10601 Machine Learning HW8

Mingyang Song

Did you *receive* any help whatsoever from anyone in solving this assignment? No.

Did you *give* any help whatsoever to anyone in solving this assignment? No.

Q1. Native Bayes (15 Points)

1.**Is the following statements True/False? Explain your reasoning.**

False. The naive Bayes classifier is based on the simplifying assumption that the feature values are conditionally independent. Conditionally independence is different from independence which means even if some features are independent they may be not independent given by a third. So we cannot use naive Bayes just based on independent features.

2.**Using a simple Naive Bayes approach, how would we classify this, Chinese Restaurant or Western-style Restaurant?**

V: CR, WR

A: Spicy, Beef, Hot, Fried, Fish, Cooked, Potato

(a)To calculate prior:

P(V=CR)=0.4 P(V=WR)=0.6

As we assume that the features are conditionally independent,

First Way(actually I am not very clear about the difference, I thought it's just two ways to get probs):

P(A=Fish|V=CR)=1/9   P(A=Fish|V=WR)=2/11

P(A=Cooked|V=CR)=0   P(A=Cooked|V=WR)=2/11

P(A=Beef|V=CR)=2/9   P(A=Beef|V=WR)=1/11

Second Way(I hope that you could write a feedback for my confusion):

P(A=Fish|V=CR)=1/2   P(A=Fish|V=WR)=2/3

P(A=Cooked|V=CR)=0   P(A=Cooked|V=WR)=2/3

P(A=Beef|V=CR)=1/2   P(A=Beef|V=WR)=1/3

(b)classify the new item:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid Fish, Cooked, Beef) = \arg\max_{v_j \in V} P(Fish, Cooked, Beef \mid v_j) P(v_j)$$

Under the assumption,

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$ First Way:

P(CR)P(Fish|CR)P(Cooked|CR)P(Beef|CR)=0

P(WR)P(Fish|WR)P(Cooked|WR)P(Beef|WR)=24/11*11*11*10

Second Way:

P(CR)P(Fish|CR)P(Cooked|CR)P(Beef|CR)=0

P(WR)P(Fish|WR)P(Cooked|WR)P(Beef|WR)=4/45

(c)we should classify this into WR.

Q2. Naive Bayes Classifier for Political Blogs

2.3 Implementation of Naive Bayes (Total points: 45)

(b)

split	Classification accuracy
1	0.75
2	0.5
3	0.66667
4	0.75
5	0.75
6	0.5
7	0.58333
8	0.66667

9	0.75
10	0.66667

Mean: 0.658334

Standard Deviation: 0.09976

(c) using split1.train and split1.test

P(w|lib)	the to of and a in that is for on i it this by with as you at be was
P(w|con)	the to of and a in that is for it i on by this was with as be have not

The two lists are different in deed. However, most of the words are the same and there are great many overlapping words. We can see that these words are prepositions, prons and conjunctives which are frequently used by all kinds of blogs.

(d) using split1.train and split1.test

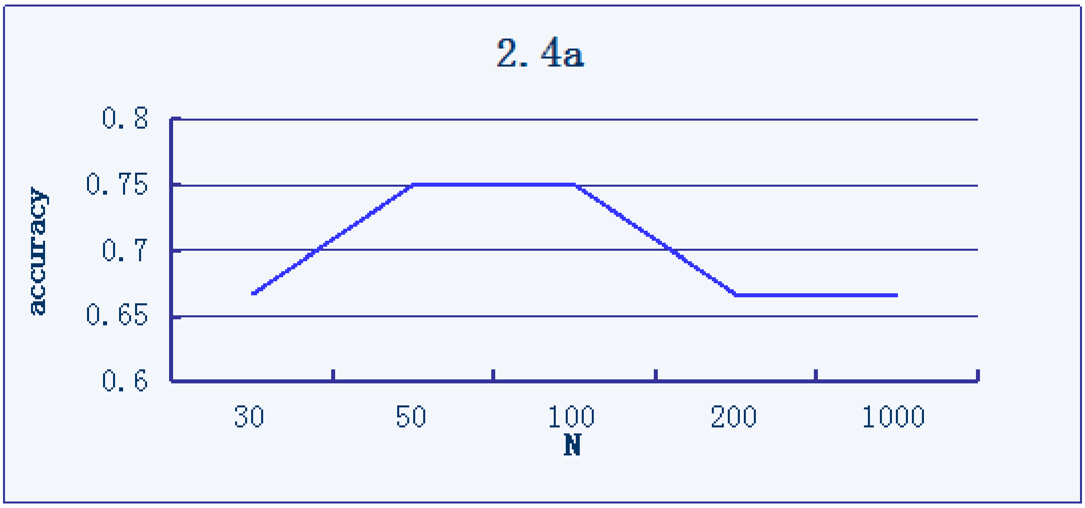
$\log \frac{p(w \mid C_{lib})}{p(w \mid C_{con})}$	rittenhouse digby 2002-2006 pst cesca _ brooklynkat tristero goldfarb dday kleiman mathew contrapositive stumble theatre pinky tex liberaloasis tx mcewan
$\log \frac{p(w \mid C_{con})}{p(w \mid C_{lib})}$	kyer discerningtexan galloway ridenour loading leftism jacoby tags spinoff pmemail domenico ellison batr morrissey hostages dsa _____ chechen bookmark almasi

