**Assignment 03 – Naive Bayes/Logistic Regression Classification**

***Chentai Yuan, Qianzhong Chen, Hao Ding***

***ECE448, Group 07***

***TA: Shuting Tao, Hanrong Zhang***

***April 24, 2023***

**Table of Contents**

**1. Image Classification 2**

**2. Text Classification 3**

**3. Offensive agent vs Your agent 4**

**4. Human vs. Your agent6**

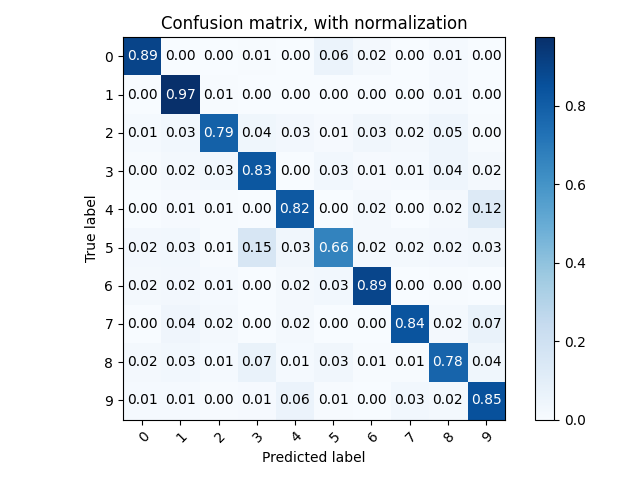
**5. Extra Credit 7**

**6. Acknowledgments 9**

**Section Ⅰ: Image Classification**

1. Classification rate

In our training set and test set, the average classification rate is 0.836, and classification rate for digit 0 is 0.89, for digit 1 is 0.97, for digit 2 is 0.79, for digit 3 is 0.83, for digit 4 is 0.82, for digit 5 is 0.66, for digit 6 is 0.89, for digit 7 is 0.84, for digit 8 is 0.78, for digit 9 is 0.85.

2. Confusion matrix

3. Test examples for highest and lowest posterior probabilities

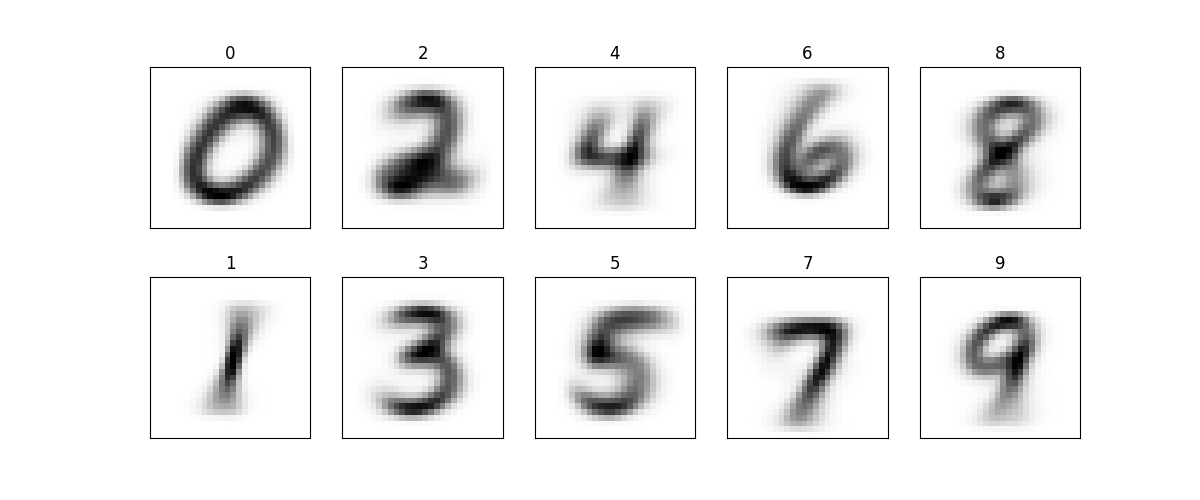
图片包含 灯光, 游戏机

描述已自动生成For highest probabilities, digits are shown in thick and clear writing patterns.

For lowest probabilities, some digits are thin, and for digit “7”, it shows like digit “1”.

