Classification

In this section, we implement multi-class classification models on the Fashion-MNIST dataset. We first use two algorithms, k-Nearest Neighbor (k-NN) and random forest, which can be directly used for multiclass classification. Then, we extend Support Vector Machine (SVM), which is a binary classifier, to handle the multiclass case.

k-Nearest Neighbors

We first implement k-NN classification model to classify the Fashion-MNIST data. The Euclidean distance is used to measure the distance between variables. The raw data are scaled in the data pre-processing step, so the distance is scale-invariant.

To determine the parameter k, we use 5-fold cross-validation and calculate the overall mis-classification rate for each k. The figure shows the relation between mis-classification rate and k. We find that the best mis-classification rate is reached at $k = 10$.

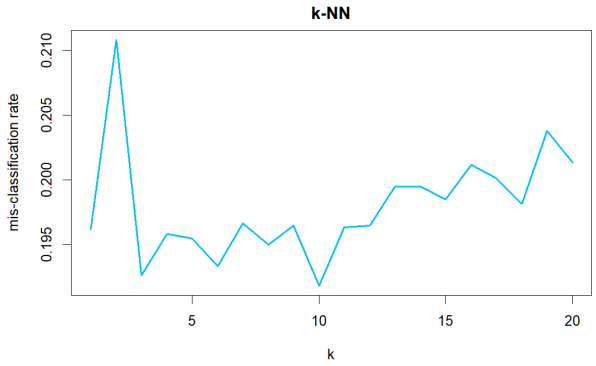

Figure 1: 5-fold CV Mis-classification Rate

Based on the result of cross-valiadation, we choose $k = 10$ to fit the k-NN model. The confusion matrix and mis-classification rate for each class are shown in the following two tables. For Fashion-MNIST data, the overall mis-classification rate of the 10-NN classifier is 0.144 and the accuracy is 0.856.

	Ytest_pred									
Ytest	0	1	2	3	4	5	6	7	8	9
0	856	1	18	17	4	0	95	1	8	0
1	3	964	9	13	1	0	10	0	0	0
2	11	0	794	11	97	0	86	0	1	0
3	30	8	13	888	37	0	24	0	0	0
4	2	0	74	22	810	0	90	0	2	0
5	0	0	0	1	0	788	10	121	6	74
6	191	1	106	19	71	0	601	0	11	0
7	0	0	0	0	0	2	0	936	0	62
8	2	1	15	3	5	0	11	6	955	2
9	0	0	0	0	0	1	1	31	0	967

(a) Confusion Matrix

class	mis−classification
0	0.218
1	0.011
2	0.228
3	0.088
4	0.21
5	0.004
6	0.352
7	0.145
8	0.028
9	0.125
overall	0.144

(b) Mis-classification Rate

Figure 2: 10-NN

Random Forest

We then classify the Fashion-MNIST data using the random forest. The confusion matrix and mis-classification rate for each class are shown in the following two tables. The overall mis-classification rate of random forest is 0.115 and the accuracy is 0.885. The results show that random forest has a better performance than 10-kNN classifer.

	Ytest_pred_tree									
Ytest	0	1	2	3	4	5	6	7	8	9
0	859	0	12	31	0	1	84	0	13	0
1	1	972	5	16	1	1	4	0	0	0
2	8	1	804	11	114	0	52	0	10	0
3	18	7	7	931	20	0	17	0	0	0
4	1	0	58	31	862	0	45	0	3	0
5	0	0	0	0	0	947	0	36	5	12
6	167	1	96	27	73	0	619	0	17	0
7	0	0	0	0	0	17	0	928	0	55
8	1	1	10	0	3	1	6	2	975	1
9	0	0	0	0	0	7	1	40	3	949

(a) Confusion Matrix

class	mis−classification
0	0.186
1	0.01
2	0.19
3	0.111
4	0.197
5	0.028
6	0.252
7	0.078
8	0.05
9	0.067
overall	0.115

(b) Mis-classification Rate

Figure 3: Random Forest

SVM

We have applied two multiclass classifiers k-NN and random forest for the Fashion-MNIST classification. Now, we extend SVM to the multiclass form to solve the classification problem. SVM is a binary classifier and it does not support multiclass classification natively. However, we can break the multiclass classification problem into several binary ones. There are two common methods to extend SVM for multiclass classification, One-vs-One approach and One-vs-Rest approach.

