问政留言文本的分类与热点挖掘

摘 要:

近年来，文本数据量不断攀升，为以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。所以使用自然语言处理技术实现留言文本的自动分类与热点挖掘变得相当重要。

对于问题一，进行文本分类。共有 9210 条留言，首先对留言详情进行命名体识别，分词和去停用词处理。然后取 tf-idf 与信息增益得分排名前 4000 的特征的交集，共得到 1811 个特征。用这些特征构建留言文本词频向量，并通过 tf-idf 得分加权后，分别用于多项分布贝叶斯，线性 SVM，随机森林与多层感知器(MLP)进行训练。结果显示线性 SVM 分类效果最好，总体 f1-score 为 0.859；多项分布朴素贝叶斯效果也较理想，总体 f1-score 为 0.847。分类文本中交通运输类的分类最困难，其召回率不到 0.700。接下来探究了 Doc2vec 模型的参数，通过控制变量对参数进行调整，在减少了模型的过拟合程度后，用于分类器训练。其中 MLP 训练结果最好，总体 f1-score 为 0.807。

对于问题二，进行热点挖掘。先基于 DBSCAN 算法构建了一个先聚类，再验证，最后合并的挖掘热点信息的模型。然后基于同一个聚类内每份留言的时间，反对数，点赞数提出了一个计算热点指标的算法，并通过改造 sigmoid 函数将该算法计算出来的值映射到 0~10 空间上，得到绝对热点指标，即 10 为满热度，0 为无热度。最后将模型用于实验数据，通过热点指数的密度分布，一定程度上说明了指标构造的合理性，并得到 2019 年 A 市热度指数最高的事件是“伊景园滨河苑捆绑销售车位”。

对于问题三，提出了对留言的答复意见的评价方案。从答复的相关性、可解释性、完整性、及时性、言辞友好性对答复做出评价。使用层次分析法对各因素的重要程度进行量化；通过答复与留言的特征项集合的交集体现相关性，答复长度与留言字符数的比值量化完整性，并规定$\beta_i = \frac{|D \cap T| + |D|}{m}$以量化答复的可解释性；将答复综合评价结果规范至[0,100]。

关键词： tf-idf 特征向量 多层感知器 Doc2vec 挖掘模型 DBSCAN

Classification and Hot Spot Mining of the Text of The Asking Political Message

Abstract: In recent years, the volume of text data has been climbing, which has brought great challenges to the work of relevant departments that mainly rely on human persons for the division of messages and hot spots. Therefore, the use of natural language processing technology to achieve automatic classification of message text and hot spot mining has become very important.

For question one, the text is classified. A total of 9210 messages, First identify the naming body of the 9210 message details, and divide the words, select to deactivate the words. Then take the intersection of the characteristics of tf-idf and the top 4000 features of the information gain score, and get a total of 1811 features. These features are used to construct the message text word frequency vector, and after weighting the tf-idf score, they are used for training in multiple distributions Bayes, linear SVM, random forest and multi-layer perceptron (MLP). The results showed that the linear SVM classification effect was the best, with 0.859 in total f1-score, and the ideal effect of simple Bayesian in multiple distributions, and 0.847 for the total f1-score. The classification of transportation categories is the most difficult in the classification text, with a recall rate of less than 0.700. Next, the parameters of the Doc2vec model are explored, and the parameters are adjusted by the control variables, which are used for classifier training after reducing the degree of overfitting of the model. Among them, MLP training results are the best, the overall f1-score is 0.807.

For question two, hot spot mining is carried out. Based on the DBSCAN algorithm, a model of first clustering, then validation, and finally merging the mining hot spot information is constructed. Then based on the time of each message in the same cluster, the number of objections, the number of likes proposed an algorithm to calculate the hot spot indicator, and by transforming the sigmoid function to the algorithm calculated the value of the value mapped to 0 to 10 space, to get the absolute hot spot indicator, that is, 10 for full heat, 0 for no heat. Finally, the model is used for experimental data, through the density distribution of hot spot index, to some extent,

to explain the rationality of the index structure, and the highest heat index in 2019 A city is "Yijingyuan Riverside Garden bundled car space."

For question three, the evaluation scheme for the reply comments on the message was put forward. The response is evaluated in terms of relevance, interpretability, completeness, timeliness, and verbal friendliness. Using hierarchical analysis to quantify the importance of each factor, the ratio of the length of the reply to the number of characters in the message is quantified by the collective present correlation between the reply and the set of features of the message, and the ratio of the reply length to the number of message characters is quantified, and prescribe $\beta_i = \frac{|D \cap T| + |D|}{m}$ to quantify the interpretability of the response. The responses to the comprehensive evaluation results are standardized to [0,100].

Keywords: tf-idf feature vector Multi-layer perceptron Doc2vec mining model DBSCAN

目录

一、挖掘目标

1.1 背景

NLP(Natural Language Processing),自然语言处理技术正在迅速成长，其中很多理论和方法在大量新的语言技术中得到广泛应用。NLP 一些经典的应用场景包括文本分类、文本情感分析、文本相似度分析等。而在进行文本分析，最常用的构建词向量的模型有词袋模型(BOW)与 word2vec[1]。

本次数据挖掘就是这门技术的一次应用。如今，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、 汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依 靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

1.2 目标

本次建模的主要任务是基于群众问政留言记录，建立一个文本分类模型，以及一个能发现热点问题的模型。对于文本分类器，它必须能有效的按照留言的第一标签对留言进行分类，除了要具备很高的精确率，还要有足够高的召回率。只有这两个指标都足够高的前提下，一个文本分类器才能代替人工，实现自动分类。对于热点挖掘模型，要尽可能将反应同一个问题的留言进行聚类，再根据一个合理的热度指数计算公式计算出各个问题的热度，从而实现对热点问题的排序。除了这两个模型外，我们还提出一套评价方案，用于对留言答复意见的评价。

二、群众问政留言分类

2.1 数据预处理

对于中文文本，在进行 NLP 之前，最重要的一步就是对文本进行合理的分词。这一步的好坏对后面模型的精确与否有深刻的影响。我们对留言文本分词采用的工具是 jieba。Jieba 是 python 的一个中文分词模块[2]，GitHub 上称其为最好的中文分词组件。这里我们注意到对于问政系统的留言中存在着大量的地名以及机构名。仅仅使用 jieba 分词，很容易将这些比较长的地名以及机构名拆分成一个个词语，这样很容易破坏留言原本的语言，给模型的训练带来困难。因为这些地名与机构名是在留言中往往以高频词出现的。

基于上面的问题，我们必须要进行命名体识别，以期望提高后面模型的精度。所谓命名体识别，就是指从文本中识别出命名性指称项。狭义上，是识别出人名、地名和组织机构名这三类命名实体。这里我们借助百度提供的 AI 开放平台进行命名体识别，得到留言文本中

的地名以及机构名后，再将它们加入 jieba 词库进行分词。

最后一步是对留言进行停用词处理。在文本处理中，停用词是指没有信息量的词，其功能及出现极其普遍，通常是一些单字，单字母以及高频的单词，在中文中如“我、的、吗、了”等，英文中如的“the、this、an、a、of”等。一般在预处理阶段便将其删除，避免对文本造成负面影响[2]。此处采用使用特定语言的停用词列表的方法对本文所用的停用词进行删除——停用词取自百度停用词表。我们对文本预处理的过程可以用图 1 表示。

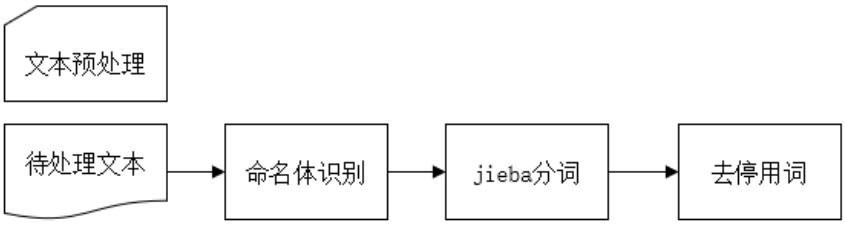


图 1　文本预处理流程图

2.2 特征抽取

文本的表示及其特征项的选取是文本挖掘的一个基本问题，它把从文本中抽取出的特征词进行量化来表示文本信息。目前人们通常采用向量空间模型(VSM)来描述文本向量,但是如果直接用分词算法和词频统计方法得到的特征项来表示文本向量中的各个维,那么这个向量的维度将是非常的大。这种未经处理的文本矢量不仅给后续工作带来巨大的计算开销,使整个处理过程的效率非常低下,而且会损害分类、聚类算法的精确性,从而使所得到的结果很难令人满意。因此,必须对文本向量做进一步处理,在保证原文含义的基础上,找出对文本特征类别最具代表性的文本特征。为了解决这个问题,最有效的办法就是通过特征选择来降维。

我们主要选择两种评估方法来进行文本特征的选取: TF-IDF 算法与信息增益。这两个评估函数通过对文本所有特征进行评估并打分，分数也就是权重。然后将所有特征按权重大小排序,提取预定数目的最优特征作为提取结果的特征子集。由此可见，特征提取的效果与评估函数的质量息息相关。为此我们尝试合并这两个评估函数，来提高特征提取的效果。

