SVM Assignment

Josh Pike

June 2022

1 Question 1

We implemented the strategy of one vs. rest for our multi-class classification using a soft-margin SVM. We did not use an instance mapping function and we used the squared hinge loss for our loss function.

2 Question 2

We estimated the performance of our trained SVM empirically using cross-validation over 100 iterations using an 80/20 split. We found the percent of correct predictions for each iteration and took its average to give an estimation of our trained model's performance, the results of which you can see in Table 1. We compared this value to the known theoretical upper error bound:

$$P(Error) \leq \frac{\mathbb{E}(N_S)}{N}$$

where $\mathbb{E}(N_S)$ is the estimated number of support vectors, and N is the total number of instances. We can can conversely write this as a lower bound on the percent of correct values by taking $1 - P(error)$:

$$P(Correct) \geq 1 - \frac{\mathbb{E}(N_S)}{N}$$

Cross-Validation % Correct	73.341%
Theoretical % Correct	67.366%

Table 1: Theoretical percent correct and percent correct averaged over 100 iterations of cross-validation

As you can see in Table 1, our cross-validation produced a result that was nearly 6% better than the theoretical lower bound. This is because the theoretical lower bound is just that: a lower bound. The data one uses to train their SVM has the potential to preform better than this bound, but cannot perform worse. Thankfully, our model performed reasonably better than this bound.

3 Question 3

Predictions submitted as "JoshPike_predictions.txt".

4 Question 4

The results of performing no transfer, hypothesis transfer, and instance transfer learning can be seen in 2. For no transfer we simply trained our SVM on the 1 vs. 7 target problem. For hypothesis transfer we first trained our SVM on the source problem 1 vs. 9 and transferred it to the target problem by using the weights obtained from the source problem and training again on the target problem. For instance transfer we trained on and found the support vectors of the source problem and then transferred these support vectors to the target problem by modulating the objective and constraints with the support vectors to train.

No Transfer % Correct	79.045%
Hypothesis Transfer % Correct	80.371%
Instance Transfer % Correct	81.167%

Table 2: Percent correct when performing no transfer, hypothesis transfer, and instance transfer

From Table 2 you can see that each of these transfer instances performed relatively closely with no transfer being worst at 79.045% correct and instance transfer best at 81.167%. We believe no transfer performed the worst because it had the least amount of training data compared to the other two strategies. Hypothesis transfer outperformed no transfer as transferring the weights provided the model with a solid foundation to adjust their model to, thus making it more accurate. We believe the case of instance transfer outperformed hypothesis transfer for two reasons: one and most notably our tuning of the coefficient on our slack variable was much more effective for instance transfer which after tuning caused its percent correct to jump past hypothesis transfer. On the other hand tuning this coefficient for hypothesis transfer did not respond well at all. Two, the actual transfer of the support vectors could provide the model with the potential to create a new weighting that was not bound by what the source problem initially produced, thus providing a more accurate model.