湖南省人民政府学位委员会办公室
湖南省教育厅学位管理与研究生教育处

2016 湖南省研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省研究生数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的试题中选择一项填写）：B

我们的参赛报名号为（如果组委会设置报名号的话）：014

所属学校（请填写完整的全名）：国防科学技术大学

参赛队员（打印并签名）：1.封宏俊

2.毛楚乔

3.刘宇捷

指导教师或指导教师组负责人(打印并签名)：

日期： 2016 年 4 月 19 日

评阅编号（由组委会评阅前进行编号）：

2016 湖南省研究生数学建模竞赛

编 号 专 用 页

评阅编号（由组委会评阅前进行编号）：

评阅记录（可供评阅时使用）：

评阅人										
评分										
备注										

湖南省第二届研究生数学建模竞赛

题 目　基于贝叶斯的改进关联规则频繁项集分类系统

摘　要：

基于互联网的法律咨询是提高法律服务效率、均衡法律资源的重要途径，但法律事务的复杂多样限制了群众的寻法效率。为改善这种状况，针对法律网站浏览数据，本文提出一种基于朴素贝叶斯的改进关联规则频繁项集分类系统，其中数据预处理模块采用了基于隐马尔科夫模型的分词过滤规则，频繁项集的挖掘采用了经典的 Apriori 算法，最后由频繁项集得到其相应条件概率实现分类。

在此，数据预处理实现对用户网页访问信息及搜索词条的加权分词处理，引入了相关数学模型以近似描述相关网页信息。本文采用了结巴分词实现语句分词，在此基础上，实现三类词语的过滤，分别是停用词、常用词以及近义词，其中使用结巴分词自带的 TF-IDF 函数获取训练数据的常用词。在提取关键词后，采用 Apriori 算法统计高频关键字并计算其关联词库，系统模型初始化完毕。

当用户输入数据时，我们采取相同的数据预处理方法并将得到的关键词与关联词库作匹配处理。根据贝叶斯准则，该系统可以实现四个功能：

1、 针对当前输入数据，预测其关键字并加以提示。

2、 根据输入数据，向用户推荐法律事务类型。

3、 通过用户交互，动态调整关联词库，以实现更精确的推荐分类。

4、 根据法律事务类型，根据关键词的相似性度量，向用户提供相应链接及律师支援。

仿真表明，该系统能准确高效地针对特定问题找到相关法律支援。

关键词：隐马尔科夫(HMM)模型　结巴分词　Apriori 算法　朴素贝叶斯

一、问题的重述与分析

1.1 问题的重述

随着我国社会经济发展，法律服务需求不断增长，但法律资源总体稀缺，且地区分布不均衡。以律师人数为例，目前全国执业律师约 27 万，北京万人律师比最高，约为 12.6，多个省份万人律师比不足 1。限制了法律服务的范围和质量。法律咨询网站应运而生，利用互联网自身的特点，普及法律常识，增强公众法律意识，提升“普法”的广度与深度，降低法律咨询的难度和成本，缩小地区差异，帮助用户，特别是经济不发达地区的用户，通过法律途径伸张公平与正义。互联网正逐步成为人们获取法律服务的重要途径。

但由于法律事务的复杂性、多样性、专业性，与当面咨询相比，现有法律网站交互功能不足，无法与用户形成有效互动。并且绝大多数用户缺乏专业法律知识，不了解以往类似案例，一般不能准确判断问题所适用的法律、法规条文，甚至不能确定法律类型，也很难使用恰当的关键词进行有效地搜索。因此，很多用户难以通过网站导航、关键词搜索等传统方式，高效、快捷地独立找到有针对性的法律知识、相关案例或专业律师，影响用户体验，限制网站发挥作用。

为提高服务效率，法律网站希望在分析用户特征的基础上，通过优化页面组织结构，增加新功能，改进服务模式等方式，使用户针对特定问题能够尽快找到相关页面或专业律师。

问题一：根据附件数据，分析用户关注领域、访问路径、用户体验等方面的特征或规律。尝试评估针对特定问题找到相关页面或专业律师的成功率和效率。

问题二：尝试通过优化页面组织结构，增加新功能（如，自动推荐），建立新型服务模式（如，向导式助手）等方式，改进服务效率，并对改进效果进行对比、评估。改进方案原则上应为自动服务，如果确实需要人工干预，则应明确人工干预的具体方式，并估计工作量。