There are no overlapping words in the two lists. These words are very rear and unique for specific blogs and seldom used by other blogs, while the words form previous question are very commonly used.

2.4 Feature Selection (Total points: 20)

(a) using split10.train and split10.test

N	Classification accuracy
30	0.66667
50	0.75
100	0.75
200	0.66667
1000	0.66667



We can see from the plot that the average accuracy goes up from 0.66667 to 0.75 from the beginning and than keep 0.75 and finally goes down to 0.66667. However, the range of accuracy is not large.

It means that to simply exclude the N frequent words from V will not make very bad influence on the accuracy except only two facts:

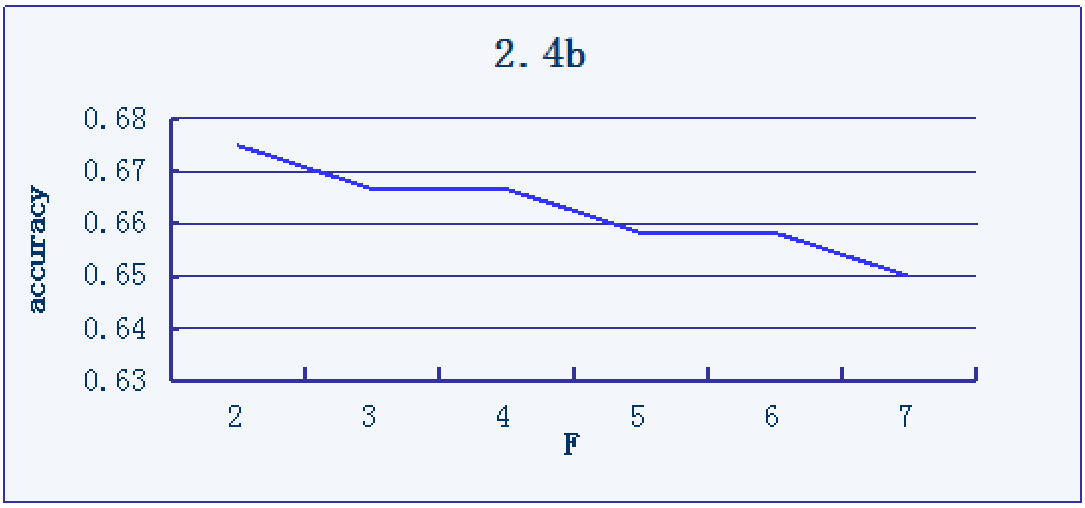
- key words which can be used to label the blogs are in the N set.
- a lot of noisy words are in V-N set which means that these noisy words will have a worse influence on the calculation of (vj)p(ai|vj) because there are no frequent words like "the", "is" to smooth the influence of each word.

Because usually for the first like 30 frequent words, they just occur in every blog without any key words for classification. Therefore, when we remove top 30, there is an increase.

Once N is not so big and keep a high accuracy, we can easily choose a proper N = 50 to make feature selection.

(b) N=50

F	Average Classification accuracy
2	0.6750
3	0.6667
4	0.6667
5	0.6583
6	0.6583
7	0.6500



We can see from the plot that as the F goes up the accuracy goes down gradually. This is because compared to N, F will have greater influence on the key words in the blogs. Some words are important for blog classification even though they only occur once or twice. The more the F becomes, the higher probability that some key words for classification will be remove from the token lists. As a result the accuracy goes down.

