# Machine Learning Homework3 Report

## 1. SVM vs. Neural Networks

### Data A:

#### Preparation

In Data A set, I choose iris dataset and breast cancer dataset to compare the classification effect of SVM and MLP in small data set. The kernel of SVM I choose Linear kernel and RBF kernel. The parameters of MLP I use default parameters of MLP from sklearn library.

In experiments, I finally get a similar result of iris dataset and breast cancer dataset. In this case, I would only initialize the iris result and analyse it in details.

Iris dataset's size is around 800. And each feature has 4 dimension. For every classification method, I vary the test set size from 0.1 to 0.9 and dimension reduction from 0 to 3.

#### Experiment

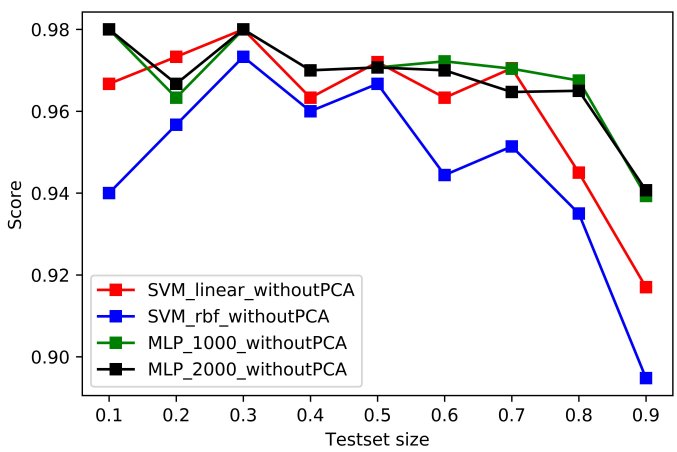


*Figure 1: All score without PCA*

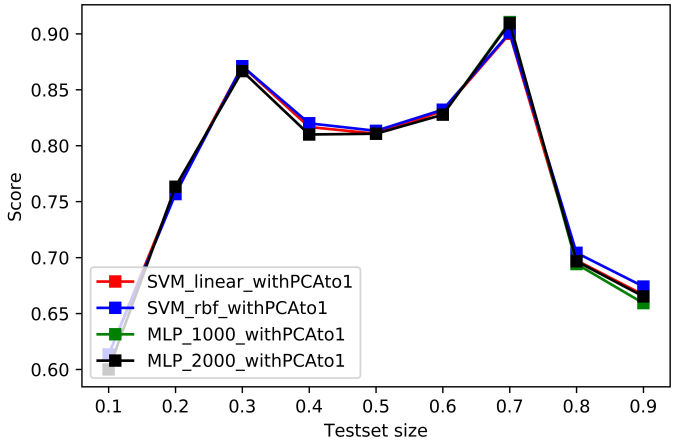



*Figure 2: All score with PCA to 1*

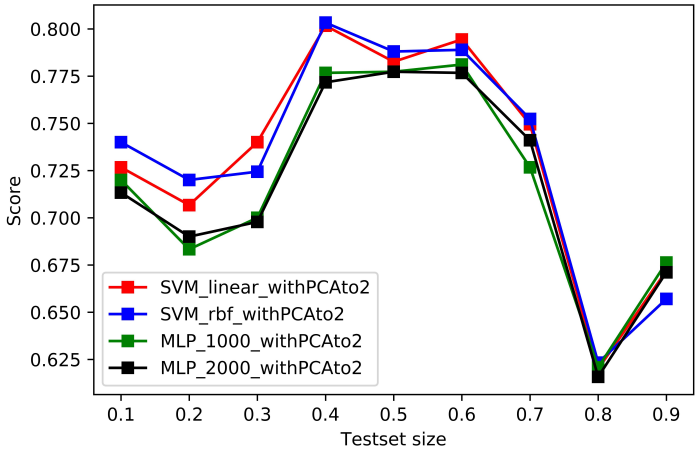



*Figure 3: All score with PCA to 2*

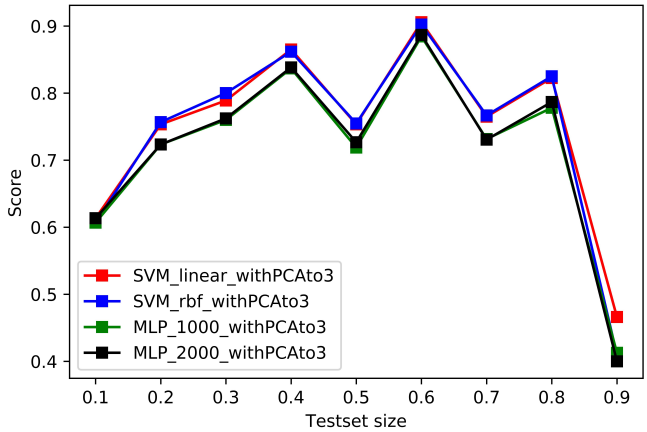



*Figure 4: All score with PCA to 3*

Figure 1,2,3,4 show the score of each classification algorithm with different size of test set. We can find when the test set's size is in 0.3 to 0.6 which means training set's size is in 0.7 to 0.4, the score of each algorithm is in a high level. And when the test set's size achieves 0.8 and 0.9, the score drops rapidly. In this case, we can conclude that when the training set accounts for 50% to 70%, this model can have a high score, which means it can output a good model.

Compare 4 methods in this case, we find all the 4 methods have a similar score in this situation. So we can conclude that when the dataset is small, SVM and MLP can both have a good classification result.

