CIS600 Fundamentals of Data nd Knowledge Mining

HW3: NBC and kNN

Due: 9pm EST, 05/23, 2019

Task description:

The data set comes from the Kaggle Digit Recognizer competition. The goal is to recognize digits 0 to 9 in handwriting images. Because the original data set is large, I have systematically sampled 10% of the data by selecting the 10th, 20th examples and so on. You are going to use the sampled data to construct prediction models using multiple machine learning algorithms that we have learned recently: naïve Bayes, kNN and SVM algorithms. Tune their parameters to get the best model (measured by cross validation) and compare which algorithms provide better model for this task.

Due to the large size of the test data, submission to Kaggle is not required for this task. However, 5% extra point will be given to successful submissions. Follow the Kaggle competition instruction (https://www.kaggle.com/c/digit-recognizer/data)

Tip: check out the Kaggle forum to see if there are some patterns other people have found that you can use to build better models.

Data Sources

- Original data file
  - original training data (77MB) https://www.dropbox.com/s/npxk66fxruv09u5/Kaggle-digit-train.csv?dl=0
  - original test data (51MB) https://www.dropbox.com/s/3wnkss7x6m4pqgx/Kaggle-digit-test.csv?dl=0
- Sample data file
  - small sample of training data (1.5MB): CSV: https://www.dropbox.com/s/v7dncz0mqklayus/Kaggle-digit-train-sample-small-1400.csv?dl=0
  - small sample of test data (1000 examples):
    CSV: https://www.dropbox.com/s/e5tokwdmkd8ggmm/Kaggle-digit-test-sample1000.csv?dl=0

Report structure:

Section 1: Introduction Briefly describe the classification problem and general data preprocessing. Note that some data preprocessing steps maybe specific to a particular algorithm. Report those steps under each algorithm section.

Section 3: Naïve Bayes Build a naïve Bayes model. Tune the parameters, such as the discretization options, to compare results.

Section 3: K-Nearest Neighbor method

Section 4: Algorithm performance comparison Compare the results from the two algorithms. Which one reached higher accuracy? Which one runs faster? Can you explain why?

Section 5: Kaggle test result (5% extra point) Report the test accuracy for the above models. Discuss whether overfitting occurs in these models.

Grading rubrics:

1. Are the models constructed correctly?
2. Is the result analysis conclusion convincing?
3. Is sufficient details provided for others to repeat the analysis?
4. Does the analysis include irrelevant content?
5. Successful submission to Kaggle?

Please submit your rmarkdown document and knitted report (in either PDF, html or word format)