文本分类与聚类B组综述报告

刘兆友 王朝阳 马宇林 刘俊艳

摘 要

近些年来，在信息技术和互联网的髙速发展下，电子文本信息的数目迅速增长人们可获得的信息越来越多。然而由于信息的杂乱无序，人们很难在浩餘的数据中找到真正需要的信息。面对我们常说的“信息发达，知识贫乏”这样的局面，如何尽最大可能对这些信息进行有效的组织和管理成为信息处理研究中重要的研究之一，文本分类与聚类技术就是解决这一问题的有效方法。能够帮助人们非常髙效并且准确的定位文本信息，为用户取得需要的信息提供强大的支持。

本文详细介绍了文本分类与聚类的主要方面，涉及文本表示、文本分类算法等相关技术以及文本聚类算法等相关技术。

**关键词：**文本分类；文本聚类；特征表示；分类算法；聚类算法

Abstract

In recent years, with the rapid development of information technology and the Internet, the number of electronic text messages has increased rapidly and people have more and more information available. However, due to the chaotic information, people find it hard to find the information they really need. Faced with what we often say "information is developed, lack of knowledge", how to organize and manage these information to the greatest extent possible becomes one of the most important research in information processing. Text classification technology is to solve this problem Useful ways. Can help people to be very efficient and accurate positioning of text messages, to provide users with the necessary information to provide strong support.

This paper introduces the main aspects of text classification in detail, including text representation, text feature extraction methods, text classification algorithms and other related technologies and text classification related applications.

**Keywords:** Text Classification; Feature Representation; Feature Selection; Classification Algorithm; Applicati

# **引** **言**

在信息技术快速发展的步伐中，我们进入了信息时代，信息借助于各种技术平台作为载体，以多种形式呈现在人们面前，网络信息已经渗透到人们日常生活的各个方面，如电子邮件、网络论坛、个人博客、微博、数字化图书馆、搜索引擎等。它们不仅改变了人们的行为方式和思维模式，影响着人们的衣、食、住、行，同时也为人们的决策提供着重要的参考信息和依据。其中涉及到的信息量呈指数级增加，信息的形式包括纯文本、音频、视频、图像、网页文本等。

随着Internet的迅猛发展以及计算机处理能力的不断提高，例如微博，社区以及SNS的兴起塑造了全新的社会生活形态，使得越来越多的用户更加偏爱在互联网和移动平台来抒发自己的情感、观点和评论。在“互联网+”行动计划不断助力企业发展的趋势下，互联网对于整体社会的影响己进入到新的阶段。用户在这些新兴的社交网络与传统的门户网站的参与和交互行为的激增，加速了文本数据的规模效应，进而导致大量的电子信息成几何级数增长，使得从“信息贫乏”时代转变为“信息过载”时代。人们所面对的问题不再是如何获取信息，而是如何从海量的信息中快速有效的提取信息。因此，如何有效的管理，过滤和筛选这些信息，成为学术界和企业界一项重要的研究课题。

伴随着信息数据急剧增长，信息表示形式也是多样化的，主要包括文本、声音和图像。其中文本数据与声音和图像相比，占用网络资源少，更容易上传和下载:并且，文本是信息的主要载体，其它形式的信息都可以通过文本标注来表示，使得文本信息在网络资源中占据主导地位。面对浩瀚的信息海洋，如何快速准确的获取用户所需的信息成为人们的迫切需要。传统通过人工手段对海量原始文档进行标注和自动分类，不仅费时费力，而且还不能达到理想的效果，己经无法适应信息社会对爆炸式增长的数字信息的管理需求。因此，如何有效的把准确的文本信息反馈给用户，使用户准确快速的获得所需信息成为关注的焦点。

文本分类是实现人机自由交互、推动人工智能发展的关键技术，可以在较大程度上处理和解决信息杂乱的问题，方便用户准确地定位所需的信息。文本分类技术的研究对于信息的高效管理和有效利用都具有极其现实的意义，成为解决文本信息管理的关键手段。其中，文本表示是文本分类的基石，因为需要将文本转化为己有算法能处理的形式，因此文本表示的准确度很大程度上将直接决定了自然语言处理任务的结果表现。然而文本向量化表示方法有着固有的缺点，传统分类方法的理论假设和实际情况相比存在较大差异，导致在现实应用中难以达到预期性能。因此，文本表示方法与文本分类技术的研究成为信息处理领域的一个前沿课题，有着重大的理论价值和现实需求。

聚类分析是数据挖掘中的一个重要研究领域。聚类是根据数据的不同特征,将其划分为不同的数据类。它的目的是使得属于同一类别的个体之间的距离尽可能的小,而不同类别上的个体间的距离尽可能的大。那么文本聚类到底用处有多大呢?

①文档聚类可以作为多文档自动文摘等自然语言处理应用的预处理步骤,比较典型的例子是哥伦比亚大学开发的多文档文摘系统Newsblaster[20]。能将每天的报文文本进行聚类处理,并按主题进行冗余消除、信息融合等智能处理,为用户的浏览提供方便;

②对用户感兴趣的文档聚类,能够挖掘用户的兴趣模式以用于信息过滤和主动推荐等信息服务。对搜索引擎返回的结果进行聚类,然后按目录树的形式提供给用户,可以使用户快速找到所需要的信息[21];

③聚类技术还可以用来改善文本分类的结果或者找出潜在的主题[22];

④数字图书馆服务。通过自组织映射神经网络等方法[23],可以将高维空间的文档结构映射到二维空间,使得聚类结果可视化和便于理解,如SOMlib系统;

⑤文档集合的自动整理。如对个人邮件进行分类,对个人短信息自动分类处理等。而微软的Ji-RongWen等人则利用聚类技术对用户提出的查询记录进行聚类,利用聚类结果来更新网站

# 文本分类与聚类概述

## 引言

文本自动分类最初是应信息检索IR系统的要求而出现的。随着全球互联网络的普及，文本自动分类对于信息处理的意义变得更加重要。在互联网上电子文档信息每天都在急剧增加，通过网络人们可以很方便地共享巨大的信息资源，但是网络信息的快速膨胀，信息资源无法有效利用。面对网上的海量信息，传统的做法，是对网上信息进行人工分类，并加以组织和整理，为人们提供一种相对有效的信息获取手段。但这种人工分类的做法存在着许多弊端：一是耗费大量的人力、物力和精力；二是分类结果一致性不高。即使分类人的语言素质较高，对于不同的人来分类其分类结果仍然不尽相同，甚至同一个人在不同时间做分类也可能会有不同的结果。网络信息的激增一方面增加了对于快速、自动文本分类的迫切需求。另一方面又为基于机器学习的文本分类方法准备了充分的资源电子化信息的自动分类处理技术正越发显示着其优越性，文本自动分类及其相关技术的研究也正日益成为一项研究热点。

## 文本分类概述

文本分类指的是用电脑对文本集(或其他实体或物件)按照一定的分类体系或标准进行自动分类标记。一个文本分类问题就是将一篇文档归入预先定义的几个类别中的一个或几个，而文本的自动分类则是使用计算机程序来实现这样的分类。

这里所指的文本可以是媒体新闻、科技、报告、电子邮件、技术专利、网页、书籍等。通俗来说，就好比你拿一篇文章，问计算机这篇文章要说的是什么，它需要计算机自动判断该文章所属于预先定义好的标签类别中的哪一类（是体育，经济还是教育等）。文本分类是自然语言处理(Natural language processing,NLP)领域的主要研究方向之一。

文本分类的方法主要分为两类:基于规则的分类方法和基于统计的分类方法。其中，基于规则的分类方法多需要该领域的知识、规则库作支撑，但是往往规则的制定以及更新的受限使得这种方法应用比较窄，更适合应用于某一具体领域。基于统计的学习方法一般是在训练集上依据某种统计或者采用某种统计学知识或定律，通过对样本统计和计算，建立相应的数据模型学习参数，并完成分类器的训练。在测试阶段，根据这些参数对待测样本预测类别。

目前，大量的基于统计的机器学习方法被应用于文本分类系统中，应用最早的机器学习方法是朴素贝叶斯(Naive Bayes，NB)。随后，几乎所有重要的机器学习算法在文本分类领域都得到了应用，比如K近邻算法(K Nearest Neighbor. KNN)、支持向量机(Support vector machine. SVM)、神经网络(Neural Nets)、最小二乘和决策树等。其中，SVM的应用是文本分类近几年来最重要的进展之一。SVM使用浅层线性模式分离模型，当不同类别的数据向量在低维空间无法划分时，SVM通过核函数映射到高维空间并寻找分类最优超平面。Bayes、线性分类、决策树以及KNN等方法的能力相对较弱，但是模型简单，效率也比较高，也有一部分研究学者基于这些方法进行修正和改进。

由于利用多个学习器可以取得比单一学习器更好的性能，因此很多学者试图通过增加学习器的数目来提高泛化能力。值得一提的是，南京大学周志华等人提出了选择性集成理论，证明了从一组学习器中选择部分学习器可以比所有学习器构建的集成学习系统更优越，在解决文本分类应用上取得了不错的效果。

## 文本聚类概述

聚类分析是一种无指导的机器学习方法，在机器学习、统计分析、模式识别、数据挖掘、生物学等许多领域得到了广泛的研究与应用。聚类的基本目的是将数据对象按照一定的标准分成若干个簇，使得同一个簇中的对象之间的相似度较大，不同簇之间的对象相似度较小。

文本聚类是在传统的聚类分析的基础上发展而来的，文本数据是半结构数据，这使得基于结构化数据的聚类算法不适用于文本聚类。 具体的聚类方法有基于层次的、基于划分的、基于密度的和基于网格的等。

