Hand Written Digit Prediction - Supervised Learning Module Assignment 2

Maryam Zolfaghar

[1]Computer Science Department, UC Davis

1. Question 1

Train a linear multi-class classification SVM with no kernel. Specify

- i) Your mapping function
  For the multivector constructions, the mapping function is:
  $$\Psi(X,y) = [\underbrace{0,\ldots,0}_{\in R^{(y-1)n}}, \underbrace{x_1,\ldots,x_n}_{\in R^{n}}, \underbrace{0,\ldots,0}_{\in R^{(k-y)n}}$$
  $$\Psi(X,y) = [\underbrace{0,0,0,0,0,0,0,0}_{\in R^{(y-1)8}}, \underbrace{x_1,\ldots,x_8}_{\in R^{8}}, \underbrace{0,0,0,0,0,0,0,0}_{\in R^{(10-y)8}}$$
  $\Psi(X,y)$ composed of $k = 10$ vectors, each of which is of dimension $n = 8$, where we set all the vectors to be all zeros vector except the y'th vector, which is set to be values of x.
  The mapping/transforming function in terms of kernel is just the identity function ($K(x,x') =< x, x' >$). It is calculating the dot product between each instance. In the case solving a dual form function, the mapping function here is the identity function and the dot product in the dual form occurs between the original instances ($< x, x' >$).

- ii) Your loss function (20 points)
  I have used "squared hinge loss"in the final classifier which is:
  $$L(y,\hat{y}) = \sum_{i=0}^{N} (max(0, 1 - y_i.\hat{y_i})^2)$$

2. Question 2

Describe a method to estimate your performance using an empirical method. Compare this estimate with a well known theoretical bound. Explain why/if there is a difference. (5 points) In order to calculate my generalization error, I have used a cross validation method in which I have splitted my data to 10 folds. In each iteration, I reshuffled the data with a fixed random seed and split them between 10 folds. I also ensure that relative class frequencies is approximately preserved in each train and validation fold. I then used the training splits for training the classifier and validation split to evaluate the model. Finally, I took the average of scores over all evaluation splits.

Results

Method	**Cross-validation Score**
Hinge loss, Linear kernel, one-vs-one (onv), ($\lambda/C = 8$)	87.65%
Hinge Loss, Linear kernel, one-vs-rest (ovr)	69.84%
Squared hinge loss, Linear Kernel, ovr	73.15%

For the second part of the question, I have used PAC bound formulas to get the lower and upper bounds.

Upper Bound - PAC Bound 2

$$\epsilon < \frac{1}{n}(T(x) + 4\log_e \frac{4}{\delta} + 4VC(h)\log_e \frac{2en}{VC(h)})$$

Lower Bound - PAC Bound 3

$$max(\frac{VC(h)-1}{32n}, \frac{1}{n}\log_e \frac{1}{\delta})$$

Results:

- One vs Rest, upper bound:
$$1/3372(905 + 17.52 - 18.36) = 0.268$$
which means 73.2% accuracy.
- One vs One, upper bound:
$$1/3372(423 + 17.52 - 18.36) = 0.125$$
which means 87.5% accuracy.
- Lower bound:
$$max(0.00007, 59.91) = 59.91$$
which means 40.08% accuracy.

The PAC bounds give the expected generalization errors and the upper bound might be higher than the empirical error. The empirical accuracies from corss-validation method are all between lower and upper bound. The upper bounds were very close to the empirical scores that show the power of PAC bound which means without using the cross-validation by only using these formulas we can get a very nice bounds; both upper and lower bounds (like a confidence interval).

3. Question 3

Submit your predictions on this test set, one prediction per line in the order given studentsdigits-test.csv. Preview the document (10 points) The results for one-vs-one approach has been saved in the 'result' folder with 'MaryZolfaghar_preds_Q3_OneVsOne.txt' filename and for one-vs-rest approach with 'MaryZolfaghar_preds_Q3_OneVsRest.txt' filename.

