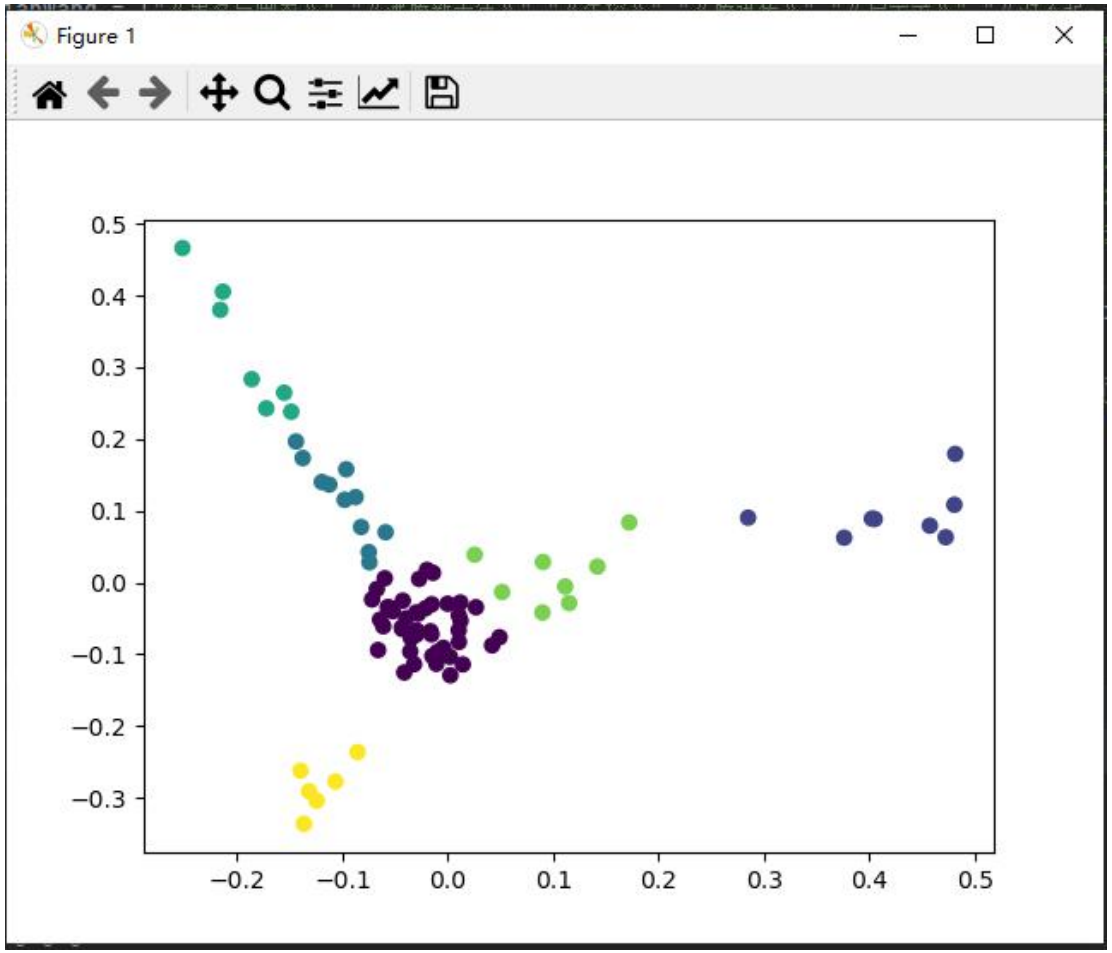
去年
nan
安全
食品
一个
中国
自己
可以
我们
他们
美国
没有
问题
这个
生产
进行
万元
现在
很多
不是
如果
这些
进行
这样
福喜



基于豆瓣读书书籍简介的聚类

K-means





K-means

```
开始聚类:              Kmeans
[0 0 0 0 0 0 0 0 0 5 1 1 1 1 5 5 5 0 1 1 1 5 5 1 0 5 4 4 4 4 4 4 0 0 0 0 0
 0 0 0 0 0 3 2 0 0 2 3 3 3 0 3 2 3 2 3 2 2 3 0 3 3 2 5 3 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0]
['《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《制造东京》', '《城市发展史》', '《神雕侠侣》', '《射雕英雄传》
['《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城 ： 遗失在日本的北京皇城建筑艺术》'] 8
['《三体》', '《齐马蓝》', '《银河帝国：基地七部曲》', '《三体Ⅱ》', '《三体Ⅲ》', '《沙丘》', '《惨败》'] 7
['《索拉里斯星》', '《星之继承者》', '《未来学大会》', '《永恒的终结》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《你一生的故事》', '《其主之声》', '《球状闪电》', '《机器人大师》', '《环界1》'] 11
['《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》'] 6
['《中世纪之美》', '《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《图像中国建筑史》', '《雪崩》'] 8
计算轮廓系数:
0.5733305955530371
It's over!!
```

```
开始聚类:              Kmeans
[0 0 0 0 0 0 0 0 0 4 1 1 1 1 4 4 4 0 1 1 1 4 4 1 0 4 5 5 5 5 5 5 0 0 0 0 0
 0 0 0 0 0 2 3 0 0 3 2 2 2 0 2 3 2 3 2 3 3 2 0 2 2 3 4 2 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0]
['《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《制造东京》', '《城市发展史》', '《神雕侠侣》',
['《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城 ： 遗失在日本的北京皇城建筑艺术》'] 8
['《索拉里斯星》', '《星之继承者》', '《未来学大会》', '《永恒的终结》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《你一生的故事》', '《其主之声》', '《球状闪电》', '《机器人大师》', '《环界1》'] 11
['《三体》', '《齐马蓝》', '《银河帝国：基地七部曲》', '《三体Ⅱ》', '《三体Ⅲ》', '《沙丘》', '《惨败》'] 7
['《中世纪之美》', '《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《图像中国建筑史》', '《雪崩》'] 8
['《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》'] 6
计算轮廓系数:
0.5664446892166937
It's over!!
```

