Spam filtering with Naïve Bayes - Which Naïve Bayes?

Authors

Vangelis Metsis, Ion Androutsopoulos, Greece Georgios Paliouras

Abstract

Naïve Bayes在之前是非常流行的一种检测spam email的方法，而且还是开源的。然而，有很多种Naïve Bayes版本，许多文献中也是没怎么提及的。作者呢，在这篇论文中讨论了5种不同版本的NB，然后在6个新的dataset中进行比较。Enron dataset是email dataset，包含约有150个users，主要是来自Enron的高级管理者的邮件，数据约有0.5M个。数据集中呢，包含了ham and spam email。但是作者自己弄了一个新的dataset，这个dataset是开源的，作者说这个新的dataset很真实，因为在时间顺序上包含了两大类，也就是ham and spam，然后在比例上也是很合适，一个dataset也就之包含一个人的messages。作者采用的实验过程可以使得这个spam filter更加个性化，能够更加个性化。作者画了ROC curve使得我们能够更好地去比价不同版本的NB

1 Introduction

尽管有几个ML的算法是应用在检测spam email上的，这些算法也是有很好的performance的，比如说Boosting, SVM等。之后，Naïve Bayes(NB) classifier就慢慢地变为了主流，不管是商业上还是开源的spam filter，都是非常流行的一个方法。究其原因，估计是简单且容易实现，线性时间复杂度且精确度又高。在众多的spam filters中，NB算是非常具有竞争力的。然而呢，有几种不同形式的NB，在许多的anti-spam 文献中都没怎么提及。

在之前的NB算法中，有两种，都是learning-based 的算法，一种是multi-variate Bernoulli model (这个的attributes是boolean形式的，就是看某个词出现了没有)，一种是multinomial form of NB (这个的attributes是term frequency形式的，就是看某个词出现了多少次)。这两者呢，之前也是有做过对比的，实现显示，multinomial NB比multivariate Bernoulli model要好。在后来是实验中发现，如果在multinomial FB这个model的数据的attributes给换成boolean的话，效果会出奇的好。

这个multi-variate Bernoulli NB，改一改之后也是可以用在连续型的数据上的(continuous attributes)，这也就产生了新的model，也就是multi-variate Gauss NB，这个model的assumption是每一个类别，每一个class中的每一个attribute都是服从normal distribution的 (也就是说每一个class，它的每一个特征的分布都是服从正态分布的)。另外，每一类的每一个attribute的分布都可以用几个不同的normal distribution的平均值来替代，每个normal distribution都是不一样啦，那么这个就是一个新的版本的NB，就是Flexible Bayes (FB). (其实就是说，一个特征中的数据，这一堆可以看做是一个分布，另外一堆可以看做是另一个分布，再把这个分布取个均值)。作者发现，如果attribute的值是$\frac{\text{term frequency}}{\text{document length}}$的话，FB的表现就会比multi-variate Gauss NB要好。之前工作中作者只比较了multivariate NB，没有跟其他版本的NB进行比较。

这篇论文呢，作者就向我们介绍了5种不同版本的NB，分别是：①Multi-variate Bernoulli NB，②Multinomial NB， TF attributes，③Multinomial NB，Boolean attributes，④Multi-variate Gauss NB，⑤Flexible Bayes. 作者也在6个新的Enron dataset中进行的评估。每一个dataset呢，都包含了一个user的ham messages，然后再加了一定比例的spam messages，这也使得dataset更加真实。作者认为这样的dataset并且这样训练的话能使得这个filter更为个性化，私人化。作者plot了ROC曲线，因为我们不知道取什么阈值比较好，有这么一个ROC，方便去比较true positive and true negative

2 Naïve Bayes classifiers

这里呢，就简单点来看，作者就只看message的文字内容，就不看header啊或者其他东西，就只看email里的内容。当然了，也可以把header那些作为一个attribute。我们可以用一个ensemble的方法，用几个不同的classifiers，一些train文字啊，一些train其他attributes之类的。