图片包含 图示

描述已自动生成

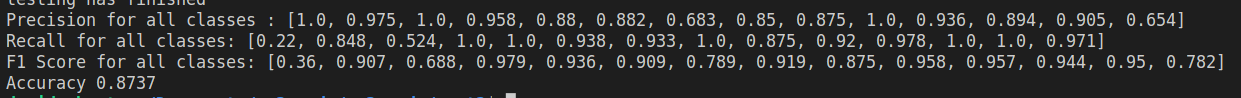
4. Feature likelihood plots

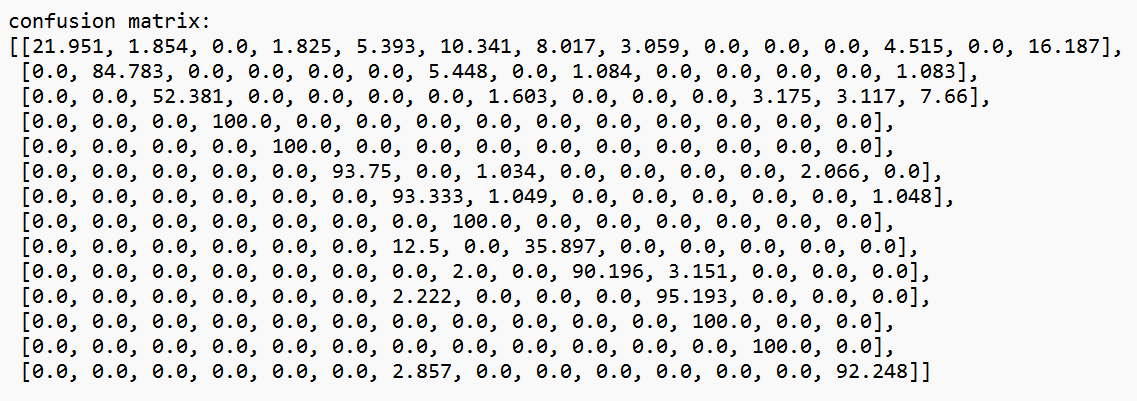
**Section Ⅱ: Text Classification**

1. Top 20 frequently occurred words of each class

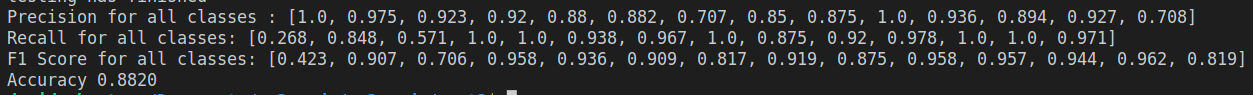


1. Prior Case

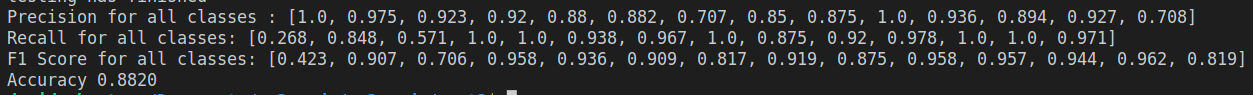




1. ML Case:



1. Uniform Distribution Case

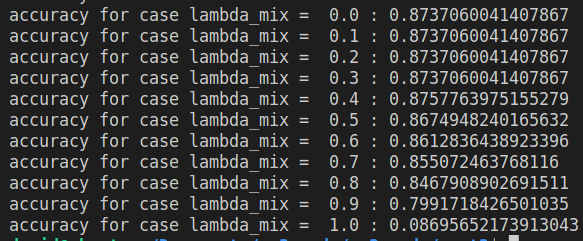


Discussion:

Not using class prior probability or Changing to uniform distribution increased the classification accuracy, meaning that including prior probability in naïve Bayes not always benefits the classification results. It may be attributed to the divergence in class distribution between training set and test set. Meanwhile, it makes great sense that ML case and uniform distribution case have exactly the same results, as posterior probability of ML case for each token should be 14 times of posterior probability of uniform distribution case, which dose not affect argmax function and final classification results.

**Section Ⅲ: Linear Classifier**

**Section Ⅳ: Bigram Classification**

****

Maximum accuracy occurs at lambda\_mix = 0.4, with an accuracy of 0.8757, pure bigram classification accuracy is 0.0870.

*Q1. Does relaxation always help?*

Relaxation of naïve Bayes assumption does not always help. When we use a lambda\_mix of 1, which means that we only consider classifying using bigrams, the accuracy is very bad. It may happen because that bigram as a combination has way more degree of uniqueness, our training set cannot cover the breadth of bigrams and a lot of bigrams that our program has not learned occurs at test set, causing the trouble that it cannot classify the texts well.

*Q2. Is N-gram with a large N a good idea?*

No. As discussed before, with the increase of N, the complexity as well as number of linguistic data increase exponentially. As the lambda\_mix requires well-tuning, the much more time spent in training the model cannot trade-off the tiny improvement in performance.

**ACKNOWLEGEMENTS**

Statement of Contribution:

Qianzhong Chen developed all three parts and extra credit part individually and his code is submitted as .py. He is responsible for Section#2 and Section#4 of the report.

Chentai Yuan mainly implemented and uploaded the Assignment of image\_main.py and naive\_bayes.py, also he join the work for other two parts. He improved the interface for training and wrote the report paper for Section#1.

Hao Ding implemented the human agent and helped with debugging. He wrote the report paper for section#3.

**REFERENCE**

1. " Assignment 3: Naive Bayes/Logistic Regression Classification - ECE448 Spring 2023 Assignment#3 Manual."
2. Wikipedia on Logistic Regression: “https://en.wikipedia.org/wiki/ Logistic\_regression”