2.2.1 TF-IDF 算法

文本分类最常用的算法就是 TF-IDF 算法，其中 TF 称为词频(term-frequency),用以计算该词描述文档内容的能力，ID 称为逆文档词频（ inverse document-frequency)，用于计算该词区分文档的能力[3]。使用 tf-idf 算法代替原始文档的词频的目的是为了降低高频词的影响以及提高低频词的影响。从经验上讲，对于在每个文档中都出现的词可以认为是无用的，而那些只出现在一部分文档中的词对分类更为有用。计算 tf-idf 的公式为 2.1。公式中t代表特征，也就是单词。d代表某一个文档。

$$tf - idf(t,d) = tf(t,d) * idf(t) \qquad 2.1$$

这里我们采用的$idf(t,d)$的计算公式如 2.2。(这里和很多参考书的标准定义是不同的)其中n是文档的总数量，$df(t)$是文档集中出现特征t的文档数量。最后加 1 是为了不忽略出现在所有文档中的单词。$tf(t,d)$代表出现在某篇文档d中特征t的数量。

$$idf(t) = ln\frac{1+n}{1+d(t)} + 1 \quad (2.2)$$

从公式中可以看出，字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

2.2.2 信息增益

信息增益也是有效的特征选择方法，该方法主要是看特征能够给分类系统带来多少信息，带来的信息越多，该特征越重要[4]。在信息论中关于信息熵的定义如式 2.3。

$$H(X) = -\sum_{i=1}^{n} P_i * log_2 Pi \quad (2.3)$$

其中X是一个变量，p_i是变量X取值x_i的概率。对与分类系统而言，它的可能取值为c_i,每一类出现的概率为$P(c_i)$。所以分类系统的熵表示为式 2.4。信息增益是对于一个一个特征而言的，对于某一个特征T，系统没有它与有它时信息熵之差就是该特征的信息增益。在考虑存在一个特征T，系统的信息量是多少时，必须要涉及到一个概念就是“条件熵”。所谓条件熵就是指

$$H(C) = -\sum_{i=1}^{n} P(c_i) * log_2 P(c_i) \quad (2.4)$$

系统的特征T被固定时，系统具有的信息熵。对于文本分类系统的特征T，只有两种情况，就是文档中存在特征t，以及不存在特征t，不存在时用$\bar{t}$来表示。条件熵公式如 2.5。$P(t)$表示T出现的概率，$P(\bar{t})$表示不存在特征T的概率。

$$H(C|T) = P(t)H(C|t) + P(\bar{t})H(C|\bar{t}) \quad (2.5)$$

将 2.5 式进一步展开，如 2.6。

$$H(C|t) = -\sum_{i=1}^{n} P(c_i|t) * log_2 P(c_i|t)$$
$$H(C|\bar{t}) = -\sum_{i=1}^{n} P(c_i|\bar{t}) * log_2 P(c_i|\bar{t}) \quad (2.6)$$

因此特征T给系统带来的信息增益就可以写成系统原本的熵与固定特征T后的条件熵之差，如式 2.7。熵表示系统的混乱程度[5]，因此系统引入一个特征后信息增益越大也就表示系统变得更有序，说明该特征更有利于分类。

$$IG(T) = H(C) - H(C|T) \quad (2.7)$$

2.3 文本向量化

为了让文本表达为计算机能够理解的形式需要将文本向量化。文本向量化的方式目前可分为两种离散化表示方法和分布式表示方法。

2.3.1 词袋模型

这是一种离散化方法。词袋模型是一个简化了表达模型，在该模型下，一个文本可以用装着这些词的袋子来表示，这种方式不考虑文法以及词的顺序。在数学上，任何一个文本可以用一个固定长度向量来表示。向量的一个元素就表示一个特征t，元素的值为整数，一般等于这个特征t在文本中出现的次数。有些时候也可以只使用 0,1 来表示。即文本出现这个特征t对应位置的元素就记为 1，不出现就记为 0。

词袋模型是非常有效的，但其也存在数个缺点。首先它失去了词序的信息，比如“杰克喜欢玛瑞”与“玛瑞喜欢杰克”的词向量是一样的。虽然可以通过 n-grams 模型去解决这一缺点，但与此同时会带来维度灾难以及使向量矩阵特别稀疏。其次，这个模型不能学习单词下潜在的语义。

对于这个模型，最常见的用法是结合$tf - idf$算法为每一个特征加权处理，这样在往往能更合理的表示一个文本[11]。同时结合信息增益等方法选取特征可以有效避免维度灾难。

2.3.2Doc2vec

这是一种分布式表示方法。Doc2vec 是一种段落向量化的非监督算法，它的提出是建立在 word2vec 基础上的。该算法能够从边长的文本(句子、段落和文档)中学习到固定长度的特征向量。这个向量能够从段落中给定的上下文样本去预测下一个词，原论文中提到两种方法分别是:PV-DM 与 PV-DBOW。我们在使用时仅用到了 PV-DM 方法，PV-DM 方法与 word2vec 中 CBOW 很像[12]。

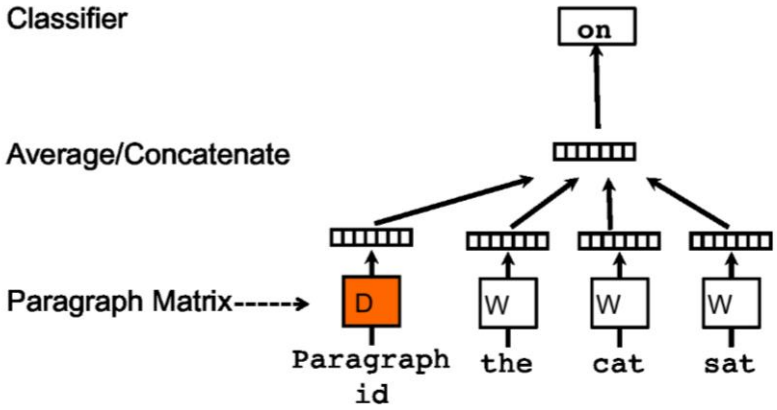


图 2　PV-DM 学习段落向量结构图

在 Doc2vec 中，每一句话用唯一的向量来表示，用矩阵D的某一列来代表。每个词也用唯一的向量来表示，用矩阵W的某一列来表示。每次从一句话中滑动采样固定长度的词，取其中一个词作为预测词，其他词作为输出词。输入词对应的词向量 Word vector 与本句话对

应的句子向量 Paragraph vector 作为输入层的输入，将本句话的向量和本次采样的词向量相加求平均或者累加构成一个新的向量x，进而使用这个向量x预测此次窗口内的预测词。

Paragraph vector 在同一个句子的若干次训练中是共享的，它可以被看做是句子的主旨。随着一句话每次滑动取若干词训练的过程中，作为每次训练的输入层一部分共享的 Paragraph vector，该向量表达的主旨将会越来越准确。

2.4 分类器

在得到文本的特征向量后，便可以将它用于传统的机器学习以及深度学习。第一问是文本分类问题，我们主要用到的分类器有朴素贝叶斯，SVM 以及 MLP。个别分类器是用来做对比的，我们只介绍主要的分类器。

2.4.1 多项式分布朴素贝叶斯

这是一种传统的分类器。理论很简单，但却很有效。朴素贝叶斯是一种有监督学习算法，它主要的理论基础是贝叶斯定理。“朴素”是一种假设：对于给定的任意两个特征间都是独立的，不需要考虑联合概率密度。贝叶斯理论陈述了式 2.8 的关系。其中y表示不同的文本类型，$(x_1, \dots, x_n)$表示文本特征向量，x_i为不同特征。

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \tag{2.8}$$

使用朴素贝叶斯假设，2.8 式将会变为 2.9 式，

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)} \tag{2.9}$$

于是我们使用的分类规则如式 2.10。我们可以使用最大后验概率去估计$P(y)$和$P(x_i|y)$;

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{2.10}$$

$$\Downarrow$$

$$\hat{y} = argmax P(y)\prod_{i=1}^{n} P(x_i|y)$$

对于不同的朴素贝叶斯分类器，它们的区别主要是$P(x_i|y)$不同的分布假设。我们使用的是多项式分布的朴素贝叶斯，对于每一个类别y其分布被向量$\theta_y = (\theta_{y1}, \dots, \theta_{yn})$参数化。$n$是特征的数量，$\theta_{yi}$就是$P(x_i|y)$，表示特征$x_i$出现在类别$y$的概率。参数$\theta_y$通过平滑的极大似然估计确定，如式 2.11。$N_{yi}$是特征$x_i$出现在类别$y$中的次数，$N_y = \sum_{i=1}^{n} N_{yi}$是类别$y$中所有特征

$$\hat{\theta}_{yi} = \frac{N_{yi} + a}{N_y + an} \tag{2.11}$$

出现的总次数。引入平滑指数a，是为了防止没有出现在学习子样特征的概率为 0。

2.4.2 多层感知器

多层感知器（MLP）理论基于神经网络。在有监督的情况下，它能够通过数据集学习到一个函数：$f(\cdot): R^m \longrightarrow R^o$。$m$是输入的维数，$o$是输出的维数。如果给予这个模型一系列特征$X = x_1, x_2, \ldots, x_m$和一个目标$y$，它能够学习到一个近似非线性的函数用于分类或回归。它和逻辑回归（Logistic Regression）的不同点在于它具有一个或者多个非线性层，也就是隐藏层[6]。图 3 展示了一个只有一个隐藏层的 MLP。

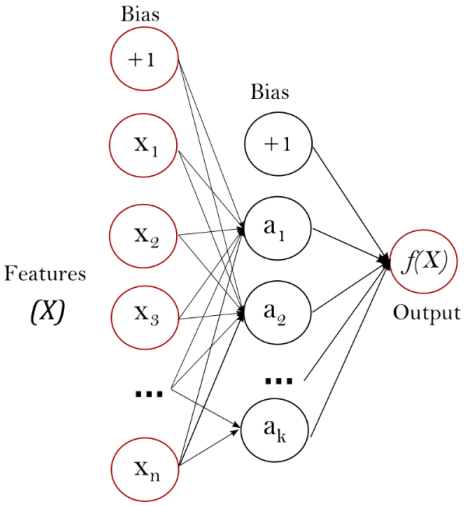


图 3　一个隐藏层的 MLP

最左边的是输入层，由一些神经元集合组成，$\{x_1, x_2, \ldots, x_m\}$表示了输入的特征。隐藏层的每个神经元都具有一个转换作用。对于一个隐藏层的 MLP，其学习函数可以表示为式 2.12。W_2, W_1, b_1, b_2是模型的参数。W_2, W_1是输入层和隐藏层的权重，b_1, b_2是加在隐藏层和输出层的偏置。$g(\cdot): R \longrightarrow R$是激活函数，我们在训练的时候选择的是双曲正切激励函数,即 2.13。

$$f(x) = W_2 g(W_1^T x + b_1) + b_2, \tag{2.12}$$

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.13}$$

对于多分类，$f(x)$将会是一个n维的向量，n是分类的类型数量。此时$f(x)$使用 softmax 函数，见式 2.14。z_i表示输入 softmax 函数的第i个元素，这和第i类是对应的。k是总共的类别数。结果是一个表示样本x输入某一类的概率的向量，且向量各元素和为 1。

