文本分类

玖强

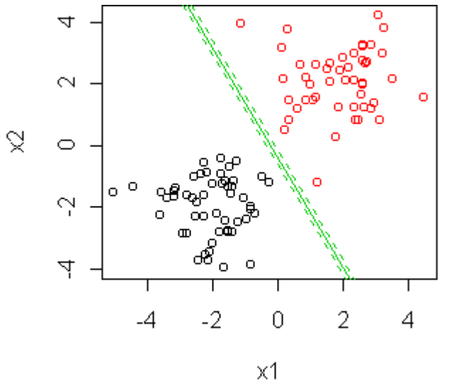
Outline

- 文本分类概述
- 无监督的机器学习算法
- 有监督的机器学习算法

文本分类概述

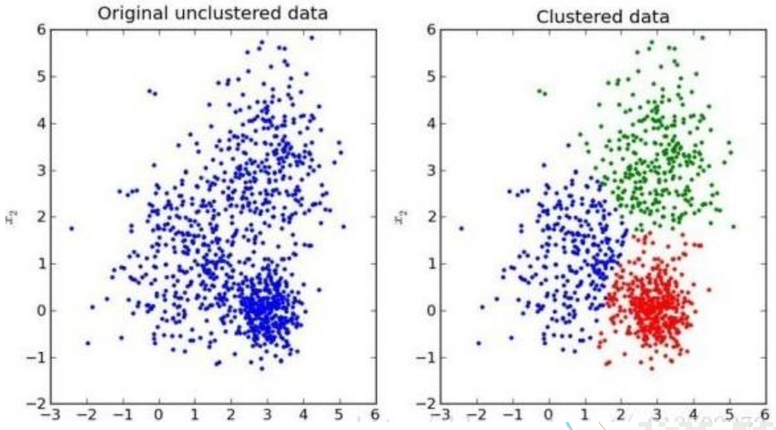
概述

- 给定大量的文档，按照某个Topic主题进行分类：
  1. **Grouping**：根据文档的主题，同一个主题的分为一组，这样称为分组；
  2. **Labeling**：然后，对所有的分组进行标注；每一个这样的分组称为一个类别(class)



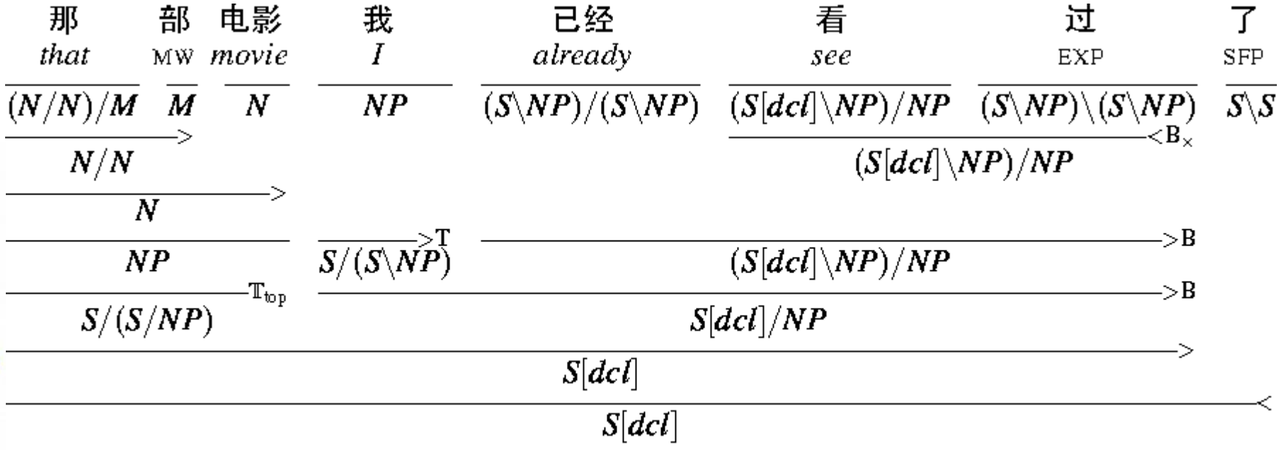
- 文本分类
  - 将文档与所属class label关联的过程
  - Classification/categorization

- 文本聚类(Clustering)
  - 目标：将给定的数据按照一定的相似性原则划分为不同的类别，其中同一类别内的数据相似度较大，而不同类别的数据相似度较小；
  - 可看作分类的一种特殊情况，聚类和分类的区别：
    - 分类是预先知道每个类别的主题，再将数据进行划分；
    - 而聚类则并不知道聚出来的每个类别的主题具体是什么，只知道每个类别下的数据相似度较大，描述的是同一个主题



分类问题的基本方法

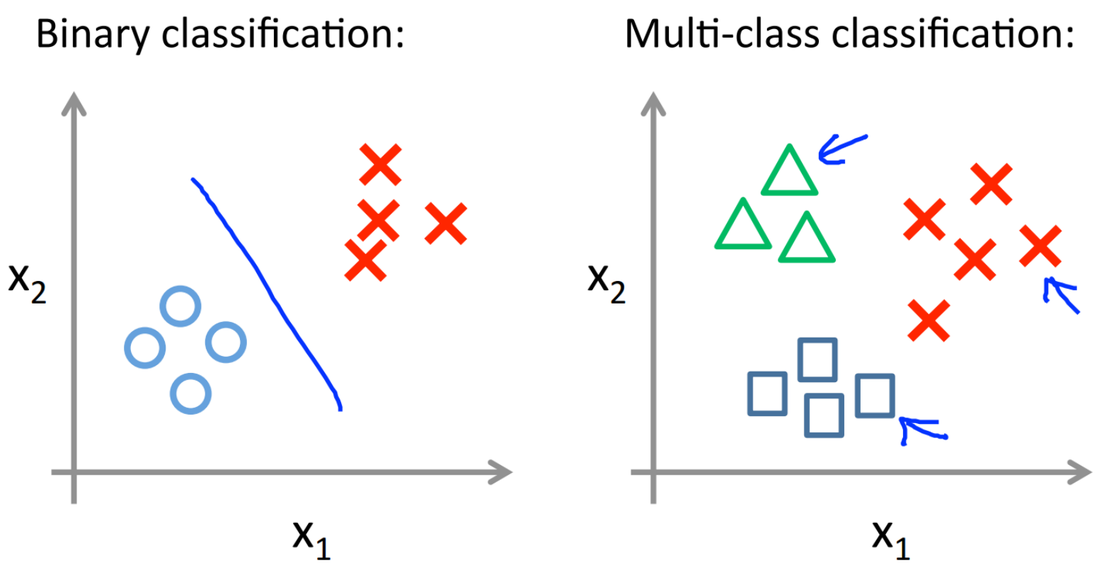
- 从数据中学习特定模式，然后通过得到的模式可以对新数据进行预测
- 这种模式可以是**pre-defined or non-defined**
- 三种类型算法
  - 有监督的学习:有训练数据，使用预定义的“训练示例”集合，训练系统，便于其在新数据被馈送时也能得出结论。系统一直被训练，直到达到所需的精度水平。
    - 标签的获取常常需要极大的人工工作量，有时甚至非常困难
  - 无监督的学习:无训练数据， 它必须自己检测模式和关系。 系统要用推断功能来描述未分类数据的模式。
  - 半监督的学习:训练数据很少



问题定义

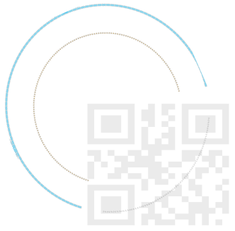
❑ 分类器

- ❑ 数据 D，这里D可以表示为文档的集合
- ❑ 类别 C={$c_1, c_2, \ldots, c_L$}，这里假设L个类别/class,每个类别分配一个label
- ❑ Binary-classification, F=D x C →{0,1}, 即<d_j,c_p>=
  - ❑ 1,如果d_j是类c_p的成员
  - ❑ 0,如果d_j不是类c_p的成员
- ❑ Multi-class classification

