UNIVERSITY OF CHINESE ACADEMY OF SCIENCES

HOMEWORK OF DATA MINING

Assignment & MessageClassify

姓名: 孙学超

学号: 2017280150290 11

December 16, 2017

1 Appoach

1.1 Feature Extraction

对于文本分类问题，无非就是判断一个文档属于哪一个类别。细化到垃圾短信识别的场景中，文本分类问题就变成给我们一个文档（短信），给定一个算法可以正确地将其划分为垃圾短信或者非垃圾短信。形式化的表示为

$$f : D \to C \tag{1}$$

其中D为文档（短信）集合，C = {垃圾短信，非垃圾短信}。

然而，由于语言的变化繁多，同样一句话可以拥有很多种表述。所以单纯的匹配整个句子是不现实，必须要对文档进行处理使用。由于针对不同的两句话，如果它们包含的词语大致相同，那么其表达的意思也会相近。所以我们想到了分词。原问题就可以形式化的规约为

$$\begin{aligned} f_1 &: D \to W \\ f_2 &: W \to C \end{aligned} \tag{2}$$

其中D为文档（短信）集合，W为词语集合，C = {垃圾短信，非垃圾短信}。

分词已有许多成熟的分词工具来实现，例如jieba、IKAnalyzer等。

分词之后，每个文档都会拥有不同的词语组合。那么，词语的维度随之也会增加。同时，有些词语对于我们的分类问题没有什么作用。例如一些在所有文档中都出现的词不会给我们区分类别带来更多的信息。所以我们需要通过一定的方法将词语集合中一些能够用来区分不同类别的词语提取出来，我们称这类词语为当前文档集合的特征词。提取特征词之后，原问题进一步规约为

$$\begin{aligned} f_1 &: D \to W \\ f_2 &: W \to F \\ f_3 &: F \to C \end{aligned} \tag{3}$$

其中D为文档（短信）集合，W为词语集合，F为特征词集合，C = {垃圾短信，非垃圾短信}。

1.1.1 Information Gain

在文本分类的特征值提取过程中，信息增益，即Information Gain，是一个非常流行的方法。但凡是特征选择，首先需要对每一个特征进行量化，进而将所有的特征按照重要程度进行排序，所以那些重要程度比较高的特征就会被保留。

在信息增益中，重要性的衡量标准就是看特征能够给系统带来多少信息，带来的信息越多，该特征就越重要，其就越该被保留。

• 信息量（熵）

那么信息的多少应该如何来评定？在信息论中，给出了有关信息量（也即是“熵”）的定义。

定义：

给定一个变量X，有n种可能的取值，分别为$x_1, x_2, x_3, ..., x_n$，并且针对每一种取值取到的概率为$P_1, P_2, P_3, ..., P_n$，那么对于变量X的熵为

$$H(X) = -\sum_{i=1}^{n} P_i \log_2^{P_i} \tag{4}$$

由上式可以看出，一个变量的变化越多，那么对于每个P_i的值就越小，n就会越大，从而它的熵就越大，携带的信息就越多。也就说明这个系统越混乱。

针对文本分类问题，类别C就是系统的变量，他可能的取值为$C_1, C_2, ..., C_n$，而每一类别出现的概率是$P(C_1), P(C_2), ..., P(C_n)$。那么此时分类系统的熵就可以表示为：

$$H(C) = -\sum_{i=1}^{n} P(C_i) \log_2^{P(C_i)} \tag{5}$$

特殊的，针对二值分类问题，类别总数为2，那么此时当前分类系统的熵就是

$$H(C'') = -P(C_1) \log_2^{P(C_1)} - P(C_0) \log_2^{P(C_0)} \tag{6}$$

总之，对于一个分类系统而言，我们可以通过计算其类别变量的熵来定义当前系统含有的信息量。

- **信息增益**

给定了信息量的定义之后，我们就可以阐述什么是信息增益。

信息增益是针对一个一个特征词而言的，即每一个特征词都对应一个信息增益值。对于任意一个特征词t，计算出当前系统有它和没它的信息量，分别记为H_t，$\bar{H}_t$。那么特征词t给系统带来的增益即为H_t - $\bar{H}_t$，这就是对于特征t的信息增益，记为IG_t。

