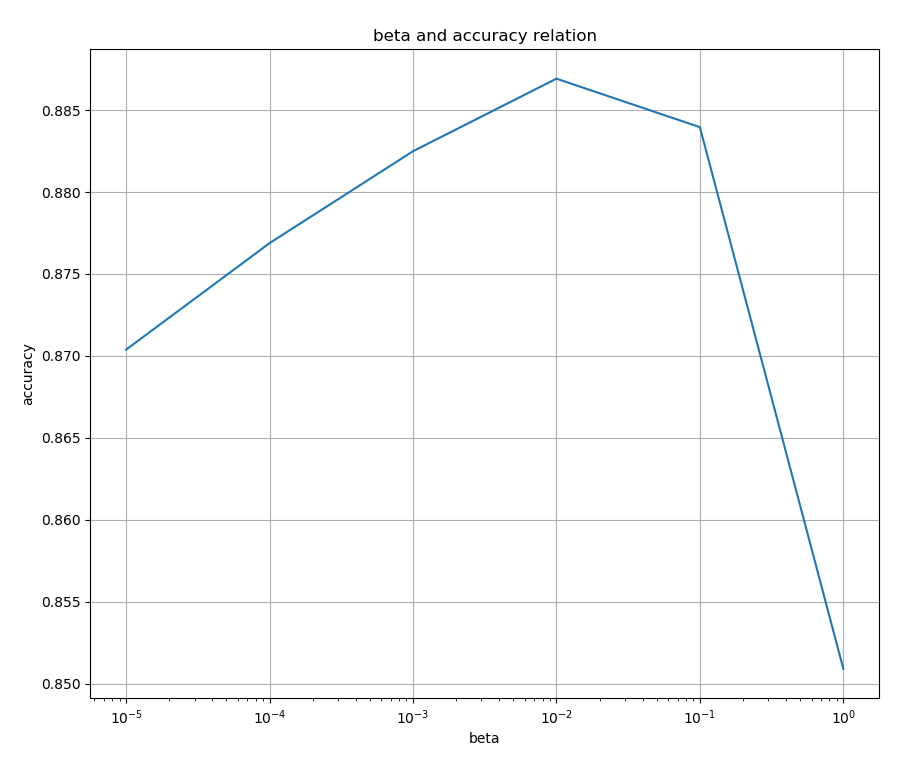
Question 4:

the NB classifier is retrained with beta value ranging from 0.00001 to 1.

The beta/accuracy relation is shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| beta | 0.00001 | 0.0001 | 0.001 | 0.01 | 0.1 | 1 |
| accuracy | 0.87038 | 0.87688 | 0.88249 | 0.88692 | 0.88396 | 0.85090 |

Plot with Matplotlib:



Observation: the beta value at 0.01 gives the best result at 0.88692. while values smaller or bigger have a tendency to reduce accuracy. The reduction is more significant when beta is very big.

Explain:

When beta is big, it is the dominant factor in the probability calculation, so it will overwhelm any estimation from real data. And when beta is too small, it’s not playing any role in the probability estimation, so it’s basically like without the smoothing.

An appropriate value of beta not too small and too big is required to get an better estimation.

Question5:

Method proposal:

Entropy is a good measure of the uncertainty of a random variable. For a word Xi that is quite indicative of the group it belongs to, we expect the entropy of the H(Y/Xi) should be low, meaning it’s distribution in all groups is highly unbalanced, favoring one or only a few of all groups.

Of course, we must consider the possibility of rare words that just by chance appears only in certain groups. A smoothing method like introducing a Dirichlet prior is necessary.

H(Y/Xi) can be calculated with using this formula:

H(Y/Xi)= ∑ -P(Yj/Xi)\*log(P(Yj/Xi)) (the sum of for all Yj)

And P(Yj/Xi) can be calculated with:

P(Yj/Xi)= P(Yj)\*P(Xi/Yj)/P(Xi)

We already calculated P(Yj) and P(Xi/Yj) in the NB algorithm. P(Xi) is just a normalizing factor which do not need to calculate. We just normalize P(Yi)\*P(Xi/Yj).

From above we can calculate H(Y/Xi) easily.