基于豆瓣读书书籍简介的聚类

K-means

《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》

《中世纪之美》', '《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《图像中国建筑史》', '《雪崩》

'《三体》', '《齐马蓝》', '《银河帝国：基地七部曲》', '《三体Ⅱ》', '《三体Ⅲ》', '《沙丘》', '《惨败》'

《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城：遗失在日本的北京皇城建筑艺术》

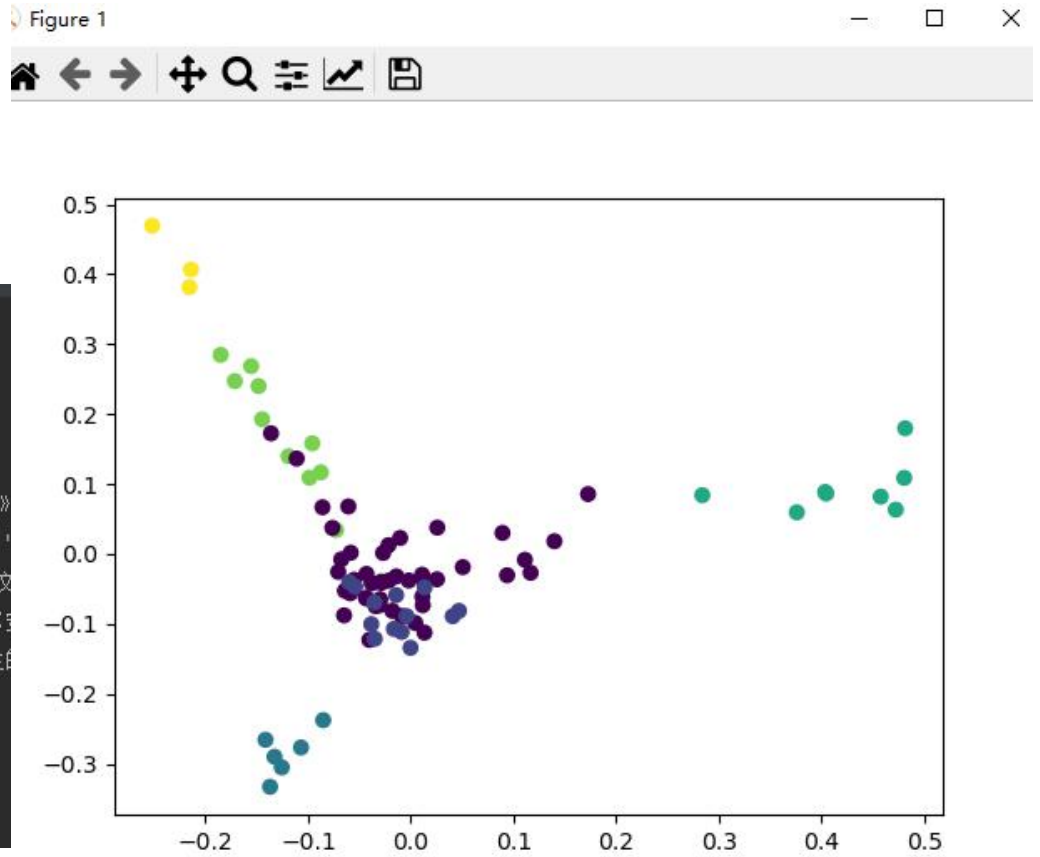
《索拉里斯星》', '《星之继承者》', '《未来学大会》', '《永恒的终结》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《你一生的故事》', '《其主之声》', '《球状闪电》', '《机器人大师》', '《环界1》

《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《制造东京》', '《城市发展史》', '《神雕侠侣》', '《射雕英雄传》', '《倚天屠龙记》', '《鹿鼎记》', '《小李飞刀:多情剑客无情剑》', '《侠客行》', '《逝去的武林:一代形意拳大师口述历史》', '《武林:一代形意拳大师口述历史》', '《蜀山剑侠传》', '《笑傲江湖》', '《献给阿尔吉侬的花束》', '《克莱因壶》', '《平面国》', '《醉步男》', '《鱼翅与花椒》', '《中国人超会吃》', '《雅舍谈吃》', '《行走的柠檬》', '《四口吃遍江户》', '《随园食单》', '《料理图鉴》', '《美食与文明》', '《五味》', '《于谦：人间烟火》', '《肚子饿万岁》', '《厨艺的常识》', '《鸭川食堂》', '《这本书好吃吗》', '《人间滋味》', '《至味在人间》', '《流动的餐桌》', '《川菜》', '《金牌主厨的面点课》', '《世间味道》

基于豆瓣读书书籍简介的聚类

Birch

```
开始聚类:              birch
输出聚类结果:
[0 0 0 0 0 0 0 0 0 3 3 3 3 0 0 0 0 3 3 3 0 0 3 0 0 2 2 2 2 2 2 0 0 0 0 0
 0 0 0 0 0 4 5 0 0 4 4 4 4 0 0 4 0 5 4 5 4 4 0 0 0 4 0 0 1 1 0 0 0 1 1 0 1
 1 0 1 1 1 1 1 0 1 1 1]
['《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》
['《鱼翅与花椒》', '《中国人超会吃》', '《随园食单》', '《料理图鉴》', '《五味》', '《于谦：人间烟火》', '《厨艺的常识》', '
['《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文
['《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《
['《索拉里斯星》', '《齐马蓝》', '《星之继承者》', '《未来学大会》', '《永恒的终结》', '《银河帝国：基地七部曲》', '《你一生
['《三体》', '《三体Ⅱ》', '《三体Ⅲ》'] 3
计算轮廓系数:
0.014277223302942345
It's over!!
```



