COMS 4771 Machine Learning (Fall 2022)
Problem Set #1

Xinhao Li - xl2778@columbia.edu

2022/10/07

Problem 1

(a) We know that the equation for the likelihood estimation is:

$$L(\theta \mid X) := P(X|\theta) = \prod_{i=1}^{n} p_\theta(\boldsymbol{x}_i) \tag{1}$$

Here, we know the density as it is given in the question, so the likelihood estimation can be rewritten as:

$$L(\theta \mid X) := \begin{cases} \prod_{i=1}^{n} e^{\theta - \boldsymbol{x}_i} & \text{if } \boldsymbol{x}_i \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Parameter setting θ that maximizes $L(\theta \mid X)$:

$$\text{argmax} L(\theta \mid X) = \text{argmax} \prod_{i=1}^{n} e^{\theta - \boldsymbol{x}_i} \text{ (if } \boldsymbol{x}_i \geq \theta) \tag{3}$$

We know that both the e^x and lnx are monotonically increasing functions. So here let's take ln of $L(\theta \mid X)$ and get:

$$\begin{aligned} \text{argmax}(ln(L(\theta \mid X))) &= \text{argmax}(ln(\prod_{i=1}^{n} e^{\theta - \boldsymbol{x}_i})) \\ &= \text{argmax}(ln(e^{\theta - \boldsymbol{x}_1} e^{\theta - \boldsymbol{x}_2} \cdot ..^{\theta - \boldsymbol{x}_n})) \\ &= \text{argmax}((\theta - \boldsymbol{x}_1) + (\theta - \boldsymbol{x}_1) + \cdots + (\theta - \boldsymbol{x}_n)) \\ &= \text{argmax}(n\theta - \sum_{i=1}^{n} (\boldsymbol{x}_i)) \end{aligned} \tag{4}$$

So now, we are trying to find θ which will maximize $n\theta - \sum_{i=1}^{n}(\boldsymbol{x}_i)$. And we know that $\boldsymbol{x}_i \geq \theta$. This means each $(\theta - \boldsymbol{x}_i)$ in the summation is smaller than 0. Therefore, in order to maximize the likelihood function, θ should be as large as possible but no larger than any of $\boldsymbol{x}_i$.

$$0 < \theta \leq \forall \boldsymbol{x}_i \tag{5}$$

In conclusion, the result is:

$$\theta_{MLE} = \min_{\mathbf{n}}(\boldsymbol{x}_\mathbf{i}) \tag{6}$$

(b) Every continuous bijective function from $\Re$ to $\Re$ is strictly monotonic. So let :

$$L(\theta \,|\, X) = h(\theta) \; and \; \theta = g(\eta) \tag{7}$$

Then:

$$L(\eta \,|\, X) = h(\theta) = h(g(\eta)) = H(\eta) \tag{8}$$

Let's take the derivative of $H(\eta)$ and we get:

$$\frac{\partial H(\eta)}{\partial \eta} = \frac{\partial h(g(\eta))}{\partial \eta} \cdot \frac{\partial g(\eta)}{\partial \eta} \tag{9}$$

This equation will be zero at the maximum and can be achieved by having $\frac{\partial h(g(\eta))}{\partial \eta} = 0$ or $\frac{\partial g(\eta)}{\partial \eta} = 0$. Since we know that $\frac{\partial g(\eta)}{\partial \eta}$ can not be zero as function g is monotonic. So we can only have $\frac{\partial h(g(\eta))}{\partial \eta} = 0$ and this leads to:

$$\frac{\partial h(g(\eta))}{\partial \eta} = 0 = \frac{\partial h(\theta_{MLE})}{\partial \eta} \tag{10}$$

In conclusion, we have:

$$g(\eta) = \theta_{MLE} \tag{11}$$

(c) Based on the Bayes theorem, we know:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{12}$$

So if we have prior knowledge, the MAP can be calculated as:

$$\begin{aligned} \theta_{MAP} &= \text{argmax}_\theta \; p(\theta \mid \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \\ &= \text{argmax}_\theta \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \mid \theta) \cdot p(\theta)}{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)} \end{aligned} \tag{13}$$

Since $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$ is independent of θ, so:

$$\theta_{MAP} = \text{argmax}_\theta p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \mid \theta) \cdot p(\theta) \tag{14}$$