在作者的实验中，每一个message最终都是用一个vector，$< x_1, \cdots, x_m >$来表示的, 这里的$x_1, \cdots, x_m$分别是$X_1, \cdots, X_m$，attributes对应的值。之前也有人用n-gram来表示这些$x_1, \cdots, x_m$，但是发现行不通。这里呢，最简单的方式就是用boolean来表示，$X_i = 1$表示这个message有包含这个token，否则就是$X_i = 0$. 另外呢，也有用term frequency (TF)来表示的，就是在这一篇message中，这个token出现了多少次。如果使用TF的话，它携带的信息自然也就更多，我们可能会认为用TF作为attribute的values会更好，但其实不然。第三种方式是normalized TF，就是$\frac{\text{term frequency}}{\text{document length}}$. 举个例子来说为什么要normalized一下，"rich"在一篇两段长的message(大约是200-400词)中出现了三次，那么这个message有较大的可能是spam，但是在更长的message中，spam的可能性会降低。

作者呢，舍弃了一些不常见的词(就是至少了5个messages是没有出现这个词的，就给舍弃了)。即使做了一些舍弃，仍旧还有很多的词要做attributes，于是作者用information gain的方法进行排序，选出m个最好的来做attributes，作者用了m=500,1000,3000分别做了实验。这里我们需要知道，information gain对boolean attribute比较容易求，对non-boolean的话就比较难求(不过，我们上课时教授讲了怎么求，哈哈)。之前也有人用其他方法来对TF value的求information gain，但是效果也还是差不多。

从Bayes理论来看，给定一个message，$\overrightarrow{x} =< x_1, \cdots, x_m >$, 问是属于哪一个category c的。于是就可以用Bayes公式

$$p(c|\overrightarrow{x}) = \frac{p(c)p(\overrightarrow{x}|c)}{p(\overrightarrow{x})}$$

从公式来看，因为分母跟哪个类别无关，所以NB classifier是要最大化$p(c)p(\overrightarrow{x}|c)$, 在spam filtering的case中，上面的Bayes公式就跟下面的公式等价

$$p(c_s|\overrightarrow{x}) = \frac{p(c_s)p(\overrightarrow{x}|c_s)}{p(\overrightarrow{x})} = \frac{p(c_s)p(\overrightarrow{x}|c_s)}{p(c_s)p(\overrightarrow{x}|c_s) + p(c_h)p(\overrightarrow{x}|c_h)} > T$$

我们一般可能设置T=0.5，然后c_s, c_h分别代表的意思是spam和ham category。通过调整T的阈值大小，我们可以选择是要更多的准确预测ham，还是要更多的准确预测spam。

这里提一下，在这里，spam filter的case中，true positive代表的意思是真实值是spam，预测值也是spam。true negative的意思是，真实值是ham，预测值也是ham。false positive的意思是真实值是ham，但是预测值是spam。false negative的意思是真实值是spam，但是预测值是ham

在这里有一个先验概率priori probability $p(c)$就是属于c类别的message除以message的总是，$p(c) = \frac{\#c}{\#total}$. 然后$p(c)p(\overrightarrow{x}|c)$的计算方式是每个版本都不一样，下面就看看如何计算吧。

2.1 Multi-variate Bernoulli NB

我们的dataset中，一共有m的attributes，让我们设定$F = \{t_1, \cdots, t_m\}$就是这些tokens(其实就是哪些词)，这m个也是用刚刚说的方法进行选择过的。这个multi-variate Bernoulli NB就是把每一个message d当做是一系列的tokens来看的，每一个t_i表示的就是d中出现的词。因此呢，d可以用一个binary vector $\overrightarrow{x} =< x_1, \cdots, x_m >$来表示，就是看这些词有没有在d中出现，有就是1，没有就是0. 这也局是multi-variate Bernoulli NB中的数据就是这样的。还有就是，每一个类别 c 的message d就可以看做是做了m次的Bernoulli trials。每一次实验，我们都看t_i表示的这个词有无出现在d中。然后每一个词出现的概率就是$p(t_i|c)$. 这个multi-variate Bernoulli NB需要假设在给定category的情况下，每一次实验的结果都是相互独立的。这也是NB需要进行的假设。很Naïve。因为词的出现说是independent是不大现实的。之后所有的NB都需要做这样相似的假设。尽管很naïve，但是分类效果很好。于是，我们可以得到

$$p(\overrightarrow{x}|c) = \prod_{i=1}^{m} p(t_i|c)^{x_i}(1-p(t_i|c))^{(1-x_i)}$$