文本聚类技术是提高搜索引擎性能的一种有效的方法。作为一种无监督的机器学习方法,聚类由于不需要训练和预先对文档标注类别,因此有一定的灵活性和较高的自动化处理能力,已经成为对文本信息进行组织的手段,被研究和应用于自然语言处理、Web挖掘等相关领域。而且在本体学习中的概念关系获取领域,文本聚类也是一种重要的识别概念关系的方法。文本聚类技术已经研究了有四十多年,随着互联网的不断发展和信息的不断膨胀,文本聚类技术的研究越来越受到重视,其应用领域不断的发展和扩大,将会成为人工智能领域一个重要的研究课题。许多国内外的学者不断投身到这项研究中,各种成果也不断涌现。

# 文本表示

这一章具体介绍了文本分类中流程中的每一个步骤。

## 文本的预处理

机器学习算法要求将文本的单词序列转换为数值表示的一个矢量。首先我们假定每个文本是单词的无序集合(Set)，不考虑单词之间的相互位置，这样一个单词代表哪一维是随机的，文本不再是一个序列，而是属性－值的集合。在给出文本中每个属性（单词）的值之前，需要对文本集进行预处理：

（1）处理文本标记

文本中存在着一些标记，例如数字、标点符号等，这一类标记往往不具有实际意义，也不能作为区分文本类别的依据，而且在某些文本分类问题中，它们的存在反而会给文本分类带来干扰，例如在垃圾邮件分类中，垃圾邮件中经常会夹杂特殊符号，从而逃避分类器的识别。所以需要将它们删除，从而避免干扰分类器性能，同时也可以减少分类负担。

（2）分词处理

对于英文文档而言，词与词之间的分隔是通过特定的间隔标记符号实现的，例如空格和标点符号等。所以遍历文档，就能够实现英文文档的分词，并获得单词列表。而对于中文文本，分词处理是一个很重要的研究方向，现在已经出现了很多中文分词方法。

（3）提取词干

对于英文单词来讲，经常会因为加了前缀或后缀而产生多种不同的形式，但是它们的含义往往是相似的。例如：英文单词“protecting”，“protected”，“protects”和“protection”，对它们进行词干化处理后，得到其词干为“protect”。对于词干提取操作，其过程在于将单词还原为其基本形式，而只保留词干。该操作可以统一单词的形式，从而减少冗余单词，降低单词数，为文本分类的后续处理节省计算成本。在提取词干的方法中，由 Dr.Martin Porter提出的波特词干器（Porter Stemmer），是目前使用最为广泛的词干提取算法之一，通常称之为波特词干算法。从本质上来讲，词干化处理可以减少特征间的依赖性，提高特征间的独立性。

（4）移去停用词

停用词（Stop Words）经常出现在文档中，却没有具体的实际意义。在英文文档中如“a”、“the”、“am”等，在中文文档中如“啊”、“在”、“的”之类。这些词也可称作虚词，包含副词、冠词、代词等，在文档中使用十分广泛，但却难以对文档分类提供帮助。因此，在研究文本分类等数据挖掘问题时，经常会将它们预先剔除，不仅可以减少存储空间、降低计算成本，而且可以避免它们对分类器的干扰。

经过以上处理后，我们就可以用特征词权重函数来度量文本中每个单词的属性值。

## 文本表示

所谓文本表示既是通过某种形式将文本字符串表示成计算机所能处理的数值向量。那么为什么要进行文本表示，根本原因是计算机不能直接对文本字符串进行处理，因此需要进行数值化或者向量化。不仅传统的机器学习算法需要这个过程，深度学习也需要这个过程，只不过这个过程可能直接包含在了深度学习网络中；同时，良好的文本表示形式也可以极大的提升算法效果。

### One-hot编码

One-Hot编码，又称为一位有效编码，主要是采用N位状态寄存器来对N个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候只有一位有效。One-Hot编码是分类变量作为二进制向量的表示。这首先要求将分类值映射到整数值。然后，每个整数值被表示为二进制向量，除了整数的索引之外，它都是零值，它被标记为1。

One hot 编码进行数据的分类更准确，许多机器学习算法无法直接用于数据分类。数据的类别必须转换成数字，对于分类的输入和输出变量都是一样的。

我们可以直接使用整数编码，需要时重新调整。这可能适用于在类别之间存在自然关系的问题，例如温度“冷”（0）和”热“（1）的标签。当没有关系时，可能会出现问题，一个例子可能是标签的“狗”和“猫”。在这些情况下，我们想让网络更具表现力，为每个可能的标签值提供概率式数字。这有助于进行问题网络建模。当输出变量使用one-hot编码时，它可以提供比单个标签更准确的一组预测。

独热编码的好处：

1.解决了分类器不好处理属性数据的问题，让特征之间的距离计算更加合理。

2.在一定程度上也起到了扩充特征的作用，比如性别本身是一个特征，经过one hot编码以后，就变成了男或女两个特征。

3.将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点。

### 词袋模型

最初的Bag of words，也叫做“词袋”，在信息检索中，Bag of words model假定对于一个文本，忽略其词序和语法，句法，将其仅仅看做是一个词集合，或者说是词的一个组合，文本中每个词的出现都是独立的，不依赖于其他词是否出现，或者说当这篇文章的作者在任意一个位置选择一个词汇都不受前面句子的影响而独立选择的。

假设有如下两句话：

Jane wants to go to Shenzhen.

Bob  wants to go to Shanghai.

将所有词语装进一个袋子里，不考虑其词法和语序的问题，即每个词语都是独立的。例如上面2个例句，就可以构成一个词袋，袋子里包括Jane、wants、to、go、Shenzhen、Bob、Shanghai。假设建立一个数组（或词典）用于映射匹配，得到[Jane, wants, to, go, Shenzhen, Bob, Shanghai]，那么上面两个例句就可以用以下两个向量表示，对应的下标与映射数组的下标相匹配，其值为该词语出现的次数。

[1,1,2,1,1,0,0]

[0,1,2,1,0,1,1]

词袋（Bag-of-words）模型非常易于理解和实施，并为定制特定的文本数据提供了很大的灵活性。它在语言建模和文档分类等预测问题上取得了很大的成功。

然而，它有一些缺点，比如：

1.词汇：词汇需要仔细的设计，特别是为了管理文档的大小，这会影响文档表示的稀疏性。

2.稀疏性：由于计算的原因（空间和时间复杂性）以及信息的原因，稀疏表示更难模拟，因为模型在如此庞大的代表空间中利用这么少的信息面临着巨大挑战。

3.含义：丢弃词序忽略了上下文，进而又影响在文档中的词语的意义（语义）。上下文和意义可以提供很多模型，如果模型可以区分相同的单词不同的排列（“这是有趣的”vs“这是有趣的”），同义词（“旧自行车”vs“二手自行车”）。

### 基于矩阵的表示方法

分布表示算法中最常用的是基于矩阵模型，该模型的思路主要是，根据文本内容构建一个词-上下文矩阵，每一行代表一个词，每一列代表一个文本或者上下文，那么每行就可以作为一个term的表示。词-上下文矩阵构造：给定一篇文章或者一个语料库，首先将其转换为term-document或者是term-context矩阵。矩阵元素值表示：矩阵中每个元素的value可以是该term的TF-IDF值，通常此种方法简洁高效，工程中应用也最为广泛。降维：对于文本数据来说，构造出来的矩阵是高维、稀疏矩阵，因此为便于后续的处理通常会采用降维方法对矩阵进行降维，保留更有意义的内容，常用的方式为SVD（Singular Value Decomposition）。基于矩阵模型的算法有LSA/LSI等和基于聚类基于神经网络的方法。LSI可以说是LSA在search场景下的一个应用，给定一个document collection和一个query，返回query对应的查询。首先将document collection 转换为term-document matrix，term通常来源于document的title，keywords list，abstract，同时也将query转换为一个vector，vector中元素的值为每个词的平均权重，可以用TF-IDF模型计算得出，从而可以根据向量计算出query对应的document。为什么LSA可以有效解词语同义和多义的问题：LSA/I 本质是挖掘词与词在文档层面共现模式，如果两个词经常同时出现，那么他们很容易被理解为具有相同的语义，同时如果两个词的背景上下文经常相同或相似，他们也被理解为具有相同语义。LSA通过捕获这些共现模式，使得在同一个主题中，具有高权重的词聚合在一起，也说明这些词语义相近。优点是能够避免简单语义鸿沟，解决此的歧义问题。缺点是对于大规模语料处理时会很耗费时间，因为文本通常维数较高。

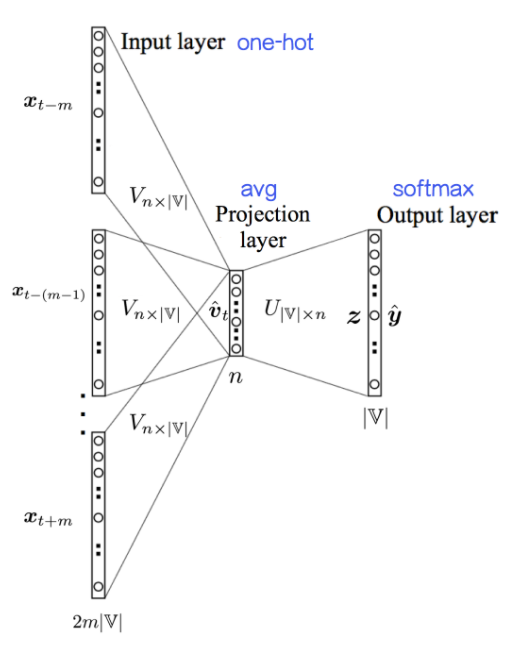
### 词向量模型

词向量模型是考虑词语位置关系的一种模型。通过大量语料的训练，将每一个词语映射到高维度（几千、几万维以上）的向量当中，通过求余弦的方式，可以判断两个词语之间的关系，例如例句中的Jane和Bob在词向量模型中，他们的余弦值可能就接近1，因为这两个都是人名，Shenzhen和Bob的余弦值可能就接近0，因为一个是人名一个是地名。