From the question and part (a), we know the $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \mid \theta)$ and $p(\theta)$. Therefore, we can show that:

$$\theta_{MAP} = \text{argmax}_\theta (\prod_{i=1}^{n} e^{\theta - \boldsymbol{x}_i} \cdot 2e^{-\theta^2} \pi^{-1/2}) \tag{15}$$

Again, let's take the ln of the θ_{MAP} and get:

$$ln(\theta_{MAP}) = n\theta - \sum_{i=1}^{n} \boldsymbol{x}_i + ln2 - \theta^2 + ln(\pi^{-1/2}) \tag{16}$$

Let's take the derivative of the equation 16 to find the θ_{MAP} and we have:

$$\frac{dln(\theta_{MAP})}{d\theta} = n - 2\theta = 0 \; \Rightarrow \theta_{MAP} = \frac{n}{2} \tag{17}$$

Problem 2

(i)
As defined in question:

$$M := A^T A \tag{18}$$

Based on the definition of symmetric matrices, we know that: Matrix M is symmetric if $M = M^T$. Here we have:

$$M^T = (A^T A)^T = A^T (A^T)^T = A^T A = M \tag{19}$$

So M is symmetric.
An $n \times d$ symmetric real matrix M is said to be positive-semidefinite if $x^T M x \geq 0$ for all x in $\Re^d$. Here we have:

$$\begin{aligned} x^T M x &= x^T A^T A x \\ &= (Ax)^T (Ax) \\ &= ||(Ax)^2|| \\ &= \sum_{i=1}^{n} \Big(\sum_{j=1}^{d} (a_{ij} x_j)^2 \Big) \geq 0 \end{aligned} \tag{20}$$

In conclusion, the matrix M is symmetric positive semi-definite.

(ii)
Based on the definition of β^k:

$$\beta^k := \beta^{k-1} + \eta A^T (b - A\beta^{k-1}) \tag{21}$$

We are asked to prove that:

$$\beta^N = \eta \sum_{k=0}^{N-1} (I - \eta M)^k v \tag{22}$$

To proof using mathematical induction, the base case is: When N = 1:

$$\beta^1 := \beta^0 + \eta A^T (b - A\beta^0) = \eta A^T b \tag{23}$$

$$\beta^1 = \eta \sum_{k=0}^{0} (I - \eta M)^0 v = \eta v = \eta A^T b \tag{24}$$

Our hypothesis is when N = k, it will satisfy that $\beta^N = \sum_{k=0}^{N-1}(I-\eta M)^k v$. And the induction step should show that when N = k+1, it also satisfy $\beta^N = \sum_{k=0}^{N-1}(I-\eta M)^k v$.

$$\begin{aligned}
\beta^{k+1} &= \beta^k + \eta A^T(b - A\beta^k) \\
&= \eta \sum_{k=0}^{k-1}(I-\eta M)^k v + \eta A^T\Big[b - A\eta \sum_{k=0}^{k-1}(I-\eta M)^k v\Big] \\
&= \eta \sum_{k=0}^{k-1}(I-\eta M)^k v + \eta A^T v - \eta A^T A\eta \sum_{k=0}^{k-1}(I-\eta M)^k v \\
&= \eta(I-\eta A^T A)\sum_{k=0}^{k-1}(I-\eta M)^k v + \eta A^T b \\
&= \eta(I-\eta M)^1 \sum_{k=0}^{k-1}(I-\eta M)^k v + \eta(1-\eta M)^0 v \qquad (25)
\end{aligned}$$

Let $a = I - \eta M$ and equation 25 becomes:

$$\begin{aligned}
\beta^{k+1} &= \eta a^1 \sum_{k=0}^{k-1} a^k v + \eta a^0 v \\
&= \eta a^0 v + \eta a^1\Big(a^0 + a^1 + ... + a^{k-1}\Big)v \\
&= \eta a^0 v + \eta a^1 v + \eta a^2 v + ... + +\eta a^k v \\
&= \eta \sum_{k=0}^{k} a^k v \\
&= \eta \sum_{k=0}^{k}(I-\eta M)^k v \qquad (26)
\end{aligned}$$

This proves that for any positive integer N: $\beta^N = \eta\sum_{k=0}^{N-1}(I-\eta M)^k v$.

