DataMining final project*

杜文哲[1+]

[1](南京大学 计算机科学与技术系,南京 210000)

DataMining final project*

Du Wen-zhe[1+]

[1](Department of Computer Science and Technology, Nanjing University, Nanjing 210000, China)

Abstract: 本次数据挖掘大作业基本上属于命名实体歧义消解和指代消解,对于每个领域的科研工作者和学生来说,搜索和阅读学术论文是他们的重要工作之一.当科研人员想深入了解某一领域时,通常会搜索该领域的研究者,并找到这些研究者的学术论文进行学习. 然而,由于很多论文作者姓名相同,特别是对于国内学者来说,很多中文名不同的学者英文名也相同(例如张三和章三的英文名同为 San Zhang). 这导致很多论文被错误划分到其他同名作者的著作中.与此同时,同一个姓名可能有不同的缩写方式(例如 J. Doe,Jane Doe 和 J. A. Doe),导致无法将同一个作者的所有论文汇总到一起.这给论文搜索和数据整理都带来很大问题. 因此,正确匹配论文和作者信息非常重要.本次任务要求从给定数据中挖掘论文和论文作者之间的关系,从而构建模型来自动预测论文是否为给定作者所著,由 sample_submission.csv 文件的格式判断为二分类问题,故问题的解决思路为从数据中抽取特征建立分类器.

Key words: Named entity ambiguity resolution,PaperAuthor-matching, random forest, KDD cup 2013

1 对于本次任务的理解

1.1 任务介绍

与作者相关的信息主要在 Author.csv 文件中,包含唯一的作者 ID,作者姓名,作者机构.同一个作者可能由于名字的不同缩写方式(例如 J. Doe,Jane Doe 和 J. A. Doe)在文件中出现多次. 与论文相关的信息主要在 Paper.csv 中,包含唯一的论文 ID,论文题目,发表年份,会议/期刊 ID,关键词等.与会议/期刊相关的信息主要在 Conference.csv 和 Journal.csv 中,主要包含会议/期刊 ID,会议/期刊的简称, 会议/期刊全称,主页地址.PaperAuthor.csv 是带有噪声的论文-作者对信息,主要包含论文 ID,作者 ID,论文中标注的作者名,论文中标注的机构.由于同一姓名的不同简写以及不同作者间可能存在的重名,因此该文件中的 PaperID-AuthorID 对很可能存在错误,不能将该文件的 PaperID-AuthorID 对当做 label 或 ground truth.[2]

1.2 个人理解

通过观察发现,author.csv 与 paper.csv 文件都有噪声,也就是对于每一个 pair (author, paper)的对应关系并不一定正确,并且数据中有大量的缺失值,例如 paper_author.csv 文件中的 affiliation 就有大量的缺失,此时若用

传统的取均值取众数加权等等方法,由于是affiliation是字符型,此类方法的效果并不好,通过检索学习了许多提取特征的处理办法,于后续列出.最后的任务是判断每个作者 ID 对应的论文 ID 是否正确,故将问题看成是二分类问题,提取不同数据表的数据进行特征的组合提取,统一合并和打入分类器中进行预测,由于时间关系最后只尝试了 sklearn 库中的 random forest,后续有时间会再尝试手动实现随机森林与其他的分类器. 而且由于任务一中测试集的 ID 出现在了训练集里,对于特征提取的要求相对不高,而任务二由于完全属于未知的泛化任务,所以相关的分数都较任务一低.

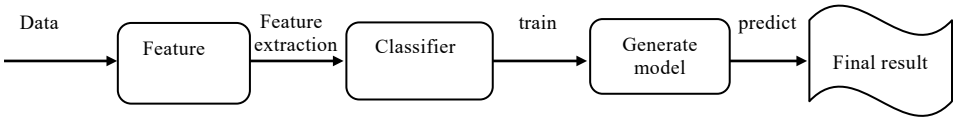

图 1　作业实现的基本框架

2　数据处理与特征提取

2.1 数据处理

首先,表 1-5 列出了相关数据集的各种信息：数据,数据类型,与相应的数据介绍. 见下表：

Data	Data type	Instruction
Id	Int	Author id
Name	Char	Author name
Affiliation	Char	The Author’s affiliation