到现在，我们仅仅需要计算两个数值就可以了，一个是系统有当前特征词t的信息量H_t，另一个是当前系统没有此特征词的信息量$\bar{H}_t$。

对于H_t，因为系统本身就包含特征词t，所以H_t的值就是公式5的值。然而对于$\bar{H}_t$，我们需要再次回到信息量最初的定义。

信息量表明了变量变化情况的多少，即信息量的大小跟变量的种数以及当前类别发生的概率有关。为便于理解，考虑下面的例子：

- **假设两个教室，每个教室中都有n个座位，同时有n个学生可以随便坐，由于一些原因两个教室发生了不同的变化：**

  教室1) 有一个座位被搬到了教室外面，同时一个学生也离开了这个教室。

  教室2) 有一个学生强行霸占了某一个座位，其他学生不能坐。

所以我们从直观上可以感觉到，两个教室的混乱程度在变化后是一样的。从而可以得出一个结论

- 1）系统不包含特征词t
- 2）系统包含特征词t，但是t已经被固定了，不能变化。

其中，1）和2）是等价的。

所以针对$\bar{H}_t$，我们就可以使用它的等价条件来计算，即系统包含这个特征词，但是这个特征词已经被固定。而针对此类问题的计算，在信息论中也已经给出，即条件熵，记为$H(C|t=t_i)$。计算公式为

$$H(C|t=t_i) = -\sum_{j=1}^{n} P(C_j|t=t_i)\log_2^{P(C_j|t=t_i)} \tag{7}$$

那么问题接踵而至，对于一个特征词t来说，它的取值含有多种$(t_1, t_2, t_3...t_n)$，当计算条件熵时，因为每一个值都有可能会发生，并且以一定的概率发生。那么对于此特征词的条件熵即为整个特征词t上对条件熵的期望

$$H(C|t) = \sum_{i=1}^{n} P_i H(C|t=t_i) \tag{8}$$

综上，信息增益就可以计算出来，对于任意一个特征词t，其信息增益记为IG(t):

$$IG(t) = H(C) - H(C|t) \tag{9}$$

特殊的，针对二值文本分类，C只有两种类别，同时每一个特征T也只有出现和不出现两种情况，将公式(9)展开得到

$$\begin{aligned} &IG(T) = H(C) - H(C|T) \\ &= -\sum_{i=1}^{n} P(C_i)\log_2^{P(C_i)} + P(t)\sum_{i=1}^{n} P(C_i|t)\log_2^{P(C_i|t)} + P(\bar{t})\sum_{i=1}^{n} P(C_i|\bar{t})\log_2^{P(C_i|\bar{t})} \end{aligned} \tag{10}$$

其中，t代表在当前系统中出现，$\bar{t}$代表在当前系统中不出现。

• 特征词提取步骤

经过上述讨论，我们可以得到使用信息增益（IG）方法来获取特征词的步骤:

1) 针对每一类别C_i，统计其包含的特征词次数
2) 针对每一特征词T，统计其在各个类别中出现的次数，并计算在各个类别中不出现的次数
3) 计算系统的熵，即信息量
4) 针对每一个特征词T，计算其信息增益
5) 取信息增益值较大的前k个特征词作为当前系统的特征词

