Nick Videtti

nvidetti@syr.edu

Assignment 7

Support Vector Machines

SVM Building, Predictions with SVM’s, Accuracy Testing, and Comparison to Past Models

9/1/2022

| A Second Attempt at Deciphering Poorly Written Digits | |
| --- | --- |
| **Introduction** | Handwriting has always been an interesting topic. The education system teaches all students to write exactly the same way, but each student’s handwriting ends up being very unique. Many grade school teachers can tell which of their students handed in a written assignment by looking at handwriting alone. The quality of handwriting is based almost exclusively on legibility.  In recent years, cursive handwriting has been removed completely from the curriculum of many elementary schools, and it can be argued that the change was long overdue. Technology has brought society to a point where it can be argued that handwriting is not needed past the very early years of school. The computer has already had a much more substantial impact on the pen/pencil than the pen/pencil ever had on the quill pen. While technology continues to become more user-friendly and climate change continues to occur, going paperless will be as feasible and desirable as ever before, including in the classroom.  The downfall of this current period is that handwriting is still needed at times, but not many outside of academia practice their handwriting, which leads to poorer handwriting in adults than children. Furthermore, most are used to reading either their own handwriting or the perfect text found in books or on screens, which makes it more difficult for the given person to read even slightly sub-par handwriting.  While technology has caused the need for a solution, it also is the reason for a possible solution. Some examples of different numbers that were handwritten have been supplied, including some very heavy detail on how the numbers were written. Using these, can poor handwriting be deciphered? This example will explore digits 0-9, but the same process can be used for letters as well.  In a previous attempt, these same digits were found to be deciphered correctly roughly 84% of the time. This is a second attempt that will hopefully result in an even higher accuracy. |
| **Analysis** | In the previous attempt, the two models that were used were Naive Bayes and Decision Tree. There were some major simplifications made, but the results of that were that the Naive Bayes model predicted with 83.77% accuracy. This time around, 16 different Support Vector Machine models will be generated.  Each of these models will represent a certain kernel and cost. Four of the 16 models will have an optimized, or “tuned”, cost which may result in them being the same as one of the other models if any of the other models using the same kernel are also already using the optimized cost. For each of these “tuned cost” models, the optimized cost is not necessarily the absolute best, but rather the best of the options between 0.001, 0.01, 0.1, 1, 10, 100, and 1000. There will be three other costs used per kernel (0.1, 1, and 10) to make 4 models per each kernel. The four kernels that will be used are linear, polynomial, radial basis, and sigmoid. This will result in the following models being generated.   | **KERNEL** | **COST** | **MODEL NAME** | | --- | --- | --- | | linear | 0.1 | linear\_0.1 | | linear | 1 | linear\_1 | | linear | 10 | linear\_10 | | linear | tuned | linear\_tuned | | polynomial | 0.1 | polynomial\_0.1 | | polynomial | 1 | polynomial\_1 | | polynomial | 10 | polynomial\_10 | | polynomial | tuned | polynomial\_tuned | | radial | 0.1 | radial\_0.1 | | radial | 1 | radial\_1 | | radial | 10 | radial\_10 | | radial | tuned | radial\_tuned | | sigmoid | 0.1 | radial\_0.1 | | sigmoid | 1 | radial\_1 | | sigmoid | 10 | radial\_10 | | sigmoid | tuned | radial\_tuned |   These models will be created for predicting what the written digits are most likely to have been as intended. The data provided are simply examples of different written digits, specifically information for each of the 784 pixels in a 28x28 pixel picture of the written digit. That is, for each row of data, there is a “label” column that indicates the actual intended digit, and 784 columns/attributes of corresponding pixel data.  There are two datasets provided, one for training and one for testing. For the training set, the “label” variable is set as a factor variable. The label column in the testing set will also be set to a factor variable, but will be removed entirely from the testing datasets and stored separately. This way, the models can be tested on the testing dataset, but can also be compared to the actual labels to check for accuracy.  Due to speed issues, the training dataset needed to be limited while training the models. While the full training data is tens of thousands of rows, only the first 1,000 were used for training the 16 models. Even this way, training all 16 models takes about 20 minutes. It could be argued that randomly selecting these 1,000 rows would be better practice, but using the same 1,000 rows of data also gives a more realistic comparison between each of the models.  To summarize, the first step is for the 16 models to be trained and tested. Secondly, the predictions for each will be stored in a dataframe, along with the actual labels. Then, each model will be tested for accuracy by comparing the predicted labels to the actual labels. Finally, the accuracy of the most accurate model created will be compared to the 83.77% accuracy of the Naive Bayes model created in the previous attempt that was discussed in the previous paper. |
| **results** | The accuracy of each of the 16 models was as follows.  (using only first 1000 rows of training dataset)    The cost seemed to not affect accuracy at all. It may be an assumption that this indicates the models are identical regardless of cost, but it is certainly possible. Further, there seems to be no difference between the radial and sigmoid kernels, so the same assumption is also valid in that case. This leaves the polynomial and the linear kernels generating the best and second best accuracy, respectively. Both of these are also more accurate than the Naive Bayes model, which had 83.77% accuracy.  This is certainly not conclusive of which model is truly better since this is only the results of one iteration of model creation, and is only based on the first 1,000 rows of training data. Ideally many iterations of models are created using the whole training dataset. Also, while cost appears to have no effect on accuracy, this is only based on a finite number of cost values. Finding the truly optimal cost, and more importantly finding if a truly optimal cost even exists, would also help generate more conclusive findings. |
| **conclusions** | After one attempt, the digits were predicted correctly for roughly 84% of the written digits. While this is not necessarily the best starting point, it certainly could have been much worse. This time, there were two methods found to be even better than that, one being roughly 89% accurate, the other being roughly 88% accurate.  In the few places of adult life where writing digits is still required, it is paramount that problems like this are able to be solved with nearly flawless accuracy and nearly negligible time as the paperless world, and eventual paperless classroom discussed earlier, to become an efficient reality. |