表 1 author.csv

Data	Data type	Instruction
Id	Int	Paper id
Title	Char	Paper name
Year	Int	Publishment year of paper
Conferenceid	Int	Conference id where paper published
Journalid	Int	Journal id where paper published
Keywords	char	Keywords of paper

表 2 Paper.csv

Data	Data type	Instruction
PaperId	Int	Paper id
AuthorId	Int	Author id
Name	Char	Author name

Affiliation	Char	The Author's affiliation

表 3 PaperAuthor.csv

Data	Data type	Instruction
Id	Int	Conference/Journal id
ShortName	Char	Short name of conference/journal
Fullname	Char	Full name
Homepage	Char	Homepage URL of conference/journal

表 4 Conference/Journal.csv

Data	Data type	Instruction
AuthorId	Int	Author id
DeletedPaperIds	Char	False label for pair(author,paper)
ConfirmedPaperIds	Char	True label for pair(author,paper)

表 5 train.csv

由于原始数据的缺失值较多,且有噪声,需要先对数据进行预处理,进行初步的特征提取操作,清楚噪声并提取出有用的信息.在 preprocess_data.py 文件中,进行了数据预处理,提取了 paper to author, author to coauthor, author's publication, combination of name and affiliation, keywords frequency 等几个特征作为主要特征进行训练.

首先,提取 PaperAuthor.csv 文件中的 paper to author,即找出一篇文章的作者列表.原因是,对于一篇论文的共同作者,我们可以认为他们是有关联的,并且出现在一篇论文的共同作者,我们或许可以将他们标记上相同的 keywords(来自 paper.csv),同时又因为 PaperAuthor.csv 出现了 affiliation, 后续也可以通过加入 affiliation 特征合并,因此,一篇文章的共同作者具有相似性的可解释性非常强.我们将同一篇文章的共同作者提取出来存入 paper2author 进行后续使用.

其次 author to coauthor 是基于上述的 paper to author 更进一步,我们在找到一篇文章的共同作者后,可以基于此,寻找 topk 个有关联的共同作者,也就是说,我们通过设置 topk 值,让不同的共同作者之间产生联系,从而形成了一个特征,这个特征是关于作者之间的.而上述的 paper to author 的特征是关于 paper 和 coauthor 的.

接着,在 Conference.csv 和 Journal.csv 文件中提取作者的论著清单,作为一个清洗完成后的数据,因为数据集 Conference 和 Journal 大体上十分相似其中的 URL 与 short-name, full-name 基本属于无关的特征,我们只需将其合并打入一个字典 author's publication 中即可.

同时,combination of name and affiliation 也是非常重要的预处理的一步,因为 paperauthor.csv 文件中的噪声主要来源于：对于一个 pair (authorid , paperid),它可能出现不止一次,所以此处我们将相应的 name 与 affiliation 合并起来,即:

pair(authorid, paperid) , name1 ## name2 ## name3, aff1 ## aff2 ## aff3

形式,通过 paperauthor.csv 可以获得论文的 Author name 与 affiliation, 同时根据 pair (authorid, paperid) 中的 author id,我们可以在 author.csv 中找到对应的 name 与 affiliation,生成 pair(authorid, paperid), name_a, aff_a, 作为两个数据特征,在后续计算字符串的距离.

最后,在 paper.csv 文件中出现了文章的 keyword, 即如上文首先所述,通过 paper_author 与 paper 两表,我们可以提取出每个作者发表过文章的关键词集合,作为该作者的属性,也就是一个特征,同时用 keywords frequency 记录关键词出现的次数,以此判断作者更贴近于哪个关键词.

2.2 特征提取

2.2.1 共同作者

每篇论文都会有多个共同作者,我们可以根据 paper to author, author to coauthor 数据特征进行进一步的处理,依照前文所说的设置 topk 值(即出现频率排名前 k 个的数据),找到每个作者相关联的前 k 个共同作者(记为集合 Co),对于来自 paper_author.csv 中的 pair (authorid, paperid),计算论文 paperid 的作者(该集合记为 A)是否出现在 authorid 的前 topk 个共同作者中,这里采用了两种方法进行计算:

1） 简单计数,即出现了几个就用一个 list 记录下来.

$$Count = | Co \cap A |$$

2） 还可以将出现 paperid 论文的 topk 个共同作者(来自于集合 Co),与同一个 pair (paperid, authorid) 中 authorid 作者的合作次数,作为一个共同作者的相关性特征,可以视为该共同作者的出现可能性.

