CIS521 – HW7

Writeup

Nathan Fraenkel

Boyang Zhang

Table:

*Pair 1 = (PC vs. Mac)*

*Pair2 = (Baseball vs. Hockey)*

*Pair3 = (PC vs. Baseball)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Error seed=1** | **Error seed=2** | **Error seed=3** | **Error seed=4** | **Error seed=5** |
| **Naïve Bayes 1** | 0.251928021 | 0.218508997 | 0.218508997 | 0.195372751 | 0.187660668 |
| **Streamwise Regression 1** | N/A | N/A | N/A | N/A | N/A |
| **Stepwise Regression 1** | N/A | N/A | N/A | N/A | N/A |
| **Perceptron 1** | 0.128534704 | 0.143958869 | 0.133676093 | 0.177377892 | 0.146529563 |
|  |  |  |  |  |  |
| **Naïve Bayes 2** | 0.244444444 | 0.166666667 | 0.255555556 | 0.2 | 0.211111111 |
| **Streamwise Regression 2** | N/A | N/A | N/A | N/A | N/A |
| **Stepwise Regression 2** | N/A | N/A | N/A | N/A | N/A |
| **Perceptron 2** | 0.122222222 | 0.088888889 | 0.1 | 0.133333333 | 0.133333333 |
|  |  |  |  |  |  |
| **Naïve Bayes 3** | 0.254681648 | 0.262172285 | 0.235955056 | 0.232209738 | 0.228464419 |
| **Streamwise Regression 3** | N/A | N/A | N/A | N/A | N/A |
| **Stepwise Regression 3** | N/A | N/A | N/A | N/A | N/A |
| **Perceptron 3** | 0.037453184 | 0.029962547 | 0.037453184 | 0.037453184 | 0.029962547 |

**🡪 IMPORTANT NOTES ABOUT THE TABLE:**

- “Naïve Bayes 1” = Naïve Bayes run on pair 1

- We were unsure about the difference between ‘training error’ and ‘test error’… so we only have one type of error.

- We were unable to successfully finish the streamwise and stepwise regression implementations, so we were unable to get results for them :\

**Questions:**

1. *What choices did you make when implementing the algorithms? List the settings of parameters like learning rate, number of words considered for each algorithm, or the λ parameter. Do you think these choices affected the results in any way? Why or why not?*

While we did finish the implementations of the regressions, we are aware that the lambda values do in fact have a big difference on the result. Lambda values come into play in ridge regression when creating the w matrix and in streamwise and stepwise regression when calculating the training error of w. A greater lambda will mean taking the inverse of a matrix with greater values at each entry (since you are adding lambda \* I), meaning the inverse matrix, and therefore w, will have smaller values. It will also increase the training error since it is defined by some number plus lambda \* length (cols). Number of words considered for each algorithm also makes a difference. Higher number of words means lower uncertainty/ more accuracy/ less error because we have a greater pool of words to operate on and ultimately learn/ adapt from.

1. *Of the three comparisons, which was the most difficult comparison across all algorithms? Why do you think it was the hardest?*

Clearly the third comparison was not the hardest because text involving PC and Baseball would have completely different nouns, adjectives, descriptors, etc. throughout the entire file. This is reflected in our table with the perceptron algorithm results, as pair 3 has the lowest error result of the three.

As for the first two, our results from the perceptron algorithm once again can guide us towards our decision. The error for perceptron run on PC vs. Mac is consistently higher (not by too much, though) than perceptron run on Baseball vs. Hockey. Therefore, we can say that pair 1 is the most difficult comparison. This makes sense, though. Baseball and hockey are both sports, but they aren’t very related besides that. The terminology used in hockey is fundamentally used in baseball, including positions, ball vs. puck, athlete names, running vs. skating, etc. However, if this were baseball vs. softball, or American football vs. rugby, for example, the error would probably be higher because of the similarities within the sports. In that vein, since PCs and Macs are both computers they share a lot more in common than two mildly related sports. They both use the same contextual language (hard drive, keyboard, mouse, display, RAM, etc.), which makes it more difficult for the algorithms to detect differences.

1. *For streamwise and stepwise regression, list the top ten features added for each of the three comparisons. (Make sure to report the words, not just feature indices.) Do these features “make sense,” i.e., would you have predicted that these words were useful for distinguishing which newsgroup a post belongs in? Why or why not?*

Unfortunately, we were not able to finish implementing these algorithms. We ran into problems trying to figure out how to get the indices right for the training error calculation in the streamwise and stepwise algorithms. More specifically, we were running into trouble at steps 2(c) in the streamwise algorithm and step 2(a)iii in the stepwise algorithm: the stage where you calculate the training error. However, we do believe that we were on the right track because 1) we (believe that we) had a correct ridge regression function, which means we were correctly getting the w matrix and 2) we correctly added the rest of the logic for the pseudocode outlined in hw7supplemental.pdf for both functions. In the end, we ran out of time and could not figure out computing the training error step in the pseudocode.