基于文本分类的形式概念分析

刘欣逸

作业	分数
得分	

2020 年 11 月 13 日

第 27 卷 第 2 期 计 算 机 学 报 Vol. 25 No. 2
2018 年 12 月 CHINESE JOURNAL OF COMPUTERS Dec. 2018

基于文本分类的形式概念分析

刘欣逸

（大连海事大学 计算机技术 辽宁省大连市 中国 116026）

摘 要： 本分类是数据挖掘技术的重要组成部分，为海量文本信息高效存储和挖掘提供了便利。在研究一般的文本分类模型基础之上，针对当前文本分类器在训练文本集较少的情况下容易出现过拟合现象，给出了一种基于形式概念分析的文本分类模型。该模型通过离散化文本的属性特征，构成形式背景，构建概念格，并把概念格中每个概念提取的分类规则作为文本分类的规则。其次，针对概念格分类规则提取的算法，该文给出了一种改进的提取分类规则的。最后，该文以卡方验证作为文本预处理时特征选择的算法，结合本文给出的文本分类模型，开发了基于形式概念分析的文本分类软件。
算法。

关键词： 文本分类；形式概念分析；概念格
中图法分类号：TP311.20 **DOI 号：** 10.3969/j.issn.1001-3695.2018.01.030

TEXT CLASSIFICATION BASED ON FORMAL CONCEPT

Xinyi Liu

Department of computer, Dalian Maritime University, City Dalian

Abstract： Text classification is an important part of data mining technology, which facilitates the efficient storage and mining of massive text information. Firstly, based on the study of the general text classification model, a text classification model based on formal concept analysis is proposed for the current text classifier in the case of less training text set. The model divides the attribute characteristics of the text, forms the background of the form, constructs the concept lattice, and classifies the classification rules extracted from each concept in the concept lattice as the rules of text classification. Secondly, for the algorithm of concept lattice classification rule extraction, this paper presents an improved algorithm for extracting classification rules. Finally, this paper uses the method of chi-square verification as feature selection method in text preprocessing, and combines the text categorization model given in this paper to develop text categorization software based on formal concept analysis.

Key words： Text Classification; Formal Concept Analysis; Concept Lattice

0 引言

随着 Internet 网的迅速发展，网络中每天将产生海量的文本数据，如何将这些数据进行归类存储（以便于后续的查找或者提取有用的信息做好准备）成为当今互联网面临的重要问题之一。文本中含有大量的信息，但由于 Internet 网的广泛性和开放性，在 Internet 网上发表的各种文字信息都是没有限制的，任何团体或者私人都可以在互联网上发布信息，因此对于搜索引擎如何规划组织这些信息，以便于查询者使用，成为当前十分有价值和意义的研究之一。

目前，搜索引擎成为全世界人们日常生活中获取信息的主要途径之一，我国人们日常生活中访问搜索引擎（如百度）的次数是上亿次级别的。搜索引擎成为人们生活中重要的获取准确信息的得力助手，包括获取各种各样的信息，改变人们生活和学习方式等，总而言之对人们的日常贡献是巨大的。搜索

引擎技术对于网页文本处理用到的技术有：采集、分词、存储、查询，对应到相关技术为：信息检索、自然语义处理、数据库、数据挖掘等技术[1,2,3]。本文主要研究的文本分类属于数据挖掘的领域。

1 文本分类

文本分类处理的过程包括两个方面，一个是文本分类模型的构建过程，另一个就是用模型进行预测的过程。文本分类模型构建过程包括：文本的预处理和文本分类器的训练。而文本的预测力过程又包括：将文本训练集进行分词、特征选词构建词典和计算每个文档词的 TF-IDF 值的三个步骤；文本分类器的训练，主要将由预处理得到的文档向量模型作为机器学习训练方法的训练集。常用的机器学习算法有朴素贝叶斯、支持向量机、神经网络和深度学习等；通过朴素贝叶斯训练得到的垃圾邮件预测的模型已投入到实际的使用之中。 而将模型运用于预测未知文本也需要进行文档预测出来的过程，将未知文档转换为对应的向量，才能作为模型的输入。

2 形式概念分析

德国数学家 Wille 在 1982 年的论文中发表提出了形式概念分析的理论。形式概念分析理论由序理论发展而来，是应用数学的一个分支。它的核心是概念格的构建，概念格体现的是“父概念”、“子概念”偏序关系，概念格中的任何一个概念集合都是按这种偏序关系来确定的。