2.2.2 字符串距离

在数据预处理时我们得到了如下表 6 的数据特征

	Author.csv	Paper_Author.csv
name	name-a	name1 ## name2
affiliation	aff-a	aff1 ## aff2

aff:affiliation.

表 6 combination of name and affiliation

因为 paperauthor 文件与 author 文件中的作者并不完全对应,这便是产生噪声的原因,所以该特征我们通过计算来自 author 表中的 name-a 与 paperauthor 表中的 name1 , name2…的字符串距离,来表示作者名字的相似度,以此能比较好的降低噪声的影响.这里距离的度量可以采用字符串动态规划的经典算法：编辑距离,最长公共子序列(LCS),最长公共子串(LSS)等. 最后将所计算出的所以距离存入列表中,最大程度地各自表示了 author 的 name 和 author 的 affiliation 与 pair(authorid, paperid) , name1 ## name2 ## name3, aff1 ## aff2 ## aff3 的相似度.

同时,也可以通过三种字符串距离度量取均值,来强化该项特征.

2.2.3 关键词

论文的关键词是非常重要和有效的特征,将论文的 title 和 keyword 进行拼接,拼接为 keywords,利用自然语言处理中常用的分词技术,把句号,冒号等无用的停词去掉.

这里同共同作者,我们可以采取两种方法计算

1)我们把作者以往写过论文的关键词抽取出来作为集合 T,记录下当前论文的 keywords 与集合 T 的重复个数：

$$num = | T \cap keywords |$$

2)出现在 keywords 中的每个关键词若也出现在了集合 T 中,我们便将它在集合 T 中出现的次数作为进行累加. 这个特征可以获得作者比较符合的关键词属性,从而若在测试集中出现了这个作者,我们便能从文章的 keyword 判断是否满足作者的关键词特征.

2.2.4 论著

统计作者发表过的论文在不同期刊和会议的数量也可以作为一个特征,这么做的原因是,可能每个作者有不同发表倾向性,即作者 A 可能偏爱发会议 B,基于此,我们对作者发表的期刊和论文进行统计作为一个特征存入列表中.也就是,我们将作者的偏爱值表示为过去论著发表的刊物集合与当前论文 paperid 的 journal/conference 的相似度,通过计数来量化这个偏好.

3 分类器

由于时间关系,仅采用了 sklearn 库中的随机森林. 简单来说,随机森林是一个包含多个决策树的分类器,并且其输出的类别是由个别树输出的类别的众数而定, 是一种集成学习的方法,也是一种准确率较高的分类器,学习的过程十分的迅速,因为内有多个决策树投票,方便平衡误差.

基本的思路为：1) 进行数据的随机选择,重复随机采样构造子数据集. 2) 用采样出的子数据集构造子决策树,每个子决策树纳入这个数据并输出一个结果. 3) 如果有了新的数据需要通过随机森林得到分类结果,就可以通过对子决策树的判断结果投票,得到随机森林的输出结果. 因而随机森林也就是集成学习的一种方法.

4 实验结果

baseline1： 只提取发文总数作者总数,作者在不同会议/期刊的发表次数,作者与其他作者的合作次数.[1]
baseline2： 不进行对三种度量距离方法取平均值强化特征的结果
all features: 对以上所有特征进行采纳分类的结果

method	result_track1(%)	result_track2(%)
baseline1(without coauthor)	85.04	
baseline2(without mean string distance)	95.14	93.10
all features	96.07	92.71

表 7 结果

5 总结

观察表 7,我们可以发现,相较于初步提取特征数据的 baseline 而言,进一步通过字符串距离计算相似度,引入论著信息,关键词,论文作者等等可以较好地提高表现.但同时,对于字符串距离取均值进一步强化特征通过图表可以看出,与不取均值的表现相差无几,各有优劣,由此得出,对于同一个类别的特征进行反复提取,就像是沉没成本,能获得提升的效果有限,不如寻找新的特征关系进行尝试.

References:

[1] Ben hm. KDD Cup 2013 Author Paper Identification, 2013. https://github.com/benhamner/Kdd2013AuthorPaperIdentification.

附中文参考文献:

[2] 黎铭. Data Mining Practice, 2021. http://www.lamda.nju.edu.cn/mijw/MiningPractice.html.