1.2 问题的分析

针对问题一，我们认为核心的数据仍是用户输入的搜索信息，其次应对用户浏览时间、访问路径等方面进行加权分析。在此基础上得到一个分类系统，进而对各类不同的法律事务进行逐一分析，以获得更好的关联规则及指导，具体的问题回答需要结合仿真结果，具体参见第五章。

针对问题二，结合关联分类模型，我们提出了基于关键词、法律事务及律师的推荐，并根据推荐反馈结果动态调整贝叶斯系统，该系统的人工干预主要体现在前期精确数据的获取上，而后期的工作应是机器学习的结果，因此人工干预并不大。

二、程序说明

本文采用 Python 语言进行数据处理及仿真，所用到的库及其版本如下表所示：

表 1 程序库及版本

beautifulsoup4-4.3.2
jieba-0.38
xlrd-0.9.4
python3.4

主程序入口为 main.py，数据应置于上层路径之下。该程序可分为训练模式以及推荐模式，在训练模式下，程序读取相应的数据文本建立系统模型，并向外写出系统模型文件。在推荐模式下，程序读取系统模型文件，对输入的信息进行预测处理，具体参见程序中的说明文件。

三、模型相关算法

3.1 结巴分词

中文分词是中文文本处理的一个基础性工作，利用结巴分词进行中文分词，把得到的数据进行预处理，以便于后期对单个关键词进行处理。结巴分词目前支持三种分词模式：

1）精确模式，试图将句子最精确地切开，适合文本分析；

2）全模式，把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；

3） 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

考虑到本文所建立模型内存空间的限制及对于运算能力的要求，拟采用精确模式进行分词。其基本实现原理有三点：

1.基于 Trie 树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）。

2.采用了动态规划查找最大概率路径，找出基于词频的最大切分组合

3.对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。

3.1.1 HMM 模型

隐马尔可夫模型（Hidden Markov Model, HMM）是统计模型，它用来描述一个含有隐含未知参数的马尔可夫过程。其难点是从可观察的参数中确定该过程的隐含参数，然后利用这些参数来作进一步的分析，例如模式识别。隐马尔可夫模型是马尔可夫链的一种，它的状态不能直接观察到，但能通过观测向量序列观察到，每个观测向量都是通过某些概率密度分布表现为各种状态，每一个观测向量是由一个具有相应概率密度分布的状态序列产生。所以，隐马尔可夫模型是一个双重随机过程----具有一定状态数的隐马尔可夫链和显示随机函数集。

中文分词可以描述为：当给定观测序列和模型参数，寻找某种意义上最优的隐状态序列。在此，马尔科夫模型中隐含状态不能直接观测但却更具有价值，通常利用 Viterbi 算法来寻找。具体而言，句子的分词方法可以看成是隐含状态，而句子则可以看成是给定的可观测状态，从而通过建 HMM 来寻找出最可能正确的分词方法。

3.2 TF-IDF 模型

TF-IDF(term frequency-inverse document frequency)[1]是一种用于信息检索与数据挖掘的常用加权技术。TF-IDF 模型的主要思想是：如果词 w 在一篇文档 d 中出现的频率高，并且在其他文档中很少出现，则认为词 w 具有很好的区分能力，适合用来把文章 d 和其他文章区分开来。

TF-IDF 实际上是：TF*IDF，TF 代表词频（term frequency），IDF 为逆向文件频率（term frequency）。具体而言，TF 表示词 w 在给定文档 d 中出现的频率,即词 w 在文档 d 中出现次数 $count(w,d)$ 和 w 在文档集合 D 中出现总次数 $count(w,D)$ 的比值：

$$TF = count(w,d) / count(w,D) \quad (1)$$

IDF 是一个词语普遍重要性的度量，即文档总数 n 与词 w 所出现文件数 $docs(w,D)$ 比值的对数：

$$IDF = \log(n / docs(w,D)) \tag{2}$$

如果包含词 w 的文档越少，也就是 $docs(w,D)$ 越小，IDF 越大，则说明词条 w 在整个文件集合中具有低文件频率，具有很好的类别区分能力。

直观上看，TF 描述的是文档中词出现的频率，IDF 是和词出现文档数相关的权重。某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