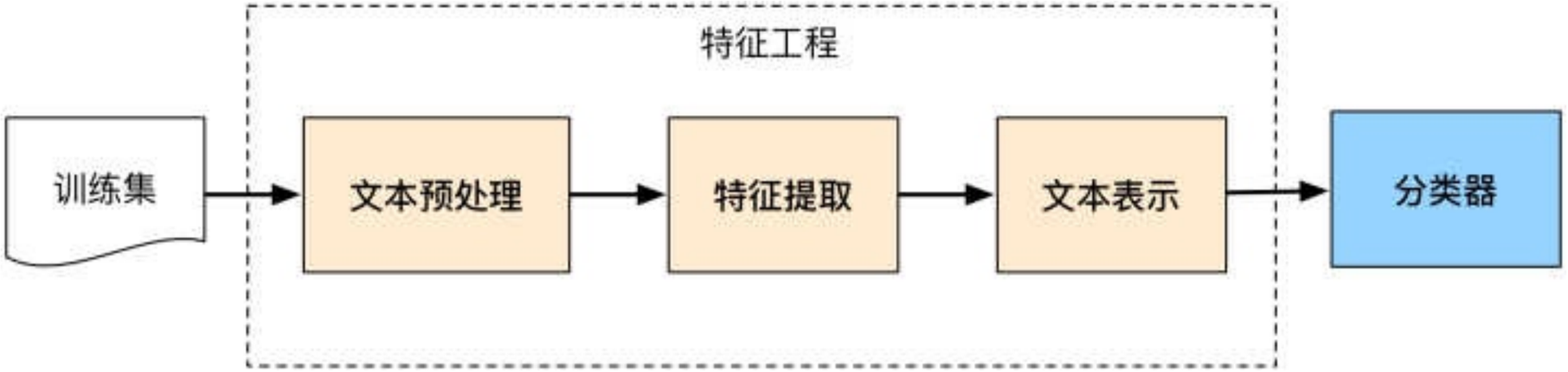


分类的概述

- 定义:给定分类体系,将文本分到某个或者某 几个类别中。
- 分类模式:
  - 二类问题(binary): 一篇文本属于或不属于某一类;
  - 多类问题(multi-class) :一篇文本属于多个类别中的 其中一个类别,多类问题可拆分成两类问题;
  - 一个文本可以属于多类(multi-label)。
- 很多分类体系:
  - Reuters分类体系、中图分类



文本分类过程

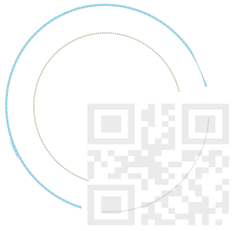


数据预处理

- 文本预处理过程：在文本中提取关键词表示文本的过程
- 中文文本处理中主要包括文本分词和去停用词两个阶段
  - 之所以进行分词，是因为很多研究表明特征粒度为词粒度远好于字粒度

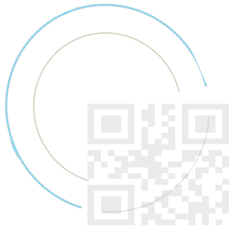
```
>>> seg = jieba.cut("我是一只小小小小鸟")
>>> print seg
<generator object cut at 0x28feaf0>
>>> seg_list = list(jieba.cut("我是一只小小小小鸟"))
>>> print seg_list
[u'\u6211', u'\u662f', u'\u4e00\u53ea', u'\u5c0f\u5c0f', u'\u5c0f\u5c0f\u9e1f']
```

- 例子
  - 夏装雪纺条纹短袖t恤女春半袖衣服夏天中长款大码胖mm显瘦上衣夏
  - 夏装 / 雪纺 / 条纹 / 短袖 / t恤 / 女 / 春 / 半袖 / 衣服 / 夏天 / 中长款 / 大码 / 胖mm / 显瘦 / 上衣 / 夏



文本表示和特征提取

- 文本表示
  - 目的：预处理后的文本转化成计算机可以理解的方式
  - 方法：**Word2Vec**
- 特征提取
  - 向量空间模型的文本表示方法的特征提取对应特征项的选择和特征权重计算两部分
  - 特征选择的基本思路是根据某个评价指标独立的对原始特征项（词项）进行评分排序，从中选择得分最高的一些特征项，过滤掉其余的特征项
  - 特征权重主要是经典的**TF-IDF方法**及其扩展方法，主要思路是一个词的重要度与在类别内的词频成正比，与所有类别出现的次数成反比。



文本表示和特征提取

- **TF-IDF(term frequency–inverse document frequency)**
  - **TF-IDF是一种用于信息检索与数据挖掘的常用加权技术，常用于挖掘文章中的关键词，而且算法简单高效，常被工业用于最开始的文本数据清洗。**
  - **TF-IDF算法步骤：**
    - 第一步，计算词频：

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数}+1}\right)$$

词频

逆文档频率

考虑到文章有长短之分，为了便于不同文章的比较，进行"词频"标准化。

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

    - 第二步，计算逆文档频率：

这时，需要一个语料库（**corpus**），用来模拟语言的使用环境。如果一个词越常见，那么分母就越大，逆文档频率就越小越接近**0**。分母之所以要加**1**，是为了避免分母为**0**（即所有文档都不包含该词）。**log**表示对得到的值取对数。

    - 第三步，计算**TF-IDF**：

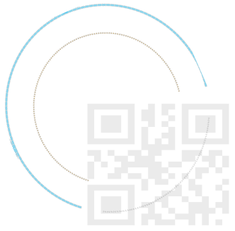
$$\text{TF} - \text{IDF} = \text{词频(TF)} \times \text{逆文档频率（IDF）}$$

文本表示和特征提取

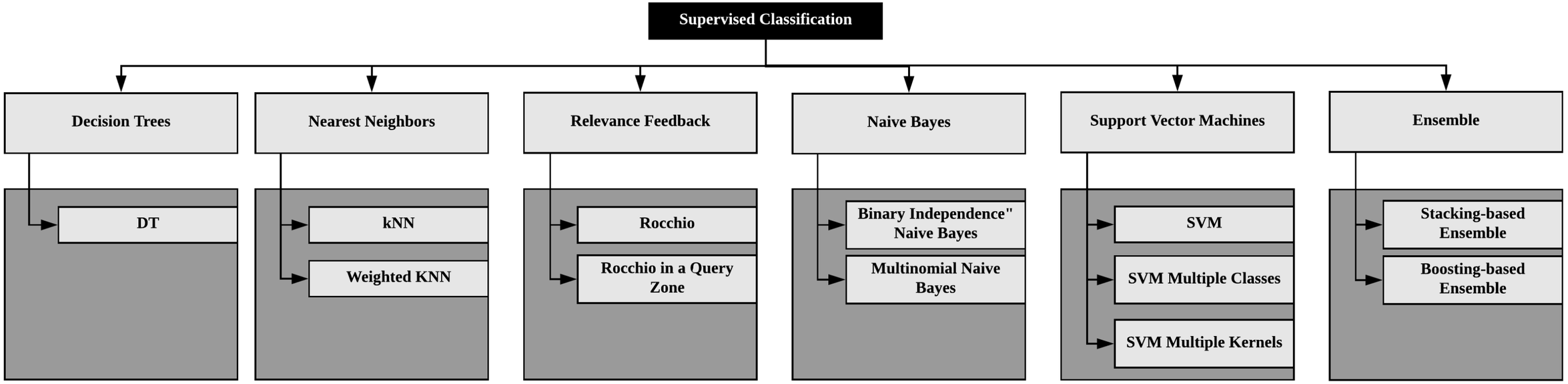
- **TF-IDF(term frequency–inverse document frequency)**
  - 优缺点:
    - **TF-IDF的优点是简单快速，而且容易理解。**
    - **缺点是有时候用词频来衡量文章中的一个词的重要性不够全面，有时候重要的词出现的可能不够多，而且这种计算无法体现位置信息，无法体现词在上下文的重要性。**
  - 解决方法：
    - **Word2vec**
    - **…**

词频

逆文档频率



分类器


Supervised Classification
Decision Trees
DT
Nearest Neighbors
kNN
Weighted KNN
Relevance Feedback
Rocchio
Rocchio in a Query Zone
Naive Bayes
Binary Independence" Naive Bayes
Multinomial Naive Bayes
Support Vector Machines
SVM
SVM Multiple Classes
SVM Multiple Kernels
Ensemble
Stacking-based Ensemble
Boosting-based Ensemble