现在常用word2vec构成词向量模型，它的底层采用基于CBOW和Skip-Gram算法的神经网络模型。

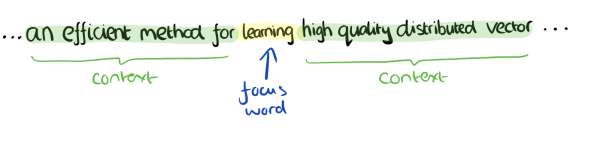
1. **CBOW**

连续词袋模型(CBOW)的结构图如下所示。



在上图中，这里输入层是由one-hot编码的输入上下文{x1,…,xC}组成，其中窗口大小为C，词汇表大小为V。隐藏层是N维的向量。最后输出层是也被one-hot编码的输出单词y。被one-hot编码的输入向量通过一个V×N维的权重矩阵W连接到隐藏层；隐藏层通过一个N×V的权重矩阵W′连接到输出层。接下来，我们假设我们知道输入与输出权重矩阵的大小。

CBOW模型的训练输入是某一个特征词的上下文相关的词对应的词向量，而输出就是这特定的一个词的词向量。比如下面这段话，我们的上下文大小取值为4，特定的这个词是"Learning"，也就是我们需要的输出词向量,上下文对应的词有8个，前后各4个，这8个词是我们模型的输入。由于CBOW使用的是词袋模型，因此这8个词都是平等的，也就是不考虑他们和我们关注的词之间的距离大小，只要在我们上下文之内即可。



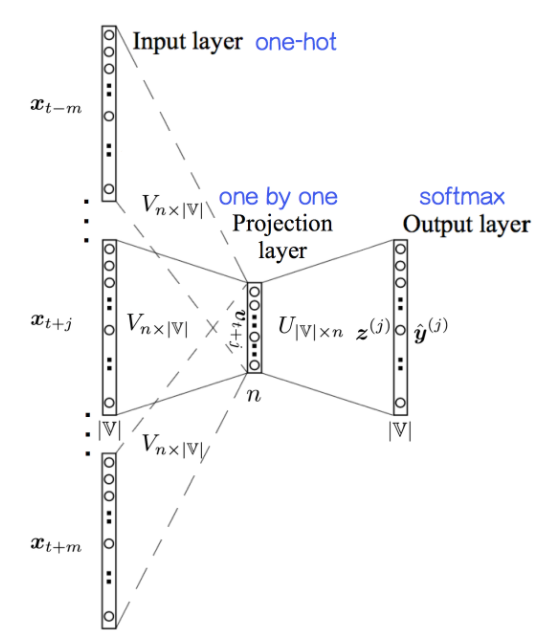
这样我们这个CBOW的例子里，我们的输入是8个词向量，输出是所有词的softmax概率（训练的目标是期望训练样本特定词对应的softmax概率最大），对应的CBOW神经网络模型输入层有8个神经元，输出层有词汇表大小个神经元。隐藏层的神经元个数我们可以自己指定。通过DNN的反向传播算法，我们可以求出DNN模型的参数，同时得到所有的词对应的词向量。这样当我们有新的需求，要求出某8个词对应的最可能的输出中心词时，我们可以通过一次DNN前向传播算法并通过softmax激活函数找到概率最大的词对应的神经元即可。

[1] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.（这篇文章就讲了两个模型：CBOW 和 Skip-gram）

[2] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. 2013, 26:3111-3119.（这篇文章针对Skip-gram模型计算复杂度高的问题提出了一些该进）

**二． Skip-gram模型**

Skip-gram模型反转了目标和上下文单词的使用，结构图如下图所示。



在这种情况下，目标字在输入处被馈送，隐藏层保持相同，并且神经网络的输出层被多次复制以适应所选数量的上下文字。以“cat”和“tree”为例，作为上下文单词，“爬”作为目标词，skim-gram模型中的输入向量为[0 0 0 1 0 0 0 0]，而两个输出层将具有[0 1 0 0 0 0 0 0]和[0 0 0 0 0 0 0 1]分别作为目标向量。代替产生一个概率向量，将为当前示例产生两个这样的向量。以上面讨论的方式产生每个输出层的误差向量。然而，将来自所有输出层的误差向量相加以通过反向传播来调整权重。这确保了每个输出层的权重矩阵WO在整个训练中保持相同。

## 特征选择

从文档中提取的特征数量很大，特别是训练文档库比较大的时候，有些特征对文档分类的作用可能并不大。因此，我们有必要对提取到的文档特征进行筛选，选出那些最能代表文档类别概念的特征。这一过程就是特征选择。

文档属性选择方法基本上是基于信息论和统计分析。特征选择的具体步骤为:

(1)从训练文档库中提取得所有特征项，构成文档特征集合F;

(2)对集合F中的每一项用下列某一种方法进行打分。譬如，选用信息增量方法，则对F中的任意N-gram项f，求IG(f)。当F中的所有项都打分完成后，按分值由高到低进行排序;

(3)假设需要选取N个文档分类属性，则从F中的选取分值高的N个项，构成最终的分类属性集FS，FS最终用于文档分类的训练与测试。

**一. 信息增量(Information Gain)**

信息增量表示文档中包含某一特征值时文档类的平均信息量。它定义为某一特征在文档中出现前后的信息嫡之差。假定c为文档类变量，C为文档类的集合，d为文档，f为特征(以下各节同此)。对于特征f，其信息增量记为IG(f)，则有:

**二. 互信息(Mutual Information)**

在统计学中，互信息用于表征两个变量间的相关性。对于特征f ,其互信息记为MI(f)，则有:

显然，当f独立于c时，MI(c,f)为0。在应用时一般取平均值：

**三．交叉熵(Cross Entropy)**

交叉熵(Cross Entropy)和信息增量相似，不同之处在于信息增量中同时考虑到了特征在文本中发生与不发生时的两种情况，而交叉熵只考虑特征在文本中发生一种情况。假定*c*为类变量，*C*为类的集合，对于特征*f*，其交叉熵一记为CE(*f*)，则有:

若只考虑单个类，则有：

# 分类算法

## KNN分类方法

基于临近点的分类(K Nearest Neighbor,K-NN)是模式识别领域非常著名的分类模型, 被应用到文本分类中的时间较早，该模型区分类别的效果显著，是无需进行参数估计的一个重要的模型。

1. KNN的基本思路

KNN分类是一种根据实际训练集进行分类的模型。该方法的基本思路是:

a.将训练文本集中的文本采用VSM模型转换为向量。

b.存储文本向量集合；

c.算出需要划分类别的文档与原始训练集合中每一个文档的近似的程度大小，并降序排列：

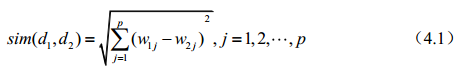
d.事先定义近邻个数K,取出排列后排名位于前K的文档。

e.需要区分类别的文档的最红划分便是排在前面K位的文本的众数类。

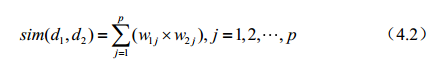
(2) 文本相似度计算

文档集合之间的类似程度的表示形式一般有欧氏距离、 矢量内积和矢量之间夹角余弦这三种：

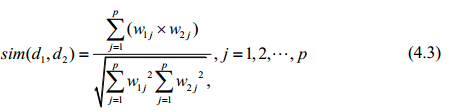
a. 欧氏距离，计算得到的 的结果越小，则说明文档 与文档 越近似。



b.向量内积，该方法与欧氏距离相反， 越大， 则说明文档 d1与文档 d2越近似。



c.夹角余弦， 计算得到的的结果越大，则表明文档 d1与文档 d2更加近似。

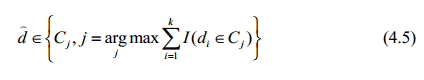


(3) 文本类别确定

记待判别文档为 d ,与其最近邻的文本集合为 {}，文本 是否属于类用示性函数表示为：



使用 K 近邻分类方法做最终类别的判定的基本思路是以最近邻 K 个文本中的众数类作为需要划分类别的文本的类，即多数表决法:



该方法认为每个近邻对于分类的影响都一样，因此对于 K 值的选取非常敏感。

(4) K近邻数选取

由 k值的选取是事先设定的，具有一定的主观性。实际应用中，数据集经常包含噪声，并且不同类别所占比例各不相同，数据集类别之间也没有明确的界限，所以当K较小时， K近邻分类方法对噪声特别敏感，相反，K值较大时，类别中占多数的那一类将会主导分类结果，同样使分类性能低下。

理论上可以采取穷举法，选择所有可能k值，然后选择最优的k值作为近邻选取个数，但是实际中，穷举法带来的计算开销通常是难以接受的，所以本文对全局搜索进行部分简化，方法如下：设 ，设数据集中最少类别包含的实例数目为count1，次少类别实例数目为count2，分两种情况判定 K 值。

1.当 count1相对来讲并不是很小时，取

从而使包含在最小类别的测试文本有足够多的近邻来保证其被正确的划分类别；

2.当count1很小时，取

从而避免属于最小类别的实例因近邻数太少而被误分。

（5）K近邻分类器的特征

K近邻分类方法的优点是原理直观，并且类别确定主要根据附近较为相似的数量一定的若干文本，总之对类与类之间存在叠加或者穿插的需要确定类别的文本集合而言，K近邻方法相较于别的办法合适一些;常见的分类方法均需要训练出稳定的模型，而K近邻方法可以直接将待测文本进行类别识别，所以具有较低的时间复杂度，并且支持增量学习。

K近邻分类算法在划分类别的时候有一个比较大的缺点就是当用于构建模型的文本在不同类别中文本数差距比较明显时，需要进行类别划分的文本极其容易就会被错误的划分到文本数目较多的类别；同时，该方法属于惰性不建模的学习方法，对于每个待分类文本求解与训练文本的相似度，计算较慢，不同于其他能够实现构建模型并实现较快分类的方法。

