朴素贝叶斯分类器项目报告

蒋旗志

1. 互信息选取特征词

在构造贝叶斯文本分类器时，首先需要确定词向量包括哪些词项。直观上，如果某一个词出现的频率越高，这个词就越重要。但有些词（如"的、是、在、了"等停用词）在所有的类别中出现的频率都很高，且在所有的文本类别中出现的频率相对稳定。这样的词对于区分不同类别的特征就没有太大意义，这类词的出现对‘它是否属于某一类别 c’带来的信息量很少。一般来说，我们希望所选择的词项能反出某一特定类别的特征，即这个词项的出现能给它是否属于某一类别 c 带来尽可能多的信息量。我们可以用互信息 MI（Mutual Information）度量某个词项 t 的出现为它属于类别 c 贡献的信息量。具体地，就是计算文档中词项 t 出现与否与文档是否属于类别 c 这两个随机变量的互信息。它们之间互信息的公式如下：

$$MI(U;C)=\sum_{e_t\in\{0,1\}}\sum_{e_c\in\{0,1\}}P\{U=e_t,C=e_c\}\log_2\left\{\frac{P\{U=e_t,C=e_c\}}{P\{U=e_t\}P\{C=e_c\}}\right\},$$

其中 U 是一个取值在{0,1}中的随机变量，表示词项 t 是否出现，e_t 为 1 时表示文档中出现词项 t，为 0 表示文档中未出现词项 t。类似地，C 也是一个取值在{0,1}中的随机变量，表示文档是否出现属于类别 c，e_c 为 1 时表示文档属于类别 c，为 0 表示文档不属于类别 c。

我们用极大似然法估计公式中出现的概率值：

$$P\{U=e_t,C=e_c\}=\frac{N_{e_t,e_c}}{N},$$
$$P\{U=e_t\}=\frac{N_{e_t,\bullet}}{N},$$
$$P\{C=e_c\}=\frac{N_{\bullet,e_c}}{N}.$$

其中 N 表示文档总数，N_{e_t,e_c} 表示类别为 c（或不为 c，即当 e_c 为 0 时）且包含词项 t（或不包含词项 t，即当 e_t 为 0 时）的文档的个数。$N_{e_t,\bullet}$ 表示包含词项 t（或不包含词项 t，即当 e_t 为 0 时）的文档的个数，而 $N_{\bullet,e_c}$ 表示类别为 c（或不为 c，即当 e_c 为 0 时）的文档的个数。

将这些估计的概率值代入互信息定义，得到计算互信息的公式：

$$MI(U;C)=\frac{N_{1,1}}{N}\log_2\left(\frac{NN_{1,1}}{N_{1,\bullet}N_{\bullet,1}}\right)+\frac{N_{1,0}}{N}\log_2\left(\frac{NN_{1,0}}{N_{1,\bullet}N_{\bullet,0}}\right)$$
$$+\frac{N_{0,1}}{N}\log_2\left(\frac{NN_{0,1}}{N_{0,\bullet}N_{\bullet,1}}\right)\frac{N_{0,0}}{N}\log_2\left(\frac{NN_{0,0}}{N_{0,\bullet}N_{\bullet,0}}\right),$$

2. 朴素贝叶斯分类器

2.1 最大后验概率

朴素贝叶斯分类器（Naïve Bayes Classifier）是一种有监督机器学习方法，在构造分类器的过程中使用了贝叶斯公式与条件独立假设。具体地，我们对文档 d 与类别 c 运用贝叶斯公式：

使用的文档表示方法为词袋模型表示（Bag of words　representation）

$$P\{\mathrm{c}\,|\,\mathrm{d}\}=\frac{P\{\mathrm{c}\,|\,\mathrm{d}\}P\{\mathrm{c}\}}{P\{\mathrm{d}\}}$$

这里 P(c|d) 称之为后验概率。直观上，d 出现在哪一类的可能性大，我们就倾向于将 d 分到哪一类中去。可以用后验概率 P(c|d)衡量 d 出现在每一类 c 中的可能性，给出一个文档的条件下，我们选择将 d 分到使后验概率 P(c|d)最大的那个类别 c_{MAP} 中去。基于上述规则做出的分类结果叫做最大后验概率类（maximum a posteriori MAP）。然而，在实际应用中我

们并不需要计算文档 d 属于每一类 c 的后验概率值，然后选择后验概率值最大的那个类作为预测类。我们关心的只是 d 属于各个不同类后验概率值的相对大小，由贝叶斯公式可知，每个后验概率同乘以 P(d) ,转化成 P(d|c)*P(c)后后验概率值的相对大小不变。也就是说，我们选出的最大后验概率类 c_{MAP} 与按下式选出的类相同：

$$c_{MAP} = \arg\max_{c \in C} P\{d \mid c\}P\{c\}$$

2. 2 条件独立假设

在计算似然概率 P(d|c)时，首先把文档 d 表示成一些特征 x_1，x_2，…，x_n。然后对每一个类别 c，我们需要估计如下概率值：

$$P\{x_1, x_2, \cdots, x_n \mid \mathrm{c}\},$$
$$\mathrm{P}\{\mathrm{c}\}.$$

换句话说，采用直接估计联合概率的方法至少需要估计 O（$|X|^n*|C|+|C|$）个参数。这只有在海量训练样本的情况下才能保证训练出来的模型不会过拟合。为了减少模型参数，使模型更实用，我们必须做出一些简化模型的假设（尽管这些假设并非十分符合实际情况）。