深度学习文本分类方法

- **传统做法主要问题:**
  - 文本表示是高维度高稀疏性的，特征表达能力很弱;
  - 此外需要人工进行特征工程，成本很高
- **深度学习方法：**
  - 文本表示是核心
  - 利用CNN/RNN等网络结构自动获取特征表达能力

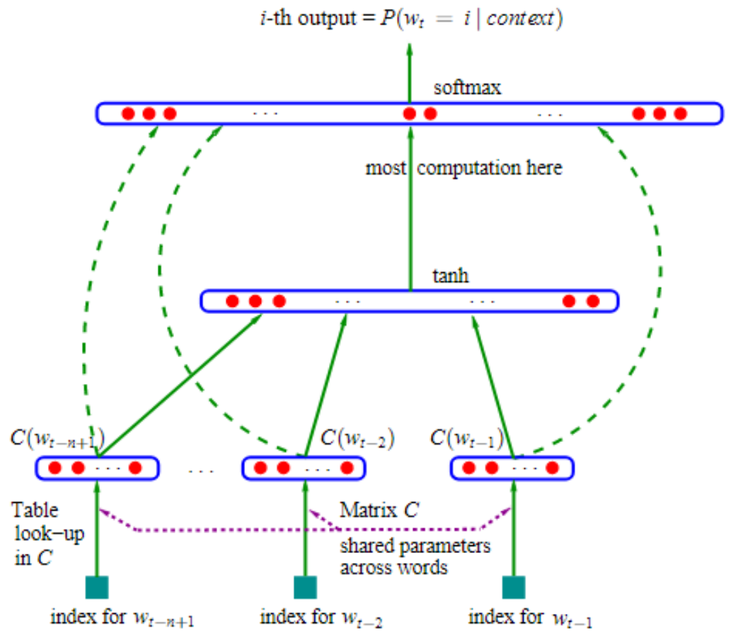
文本的分布式表示：词向量（Word Embedding）

❑ 分布式表示（Distributed Representation）

- ❑ 基本思想是将没歌词表达成N维稠密、连续的实数向量，与之相对的one-hot encoding向量空间只有一个维度是1，其余都是0。
- ❑ 分布式表示最大的优点是具备非常powerful的特征表达能力，比如n维向量每一维k个值，就可以表征k^n个概念了

❑ 神经网络语言模型（NNLM，Neural Probabilistic Language Model），采用的是文本分布式表示，即每个词表示为稠密的实数向量。NNLM模型的目标为

$$f(w_t, \cdots, w_{t-n+1}) = P(w_t|w_1^{t-1})$$



词的分布式表示即词向量（Word Embedding）是训练语言模型的一个附加产物，即图中的Matrix C。

❑ 词向量真正火起来是Word2Vec工具包

- ❑ 文本的表示通过词向量的表示方法，把文本数据从高维度高稀疏性的神经网络难处理的方式，变成了类似图像、语言的连续稠密数据。深度学习算法本身有很强的数据迁移性，很多之前在图像领域很适用的深度学习算法比如CNN等也可以很好的迁移到文本领域了。

深度学习文本分类模型

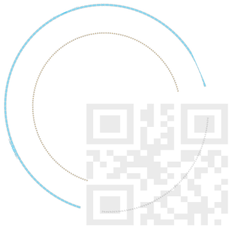
- **FastText**

Joulin, Armand, et al. "Bag of tricks for efficient text classification." *arXiv preprint arXiv:1607.01759* (2016).

Bag of Tricks for Efficient Text Classification

Armand Joulin **Edouard Grave** **Piotr Bojanowski** **Tomas Mikolov**
Facebook AI Research
{ajoulin,egrave,bojanowski,tmikolov}@fb.com

- **还是Mikolov的工作**
- **优点：非常简单！**



深度学习文本分类模型

❑ FastText

Joulin, Armand, et al. "Bag of tricks for efficient text classification." *arXiv preprint arXiv:1607.01759* (2016).

- ❑ 是把句子中所有的词向量进行平均（某种意义上可以理解为只有一个Avg Pooling的特征CNN），然后直接接SoftMax层。
- ❑ 也加入了一些n-gram特征的trick来捕获局部序列信息
- ❑ 看起来好像没什么贡献~~
- ❑ 当然带来思考也是好文章！

$$P(n_{l+1}) = \prod_{i=1}^{l} P(n_i).$$

整个文章唯一单独公式

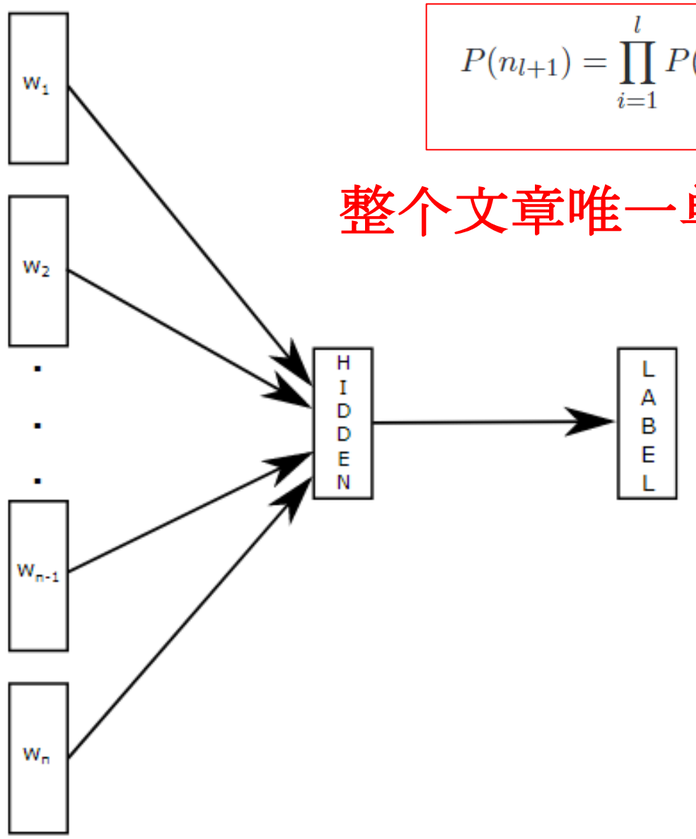


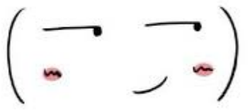
Figure 1: Model architecture for fast sentence classification.

深度学习文本分类模型

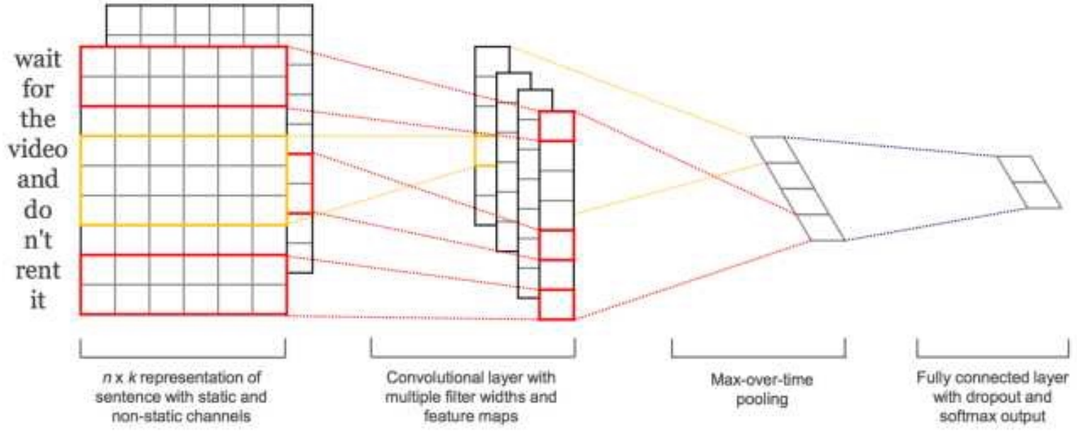
❑ TextCNN

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

- ❑ **FastText中用的n-gram特征Trick恰恰说明了局部序列信息的重要意义!**
- ❑ **从图像分类看，CNN核心在于捕捉局部相关性，具体到文本分类任务重可以利用CNN来提取句子中类似n-gram的关键信息**



Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks." *Pattern Recognition* (2017).