1.2 Classify Solution

1.2.1 Naive Bayes

• 贝叶斯定理

在统计学领域，贝叶斯学派虽然从其诞生到一百年前一直都不是主流学派，但是随着社会的发展，贝叶斯学派理论的重要性越发突出。

贝叶斯学派的思想可以概括为先验概率+数据=后验概率。也就是说我们在实际问题中需要得到的后验概率，可以通过先验概率和数据一起综合得到。数据就是试验得到的数据，它对于后验概率的影响是毋庸置疑的。存在争议的就是先验概率对后验概率的影响。一般来说先验概率就是我们对于数据所在领域的历史经验，但是这个经验常常难以量化或者模型化，于是贝叶斯学派大胆的假设先验分布的模型，比如正态分布，beta分布等。这个假设一般没有特定的依据，因此一直被其他学派认为很荒谬。虽然难以从严密的数学逻辑里推出贝叶斯学派的逻辑，但是在很多实际应用中，贝叶斯理论却展现出非常好的效果，比如垃圾邮件分类，文本分类。

定理：条件独立公式

给定两个变量**X**，**Y**，如果**X**和**Y**相互独立，则有：

$$P(XY) = P(X)P(Y) \tag{11}$$

定理：条件概率公式

给定两个变量**X**，**Y**，

$$P(Y|X) = \frac{P(XY)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)} \tag{12}$$

根据公式(11, 12)，可以很容易得到贝叶斯公式。

定理：贝叶斯公式

给定两个变量**X**，**Y**，则有

$$P(Y_k|X) = \frac{P(X|Y_k)P(Y_k)}{P(X)} = \frac{P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|Y_k)P(Y_k)}{P(X)} \tag{13}$$

其中，（$Y_1, Y_2, ..., Y_k$）是变量**Y**的一个划分，即$\sum_k P(Y_k) = 1$。

• 朴素贝叶斯模型

从统计学回归到我们原始的数据分析当中。假如，我们分类模型的样本是：

$$(x_1^1, x_2^1, ..., x_n^1, y_1), (x_1^2, x_2^2, ..., x_n^2, y_2), ..., (x_1^m, x_2^m, ..., x_n^m, y_m)$$

即我们有**m**个样本，每个样本有**n**个特征，特征输出有**K**个类别，定义为$C_1, C_2, ..., C_K$。

这样对于上文中提到的先验概率，我们就有一个量化的指标。可以把每个类别在样本（训练集）中出现的频率作为当前分类的先验概率。也就是说，通过样本我们可以学习到朴素贝叶斯的先验分布$P(Y = C_k)(k = 1, 2, 3, 4....K)$。

但是对于贝叶斯公式(13)中的$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|Y_k)$却很难求出，这是一个**n**维的条件分布。一般情况下**n**的值会非常大。所以朴素贝叶斯模型在这里做了一个大胆的假设，即**X**的**n**个维度之间相互独立，这样就可以得出：

$$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|Y_k) = P(X_1 = x_1|Y_k)P(X_2 = x_2|Y_k)...P(X_n = x_n|Y_k)$$

这样以来对于每一个$P(X_i = x_i|Y_k)$就可以通过统计样本中的频率来得到。从上式可以看出，一个很难的分布被简化，但是却带来了预测的不准确性。即当特征之间不独立时，贝叶斯模型就会捉襟见肘。但是一般情况下，样本的特征独立这个条件是弱成立的，尤其在数据量比较大的时候。

最后贝叶斯公式(13)中只剩下$P(X)$一个量我们不能利用样本求出。但是针对同一条数据，其特征是相同的，那么P(X)也必然相同。对于分类问题，我们只需要计算当前的特征属于哪一个类别，也就是针对每一个类别求出使得贝叶斯公式$P(Y_k|X)$最大的类别，因为对于同一条数据P(X)是相同的，那么只需要最大化$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|Y_k)P(Y_k)$即可。

由上述讨论可知，朴素贝叶斯算法的计算公式为

$$C_{result} = \underbrace{\arg\max}_{C_k} P(Y = C_k) \prod_{j=1}^{n} P(X_j = x_j^{(test)}|Y = C_k) \tag{14}$$