朴素贝叶斯模型采用了如下假设减少参数：

- 词袋模型假设：假设词在文档中出现的位置不重要

- 条件独立假设：假设在给定类别 c 的条件下，特征概率 $P(x_i|c_j)$之间是相互独立的，即有：

$$P\{x_1, x_2, \cdots, x_n \mid \mathrm{c}\} = P\{x_1 \mid \mathrm{c}\} \bullet P\{x_2 \mid \mathrm{c}\} \bullet \cdots \bullet P\{x_n \mid \mathrm{c}\}$$

2.3 拉普拉斯平滑

接下来需要估计每个特征词的似然概率 $P(x_i|c_j)$，一个直观的想法就是用特征词在每个类别出现的频率作为概率的估计，也就是 $P(x_i|c_j)$的极大似然估计：

$$\hat{P}\{x_i \mid \mathrm{c}_j\} = \frac{count(x_i, \mathrm{c}_j)}{\sum_{x_m \in V} count(x_m, \mathrm{c}_j)}$$

上式分母为 j 类文档单词总数，分子为第 i 个特征词在 j 类文档中出现的次数

这种估计最大的问题是如果某个特征词的概率的估计为零，则不管前面出现的词使判定某个类的概率多高，则最后计算出来的属于此类的后验概率为零。比如说，我们根据脚掌形状，脖子长度，腿长，羽毛颜色等一系列特征来判断一个动物是不是天鹅，则最大释然估计得出的分类完全是基于已经观察到的数据的，设想用最大似然估计计算出的概率判断一只动物是不是天鹅，根据脚掌形状，脖子长度，腿长，等一系列特征来看它属于天鹅的概率都很高，但仅仅因为它的羽毛为黑色而已经观察的数据中没有羽毛为黑色的天鹅就足以使它为天鹅的后验概率为 0.这显然是不合理的。

一种改进方法是每个词在计数时都多加 1 以避免出现某个特征概率为零的情况，这种方法称之为拉普拉斯平滑。事实上，我们可以在每个词计数时多加大于 0 的量α。

到目前为止，我们已经确定了模型训练阶段的需要计算的量有：

$$\hat{P}\{c_j\} = \frac{\#doc_j}{\#doc}, \mathrm{j} \in \{1, \cdots, |\mathrm{C}|\},$$
$$\hat{P}\{x_i \mid \mathrm{c}_j\} = \frac{n_i + \alpha}{N_j + \alpha |V|}, \mathrm{i} \in \{1, \cdots, \mathrm{n}\}, \mathrm{j} \in \{1, \cdots, |\mathrm{C}|\}.$$

先验概率的估计为某类文档数比总文档数，特征概率估计为某个特征词在某类出现次数加上平滑参数比上此类文档单词总数加上 n 倍平滑系数。|V|为特征的数量，此处为 n，之所以乘以|V|倍可理解为每个特征词不管出现与否都会有一个α，一共有|V|个特征词乘以|V|。

2.4 取对数避免下溢

在实际执行中因为考虑到浮点数下溢问题将先验概率的连乘取对数变成和式，由于我们只关心概率的相对大小，且对数函数是单调的，因此保持概率值之间相对大小关系。

3. 程序思路及运行结果

3.1 数据预处理

数据预处理主要是通过 dirTextToLargetext(dir, largefile),joinFile(filename),translate(bytesstr),findLabel(filename)四个函数完成。dirTextToLargetext(dir, largefile)将指定目录下的所有文本文件（连同标签数据）转移到一个大文件中，转移后的大文件中每一行对应于原始数据一个文档，且行首为标签信息，标签信息与文本信息之间有一个空格，利用文本信息 translate(bytesstr)函数把所有分中文字符去掉。dirTextToLargetext(dir, largefile)调用 joinFile(filename)和 findLabel(filename)函数，joinFile(filename)将指定文件变成去除其它字符后（通过调用 translate(bytesstr)函数）的长字符串，findLabel(filename)函数通过文件名找到文件所属标签。

在数据预处理过程中发现有一个不属于其它 10 个类别的 Military 类文件,简化处理起见，将其从原始数据中删除。

3.2 选择特征词

本程序通过计算每个词对每个类别的互信息选择特征词，选择过程主要由 readDataIntoDict(largefile),buildClassWorddictandCompuNdot(bigdict),compuMI(cla

ssworddict, ndotcx, ntxdot, n)和 selectFeatureword(mi, filename, featurenum)四个函数完成。readDataIntoDict(largefile)以数据预处理步合并好的文件的文件名为参数，返回一个字典 bigdict，这个字典以文档类别为键，值为此类文档组成的列表，列表每个元素为一个由某个原始文件转化而来的字符串。buildClassWorddictandCompuNdot(bigdict)已上个函数返回的字典为参数，返回 classworddict, ndotcx, ntxdot，n 四个参数，classworddict 是一个双重字典，用来保存每个词语在每个类别中出现的文档数（也就是公式中的 N_{11}）ndotcx, ntxdot 两个字典分别用来保存某类文本的文档数与包含某个特征词的文档数，n 为文档总数。compuMI(classworddict, ndotcx, ntxdot, n)用前面返回的结果计算所有词对所有类别的互信息，它返回一个双重字典 mi 保存这些互信息。selectFeatureword(mi, filename, featurenum)以计算出的互信息 mi 为参数，将每个类别互信息最大的前 featurenum 个词写入到 filename 文件中去。

