CSE291 HW3
Chao Yu A99049546
Wen Liang A53214852

Covtype:
At first we choose to use Minkowski distance metric
For 2 folds, 5 folds cross validation, we find out when K=1, the cross validation error for 50 percent, 75 percent and full data set are all the lowest.
For leave one out, since it takes too much time run the dataset, we decide to choose 50000 training data to do leave one out cross validation. The result is the same as fold cross validation, the accuracy is the highest when K=1.
For K=1, the average accuracy for full data, 5 folds is 96.2754%, 2 folds is 95.6246%.
The average accuracy for 75 percent data, 5 folds is 95.9463%, 2 folds is 95.0502%
The average accuracy for 50 percent data, 5 folds is 95.1445%, 2 folds is 93.8332%

When we use K=1 to the test data, the accuracy is 96.4517%, the error is 3.5483%. The accuracy for the test set is higher than the cross validation accuracy except for full data, F=5. In addition, from the accuracy shown above, 5 fold cross validation is better than the 2 fold cross validation.

Then we choose ellucian metric.
Using 5 folds cross-validation, we find out best K is is still 1.

From the result, we can see cross validation for full data performs best. The larger the training data is, the more accuracy the validation set and test set will be.

The method I choose the best k is: when I do the cross validation, I test different values of k each time, then I can get the average accuracy for each k. My testing k is: 1, 3, 5, 8, 10, 12, 15, 20, 50, 100 and using Minkowski distance metric and euclidean metric. We find out using Minkowski distance metric and euclidean metric, the accuracy does not change very much.

SVM:
We randomly pick 5000 training data and as our data sample after doing the feature selection. The features decreases from 54 to 11.
Then we use cross validation to test the validation set with four different kernels, which are Linear,Polynomial,RBF and Sigmoid. For each kernel, we do cross valid with fold =2 and 5. We find out the best kernel is linear, with the average accuracy 65.7% with 5 folds and average accuracy 65.3% with 2 folds. The CV errors are 34.3% and 34.7% correspond to the two accuracies above. The test accuracy is 67.0% , which is slightly higher than the validation set.

Same as KNN, when F=5, the result is better. The accuracy is also related with the size of training data. Full data performs best.

MNIST - KNN

1. Using the training data, do F-fold cross-validation and find an optimal K

- Minkowski Distance:
- Full data, 5-folds:
- mean of k=1,accuracy=0.928800
  mean of k=3,accuracy=0.927400
  mean of k=5,accuracy=0.929000
  mean of k=8,accuracy=0.923200
  mean of k=10,accuracy=0.920400
- Full data, 2-folds:
  mean of k=1,accuracy=0.913000
  mean of k=3,accuracy=0.911600
  mean of k=5,accuracy=0.913600
  mean of k=8,accuracy=0.907200
  mean of k=10,accuracy=0.903800
- Full data, leave one out:
  mean of k=1,accuracy=0.933000
  mean of k=3,accuracy=0.935000
  mean of k=5,accuracy=0.931400
  mean of k=8,accuracy=0.928400
  mean of k=10,accuracy=0.927800
- 50% data, 5-folds:
  mean of k=1,accuracy=0.908800
  mean of k=3,accuracy=0.906400
  mean of k=5,accuracy=0.910400
  mean of k=8,accuracy=0.903600
  mean of k=10,accuracy=0.901200
- 50% data, 2-folds:
  mean of k=1,accuracy=0.896800
  mean of k=3,accuracy=0.888800
  mean of k=5,accuracy=0.885200
  mean of k=8,accuracy=0.877600
  mean of k=10,accuracy=0.875600
- 50% data, leave one out:
  mean of k=1,accuracy=0.917600
  mean of k=3,accuracy=0.918000
  mean of k=5,accuracy=0.920800

