Handwritten Digits Recognition via Principal Component Analysis

Zhao Yuqi
Department of Mathematics
HKUST

CONTENTS

Handwritten Digits Recognition via Principal Component Analysis

***Abstract*—This project aim to recognize handwritten digits (0-9) by Principal Component Analysis(PCA) combined with some other technical tools. Specifically, it includes three different methods, based on PCA combined with distance from subspaces, logistic regression, and support vector machine(SVM) respectively. The experiment result indicates that the chain of PCA and SVM get the best performance among the three method, which archives above 98% accuracy.**

I. Introduction

Handwritten recognition of numeric digits has great importance in those situations such as recognizing zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand and so on.

The challenges of solving this problem are from two different aspects. The handwritten digits are of different size, thickness, or orientation and position relative to the margins. And the dimensions of the data might be very high as it is image data with at least hundred of pixels.

In this project, PCA was used to deal with those challenges simultaneously. Benefited from PCA, we can obtain low-dimension features of the data, which was tractable to analysis furthermore.

It must be mentioned that in this project, we actually used PCA in two different approaches: within data of each digits, and among the whole data. In previous case, we implement PCA algorithm to each different digit thus we gain ten affine spaces which can be regarded as features for each digit. For the later case, we implement PCA algorithm among the whole data, which is classical unsupervised dimension reduction method.

After having feature of the digits, some classical classification algorithms can be implemented to classify the image. Implementations based on distance, logistics regression and SVM was presented. we compared those three methods and finally result the pipeline of PCA and SVM obtain the best result.

II. Dataset and Set Up

Our experiments based on the dataset from course MATH6380: A Mathematical Introduction to Data Analysis, instructed by YUAN YAO. It could be downloaded on

www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/zip.digits/.

The dataset contains 7291 data of digits 0-9. Each example is a 256 vector, that can be easily convert into 16×16 gray image. Picture 1 shows some samples of image data. We split data of each digits, using 70% of each digit data for training, and the rest for test.

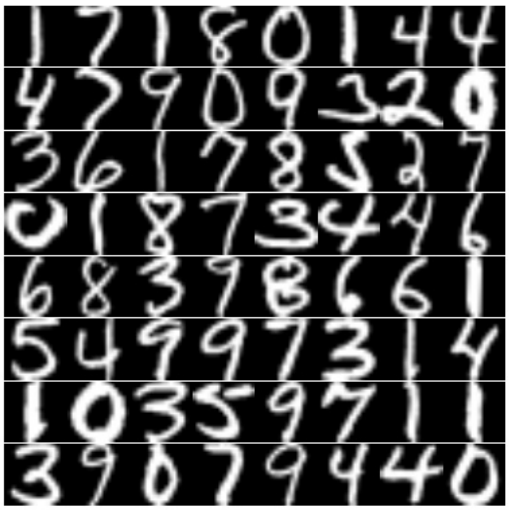

Fig. 1. Sample digits used for training the classifier

III. PCA based on Distances from Subspaces

Methodology

Notice that the PCA algorithm is to look for a k-dimensional affine space in $\mathbb{R}^p$ ($p = 256$ in this case) to best approximate n samples. Mathematically, the best approximation in terms of Euclidean distance is given by the following optimization problem.

$$\min_{\beta,\mu,U} I := \sum_{i=1}^{n} \|X_i - (\mu + U\beta_i)\|^2$$

The solution of this problem is (μ, U), determining a affine space, where μ is intercept, and U is an basis of the affine space. Suppose each digit was located in a affine space plus some perturbation as

$$X_i^{(k)} = \mu^{(k)} + U^{(k)}\beta_i^{(k)} + \epsilon,$$

where k is from 0 to 9, and $X_i^{(k)}$ means the image data of digit 'k'. Therefore $\mu^{(k)} + U^{(k)}\beta_i^{(k)}$ is actually the projection of $X_i^{(k)}$ onto the affine space of digit 'k'.