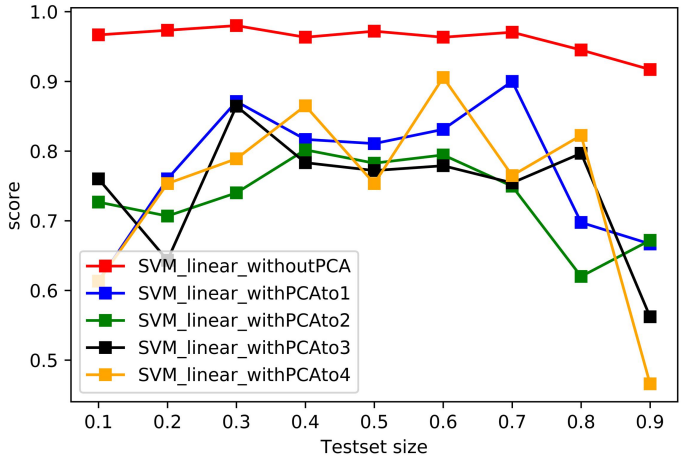



*Figure 5: SVM with linear kernel*

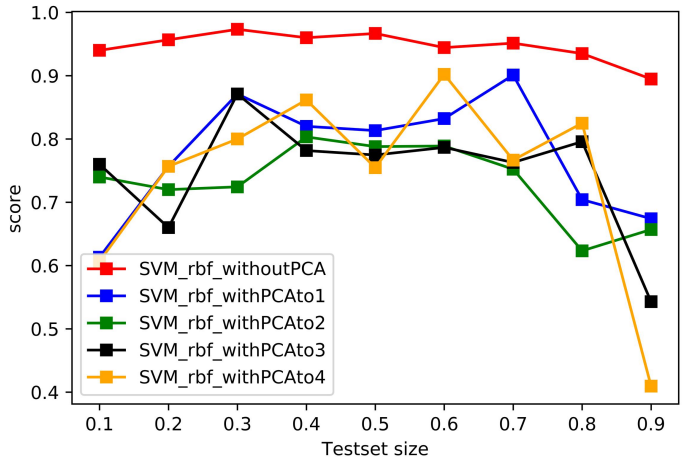



*Figure 6: SVM with RBF kernel*

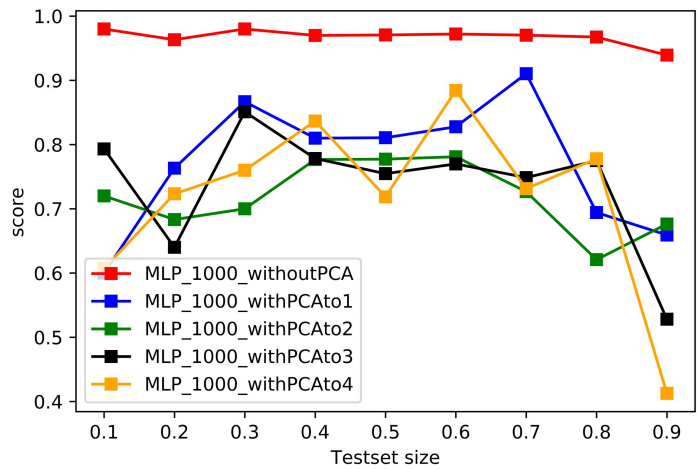



*Figure 7: MLP runs 1000 iteration*

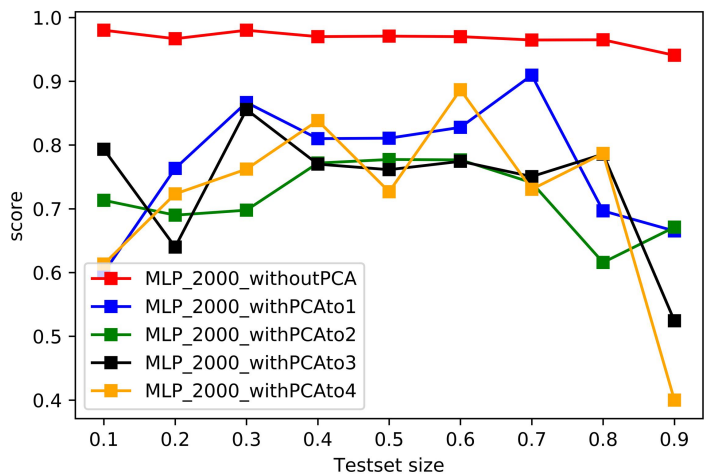



*Figure 8: MLP runs 2000 iteration*

In this part, I compare each algorithm with different dimension. We can find that the highest score is the red line which is without PCA effect. So in this situation, we can conclude that when the dimension of data set is small, retaining dimensions as much as possible is a better method to finish classification task.

The result of other data set which is breast cancer is similar to this result. In order to not place to much photos, I would like not to place rest photos in this report.

In conclusion, when the data set is small, SVM algorithm and neural network such as MLP have a similar ability to classify the data. However in the big dataset, neural network has a much better classification than SVM, I will show the detail of that below.