基于豆瓣读书书籍简介的聚类

Birch

《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》

《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城：遗失在日本的北京皇城建筑艺术》

《索拉里斯星》', '《齐马蓝》', '《星之继承者》', '《未来学大会》', '《永恒的终结》', '《银河帝国：基地七部曲》', '《你一生的故事》', '《沙丘》', '《其主之声》', '《惨败》

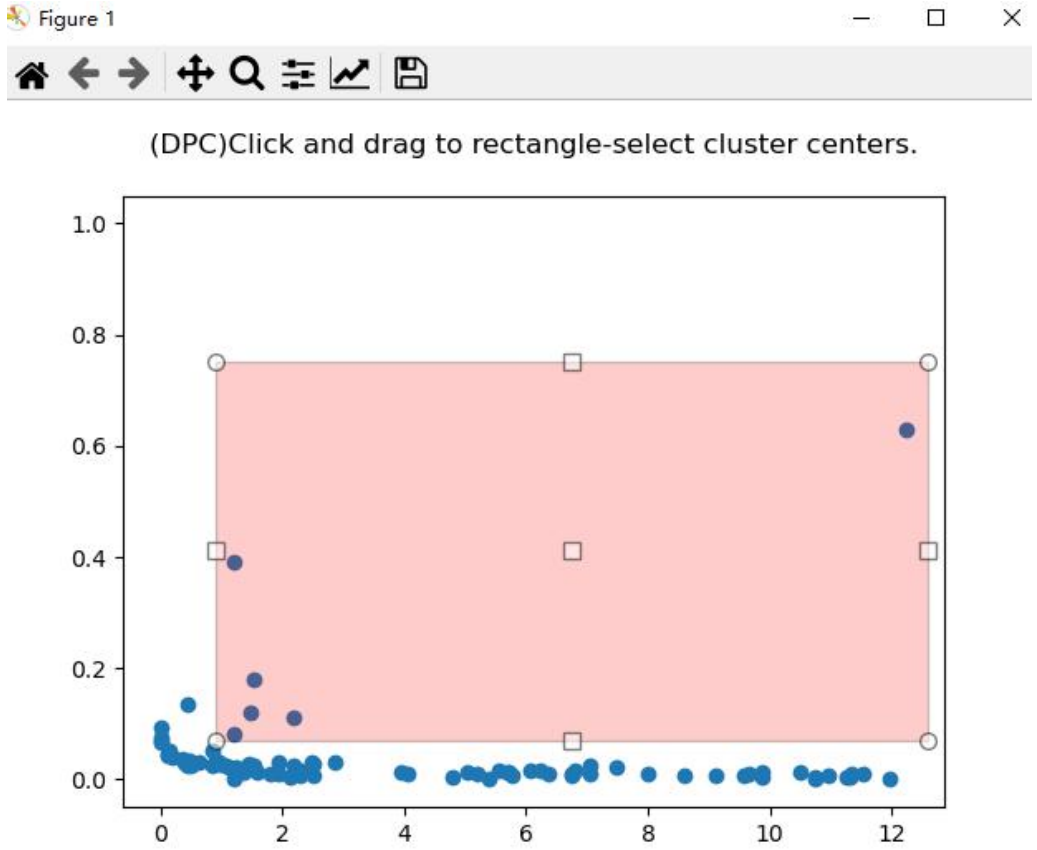
《三体》', '《三体 II 》', '《三体 III 》

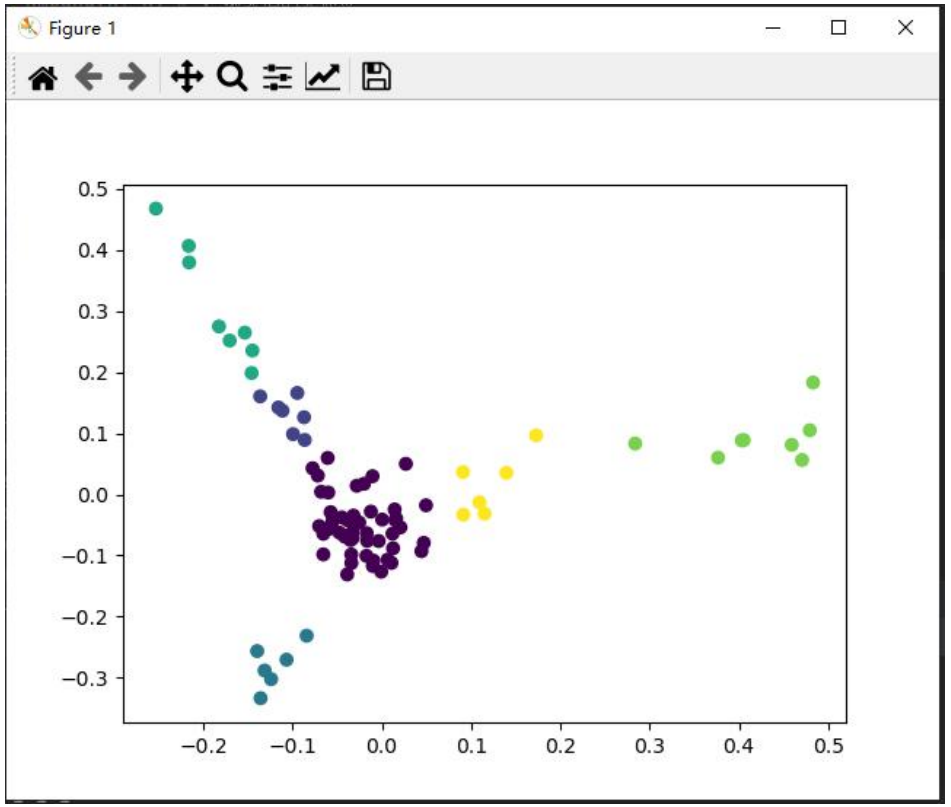
《鱼翅与花椒》', '《中国人超会吃》', '《随园食单》', '《料理图鉴》', '《五味》', '《于谦：人间烟火》', '《厨艺的常识》', '《鸭川食堂》', '《这本书好吃吗》', '《人间滋味》', '《至味在人间》', '《川菜》', '《金牌主厨的面点课》', '《世间味道》