2.1 形式背景和概念

定义 2.1 假设有一个背景$K=(U,M,I)$，若$A\subseteq U$，$B\subseteq M$,令

$$f(A)=\{m\in M|\forall u\in A,(u,m)\in I\},$$

$$g(B)=\{u\in U|\forall m\in A,(u,m)\in I\}$$

假如 A,B 满足 f(A)=B,g(B)=A，则背景中的一个概念用二元组(A,B)表示。B 是概念的内涵， A 称为概念的外延。形式背景生成所有的概念利用 B(K)和 B(U,M,I)简单表述。

定理 2.2 设(U,M,I) 是一个背景，若$X\subseteq U$,则有(g(f(X))， f(X))一定是概念，若$Y\subseteq M$,则(g(Y)， f(g(Y)))也一定是概念。反之概念都有(g(f(X))， f(X))且$X\subseteq U$的形式或(g(Y)， f(g(Y)))且$Y\subseteq M$的形式。

2.2 概念格的相关概念

定义 2.3 如果（A_1,B_1），（A_2,B_2）是一个背景上的连个概念，并且$A_1\subseteq B_1$（等价于$B_2\subseteq B_1$）因为$f(A_1)\supseteq f(A_2)$则称（A_1,B_1） 是 （A_2,B_2） 的子概念，（A_2,B_2）是（A_1,B_1）的父概念，并记作（A_1,B_1）≤（A_2,B_2），关系≤称为是概念的“层次序”，(简称“序”)。背景 K=(G,M,I)的所有概念用这种序组成的集合 B=(U,M,I) 表示,称它为背景 K=(G,M,I) 的概念格

2.2 概念格构造算法

当前概念格的生成算法主要分为两种：一种是概念格的批处理构造算法，另一种是概念格的增量构造算法。接下来介绍概念格构造主要的一些算法。

两项任务构造概念格批处理算法需要完成：一是要生成形式背景的所有概念，也就是生成概念格的节点。二是要建立这些节点之间偏序的关系，弄清子节点和父节点。根据这连个要求，将批处理算法分为两种模型：任务交叉生成模型和任务分割生成模型。两种模型构造概念和概念的关系的顺序不一样，任务分割生成模型算法顺序是：在生成背景全部的概念集之后，再找到概念之间的关系；任务交叉生成模型的顺序是：在生成概念的同时生成对应的关系。

增量算法的原理是基本一致的，它的主要原理是将要插入的概念与已经构建的概念格进行对比，然后与构建的概念格中的所有的概念进行交的操作，根据得到的结果再进行下一步，批处理算法和增量处理算法主要的不同在于生成概念之间的关系。增量的算法的代表有：Capineto 算法和 Godin 算法。

3 形式概念分析的文本分类

基于形式概念分析的文本分类模型的构建过程和一般的文本分类模型过来过程是相似的。由于自然语义发展比较缓慢，因此文本分类主要是基于统计学习。现在大多

文本分类是建立在特征词向量空间的，并没有考虑到文中语义和词与词之间的联系，特征词的维数随着训练集数量的增加会变得很大，导致构建模型的空间复杂度和时间复杂度大大增加。此外当文本训练集很少的情况下容易出现过拟合的现象。本文中我们借助概念格理论，将通过概念格挖掘分类规则。将文本作为对象，对文本训练集进行分词和特征抽取，从而构建模型的输入空间，也称为字典；然后将每个训练集文本对象通过字典，构建成文档对象对应的属性形式；从而形成了基于文档对象属性的形式背景，接着构建概念格，然后从概念格中分类规则，根据分类规则实现文本分类的目的。

3.1 整体结构

文本分类系统分为两个阶段：训练阶段，分类阶段。 训练阶段的功能是：确定模型的输入空间和根据机器学习的算法训练得到文本的分类规则。这里包括的主要模块有：文本的预处理和分类模型的训练。文本的预处理是将训练的文本进行分词，构建字典来确定模型的输入空间，然后计算每个文档的 TF-IDF 值，然后根据 TF-IDF 进行离散化，然后将文档表示为离散化后的特征向量的形式。分类模型的训练是将文档离散化后的特征向量填入表中构成形式背景，然后构建概念格（概念格的构建是构建模型的核心），因为接下来的一个步骤是从概念中提取分类规则。分类规则是最终的目的，用于对未知样本进行分类。

