Prototype selection for nearest neighbor

Qimin Chen
A53284263
qic003@ucsd.edu

1 Prototype selection

I firstly applied PCA (Principal Components Analysis) on original MNIST dataset to reduce the dimensionality from 28×28 to 25, then each class of MNIST (roughly 5000 images per class) was clustered by K-means into $M/10$ clusters and the center of clusters are used to represent the corresponding classes.

2 Pseudocode

As described above, PCA and K-means are used to implement prototype selection. The original MNIST training set is read as 60000×784 matrix so each image is converted to one-hot.

Algorithm 1 Prototype selection

```
Input: original MNIST training set $T$ with $60000 \times 784$ dimension and integer $M$.
Output: reduced training set $R$ with $M \times 25$ dimension.
apply PCA on training set and testing set to reduce dimensionality;
reduced_training = [ ]
for $label = 0 \cdots 9$ do
    model = KMeans(T, M/10)
    reduced_training += model.cluster_centers
end for
```

3 Experimental results

The confidence intervals is computed by equation (1):

$$(\bar{x} - z^* \frac{\sigma}{\sqrt{n}}, \bar{x} + z^* \frac{\sigma}{\sqrt{n}}) \tag{1}$$

where $\bar{x}$ is accuracy mean, z^* is critical value which is 1.96 in 95% confidence intervals, σ is the standard deviation of accuracy and n is the number of accuracy.

Table 1 shows the comparison of average accuracy and confidence intervals along with accuracy standard deviations in parentheses over 10 times for each value of $M = 100, 500, 1000, 5000, 10000$ between Prototype selection and Random selection. It is apparent that the performance of Prototype selection is better than Random selection. Also, PCA helps speeding up nearest neighbor classification because it reduces the dimensionality of dataset.

4 Critical evaluation

My method is a clear improvement over random selection as seen from Table 1. Since I have already applied PCA on dataset so that it helps improving the accuracy and saving training time as well, I would like to try on other methods like k-NN where k is greater than 1 or neural networks.

Experiment	Prototype selection	Random selection
M = 100	92.19 ± 0.0011 % (0.0012)	73.83 ± 0.0162 % (0.0184)
M = 500	94.97 ± 0.0010 % (0.0011)	87.35 ± 0.0036 % (0.0041)
M = 1000	95.69 ± 0.0007 % (0.0008)	90.40 ± 0.0022 % (0.0025)
M = 5000	96.59 ± 0.0012 % (0.0014)	94.37 ± 0.0012 % (0.0013)
M = 10000	96.81 ± 0.0009 % (0.0010)	95.58 ± 0.0013 % (0.0015)

Table 1: Averaged accuracy and confidence intervals along with accuracy standard deviations in parentheses over 10 times experiments per M.