In One-vs-One approach, we fit SVM models for every two classes. Each classifier separates points of two different classes. Suppose we have k classes, we then fit $\frac{k(k-1)}{2}$ SVM models. In prediction stage, we input data into all binary classifiers. Each binary classifiers will decide a class that the input is belonged to. We let those binary classifiers vote for the class of input, and the prediction result is the class that most classifiers vote for.

In One-vs-Rest approach, we fit SVM models to distinguish points of one certain class from the other classes. Suppose we have k classes, we then fit k SVM models. In prediction stage, we input data into all One-vs-Rest classifiers. Each classifier will give a probability that the input is belonged to that class. The prediction result is the class with the largest probability.

One-vs-Rest approach is more computationally efficient than One-vs-One approach, since it only need to fit k SVM models while One-vs-One approach need to fit $\frac{k(k-1)}{2}$ models. However, in One-vs-Rest approach, the training data is unbalanced since the ratio of training data from each class is $1 : (k-1)$, which may cause biase.

We fit the multiclass SVM model using One-vs-One approach, with using radial basis kernel. The overall mis-classification rate of kernel SVM is 0.092 and the accuracy is 0.908. The kernel SVM classifier outperforms k-NN and random forest.

In addition, a linear SVM model is constructed to compare with the RBF kernel. The overall mis-classification rate of linear SVM is 0.188 and the accuracy is 0.812.

Ytest	Ytest_pred_svm 0	1	2	3	4	5	6	7	8	9
0	878	0	13	18	1	0	81	0	9	0
1	3	987	1	7	0	0	1	0	1	0
2	16	1	834	14	74	0	59	0	2	0
3	24	7	8	919	26	0	15	0	1	0
4	3	1	58	22	877	0	38	0	1	0
5	0	0	0	0	0	952	1	27	6	14
6	123	1	52	27	55	0	732	0	10	0
7	0	0	0	0	0	10	0	961	0	29
8	4	0	4	3	2	2	4	1	980	0
9	0	0	0	0	0	5	0	29	3	963

(a) Confusion Matrix

class	mis−classification
0	0.165
1	0.01
2	0.14
3	0.09
4	0.153
5	0.018
6	0.214
7	0.056
8	0.033
9	0.043
overall	0.092

(b) Mis-classification Rate

Figure 4: Kernel SVM (rbf)

Though linear SVM model does not perform better than k-NN and random forest, SVM with RBF kernel have a quite good performance on classification, which achieves an overall accuracy of 90.8%.

Ytest	Ytest_pred_lsvm 0	1	2	3	4	5	6	7	8	9
0	773	8	18	35	5	1	135	0	23	2
1	3	980	2	12	2	1	0	0	0	0
2	33	1	725	15	106	0	112	0	8	0
3	78	32	13	822	28	0	26	0	1	0
4	3	0	173	40	688	0	94	0	2	0
5	2	0	0	1	0	890	0	65	3	39
6	193	5	144	44	87	0	509	0	17	1
7	0	0	0	0	0	63	0	886	1	50
8	9	0	18	4	6	7	22	5	928	1
9	0	0	0	0	0	23	0	56	0	921

(a) Confusion Matrix

class	mis−classification
0	0.165
1	0.01
2	0.14
3	0.09
4	0.153
5	0.018
6	0.214
7	0.056
8	0.033
9	0.043
overall	0.188

(b) Mis-classification Rate

Figure 5: Linear SVM

Conclusion

In this part, we build k-NN, random forest, linear SVM and kernel SVM to classify Fashion-MNIST data. The performace of different algorithms are shown in the table. The accuracy of four methods: kernel SVM > random forest > k-NN > linear SVM.

Table 1: Model Summary

	kNN	Random.Forest	Linear.SVM	Kernel.SVM
Accuracy	0.8559	0.8846	0.8122	0.9083