## Rocchio分类方法

### 基本原理

它是一种基于向量空间模型进行文本分类的的思路。这种方法提供了一种将文本的相关反馈信息融到向量空间模型的方法。相关反馈是一种信息检索中常用的技术，它帮助用户逐步完善基于之前搜索结果的查询。它是由用户根据所需信息检索到的相关文档，给出的进一步用户反馈组成的。用户的反馈信息也可以不断的加入到这种分类器中，被用来逐步改善用户特征，从而进行分类器的训练。而一些线性分类器由类别的显式描述（或原始文档）组成，Rocchio算法则是一个可用于推导出这种线性的且带有描述风格的分类器。

### 基本步骤

Rocchio算法的基本步骤为：

1. 先把属于每一个类别的样本文档转换成文档向量。
2. 求属于每一个类别的样本文档的质心向量（原型向量）。
3. 当给定一个待分类文本时，确定该待分类文本向量，计算待分类文本与各类别的原型向量之间的距离，新文档的类别为距离最近的原型向量对应的类别。

具体来说便是，首先，通过将所有训练文本向量简单的算术平均计算每类文本集合的中心向量。原型向量的计算方法有几种，其中最简单的就是对于一个类的所有文本的特征向量求平均值的结果作为该离别文档的原型向量。然后，当新文本到达后，对该文档进行粉刺处理，将文本表示为特征向量，最后计算新文本特征向量和每类中心向量之间的相似度，常见的计算待分类文本与各个类别的原型向量之间的距离有几种方式，其距离可以是向量点积、向量之间夹角的余弦值或其他函数。

### 核心思想

对于Rocchio的基本思想其实不难解释，对于一个词集，和一个分类，总有某些词，这些词一旦出现属于这个分类的可能性就会增加，而另一些词一旦出现属于这个分类的可能性就会降低，那么累计这些正面的，和负面的影响因素，最后由文档分离出的词向量可以得到对于每个类的一个打分，打分越高属于该类的可能性就越大。

## 朴素贝叶斯

(1)朴素贝叶斯

1960年 Maron和 Kuhns首先提出了朴素贝叶斯分类方法，是一种基于概率模型的分类方法。贝叶斯分类就是通过观测集数据特征对原始类别先验分布进行修正得到后验分布的一种分类模型。这一理论的根本就是假定任何事情都是不一定的，都具有不一样的分布。

Lewis阐释了NB在信息检索和文本分类领域的应用。朴素贝叶斯法（Naïve Bayes）是基于贝叶斯定理与特征条件独立假设的分类方法，属于统计学分类方法。简单来说，朴素贝叶斯分类器假设在给定样本类别的条件下，样本的每个特征与其他特征均不相关，对于给定的输入，利用贝叶斯定理，求出后验概率最大的输出。朴素贝叶斯法实现简单，学习与预测的效率均较高，在文本分类领域有广泛的应用。

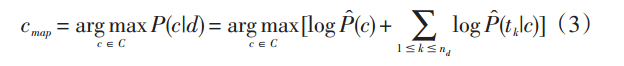
建立朴素贝叶斯分类器有两种方法，第一种是多项式模型（multional Naïve Bayes）,另外一种是多元伯努利模型(multicatiate Bernoulli model),也称为二值独立模型。

多项式NB是一种生成式模型，文档 d属于类别c的概率可以通过如下公式获得:

其中，P(c)是文档 d属于类别 c的先验概率，<t1,t2,t3,...,tnd>是文档d中的词项，nd是d中所有词项的数量。在已知文档d属于每个类别ci的先验概率后，需要找到文档d最有可能的类别，对于朴素贝叶斯而言，也就是具有最大后验概率（maximum a posteriori, MAP）估计值的类别，即公式。



计算条件概率的乘积，可能会导致浮点数下界溢出，所以引入对数，见下公式。



多元伯努利模型中，对于词项表中的每个词项都对应一个二值变量，分别表示该词项在文档中存在与否，存在用 1表示，不存在用 0表示。 P(d|c)= P(<e1,...,ei,...,eM>|c),<e1,...,ei,...,eM>是一个M维的布尔向量，表示每个词项在文档d中是否存在,其Cmap可以通过下公式计算得到。



多项式模型和多元伯努利模型的主要区别在于：

1.文档表示是否考虑词项在文档中的出现的频率。多项式NB不仅考虑词项在文档中的出现频率，且徐考虑词项出现的频率；而二者则仅koala词项是否出现。

2.文档中未出现的词项对于分类的作用不同。在多项式NB中，未出现的词项不参与分类，而在多元伯努利模型中，对未出现的词项也要建模，即未出现词项也要作为一个因子参加P(c|d)的计算以决定文档类别。

相较于多元伯努利模型，多项式 NB分类方法具有如下优点：

①当有多个同等重要的特征联合起来对分类决策起作用时，多项式 NB能够表现出更好的分类效果；

②多项式 NB对噪音特征和概念漂移具有更强的鲁棒性；

③速度快。不论训练还是分类过程其时间复杂度都是线性的。

抛开两种方法的具体区别，Rish等发现尽管朴素贝叶斯分类器的独立性假设欠合理，概率估计也不那么精确，但是其分类决策依然在研究与应用中取得了很好的效果。通过在分类错误数据集上的分布熵影响试验证实：在低熵情形下，朴素贝叶斯分类器具有很好的分类性能。特别是在以下两种情形表现更优，一是具有完全独立的特征，二是具有函数相关的特征。该研究还证实，朴素贝叶斯分类器的准确性与特征向量独立性的程度间没有直接关联；在朴素贝叶斯分类器中损失部分特征所包含的类别信息却可以获得更优的分类准确率。

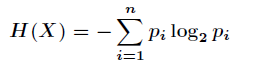
## 决策树算法

### 基本思想与定义

决策树的构造过程不依赖领域知识，它使用属性选择度量来选择将元组最好地划分成不同的类的属性。所谓决策树的构造就是进行属性选择度量确定各个特征属性之间的拓扑结构。构造决策树的关键步骤是分裂属性。所谓分裂属性就是在某个节点处按照某一特征属性的不同划分构造不同的分支，其目标是让各个分裂子集尽可能地“纯”。尽可能“纯”就是尽量让一个分裂子集中待分类项属于同一类别。分裂属性分为三种不同的情况：1、属性是离散值且不要求生成二叉决策树。此时用属性的每一个划分作为一个分支。2、属性是离散值且要求生成二叉决策树。此时使用属性划分的一个子集进行测试，按照“属于此子集”和“不属于此子集”分成两个分支。3、属性是连续值。此时确定一个值作为分裂点split\_point，按照>split\_point和<=split\_point生成两个分支。构造决策树的关键性内容是进行属性选择度量，属性选择度量是一种选择分裂准则，是将给定的类标记的训练集合的数据划分D“最好”地分成个体类的[启发式方法](http://en.wikipedia.org/wiki/Heuristic_(computer_science))，它决定了拓扑结构及分裂点split\_point的选择。属性选择度量算法有很多，一般使用自顶向下递归分治法，并采用不回溯的[贪心策略](http://en.wikipedia.org/wiki/Greedy_algorithm)。这里介绍[ID3](http://en.wikipedia.org/wiki/ID3_algorithm)和[C4.5](http://en.wikipedia.org/wiki/C4.5_algorithm)两种常用算法。

### 核心算法类别

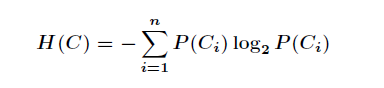
ID3算法介绍。ID3算法是决策树的一种，它是基于奥卡姆剃刀原理的，即用尽量用较少的东西做更多的事。ID3算法，即Iterative Dichotomiser 3，迭代二叉树3代，是Ross Quinlan发明的一种决策树算法，这个算法的基础就是上面提到的奥卡姆剃刀原理，越是小型的决策树越优于大的决策树，尽管如此，也不总是生成最小的树型结构，而是一个启发式算法。在信息论中，期望信息越小，那么信息增益就越大，从而纯度就越高。ID3算法的核心思想就是以信息增益来度量属性的选择，选择分裂后信息增益最大的属性进行分裂。该算法采用自顶向下的贪婪搜索遍历可能的决策空间。在信息增益中，重要性的衡量标准就是看特征能够为分类系统带来多少信息，带来的信息越多，该特征越重要。在认识信息增益之前，先来看看信息熵的定义熵这个概念最早起源于物理学，在物理学中是用来度量一个热力学系统的无序程度，而在信息学里面，熵是对不确定性的度量。在1948年，香农引入了信息熵，将其定义为离散随机事件出现的概率，一个系统越是有序，信息熵就越低，反之一个系统越是混乱，它的信息熵就越高。所以信息熵可以被认为是系统有序化程度的一个度量。假如一个随机变量http://images.cnitblog.com/blog/571227/201412/112108114467870.png的取值为http://images.cnitblog.com/blog/571227/201412/112108491346383.png，每一种取到的概率分别是http://images.cnitblog.com/blog/571227/201412/112110282909943.png，那么http://images.cnitblog.com/blog/571227/201412/112110572751166.png的熵定义为



意思是一个变量的变化情况可能越多，那么它携带的信息量就越大。对于分类系统来说，类别http://images.cnitblog.com/blog/571227/201412/112119378211796.png是变量，它的取值是http://images.cnitblog.com/blog/571227/201412/112120381502926.png，而每一个类别出现的概率分别是：



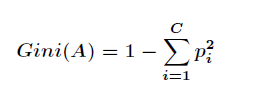
而这里的http://images.cnitblog.com/blog/571227/201412/112122176655828.png就是类别的总数，此时分类系统的熵就可以表示为：



以上就是信息熵的定义，接下来介绍信息增益。信息增益是针对一个一个特征而言的，就是看一个特征http://images.cnitblog.com/blog/571227/201412/121352330406260.png，系统有它和没有它时的信息量各是多少，两者的差值就是这个特征给系统带来的信息量，即信息增益。