(iii)
Because $M = A^T T$, so M is invertible. From the question, we also know that η is always greater than 0. So we can say that $I - \eta M$ is invertible. We know that the eigenvalues of M is λ_i for all $i = 1...d$. Then the eigenvalues of $\eta\sum_{k=0}^{N-1}(I-\eta M)^k$ is: $\eta\sum_{k=0}^{N-1}(I-\eta\lambda_i)^k$ for all $i = 1...d$.

(iv)

$$
\begin{aligned}
||\beta^N - \hat{\beta}||_2^2 &= ||\eta \sum_{k=0}^{N-1} (I - \eta M)^k M\hat{\beta} - \hat{\beta}||_2^2 \\
&\leq ||\eta \sum_{k=0}^{N-1} (I - \eta M)^k M - I||_2^2 \; ||\hat{\beta}||_2^2 \\
&\leq \lambda_{max}\Big[(\eta \sum_{k=0}^{N-1} (I - \eta M)^k M - I)^T (\eta \sum_{k=0}^{N-1} (I - \eta M)^k M - I)\Big] \; ||\hat{\beta}||_2^2 \qquad (27)
\end{aligned}
$$

As we know $\eta \sum_{k=0}^{N-1} (I - \eta M)^k M - I$ is symmetric, so the above equation becomes like:

$$
\begin{aligned}
||\beta^N - \hat{\beta}||_2^2 &\leq \lambda_{max}\Big[(\eta \sum_{k=0}^{N-1} (I - \eta M)^k M - I)^2\Big] \; ||\hat{\beta}||_2^2 \\
&\leq max_i\Big[(\eta \sum_{k=0}^{N-1} (I - \eta \lambda_i)^k \lambda_i - I)^2\Big] \; ||\hat{\beta}||_2^2 \qquad (28)
\end{aligned}
$$

Since $1 + x + x^2 + ... + x^n = \frac{1-x^{n+1}}{1-x}$, so here we have:

$$
||\beta^N - \hat{\beta}||_2^2 \leq max_i\Big[(\eta\lambda_i \frac{1 - x^N}{1 - x} - 1)^2\Big] \; ||\hat{\beta}||_2^2 \qquad (29)
$$

where $x = I - \eta\lambda_i$. Finally we have:

$$
||\beta^N - \hat{\beta}||_2^2 \leq max_i\Big[I - \eta\lambda_i\Big]^{2N} \; ||\hat{\beta}||_2^2 \qquad (30)
$$

As mentioned in the question, $\lambda_i < \frac{1}{\eta}$, so we have $1 - \eta\lambda_i > 0$. As $\eta > 0$, we want λ_i as small as possible. So:

$$
\begin{aligned}
max_i\Big[I - \eta\lambda_i\Big]^{2N} &= (1 - \eta\lambda_{min})^{2N} \\
&\leq e^{-2\eta\lambda_{min}N} \qquad (31)
\end{aligned}
$$

In conclusion, we have:

$$
||\beta^N - \hat{\beta}||_2^2 \leq e^{-2\eta\lambda_{min}N} \; ||\hat{\beta}||_2^2 \qquad (32)
$$

Problem 3

(a) First show that the asymptotic error rate of 3-NN classifier is at most 1.6 times Bayes optimal classifier.

$$\begin{aligned}\lim_{n\to\infty} P_{y_t,D_n}[e|x_t] &= \lim_{n\to\infty}\int P_{y_t,y_n}[e|x_t X_n]P[X_n|x_t]dX_n \\ &= \lim_{n\to\infty}\iiint P_{y_t,y_n}[e|x_t,x_{n1},x_{n2},x_{n3}]P[x_{n1},x_{n2},x_{n3}|x_t]dx_{n1}dx_{n2}dx_{n3} \\ &= \lim_{n\to\infty}\iiint P_{y_t,y_n}[e|x_t,x_{n1},x_{n2},x_{n3}]P[x_{n1},x_{n2},x_{n3}|x_t]dx_{n1}dx_{n2}dx_{n3} \quad (33)\end{aligned}$$