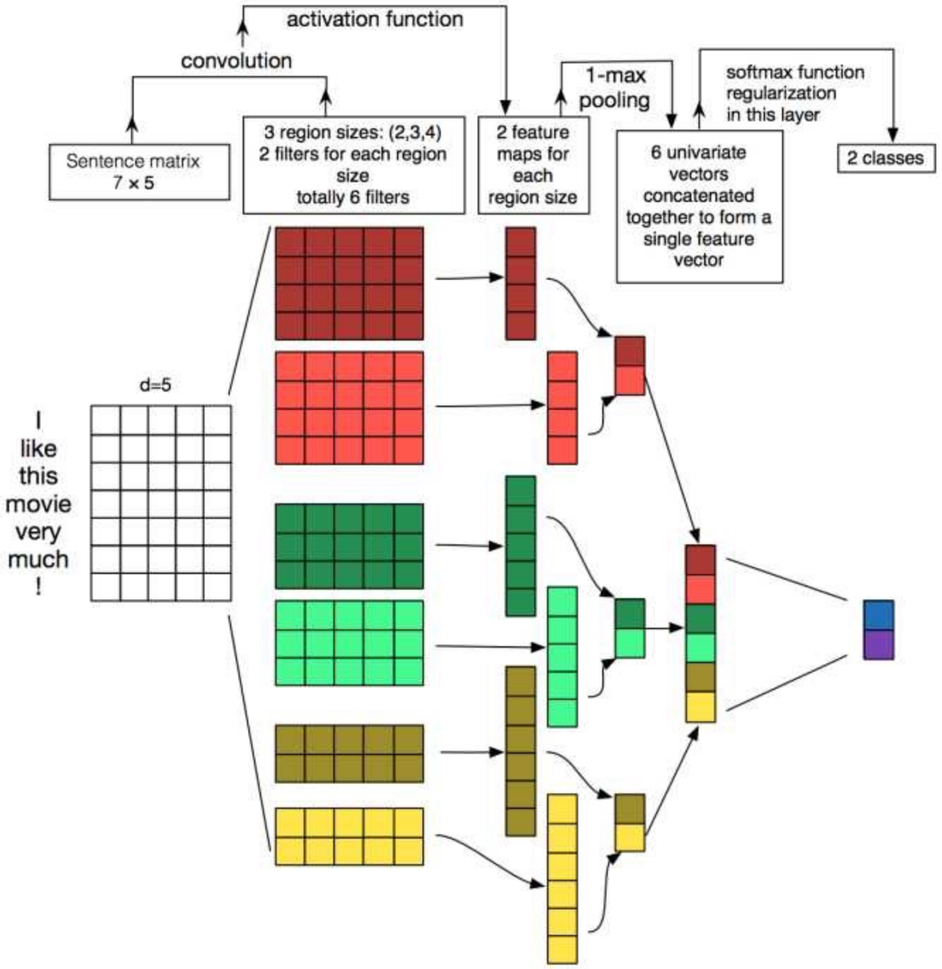
深度学习文本分类模型

❑ TextCNN

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

❑ 解读：

1. **第一层是图中最左边的7乘5的句子矩阵，每行是词向量，维度=5**
2. **然后，经过有 filter_size=(2,3,4) 的一维卷积层，每个filter_size 有两个输出 channel**
3. **第三层是一个1-max pooling层，这样不同长度句子经过pooling层之后都能变成定长的表示了**
4. **最后接一层全连接的 softmax 层，输出每个类别的概率**



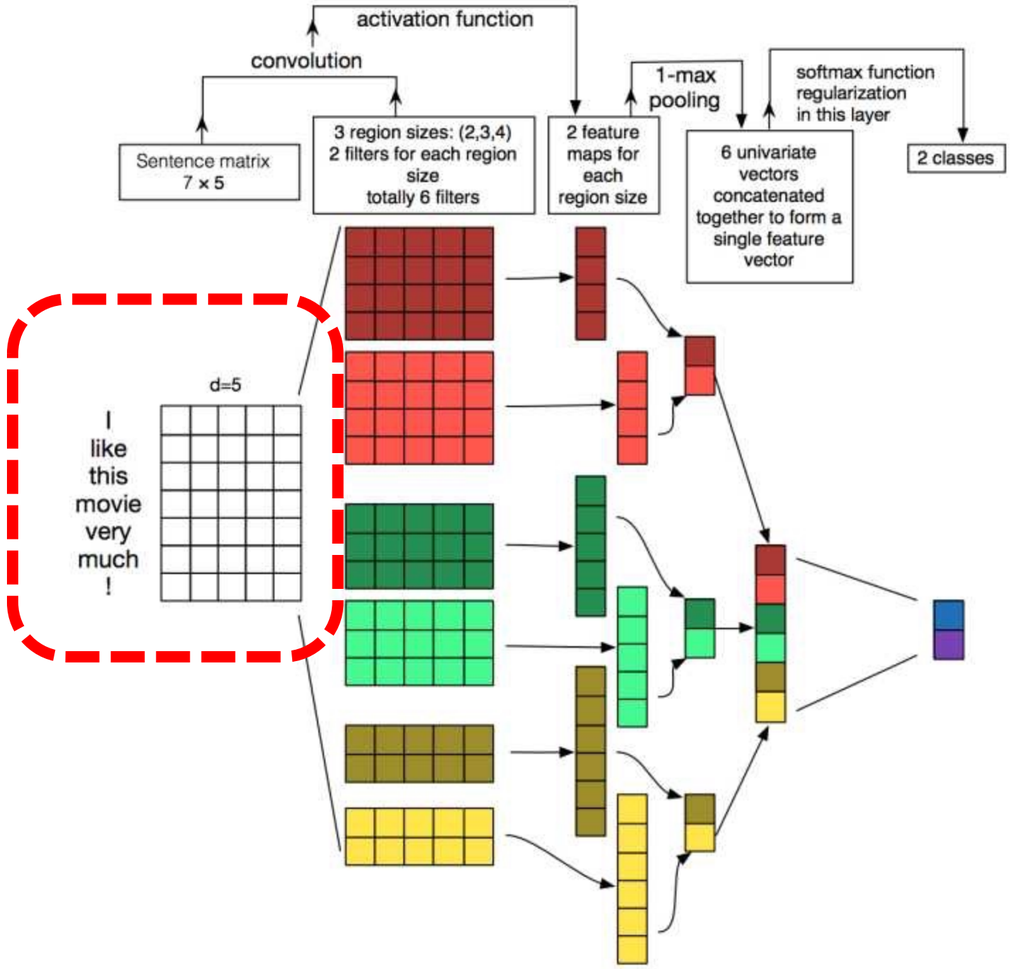
深度学习文本分类模型

❑ TextCNN

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

❑ 特征

- ❑ 有静态（static）和非静态（non-static）方式
- ❑ static方式采用比如word2vec预训练的词向量，训练过程不更新词向量。特别是数据量比较小的情况下，采用静态的词向量往往效果不错
- ❑ non-static则是在训练过程中更新词向量。它是以预训练（pre-train）的word2vec向量初始化词向量，训练过程中调整词向量，能加速收敛。如果有充足的训练数据和资源，直接随机初始化词向量效果也是可以的