C4.5只是在ID3的基础上做了改进，使用信息增益率，而不是信息增益。

CART算法是一种二分递归分割技术，把当前样本划分为两个子样本，使得生成的每个非叶子结点都有两个分支，因此CART算法生成的决策树是结构简洁的二叉树。由于CART算法构成的是一个二叉树，它在每一步的决策时只能是“是”或者“否”，即使一个feature有多个取值，也是把数据分为两部分。在CART算法中主要分为两个步骤（1）将样本递归划分进行建树过程（2）用验证数据进行剪枝。第一个过程进行递归建立二叉树，那么它是如何进行划分的 ？设http://images.cnitblog.com/blog/571227/201501/091521164535431.png代表单个样本的http://images.cnitblog.com/blog/571227/201501/091522035317783.png个属性，http://images.cnitblog.com/blog/571227/201501/091522421257425.png表示所属类别。CART算法通过递归的方式将http://images.cnitblog.com/blog/571227/201501/091523317818089.png维的空间划分为不重叠的矩形。划分步骤大致如下：（1）选一个自变量http://images.cnitblog.com/blog/571227/201501/091528272651984.png，再选取http://images.cnitblog.com/blog/571227/201501/091531419535737.png的一个值http://images.cnitblog.com/blog/571227/201501/091532214533937.png，http://images.cnitblog.com/blog/571227/201501/091532442966491.png把http://images.cnitblog.com/blog/571227/201501/091533267036202.png维空间划分为两部分，一部分的所有点都满足http://images.cnitblog.com/blog/571227/201501/091534271563502.png，另一部分的所有点都满足http://images.cnitblog.com/blog/571227/201501/091535186096649.png，对非连续变量来说属性值的取值只有两个，即等于该值或不等于该值。（2）递归处理，将上面得到的两部分按步骤（1）重新选取一个属性继续划分，直到把整个http://images.cnitblog.com/blog/571227/201501/091536471093412.png维空间都划分完。在划分时候有一个问题，它是按照什么标准来划分的 ？ 对于一个变量属性来说，它的划分点是一对连续变量属性值的中点。假设http://images.cnitblog.com/blog/571227/201501/091549258753804.png个样本的集合一个属性有http://images.cnitblog.com/blog/571227/201501/091550106258598.png个连续的值，那么则会有http://images.cnitblog.com/blog/571227/201501/091551330314125.png个分裂点，每个分裂点为相邻两个连续值的均值。每个属性的划分按照能减少的杂质的量来进行排序，而杂质的减少量定义为划分前的杂质减去划分后的每个节点的杂质量划分所占比率之和。而杂质度量方法常用Gini指标，假设一个样本共有http://images.cnitblog.com/blog/571227/201501/091603514532531.png类，那么一个节点http://images.cnitblog.com/blog/571227/201501/091604407344010.png的Gini不纯度可定义为



其中http://images.cnitblog.com/blog/571227/201501/091606025624178.png表示属于http://images.cnitblog.com/blog/571227/201501/091606342504996.png类的概率，当Gini(A)=0时，所有样本属于同类，所有类在节点中以等概率出现时，Gini(A)最大化，此时http://images.cnitblog.com/blog/571227/201501/091608573281118.png。

### 特点与剪枝工作

决策树是通过对训练数据集重复分组来构造的． 如果训练数据集中的数据能准确地反映分析对象的本质，则通过该训练数据集所得到的决策树将可以准 确地对该问题进行分类．然而，由于实际问题中存在许多不确定的因素，当用决策树构造算法对这类数据 分类时，所得到的决策树将会变得大而复杂，由此得到的知识规则集也会变得大而复杂．然而，研究证明， 大而复杂的决策树并不意味着可以得到更准确的规则集[1]．因此，对决策树进行剪枝非常必要．当前存在 许多种不同的剪枝方法，分为事前剪枝和事后剪枝两大类[2]，后者应用得较广泛．事后剪枝算法又可以分 为两类，一类是把训练数据集分成树生长集和树剪枝集；另一类算法则在树生长和树剪枝阶段都使用同一 训练数据集．事前剪枝的缺点是使树的生长可能过早停止，因此应用较少。

## SVM支持向量机算法

### 基本原理

支持向量机(support vector machine)是一种分类算法，通过寻求结构化风险最小来提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。通俗来讲，它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

### 具体方法

#### (1)超平面

在n维空间中找到一个分类超平面，将空间上的点分类。如下图是线性分类的例子。



它分类的思想是，给定给一个包含正例和反例的样本集合，svm的目的是寻找一个超平面来对样本根据正例和反例进行分割。

如果不关注空间的维数，这种线性函数就是前言中所说的那个统一的名称——超平面（Hyper Plane）！

在样本空间中，划分超平面可通过如下线性方程来描述：

http://img.blog.csdn.net/20160418171323695?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

假设它已经完成了对样本的分隔，且两种样本的标签分别是{+1，-1}，那么对于一个分类器来说，g(x)>0和个g(x)<0就可以分别代表两个不同的类别，+1和-1。

(2)最大间隔超平面

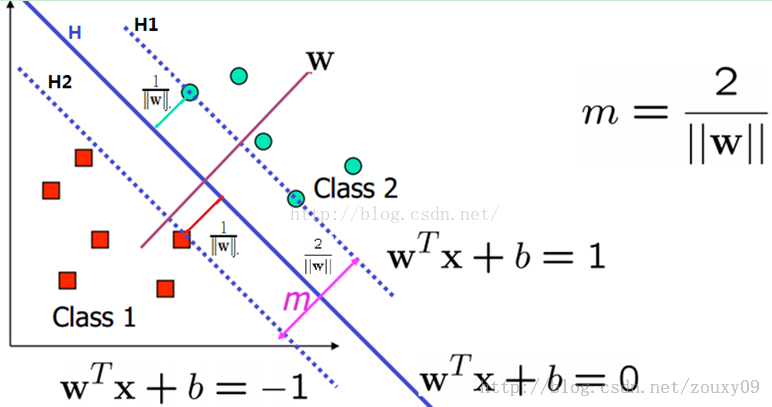
但光是分开是不够的，SVM的核心思想是尽最大努力使分开的两个类别有最大间隔，这样才使得分隔具有更高的可信度。而且对于未知的新样本才有很好的分类预测能力（在机器学习中叫泛化能力）

那么怎么描述这个间隔，并且让它最大呢？SVM的办法是：让离分隔面最近的数据点具有最大的距离。

为了描述离分隔超平面最近的数据点，需要找到两个和这个超平面平行和距离相等的超平面：

H1: y = wTx + b=+1 和 H2: y = wTx + b=-1

如图所示：



在这两个超平面上的样本点也就是理论上离分隔超平面最近的点，是它们的存在决定了H1和H2的位置，**支撑**起了分界线，它们就是所谓的**支持向量，**这就是支持向量机的由来

有了这两个超平面就可以顺理成章的定义上面提到的间隔（margin）了

二维情况下 ax+by=c1和ax+by=c两条平行线的距离公式为：

http://img.blog.csdn.net/20160418172327552?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

可以推出H1和H2两个超平面的间隔为2/||w||，即现在的目的是要最大化这个间隔。

所以support vector machine又叫Maximum margin hyper plane classifier等价于最小化||w||。为了之后的求导和计算方便，进一步等价于最小化

http://img.blog.csdn.net/20160418172433113?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

假设超平面能将样本正确分类，则可令：

http://img.blog.csdn.net/20160418172550993?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

两个式子综合一下有：

http://img.blog.csdn.net/20160418172643396?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

这就是目标函数的约束条件。现在这个问题就变成了一个最优化问题：

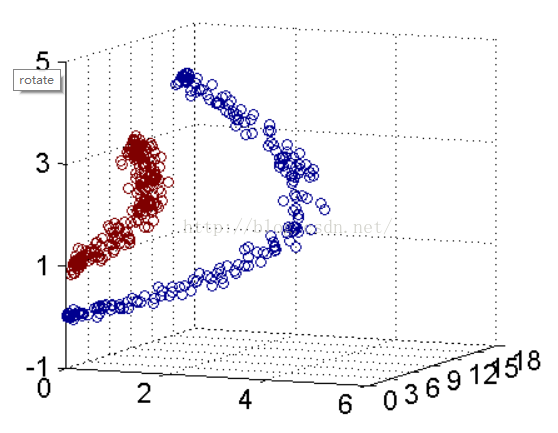
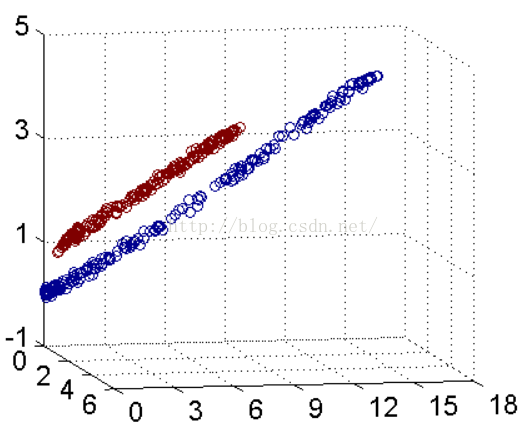


而且这是一个凸二次规划问题，一般的解决方法有两种1是用现成的优化工具包直接求解，2是使用Lagrange Duality找到一种更有效的方法求解。

其中方法2具有两个优点:a、更好解。b、可以自然地引入核函数，推广到非线性分类

#### 核函数

前述方法对线性不可分的样本集无能为力。但是一个低维的样本集映射到高维则可以变成线性可分（如图所示），那样才能使用SVM工作。

设映射函数为Φ(•)，则映射后的空间分类函数变成

http://img.blog.csdn.net/20160418173433352?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

但是，如果拿到低维数据直接映射到高维的话，维度的数目会呈现爆炸性增长。所以这里需要引入核函数（kernal function）。

核函数的思想是寻找一个函数，这个函数使得在低维空间中进行计算的结果和映射到高维空间中计算内积<Φ(x­1), Φ(x2)>的结果相同。这样就避开直接在高维空间中进行计算，而最后的结果却是等价的。现在，分类函数就变成了这样：

http://img.blog.csdn.net/20160418173515336?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQv/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