4. Question 4

Implement both types of transfer learning SVM (hypothesis and instance transfer) to train 1 vs 7 (target problem) by transferring in 1 vs 9 (source problem). Report your error estimate for the target problem with i) no transfer, ii) hypothesis transfer and iii) instance transfer. Which performs better? Why? (20 points)

Method	Cross-validation Score
No transfer with sklearn	87.01%
No transfer, target, with solver	87.53%
No transfer, source, with solver	88.93%
Hypothesis transfer with solver	89.26% (after tuning λ/C as a regularization term)
Instance transfer with solver	87.779% (transferred 11 support vectors on average)

To calculate the cross validation score in both types of transfer learning, I have used a StratifiedShuffleSplit cross validation model with 10 splits. I used one cross-validation for folding X1 data (i.e, digit 1s) between source and target problem. Then there is huge unbalanced data, so I have set test size to 50% and then assigned a weight to X1 data. Then I have used another cross validation model for the target problem which has 10% for the test size. According to the results, there was a little improvement for hypothesis and instance transfer learning compared to when I did not use any transfer learning. The reason can be the target problem is already has a lot of samples and the classifier can separate them with a quite high accuracy. However, the source problem has a higher accuracy and in hypothesis transfer when using weights from the source problem it is an informative extra information for the new problem. It is informative since in both problems digit 1s are positive samples. And we have the bottom half coordination of data points which is a bit similar for digit 7 and 9. So the information from the pre-trained source model is informative to use in the new target problem but it could not help to improve that much because in the target problem the classifier already can classify data with a quite high accuracy.

5. Question 5-Bonus Questions

Kernelize your approach for question 1). What Kernel do you use? What is your new error estimate? Submit a new predictions file for the data set (20 points) I have tried all of these kernels: ”RBF”, ”sigmoid”, ”polynomial”. The summary of results using cross-validation is in the following table and figures. The result for the test data using the best kernel (RBF with gamma=0.001) has been saved in folder ’results’ with ’MaryZolfaghar_preds_Q5_OneVsRest.txt’ filename.

Method	**Cross-validation Score**
polynomial kernel with ovo (degree=4)	95.95%
polynomial kernel with ovr (degree=4)	95.87%
RBF kernel with ovo (gamma=0.001)	96.93%
RBF kernel with ovr (gamma=0.001)	96.71%
Sigmoid kernel with ovo (gamma=1e-5)	78.45%
Sigmoid kernel with ovr (gamma=1e-5)	64.66%

Polynomial (Fig.1) and Sigmoid (Fig.3) kernels, one-vs-one can solve the problem better. But we can see with RBF kernel (Fig.2), even one-vs-rest method handled the problem quite well.

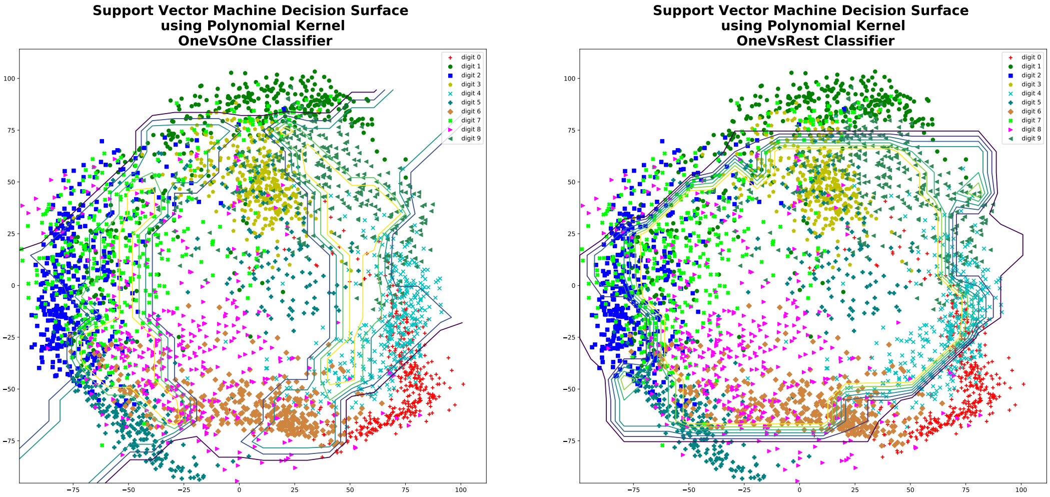

Figura 1. SVM results - Polynomial kernel with one-vs-one (left) and one-vs-rest approach (right)

