A Short Exploration of Dimension Reduction: Handwritten Digit Recognition and PCA Family

Yi-Su Lo and Guiyu Cao

Department of Mathematics
The Hong Kong University of Science and Technology

Problem and data

- The data is a set of images for handwritten ZIP Code digits $0, 1, \dots, 9$, scanned from envelopes. Every sample could be regarded as a data point in $[0,1]^{256}$.

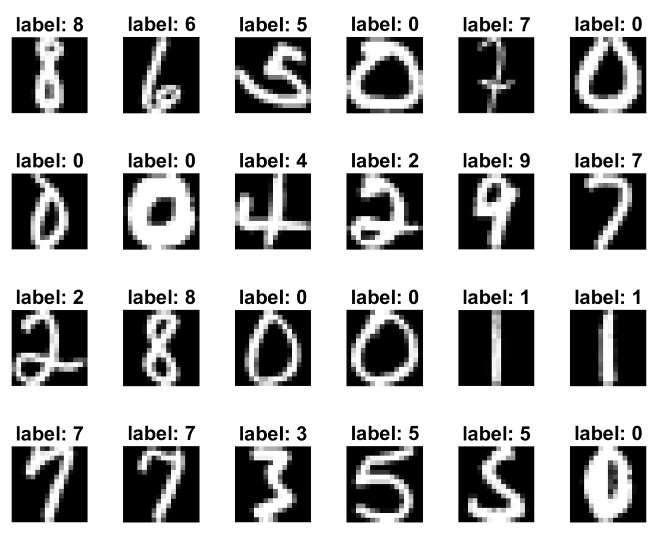


Figure 1: Some example samples in the training set.

Digit	0	1	2	3	4	5	6	7	8	9	Total
Training set	1194	1005	731	658	652	556	664	645	542	644	7291
Test set	359	264	198	166	200	160	170	147	166	177	2007

Table 1: Number of samples for training and test sets.

- In this mini-project, we try to complete the handwritten digit recognition task by the following steps
  (a) carry out principal component analysis(PCA), sparse PCA, or kernel PCA for dimension reduction;
  (b) apply support vector machine (SVM) to train classifiers.

Methods and software

Principal component analysis (PCA)

- For a given data set $\{x_i : i = 1, 2, \dots, n\}$, it solves

$$\min_{\mu, \beta_i, U} \sum_i \| x_i - (\mu + U\beta_i) \|,$$

where μ is the sample mean i.e. $\mu = \frac{1}{n}\sum_i x_i$, U consists of the *principle components* and $\sum_i \beta_i = 0$. Which is equivalent to

$$\min_L \| X - L \| \qquad \text{s.t.} \qquad \text{rank}(L) \le k,$$

where X is the data matrix collecting all data points.

Sparse principal component analysis (SPCA)

- SPCA extends the classic PCA by adding sparsity constraint on the input variables.
- Looking for sparse principle components, i.e. $\#\{Y_{ij} \neq 0\}$ are small. Using 1-norm convexification, we have the following SDP form for SPCA

$$\begin{aligned} \max \ & \text{trace}(\Sigma Y) - \lambda \|Y\|_1 \\ \text{s.t.} \ & \text{trace}(Y) = 1 \\ & Y \succeq 0 \end{aligned}$$

- SPCA realized with Thomas Bühler and Matthias Hein's Matlab code available at https://github.com/tbuehler/sparsePCA

Kernel principal component analysis (KPCA)

- KPCA is an extension of PCA using techniques of kernel methods.
- Consider the polynomial kernel (or Gaussian kernel)

$$k(x, y) = (x^T y + 1)^2 \ (\text{or } \ e^{-\frac{\|x-y\|^2}{2\sigma^2}})$$

construct the normalized kernel matrix of the data

$$\tilde{K} = K - 2\,1_{1/n}K + 1_{1/n}K1_{1/n}$$

Then, solve an eigenvalue problem

$$\tilde{K}\alpha_i = \lambda_i \alpha_i$$

Finally, data can be represented as

$$y_j = \sum_{i=1}^{n} \alpha_{ji} K(x, x_i), \ j = 1, \cdots, d.$$

- Matlab code for KPCA by Quan Wang: https://www.mathworks.com/matlabcentral/fileexchange/39715-kernel-pca-and-pre-image-reconstruction