《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《中世纪之美》', '《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《制造东京》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《城市发展史》', '《图像中国建筑史》', '《神雕侠侣》', '《射雕英雄传》', '《倚天屠龙记》', '《鹿鼎记》', '《小李飞刀:多情剑客无情剑》', '《侠客行》', '《逝去的武林:一代形意拳大师口述历史》', '《武林:一代形意拳大师口述历史》', '《蜀山剑侠传》', '《笑傲江湖》', '《献给阿尔吉侬的花束》', '《克莱因壶》', '《平面国》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《醉步男》', '《球状闪电》', '《机器人大师》', '《雪崩》', '《环界1》', '《雅舍谈吃》', '《行走的柠檬》', '《四口吃遍江户》', '《美食与文明》', '《肚子饿万岁》', '《流动的餐桌》

基于豆瓣读书书籍简介的聚类

DPC





基于豆瓣读书书籍简介的聚类

DPC

```
开始聚类:       DPC
[1 1 1 1 1 1 1 1 1 5 5 5 5 6 6 6 1 5 5 5 6 6 5 1 6 3 3 3 3 3 3 1 1 1 1 1
 1 1 1 1 1 2 4 1 1 4 4 1 2 1 2 4 2 4 2 4 4 2 1 2 1 4 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
['《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《中世纪之美》', '《制造东京》', '《城市发
['《索拉里斯星》', '《永恒的终结》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《你一生的故事》', '《其主之声》', '《球状闪电》'] 7
['《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》'] 6
['《三体》', '《齐马蓝》', '《星之继承者》', '《银河帝国：基地七部曲》', '《三体Ⅱ》', '《三体Ⅲ》', '《沙丘》', '《惨败》'] 8
['《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城 ： 遗失在日本的北京皇城建筑艺术》'] 8
['《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《图像中国建筑史》'] 6
计算轮廓系数:
0.5408256899760746
It's over!!
```

基于豆瓣读书书籍简介的聚类

DPC

《心理健康论文一》', '《心理健康论文二》', '《心理健康论文三》', '《心理健康论文四》', '《心理健康论文五》', '《心理健康论文六》

《三体》', '《齐马蓝》', '《星之继承者》', '《银河帝国：基地七部曲》', '《三体 II 》', '《三体 III 》', '《沙丘》', '《惨败》

《认识建筑》', '《造房子》', '《建筑的故事》', '《贝聿铭全集》', '《建筑也可以很好玩：欧洲篇》', '《建筑家安藤忠雄》', '《安藤忠雄》', '《手绘紫禁城：遗失在日本的北京皇城建筑艺术》

《建筑师》', '《我还未读懂漫山白雪》', '《穿墙透壁》', '《生活艺术家的手作私宅》', '《空间的诗学》', '《图像中国建筑史》

《索拉里斯星》', '《永恒的终结》', '《小镇奇谈》', '《仿生人会梦见电子羊吗？》', '《你一生的故事》', '《其主之声》', '《球状闪电》

《中国历代政治得失》', '《明朝那些事儿》', '《红星照耀中国》', '《枪炮、病菌与钢铁》', '《巨人的陨落》', '《第三帝国的到来》', '《东晋门阀政治》', '《毛泽东选集》', '《全球通史》', '《中世纪之美》', '《制造东京》', '《城市发展史》', '《神雕侠侣》', '《射雕英雄传》', '《倚天屠龙记》', '《鹿鼎记》', '《小李飞刀:多情剑客无情剑》', '《侠客行》', '《逝去的武林:一代形意拳大师口述历史》', '《武林:一代形意拳大师口述历史》', '《蜀山剑侠传》', '《笑傲江湖》', '《献给阿尔吉侬的花束》', '《克莱因壶》', '《未来学大会》', '《平面国》', '《醉步男》', '《机器人大师》', '《雪崩》', '《环界1》', '《鱼翅与花椒》', '《中国人超会吃》', '《雅舍谈吃》', '《行走的柠檬》', '《四口吃遍江户》', '《随园食单》', '《料理图鉴》', '《美食与文明》', '《五味》', '《于谦：人间烟火》', '《肚子饿万岁》', '《厨艺的常识》', '《鸭川食堂》', '《这本书好吃吗》', '《人间滋味》', '《至味在人间》', '《流动的餐桌》', '《川菜》', '《金牌主厨的面点课》', '《世间味道》