从输出特征词的结果来看互信息是一种较为可靠的方法，例如 Politics 类选出的特征词为“总统 访问 两国 主席 友好 会见 会谈 外交 关系 外长 总理，...”

3.3 模型训练与预测

模型训练与预测部分由 readDataIntoList(largefile)， readFeaturewordtoset(filename)，listtoDict(trainlist)， naiveBayes(traindict, priors,featurewords,alpha)，bayesPredict(testlist, featurepro, priors, featurewords,)， crossValidate(datalist, turns_num)完成。前三个函数用于读入经过预处理的数据为方便处理与训练的格式，naiveBayes(traindict, priors,featurewords,alpha)用来训练模型，其中 alpha 是平滑系数，featurewords 是用某种方法选出的特征词的集合。bayesPredict(testlist, featurepro, priors, featurewords,)用来测试模型，eaturepro, priors 为训练后得出的模型似然概率先验概率的对数值。

crossValidate(datalist, turns_num)用来对模型进行交叉验证，turns_num 为验证的折数。

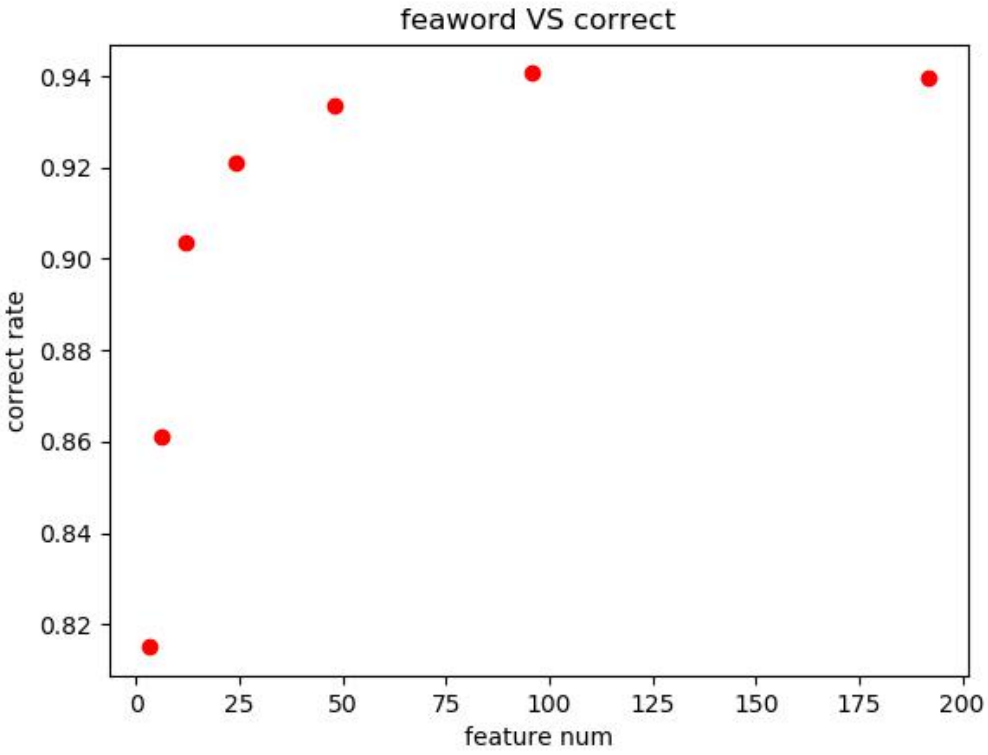
3.4 运行结果

在选择平滑系数为 1，交叉验证折数为 5 的情况下模型平均正确率为 94.1%，交叉验证折数为 7 时模型平均正确率为 94.4%，交叉验证折数为 10 时，平均正确率为 94.3%

在选择平滑系数为 0.1，交叉验证折数为 5 的情况下模型平均正确率为 94.6%，而取平滑系数为 10，平均正确率为 92.4%

此外，在模型输出中还计算了混淆矩阵，以此为基础计算模型在宏平均以及微平均下的召回率与准确率。在观察输出时发现宏平均的两个指标在通常情况下总是小于微平均的对应指标，这可能是在各类样本数量不相同时的一个普遍现象。

从下图我们可以看出，朴素贝叶斯模型对特征词的稳定性相当好，只需要很少的特征词就可以获得相当高的准确率，且几乎不受不相关特征词的干扰（后面项目我们会看到，logistic 回归模型对特征词个数十分敏感）。



2017.10.16