Based on above assumption, for a unknown label data X_i, the most natural method is to compare the distances from X_i to affine spaces of each digit 'k'. The shorter distance to affine space of 'k_i', the more likely X_i belongs to digits 'k_i'. accordingly we can easily classify X_i to the digits whose affine space give the shortest distance.

Algorithm

Training Part:

Step 1 Split Dataset to ten subsets. Samples represent same digits clustered into same subset.

Step 2 Implement PCA to each subset. Obtain $(\mu^{(k)}, U^{(k)})$ as features of digit 'k'.

Prediction Part:

Step 3 For given X_i, compute $U^{(k)}(X_i - \mu^{(k)})$ for each k.

Step 4 Find k_i such that $\|X_i - U^{(k_i)}(X_i - \mu^{(k_i)}))\|$ is the smallest one. The k_i would be our prediction.

Experiment Result

As we mentioned before, we randomly take 70% samples of each digit image data subset, and implement PCA for each subset. Set 30 components in PCA algorithm for each subsets. One of experiment gives the confusion matrix below:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	356	1	0	1	0	1	0	0	0	0
1	0	301	0	0	0	0	0	0	1	0
2	1	1	213	1	1	0	0	0	3	0
3	0	0	0	192	0	3	0	0	2	1
4	2	0	2	0	189	0	0	0	0	3
5	2	0	0	2	0	158	1	0	2	2
6	4	1	0	0	0	2	193	0	0	0
7	0	3	2	0	3	0	0	177	1	8
8	1	5	2	0	1	3	0	0	149	2
9	0	0	0	1	0	0	0	2	1	190

The confusion matrix element A_{ij} indicate the number of images that true digit is 'i' and the prediction is 'j'. Thus the diagonal element gives the correct classification. Therefore the accuracy would be

$$accuracy = \frac{\sum_{i=0}^{9} A_{ii}}{\sum_{i=0}^{9}\sum_{j=0}^{9} A_{ij}}.$$

In this case finally gives 96.58%.

IV. PCA with Logistic Regression and SVM

Methodology

Different from the usage of PVC within the dataset of each digit, in this section, we implement PCA among the whole dataset, to catch features of all digits instead of single digit.

Once we have features of whole dataset, for any given digits image, we can reduce the dimensions of input image to be classified. Therefore, some classic classifier algorithms could be used to predict which digit for this image. Here we use logistic regression and SVM.

Algorithm

Training Part:

Step 1 Implement PCA to the dataset. Obtain (U, μ)

Step 2 For X_i in train dataset, use UX_i and its corresponding digit as input to train logistic regression or SVM.

Prediction Part:

Step 2 For given X_i, compute UX_i as its low-dimension input.

Step 3 Use logistic regression or SVM trained before to obtain its prediction.

Experiment Result

We choose a set of number $[30, 35, 40, 45, 50, 65, 70, 80]$ as number of components in PCA algorithm, to evaluate the PCA with logistic regression performance. Picture 2 shows the accuracy depend on number of components in this algorithm. We note that 50 components gives the best performance in this situation. Less component may lead to not enough features to classify. However, too many components may include features of noise, which give rise to low accuracy.

The highest accuracy occurred when we choose 50 components, archiving 94.25%.

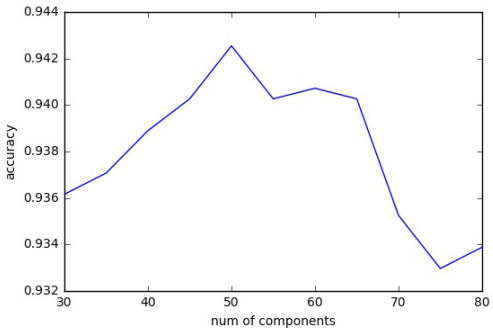

Fig. 2. PCA with logistic regression

For PCA with SVM algorithm, we set numbers of components were $[30, 35, 40, 45, 50, 65, 70, 80, 85, 90, 95]$. 'One verse one' strategy was used to train the model. We set different kernels of SVM in our experiment, including ref kernel, polynomial kernel with degree 1, 2, 3 and 4. Picture 3 shows the different results of different settings.