于是判定这个message是否是spam的公式就变成了

$$\frac{p(c_s)\prod_{i=1}^{m} p(t_i|c_s)^{x_i}(1-p(t_i|c_s))^{(1-x_i)}}{\sum_{c\in\{c_s,c_h\}} p(c)\prod_{i=1}^{m} p(t_i|c)^{x_i}(1-p(t_i|c))^{(1-x_i)}} > T$$

这里我们就想，那每个词出现的概率$p(t|c)$怎么求呢？我们在计算概率的时候得加上Laplacian smoothing，防止说某个词在词库中没有出现的情况，计算概率的公式如下

$$p(t|c) = \frac{1+M_{t,c}}{2+M_c}$$

这里，$M_{t,c}$就是指类别c中包含token t 的message有多少个。M_c代表的意思就是类别c一共有多少个。举个例子来说，假设apple一共在50个spam email都有出现，然后spam email的数量是300, 然后在70个ham email里都有出现，ham email一共有500篇，那么，$p(t_{apple}|c_s) = \frac{50}{300} = \frac{1}{6}$，$p(t_{apple}|c_h) = \frac{70}{500} = \frac{7}{50}$.

2.2 Multinomial NB, TF attributes

这里的Multinomial NB，它的特征是term frequency，我们把每一个message d都看做是一个很多tokens的组成，当然了，这个token是可以重复多次的。所以呢，d就可以用vector $\overrightarrow{x} =< x_1, \cdots, x_m >$来表示.$x_i$表示的是这个token在d中出现了多少次。每一个message的组成可以说是相当于从一个很大的bag，从中有放回地去拿这个词。用|d|来表示这个message的length。要组成这个message，就要有放回地拿|d|次。对于每一个token，它的概率是$p(t_i|c)$.这是一个多项式分布，multinomial distribution，所以，给定某个类别，要组成一条特定的message d的概率是

$$p(\overrightarrow{x}|c) = p(|d|)|d|!\prod_{i=1}^{m}\frac{p(t_i|c)^{x_i}}{x_i!}$$

我们先来解释下这个公式。假设这个message的长度是|d|，也就是说一共有d个位置，然后某个token t_1出现了x_1次，也就是说|d|中有x_1个位置是属于token的，那么它的排列方式就有$C_d^{x_1} = \frac{|d|!}{x_1!(|d|-x_1)!}$. 等下一个token t_2的时候，这个token出现了x_2次，那么就只剩下$|d| - x_1$个位置可以给t_2放了，于是就有$C_{d-x_1}^{x_2} = \frac{(|d|-x_1)!}{x_2!(|d|-x_1-x_2)!}$. 所以，所有的m个词放在一块组成message d的方式一共有

$$\frac{|d|!}{x_1!(|d|-x_1)!} \times \frac{(|d|-x_1)!}{x_2!(|d|-x_1-x_2)!} \times \cdots \times \frac{(|d|-x_1-\cdots-x_{m-1})!}{x_m!0!} = \frac{|d|!}{\prod_{i=1}^{m} x_i!}$$

然后我们再从词袋中，bag of word中抽出x_i个token t_i概率算上去，这也就是上面公式的由来。

从公式我们可以看到，有个message d的长度在那，所以，我们需要假设|d|跟类别是无关的。但是这个假设是稍微有点小小的问题的。因为毕竟每封邮件的长度都是不一样的。

现在，这个判别式就变成了

$$\frac{p(c_s)\prod_{i=1}^{m} p(t_i|c_s)^{x_i}}{\sum_{c\in\{c_s,c_h\}p(c)} \prod_{i=1}^{m} p(t_i|c)^{x_i}} > T$$

然后这里的概率$p(t|c)$的概率计算就是，当然了，得考虑词没有出现的情况，所以得加上Laplacian smoothing，计算公式就是

$$p(t|c) = \frac{1+N_{t,c}}{m+N_c}$$

这里呢，$N_{t,c}$的意思就是在c类别的信息中，这个token一共出现了多少次。假设一共有50封spam邮件，然后每封中都出现了3个apple这个词，那么，这个$N_{apple,spam} = 3 \times 50 = 150$. 然后$N_c = \sum_{i=1}^{m} N_{t,c}$。就是累加起来。