3.3 基于 Apriori 算法的数据关联分析

在法律咨询中，关键字之间必然存在极大的关联性，直接匹配并计算其频率的时间及空间复杂度过大，且其许多关键字之间并无关联，因此剪掉不必要的匹配及搜索是关键的。本文采用 Apriori 算法[2]分析数据的关联性。Apriori 算法是一种挖掘布尔关联规则的频繁项集算法，其核心思想是通过候选集生成和情节的向下封闭检测两个阶段来挖掘频繁项集。该关联规则在分类上属于单维、单层、布尔关联规则。

3.3.1 基本定义

关联规则（Association Rules）反映一个事物与其他事物之间的相互依存性和关联性。如果两个或者多个事物之间存在一定的关联关系，那么，其中一个事物就能够通过其他事物检测到。假设通过事件 X 可以推导得到 $Y: X \to Y$，此时我们称 X 和 Y 为关联规则的先导(antecedent)和后继(consequent)。

相关如下定义：设 $I = \{i_1, i_2, \ldots, i_m\}$ 是 m 个不同元素的集合，每个元素 $i_k (k = 1, 2, \ldots, m)$ 称为一个项目(Item)，将集合 I 称为项目集合(Itemset)，简称为项集。项集的元素个数称为项集的长度，长度为 k 的项集称为 k-项集。设 D 是交易数据库，其中的每一事件 T 是 I 中一组项目的集合，即 $T \subseteq I$。对于项集 $X \subseteq I$，如果 $X \subseteq T$，则称事件 T 支持 X。设 $\text{count}(X \subseteq T)$ 为交易集 D 中包含 X 的交易数量，则项集 X 的支持度表示其出现的频率：

$$\text{support}(X) = \frac{\text{count}(X \subseteq T)}{|D|} \tag{3}$$

对于关联规则 $R: X \to Y$，其中 $X \subseteq I$，$Y \subseteq I$，并且 X，Y 不相交，定义其支持度为交易集中同时包含 X 和 Y 的交易数与所有交易数之比：

$$\text{support}(X \to Y) = \frac{\text{count}(X \cup Y)}{|D|} \tag{4}$$

定义其置信度为包含 X 和 Y 的交易数与包含 X 的交易数之比：

$$\text{confidence}\left(X \to Y\right) = \frac{\text{support}\left(X \cup Y\right)}{\text{support}\left(X\right)} \tag{5}$$

关联规则的最小支持度(Minimum Support)表示发现关联规则所必须满足的最小阈值，记为 $\text{sup}_{\min}$。定义支持度大于或等于 $\text{sup}_{\min}$ 的项集称为频繁项集，反之则成为非频繁项集。如果 k-项集满足 $\text{sup}_{\min}$，称为 k-频繁项集，记作 L_{k}。关联规则的最小置信度 $\text{conf}_{\min}$(*Minimum Confidence*) 表示关联规则需要满足的最低可靠性。如果规则 $R: X \to Y$ 满足 $\text{support}(X \to Y) \geq \text{sup}_{\min}$ 且 $\text{confidence}(X \to Y) \geq \text{conf}_{\min}$，称关联规则 $X \to Y$ 为强关联规则，否则称为弱关联规则。

具体而言，关联规则挖掘可分为两步，一是找出所有频繁项集，即所有支持度大于或等于最小支持度的项集。二是由频繁项集产生强关联规则，这些规则必须大于或等于最小支持度和最小置信度。

3.3.2 Aprior 算法的基本流程

1)找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。

2)由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度

3)使用第 1 步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来.

4)用递归的方法生成所有频集。

3.3.3 频繁模式树（Frequent Pattern tree, FP-Tree）

Apriori 算法需要多次扫描数据表，且会产生极大的频繁集，在大数据的情况下，这使得算法的时间及时间复杂度过大。要降低该算法的复杂度，需要从频繁集的定义入手。不难发现，对于一个频繁项集而言，其所有子集一定是频繁的。若某项集是非频繁的，则其所有的子集也一定是非频繁的。因此，频繁集可以构成称树状，已达到分而治之的目的，这就是 FP-grouth 算法的由来。