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_{l=k}^{k} \exp(z_l)} \tag{2.14}$$

MLP 使用的损失函数有几种。对于分类问题损失函数是交叉熵(二分类情况见式 2.15)，我们选择的正则化方法是 L2(aka 惩罚函数)，也就是$a\|W\|_2^2$。其中a是控制惩罚程度的超参数。

$$Loss(\hat{y}, y, W) = -ylny - (1-y)\ln(1-\hat{y}) + a\|W\|_2^2 \quad 2.15$$

要训练出 MLP 的各个参数(w, b)，我们采用的是 Adam 算法。Adam 是一种可以替代传统随机梯度下降过程的一阶优化算法，它能基于训练数据迭代地更新神经网络权重。随机梯度下降算法保持单一的学习率 (即 alpha) 更新所有的权重，学习率在训练过程中不会改变。而 Adam 通过计算梯度的一阶矩估计和二阶矩估计而为不同的参数设计独立的自适应性学习率。

三、群众问政留言聚类

问题二要求实现对热点问题的挖掘，即要实现对反映同样问题的留言聚类。通过对文本的分析，我们发现留言具有以下两个特点：

1. 留言的核心内容都集中在留言主题，也就是留言主题即能代表整段留言。只需要对留言主题实现聚类即可。
2. 问政平台的留言具有特殊性，留言主题中存在大量机构名和地名。[15]

基于以上这两个特点，我们通过 DBSCAN 聚类算法来实现对留言文本的聚类。

3.1DBSCAN 算法

基于密度的聚类算法是数据挖掘中常用的算法，DBSCAN 算法认密度高的区域可以聚为一类，而密度低的则不能。其核心思想是用一个点的ε邻域内的邻居点数衡量改点所在的密度空间。基于这样一个一般的思想，它可以找出形状不规则的聚类。而且我们不需要事先知道聚类的个数。DBSCAN 算法有两个重要的参数：Eps和$MinPts$，前者为定义密度时的领域半径，后者为定义核心点时的阀值，也就是称为一个 cluster 所需要的最小的点数。为了实现 DBSCAN 聚类算法，待聚类的点被定义为三种类型：核心点、边界点、噪音点。要成为核心点必须要保证在规定的邻域$r = Eps$内，存在至少$MinPts$个点(包括自己)。下面给出文献中直接密度可达与密度可达的定义(这里尽量少采用数学语言)。[7]

定义 1 直接密度：可达如果x, y都是样本点，且x是核心点，y在x的邻域内，那就称y是从x的直接密度可达的。

定义 2 密度可达：对于$p^1, p^2, \dots, p^m$都是样本点，其中$m \geq 2$。如果p^{i+1}是从p^i的直接密度可达的，$i = 1,2, \dots, m-1$，则称p^m是从p^1密度可达的。

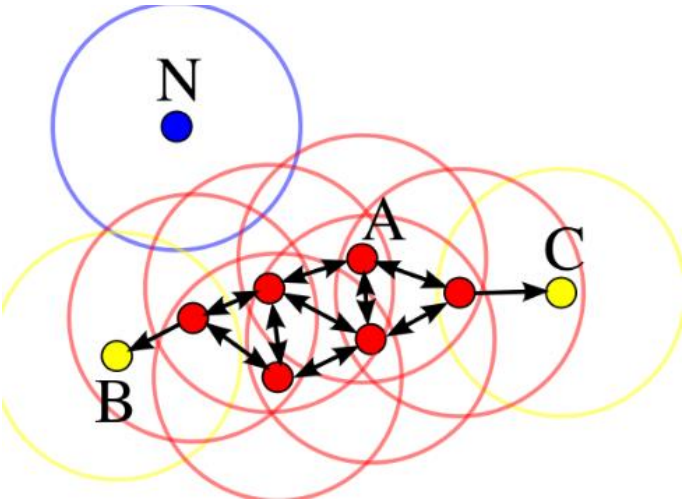


图 4　DBSCAN 聚类($MinPts = 4$)

在图 4 里面，设置$MinPts = 4$，也就是至少要 4 个点才能聚为一类。点 A 和其它红色的点都是核心点，因为在这些点以ε为半径的区域内都至少存在 4 个点(包括这个点本身)。这些红点都是可以相互达到的，它们形成了一个聚类。而点 B 和点 C 不是核心点，但是它们都能过通过 A 密度可达(或者与 A 密度可达的其它核心点)，这类点叫边界点，它们也属于这个聚类。点 N 不能够由任何一个核心点直接密度可达，它是噪音点，不属于任何一个聚类。注意这个例子里只存在一个聚类，边缘点往往位于一个或者几个族的边缘地带，可能属于一个族，也可能属于另外一个族，其族的归属不明确。

在该算法中对于特征向量间距离计算我们采用的是余弦相似度。如果x, y是行向量，那么它们的余弦相似度k被定义为式 3.1。很明显，如果两个向量的每一个元素越接近，它们在空间的夹角就越小，其余弦值k越接近于 1，那么两个文档就越相似，反之亦然。

$$k(x,y) = \frac{xy^T}{\|x\|\|y\|} \tag{3.1}$$

我们决定使用这个算法主要是它具有以下几个优点：1.DBSCAN 不需要固定聚类的族数，这恰好满足了对留言的聚类，聚类族数不是先验的。这和 k-means 算法是不同的；2.它能够识别任意形状的聚类；3.这个算法能够排除出噪音点，对于异常点是十分鲁邦的。在对热点聚类中就存在大量的噪音点，也就是某个留言问题只出现过一次的情况很多。

3.2 基于 DBSCAN 的聚类器

为了实现对文本聚类，依据留言的特征，我们想出了一套以 DBSCAN 聚类算法为基础专门用于问政平台热点问题挖掘的聚类方法。这个方法带有一点贪心的思想在里面。

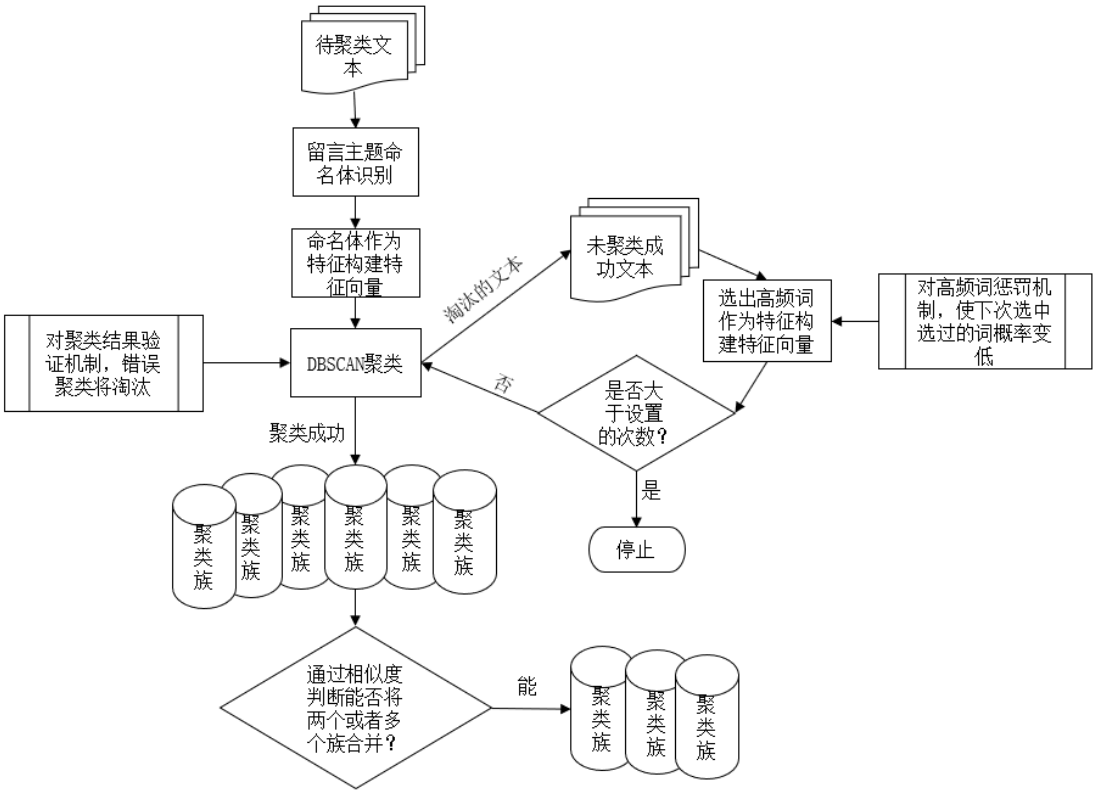


图 5　基于 DBSCAN 聚类流程图

图 5 是该方案的流程图。第一步是使用命名体作为特征词进行聚类。对留言主题进行命名体识别，得到的机构名以及地名作为特征词用来构建词向量。关于词向量的构建这里我们采用二进制的形式，也即是出现该特征记为 1，未出现记为 0。我们将留言主题和留言详情中的文本合并了去构建相应的文本向量，而不是仅仅只看某一特征是否出现在留言主题里。接下来设置 DBSCAN 参数得到聚类。

DBSCAN 聚类结果并非都是同种热点问题，我们还需要对这个结果中的每一个聚类结果进行验证。验证方式依然使用 DBSCAN 算法，只不过需要重新提取特征，这时特征提取的方法是取部分高频词作为特征词用于特征向量的构建。这时高频词的提取是基于留言详情的。通过二次聚类以后将聚类成功的留言送入聚类成功的集合，淘汰掉的留言送入未聚类的集合。