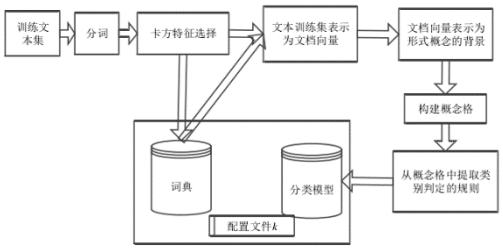

图 4-1 基于形式概念分析的文本分类模型构建的总流程图

图 1 基于形式概念分析总流程图

3.2 基于概念格的文本分类模型的构建

本文运用到的形式概念提取分类规则算法和 Sugiyama 提出的 SELF 算法的思想类似，其算法的主要思想是通过构造相应的概念格，然后通过概念中对象的类别来得到相应的类别判定规则；因为在形式概念中概念的生成过程实际上就是属性聚类的一个过程，因此每一个概念相当于训练集在某一种或者多种属性上的聚类过程，Sugiyama 提取分类规则的核心思想任务如果一个概念中所有的对象集属于同一类，那么就任务概念中对应的属性集具有旁别类别的能力，可以将属性作为判定类别的一条规则。SELF 算法主要在 JSM-method 算法的基础之上加入了连续属性的离散化的模块。基于形式概念分析的文本分类模型的构建的核心其实就是概念格的构建，概念格是形式概念分析数据的核心。得到概念格之后，就可以从中得到挖取相关的分类规则，而分类规则的提取是形式概念分析应用的重要手段，需要通过不同提取概念格中的有用信息方法来实现。本文主要的改进也是对提取分类规则的方法进行了改进。

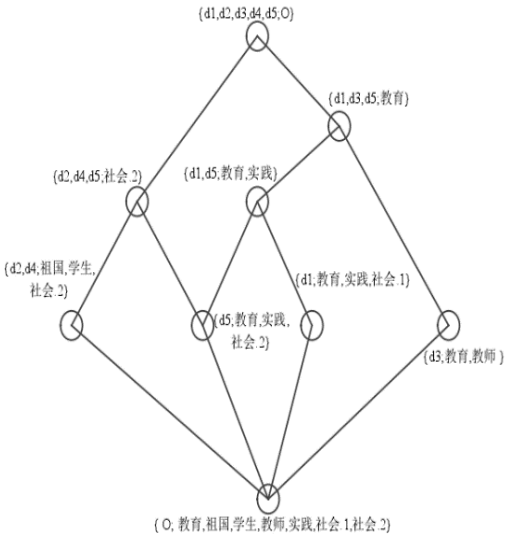

图 2 生成的概念格

3.3 基于形式概念分析的文本分类模型

综上所述，我们得到基于概念格的文本分类模型的构建过程。我们把分类的模型

构建分为两个模块，文本预处理的模块和基于形式概念挖掘分类规则的模块。第一个我们首先要做的是文本的预处理，预处理我们首先是将训练集的文本进行分词，并且通过特征选择的方法来实现输入特征降维的效果，然后把降维后的词存放在一起构成字典，最后是计算每一个文档对应的词的 TF-IDF 的值，然后通过离散化得到文档对象和它属性的集合，为形式背景的构建做好

准备。

第二个就是根据预处理得到的对象和属性集来构建对应的形式背景，接着是构建概念格，概念格的构建是应用形式概念分类的核心，因为我们的分类规则是根据概念格的性质来进行操作的。当然论文的另一个创新点，是改进从概念格中提取分类规则的方法。最后是利用提取的分类规则预测未知的样本。

4 文本分类模型软件设计

软件实现的功能包括五个基本模块：文本的预处理、形式背景构建、概念格生成、分类规则提取、为分类文本的预测，整体功能实现结构体如图 3 所示：

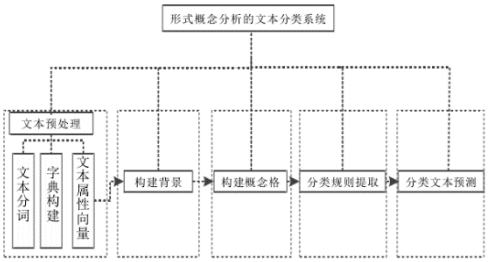

图 3 软件实现的模板功能

4.1 预处理模块

文本预处理模块分为分词、建立词典、建立文档向量属性三个小模块。

(1)分词：首先进行的是中文分词的处理。软件在分词系统之后还对得到的词语进行了过滤，去除了停止此(介词等)以及一个字的词，从而留下对文本分类贡献比较有意义的词，并且统计了每个词在每篇文本中出现的次数，这为后面选取卡方验证和计算 TF-IDF 做好准备。