其中k就是核函数

由于对任意数据集找到它合适的映射是困难的且没有必要，所以通常会从常用核函数中选择。常用核函数例如：多项式核函数、高斯核核函数、线性核函数、字符串核函数

上述方法叫做核方法。事实上，任何将计算表示为数据点内积的方法都可以用核方法进行非线性扩展。

# 文本聚类算法

## 基于密度的聚类算法

通常情况下，绝大多数基于分层或划分的聚类算法都是通过一般的距离方法来计算两个对象之间的距离，并由此形成类簇，这类方法优点就是方法简单易于运用，但是它们只能发现有规则的球形聚类，而无法发掘其他形状的聚类，这也是这类方法的一个局限性。另一方面，基于划分的聚类算法在面对数据集中的噪声数据处理效果较差。

目前，很多新提出的聚类方法都结合了密度聚类方法的思想以吸纳其部分优点。基于密度的聚类算法将簇看成是数据空间中被低密度区域分割开的高密度区域。密度定义为单位体积内的点数，簇内部密度比簇外大。基于密度的聚类算法认为,类别即是向任意方向按相同密度扩张的连通区域。这种算法主要需要考虑数据空间的密度,连通性与边界区。对于非凸的,不规则的形状,基于分层或划分的算法往往难以处理,然而基于密度的算法却能很好地处理此类问题。当数据密度超过了事先给定的阈值，则可以判定这些数据是在一个类簇之中，当然一般情况下都会给类簇设定一个最少包含对象数据的阈值，只有当类簇满足类簇密度以及最少包含对象数这两个阈值时，其才能被确定为一个类簇。正是因为如上特点，基于密度的聚类方法较其他密度聚类算法具有对噪声数据不敏感和可以发现任意形状的类簇的优势。在密度聚类方法中较为著名要属DBSCAN算法以及OPTICS算法。由于密度是一个局部概念，此类算法又称为局部聚类(Local Clustering)；基于密度的聚类通常只扫描一遍数据库，所以也称为单遍扫描聚类(Single Scan Clustering)。

但是密度聚类算法也有不少缺点，首先，当数据分布比较稀疏离散时，其聚类效果会比较差。其次，当数据量比较大时内存等相关硬件消耗过大，最后则是聚类最少包含对象数(Minpts)以及扫描半径(Eps)，这两个输入参数选择是否恰当关系到聚类的最终质量。

### DBSCAN算法

随着“信息爆炸”、“大数据”等概念的兴起，数据挖掘领域对于聚类分析算法处理不同类型数据集能力的要求越来越高，而一些传统的聚类算法如 K-Means、BIRCH 等只能聚类球形的凸形状数据集，适用范围不够广泛。基于密度的聚类算法不需要预先指定聚类簇数，而且可以在含有噪声数据的数据集中识别任意数量和形状的聚类，DBSCAN作为基于密度算法的经典代表，在聚类分析中得到越来越多的应用。DBSCAN算法是由伊斯特等人在20世纪末提出的，是一种经典的密度聚类算法。该算法是通过搜寻数据集中某一特定对象在确定扫描半径内的相邻对象，当其周围对象数量满足类簇形成的最少包含对象数，则这些对象将形成一个类簇，DBSCAN方法的具体相关基本概念定义如下：

(1)对象扫描半径Eps(R)：算法中给定数据集特定对象搜寻其相邻对象的扫描邻域半径；也常常用ε表示。

(2)Minpts(M)：由某一特定对象及其相邻对象构成数据类簇所满足的最少包含对象数；

(3)核心对象P：对象P属于数据集D(PD)，并且在P周围的扫描半径R内存在不少于M数量的相邻对象，即：

其中，表示在对象P的以半径为R的邻域内的相邻对象数，这些对象满足以下条件：

其中，Q是数据集D的对象，表示通过某一特定距离函数计算得到的Q和P之间的距离。

(4)边界对象B：边界对象是数据集中不满足核心对象的条件的一些数据对象。边界对象通常是指该对象在其扫描半径R邻域内所包含的对象不满足算法给定的聚类最少包含对象数M，具体定义如下：

其中，对象B的扫描半径内的对象数少于算法最小包含对象数M这一阈值。通常情况下，边界对象以及核心对象之间都是通过密度可达对象和密度相连对象连接。

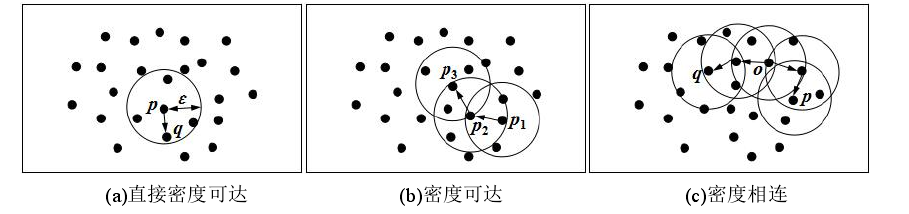
(5)直接密度可达：在数据集D中，如果一个对象Q和另一个对象P的直接密度可达，则它们之间满足两个条件：

①P是一个核心对象，即满足条件：

②，Q属于P的R-邻域对象。

(6)密度可达：对于对象链：是从关于Eps和MinPts直接密度可达的，则对象是从对象关于Eps和MinPts密度可达的。

(7)密度相连：如果数据集D中有3个对象P、A和B，对象P和对象A是密度相连，则满足该对象扫描半径R以及聚类最少包含对象阈值M的条件下，对象P以及对象A和对象B之间的关系是密度可达。



(8)聚类：对于给定数据集D，聚类C是在条件R以及M下形成，且是D的一个数据子集，那么它满足以下两个条件：

①,如果P属于聚类C(P是核心对象)以及Q和P在条件R和M下密度可达，那么。

②，那么对象P和对象Q之间是密度相连。

(9)噪声对象：通过DBSCAN算法形成的聚类是使用核心对象以及所有和核心对象拥有密度可达或密度相连这一关系对象形成的，假使某一对象属于数据集D，但又不属于各个密度聚类，那么其就是噪声对象。

DBSCAN算法执行步骤具体描述如下：

·输入：数据集D，半径参数Eps，密度阈值Minpts

·输出：聚类结果及噪声数据

步骤1：从数据集D中随机抽取一个未被处理的对象P

步骤 2：如果抽出的点是核心点，则找出所有从该点密度可达的对象，形成一个簇。

步骤 3：如果抽出的点是边缘对象，则跳出本次循环，寻找下一个点。

步骤 4：重复执行步骤1、2、3，直到数据集中所有对象都为“已处理”。

通过以上描述，可以发现基于密度的聚类就是一组“密度相连”的对象，以实现最大化的“密度可达”。不包含在任何聚类中的对象就是噪声数据。

DBSCAN算法需要计算数据集中的任意两个对象之间的距离，因此该算法的复杂度是O(n2)，其中n是数据集中所包含的对象数。如果D中的所有对象都预先建立了空间索引那么其时间对于一个对象P扫描其邻接对象的时间复杂度为O(n)，所以对于整个数据集的耗时为O(nlogn)。

存在的问题：

（1）在聚类过程中，DBSCAN一旦找到一个核心对象，即以该核心对象为中心向外扩展．此过程中核心对象将不断增多，未处理的对象被保留在内存中。若数据库中存在庞大的聚类，将需要很大的内存来存储核心对象信息，其需求难以预料。

（2）输入参数敏感．确定参数Eps，MinPts困难，若选取不当，将造成聚类质量下降。

（3）由于在DBSCAN算法中，变量Eps，MinPts是全局惟一的，当数据分布不均匀时聚类质量较差。

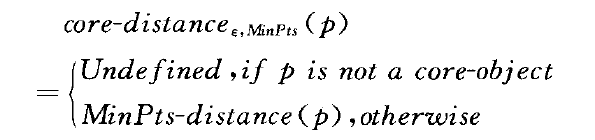
### OPTICS算法

一直以来，研究者对如何提高聚类算法的性能费尽心思，因为性能的好坏通常是衡量聚类算法优劣的重要标准。然而，很多时候更为本质的问题在于语料本身，大多数聚类算法对语料非常敏感，在一个语料上取得优良性能的算法常常在另一个语料上效果不尽如人意。与经典聚类算法重点关注性能不同，Ankerst M等于1999年提出的OPTICS(Ordering Points To Identify the Clustering Structure)算法更为关注如何直观地反映语料自身的潜在结构。OPTICS是一种基于密度的聚类算法，它从一个随机选定的对象出发，朝着数据最为密集的区域扩张，最终将所有对象组织成一个能够反映语料结构的可视化有序序列。

**一．相关概念**

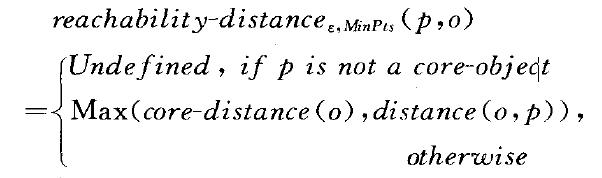
直接密度可达、密度可达、密度相连的概念与上文中DBSCAN算法中的概念相同。可以知道，密度可达是直接密度可达的传递；密度相连则是从同一点密度可达的任意两点的对称关系。由此，如果从某个选定的核心点出发，不断向密度可达的区域扩张，将得到一个包括核心点和边界点的最大化区域，区域中任意两点密度相连，这即为一个聚类簇。DBSCAN算法就是通过上述过程搜索和提取尺度为e的所有簇。为了具备更为精细的刻画能力，OPTICS引入了核心距离和可达距离的概念：

（1）核心距离(core-distance)：假定点p包含MinPts个邻居的最小半径为MinPts-distance(p)，那么P的核心距离定义为：



也就是说，核心距离是一个点成为核心点的最小邻域半径。

（2）可达距离(reachability-distance)：假定p是某点o的ε邻域中的点，那么p与o相关的可达距离定义为：



可见，p与o相关的可达距离即是从o直接密度可达p的最小距离。该距离与空间密度直接相关，如果该点的所在空间密度大，它从相邻点直接密度可达的距离就小，反之亦然。如果我们想要朝着数据尽量稠密的空间进行扩张，那么可达距离最小的点是最佳的选择。为此，OPTICS算法用一个可达距离升序排列的有序种子队列(OrderSeeds)存贮待扩张的点，以迅速定位稠密空间的数据对象。