Here the error rate can be defined as: (1 - the cases that majority of the nearest neighbors have predict the label correctly), which means for 3-NN classifier there are two possibilities. One is the case that two nearest neighbors predict the labels correctly but the last one does not (total three combinations). The second case should be that three nearest neighbors all predict the labels correctly (one combination).
Let p_{n1}, p_{n2}, p_{n3} be the probabilities that the nearest neighbors n_1, n_2, n_1 have the same label as y. The actual expression for p_{n1} is:

$$p_{n1} = P(y_{n1} = y|x_{n1}) \quad (34)$$

And p_t be the probability that : $p_t = P(y_t = y|x_t)$. So the above equation can be written as:

$$\begin{aligned}\lim_{n\to\infty}\iiint 1 - \Big[&\sum_{y\in Y} p_{n1}p_{n2}(1-p_{n3})p_t - \sum_{y\in Y} p_{n1}p_{n3}(1-p_{n2})p_t \\ &- \sum_{y\in Y} p_{n2}p_{n3}(1-p_{n1})p_t - \sum_{y\in Y} p_{n1}p_{n2}p_{n3}p_t\Big]dx_{n1}dx_{n2}dx_{n3} \quad (35)\end{aligned}$$

As n approaches infinity, we have $x_{n1} = x_t, x_{n2} = x_t, ..., y_{n1} = y_t, y_{n2} = y_t, ...$ Therefore. we can write equation 35 as:

$$\lim_{n\to\infty}\iiint 1 - 3\sum_{y\in Y}[P(y_t = y)|x_t]^3[1 - P(y_t = y)|x_t] - \sum_{y\in Y}[P(y_t = y)|x_t]^4 dx_{n1}dx_{n2}dx_{n3} \quad (36)$$

$$= 1 - 3\sum_{y\in Y}[P(y_t = y)|x_t]^3 + 2\sum_{y\in Y}[P(y_t = y)|x_t]^4 \quad (37)$$

We know that $P(y_t = y|x_t) \in (0, 1)$. So $[P(y_t = y)|x_t]^3 \geq [P(y_t = y)|x_t]^4$. And this leads to:

$$-3\sum_{y\in Y}[P(y_t = y)|x_t]^3 + 2\sum_{y\in Y}[P(y_t = y)|x_t]^4 \leq -3[P(y_t = y)|x_t]^3 + 2[P(y_t = y)|x_t]^4 \quad (38)$$

If Bayes classifier returns y_t^* at point x_t, then we have:

$$\begin{aligned} 1 - 3\sum_{y\in Y}[P(y_t = y)|x_t]^3 + 2\sum_{y\in Y}[P(y_t = y)|x_t]^4 & \\ \leq 1 - 3[P(y_t = y_t^*)|x_t]^3 + 2[P(y_t = y_t^*)|x_t]^4 & \\ \leq k(1 - P(y_t = y_t^*|x_t)) & \end{aligned} \tag{39}$$

So here, the problem becomes to maximize the k value:

$$k = maximize(\frac{1 - 3[P(y_t = y_t^*)|x_t]^3 + 2[P(y_t = y_t^*)|x_t]^4}{1 - P(y_t = y_t^*|x_t)}) \tag{40}$$

By solving equation 40 using wolframalpha ($P(y_t = y_t^*|x_t) \in (0, 1)$), we have:

$$k \approx 1.5282 \tag{41}$$

In conclusion, this means that the asymptotic error rate of 3-NN classifier is at most 1.6 times Bayes optimal classifier.

For the best 5-NN asymptotic error rate, it basically follows the same steps as I mentioned before, but just the derivation process is more complicated since we have to consider more possibilities. This time the cases that majority of the nearest neighbors have predict the label correctly have following:

- $C_5^5 = 1$ for the cases that 5 of 5 nearest neighbors have predict the label correctly.
- $C_5^4 = 5$ for the cases that 4 of 5 nearest neighbors have predict the label correctly.
- $C_5^3 = 10$ for the cases that 3 of 5 nearest neighbors have predict the label correctly.