其中$x_j^{(test)}$为测试集中特征X_j对应的值。

朴素贝叶斯参数设置

从上一节中得知，在使用贝叶斯分类时，只需要最大化$P(Y = C_k)\prod_{j=1}^{n} P(X_j = x_j^{(test)}|Y = C_k)$即可。对于$P(Y = C_k)$,比较简单，通过极大似然估计我们很容易得到$P(Y = C_k)$为样本类别$C_k$出现的频率，即样本类别$C_k$出现的次数$m_k$除以样本总数m。

然而对于$P(X_j = X_j^{(test)}|Y = C_k)(j = 1, 2, 3, ..., n)$而言，则取决于我们的先验条件：

a) 如果我们的X_j是离散的值，那么我们可以假设X_j符合多项式分布，这样得到$P(X_j = X_j^{(test)}|Y = C_k)$ 是在样本类别C_k中，$X_j^{(test)}$出现的频率。即：

$$P(X_j = X_j^{(test)}|Y = C_k) = \frac{m_{kj^{(test)}}}{m_k}$$

其中m_k为样本类别C_k出现的次数，而$m_{kj^{test}}$为类别为C_k的样本中第j维特征$X_j^{(test)}$出现的次数。

但是有些时候，可能某些类别在样本中没有出现，这样就会导致$P(X_j = X_j^{(test)}|Y = C_k)$为0，从而影响后验的估计。为了解决这种情况，我们引入了拉普拉斯平滑，即此时有：

$$P(X_j = X_j^{(test)}|Y = C_k) = \frac{m_{kj^{(test)}} + \lambda}{m_k + n\lambda}$$

其中n为特征词数目；λ 为一个大于0的常数，常常取为1。

b) 如果我们我们的X_j是非常稀疏的离散值，即各个特征出现概率很低。这时我们可以假设X_j符合伯努利分布，即特征X_j出现记为1，不出现记为0。即只要X_j出现即可，我们不关注X_j的次数。这样得到$P(X_j = X_j^{(test)}|Y = C_k)$是在样本类别$C_k$中，$X_j^{(test)}$出现的频率。此时有：

$$P(X_j = X_j^{(test)}|Y = C_k) = P(X_j|Y = C_k)X_j^{(test)} + (1 - P(X_j|Y = C_k))(1 - X_j^{(test)})$$

其中，$X_j^{(test)}$取值为0和1.

• **朴素贝叶斯算法优缺点比较**

朴素贝叶斯算法基于传统数学理论，拥有强大的理论基础。但是也有其不足之处。

朴素贝叶斯算法的主要优点有：

1) 朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。
2) 对小规模的数据表现很好，能够处理多分类任务，适合增量式训练，尤其是数据量超出内存时，我们可以一批批的去增量训练。
3) 对缺失数据不太敏感，算法也比较简单，常用于文本分类。

朴素贝叶斯算法的主要缺点有：

1) 理论上，朴素贝叶斯模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为朴素贝叶斯模型给定输出类别的情况下,假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有半朴素贝叶斯之类的算法通过考虑部分关联性适度改进。
2) 需要知道先验概率，且先验概率很多时候取决于假设，假设的模型可以有很多种，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。
3) 由于我们是通过先验和数据来决定后验的概率从而决定分类，所以分类决策存在一定的错误率。
4) 对输入数据的表达形式很敏感。

2 Experiment

2.1 Information Gain & Naive Bayes

2.1.1 Word Segmentation

对于文档的分词采用了中文分词工具包–jieba分词。利用jieba分词工具包中的cut()函数，将源数据中的每一条数据传入cut函数，并采用全模式分词，输出到新的文件中。新文件的格式为：

$$w_1|w_2|w_3|....|w_n$$

其中w_i为经过分词之后的词语。针对每一条数据其词语的个数并不固定，即n的大小不固定。

• **分词处理步骤**

1) 读取输入文件，对每一行数据调用jieba分词包中的cut函数，并传入相应参数，使其进行全分词模式
2) 获得cut函数的结果数据，进一步添加“|”进行分割不同的词语，同时删除其中的“x”字符。