结果分析

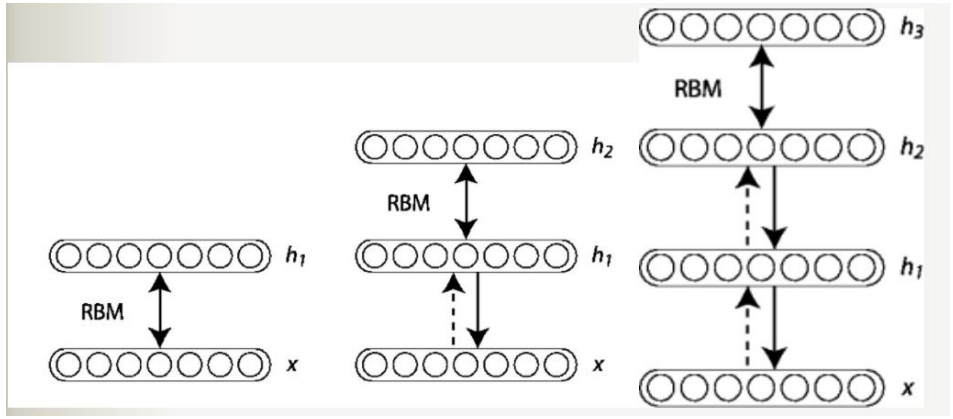
K-means	时间复杂度低，算法简单 需要手动输入参数K 由于随机性，算法不稳定
Birch	速度快、对异常值影响不大 对高斯簇效果不好
DPC	对任意形状均可聚类，可通过决策图直观推断出簇个数 dc等参数需要提前输入，决策图的选择具有主观性 对密度差异大的难以区分

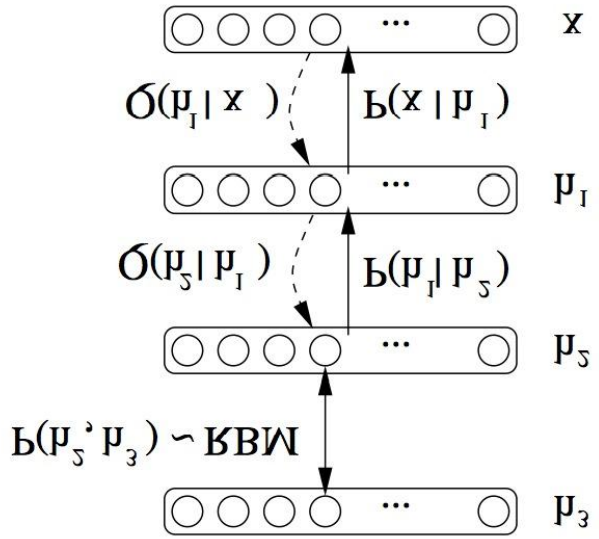
5 最新研究进展

近年来的文本聚类研究与传统的**基于密度，基于划分，基于层次，基于网格**等聚类算法相比，更多的结合了**深度学习**，这些方法按照时间顺序可以分为4个大类：

1. Deep Belief Networks
2. Convolutional neural network
3. Recurrent neural network
4. Autoencoder

Deep Belief Network





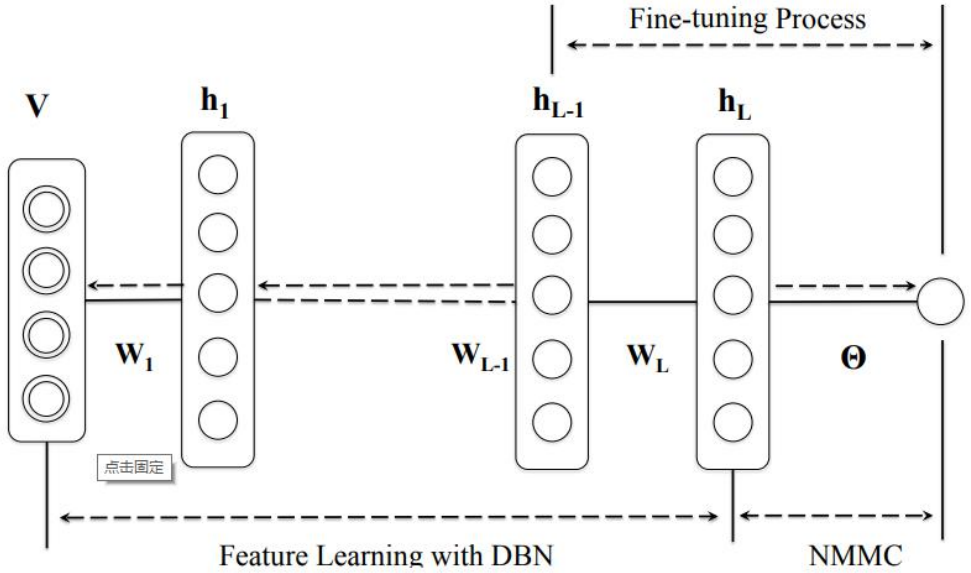
Deep Belief Network 是Hinton等在2006年提出了一种新的方法来求得这种比较接近最优解的初始权重。

在使用上述的逐层无监督方法学得节点之间的权重以及节点的偏置之后(亦即初始化)。

使用逐层无监督方法来初始化了权重值, 使其比较接近最优值, 解决了之前多层神经网络训练时存在的问题, 能够得到很好的效果。

DBN文本聚类应用

年份	DBN
2015	Deep learning with nonparametric clustering
2016	A personalized Markov clustering and deep learning approach for Arabic text categorization

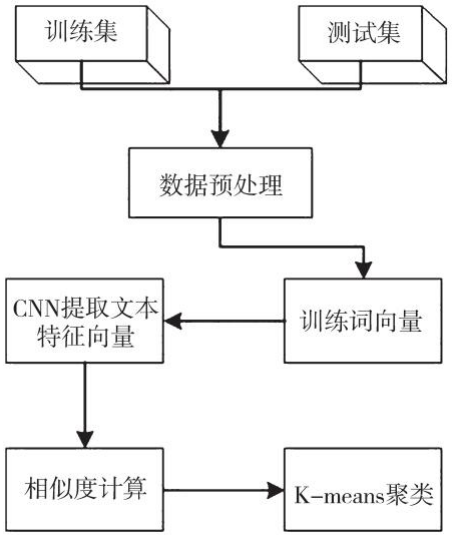


Deep Learning with Nonparametric Clustering ——Gang Chen

最大边际聚类(maximum margin clustering, MMC)是一种判别方法，将每个实例的标签作为潜在变量，使用支持向量机(SVM)进行大边际聚类。