第二步就是选高频词聚类。重要的事依然是特征选取，这时候的特征我们选择的是留言主题中出现的高频词，我们考虑到高频词往往能说明一些文本的共性，更有可能反映同一类问题。同时我们设置一个高频词惩罚机制，降低选中的高频词下次选中的概率。通过次数的设置，对未聚类的文本反复使用 DBSCAN 算法聚类。每次聚类后结果也要执行上面的验证机制。

最后一步就是聚类结果的合并。留言经过上面的聚类加验证连续两次 DBSCAN 算法操作后会出现一个问题，那就是同一类的文本可能会被聚成几类。出现这个问题原因有两个，一是我们设置的$MinPts$参数不是很大，应该说不能设置的很大，设置大了该模型发现热点问题的概率将降低，也即是只能将一小部分热点问题发现，大部分都会被认为噪音文本，或者聚成一大类。二是我们的Eps参数设置也很小，这是为了增强其识别热点的能力。举个例子，如果特征中有“南京”和“新街口”这两个个词，如果Eps太大，那么识别到的结果是将会是

所有南京发生的事件聚为一类，而Eps设置的足够小出现“新街口”这一词的文本就会单独聚为一类。而Eps设置得太小，本来属于“新街口”发生的同一类热点问题又会被聚为不同的几类。我们宁愿希望出现第二种情况也不出现第一种，因为第二种情况虽然会将同一类分为了两类，但是其至少识别出了热点。对于第二种情况Eps太小聚类后，我们不得不在进行热点的合并即把原本属于同一类得热点合并到一起。执行这部操作所用的算法是使用余弦公式 3.1 计算两个文本的相似度。

通过设置适当的相识度阀值，可将两个文档归为一类。计算的方法是选取同一个聚类族内出现的高频词，构建特征向量，再利用 3.1 式求出相似度。

3.3 热度指标计算方法

热点的定义如下，热度指标是对其的量化。为了更加准确的发现热点话题，一个合理的热点公式是十分重要的。我们现在能够使用建立热度指标的信息有：聚类好的每一类热点问题的留言数量、留言时间、点赞量、反对量。留言数量和点赞量明显与热度是正相关的；留言时间越集中，热度值也越高；反对量与热度负相关。基于这些关系，我们给出计算热度值得公式如式 3.2。

> 定义三 热点：指在过去或者当前的某一时间段内，被比较多的人关注或集中关注的信息点。

$$score = \frac{1}{\ln(std_t + 10)} * \sum_{i=1}^{n} a * \frac{\ln(sup_i + 1) + 1}{\ln(\mathrm{opp_i} + 1) + 1} \tag{3.2}$$

3.2 公式中，n表示同一个聚类族内留言数量。我们认为每一篇留言都会为留言所代表得主题提供一个热度值，不妨把它叫做热度因子。那么每一篇留言的热度因子计算公式就是$a * (\ln(sup_i + 1) + 1)/(\ln(opp_i + 1) + 1)$,$a$是一个自定义常数，可以代表每篇留言的初始热度。$sup_i$是留言的点赞量，$opp_i$是留言的反对量，取对数的目的是为了防止某一篇留言的热度因子产生爆炸，因为观察发现有的留言中点赞量达到几百，这相当与是对高点赞、高反对一种抑制作用。

公式前面的系数$1/\ln(std_t + 10)$,std_t表示同一族聚类里面留言时间的标准差。它的计方法是将留言时间两两相减，然后求得到数组的标准差。

四、答复意见质量评价方案

4.1 评价标准权重分配

针对相关部门对留言的答复意见，从答复的相关性、可解释性、完整性、及时性、言辞友好性对答复意见做出评价。使用层次分析法对各因素的重要程度进行量化。

表 1

标度	含义
1	同样重要
3	稍微重要
5	明显重要
7	非常重要
9	极其重要
2, 4, 6, 8	上述两相邻判断的中值
倒数	因素 i 与 j 比较的判断为 k，则因素 j 与 i 的判断为 1/k

表 1　判断矩阵元素的标度方法

	相关性	完整性	可解释性	及时性	言辞友好性
相关性	1	1	2	4	8
完整性	1	1	2	4	8
可解释性	1/2	1/2	1	2	4
及时性	1/4	1/4	1/2	1	2
言辞友好性	1/8	1/8	1/4	1/2	1

评价因素	相关性	完整性	可解释性	及时性	言辞友好性
权重	0.35	0.35	0.17	0.09	0.04

最大特征根 $\lambda = 5.008$，一致性指标 $CI = 0.002$;

一致性比率 $CR = \frac{a_1 CI_1 + a_2 CI_2 + \cdots + a_m CI_m}{a_1 RI_1 + a_2 RI_2 + \cdots + a_m RI_m} < 0.1$，因此满足一致性检验。

作为留言答复，首先需契合留言主题，与用户留言内容高度相关，同时答复内容应完整详实，能够切实解决留言用户的疑惑，其次语义表达准确，具有一定可解释性；相关部门应尽可能及时回复，且回复内容措辞友好，对待用户礼貌尊重。

根据各项评价标准独立评价答复意见后，按权重对答复做出综合评价。答复评价如下:

$$S = 0.35\alpha_1 + 0.35\alpha_2 + 0.17\alpha_3 + 0.09\alpha_4 + 0.04\alpha_5$$

其中，$\alpha_i (i = 1,2,\ldots,5)$依次为上述 5 个因素自身评分值，S取值范围[0,100]。

4.2 相关性评价

对相关部门答复意见进行分词处理并基于 TF-IDF 算法计算词的权重，根据选择的特征项后，以特征向量表示提取的文本，通过布尔模型计算答复意见与留言主题的相关性。

布尔模型是基于集合论和布尔代数的一种简单检索模型，其形式清楚简单，易于实现。基于布尔模型，通过计算答复意见特征项集合与留言主题特征项集合的交集判断答复意见与留言主题的相关性，具体公式如下：

$$sim(D,T)=\frac{|D\cap T|}{|T|},\qquad \alpha_1=\frac{sim(D,T)}{sim(D,T)_{max}}\times 35$$

其中，D为答复意见特征项集合，T为留言主题特征项集合，$sim(D,T)$为答复意见与留言主题的相关度，答复意见与留言主题的相关性与交集$D\cap T$中元素个数正相关；规定该项满分 35 分。

4.3 其他评价标准

留言答复的完整性可通过句长和字数体现，通常认为答复意见文本越长，所含信息量越大，文本价值越高；同时亦考虑留言本身所言问题是否复杂。相对来说，字数更多、更详尽的答复能够使用户满意度更高，因此选取答复字符数与留言字符数比值γ对留言答复的完整性进行量化：$\alpha_3=\frac{\gamma_i}{\gamma_{max}}\times 17$，分数规范至[0,17]。

留言答复的可解释性指文本可理解的程度，其合理性，通过 $\beta_i=\frac{|D\cap T|+|D|}{m}$ 实现量化，$D\cap T$为留言与答复特征项集合的交集，m为答复的长度即字符数；答复一方面要切合问题，一方面语义无问题，使用可检测的词数体现；分数规范至[0,35]。

答复的及时性可通过答复时间与留言时间的时间差来表示，及时性与时间差成反比，即留言与答复间隔越久，答复的及时性越低。分数规范至[0,9]。

答复的语气态度是否文明友好常体现在“您”“你好”“感谢”“请”等敬语、文明用语的使用上，根据文本中敬语的数量对答复的措辞友好性进行量化评价，敬语数量越多，答复的语气措辞越友好；并使用各条回复中敬词数与测试集中单个回复中敬词数最大值的比值进行量化，分数规范至[0,4]。

五、实验结果分析

5.1 留言文本分类实验

5.1.1 特征数量的确定

我们使用 python 的 scikit-learn 机器学习框架来训练各种分类器以及 DBSCAN 聚类器；使用 gensim 来实现 Doc2vec。接下来是使用第二部分提到的理论进行实验。附件二中提供的数据如图 6，可见交通运输这一类的数据量是比较少的。一共有 9210 条数据，首先对留言详情进行命名体识别，分词，去停用词。我们以随机抽样的方式选择 75%的数据集作为训练集，一共 7368 条留言。剩余的作为测试集。

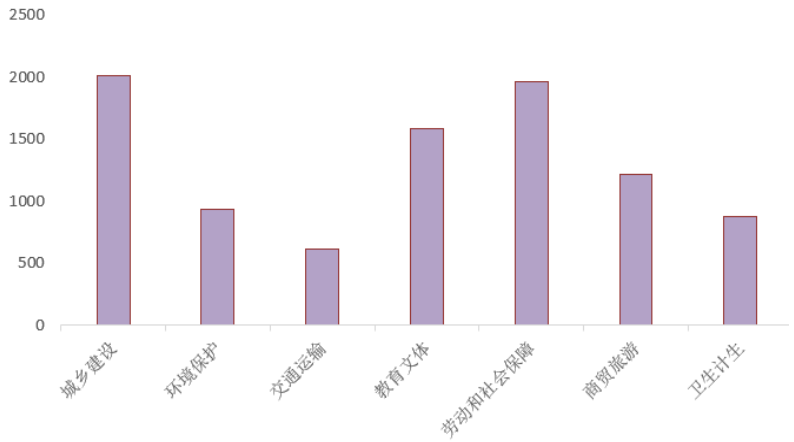


图 6　留言一级标签详情

在建立词袋模型的时候，最重要的就是特征提取[19]。我们采用信息熵和$tf-idf$结合的方法提取。那么提取的特征数量多少合适呢？为了先找到最佳的特征提取数量，即要保证训练集不能过拟合和测试集要有足够高的 f1-score。我们先采用伯努利朴素贝叶斯模型探究一下最佳特征数。虽然不同的模型最佳值可能会不同，但都在一个相应的不会差距太大的范围内。

我们对 75%的测试集分词处理后，去除词频数低于 10 的低频词后，得到作为候选的特征词一共 44965 个。接下来分别求出候选特征词的信息增益，选取信息增益大的特征词用于建模。图 7 显示了随着选取信息增益靠前的词数变化训练模型 5 折交叉验证平均 f1-score 与测试集 f1-score 的变化的情况。