(2)特征选择：根据文本处理的一般流程我们知道，除了分类算法以外，文本的输入特征空间对文本模型的影响也是很大的。因为如果文本训练集很大，那么通过分词得到的字典的词数会很多，字典的词一多会导致输入空间的维度很大，能达到上万维的空间，这样会导致算法的计算量大大的增加，因此需要通过相应的方法进行特征的降维。那么在文本处理过程中计算词的互信息，文档的频率这些都是选择特征的好方法。

(3)离散化形式背景生成文本属性向量：文本预处理最后是将每个文档生成对应的属性向量。这里面首先在每一篇文档里面找字典里面对应的词，然后通过词在文档中出现的次数和在所有训练集文档中出现的次数来计算 TF-IDF 的值，这样就能够初步确认每个文档训练集的向量模型。然后通过离散化属性的值的算法进行离散化，得到最终的属性。

4.2 概念格构建模块

概念格的构造过程其实是概念进行聚类的过程，是形式概念应用的重要前提。对于形式背景中属性大小是数量级以上的话，概念格的构造算法是当前形式概念研究的主要问题。目前概念格构造的算法有两种方法：批处理和增量算法。虽然算法不同，但是对于同一个形式背景，最终构造得到的概念格是唯一的，所以数据形式背景的分布对概念格是没有影响的。本次试验用的概念格生成算法是：批处理算法，先得到所有的概念节点，然后再通过概念节点的关系得各个概念之间的关系，得到父节点和子节点的关系，也就是概念格的边。

4.3 分类规则提取模块

算法的具体流程为：从概念格的最顶端遍历每个节点，若一个概念节点中的对象属于一类，那么这个属性就作为一个规则。

4.4 实验评估

实验通过查准率、查全率和 F 值对构建的文本分类模型进行评估[11]。查全率越大说明这个类别的数据被分类正确的概率越大，而准确率，查准率意味着分类器预测一个类的准确程度，而 F 值是查全率和查准率的综合。 本文通过复旦大学李荣陆提供文本分类语料库[12]。针对其中的教育类的 100 个文本集和医疗类的 100 个文本集的数据，将其中的教育类的 30 个数据和医疗类的 30 个数据作为训练集，而将另外的教育类的 70 个数据集和医疗类的 70 个数据集，总共 140 个测试集，下面是本文构建的基于形式概念分析的文本分类模型的分类测试实验结果，见表 1。

表 1 基于形式概念文本分类模型实验结果

文本训练集个数	教育 P	教育 R	教育 F	医疗 P	教育 R	医疗 F
3	0.617	0.82	0.707	0.739	0.485	0.586
5	0.913	0.600	0.724	0.702	0.942	0.804
7	0.817	0.957	0.881	0.948	0.785	0.859
9	0.822	0.928	0.872	0.918	0.800	0.855
10	0.839	0.971	0.901	0.966	0.814	0.884
12	0.809	0.971	0.883	0.964	0.771	0.857
15	0.690	0.985	0.812	0.957	0.557	0.709
20	0.742	0.985	0.846	0.978	0.657	0.786
30	0.673	1.00	0.804	1.000	0.514	0.679

由于 F 值相比 R 指标或者 P 指标，更能综合的反映文本分类的效果，因此本次试验主要将 F 值作为评估的标准。为了更加直观的看到随着训练集的增加,F 值的变化，通过下面的折线图 4 来观察。

