Machine learning HW6 report

Q1. Since there are some parameters you need to tune for, please check how to do grid search for finding parameters of best performing model. For instance, in C-SVC you have a parameter *C*, and if you use RBF kernel you have another parameter γ, you can search for a set of (C,γ) which gives you best performance in cross-validation.

For LIBSVM package, grid.py file is provided to do this job. In SVM, the most annoying thing is to tune the hyper-parameter. However, there's no more quick way to find but try one by one. First, you can give the input to grid.py, and it will try each parameter pair and evaluate by cross validation.

Conversely, this process is very time consuming. It's not effective if I use the whole data to get the result. As the result, I choose 0.1 part of whole data (in this homework, 50 data points are selected) randomly to tune the hyper-parameter. After I get the best c and γ, the whole data will be used to train the final SVM.

Q2. first use PCA to project all your data onto 2D space - use different colors to draw samples with different digit class

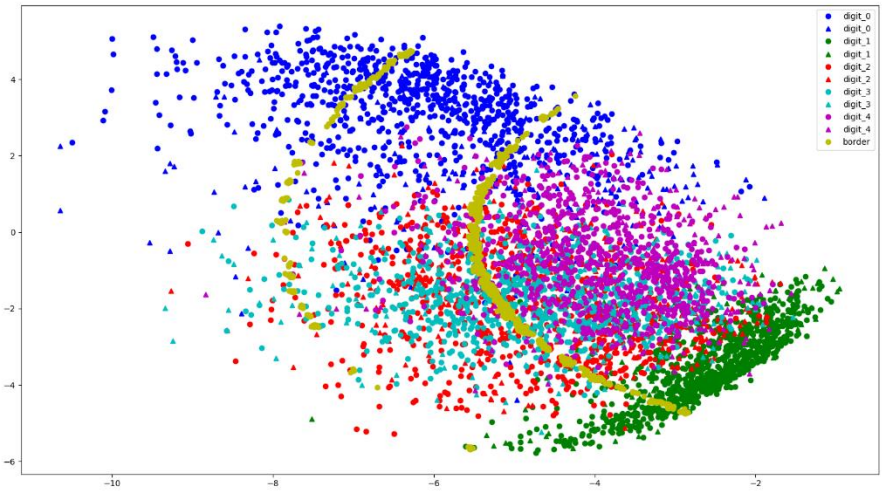

Figure 1. My final visualization results

The corresponding result is shown in Figure 1. The green, shallow blue, deep blue, red and pink are the data points which are represent the different digits after PCA. For PCA, the projected basis is referred by the data distribution, and it cannot separate the distribution which just like t-SNE. As the result, the performance of visualization is bad toward MNIST, and there're too crowded that it's hard to see the boundary.

Q3. With all the data samples are shown by "dots", the "support vectors" should be shown with different symbols, e.g. square, triangle, cross

In Figure 1, the support vectors are marked as triangle symbols. After training, we can get the support vector from the model file. However, its' a little troublesome that there're two questions in this task. First, there's slight difference between the feature of original data and the feature which model file store. If we just map toward the original array, the index of support vector cannot be found. As the result, we should do the quantization by ourselves, get the approximate one and regard it as the original support vectors.

Second, in order to save the size of model, the model adopt sparse method to store the feature of support vector. There're only non-zero value will be stored, and the other feature will be discarded. For our work, we should recover as array representation, and it can be processed furthermore.

Q4. You can also try to plot decision boundary:)

In LIBSVM, it's not trivial to get the decision boundary directly. After my experiment, I fail to draw the boundary at first. As the result, I try to do the other two experiments.

The first idea I thought is trying to project the grids of points in 2D space into original feature space, and do the SVM classification to check if there're labels difference between nearest points. If the different label exists, we can regard it as the decision boundary. However, there's a critical issue which lead my failure: the information missing. After I try to do the reverse of PCA, the orthogonal projection matrix is a square matrix whose dimension is the number of feature.

However, the dimension of points is only 2 since they are generated in 2D space firstly. To recover the missing dimension, I choose to fill the value randomly. However, the PCA is data driven projection method, and I cannot recover the data appropriately. As the result, I cannot get any points which neighbor has different labels.

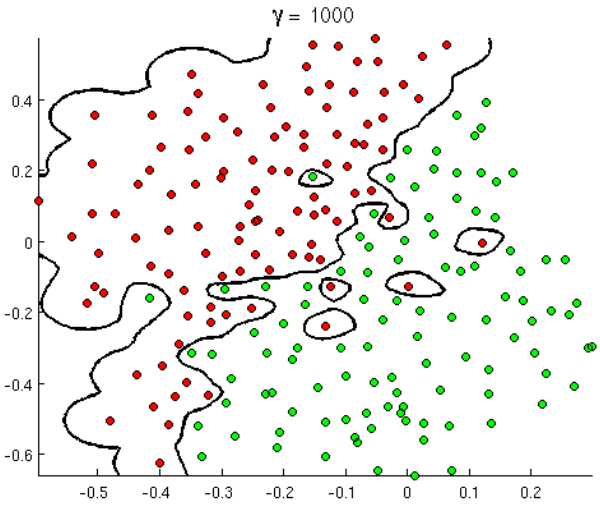

Figure 2. The ideal phenomenon of RBF kernel

The second idea I try is trying to generate some points near the training data, and project as the usual PCA order. The generated points are the original training points with some random noise. In my work, RBF kernel is adopted, and the shape of projected RBF boundary is just like the circle around the training data points. The ideal result is shown in Figure 2. The distance that the boundary to the centered data points is γ. However, there're lots of information missing during PCA, so the shape of the boundary is distorted, and it didn't act as normal circle.

However, there're two problems in my approaches. First, the distribution of generated data points will be also distorted in 2D space. Even though the data points are generated around hyper-ball in hyper-space, it will be squeezed which just like some duplicated points. Second, the generated points might not have corrected predict label. By these two reasons, the boundary that I get isn't perfect. I use deep yellow (土黃色) color to represent the boundary in Figure 1.

Q5. submit a report

In this homework, I learn how to use LIBSVM which is a well know toolkit to perform support vector machine classification. Second, I experience how to use cross validation mechanism to tune the hyper-parameters. The PCA technique is also adopted to visualize the result. At last, I try some methods to describe the boundary.

SVM is a powerful algorithm. By using LIBSVM, we can demonstrate SVM algorithm easily. However, by learning the theory and knowledge behind this algorithm, I can choose the appropriate parameters and use SVM with more efficiency. After cross validation, around 97% accuracy can be obtained.