## Data B:

### Preparation

In Data B set, I choose MNIST dataset to compare the classification effect of SVM and Neural Network. The kernel of SVM I choose Poly kernel and RBF kernel. The Neural Network I use a CNN which contains two convolution layers and two FC layers which the last FC layer is a soft—max layer.

First I will introduce the MNIST dataset. This dataset consists of handwriting numbers and labels. We can train a SVM model and a CNN model to assess the accuracy of each methods.

In this experiment, I apply SVM and CNN on the dataset.

Each single sample is a 28 pixels * 28 pixels picture which is a 784 dimensions sample. And the dataset contains 55000 samples. So I would vary the sample size from 10000 to 55000 and step by 5000 in SVM.

In CNN I will change the times of iteration to figure out whether the accuracy is positively correlated with the number of iterations.

### Experiment

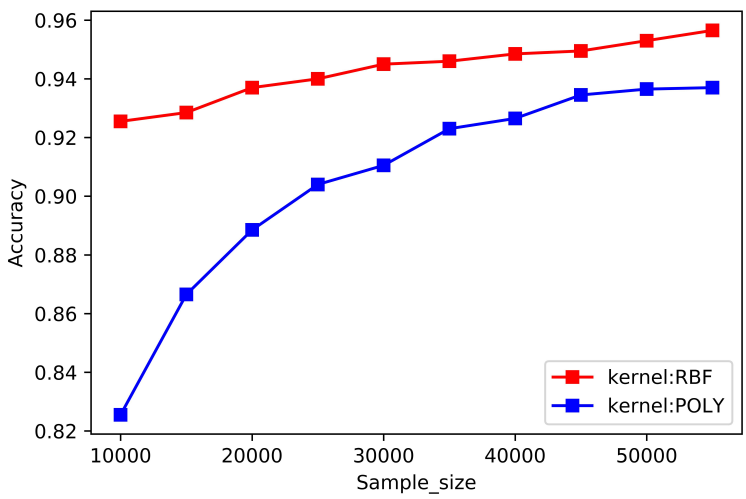



*Figure 9: Accuracy*

Figure 9 shows us how the accuracy changes with the change of sample size.