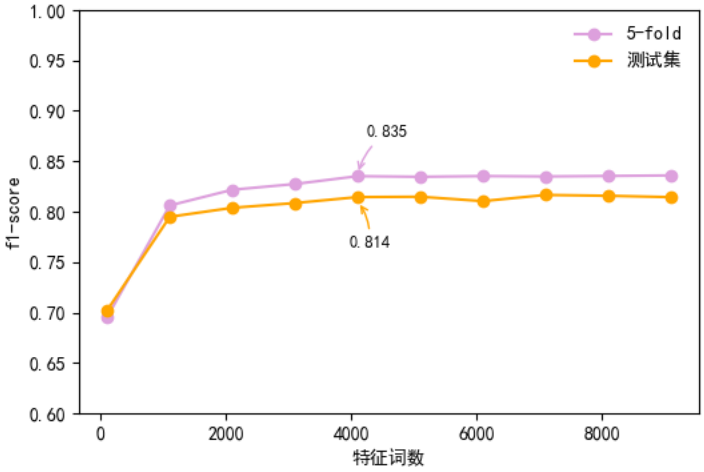


图 7　五折交叉验证平均 f1-score 与测试集 f1-score 随特征词数的变数

从图 7 中可以看出随着特征词数的增加，当达词数到 4000 的时候，测试集 f1-score 基本不再变化。此时交叉验证 f1-score 平均值为 0.835，测试集 f1-score 的值为 0.814。

通过上面的分析，我们将特征词数定在 4000 左右。上面所使用的模型是伯努利分布的朴素贝叶斯模型，是十分简单的模型，都没有考虑到特征的权重，只考虑了留言文本中有没有出现某一个特征。但是朴素的·模型训练结果还是不错的达到了 0.814。我们接下来是通过对特征加权，以及继续优化特征的选择来提高测试集的 f1-score.

5.1.2 优化特征

利用 tf-idf 选取得分高的前 4000 个特征与上面的 4000 个特征取并集后，还剩下 1811 个词。部分词及权重如表 2。

特征词	权重
医院	0.2328
医生	0.2882
生育	0.1580
患者	0.1350
…	…

表 2 卫生计生的部分特征词及权重

通过取并集维数下降了一倍，之后我们再利用 tf-idf 加权特征矩阵(这时我们构造初始矩阵采用的是词频计数即 Count，而不是二进制形式 Occurrence)。

我们利用多项式分布的朴素贝叶斯训练模型，并设置平滑系数$a = 1.0e - 8$。得到结果见表 2。可看到训练结果的总体 f1-score 为 0.848，较上面的基础模型有了显著提高。表中显示了对环境保护的分类结果是很好的，f1-score 达到了 0.914；其它标签的分类情况也还不错，都在 0.8 以上；特别值得关注的就是交通运输这一类，它的召回率只有 0.640。我尝试添加新的特征来增加交通运输这一类的召回率。如表 3 使我们添加 2000 个交通运输这类留言的低频词后分类情况。

	Label	Precision	Recall	F1	Support
0	城乡建设	0.772321	0.875949	0.820878	395
1	环境保护	0.900000	0.929348	0.914439	184
2	交通运输	0.825581	0.639640	0.720812	111
3	教育文体	0.892086	0.879433	0.885714	282
4	劳动和社会保障	0.868360	0.921569	0.894174	408
5	商贸旅游	0.879070	0.744094	0.805970	254
6	卫生计生	0.921875	0.850962	0.885000	208
nan	总体	0.865613	0.834428	0.846712	1842

表 3 多项式分布的朴素贝叶斯训练结果

显然结果不是很理想，交通运输这类的召回率提高了，但是其精确度下降了很多。不但如此，其它类的召回率和精确度都有所下降，这是一种得不偿失的做法。所以我们选择之前的 1811 个特征词进行后续试验。交通运输这类召回率相对比较低的原因可能与这一类的样本个数有关，在所有样本中，交通运输只占了 1/15，这虽然不是极不平衡的样本，但会对结果存在一定影响。

	Label	Precision	Recall	F1	Support
0	城乡建设	0.770142	0.822785	0.795594	395
1	环境保护	0.899471	0.923913	0.911528	184
2	交通运输	0.722222	0.702703	0.712329	111
3	教育文体	0.846690	0.861702	0.854130	282
4	劳动和社会保障	0.834842	0.904412	0.868235	408
5	商贸旅游	0.844340	0.704724	0.768240	254
6	卫生计生	0.917582	0.802885	0.856410	208
nan	总体	0.833613	0.817589	0.823781	1842

表 4 添加低频词后分类情况

现在我们把特征固定在了 1811 维。接下来我们尝试其他的分类器模型看看结果能不能得到一定的优化。

5.1.3 更多的分类器

5.1.2 中，我们只使用了多项式分布的朴素贝叶斯以及伯努利分布的朴素贝叶斯。结果都较为理想。下面我们尝试使用更多的分类器，比较它们应用在这 1811 个特征上的优劣。

我们采用如图 8 的方式训练模型。首先将留言文本成两部分：75%的文本用于训练模型，作为训练集;25%模型用于测试模型,作为测试集。首先要初始化模型参数，然后在训练模型时采用 5 折交叉验证，得到模型后用于测试集。测试的结果需要与交叉验证的结果比较，如果二者的结果相差太远，就说明模型泛华能力差，出现了过拟合或者欠拟合。根据比较结果调整模型参数。如此下去，得到最优的模型参数。

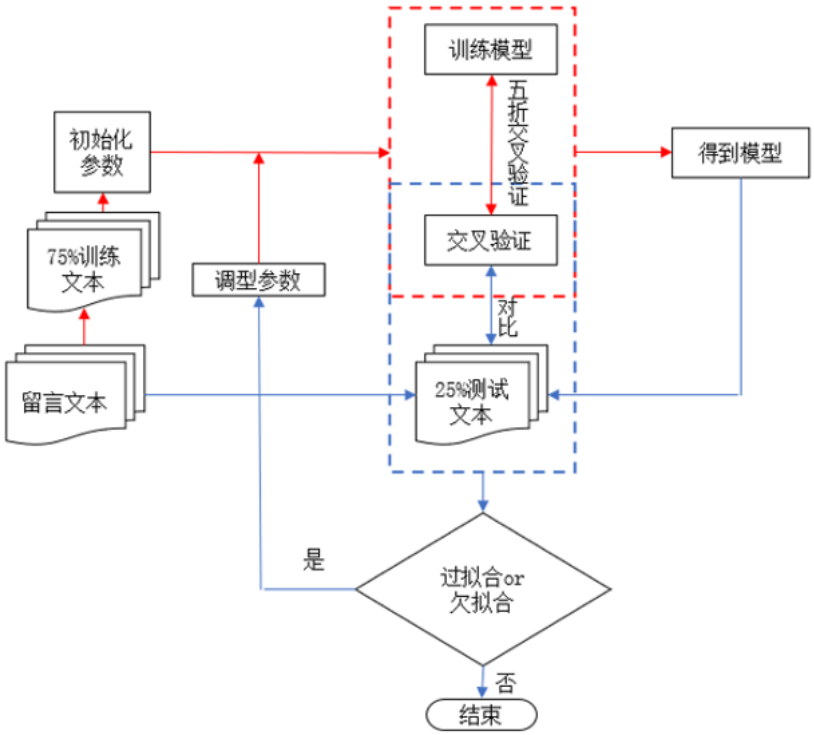


图 8. 分类器训练流程图

图 9 是将同样的特征矩阵用于 4 中不同的分类器训练的结果，左边的一幅图是各个分类器对测试集训练结果的 f1-score 得分，右边的两幅图分别是相应的 Recall 与 Precision。在训练调整参数的过程中为了防止过拟合，我们使用了交叉验证。交叉验证的情况如图 10。对比图 9 和图 10 可知结果是可靠地，能很好的应用于测试集。从图 9 中可以看出各种分类器训练的结果差距不是很大，他们对各类的分类情况也基本一致(线的起伏基本是一致的)。‘’总体‘’这一个标签代表得 f1-score 是所有分类标签 f1-score 的平均值，橙色的线，即支持向量机相对是最好的分类器，它的总体 f1-score 值由表 5 可知是 0.859。而随机森林模型最不理想，是 0.817。

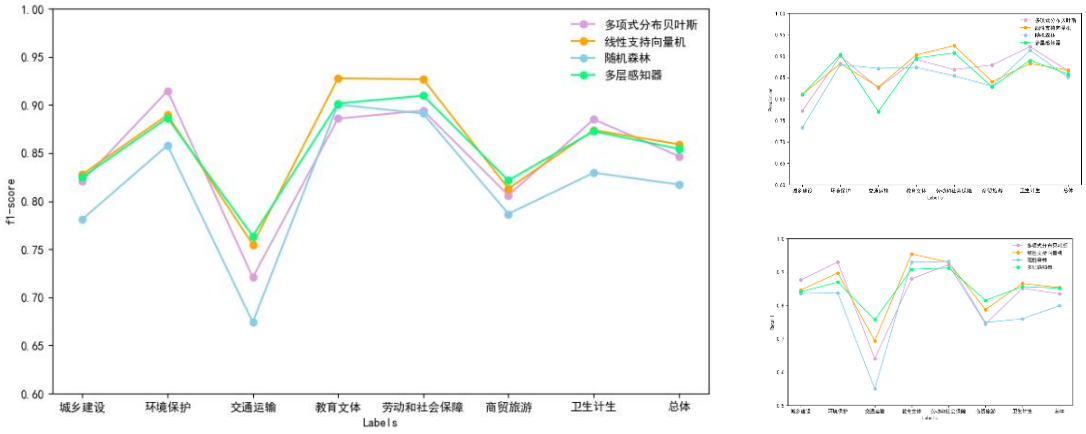


图 9.　4 种分类器用于测试集结果对比(词袋模型)