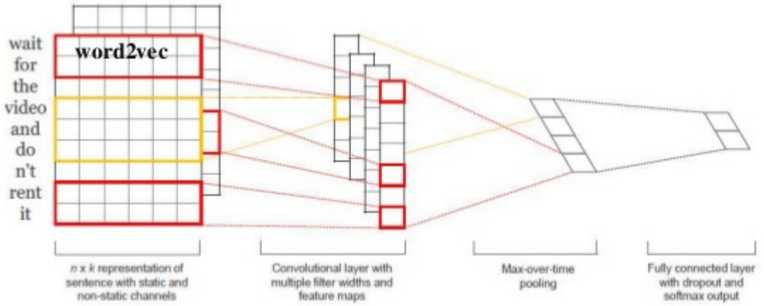
深度学习文本分类模型

❑ TextCNN

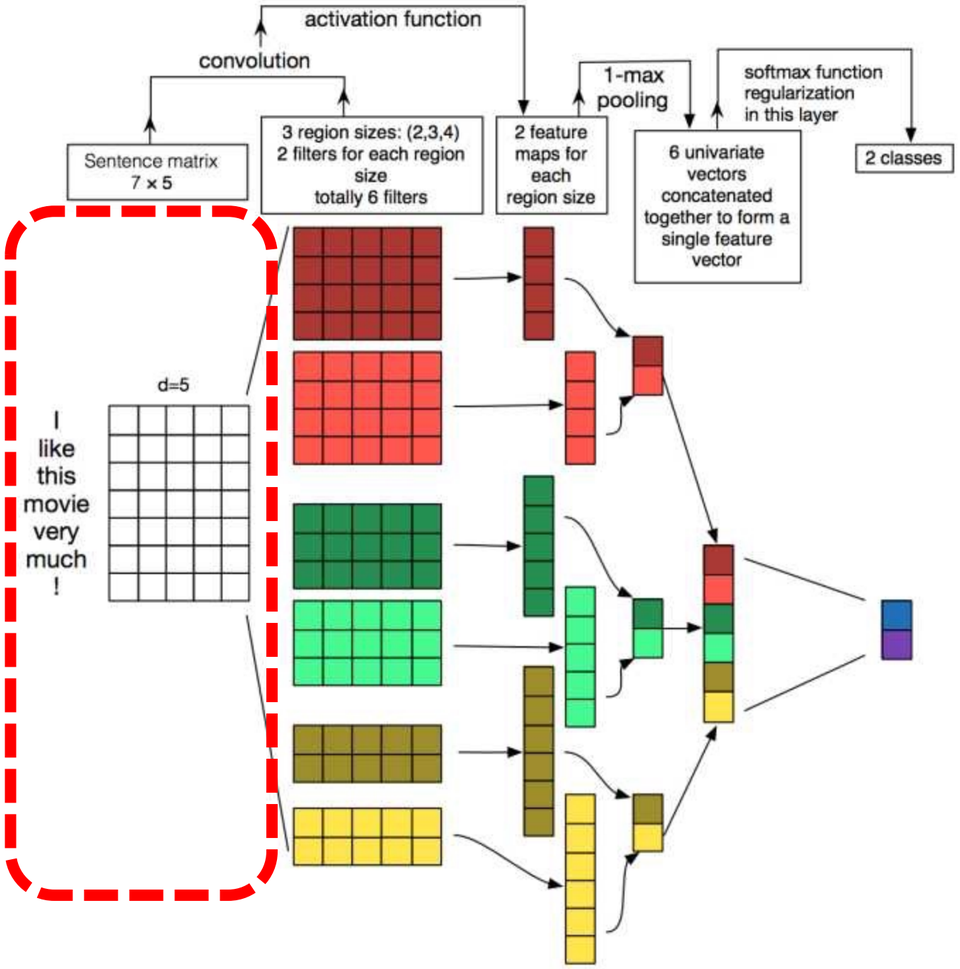
Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