针对无监督聚类问题的深度学习，提出一种**无参数聚类的深度置信网络**。利用了深度学习在特征表示和降维方面的优势。然后在最大边距框架下进行无参数聚类，该框架是一种判别聚类模型。最后，在深度置信网络中对模型参数进行了优化。

深度学习如CNN在图像领域的成功应用，给文本聚类也带了新的思考。

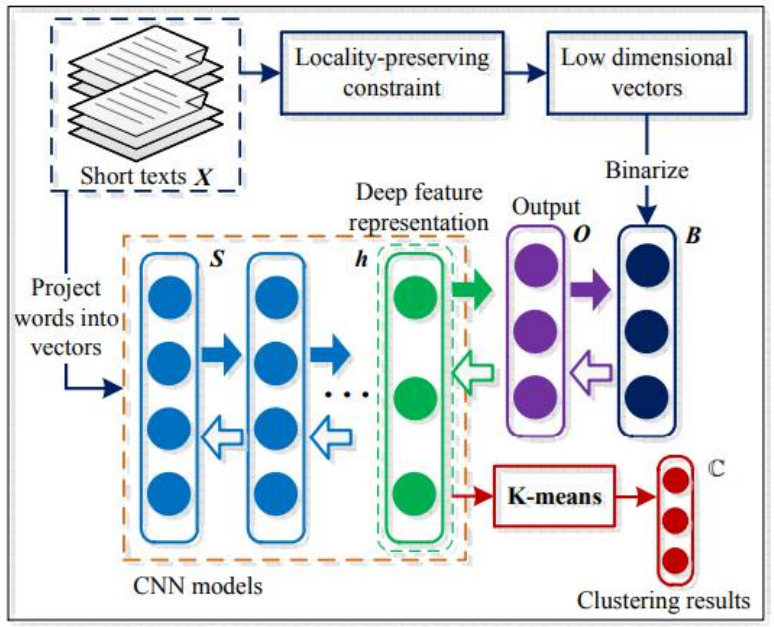


Convolutional neural network

应用CNN的基础方法如左图所示。首先通过word2vec或其他模型来获得语料库中所有词的词向量；然后将文本转换为词向量矩阵，再经过CNN对文本的词向量矩阵进行特征提取，获得文本的特征向量表示；选取合适的相似度计算方法计算文本向量的相似度；最后使用K-means方法进行聚类。

CNN文本聚类应用

年份	CNN
2015	Short Text Clustering via Convolutional neural networks (STCC)
2016	Semi-supervised clustering for short text via deep representation learning
2017	Self-Taught Convolutional Neural Networks for Short Text Clustering (dubbed STC2)



Short Text Clustering via Convolutional Neural Networks
——Jiaming Xu, Peng Wang

由于短文本的稀疏性，使得聚类成为一个具有挑战性的问题。本文提出了一种基于卷积神经网络的短文本聚类算法(STCC)，通过不使用任何外部标签/标签的自学学习框架，考虑学习特征的一个约束，更有利于聚类。

首先，我们将原始关键字特征嵌入到具有局部保持约束的紧凑二进制码中。然后，研究单词嵌入并将其输入卷积神经网络以学习深度特征表示。在获得学习过的表示之后，我们使用K-means对它们进行聚类。

年份	RNN
2018	Comparison of deep learning based concept representations for biomedical document clustering
2018	Text clustering algorithm based on deep representation learning
2019	Sequential embedding induced text clustering, a non-parametric Bayesian approach
2019	ADC: Advanced document clustering using contextualized representations

Recurrent neural network

应用LSTM或Bi-LSTM等的基础做法和CNN类似，也是用于文本的特征提取和表示。一个包含n个单词的文本的概率表示可以是如下两种形式：

$$p(t_1, t_2, \ldots, t_n) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \ldots, t_{k-1})$$

$$p(t_1, t_2, \ldots, t_n) = \prod_{k=1}^{N} p(t_k | t_n, t_{n-1}, \ldots, t_{k+1})$$

$p(t_k | t_1, t_2, \ldots, t_{k-1})$ 表示给定前这k-1个单词后，第k个单词为t_k的概率。

类似地，$p(t_k | t_n, t_{n-1}, \ldots, t_{k+1})$ 表示给定第k+1到n个单词后，第k个单词为t_k的概率。

Recurrent neural network

一个基础的应用LSTM来获取文本的特征的方法为：将包含n个单词的文本输入LSTM，得到每个时间步下的隐状态h_k：

$$h_k = [\overrightarrow{h_k}, \overleftarrow{h_k}]$$

其中，$\overrightarrow{h_k} = f(\overrightarrow{h_{k-1}}, x_k)$ ，$\overleftarrow{h_k} = f(\overleftarrow{h_{k+1}}, x_k)$，这里用函数f来表示LSTM的操作，$x_k$表示文本的第k个单词。