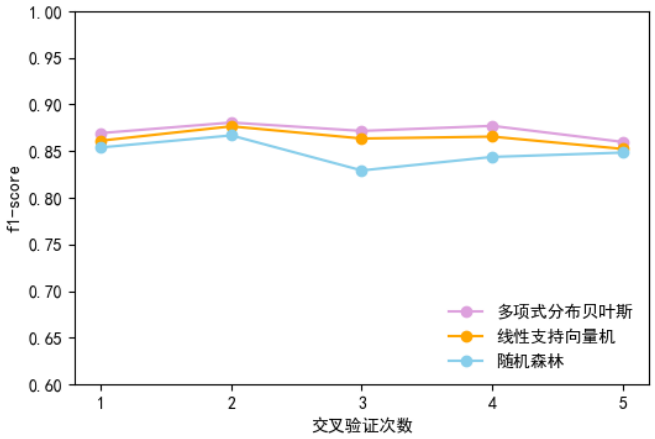


图 10.　3 种分类器交叉验证情况(词袋模型)

从图 9 和表 5 中也可以看出各个标签分类和好坏程度相差很大。分类最好的三类是环境保护，教育文体，劳动和社会保障，它们的 f1-score 得分基本在 0.9 左右，它们的召回率和精确度也是最高的。分类最差的是交通运输这一类，其 f1-score 得分最高只有 0.764，它的精确度与其他类差距不是很差，但是召回率特别低。这是以后改进模型的一个主要的方向。

	城乡建设	环境保护	交通运输	教育文体	劳动和社会保障	商贸旅游	卫生计生	总体
MultinomialNB	0.820	0.914	0.720	0.886	0.894	0.806	0.885	0.847
LinearSVC	0.828	0.889	0.754	0.928	0.927	0.813	0.874	0.859
RandomForest	0.781	0.857	0.674	0.900	0.891	0.787	0.829	0.817
MLP	0.825	0.886	0.764	0.910	0.910	0.821	0.876	0.854

表 5　4 种分类器用于测试集结果 f1-score(词袋模型)

表 5 中前三种分类器属于机器学习后一种属于深度学习。此外，虽然多项式朴素贝叶斯理论很简单，而且计算量也是最少的，训练时间也是最快的，但是它的结果却并不最次。

就实验结果来看，线性 SVM 是分类效果最好的模型。其参数设置为：惩罚函数为‘‘L2‘‘，损失函数为 SVM 的标准损失函数额平方即：“squared_hinge”，“dual”参数设置为 False，表示选择解决原始最优化问题的算法，不选择解决对偶问题的算法。这个在样本数大于特征数的时候更好。宽容度 tol 设置为 1e-3，正则化系数 C 为 1，最大迭代次数为 1000。

MLP 训练结果对交通运输分类的效果最好，这是深度学习算法的优势。设置参数如下：隐藏层为 200 层，激励函数为“relu”，即$f(x) = \max(0, max)$。解决权重最优化的算法为“adam”.初始学习率是 0.001，最大迭代次数为 200 次，惩罚函数“L2”的惩罚系数 alpha 为 0.0001，宽容度为 1e-5，epoch 设置为 10。

接下来我们是用 Doc2vec 的方法构建特征向量，用于各种分类器的训练。

我们首先将留言详情作为训练 Doc2vec 模型的语料库。在训练的时候训练集和测试集都留言详情都将被用于训练 Doc2vec 模型，而不是只取训练集的留言详情。这样在每一次预测的时候 Doc2vec 模型必须要用待预测的留言详情作为语料库更新该模型，相应的分类器的参数也需要随着 Doc2vec 产生的段落向量不同重新训练。当然，如果语料库足够大，也可以建立一个不在线学习的 Doc2vec 模型，直接使用它推测待分类留言的文本向量。我们尝试了使用 75%的数据作为语料库去训练 Doc2vec 模型，通过该模型推测预测留言特征向量，我们设置的是 1000 维，然后使用上面设置的线性 SVM 去分类，f1-score 只有 0.77，是很不理想的。我们认为剩余 25%待预测的文本应该是先验的，就是在预测分类时我们必须要知道用于预测的留言文本才行，所以可以将这部分信息也利用起来用于训练 Doc2vec 模型。

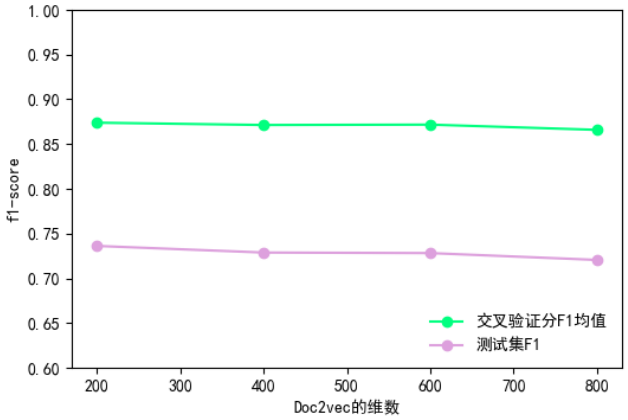


图 11　Doc2vec 模型 vector_size 与结果和检查验证 F1 值关系

由上图易知，Doc2vec 模型存在过拟合，下面我们进行探究来确定模型最佳的参数。我们探究的方法采用控制变量，当然这种方法可能并不能实现总体最优，但在一定程度上能放映实现模型参数最优化的方向。

首先模型参数初始化，我们设置初始参数如下：

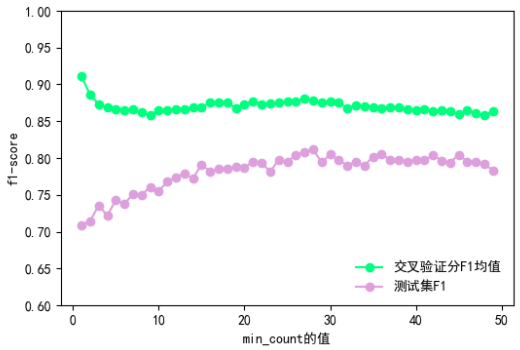
参数	dm	vector_size	window	min_count	negative	epochs
值	1	Variable	5	3	10	10

其中 vector_size 作为变量。随着 vector_size 的变化，训练集五折交叉验证 F1 的均值与测试集 F1 值得变化情况如图 1。很明显，维数对过拟合的影响并不是很大。我们不妨就将其设置为 1000。

接下来探究 min_count，对结果的影响。设置初始化参数如下：

参数	dm	vector_size	window	min_count	negative	epochs
值	1	1000	5	Variable	10	10

由图 12 可以看出，随着 min_count 的增大，也就是模型中不考虑低词频词，模型的欠拟合程度有所减少。当 min_count 为 30 左右时，过拟合程度已经达到最小。我们不妨将 min_count 的值取为 30,。此时测试集 F1 值为 0.807，交叉验证的 F1 值为 0.875。



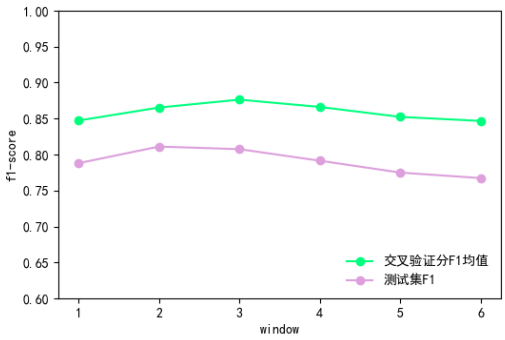


图 12 （左） Doc2vec 模型 min_count 与结果和检查验证 F1 值关系

图 13 （右） Doc2vec 模型 window 与结果和检查验证 F1 值关系

接下来探究 window 参数对过拟合程度的影响，我们将 min_count 参数设置为上面探究到的最优值。得到结果如图 13，可知 window 取 3 时过拟合程度最小，此时测试集 F1 值为 0.817，交叉验证的 F1 值为 0.865。我们将 window 设置为 2，继续调参的实验。

最后运行可知参数 negative 对过拟合程度的影响不是很大。我们将其设置为 10。

通过一步步设置最优值，类似于贪心算法。这样得到的结果不一定是最优的，但是在一定程度上定性地说明了这些参数对模型欠拟合程度的影响。最终我们确定参数为：

参数	dm	vector_size	window	min_count	negative	epochs
值	1	1000	2	30	10	10

我们使用 PV-DM 算法。Vector_size 是我们希望文本的特特征向量的维数；Window 表示一个句子中当前的词语与预测的词语最大的距离；min_count=30 表示忽略整个语料库词频比 30 低的词语；negative 表示执行负采样；epochs 表示将整个语料库重复训练 10 次。这些参数是我们手动调试的，只是我们调试过程中最优的。

用上面这些参数训练得到的 1000 维特征向量训练分类器模型结果如图 14。从实验结果来看，四种分类器的总体的 f1-score 为 0.805 左右，而交叉验证的平均 f1-score 基本为 0.86。模型任然存在轻微的过拟合，但已经有了不少改善。然而总体来说，这个模型在留言文本分类方面还是不及词袋模型。

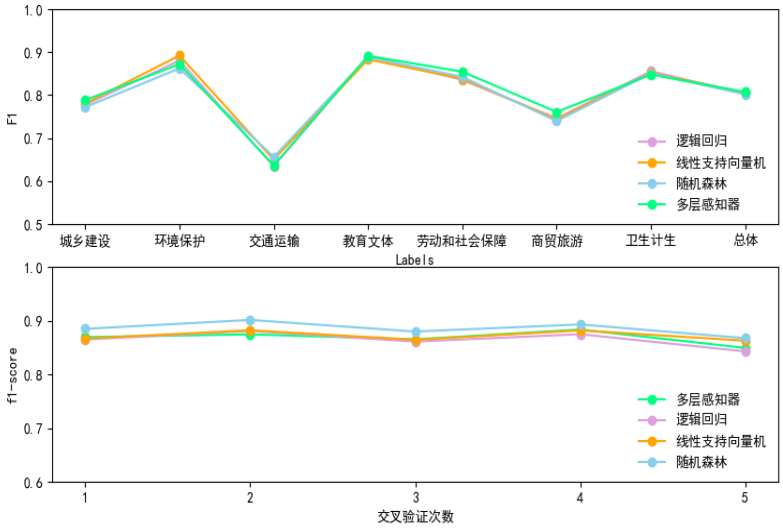


图 14　Doc2vec 模型分类情况

下图是使用 Doc2vec 模型生成的文本向量用于不同的分类器进行测试得到的分类的结果。这里使用逻辑回归替换了朴素贝叶斯，因为多项式分布的朴素贝叶斯用于分类的特征向量中不能存在负数。除了逻辑回归，所有分类器参数和上面设置的一样。图 15 是前三种分类器进行的 5 折交叉验证情况，与图 16 对比排除过拟合以及欠拟合，除了随机森林模型参数设置存在问题(交叉验证与预测结果 f1-score 值偏差较大), 其余模型训练的结果十分可靠。