❑ 通道（Channels）

- ❑ 文本的输入的channel通常是不同方式的embedding方式（比如 word2vec或Glove）



- ❑ 类似的图像里面是rgb通道



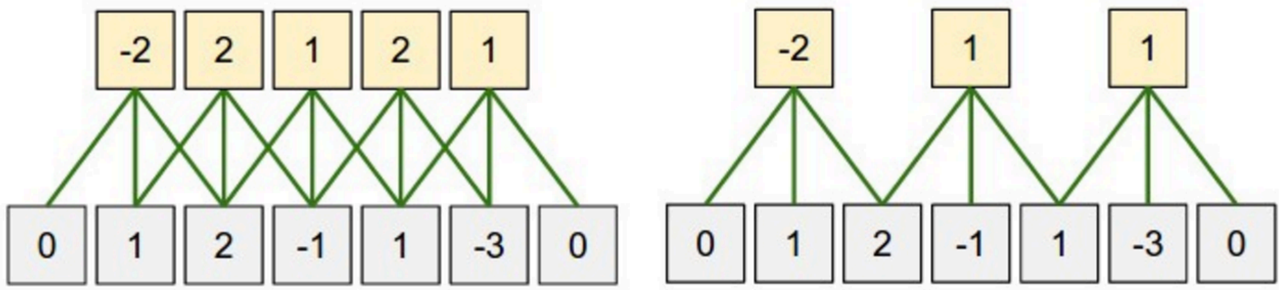
深度学习文本分类模型

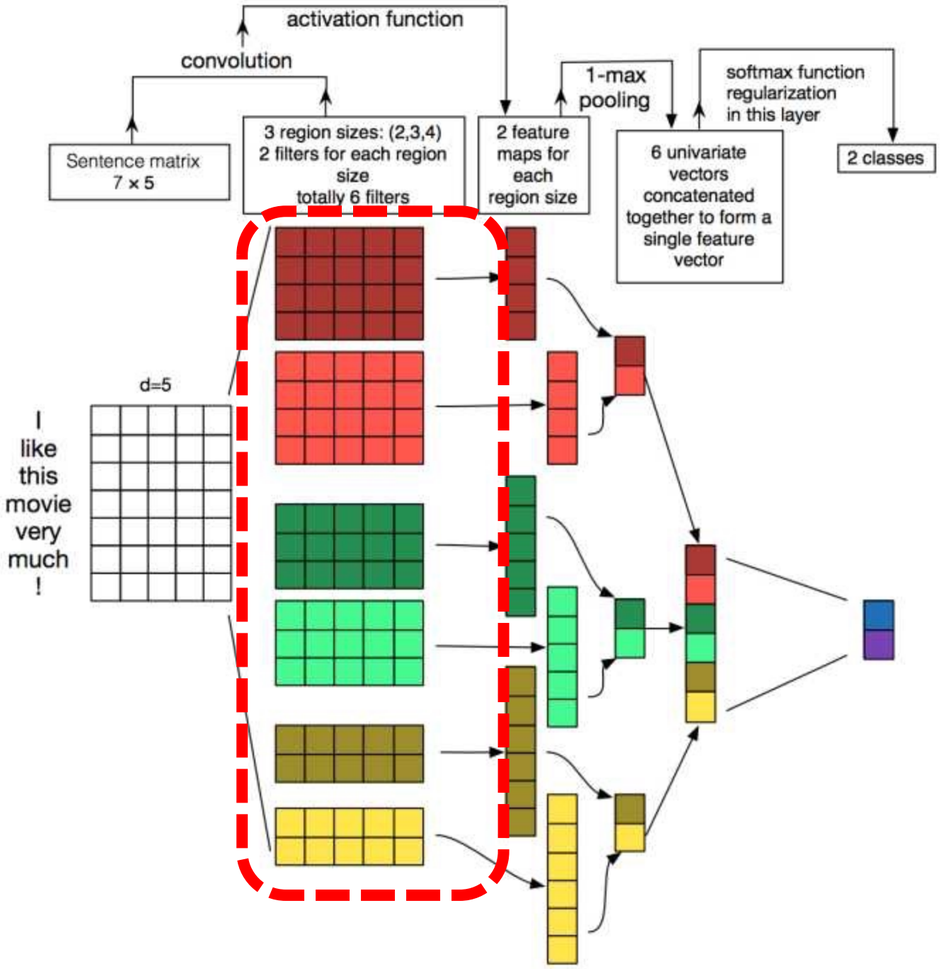
❑ TextCNN

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

❑ 一维卷积（conv-1d）

- ❑ 一维卷积带来的问题是需要设计通过不同 **filter_size** 的 **filter** 获取不同宽度的视野





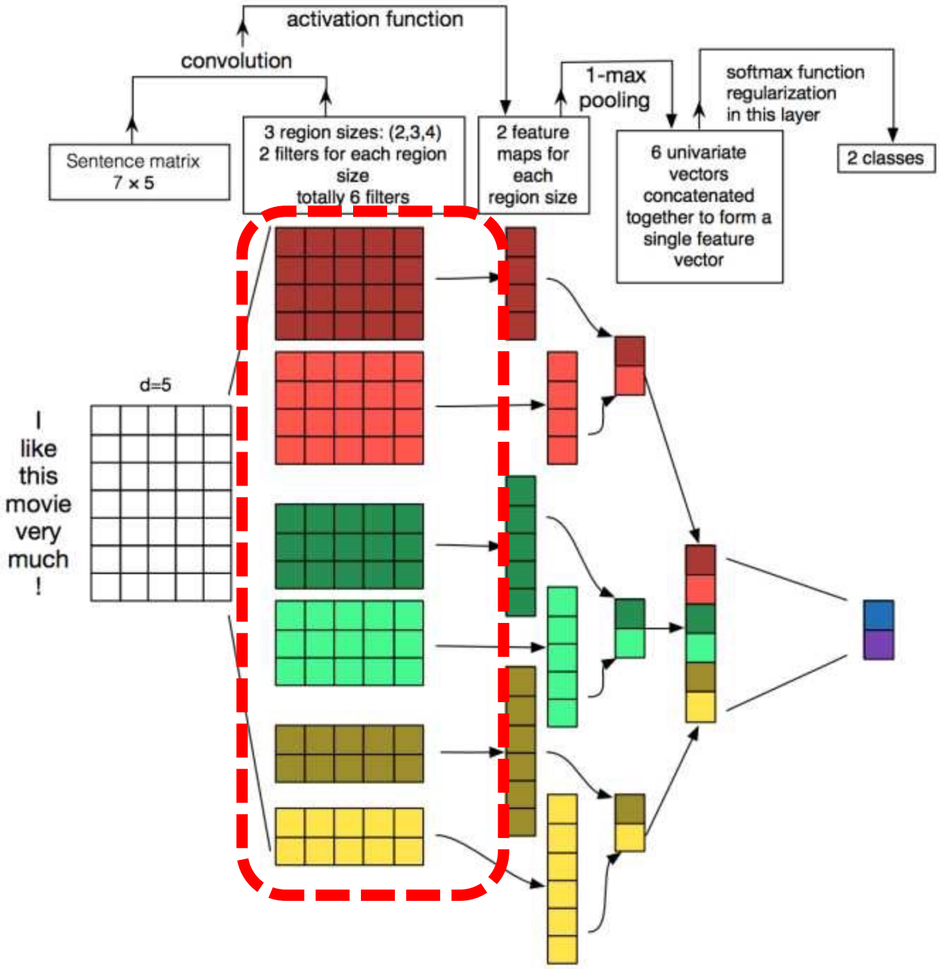
深度学习文本分类模型

❑ TextCNN

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

❑ Pooling层

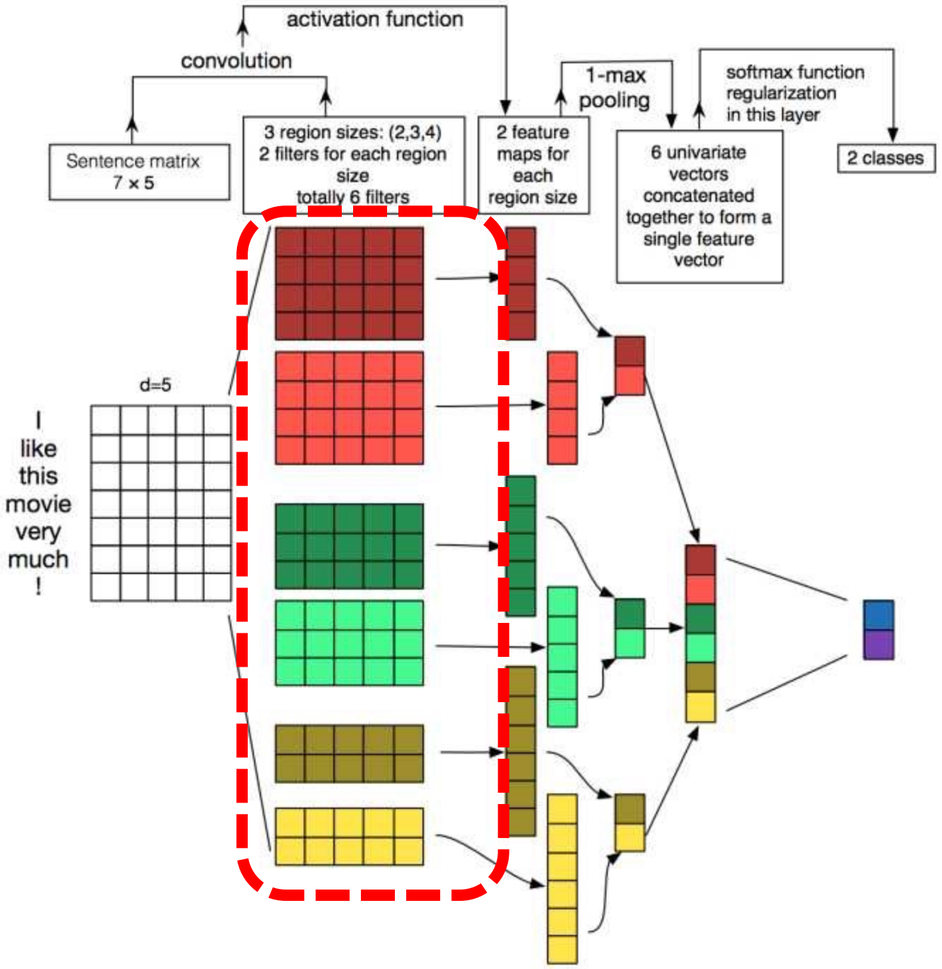
- ❑ **Pooling阶段保留 k 个最大的信息，保留了全局的序列信息**
- ❑ **比如在情感分析场景，举个例子：**
- ❑ **“我觉得这个地方景色还不错，但是人也实在太多了”**
- ❑ **虽然前半部分体现情感是正向的，全局文本表达的是偏负面的情感，利用 k-max pooling能够很好捕捉这类信息。**



深度学习文本分类模型

❑ TextCNN

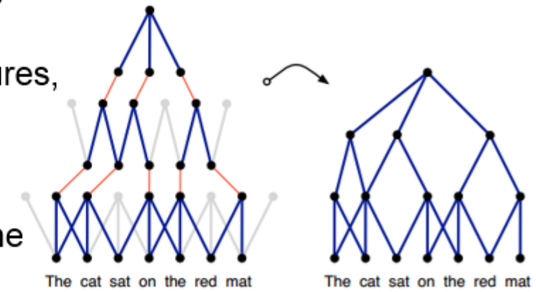
Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).



❑ Better Pooling

❑ K-max pooing

- Select the top k highest values
- Preserve the order of the features, but insensitive to their specific positions
- Discern the number of times the feature is highly activated in c.



❑ Dynamic K-Max Pooling：k的大小与卷积得到的feature map长度、以及当前pooling层数有关，公式如下：

$$k_l = \max(k_{top}, \lceil \frac{L-l}{L} s \rceil)$$

深度学习文本分类模型

❑ TextCNN

Conneau, Alexis, et al. "Very deep convolutional networks for text classification." Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Vol. 1. 2017.

❑ Deeper is better?

Very Deep Convolutional Networks for Text Classification

Alexis Conneau
Facebook AI Research
aconneau@fb.com

Holger Schwenk
Facebook AI Research
schwenk@fb.com

Yann Le Cun
Facebook AI Research
yann@fb.com

Loïc Barrault
LIUM, University of Le Mans, France
loic.barrault@univ-lemans.fr

Depth	Pooling	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
9	Convolution	10.17	4.22	1.64	5.01	37.63	28.10	38.52	4.94
9	KMaxPooling	9.83	3.58	1.56	5.27	38.04	28.24	39.19	5.69
9	MaxPooling	9.17	3.70	1.35	4.88	36.73	27.60	37.95	4.70
17	Convolution	9.29	3.94	1.42	4.96	36.10	27.35	37.50	4.53
17	KMaxPooling	9.39	3.51	1.61	5.05	37.41	28.25	38.81	5.43
17	MaxPooling	8.88	3.54	1.40	4.50	36.07	27.51	37.39	4.41
29	Convolution	9.36	3.61	1.36	4.35	**35.28**	27.17	37.58	**4.28**
29	KMaxPooling	**8.67**	**3.18**	1.41	4.63	37.00	27.16	38.39	4.94
29	MaxPooling	8.73	3.36	**1.29**	**4.28**	35.74	**26.57**	**37.00**	4.31

Table 5: Testing error of our models on the 8 data sets. No data preprocessing or augmentation is used.