The result shows that the SVM method is very robust, with high performance. Almost in all situations, the accuracy could achieve above 97%. The highest accuracy occurred when we chose polynomial kernel with degree 3, and number of components was around 80, archiving 98.17%.

V. Error Analysis

From the experiments we find PCA technique can catch the feature of digit image effectively. Combined with some classifier algorithms, it is highly accurate to classify the image of digit.

However still some digits can not be classified correctly. From analysis of misclassified data, most digits was very easy to be recognized by human.

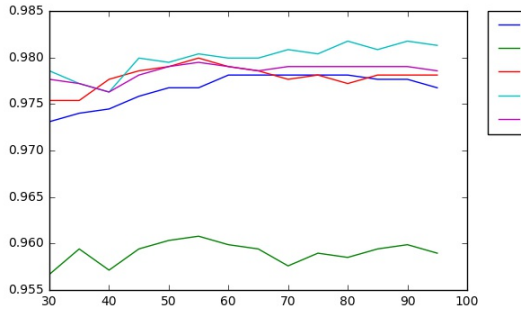

Fig. 3. PCA with SVM

Fig. 4. An example of misclassification: truth: 9; prediction 8

Picture 4 show a example of misclassification of our most powerful algorithm, chain PCA with SVM with polynomial kernel and degree 3. Similar situation happened in experiments of other methods. This might be caused by lack of data, or nonlinearity of digit structure which can not be catch by PCA, which need to be analyzed furthermore.

VI. Conclusions

In this project, we experiment three methods to recognize handwritten. One of them based on distance, with PCA within each digit sample. And the other two implement PCA to reduce dimensions of input then use logistic regression and SVM to classify respectively. Even though it seems that the PCA algorithm can not tell topological structure of image, all methods preform well, and the chain PCA and SVM obtain best score (accuracy $> 98\%$), which is a highly competitive result.

Appendix A
Code by Python

All code were updated to Github (https://github.com/yzhaobk/hand_written.git).

Programming language was Python, you can easily use jupyter notebook (http://jupyter.org) to check the code.

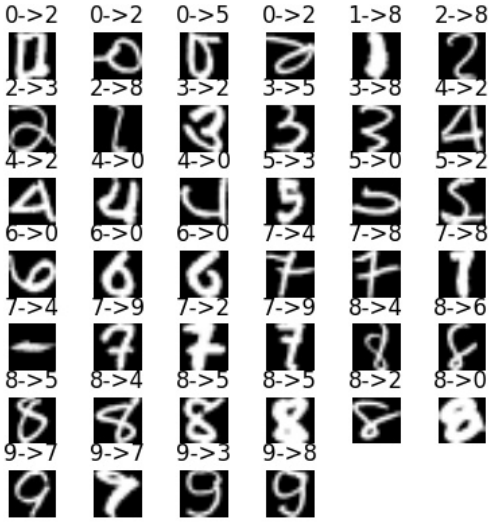

Fig. 5. All the 40 misclassification of test data in one experiment using chain PCA and SVM with poly kernel degree 3. $i \to j$ on the top of each image denotes that 'i' is misclassified as 'j'.

Appendix B
Some Misclassification Image

References

[1] Yuan Yao *A Mathematical Introduction to Data Science.* 2017 Spring, HKUST http://math.stanford.edu/~yuany/course/2016.spring/.

[2] Raschka, Sebastian. *Python Machine Learning* Birmingham, UK: Packt Publishing, 2015 published. https://github.com/rasbt/python-machine-learning-book

[3] Gaurav Jain, Jason Ko, *Handwritten Digits Recognition*, 2008.