Support vector machine (SVM)

- In a data set $\{(x_i, y_i) : y_i = \pm 1, i = 1, 2, \dots, n\}$, adopt the *soft-margin* SVM which solves

$$\min_{\beta, b, \xi_i} \frac{1}{2}\beta^T\beta + C\sum_i \xi_i^2$$

$$\text{subject to } \begin{cases} y_i f(x_i) \ge 1 - \xi_i, \\ \xi_i \ge 0, \end{cases} \forall\, i,$$

where ξ_i's serve as *slack variables*.
- We solve it with built-in funcitn in Matlab and user-defined codes.

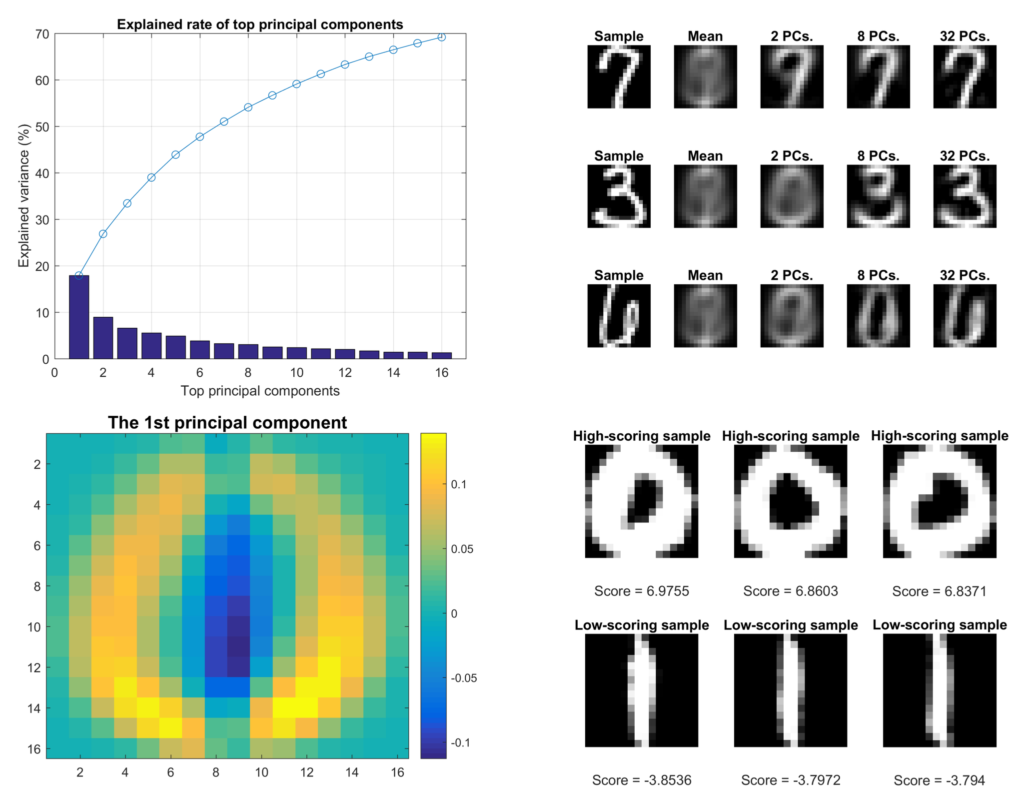
Experiments and Results

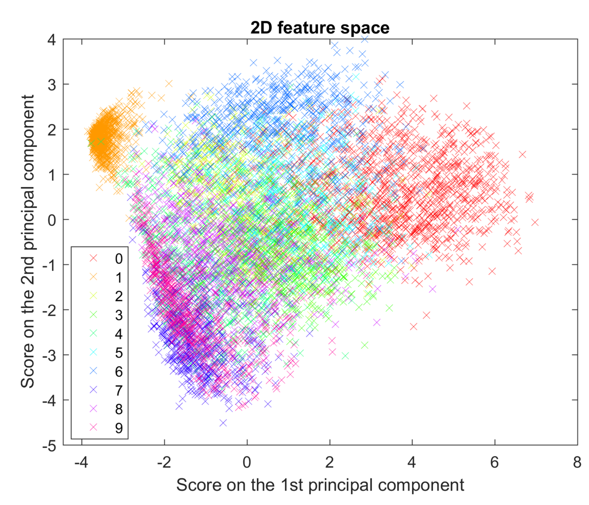
We examine three dimension reduction techniques and apply SVM for classifier training:

(a) PCA + SVM,
(b) SPCA with sparsity requirement 64 + SVM,
(c) SPCA with sparsity requirement 8 + SVM,
(d) KPCA with polynomial kernel + SVM,
(e) KPCA with Gaussian kernel + SVM.

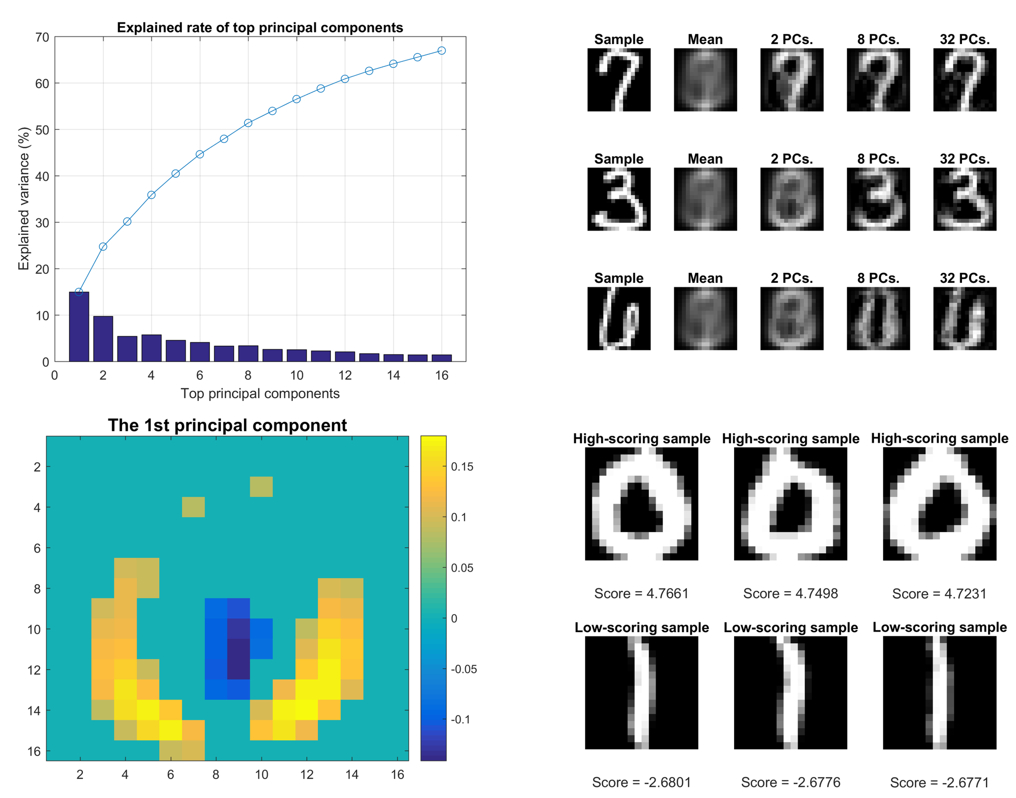
Glimpse into dimension reduction results