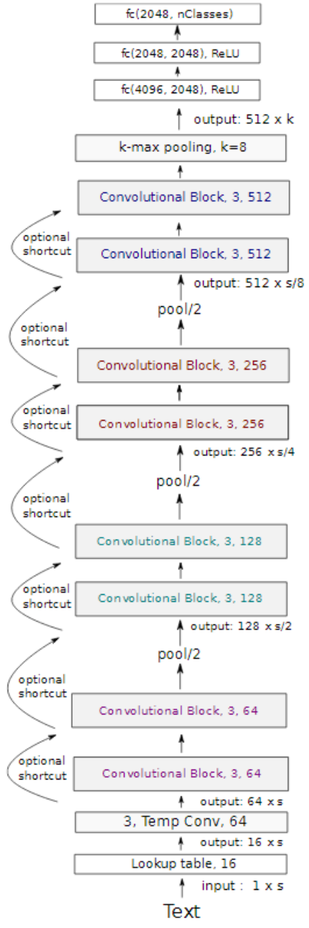
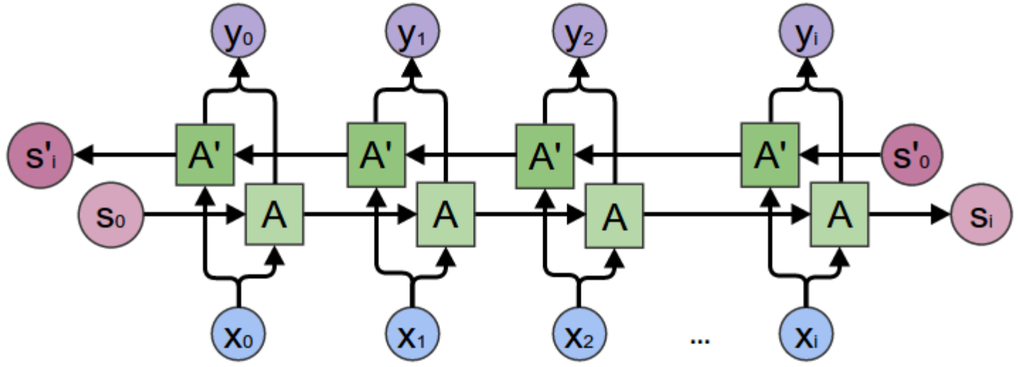


Figure 1: VDCNN architecture.

深度学习文本分类模型

- **CNN问题:**
  1. 固定 **filter_size** 的视野，一方面无法建模更长的序列信息;
  2. 另一方面 **filter_size** 的超参调节也很繁琐
- **CNN本质是做文本的特征表达工作**
- **而自然语言处理中更常用的是递归神经网络（RNN, Recurrent Neural Network），能够更好的表达上下文信息。**
  - **E.g., 在文本分类任务中，Bi-directional RNN（实际使用的是双向LSTM）从某种意义上可以理解为可以捕获变长且双向的的 “n-gram” 信息。**



深度学习文本分类模型

TextRNN

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. "Recurrent neural network for text classification with multi-task learning." arXiv preprint arXiv:1605.05101 (2016).

- **利用最后一个词的结果直接接全连接层softmax输出**
- **一个改进是：ctc**
  - **CTC（connectionist temporal classification）是 sequence-to-sequence learning 的一个重要里程碑。它为 RNN 设计了一种新型的目标函数，使得输入、输出序列不必等长，也不需要在训练前对输入、输出序列进行对齐。**
  - **是一种改进的RNN模型. CTC解决不等长的方法是，在标注符号集中加一个空白符号blank，然后利用RNN进行标注，最后把blank符号和预测出的重复符号消除。比如有可能预测除了一个"--a-bb"，就对应序列"ab"。这样就让RNN可以对长度小于输入序列的标注序列进行预测了。**

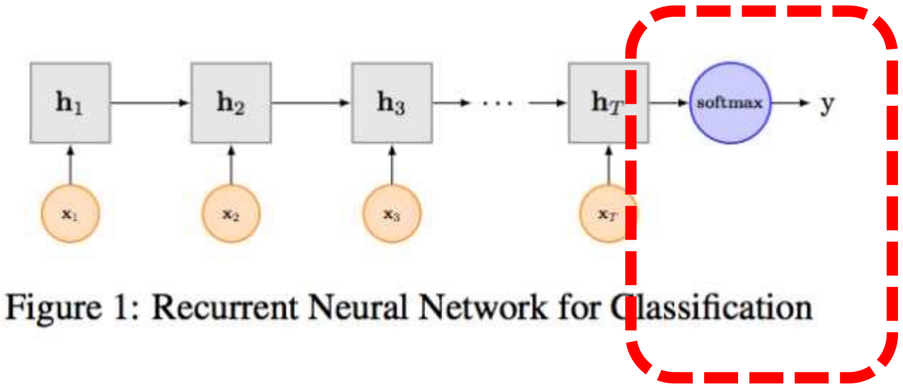


Figure 1: Recurrent Neural Network for Classification

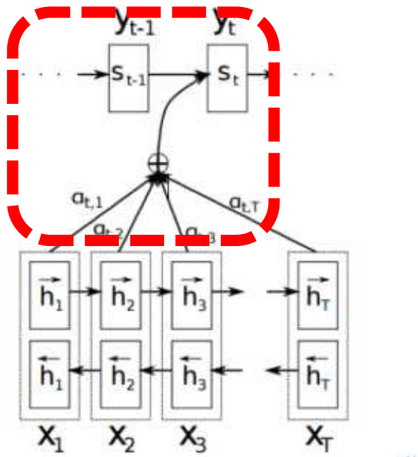
《Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks》Alex Graves
《EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding》Yajie Miao

深度学习文本分类模型

❑ TextRNN + Attention

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

- ❑ 注意力（**Attention**）机制是自然语言处理领域一个常用的建模长时间记忆机制，能够很直观的给出每个词对结果的贡献
- ❑ 基本成了**Seq2Seq**模型的标配了，文本分类从某种意义上也可以理解为一种特殊的**Seq2Seq**，所以考虑把**Attention**机制引入近来
- ❑ **Attention**的核心是在翻译每个目标词（或 预测商品标题文本所属类别）所用的上下文是不同的，这样的考虑显然是更合理的。

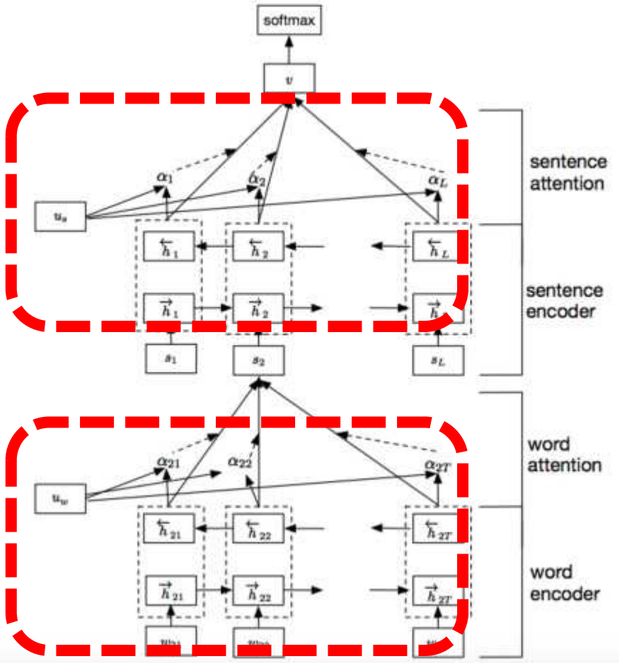


深度学习文本分类模型

❑ TextRNN + Attention

Yang, Zichao, et al. "Hierarchical attention networks for document classification." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