By directly using equation 36, we have:

$$\begin{aligned} \lim_{n\to\infty} \iiiint \int 1 - 10\sum_{y\in Y}[P(y_t = y)|x_t]^4[1 - P(y_t = y)|x_t]^2 & \\ -5\sum_{y\in Y}[P(y_t = y)|x_t]^5[1 - P(y_t = y)|x_t] & \\ -\sum_{y\in Y}[P(y_t = y)|x_t]^6 \; dx_{n1}dx_{n2}dx_{n3}dx_{n4}dx_{n5} & \end{aligned} \tag{42}$$

$$= 1 - 6\sum_{y\in Y}[P(y_t = y)|x_t]^6 + 15\sum_{y\in Y}[P(y_t = y)|x_t]^5 - 10\sum_{y\in Y}[P(y_t = y)|x_t]^4 \tag{43}$$

Following the same procedures as before and we will have:

$$
\begin{aligned}
&1 - 6\sum_{y\in Y}[P(y_t = y)|x_t]^6 \\
&+15\sum_{y\in Y}[P(y_t = y)|x_t]^5 \\
&-10\sum_{y\in Y}[P(y_t = y)|x_t]^4 \\
&\qquad\qquad \leq 1 - 6[P(y_t = y_t^*)|x_t]^6 + 15[P(y_t = y_t^*)|x_t]^5 - 10[P(y_t = y_t^*)|x_t]^4 \\
&\qquad\qquad \leq k(1 - P(y_t = y_t^*|x_t)) \qquad (44)
\end{aligned}
$$

So here, the problem becomes to maximize the k value:

$$k = maximize(\frac{1 - 6[P(y_t = y_t^*)|x_t]^6 + 15[P(y_t = y_t^*)|x_t]^5 - 10[P(y_t = y_t^*)|x_t]^4}{1 - P(y_t = y_t^*|x_t)}) \qquad (45)$$

By solving equation 45 using wolframalpha ($P(y_t = y_t^*|x_t) \in (0, 1)$), we have:

$$k \approx 1.5011 \qquad (46)$$

In conclusion, this means that the asymptotic error rate of 3-NN classifier is at most 1.51 times Bayes optimal classifier.

Problem 4

(a)
In order to systematically embed text data in a Euclidean space, I first combined both the "spam" and "ham" emails in order to generate all works that appearing in these emails. After having all works available, I am going to do word stemming. This is done by using the open source package 'nltk'. By applying the word stemming and remove all non-word characters and punctuations, I have 42730 "words" in total. By looking at the words generated after the stemming, I realized most of the works from the 42730 words set have only frequency even below 10. Remember that we have more than 5000 emails, therefore I further filtered the words set by choosing the high frequency words in the set. In the context, I choose the works that appear at least 50 times and overall we have 1832 words. Definitely this kind of selecting word will affect the accuracy and I will test this part later once the classifiers have been built. Finally, the words in each email will use this vocabulary to embed into a dictionary where the key is the email filename, and the value is a list which counts how often each work appears. Below shows an example data frame which includes 5 emails and 4 words.

file	0	1	2	3	4
3287.2001-01-09.farmer.ham.txt	0	0	0	0	1
2339.2000-09-26.farmer.ham.txt	0	0	0	0	0
3782.2001-03-14.farmer.ham.txt	0	0	0	0	0
3666.2005-02-02.GP.spam.txt	0	0	0	0	0
0770.2000-03-28.farmer.ham.txt	0	0	0	0	0

(b)
For question (b), I built two classifiers from scratch, one is the naive bayes classifier and the other is the nearest neighbor classifier with "manhattan", "euclidean", and "max_distance" distance metrics implemented. For the detailed implementation please look at the source code.

(c)
Let's first compare the accuracy of different classifiers at different train/test split ratio. We will use the most 50 frequent works to vectorize all emails first. For the nearest neighbor classifier, we will first use the euclidean distance as the metrics. From the figures shown below we can see that over all train/test split ratio and all (from 1 to 5) nearest neighbors considered, **naive bayes classifier always perform better than nearest neighbor classifier**. Noticed that the both classifiers are trained and tested on the same dataset. The best accuracy for the two classifiers has been tabulated in table 1:
As I mentioned before, for these tests, only the euclidean distance has been considered.