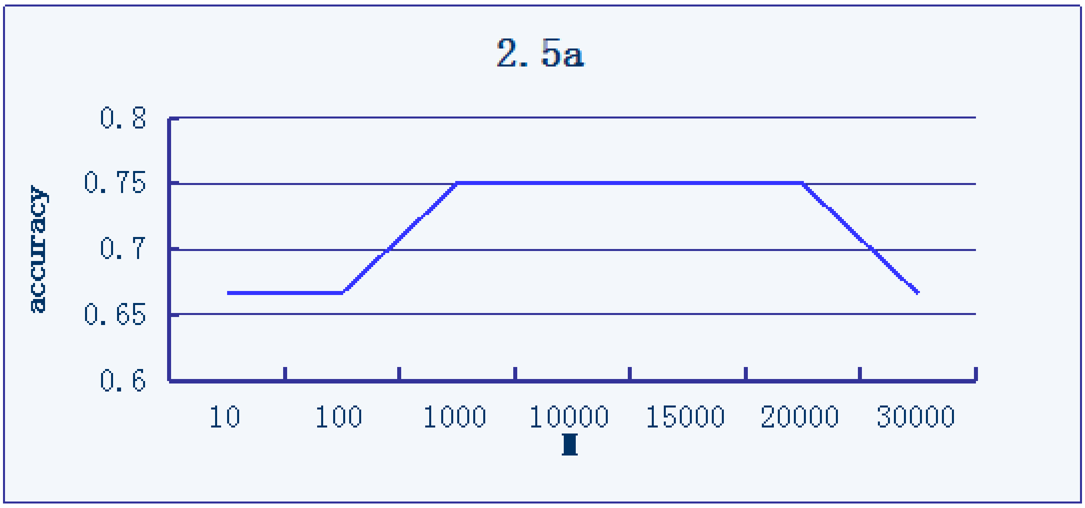
2.4 Smoothing (Total points: 20)

(a)Uniform reference distribution

using split3.train and split3.test

M	Classification accuracy

10	0.66667
100	0.66667
1000	0.75
10000	0.75
15000	0.75
20000	0.75
30000	0.66667



The curve goes up at first and get the top then goes down as M goes larger.

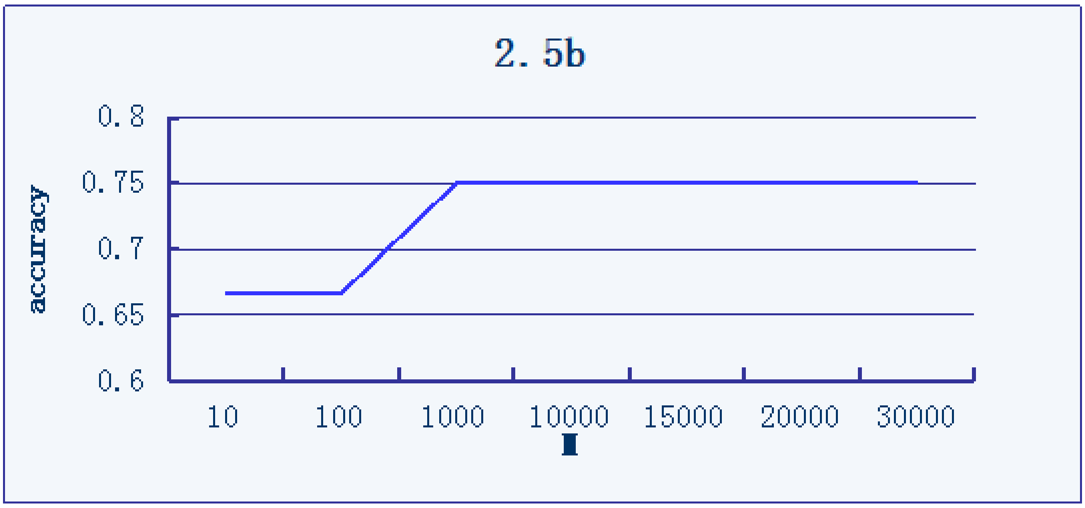
M is the weight of $\frac{1}{|V|}$. As $\frac{freq + M\frac{1}{|V|}}{tokens + M}$, we can see whether M is too small or it's too large, the accuracy is low. When M is too small, the probability of unobserved word will be very small which will make the result so close for differing. When M is too big, the probability goes to uniform distribution which means every word has similar probability. Then there is no difference between key words, stop or noisy words.