It shows that both accuracy with kernel RBF and accuracy with kernel POLY increase with sample size becoming larger. It's obvious that the RBF has a more accuracy compare with POLY. So it can conclude that in this situation RBF kernel plays better than POLY kernel.

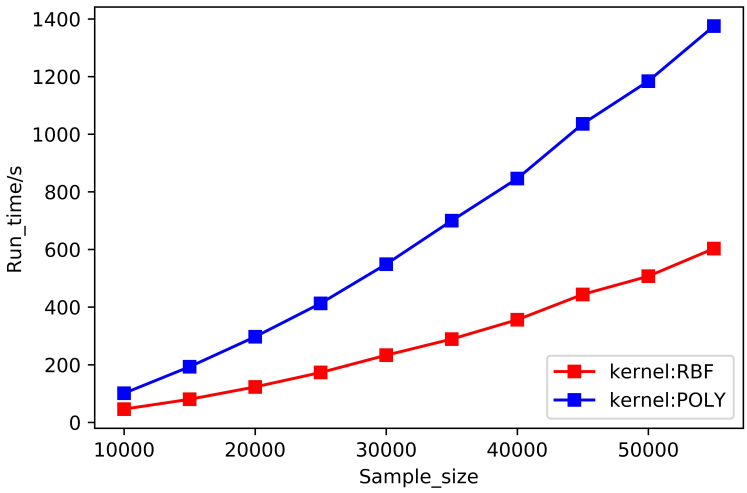



*Figure 10: Run time*

Figure 10 shows that different kernel function has a different time to calculate the results. With the increase of sample size, the running time of POLY kernel increases faster than that of RBF. When sample size becomes 55000, POLY kernel needs nearly 1400 seconds however RBF kernel just need 600 seconds.

From figure9 and figure10 we can see POLY kernel runs longer but with a worse accuracy. So in this experiment we are supposed to choose RBF as the kernel function.

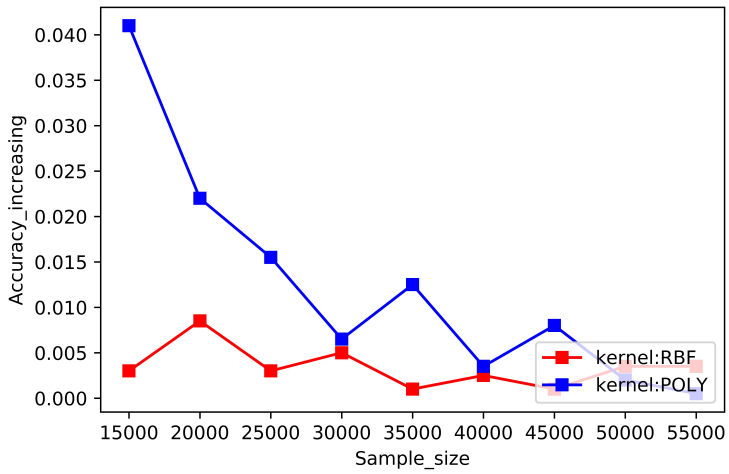



*Figure 11: Accuracy increasing*

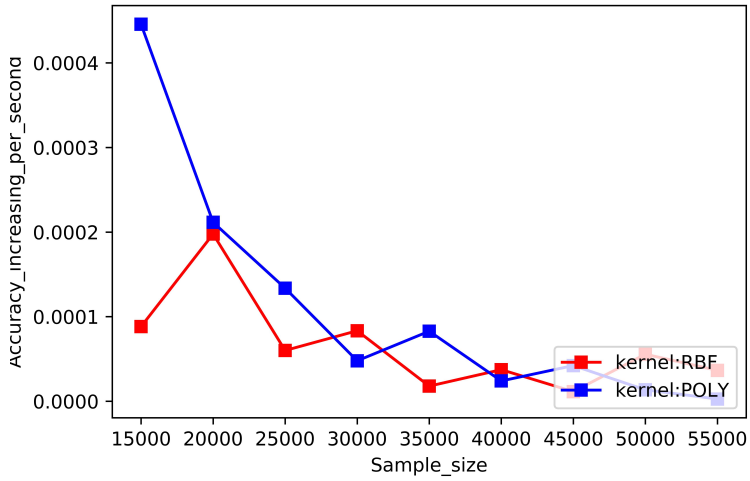



*Figure 12: Accuracy increasing per second*

Figure 11 shows how the accuracy increase with the sample size increase per 5000. Because of low accuracy of POLY from start, the accuracy increase rapidly in first 5000 samples. However, the accuracy increasing of RBF is much stable. In this situation we can conclude that with the sample size growth the accuracy increasing rate becomes lower.