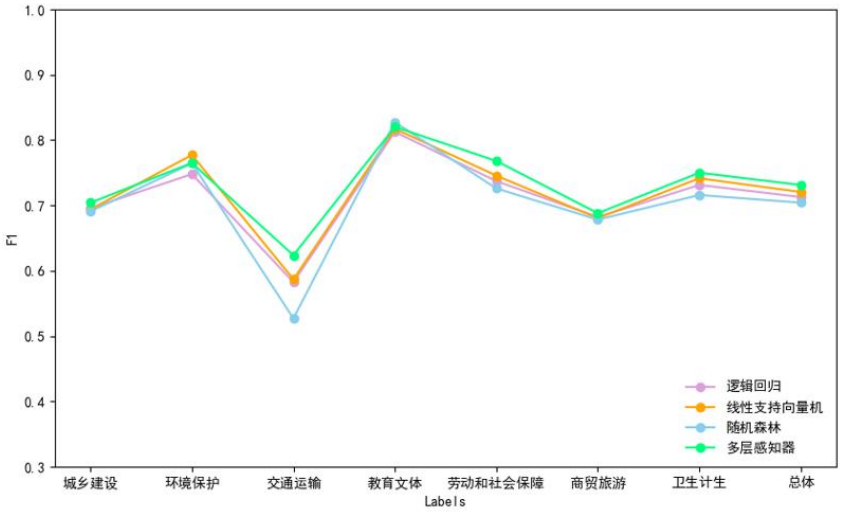


图 15　4 种分类器用于测试集结果对比——f1-score(Doc2vec)

由图 15、16 我们可以看到 Doc2vec 训练出来的特征向量的针对本数据与问题分类结果仅仅差强人意。随机森林模型存在一点欠拟合，其在交通运输上的召回率只有 0.640，但是其精确率为 1.00。我们以后可以通过降低精确率来提高其召回率来优化这个模型。其它三个回归模型分类的情况都不及之前好，其中多层感知器 MLP 的 f1-score 为 0.807，相对于传统文本分类方法(机器学习+词袋模型),现在基于深度学习(深度学习+Doc2vec)文本分类虽具有一定优越性（较大程度上降低维度，提升效率），但针对本问题，词袋模型分类结果相对更理想一点。

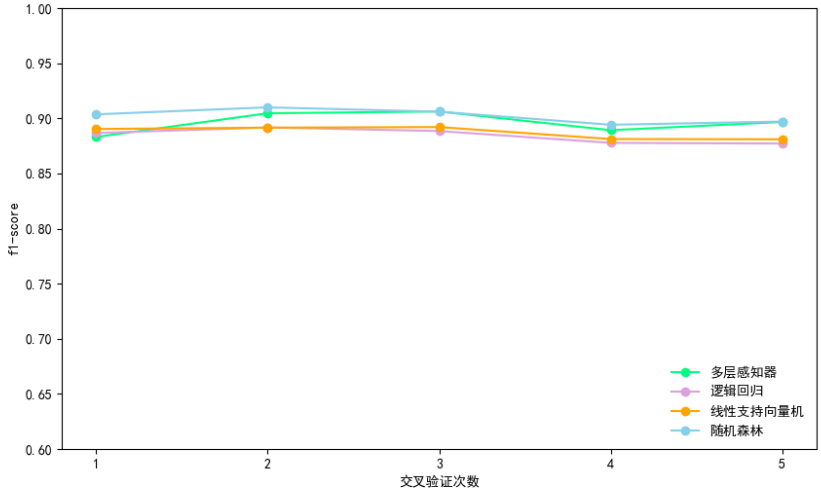


图 16　4 种分类器交叉验证情况(Doc2vec)

5.2 留言文本聚类实验

5.1 中对留言分类勉强算是成功，接下来我们利用第三部分构建的聚类器来进行实验。我们详细说一下聚类过程以及过程中参数设置。

一共有 4326 个待识别热点的样本，首先是命名体识别、分词和去停用词。通过百度 API，在主题中一共存在 1499 个命名体(我们只选取了机构名和地名)。我们对命名体进行了一些细微的处理，比如“西地省巴泥公社装饰公司”和“A3 区住建局对文兴物业公司”等等这类大地名包含小地名或者机构名，我们将他们拆开来““西地省”和“巴泥公社装饰公司”…。首先利用这些命名体聚类，设置 DBSCAN 算法的参数为 eps=0.2，metric='cosine'，min_samples=3。对于参数得调试我们通过的是人工的反馈，因为使用聚类的评估函数 CH-score 等等，其结果并不是很理想。参数设置好后，以命名体为特征进行聚类，一共得到 150 类，验证后还剩下 72 类，还剩 3967 分留言未聚类。接下来继续聚类，首先是取高频词作为特征向量，取得个数为$\sqrt{n}+3$个特征，n是未聚类文本经过验证后，使用 DBSCAN 算法聚类，这时的参数设置为 eps=0.5，metric='cosine'，min_samples=3，之后进行验证，验证使用 DBSCAN 的参数为 eps=0.5，metric='cosine'，min_samples=3。将验证合格的聚类族存储起来。这样经过两次后剩余未聚类文本维持在 3840 不变。

聚好类后进行验证，其实也就是一个推荐合并系统，首先也是特征词的提取，在对两个聚类族进行相似性验证时，分别选择两个聚类族中前 12 个高频词(主题中的高频词，不足 12 个取所有的词)一共 24 个词构建特征向量。设置余弦相似度 0.4 为合并的阀值，也就是比这个值低就合并。实验中发现该推荐系统并不能 100%推荐正确，但在 0.3 的阀值下其召回率基本唯一。

实验发现，通过这个系统除了能找到应该合并的聚类族，也能找到异常的聚类族。比如推荐系统中的某一聚类族和几个聚类族都相似，那么它也极可能是错误的聚类，因为这类文本的前 12 个高频词是没有规律的，即是这类聚类族不存在一个共同的主旨。

聚好类后便开始用热度公式计算热度。按照 2.13 式的热度定义的公式计算出的值没有固定的范围，只能反映各个类别之间相对热度的大小，计算结果如表 6。比如我现在知道了

‘’伊景园滨河苑捆绑销售车位‘’这一事件的热度值是 17.88，但是我并不能确定它是不是一个热点事件，必须要和其它事件对比才能知道结果。

为了避免上面的缺点，我们试图建立一个绝对的热度指标，该标准的热度值在 0~10 内（包括 0,10）。10 为满热度，可以代表一年中最热点的事件；0 表示没有热度。，在 2.13 的公式上做进一步修正，分析表 6 可以发现，17.88 的热度值和其它的值偏离很大，我们可以将‘’伊景园滨河苑捆绑销售车位‘’这件事的热度值定为绝对标准中的 10。而其它事件基本在 5.82 一下，我们将热度值为 5 的事件定为绝对标准中的 7。这样就建立了一个绝对标准。

热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
1	12	17.88	2019-07-07至2019-09-01	伊景园滨河苑/	关于伊景园滨河苑捆绑销售车位的投诉
2	0	9.62	2019-01-08至2019-09-09		请加快A市国家中心建设刻不容缓
3	10	5.82	2019-04-17至2019-09-06	开元路/	反映A市地铁3号线松雅西地省站西北方向10万民众安全问题
4	16	5.40	2019-11-02至2019-12-26	暮云街道/	A2区丽发新城附近修建搅拌厂噪音、灰尘污染
5	34	4.92	2019-01-02至2019-09-24	泉塘/	问问A市经开区东六线以西泉塘昌和商业中心以南的有关规划
6	40	4.54	2019-02-14至2019-09-09	星沙街道/	A7县星沙街道凉塘路的旧城改造什么时候会启动？
7	44	4.43	2019-07-21至2019-09-25	润芳园小区/	A5区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空气
8	61	4.35	2019-03-24至2019-03-29	郝家坪小学/	A3区郝家坪小学什么时候能改扩建?
9	11	4.26	2019-03-25至2019-05-28	中茂城/	A7县星沙中茂城恶意拖欠购房资金不退还！
10	9	4.25	2019-10-29至2019-12-19	市政府/	请求将原A市属下区中学收回市教育局的申诉

表 6　2.13 计算热点问题热度指数表

现在需要使用函数，将表 6 中的热度指数映射到 0~10 空间上，我们规定结果只取到小数点后一位。我们采用的映射函数如式 5.2。这个函数是仿照 sigmoid 函数得到的，其中x是式 2.13 中的$score$，　只需要将 2.13 中算出的$score$代入该函数得到最后的绝对热度指标。

$$f(x)=\frac{10}{1+e^{-\left(\frac{x}{2}-1.65\right)+1}} \tag{5.1}$$

图 17，是该函数的图像。从该图像可以看出，对于很较高的热度值将统统映射为 10，而这个并非不合理，因为我们是将 A 市 2019 年热度值最高的事件规定为为 10。