假设隐状态的维度为d，可以采取三种方式计算文本的最终特征表示

①Max-pooling 对于n个隐状态，选取每个维度上的最大值构成文本特征

②Mean-pooling 对于n个隐状态，计算每个维度上的均值构成文本特征

③ last-time 直接选取第1个和第n个隐状态构成文本特征

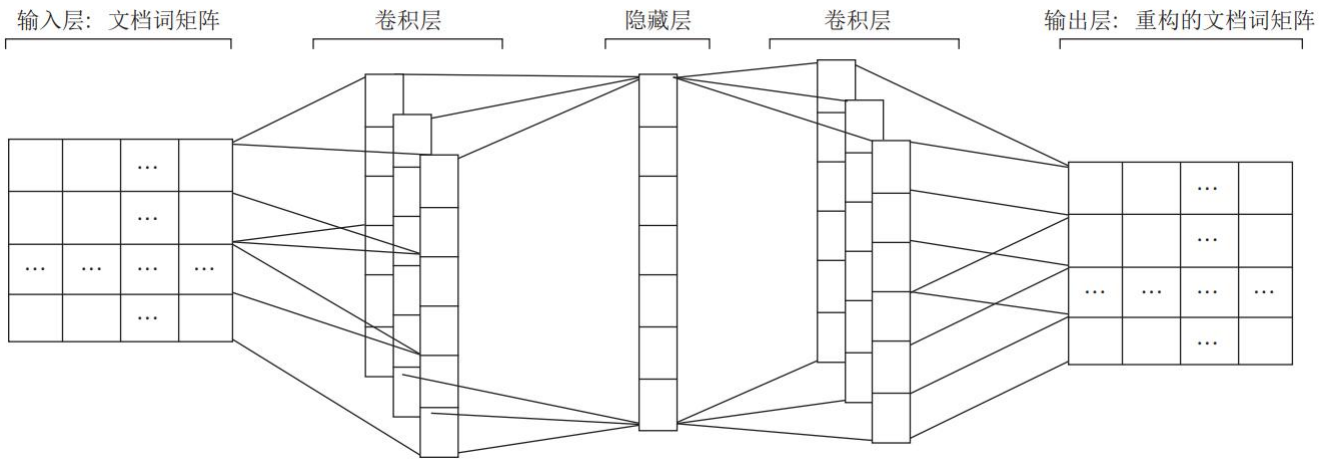
Autoencoder

自编码器被广泛应用于文本聚类前的特征降维。一类方法是先使用自编码器学习从数据域到低维潜在空间的非线性映射，随后在学到的低维空间进行聚类。这些方法将自编码器作为一个预处理阶段，与后续的聚类阶段分开设计。也就是说，特征学习和聚类是按顺序进行的。

当前应用自编码器到文本聚类的一个趋势是希望模型能同时进行特征学习和聚类。这类方法认为，将特征学习和聚类分开设计时，自编码器在低维特征学习过程中不是以促进聚类效果为目标的，学到的数据的潜在低维不一定适合于聚类，因此考虑联合特征降维和聚类。

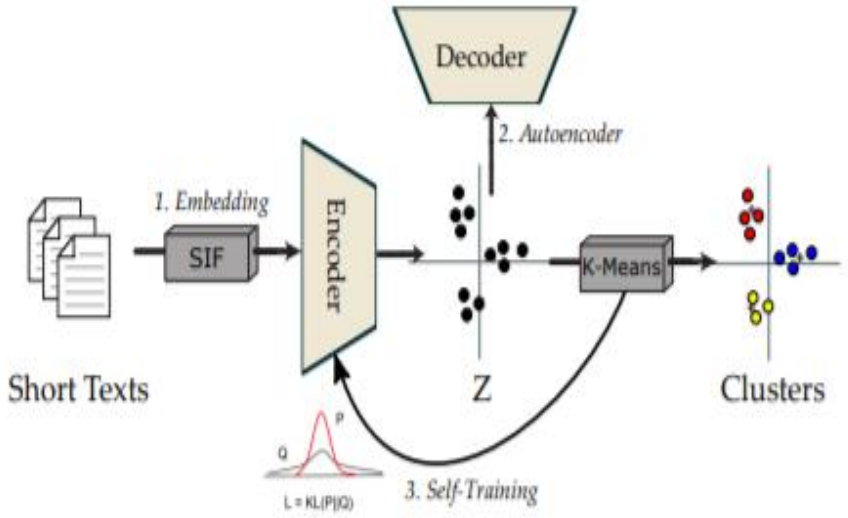
Autoencoder

一个应用自编码器的基础方法如下图所示，首先通过word2vec获得词向量表示，文本矩阵则是由词向量堆叠而成，通过自编码器学习一个文本矩阵的潜在低维表示，使用k-means或谱聚类等聚类方法对学到的低维向量进行聚类。



Autoencoder文本聚类应用

年份	Autoencoder
2019	A self-training approach for short text clustering
2020	Patent document clustering with deep embeddings(DEC)
2020	Semi-supervised Network Embedding with Text Information



A Self-Training Approach for Short Text Clustering

——Amir Hadifar Lucas Sterckx

短文本聚类是一个具有挑战性的问题。当采用传统的bag-of-words或TF-IDF表示时，这将导致短文本的稀疏向量表示。

当采用传统的bag-of-words或TF-IDF表示时，这将导致短文本的稀疏向量表示。本文利用深度聚类，提出从自动编码器和句子嵌入中学习鉴别特征，然后利用聚类算法的分配作为监督来更新编码器网络的权值。