频繁模式树使用了一种紧缩的数据结构来存储查找频繁项集所需要的全部信息。FP-Tree 算法[3]的基本数据结构，包含一个一棵 FP 树和一个项头表，每个项通过一个结点链指向它在树中出现的位置。基本结构如下所示。需要注意的是项头表需要按照支持度递减排序，在 FP-Tree 中高支持度的节点只能是低支持度节点的祖先节点。

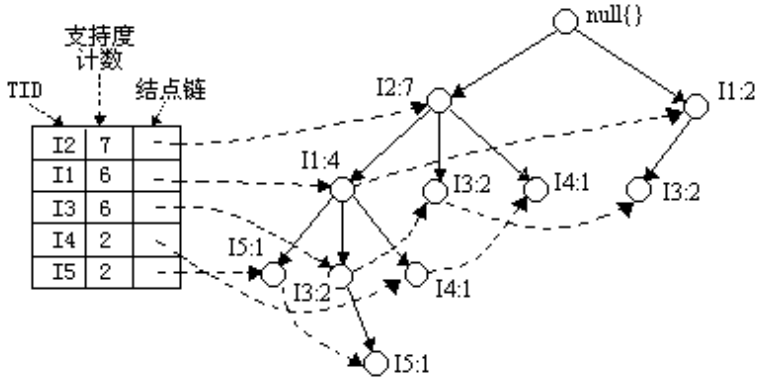

图 1　FP-Tree 的实现

产生 FP-Tree 主要分为以下两步：

1.构造项头表：扫描数据库一遍，得到频繁项的集合 F 和每个频繁项的支持度。把 F 按支持度递降排序，记为 L。

2.构造 FP-Tree：把数据库中每个事物的频繁项按照 L 中的顺序进行重排。并按照重排之后的顺序把每个事物的每个频繁项插入以 null 为根的 FP-Tree 中。如果插入时频繁项节点已经存在了，则把该频繁项节点支持度加 1；如果该节点不存在，则创建支持度为 1 的节点，并把该节点链接到项头表中。

3.4 向量空间模型

向量模型[4]把文档和查询串都视为词所构成的多维向量，而文档与查询串的相关性即对应于向量间的夹角，从而得到查询串与不同类型文档之间的相关性。向量模型认识到布尔模型中的二元权重的局限性，从而提出了一个适合部分匹配的框架。它在查询串和文档之间分配给索引术语非二元的权重，这些术语权重反映了数据库中的每篇文档与用户递交的查询串的相关度，并将查询返回的结果文档集按照相关度的降序排列，所以向量模型得到的文档是部分地匹配查询串。向量模型的优点在于根据秩（rank）返回的结果集要比布尔模型返回的结果集在感觉上更加符合检索用户的需要。

向量模型的优点在于：

1）术语权重的算法提高了检索的性能；

2）部分匹配的策略使得检索的结果文档集更接近用户的检索需求；

3）根据结果文档对于查询串的相关度通过 Cosine Ranking 公式对结果文档进行排序。

3.5 朴素贝叶斯分类器 (Naive Bayes Classifier, NBC)

朴素贝叶斯分类器[5]具有坚实的数学基础以及稳定的分类效率，同时期所需的估计参数很少，对缺失数据不太敏感，因此，本文才用朴素贝叶斯模型对关键词作法律事务分类。

在关联规则分析中，我们已获得不同规则的支持度，即当法律事务类型为 B 时，包含关键词组 A 的概率 $P(A|B)$，根据贝叶斯准则，我们容易求得当搜索信息中包含关键词组 A 时，所需信息为法律事务 B 的概率：