mean of k=8,accuracy=0.911600
mean of k=10,accuracy=0.908800

- 75% data, 5-folds:
- mean of k=1,accuracy=0.921333
  mean of k=3,accuracy=0.921067
  mean of k=5,accuracy=0.920000
  mean of k=8,accuracy=0.919467
  mean of k=10,accuracy=0.917333
- 75% data, 2-folds:
  mean of k=1,accuracy=0.909067
  mean of k=3,accuracy=0.903467
  mean of k=5,accuracy=0.906933
  mean of k=8,accuracy=0.900800
  mean of k=10,accuracy=0.894933
- 75% data, leave one out:
- mean of k=1,accuracy=0.925867
  mean of k=3,accuracy=0.928533
  mean of k=5,accuracy=0.926933
  mean of k=8,accuracy=0.926400
  mean of k=10,accuracy=0.922667
- Euclidean Distance:
- Full data, 5 folds
- mean of k=1,accuracy=0.928400
  mean of k=3,accuracy=0.926000
  mean of k=5,accuracy=0.928000
  mean of k=8,accuracy=0.920200
  mean of k=10,accuracy=0.917000

The best model is minkowski distance with k = 5, which get the best result in 5 fold cross-validation.

2. Use the best K and its associated TD and compare the error rate to CV error

The result is 0.935.

The mean error rate of leave one out cross-validation is closest to the error rate on test set. This is because it considers more about the variance in the training set and gets a more robust result than other choices of F. Then its expectation of error is closer to the test set and to other data than other methods. However, it cost too much time to compute.

3. Report the stability of the result with respect to the size of TD.

The results from full data and 75% data is pretty close to each other but the KNN classifiers using 50% data have lower accuracy. We can see that the performance of KNN algorithm is highly depends on the size of training data. The more data they have, they can get a better generalization and a more stable result. However, if the current data is big enough, the improvement can be smaller and smaller, just like the example of full data and 75% data.

4. The methodology for choosing the best K and distance metric.

We need to use the cross-validation to evaluate machine learning models. Although the leave one out method is more robust, we still need to consider the efficiency in the experiment. Then,

we should choose a reasonable number of folds K (we choose 5 here) to try each model and get a mean error rate for each folds. We use this mean error rate to evaluate models and choose best K and distance metric.

5. Bias and Variance

For classifiers:

Bias = L(y, f(x))

Variance = E[L(h(x), y)]

When K is large, KNN algorithm considers more votes from neighbors and get a smaller bias according to the central limit theory (more data you have, closet to the true distribution). However, it will suffer from a variance and vulnerable to a more unstable dataset. When K is smaller, KNN algorithm considers less votes and get a smaller variance. However, the K is pretty limited and this may introduce high bias to the model.

MNIST - SVM

1. Using training data, do F-fold cross-validation(CV)

- 5-fold, linear kernel:
- accuracy: 0.886416666667
- 2-fold, linear kernel:
- accuracy: 0.863458781333
- 5-fold, polynomial kernel (gamma = 0.001 C = 10.0):
- accuracy: 0.954333333333
- 5-fold, gaussian kernel (PCA, gamma = 1e-06 C = 1000):
- Accuracy: 0.962583333333
- 5-fold, sigmoid kernel (PCA, gamma = 1e-09 C = 1000):
- Accuracy: 0.8915
- The best kernel and best parameter gaussian kernel with gamma = 1e-06 C = 1000

2. Test the test set error with best kernel and parameter and compare the error rate to CV error

- 0.966285714286
- This result is very close to the result of 5-fold cross-validation

3. The stability of result with respect to the size of the TD.

The SVM is not so sensitive to the size of training data they used. The SVM is just trying to find optimal boundaries between classes according to the hinge loss. When the size of training data is small but linearly separable, the SVM still has very good result.

4. The methodology for choosing the best kernel and parameters

The method is similar to the previous one, use cross validation and evaluate by mean error of each fold.

Comparison between KNN and SVM:

1. Accuracy. For linearly separable problems, SVM with linear kernel can give us a better result. For more complex conditions, KNN can give us a very good result. The SVM with different kernels also provides SVM ability to solve more complex problems.

2. Training time. The KNN has rather small training time. It just put the training set into the memory. The SVM need to compute optimal boundaries for each class and spends more time. Moreover, the SVM with polynomial kernel cost a great amount of time.
3. Test time. The test time of KNN is notoriously long because KNN has to compare this test sample with each prototypes which is O(number of TD) time. However, SVM just uses its boundaries to classifier test samples only cost constant time.
4. Cost of memory. The KNN need to save all the prototypes in the memory but SVM only save some weights and parameters.