So next, I would like to test how different distance metrics will affect the accuracy of the classifiers (only tested on the 2-NN classifier since it is the optimal among all). From figure 5, we can clearly see that the classifier using euclidean distance performs best and the classifier using the max_distance performs worst. The last thing I want to test is the works vocabulary used. For the above tests, I only use the words that appear at least 50 times. Figure 6 shows how how different vocabulary will affect the accuracy of the classifiers. We can see that overall, both classifiers using the words vocabulary which contains the words that appear

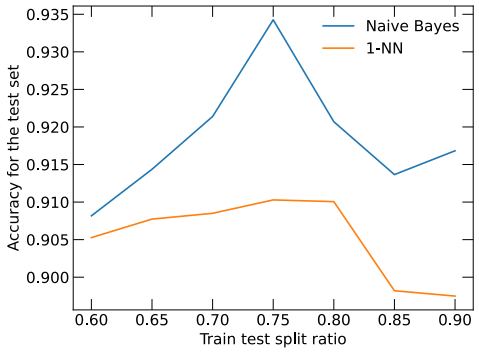

Figure 1: Naive Bayes compared with 1-NN, Euclidean

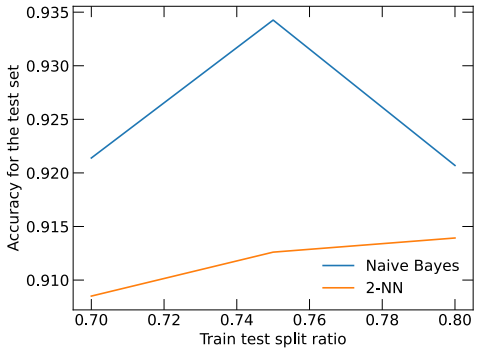

Figure 2: Naive Bayes compared with 2-NN, Euclidean

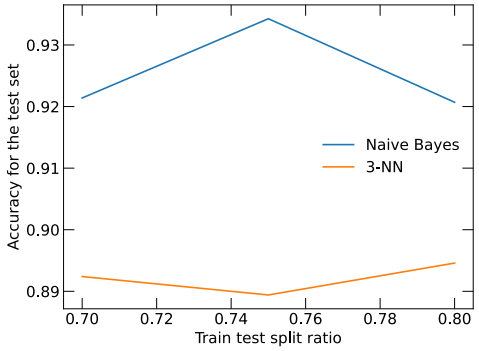

Figure 3: Naive Bayes compared with 3-NN, Euclidean

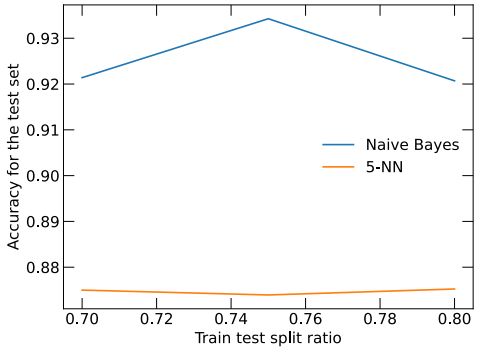

Figure 4: Naive Bayes compared with 5-NN, Euclidean

at least 50 times perform the best among other classifiers using different words vocabulary. However, one thing we need to pay attention on is that actually the best works vocabulary (>50) so far has less words than the rest two vocabularies. This means, sometimes, if we considered more irrelevant/noise data in the dataset, it will lead to bad results.

Table 1: Accuracy for two classifiers

	Accuracy
Naive Bayes Classifier	93.5%
Nearest Neighbor Classifier (euclidean dist., 2 NN)	91.5%

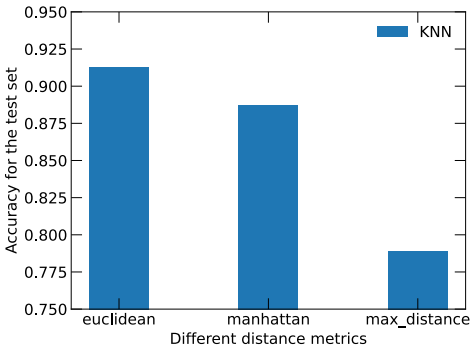

Figure 5: Different distance metrics comparison

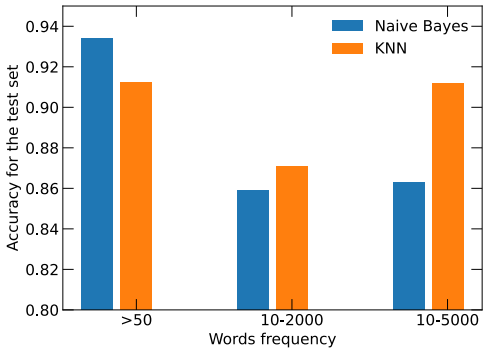

Figure 6: Different work vocabulary comparison

Appendix

- preprocessing_featurizing.py
- classifier.py
- run.py