2.1.2 Feacture Extraction

在本次试验中，对于特征词的提取采用了信息增益（Information Gain）的方法。其主要思想已经在前一节中详细讲述。

- **处理思路**

对于垃圾短信分类的问题的形式化表示为

$$f : D \to C \tag{15}$$

其中D为文档（短信）集合，C = {垃圾短信，非垃圾短信}。

其分类的类别数为2，针对每一个特征词也只有出现和不出现两种情况，所以我们可以将在前面介绍的关于信息增益的公式规约成

系统信息量:

$$H(C) = -(P_0 \log_2^{P_0} + P_1 \log_2^{P_1}) \tag{16}$$

其中P_0为非垃圾短信类别中的词语总数占所有文档词语总数的比例，P_1为垃圾短信类别中的词语总数占所有文档词语总数的比例。

条件熵:

$$\begin{aligned} &H(C|t) = P_i H(C|t = t_0) + P_i H(C|t = t_1) \\ &H(C|t = t_i) = -(P(C_0|t = t_i) \log_2^{P(C_0|t=t_i)} + P(C_1|t = t_i) \log_2^{P(C_1|t=t_i)}) \end{aligned} \tag{17}$$

其中t_0代表特征t不出现，t_1代表特征t出现；C_0代表非垃圾邮件类别，C_1代表垃圾邮件类别。

信息增益:

$$\begin{aligned} &IG(T) = H(C) - H(C|T) \\ &= -\sum_{i=1}^{n} P(C_i) \log_2^{P(C_i)} + P(t) \sum_{i=1}^{n} P(C_i|t) \log_2^{P(C_i|t)} + P(\bar{t}) \sum_{i=1}^{n} P(C_i|\bar{t}) \log_2^{P(C_i|\bar{t})} \end{aligned} \tag{18}$$

利用信息增益来提取特征词主要分为以下几步：

1) 统计正例和负例文档中词语的个数，当遇到相同词语时，作为两个词语计数。
2) 统计在系统中出现的每个词语分别在正例文档和负例文档中出现的次数。同一文档两次出现相同词语作为两个词语计数。
3) 根据公式（16）计算当前系统的信息量（熵）。
4) 根据公式（18）计算当前系统中的信息增益。并统计出前K个信息增益值较大的特征词。

经过上述步骤之后，最终能够得到所有特征词对于当前系统的信息增益值，并输出到文件中。

• **参数设置**

经过上述步骤之后，一般情况下我们都会得到最终的结果。但是仍然有特殊情况。当我们统计某个词语在正例文档中出现的次数为0时，那么根据公式（18）可得，$P(C_1|t=t_j)$为0，就会出现错误。为防止此类错误的发生，我们设置参数λ进行拉普拉斯平滑，即在统计每个词语在正负例文档中出现的频次时，同时加入参数λ。所以在计算$P(C_i|t=t_j)$时

$$P(C_i|t=t_j)=\frac{m_{ij}+\lambda}{m_j+n\lambda} \tag{19}$$

其中，n为类别数，即n=2。

2.1.3 Classify

经过特征词的提取之后，我们可以得到训练集中的特征词。接下来就是利用朴素贝叶斯算法来训练当前模型。

• **处理思路**

由上几节的讨论，我们知道，对于朴素贝叶斯算法的实质就是最大化公式（14）。对于垃圾短信分类问题我们也可以将其规约

$$C_{result}=\underbrace{\arg\max}_{C_k}P(Y=C_k)\prod_{j=1}^{n}P(X_j=x_j^{(test)}|Y=C_k) \tag{20}$$

其中C_k只有两种情况，C_0和C_1。

训练模型的步骤为

1) 统计正负样本的特征词总数，计算正负例各自的先验概率
2) 统计各特征词在正负样本中出现的频率，从而得到各特征词在正负分类条件下的条件概率
3) 当新的数据到来时，利用贝叶斯分类公式（20）,计算针对正负类别的后验概率，获得使其最大的类别即为新数据所属类别。