$$P(B|A)=\frac{P(AB)}{P(A)}=P(A|B)\cdot\frac{P(B)}{P(A)} \tag{6}$$

由此获得关于关联规则 $A\rightarrow X$ （X 为所有法律事务）的置信度，取其中较大者作为其相应的法律事务。

四、模型的建立与求解

系统可分为三个模块，分别是数据预处理器、Apriori 模型以及贝叶斯模型，下面进行简要分析。

4.1 数据预处理器

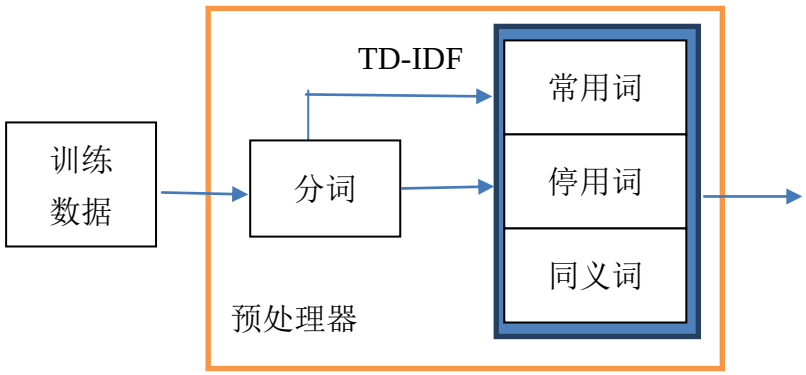

图 2. 数据预处理结构图

数据预处理器可分为三步：

一、对输入信息进行分词处理，并将所有分词汇集组成一个文本。

二、对文本采用 TF-IDF 算法，得到常用词组，根据相关信息划定停用词以及定义同义词。

三、对各条分词进行过滤，得到处理后的关键词集，根据用户浏览时间确定该记录的权值，一般而言，权值与浏览时间成正比，在此才用 log 函数来拟合，结合相关数据，设定其基底为 1000, 设定权值的最小值为 1，则记录权值的计算公式为：$\log_{1000}(time+1000)$

```
# Data preprocessing
@input title
@output keywordSet with per frequency

# text consists of the title
common_words = td_idf(text)
# stop_words, synonym_words get from the internet
for i in range(title.count)
    word = title[i]
    if word in stop_words or common_words
        continue
    if word in synonym_words
        word = the synonym of title[i]
    frequency_of_word += weight_of_word
    if word not in keywordSet
        keywordSet.add(word)

return keywordSet, frequency
```

4.2 Apriori 模型

基于对现有数据的分析，本文调整了 Apriori 模型，具体步骤如下：

1、 初始化支持度系数为 0.25，各关键词的词频源自数据预处理模块，因各法律事务的判断依赖于关键词，部分事务的数据量较小，因此设定各事务至少包含 15 个关键词，否则将自动降低支持度系数，重新遴选频繁项集。

2、 根据算法的相关规则，生成 k-频繁项集，由于数据量较大，可对 k 进行限制，根据仿真经验，k 值一般设置在[3,5]之间，以确保其准确性以及运算效率。

```
# Apriori
@imput frequency, keywordSet
@output keywordDict

L1 = frequency
k = 3
for i in range(2, k+1)
    Li is the combination of L1 excluding
        the subsets which are not in Li-1
    search Li in the whole data table and
        get the frequency of each, named FreqDict
    Update Li according to FreqDict
```

4.3 贝叶斯模型

在获得 Apriori 模型后，对用户输入的相关信息作相同的预处理，然后统计其长度不大于 k 的关键词组合在 Apriori 模型中出现的次数，在此，可作如下假定，组合关键词的数据库全集中的频数应与在 Apriori 模型中的频数大致相等，这是由频繁项集的定义所支持的。假定出现组合关键词的事件为 $A_j(j=1,2;\cdots,\mathrm{M}$，其频数为 $\mathrm{count}(A_j)$，其中 M 为组合关键词的集合长度。对于所有法律事务 $B_i(i=1,2,\cdots,\mathrm{N})$，其频数为 $\mathrm{count}(B_i)$，其中 N 为法律事务总数。根据上述计算可知 $\mathrm{P}(A_j \mid B_i)=\mathrm{count}(A_j)/\mathrm{count}(B_i)$，根据贝叶斯准则，可得

$$\mathrm{P}\left(B_i \mid A_j\right)=\mathrm{P}\left(A_j \mid B_i\right)\cdot\frac{\mathrm{P}\left(B_i\right)}{\mathrm{P}\left(A_j\right)} \tag{7}$$

其中 $\mathrm{P}\left(A_j\right)=\sum_{i=1}^{N}\mathrm{P}\left(A_j \mid B_i\right)\cdot\mathrm{P}(B_i)$，由此求得等式左边。在计算过程中，使用以下方法简化计算：