Figure 12 shows that when sample size exceeds 25000, the increasing accuracy per second is less and less so that the efficiency is low.

In this experiment, SVM classification method's best accuracy is around 96%, exactly the highest accuracy is 95.65%. Next I will show the result of a simple CNN network.

The neural network I choose shows below.

*Figure 5: Neural network*

This neural network contains two convolution layers and two pool layer which use max pooling. Follow are two FC layers and the output layer I choose a soft—max layer. The times of iteration I vary from 1000 to 20000 to test its accuracy with the increasing of iteration times.

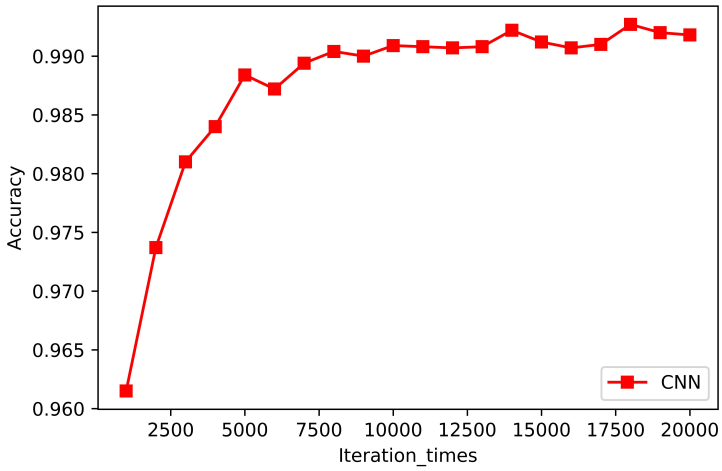



*Figure 13: Iteration times_Accuracy*

Figure 13 shows that the accuracy increases rapidly in the small times of iteration. And when the iteration times achieve around 7000, the accuracy growth rate tends to be stable. However, we can see the worst accuracy of CNN is still around 96%, interestingly the highest accuracy of SVM classification is less than 96% and take about 600 seconds however CNN just takes 85 seconds.

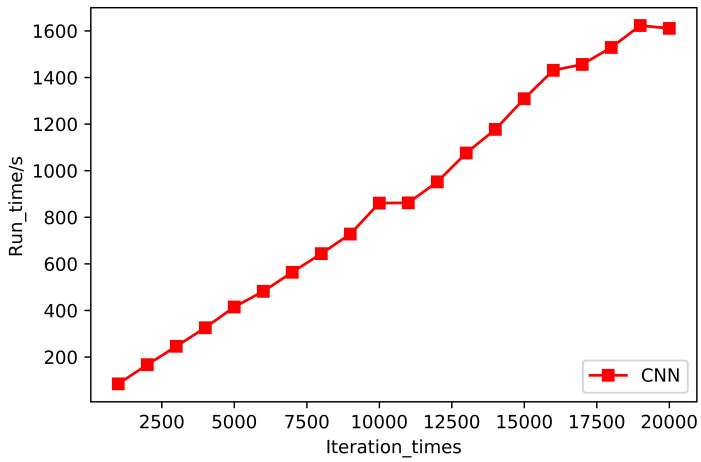



*Figure 14: Iteration times_run time*

Figure 14 shows that the running time increases linearly with the times of iterations. That's because every single iteration time takes a same time.

## Comparison

In the experiment above, it's obvious that CNN classification has a better

accuracy than SVM classification methods. The worst accuracy of CNN is still better than the best accuracy of SVM(both RBF kernel and POLY kernel).