❑ **加入Attention之后最大的好处自然是能够直观的解释各个句子和词对分类类别的重要性。**

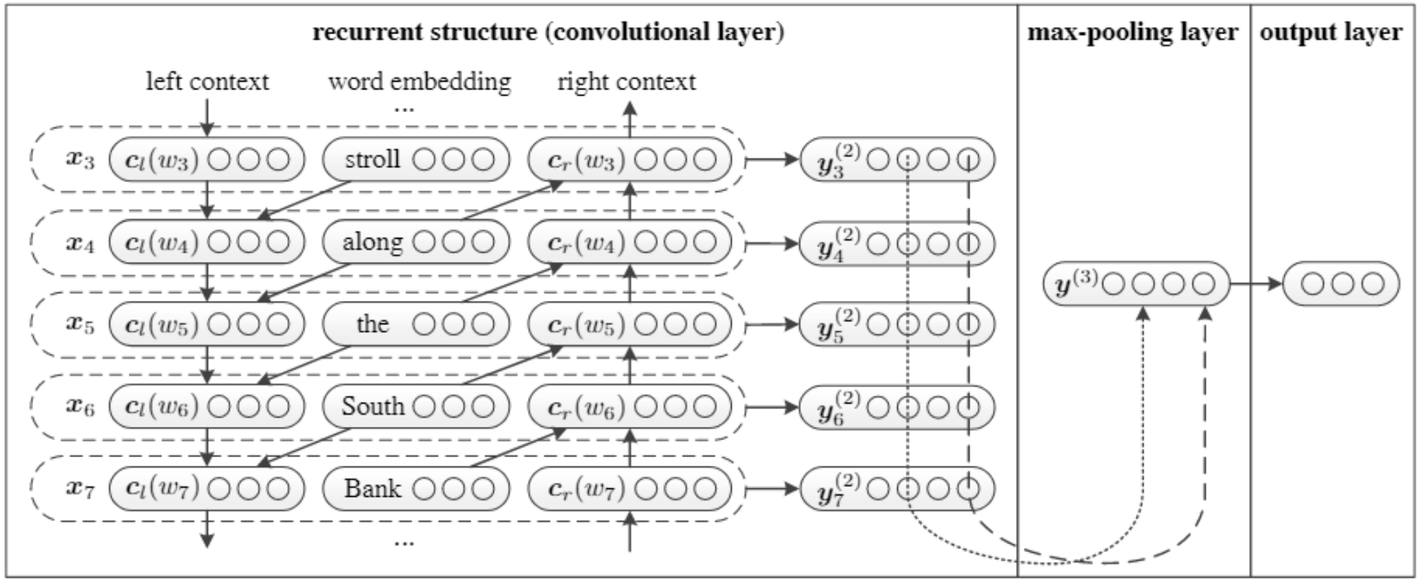


深度学习文本分类模型

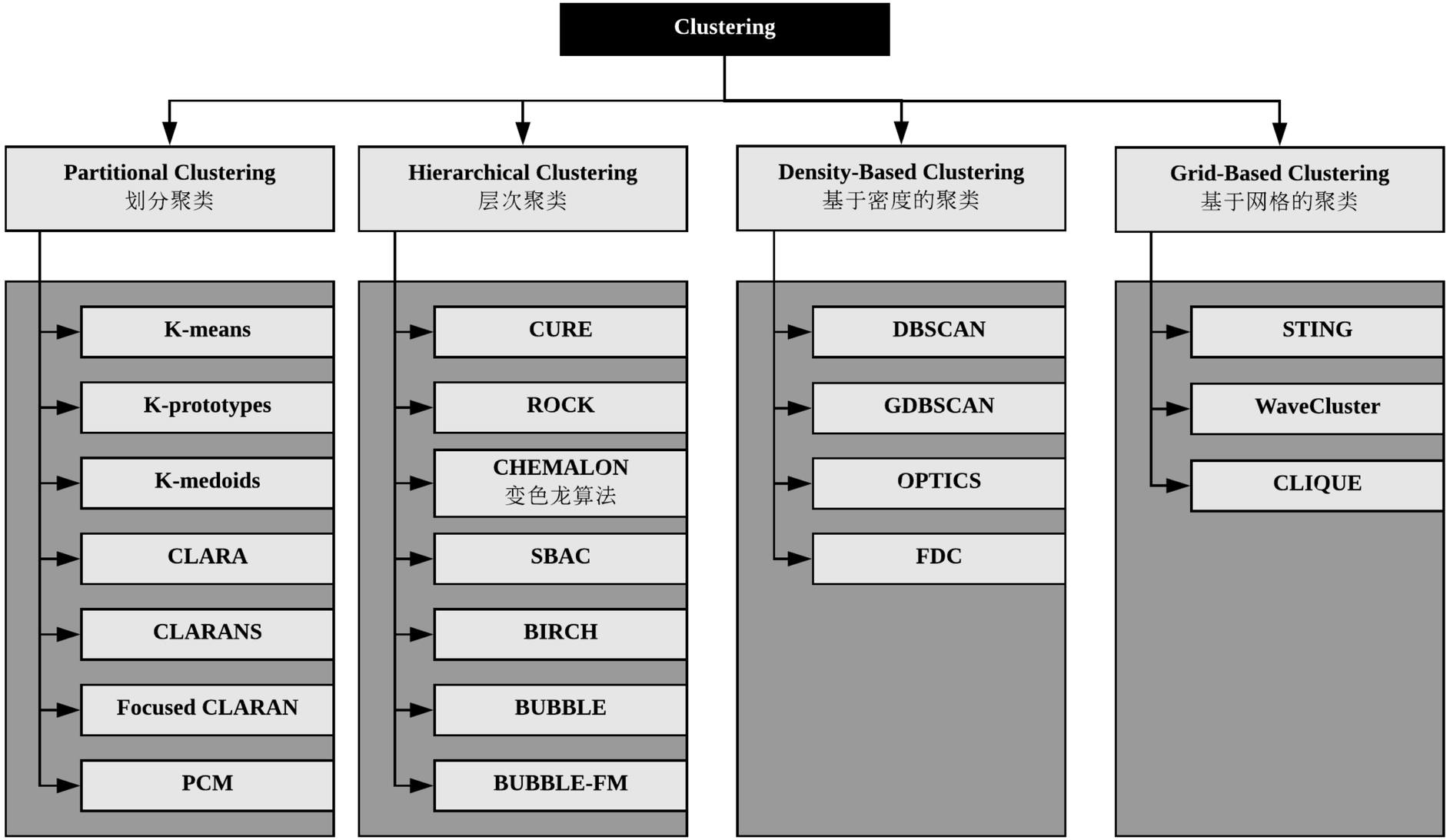
❑ TextRCNN（TextRNN + CNN）

Lai, Siwei, et al. "Recurrent Convolutional Neural Networks for Text Classification." AAAI. Vol. 333. 2015.

- ❑ 利用前向和后向**RNN**得到每个词的前向和后向上下文的表示
- ❑ 这样词的表示就变成词向量和前向后向上下文向量**concat**起来的形式
- ❑ 最后再跟**TextCNN**相同卷积层，**pooling**层即可，唯一不同的是卷积层 **filter_size = 1**就可以了，不再需要更大 **filter_size** 获得更大视野，这里词的表示也可以只用双向**RNN**输出。



无监督的分类算法

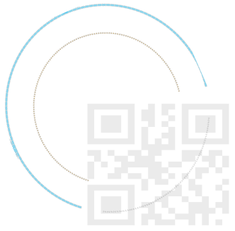

Clustering
Partitional Clustering
划分聚类
Hierarchical Clustering
层次聚类
Density-Based Clustering
基于密度的聚类
Grid-Based Clustering
基于网格的聚类
K-means
K-prototypes
K-medoids
CLARA
CLARANS
Focused CLARAN
PCM
CURE
ROCK
CHEMALON
变色龙算法
SBAC
BIRCH
BUBBLE
BUBBLE-FM
DBSCAN
GDBSCAN
OPTICS
FDC
STING
WaveCluster
CLIQUE

K-MEANS算法

- 从D中随机选择k个初始参照点
- 以此参照点作为质心,对D进行划分 • 然后重新计算Cluster的质心
- 对D重新划分
- 重复以上过程,直到质心不再改变

Bisecting K-Means

- K=2
- (1) 所有文档归为一个cluster, 并将其用k -means算法划分成两个聚类
- (2)选择当前误差平方和(SSE)最大的cluster,利用2-means对其进行划分
- (3)如果当前所有的cluster中文档的数 量<=s, s为指定的阈值,算法结束;否则 ,转到第(2)步继续处理



K-MEANS算法种子的选择

- 聚类结果与初始种子节点的选择是相关的
- 随机选择的种子可能会导致收敛很慢或者收敛到局部最优
- 采用启发式方法或其他方法选择好的种子

END