As a result, we need to find a balance of M to have a better probability for each word. A better probability means, we need to smooth the unobserved word and give a proper one to the key words for classification.

(b) Non-uniform reference distribution

using split3.train and split3.test

M	Classification accuracy
10	0.66667
100	0.66667
1000	0.75
10000	0.75
15000	0.75
20000	0.75
30000	0.75

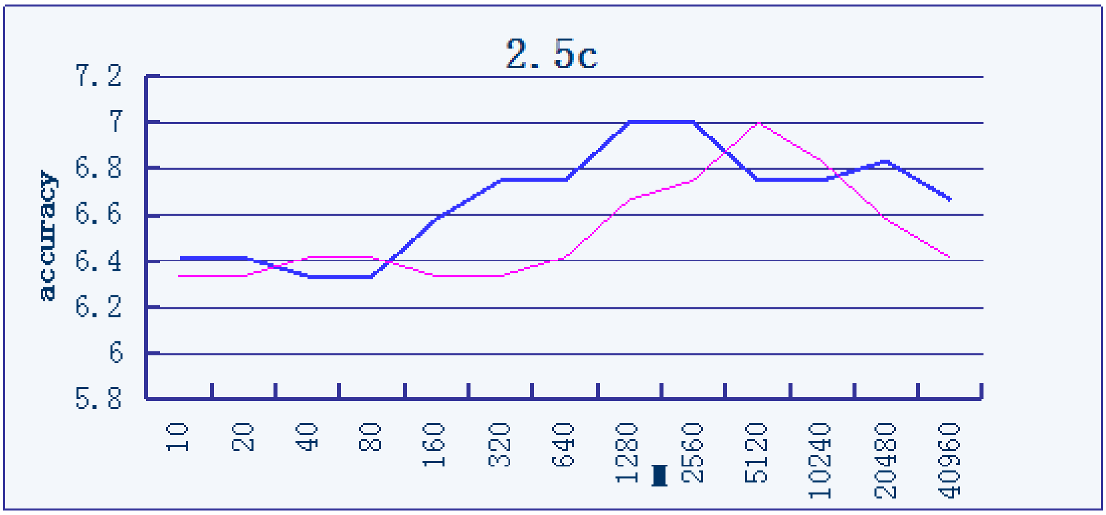


The curve goes up at first and get the top as M goes larger. Actually it should go down later even though the experiments did not give a final data.

When using Non-uniform reference distribution, the trend is almost the same as uniform distribution. A small M will let probability difficult to differ as there are too small probabilities for unobserved word, and a large M will let probabilities uniform which also have a bad influence on telling the difference. There is also a point that M will give a better description for the probability of every word.

(c) comparison

M	(a) avg accuracy	(b) avg accuracy
10	6.416666667	6.333333333
20	6.416666667	6.333333333
40	6.333333333	6.416666667
80	6.333333333	6.416666667
160	6.583333333	6.333333333
320	6.75	6.333333333
640	6.75	6.416666667
1280	7	6.666666667
2560	7	6.75
5120	6.75	7
10240	6.75	6.833333333
20480	6.833333333	6.583333333
40960	6.666666667	6.416666667



Firstly, I choose (a).

Secondly, the difference of (a) and (b) can be seen from the plot.

Here we can see a right shift of curve (b) from curve (a). Actually, $q = \frac{all\_freq + 1}{all\_token + |V|}$.

When M is the same, $q = \frac{all\_freq + 1}{all\_token + |V|}$ is kind of larger than $\frac{1}{|V|}$ because we add numbers simultaneously to numerator and denominator. The function of all_freq and all_token is to smooth q from probability. It means based on a uniform $\frac{1}{|V|}$, frequent word has a higher q than a rear word. This leads to a increase in probability for every word. But more increase for more frequent. The trend moves a little earlier and a little tighter. However (b) is poster than (a) and smaller than (a) based on the same M. That is why there is a right shift.

So it depends which method we choose. However we can choose (a) as it looks better for this task when M goes to a more median value.