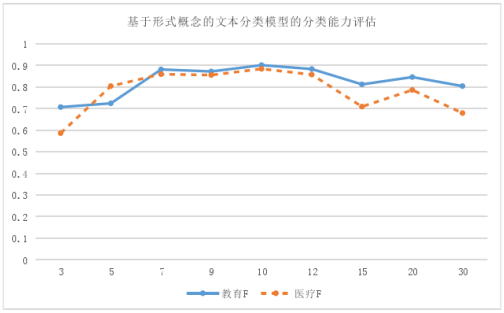

图 4 F 值试验结果折线图

由图 4 的实验结果可以观察得到，当文本训练集非常少的情况下，模型的 F 值小，分类能力差，例如，当每个类别文档只有三个时，模型的 F 值为 0.6 和 0.7。随着训练集文档的增加模型的 F 值也增加，即分类能力也在提升。当每类训练集的文档数量增加到 10 个的时候，模型的 F 值为 0.9 左右，这个时候模型的分类能力最强。但是这之后，随着训练集的增加，可以看到 F 值不增加，反而下降，分类能力减弱。分析出现这种现象的原因，可能是由于随着训练集的增多，在预处理的时候构建的字典就会越大，加入了减弱模型分类能力的词；能想到解决这个问题的方法是：随着文本训练集的增加，通过提高加入字典的卡方值的条件，来限制加入字典的词。一般来说，文本分类模型的 F 值能到 90%以上就能够被认为是成功的。由图 4 可以观察到，适当的调整训练集，模型的 F 值能够到达 90%左右，因此模型的分类能力还是比较强的，验证了本文基于形式概念分析的分类模型的正确性和有效性。

5 结论

由于当今互联网每天都产生大量的文本信息，如何更好的存储管理文本信息，以及从这些文本中挖掘有用的知识成为研究的热点，文本分类是存储和挖掘文本信息重要的手段近年来受到广大学者的重视。

（1）一种基于形式概念分析的文本分类模型。对提出基于形式概念分析的文本分类模型进行了原理的讲述，从文本的预处理（构建文档的向量表达模型），到构建所有训练集文档的形式背景，再到概念格的构建，以及从概念格中提取分类规则的算法，并对提取的分类规则做了改进。模型构建的每一步都给出了具体的实例，关键的步骤都给出了具体的操作方法。

(2) 提出了分类规则的改进算法。研究基于分析形式概念分类规则提取的主要算法，根据属性和规则之间的关系，提出了分类规则的改进算法，将固定规则比较转换为属性权重的比较。该算法不仅可以提取更多的分类规则，同时也增加了规则的应用范围，也减少了判定时的复杂度，并且给出了算法的实现步骤和应用。

(3) 通过软件实现了对模型构建的过程。软件的功能实现了文本的预处理(分词，建立字典等)、训练集文档形式背景的构建、通过背景构建概念格、通过编写算法 4.2 实现文本分类规则提取的功能，并通过两类文档(每类 3 篇)演示了模型的构建过程。最后通过公开的文本集测试了和标准的验证指标，验证了模型的正确性和有效性。

参考文献

[1] Obitko M, Snásel V, Smid J. Ontology Design with Formal Concept Analysis[J]. CLA, 2004, 128(3): 1377-1390.

[2] Formica A. Ontology-based concept similarity in Formal Concept Analysis[J]. Information Sciences, 2006, 176(18): 2624-2641.

[3] Bendaoud R, Napoli A, Toussaint Y. Formal

Concept Analysis: A Unified Framework for Building and Refining Ontologies[M]. Berlin: Springer Berlin Heidelberg, 2014: 21-28.

[4] Eisenbarth T, Koschke R, Simon D,et al. Locating features in source code[J]. IEEE Transactions on software engineering, 2003, 29(3): 210-224.

[5] Al-Ekram R. Software Reliability Growth Modeling and Prediction[J]. Department of Electrical and Computer Engineering, 2012, 30(1): 34-38.

[6] Stumme G, Maedche A. FCA-Merge: Bottom-up merging of ontologies[C]//IJCAI, Berlin, Germany, 2012: 225-230.

[7] Robertson, S.E, Rijsbergen V,etc. Probabilistic models of indexing and searching[J]. Oddy R.n.et Al.information Retrieval Research, 1980, 3(3): 54-56.

[8] Grove C. The Lexical Approach: The State of ELT and a Way Forward, by Michael Lewis[J]. Tesol Quarterly, 1993, 28(4): 828-828.

[9] Bengio Y. Learning deep architectures for AI[J]. Foundations and trends® in Machine Learning, 2009, 2(1): 123-127.

[10] Weston J, Ratle F, Mobahi H, et al. Deep learning via semi-supervised embedding[M]. Berlin: Springer Berlin Heidelberg, 2012: 639-655.

[11] Joachims T. Text categorization with support vector machines: Learning with many relevant features[C]//European conference on machine learning. Springer Berlin Heidelberg, Germany, 1998: 137-142.

[12] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]//ICML, New York , America. 1996: 148-156.