1、 在计算关键词组合时，应摒弃不在频繁项集中的关键字。

2、 关键字的组合阶数应与建模时一致，避免不必要的计算量。

```
# Bayes
@input DictList, userKeyword
@output lawType

k = 3
for each in userKeyword
    if each not in DictList
        remove

for i in range(1, k+1)
    wordList = Combination(userKeyword) with length i
    frequence = []
    for word in wordList
        for keywordDict in DictList
            f = get the max frequence for word in keywordDict
            and divide it by the frequence for word in DictList

        frequence.append(f)

sort(frequence) and get the priori 3 lawTypes
```

五、仿真结果及问题回答

1、 仿真结果

根据上述系统模型，本文采用归类后的数据进行训练，并用它作学习样本加以预测，仿真中提取的部分法律事务关键词信息如下：

企业注册：个人 税法 经营 登记 法人 营业执照 管理 个体 公司 规定 工商 纳税 变更 资料 注册

劳动合同效力：规定 解除 期限 赔偿 劳动 合同 单位 通知 固定 员工 工作 续签 签订 协议 公司

财产分割：结婚 财产 房子 房产 共同 子女 协议 名字 房产证 夫妻 户口 购房 离婚 婚姻 父母

在以训练数据作为学习数据时，得到各种法律事务的误判率，部分数据如下表所示，值得注意的是，具有较高误判率的法律事务主要源自错误的数据类型，即记录中的关键词与法律事务并不匹配，对于这种状况，可采取两种方式解决，一为人工筛选出优质的训练数据，二是增大各种法律事务类型的关键词，后者将使处理复杂度增大，在本文中仅法律事务的下限值设为 15。

表 2 各事务类型的误判率

法律事务类型	误判率
赡养	0.3771217712177122
无效婚姻	0.38524590163934425
债务转让	0.5662650602409639
债务承担	0.7101449275362319
土地出转让	0.45652173913043476
破产清算纠纷	0.1694915254237288
婚前财产	0.9221411192214112
生育权	0.3777777777777777
保险诈骗	0.2777777777777778
领养	0.6176470588235294
保险合同纠纷	0.0

2、 问题回答

1、从仿真的结果出发，我们发现了不同法律事务类型具有不同的服务效率，其中以劳动合同效力、故意伤害、伤害赔偿等法律事务类型误判率最高，这体现了这些领域复杂度更大；其次，我们发现了访问时间越长，该法律事务与关键词匹配的程度也会显著提高。从仿真入口来看，直接进入与从其他搜索引擎处进入的用户相比，其关键词并未体现明显的精确。综上，用户针对特定问题找到相关页面的成功率因法律事务类型不同而不同，具体可参考误判率，通过人工核对，我们发现误判率中有近 20%的数据与法律事务不相匹配，因此，成功率可大致估算为 S1=1-0.8*误判率。通过对律师的信息分析，我们发现大部分律师的推荐只能以地域为主，因为他们的标签以及说明与法律事务类型并无直接的关联关系。显然，即使法律事务类型判断正确，但找到相应专业律师却依然无法保证，因此，找到专业律师的成功率可粗略估算为 S2=S1*delta,其中 delta 为小于 1 的任意值。

2、为提高网站服务效率，基于本文提出的分类系统，本文对网站提出了如下改进意见及新功能：

a.利用机器分类进而人工分类的方法获取更精确且数据量更大的样本，使得词库内基本涵括相应法律事务的基本用词。

b.设计交互式输入，对用户已输入的信息作法律事务类型匹配，并推送词库中相应的频繁项集，根据用户的选择进而对词库中的词频等进行动态调整。

c.将最为常用的法律专业词映射为多维空间向量，通过计算向量间的相似性向用户推荐已有回答或是相关信息界面。

参考文献

[1]李春梅. 基于 TF-IDF 的网页新闻分类的研究与应用[J]. 贵州师范大学学报(自然科学版), 06:106-109, 2015.
[2]吕萍丽,段滋明. 浅谈 Apriori 算法[J]. 电脑知识与技术,04:715-716,2011.
[3]况莉莉. Apriori 算法与 FP-tree 算法的探讨[J]. 淮北煤炭师范学院学报(自然科学版),02:44-49,2010.
[4] Ian H.Witten、Eibe Frank,数据挖掘实用机器学习技术, 机械工业出版社, 2006.
[5]邸鹏,段利国. 一种新型朴素贝叶斯文本分类算法[J]. 数据采集与处理, 01:71-75,2014.