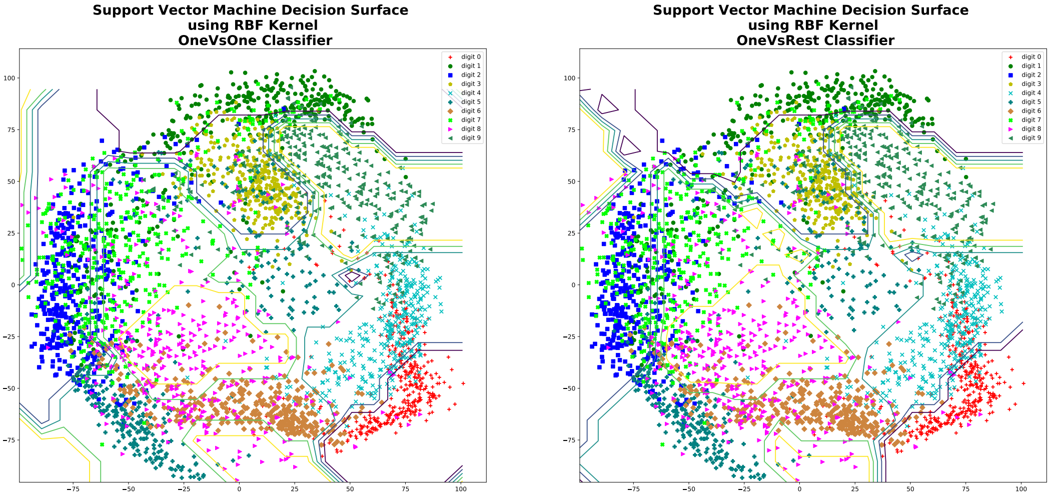

Figura 2. SVM results - RBF kernel with one-vs-one (left) and one-vs-rest approach (right)

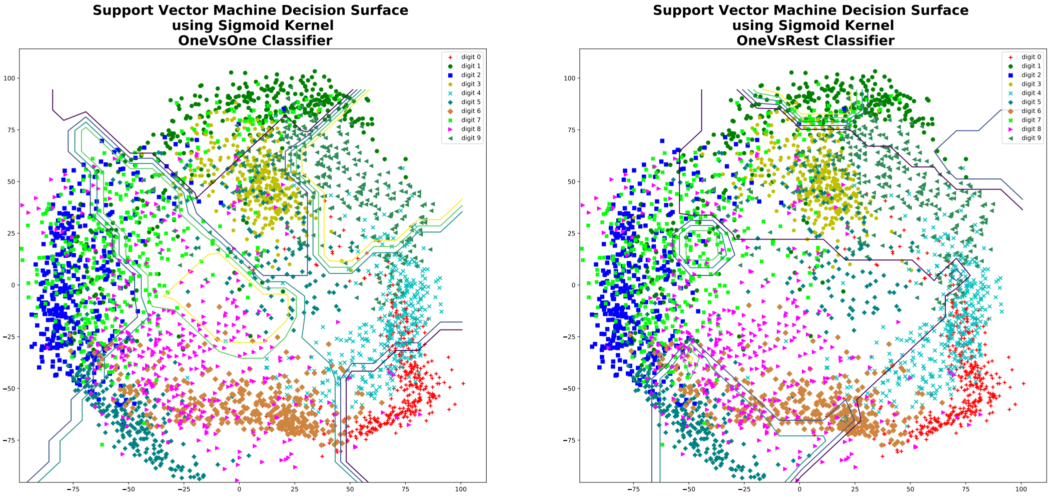

Figura 3. SVM results - Sigmoid kernel with one-vs-one (left) and one-vs-rest approach (right)