使用朴素贝叶斯分类器进行文本分类[实现说明]

YuDianhai@gmail.com

2014-6-18

回顾朴素贝叶斯（NB）分类器：

$$p(y_k|x) = \frac{p(y_k)p(x|y_k)}{p(x)} \propto p(y_k)p(x|y_k) = p(y_k)\prod_{i=1}^{d} p(x_i|y_k)$$

对于文本分类任务，即对一篇文章进行分类，是 NLP 中最常见的机器学习任务。一般情况下，类别从几个到几十不等，或者更多。使用 NB 分类器进行文本分类，我们需要首先考虑特征是什么，即x如何表示；$p(x_i|y_k)$的物理意义是什么，如何计算。

对于特征方面，文本分类常规都是使用 bag-of-words 的特征，即以文章中出现的词作为特征，而不考虑词语出现的顺序。NB 分类器也一般使用这种形式。那么特征空间的大小，就取决于词表（vocabulary）的大小，即语料集合中不重复词的个数。对于汉语来说，一般几万到百万不等。

在 bag-of-words 的特征体系下，特征空间是确定了的，但是具体x_i的取值以及对应的$p(x_i|y_k)$的物理意义却可以有不同的考虑，对应着不同的参数计算公式及分类器训练和预测的实现。这取决于我们是否考虑词语在文章中出现的频次。

一、伯努力（Bernoulli）NB

先看不考虑词频的情况。即只看某词语在某文章中是否出现，而不管出现了具体是多少次。这种假设下，每维特征的取值为 0-1，此时对应的 NB 分类器又被称为**伯努力（Bernoulli）NB 分类器**。

比如，如果词表是{2014、年、巴西、世界杯、足球赛、举行、是、第、20、届、球队}，某文档是“2014 年巴西世界杯足球赛是第 20 届世界杯足球赛”，那么特征空间是 11，该文档特征向量是：

$$\mathrm{x} = (1,1,1,1,1,0,1,1,1,1,0)$$

此时，$p(x_i|y_k)$的物理意义可认为是：若文章为 k 类别，则第 i 特征（词表第 i 个词）出现或者不出现的概率。那么：

$$p(x_i = 0|y_k) = 1 - p(x_i = 1|y_k)$$

习惯的，我们经常用$p(x_i|y_k)$来作为$p(x_i = 1|y_k)$同等含义的一种表示。那此时，原 NB 模型的表达式可以写为：

$$p(y_k|x) \propto p(y_k)\prod_{i=1}^{d}[\delta_i p(x_i|y_k) + (1-\delta_i)(1-p(x_i|y_k))]$$

δ_i=1 表示第 i 个词在该文档中出现了，没出现则δ_i=0.

此时要非常注意，计算文章属于某个类别的得分的时候，不只要考虑该文章的 word，还要考虑在词表中的但是在该文章中没出现的 word！这类词对得分的贡献是$1 - p(x_i|y_k)$。因此伯努力 NB 下，分类的时间复杂度是 O(Cd)，C 是类别数，d 是词表大小。

当然可以有加速的策略。一般情况下，词表会比较大，而一篇文章中实际出现的不重复

词（记作 M）要少。所以可以先预先按所有词都没出现计算出一个定值，然后进行置换。

$$s_k = \prod_{i=1}^{d} 1 - p(x_i|y_k)$$

$$p(y_k|x) = s_k \prod_{j=1}^{M} \frac{p(x_j|y_k)}{1 - p(x_j|y_k)}$$

那么伯努力 NB 下，p的参数估计表达式是多少呢？假设根据如上定义，及最大似然估计，可以得到：

$$p(x_i|y_k) = \frac{\sum_{t=1}^{N} \delta(y^t = y_k)\delta(i\ in\ t)}{\sum_{t=1}^{N} \delta(y^t = y_k)} = \frac{\text{第 k 类文章中出现过第 i 词的文章数}}{\text{第}k\text{类文章数}}$$

可见，对于高频词，对应的这种条件概率是非常高的。比如“的”（假设没去除停用词），其对应的条件概率值很可能会接近于 1.