2.3 Multinomial NB, Boolean attributes

这个Multinomial NB with boolean attributes的计算方式跟Multinomial NB TF attributes是差不多的，$p(t|c) = \frac{1+N_{t,c}}{m+N_c}$,每个token的概率计算方式也是一样的。判别式那些都是一样的。它跟multi-variate Bernoulli NB的区别是不计算$(1 - p(t_i|c))^{(1-x_i)}$. 还有一个区别是Laplacian的方式也不一样。就这两个区别。

我们可能会感到有点奇怪，就是为什么Multinomial NB with boolean attributes的结果会比Multinomial NB with TF attributes的效果要好，TF包含的信息肯定比boolean多。有人解释说，multinomial NB with TF attributes跟 NB with attributes that follow Poisson distribution in each category是等价的，就是说，用NB，然后它的每一个分类的特征都要服从Poisson distribution，这个方式就跟multinomial NB with TF attributes是一样的。所以，如果message 的长度是跟类别无关的话，也就是我们的assumption，那么，如果TF在现实中是不服从Poisson distribution的话，multinomial NB with boolean attributes的性能就会表现更佳。

2.4 Multi-variate Gauss NB

这个Multi-variate Bernoulli NB的attributes是boolean values。但是其实也是可以改成real-valued的，前提我们得assume每个特征是都follow normal distribution的 $g(x_i; \mu_{i,c}, \sigma_{i,c})$ in each category. 计算方式就是

$$g(x_i; \mu_{i,c}, \sigma_{i,c}) = \frac{1}{\sigma_{i,c}\sqrt{2\pi}} e^{-} \frac{(x_i - \mu_{i,c})^2}{2\sigma_{i,c}^2}$$

然后我们是可以计算出mean $\mu_{i,c}$的和deviation的 $\sigma_{i,c}$. 当然了，我们还是得有一个假设，就是每一个特征都是相互独立的，现在我们就可以算概率了

$$p(\overrightarrow{x}|c) = \prod_{i=1}^{m} g(x_i; \mu_{i,c}, \sigma_{i,c})$$

然后判定是否是spam message的公式就是

$$\frac{p(c_s) \prod_{i=1}^{m} g(x_i; \mu_{i,c_s}, \sigma_{i,c_s})}{\sum_{c \in \{c_s, c_h\}} p(c) \prod_{i=1}^{m} g(x_i; \mu_{i,c}, \sigma_{i,c})} > T$$

这也就意味着我们可以使用normalized TF attributes。normalized之后，我们只使用正数的值，不使用负数的值，当然了，这也是一个问题，因为这样又不符合normal distribution了。

2.5 Flexible Bayes

那Multi-variate Gauss NB出现的问题应该怎么解决呢？于是就有了Flexible Bayes, 方法就是在同一个特征中，找出多个不同的normal distribution，这样就会有不同的mean values，但是会有相同的deviation，然后对于这些个mean value取均值，概率公式如下

$$p(x_i|c) = \frac{1}{L_{i,c}} \sum_{l=1}^{L_{i,c}} g(x_i; \mu_{i,c,l}, \sigma_c)$$

$L_{i,c}$的意思就是有多少个不同的normal distribution。然后deviation用相同的值，$\sigma_c = \frac{1}{\sqrt{M_c}}$. M_c就是属于c类别的message一共有多少个。这样子就有效缓解了Multi-variate Gauss NB的问题。虽然说FB这个方法能够更加real-valued的分布，但是违反了我们的assumption，就是这个特征的数据总体是normal distribution的。

所有以上五种方法的训练时间复杂度都是$O(mN)$.N是指一共有多少条message，m是指有多少个特征。分类的时间复杂度是前四个为$O(m)$,FB是$O(mN)$,因为要算L_i distribution，多了求和这一步。