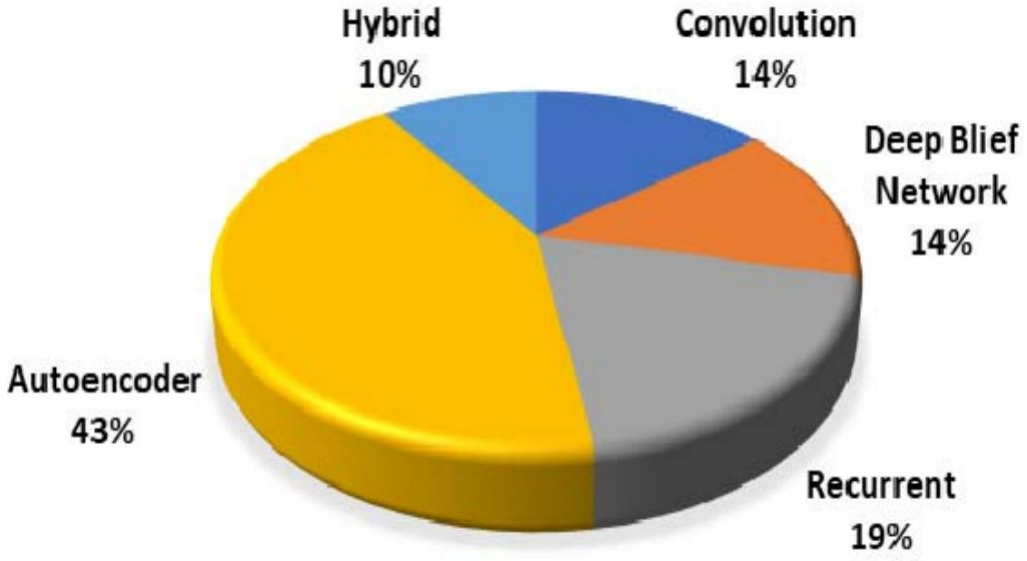
5.1 深度学习方法——趋势

从近5年比较有成效的研究成果来看，自编码器的应用是一个主要趋势

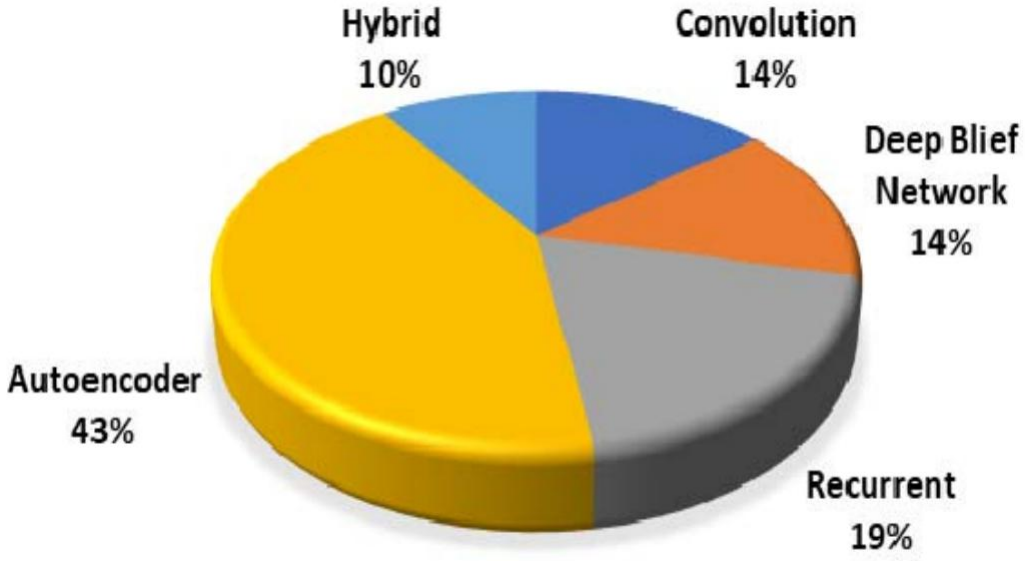
年份	CNN
2015	Short Text Clustering via Convolutional neural networks (STCC)
2016	Semi-supervised clustering for short text via deep representation learning
2017	Self-Taught Convolutional Neural Networks for Short Text Clustering (dubbed STC2)
2019	Deep divergence-based approach to clustering
年份	**RNN**
2018	Comparison of deep learning based concept representations for biomedical document clustering
2018	Text clustering algorithm based on deep representation learning
2019	Sequential embedding induced text clustering, a non-parametric Bayesian approach
2019	ADC: Advanced document clustering using contextualized representations
年份	**DBN**
2015	Deep learning with nonparametric clustering
2016	A personalized Markov clustering and deep learning approach for Arabic text categorization

年份	Autoencoder
2016	Unsupervised deep embedding for clustering analysis
2016	Variational deep embedding: An unsupervised and generative approach to clustering(VaDE)
2017	Deep clustering with convolutional autoencoders
2017	Improved deep embedded clustering with local structure preservation
2017	Towards k-means-friendly spaces: Simultaneous deep learning and clustering
2017	Deepcluster: A general clustering framework based on deep learning(DC-GMM)
2018	Spectralnet: Spectral clustering using deep neural networks
2019	Deep co-clustering
2019	A self-training approach for short text clustering
2020	Patent document clustering with deep embeddings(DEC)
2020	Semi-supervised Network Embedding with Text Information

混合和集成深度学习算法：在大多数研究中，混合深度学习算法利用了每种算法的优势，通常会获得比较出色的效果，胜过其他单独的深度学习算法；下图展示了当前深度学习方法的比例，可以看到，混合算法的研究仍数少数，未来有望出现更多新的成果。



填补空白：还有很多深度学习算法没有被应用到文本聚类这个领域，可以尝试将这些算法扩展到文本聚类领域。例如，深度强化学习（Deep Reinforcement Learning）已经在文本情感摘要等方面有所应用，但是还没有任何方法将其用于文本聚类；此外，目前还很少有工作将预训练的深度模型应用到文本聚类上，并对深度学习方法的有效性作解释。



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

感谢各位专家老师
请您批评指正

德以明理 学以精工