再重复强调一下，此时的概率意义约束是：

$$p(x_i = 1|y_k) + p(x_i = 0|y_k) = 1$$

看一下伯努力 NB 下参数平滑的问题。使用加 1 平滑，即拉普拉斯平滑，此时在保证概率意义下，其平滑公式应该为：

$$p(x_i|y_k) = \frac{\text{第 k 类文章中出现过第 i 词的文章数} + 1}{\text{第}k\text{类文章数} + 2}$$

提醒一下，此处分母加的值是 2，而不是词表大小。注意，平滑一定要使得平滑之后仍满足概率意义。

二、多项式（Multinomial）NB

当我们考虑文章内词语的频次，而不只是考虑出现或未出现，此时特征的取值不再是 0-1，不过总的特征空间大小仍未变化。拿前面的例子来做对照，词表是{2014、年、巴西、世界杯、足球赛、举行、是、第、20、届、球队}，某文档是“2014 年巴西世界杯足球赛是第 20 届世界杯足球赛”，此时该文章的特征向量为：

$$x = (1,1,1,2,2,0,1,1,1,1,0)$$

此时对应的 NB 一般称为多项式 NB。设 m 为文章内的总词频数，对应的模型表达式应该如下：

$$p(y_k|x) \propto p(y_k)p(x|y_k) = p(y_k)\frac{m!}{\prod_{i=1}^{d} x_i!}\prod_{i=1}^{d} p(w_i|y_k)^{x_i}$$

$$\propto p(y_k)\prod_{i=1}^{d} p(w_i|y_k)^{x_i}$$

之所以可以省掉这个多项式系数，是因为它是和类别y_k无关的。

而此时，$p(w_i|y_k)$的意义应该是，第 k 类别的所有文章中第 i 词的分布概率：

$$p(w_i|y_k) = \frac{\sum_{t=1}^{N} \delta(y^t = y_k)x_i^t}{\sum_{i=1}^{d}\sum_{t=1}^{N} \delta(y^t = y_k)\, x_i^t} = \frac{\sum_{t=1}^{N} \delta(y^t = y_k)x_i^t}{\sum_{t=1}^{N} \delta(y^t = y_k)\, m^t} = \frac{\text{第 k 类文章第 i 词总词频}}{\text{第}k\text{类文章内总词频}}$$

可见，多项式 NB 下，即使极高频词，其$p(w_i|y_k)$也很难接近于 1，另外其在模型中

作用的时候是：$p(w_i|y_k)^{x_i}$。

这时候的概率约束是：

$$\sum_{i=1}^{d} p(w_i|y_k) = 1$$

因此对应的加 1 平滑为：

$$p(w_i|y_k) = \frac{\sum_{t=1}^{N} \delta(y^t = y_k) x_i^t + 1}{\sum_{t=1}^{N} \delta(y^t = y_k)\, m^t + d}$$

提醒一下，此时分布加的值是词表大小！

再看一下预测的效率问题，直观上看仍然为 O(Cd)，不过在多项式 NB 下，可以更直观的简化：

$$p(y_k|x) \propto p(y_k) \prod_{i=1}^{d} p(w_i|y_k)^{x_i} = p(y_k) \prod_{j=1}^{l} p(u_j|y_k)$$

u^j表示第 j 位置的词语。这样文档内重复的词语，已经展开相乘，而没出现的词语$p(w|y_k)^0$本来就是 1，可以忽略掉。因此对于新文章，只要按照词语顺序扫描完毕，即可得到每个类别下的得分。实际预测时间复杂度为 O(Cm)，m 是总词频。

三、实际实现的一些其他注意事项

1、 训练时候对于词语平铺的文本，应该要做词的聚合，即行程 bag-of-words 的形式比较有利于后续统计计算，特别是对于伯努利 NB 必须做去重。当然，对于多项式 NB，也可以顺次扫描累加。

2、 预测时候的概率连乘，为了防止精度损失，可以改用取 log 相加。

3、 对于短一些的文本，伯努利 NB 即可；对于长文本，考虑词频的多项式 NB 即可。当然也可以使用 tf-idf 等特征值，仿照多项式 NB 的形式。

4、 预测时候，对于词表中出现但是本文章未出现的词语，伯努利 NB 下对得分有贡献，多项式 NB 下不用考虑；对于在词表中未出现的词，都可以不予以考虑，因为未登录词对各个类别的贡献是一样的。

四、实验

使用 C++程序实现基于 NB 的文本分类。

首先进行数据预处理，划分训练集和测试集（具体比例可以指定）。统计类别及词表。这时候可以搭配一下停用词表，去除一些高频无意义词。

训练时候实现了伯努利 NB 和多项式 NB 两种计算策略，并对应不同的预测策略。训练时候才用的 NB 类型在参数文件里有存储，预测时候可以自动判断。

在给定的一个三分类新闻语料上，按照 20%的比例随机划分测试集，初步实验结果如下（只给出整体 Accuracy）：

	使用停用词表	不使用停用词表
Multinomial NB	97.94%	97.47%
Bernoulli NB	91.46%	90.20%

运行环境输出示例如下：

```
processing data ....

training step .....
Dict infomation:
category: 3
word: 37623
Begin to train Multinomial NB..
Total documents number: 2433
p(y):
0.330045 0.340732 0.329223

testing step .....
Dict infomation:
category: 3
word: 37623
M
This is a Multinomial NB  model!
Accuracy:       97.4724
```