**二. OPTICS算法**

算法具体过程如下：

步骤1：有序种子队列初始为空，结果队列初始为空；

步骤2：如果所有点处理完毕，算法结束；否则选择一个未处理对象放入有序种子队列：

步骤3：如果有序种子队列为空，返回步骤2，否则选择第一个对象p进行扩张：

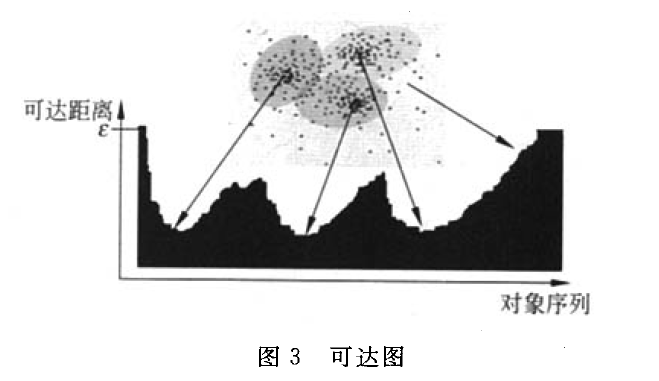
步骤3.1：如果p不是核心点，转步骤4；否则，对p的Eps邻域内任一未扩张的邻居q：

步骤3.1.1：如果q已在有序种子队列中且从P到q的可达距离小于旧值，则更新q的可达距离，并调整q到相应位置以保证队列的有序性；

步骤3.1.2：如果q不在有序种子队列中，则根据p到q的可达距离将其插入有序队列；

步骤4：从有序种子队列中删除p，并将p写入结果队列中，返回步骤3。

算法的时间复杂度为O(n2)。以结果队列中的点序列作为横坐标，可达距离作为纵坐标，将得到如下图所示的可达图(teachability plots)。图中可达距离小的波谷区域代表数据稠密的簇内的点；波峰区域则代表数据稀疏的边界点。因此，可以在遍历可达图的过程中，通过陡峭下降区域(steep down region)和陡峭上升区域(steep up region)来辨别和提取聚类簇。



## 基于层次的聚类算法

一个层次的聚类方法是根据给定的簇间距离度量准则,构造和维护一棵由簇和子簇形成的聚类树,直至满足某个终结条件为止。根据层次分解是自底向上还是自顶向下形成,层次聚类方法可以分为凝聚的（agglomerative）和分裂的(divisive)。一个纯粹的层次聚类方法的聚类质量受限于如下特点一旦一个合并或分裂被执行,就不能修正。

层次聚类方法（Hierarchical Clustering Method）是一种发展比较早、应用广泛的聚类方法，按采用“自顶向下（Top- Down）”和“自底向上（Bottom- Up）”两种方 式 ， 分 别 被 称 为 分 解 型 层 次 聚 类 法（DivisiveHierarchical Clustering） 和 聚 结 型 层 次 聚 类 法(Agglomerative Hierarchical Clustering）。层次聚类方法采用一种迭代控制策略，使聚类逐步优化。它是按照一定的相似性判断标准，合并最相似的部分或者分割最不相似的部分。目前，层次聚类的研究中出现了一些结合其他聚类方法的多阶段聚类分析技术，比较有代表性的有BIRCH（balanced iterative reducing and clustering using hierarchy）、CURE（clustering using representatives）、Chameleon等，它们都是基于层次思想的组合算法，接下来我们对其基本思想进行简要介绍。

### CURE算法

绝大多数聚类算法或者擅长处理球形和相似大小的聚类，或者在存在孤立点时变得比较脆弱。CURE采用了一种新颖的层次聚类算法，该算法选择基于质心和基于代表对象方法之间的中间策略。它不同于单个质心或对象来代表一个类，而是选择数据空间中固定数目的具有代表性的点。一个类的代表点通过如下方式产生：首先选择类中分散的对象，然后根据一个特定的分数或收缩因子“收缩”或移动它们。在算法的每一步，有最近距离的代表点对（每个点来自于一个不同的类）的两个类被合并。每个类有多于一个的代表点使得CURE可以适应非球形的几何形状。类的收缩或凝聚可以有助于控制孤立点的影响。因此，CURE对孤立点的处理更加健壮，而且能够识别非球形和大小变化比较大的类。针对大型数据库，CURE采用随机取样和划分两种方法组合：一个随机样本首先被划分，每个划分被部分聚类。

CURE算法的思想主要体现在如下几个方面：

（1）CURE算法采用的是凝聚层次聚类。在最开始的时候，每一个对象就是一个独立的类，然后从最相似的对象开始进行合并。

（2）为了处理大数据集，采用了随机抽样和分割（Partitioning）手段。采用抽样的方法可以降低数据量，提高算法的效率。在样本大小选择合适的情况下，一般能够得到比较好的聚类结果。另外，CURE算法还引入了分割手段，即将样本分割为几个部分，然后针对各个部分中的对象分别进行局部聚类，形成子类。再针对子类进行聚类，形成新的类。

（3）传统的算法常常采用一个对象来代表一个类，而CURE算法由分散的若干对象，在按收缩因子移向其所在类的中心之后来代表该类。由于CURE算法采用多个对象来代表一个类，并通过收缩因子来调节类的形状，因此能够处理非球形的对象分布。

（4）分两个阶段消除异常值的影响。CURE算法采用的是凝聚层次聚类。在最开始的时候，每一个对象就是一个独立的类，然后从最相似的对象开始进行合并。由于异常值同其它对象的距离更大，所以其所在的类中对象数目的增大就会非常缓慢，甚至不增长。第一个阶段的工作，是将聚类过程中增长非常缓慢的类作为异常值除去。第二个阶段的工作（聚类基本结束的时候）是将数目明显少的类作为异常值除去。

（5）由于CURE算法采用多个对象来代表一个类，因此可以采用更合理的非样本对象分配策略。在完成对样本的聚类之后，各个类中只包含有样本对象，还需要将非样本对象按一定策略分配到相应的类中。

### BIRCH算法

BIRCH(Balanced Iterative Reducingand Clustering Hierarchies)是专门针对大规模数据集提出的聚集型层次聚类算法,它综合了层次凝聚和迭代的重定位方法。首先用自底向上的层次算法,然后用迭代的重定位来改进结果。它的主要思想是扫描数据库,建立一个初始存放于内存中的聚类特征树,然后对聚类特征树的叶结点进行聚类。

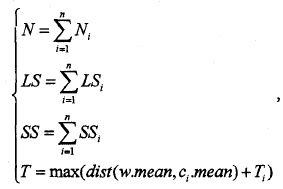
聚类特征的定义(CF)：一个聚类特征(CF)是一个三元组（N,LS,SS）其中N是簇中的点的数目,LS是N个点的线性和,SS是N个点的平方和。

聚类特征树的定义(CF树):一颗CF树是一个带有分支因子B的平衡树,每一个内部结点对于每一个子结点都包含一个CF三元组。每个叶结点也代表一个簇,并且对于其中每个子簇都包含一个CF条目。在叶结点中的子簇要有一个不超过给定阈值T的直径。

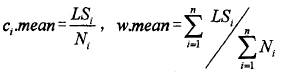
合并定理:假定n(大于1)之个簇c进行合并,n个簇的聚类特征表示为

3c1380285d71ca300ba88544fa08c22,那么合并后簇为W,其聚类特征为

5b9c6f8e352c8dbb9d7bc97cbe508fb



其中



合并后簇的聚类特征精确地表示了两个聚类合并后的渐增性。

在层次聚类方法中,要按照一定的相似性判断标准合并最相似的部分,或者分割最不相似的两个部分,判断各个类之间的相似程度的准则是假设和是聚结过程中同一层次上的两个类,和马分别是和两个类中的对象数目,夕为中的任意一个对象,户为几中的任意一个对象,关为中对象的平均值,石为中对象的平均值。

BIRCH聚类算法步骤：

BIRCH聚类算法利用特征树结构对数据集进行聚类,算法主要过程为首先, 将所有初始数据扫描,建立一个原始化聚集CF树，尽最大可能使得特征树包含所有信息，然后,用聚集特征代替原有数据集进行聚类。

在第一阶段,CF树是随着原始数据的加入而自动形成的：一个对象被放入那个离它最近的叶子结点中去。如果放入以后这个簇的半径大于阈值T的话,那么这个叶结点就会被分割。插入过程类似于B+树构建中的插入和结点分裂。聚集特征树的大小可以通过调节参数来改变,如果要存储树需要的内存大于主存,可以定义一个较小的阈值，然后通过提升阈值重新建立一个聚类CF树,这个重建过程并不需要将整个记录扫描一次，而是建立在原有树的叶子结点的基础之上的。因此，建立一个树数据记录只需要被扫描一次。当树建好以后,可以在第二阶段用其他的聚类算法对聚类特征进行聚类。

## 基于划分的聚类方法

对于给定的包含n个数据对象的数据库，通常基于划分的方法要求用户给定构建数据的最终划分数目k，每个划分表示一个聚簇，并且k<n。也就是说，算法将整个数据集划分为k个组，同时满足：每个聚簇至少包含一个数据对象且每个数据对象必须唯一的属于一个聚簇。对于一个给定要构建的数目k，基于划分的方法是，首先创建一个初始划分，通常采用的方法是随机选取k个数据对象作为初始点，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，采用的准则是：在同一个聚簇中的数据对象尽可能相近或相关，不同的聚簇中的数据对象尽可能相异或无关。根据对象在划分之间移动的衡量参数和聚簇的表示方法不同，基于划分的方法主要有K-Means算法，K-Medoids算法及最近邻算法等等，下面我们将逐一对其原理进行介绍和分析。