• **参数设置** 与信息增益提取特征值类似，在大多数情况下我们可以获得最终的结果，但是也仍然有特殊情况，使得我们最终的结果偏离较大。

当某一个特征词在某一类别下其出现的次数为0时，那么对于条件概率

$$P(X_j=X_j^{(test)}|Y=C_k)=\frac{m_{kj^{(test)}}}{m_k}$$

的值为0，这样会导致由于某几个特征，使得所有特征的对结果的贡献都没有意义。所以我们在此设置一个参数λ，引入拉普拉斯平滑，将公式规约为

$$P(X_j=X_j^{(test)}|Y=C_k)=\frac{m_{kj^{(test)}}+\lambda}{m_k+n\lambda}$$

其中n为特征词的数量，λ一般设置为1。

		F_{Num} = 50		F_{Num} = 105		F_{Num} = 500	
		$\lambda = 1.0$	$\lambda = 10000000$	$\lambda = 1.0$	$\lambda = 10000000$	$\lambda = 1.0$	$\lambda = 10000000$
1th	Percision	0.9827	0.9384	0.9877	0.9404	0.9935	0.9365
	Recall	0.9486	0.9557	0.9477	0.9587	0.9499	0.9588
	F1	0.9654	0.9470	0.9672	0.9495	0.9712	0.9475
2th	Percision	0.9836	0.9396	0.9886	0.9417	0.9939	0.9373
	Recall	0.9481	0.9555	0.9467	0.9586	0.9495	0.9580
	F1	0.9655	0.9475	0.9672	0.9501	0.9712	0.9475
3th	Percision	0.9834	0.9384	0.9883	0.9402	0.9939	0.9362
	Recall	0.9471	0.9547	0.9472	0.9571	0.9487	0.9578
	F1	0.9649	0.9465	0.9673	0.9486	0.9708	0.9469
4th	Percision	0.9829	0.9391	0.9883	0.9412	0.9940	0.9369
	Recall	0.9477	0.9556	0.9472	0.9584	0.9493	0.9582
	F1	0.9650	0.9473	0.9673	0.9497	0.9711	0.9475
5th	Percision	0.9836	0.9397	0.9885	0.9416	0.9939	0.9381
	Recall	0.9493	0.9564	0.9492	0.9604	0.9497	0.9587
	F1	0.9661	0.9480	0.9685	0.9509	0.9713	0.9483
Average	Percision	0.9832	0.9391	0.9882	0.9416	0.9939	0.9370
	Recall	0.9482	0.9556	0.9472	0.9586	0.9494	0.9583
	F1	0.9654	0.9472	0.9673	0.9497	0.9711	0.9475

表 1: λ是拉普拉斯平滑参数，F_{Num}是取特征词的数量

2.1.4 Evaluation

对于模型的评价，我们采用percision、recall和F1三个指标进行评价。同时采用5-fold的方法来分割训练集，针对每一组，其中4/5的训练集用来训练模型，1/5的训练集用来评价测试。

由表可得,λ对结果的影响远小于F_{Num}。因为λ是拉普拉斯平滑的参数。从拉普拉斯平滑的公式可知，其意义主要是在计算概率之前添加进一部分先验概率，避免求得概率是0。而这个先验概率是我们简单的假设在每一类的文档中对于任何一个特征词的个数都是λ得到的。因为训练集比较大，所以只有λ的数量级与训练集相当时才会对结果产生影响，否则产生的影响不会很大。

然而特征值的个数对结果的影响比较明显，随着特征值个数的增加，各项指标越高。但是由表中可以看到，当个数增加至105时，其F1就已经到达了0.96左右。个数增加至500时，F1指标也仅仅到达了0.97左右。因为信息增益较大的特征词已经参与评价，再添加更多的特征词，其带来的效果也不会提高很多。同时随着个数的增多，运行速度也在下降，所以适当选取特征词个数是很有必要的。