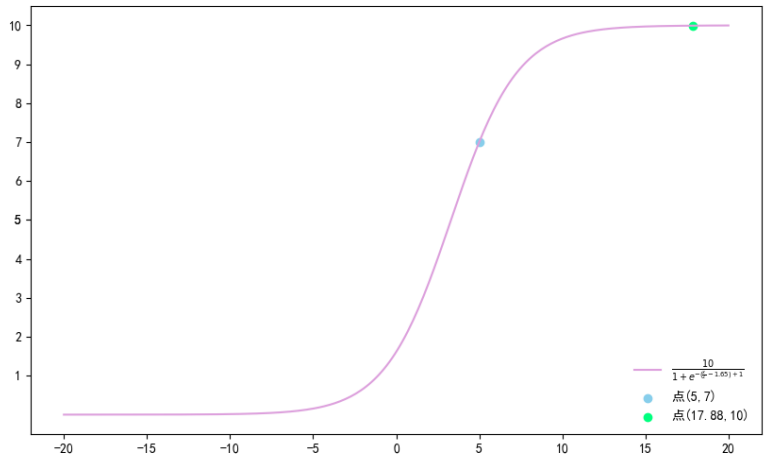


图 17　5.1 式函数图像

下面我们打算进一步探究这个热度评价公式的合理性。首先绘制 2019 年 A 市各热点事件的绝对热度指数的直方图和核密度估计曲线，如图 18。该图反映了热点事件的分布情况，。图中显示了高热度事件是比较稀少的，大多数热点事件的热度指数都集中在 3。这和实际情况是一致的。

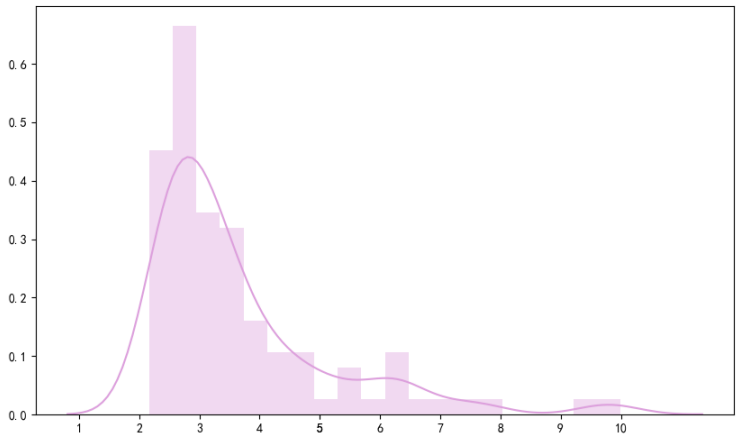


图 18　热点指数分布图

热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
1	12	10.0	2019-07-07至2019-09-01	广铁/	关于伊景园滨河苑捆绑销售车位的投诉
2	0	9.6	2019-01-08至2019-09-09		请加快A市国家中心建设刻不容缓
3	10	7.8	2019-04-17至2019-09-06	开元路/	反映A市地铁3号线松雅西地省站西北方向10万民众安全问题
4	16	7.4	2019-11-02至2019-12-26	暮云街道/	A2区丽发新城附近修建搅拌厂噪音、灰尘污染
5	34	6.9	2019-01-02至2019-09-24	泉塘/	问问A市经开区东六线以西泉塘昌和商业中心以南的有关规划
6	40	6.5	2019-02-14至2019-09-09	星沙街道/	A7县星沙街道凉塘路的旧城改造什么时候会启动?
7	44	6.4	2019-07-21至2019-09-25	润芳园小区/	A5区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空气
8	61	6.3	2019-03-24至2019-03-29	郝家坪小学/	A3区郝家坪小学什么时候能改扩建?
9	11	6.2	2019-03-25至2019-05-28	中茂城/	A7县星沙中茂城恶意拖欠购房资金不退还!
10	9	6.2	2019-10-29至2019-12-19	市政府/	请求将原A市属下区中学收回市教育局的申诉

表 7　绝对热度指数表

图 19 是热度指数排名前 5 名的热点事件的时间分布图。图中显示各个留言的时间分布，我们在设计公式时使用了时间的标准差的倒数作为系数，也就意味着时间分布越集中，热点指数的值会越高；时间分布越分散，热度值会有所降低。热度指数最高的事件，即" 伊景园滨河苑捆绑销售车位"，其留言的时间是很集中的，说明它是 A 市 8 月，9 月需亟待解决的一件事。热度指数排名第二的事件是"加快 A 市国家中心建设"，其留言的时间分布很分散，几乎全年各个时段都有留言，但留言数量特别多。

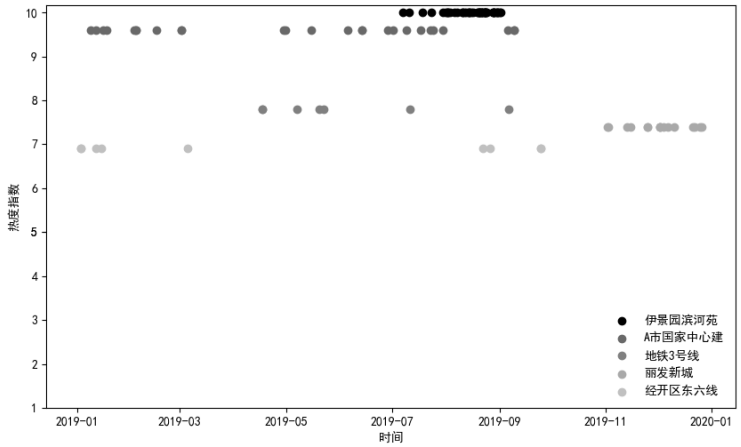


图 19 热度指数排名前 5 的热点事件留言的时间分布

六、模型评价

6.1 特征如何继续优化

我们根据 tf-idf 和信息增益的交集来选取特征，将特征向量的维数降低到了 1811 维，然后通过 tf-idf 加权。训练过程中对交通运输这一类的召回率很低，只有 0.64，这与原始数据分布不平衡有一定关系，我们可以通过对不同类型选取不同的特征数量来进行优化。当我们通过选取交通运输这一类的低频词添加到特征中进行模型训练。测试集交通运输的召回率确实有了提高，但是对其它类有了影响。选特征是个和复杂的过程，将来可以通过慢慢微调来获得最好的结果。

在调试最佳特征维数时，我们使用的是伯努利分布的朴素贝叶斯作为反馈，也就是要使伯努利分布朴素贝叶斯分类器用于测试集的总体 f1-score 达到最高。对不同的分类器可能存在不同的特征维数，这就需要逐一训练，来确定不同分类器的不同特征维数。

对特征加权也可以继续优化。问政留言文本会出现很多命名体，而大部分命名体都与文本的主题有关，我们可以给命名体赋予更高的权重，以期望提高分类的 f1-score。

6.2 模型参数如何继续优化

文中用到的各种模型的参数大都是手动调试的，为了提高训练的精度，我们可以使用启发式算法去优化这些参数，比如遗传算法与蚁群算法。在使用这些算法时，必须要得到一个反馈来调整启发式算法的步进方向。我们可以使用测试集的 f1-score 作为反馈，步进方向是使这个值越高越好。但这样就会出现过拟合问题，也就是测试集也纳入了模型的训练，但再来新的测试集时拟合效果会变差。所以我们必须还要考虑使用交叉验证。通过交叉验证与测试集的 f1-score 来得到一个合理的反馈用于启发式算法的步进，直到找到可能是最优的参数设置。

在实验中，聚类算法的 CH 指标用于提升聚类器参数反馈指标不是很合理，值太大会导致过度聚类，大小了聚类程度又不足。所以要想来优化 DBSCAN 的参数，就必要找到一个 z 折中的 CH 值来进行聚类结果的好坏评价。然后才能用于 DBSCAN 模型参数的训练。

参考文献

[1]Bird,S. Klein,E. Loper,E. Python 自然语言处理[M].人民邮电出版社，2014.

[2]黑马程序员. Python 数据分析与应用：从数据获取到可视化[M].中国铁道出版社, 2019.

[3] 程 ， 刘 ， 闵 , et al. 一种低频词词向量优化方法及在短文本分类中的应用 [J/OL], : 1–11.

[4] R. Baeza-Yates and B. Ribeiro- Neto. Modern Information Retrieval (2011) [M]. Addison Wesley,pp.68-74.

[5]C.D. Manning, P. Raghavan and H. Schuetze. Introduction to Information Retrieval (2008) [M]. Cambridge University Press, pp. 234-265.

[6]宗成庆. 统计自然语言处理[M]. 清华大学出版社，2013.

[7] 基于语义的高质量中文短信文本聚类算法-【维普官方网站】-www.cqvip.com 维普网 [EB/OL].

[8] 刘金岭. 基于语义的高质量中文短信文本聚类算法 [J/OL], , 35(10): 201– 202.

[9] 王 ， 贾 ， 李 . 基于 K-means 算法改进的短文本聚类研究与实现 [J/OL], , 43(12): 76–80.

[10] 网络舆情热点发现与分析研究–《西南交通大学》2011 年硕士论文 [EB/OL]. .

[11] 曾 ， 袁 ， 邵 , et al. 文本特征提取的研究进展 [J/OL], , 11(6): 706–715.

[12] LE Q V, MIKOLOV T. Distributed Representations of Sentences and Documents[J/OL],

[13]张晓辉，李莹，王华勇等.应用特征聚合进行中文文本分类的改进 KNN 算法. 东北大学.2003

[14]张晓雷. 面向 Web 挖掘的主题网络爬虫的研究与实现[D].西安电子科技大学,2012.

[15] 杨 . 基于线性支持向量机的文本分类应用研究 [J/OL], (3): 146–148.

[16] 段 ， 姚 . 政媒融合问政平台非正式文本自动分类匹配研究 [J/OL], : 1–9.

[17] 王 ， 葛 ， 张 , et al. 增量式聚类的新闻热点话题发现研究 [J/OL], (3): 46–50.

[18] 彭敏 , PENG MIN H J. 基于频繁项集的海量短文本聚类与主题抽取 [J/OL], 52(9): 1941.

[19] 基于向量空间模型的文本聚类算法–《计算机工程》2008 年 18 期 [EB/OL]. .