3 Datasets and methodology

4 Experimental results

4.1 Size of attribute set

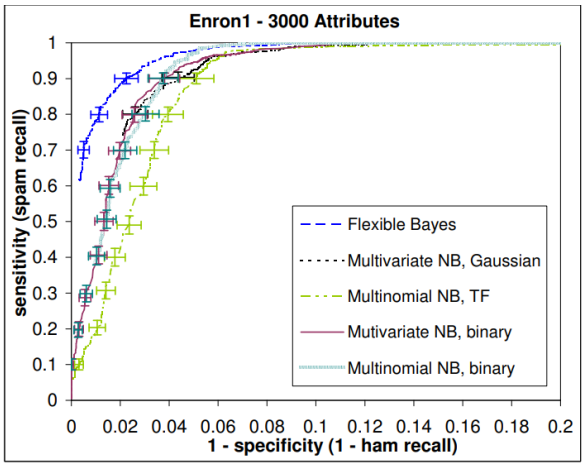
4.2 Comparison of NB versions

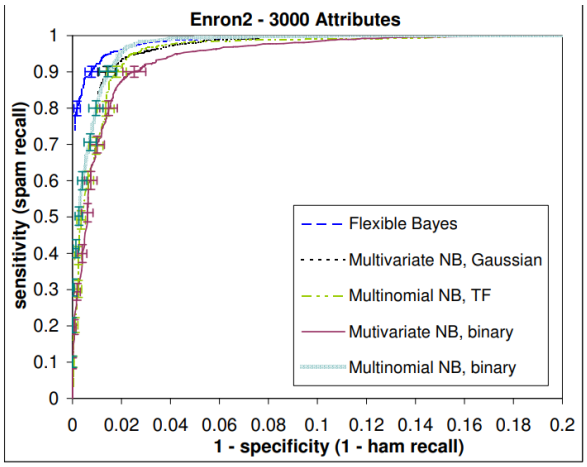
这里先说一些recall，这里呢，spam定为positive，ham定为negative。spam recall ($\frac{TP}{TP+FN}$),然后ham recall就是 ($\frac{TN}{TN+FP}$). 什么意思嘞？以spam recall来说，就是在所有分类为spam的message中，真正分类正确的比例。recall其实就是求全率，希望把所有的spam都给找出来。spam recall和spam misclassification rate互补。为了评估不同的NB分类器，作者画了ROC curve，我们可以看不同阈值的情况下分类情况是怎样的。所以，y轴就是sensitivity (spam recall), x轴就是 1-specificity (ham recall的互补，就是ham misclassification rate).

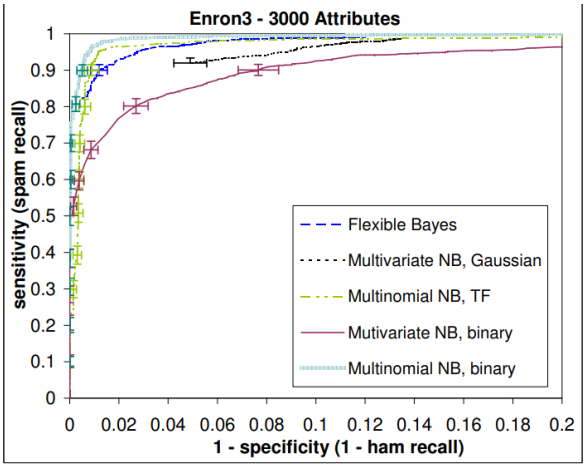
我们还得来看一下这六个dataset的混合比例情况，如下表

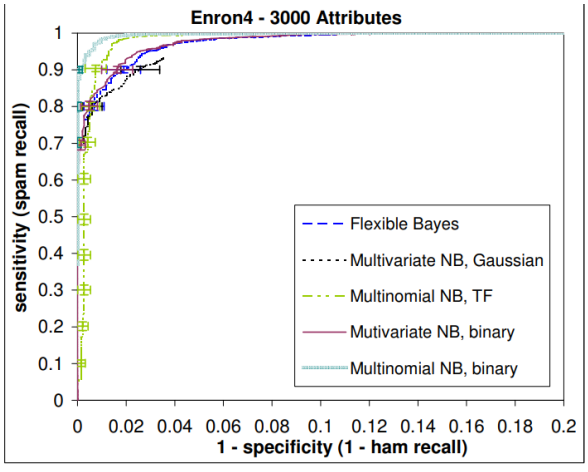
ham + spam	ham:spam	ham, spam periods
farmer-d + GP	3672:1500	[12/99, 1/02], [12/03, 9/05]
kaminski-v + SH	4361:1496	[12/99, 5/01], [5/01, 7/05]
kitchen-l + BG	4012:1500	[2/01, 2/02], [8/04, 7/05]
williams-w3 + GP	1500:4500	[4/01, 2/02], 12/03, 9/05]
beck-s + SH	1500:3675	[1/00, 5/01], [5/01, 7/05]
lokay-m + BG	1500:4500	[6/00, 3/02], [8/04, 7/05]