The sample size is around 55000. It's not a too big data set. However, SVM does not perform well in this classification task. 95% accuracy is not a good classification accuracy. Because CNN almost reach 99% accuracy just takes 600 seconds. So I don't think SVM has any strengths in classify big data sets because the smallest time complexity of SVM is $O(n^2)$. I think SVM would has a more great performance in small data sets.

When it comes to neural network, I think it is a great method to classify data in both big data sets and small data sets because its run time depends on its iteration times linearly.

## 2. Casual discovery algorithms

In this section, we are supposed to implement a casual discovery algorithms on a real world problem. In order to finish this task, we have some preparation to do. First, I choose Tetrad as mainly tools. And then I gather data from a website which is data.world. After compare each data, I choose wiki4HE from UCI as my data set. The introduction of this data set will given by the link to this data set.

The brief of this data set is Ongoing research on university faculty perceptions and practices of using Wikipedia as a teaching resource.

In this task, I choose FCI algorithm to solve this problem. Below is a brief introduction of FCI algorithm.

第一步，构造一个非定向边构成的完全图，对于顶点 $X_i$ 和 $X_j$，在给定 $X_i$ 的邻接子集 $M$ 条件下，对两个顶点做条件独立检验，如果条件独立关系成立，则移除 $X_i$ 和 $X_j$ 之间的边，并且把相应的邻接子集 $M$ 保存到分离集 $sepset(X_i, X_j)$ 和 $sepset(X_j, X_i)$ 中。第一步得出的骨架是最终骨架的超集。

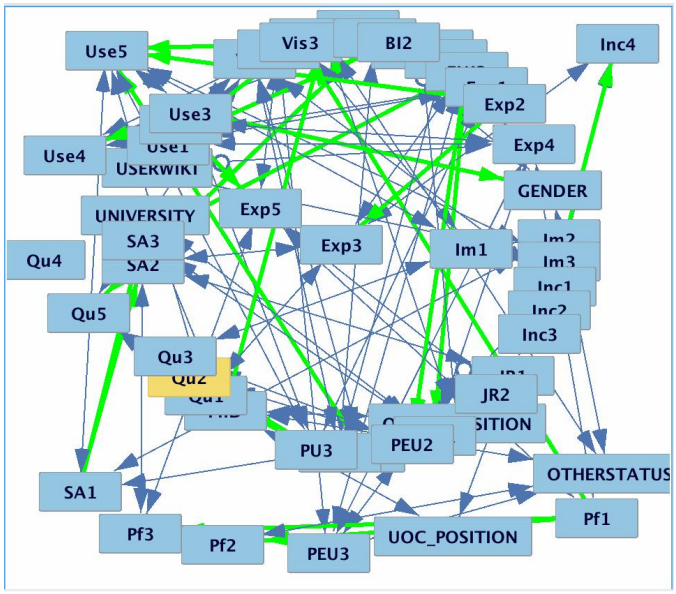
第二步，对于开放三元组 $\langle X_i, X_j, X_k \rangle$，若 $X_j$ 不在 $sepset(X_i, X_k)$ 和 $sepset(X_k, X_i)$ 中，则把此三元组定向成 $v$ -- 结构。

第三步，对于 $X$ 中所有顶点 $X_i$，找出它的 $pds$ $(X_i, \cdot)$。然后对于所有 $X_i$ 的邻接顶点 $X_j$，检验 $X_i \perp X_j | (Y \cup S)$ 是否存在，$Y \subseteq pds(X_i, \cdot) \setminus \{X_i, X_j\} \cup pds(X_j, \cdot) \setminus \{X_j, X_i\}$。若存在则移除两顶点之间的边，并把相应的邻接子集 $M$ 保存到分离集 $sepset(X_i, X_j)$ 和 $sepset(X_j, X_i)$ 中。这样就破坏了图的骨架，需要重新定向 $v$ -- 结构。所以最后把所有的边重新改为非定向边。

第四步，重复第二步，重新定向 $v$ -- 结构。

第五步，利用 ZHANG,J.(2008) 中的定向准则，尽可能多的给边定向。

After implement FCI algorithm on Wiki4HE data set, I finally get a model which describes the casual relationship of all the parameters.




This is a brief review of the model. The arrow means casual relationship of different parameters. In this case, we can find each parameter whether has a casual relationship of other parameters.