### K-Means聚类

（1）基本定义

假设数据集D包含n个欧式空间中的对象，划分方法把D中的对象分配到k个簇当中，使用一个目标函数用来评估划分的质量，使得簇内对象相互相似，而与其他簇中的对象相异。基于形心的划分技术就是使用簇Ci的形心代表该簇。在这里，我们使用簇内对象的均值作为簇Ci的形心，称之为K-Means聚类[10]。

（2）算法流程

K-Means聚类算法思想简单，效果却很好，是最有名的聚类算法。该聚类算法的具体步骤如下：

1. 随机初始化K个样本作为初始聚类中心。
2. 计算每个样本点到K个中心的距离，选择最近的中心作为其分类，直到所有样本点分类完毕。
3. 分别计算K个类中所有样本的质心，作为新的中心点，完成一轮迭代。
4. 重复步骤2），3），直到分类结果不再改变。

（3）优缺点比较

K-Means算法简单有效，易于实现，并且能应用到大大规模的数据集当中，但它常常终止于一个局部最优解，且需要提前设定分类数目k，并对噪声敏感。

### K-Medoids聚类

（1）基本定义

由于k-均值算法对离群点敏感，当有个别离散对象远离大多数数据时，该算法会将其分配到其中一个簇内，这会严重地扭曲该簇的均值。因此，我们采用K-Medoids方法[7]对其进行改进。K-Medoids算法相较于K-Means算法的不同点在于对中心点的选择标准不同。在K-means中，我们将中心点取为当前簇内所有数据点的平均值，而在K-medoids算法中，我们从当前簇内选取这样一个点——它到当前簇内的所有其它点的距离之和最小。

（2）算法流程

该具体的实现方法如下所示[8]：

1）任意选取K个对象作为初始聚类中心。

2）计算每个样本点到K个中心的距离，选择最近的中心作为其分类，直到 所有样本点分类完毕。

3）在每个类Oi中，顺序选取一个Or，计算用Or代替Oi后的消耗。若消耗变大，则保留原来的中心点，若消耗变小，则用Or代替Oi成为新的簇内中心点。

4）重复步骤2、3，直到不再发生变化。

值得注意的是，上述过程中，除了每次选出的簇内中心点保持不变外，其余点均要进行重新分配，得到一个新的簇类。

（3）优缺点比较

可以看出，K-Medoids算法在处理噪声或者离群点时，比k-Means算法更鲁棒，这是因为中心点不像均值那样容易受离群点或其他极端值得影响。然而，k-Medoids算法的每次迭代的复杂度是O(k(n-k)2)。当n和k的值较大时，这种计算的开销非常大，远高于k-Means算法。因此，当在大型数据集上运算时，可以采用抽取随机样本的方法，在样本空间上进行k-Medoids聚类。

## 基于模型的聚类算法

基于模型的聚类主要思想是假设数据集中的数据分布符合特定的数学模型，通过数学模型来发现聚类。它给每一个聚簇假定了一个模型，然后去寻找能够很好满足这个模型的数据集。这个模型可能是数据点在空间中的密度分布函数，它由一系列的概率分布决定，也可能通过基于标准的统计数字自动决定聚类的数目。它的一个潜在假定是：目标数据集是由一系列的概率分布所决定的。

主要有两种基于模型的方法：一种是统计学的方法，代表性算法是COB-WEB算法；另一种是神经网络的方法，代表性的算法是自组织映射神经网络。

### COBWEB聚类方法

COBWEB算法是一种增量概念聚类算法。这种算法不同于传统的聚类方法，它的聚类过程分为两步：首先进行聚类，然后给出特征描述。以一个分类树的形式来创建层次聚类，它的输入对象用分类属性-值对来描述。COBWEB的优点为：可以自动修正划分中类的数目；不需要用户提供输入参数。缺点为：COBWEB基于这样一个假设：在每个属性上的概率分布是彼此独立的。但这个假设并不总是成立。且对于偏斜的输入数据不是高度平衡的，它可能导致时间和空间复杂性的剧烈变化，不适用于聚类大型数据库的数据。因此，分类质量不再是单个对象的函数，而且也加入了对聚类结果的特征性描述。

### SOM聚类方法

（1）基本定义

自组织特征映射模型是对生物神经系统进化过程的计算机模拟,它的网络结构分为上下两层,下层为输入层,上层为输出层。输出层神经元构成一维或二维格形,格形确定了神经元在空间中的邻域关系;且输出层神经元与每个输入结点都有连接。若输入为n维向量,则每个输出神经元对应一个n维权值向量wj。自组织特征映射模型可以实现自组织功能,即SOM聚类[15]。

自组织的目的就是通过调整输入和输出之间的权系数wij,使神经网络收敛于一种表示形态。每执行一次学习,则SOM网络中就会对外部输入模式执行一次自组织适应过程。其结果是强化现行模式的映射形态,弱化以往模式的映射形态。

（2）算法流程

首先构造神经网络:

训练样本为L个互异的文本,设输入向量为X=[x1, x2, … , xn], n为全部文本中所有不重复的词的个数,构成输入神经网络的维数;权值向量为Wi=[wi1, wi2, … , win](i= 1,2,… ,m), m为输出神经网络的维数,即分类的个数,可根据实际情况设定,或者取输入维数的百分比。

其次训练SOM网络:

Step1：初始化。对初始对初始权值向量Wi(0)选择一组较小的随机值。同时,设置初始学习速率η0(0 < η0 < 1);设置迭代总次数为T, T一般取500次左右。

Step2：取一个文本作为训练样本,把该样本中的词作为本次训练的输入向量节点,数值为归一化后的词频再乘以该词的权值。

Step3：确定输出最大的神经元为获胜神经元。当输入向量为x且第c个神经元获胜时,等价于输入向量和权值向量的欧氏距离最小。

Step4：调整网络权值。学习速率随时间增加不断减小,拓扑邻域随着迭代次数的增加不断收缩。

Step5：用下一个样本继续训练网络,直到在特征映射里没有观察到明显的变化为止。

（3）算法优缺点

SOM神经网络在聚类方面有如下优点：

1. 无须用户指定聚类数目，网络通过学习过程自适应地确定聚类数目；
2. 因其采用“胜者全得”的学习策略，对噪音数据不敏感；
3. 具有可视化的优点;它采用的邻域学习策略能使数据从高维映射到低维时保持其拓扑结构不变，输出层神经元连接权矢量的空间分布能正确地反应输入模式的空间概率分布；

因此，SOM网络不但能学习到输入模式的类别特征，而且能够学习到输入模式在原始空间中的拓扑结构特征和概率分布，从而具备可视化的优点。无导师学习现在发展的还不成熟，传统SOM网络在文本聚类领域的应用还存在着许多的不足；

1）网络输出层结点的初始结构需要用户预先给出;输出层结点的初始拓扑结构与输入模式在在原始数据空间中的拓扑结构一致时，网络才会达到好的学习效果。但是由于文本数据高维性的特点，人们很难预先给出与原始数据空间中相一致的网络输出层拓扑结构。

2）网络训练时，有些输出层神经元的连接权值与输入模式相差很大，始终不能获胜，成为“死神经元”；其权值得不到任何学习训练的机会，进而影响文本聚类。

参考文献

[1] Zhang X, Zhao J, Lecun Y. Character-level Convolutional Networks for Text Classification[J]. 2015:649-657.

[2] Zhang X, Lecun Y. Text Understanding from Scratch[J]. Computer Science, 2015.

[3] 单丽莉,刘秉权,孙承杰. [文本分类中特征选择方法的比较与改进](http://kns.cnki.net/kcms/detail/detail.aspx?filename=HEBX2011S1069&dbcode=CJFQ&dbname=CJFD2011&v=)[J]. 哈尔滨工业大学学报. 2011(S1)

[4]Sebastiani, Fabrizio. "Machine learning in automated text categorization." ACM computing surveys (CSUR) 34.1 (2002): 1-47.

[5] 徐君军. 文本分类中特征选择算法的研究与改进[D].杭州电子科技大学,2016.

[6]王博, et al. "文本多分类中的特征选择研究." 计算机工程与科学32.8 (2010): 90-93.

[7]Wu, Ke, et al. "A probabilistic approach to feature selection for multi-class text categorization." Advances in Neural Networks–ISNN 2007 (2007): 1310-1317.

[8]Yan, Jun, et al. "OCFS: optimal orthogonal centroid feature selection for text categorization." Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2005.

[9]Xia, Ying, et al. "DEMST-KNN: A Novel Classification Framework to Solve Imbalanced Multi-class Problem." Computer Science On-line Conference. Springer, Cham, 2017.

[10]Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition40.7 (2007): 2038-2048.

[11] 邱定, 张激, 王金华,等. 基于Rocchio和KNN提出的新的文本分类技术[J]. 自动化与仪器仪表, 2017(8):107-110.

[12]Chen, Jingnian, et al. "Feature selection for text classification with Naïve Bayes." Expert Systems with Applications 36.3 (2009): 5432-5435.

[13] Oates T，Jesen D,the effects of training set sizes on decision tree[A]．Proc of the 14th Int'l Conf on Machine learning [c]．Nashville：Morgan Kaufman，1997．254-262．

[14] Breslow L A，Aha D w．Simlifying decision trees：a survey[J]．Knowledge Engineering Review，1997，12(1)：1-40．

[15]陈佳希. 基于支持向量机的文本分类[J]. 电子世界,2017,(07):64.

[16]David E Rumelhart , Geoffrey E Hintont, and Ronald J Williams. Learning representations by backpropagating errors.Nature,323(6088):533-536,1986.

[17]王泽. SVM算法在中文文本分类算法中的应用策略研究[J]. 通讯世界,2016,(01):225-226.

[18]Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. Journal of Machine Learning Reasearch (JMLR),3:1137-1155,2003

[19]Hongguang, Liu, Ma Shuanggang, and Liu Guifeng. "Classifying Chinese News Texts with Denoising Auto Encoder." Data Analysis and Knowledge Discovery 32.6 (2016): 12-19.