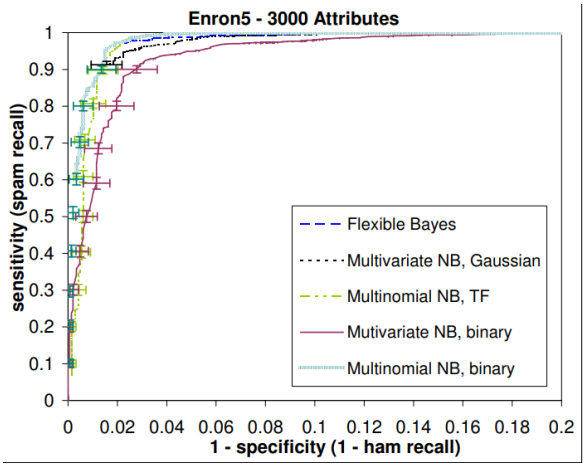
下图就是在6个不同的数据集中进行对比5个不同的NB classifiers。

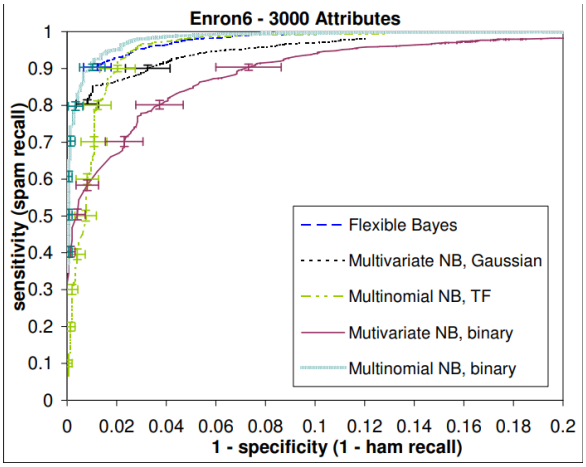












我们看这个curve，希望呢，spam recall要高但是ham misclassification rate要低，所以线偏左上角的就越好。我们从这个六张图中可以看出，总体上来讲multinomial NB with binary attributes是最好的，因为在其中4个dataset中，这个版本的NB都是最好的。而且，在第一第二个dataset中，multinomial NB with binary attributes跟最好的(FB)差距也不大，不能说很差。一般来说，NB-based spam filters，我们可能使用的方法是multinomial FB with TF attributes or multi-variate Bernoulli with binary attributes. 从这六张图中，我们可以看出，multi-variate Bernoulli with binary attributes基本上是好差的。

作者也对比了FB和multi-variate Gauss NB，发现呢，FB效果比较好。

现在目前比较好的就是FB和multinomial NB with binary attributes，但就复杂度来说，还是multinomial NB with binary attributes比较好。作者也为我们总结了5个版本的ham recall和spam recall的表，如下

NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6	Avg.
FB	90.50	93.63	96.94	95.78	99.56	99.55	95.99
MV Gauss	93.08	95.80	97.55	80.14	95.42	91.95	92.32
MN TF	95.66	96.81	95.04	97.79	99.42	98.08	97.13
MV Bern.	97.08	91.05	97.42	97.70	97.95	97.92	96.52
MN Bool.	96.00	96.68	96.94	97.79	99.69	98.10	**97.53**

Table 4: Spam recall (%) for 3000 attributes, $T = 0.5$.

NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6	Avg.
FB	97.64	98.83	95.36	96.61	90.76	89.97	94.86
MV Gauss	94.83	96.97	88.81	99.39	97.28	95.87	95.53
MN TF	94.00	96.78	98.83	98.30	95.65	95.12	96.45
MV Bern.	93.19	97.22	75.41	95.86	90.08	82.52	89.05
MN Bool.	95.25	97.83	98.88	99.05	95.65	96.88	**97.26**

Table 5: Ham recall (%) for 3000 attributes, $T = 0.5$.

我们会发现multinomial NB with binary attributes就是最好的。

4.3 Learning curves

5 Conclusions and future work

作者呢，为我们介绍了5个不同的NB，结果比较好的是FB和multinomial NB with binary attributes，这两个在其他文献中是比较少提及的。最好的是multinomial NB with binary attributes。