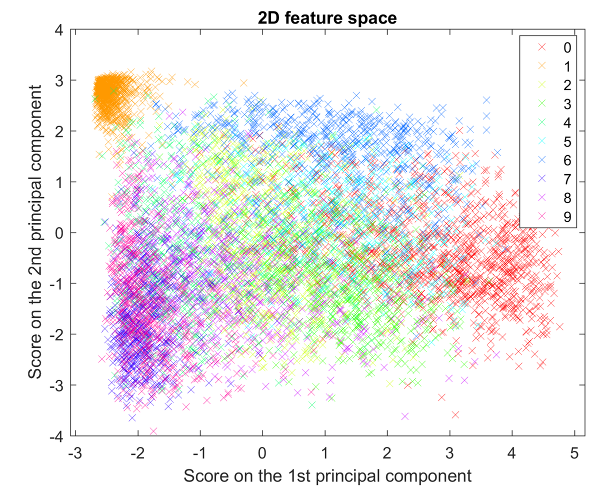
PCA



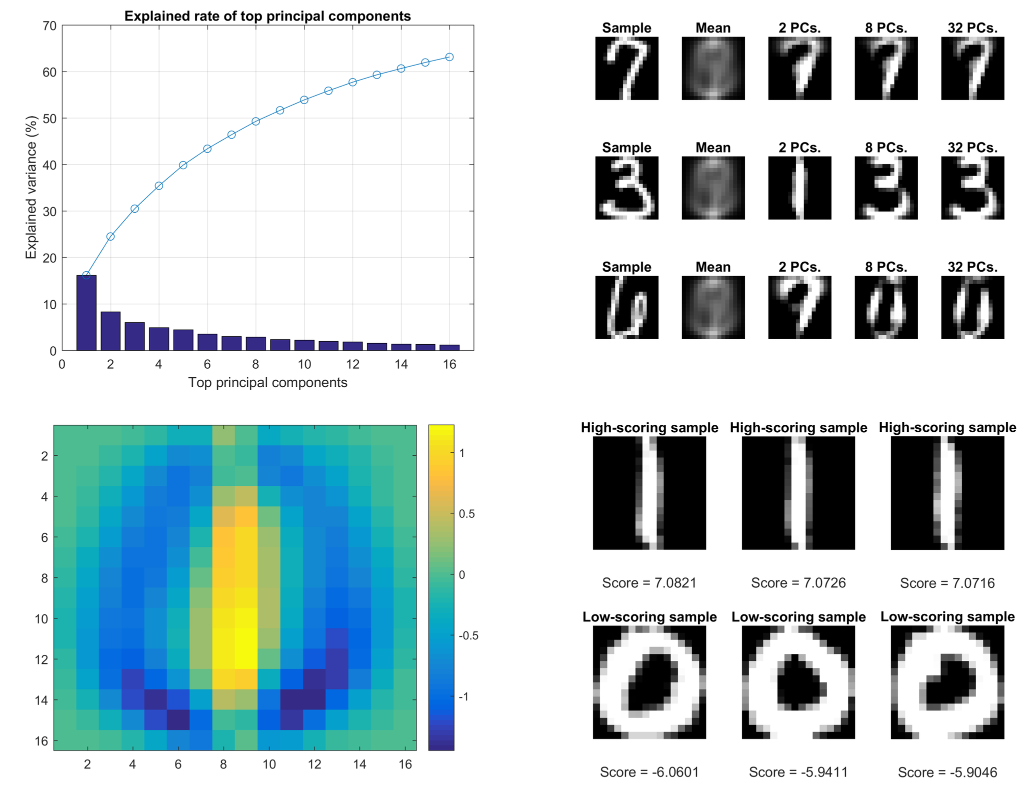


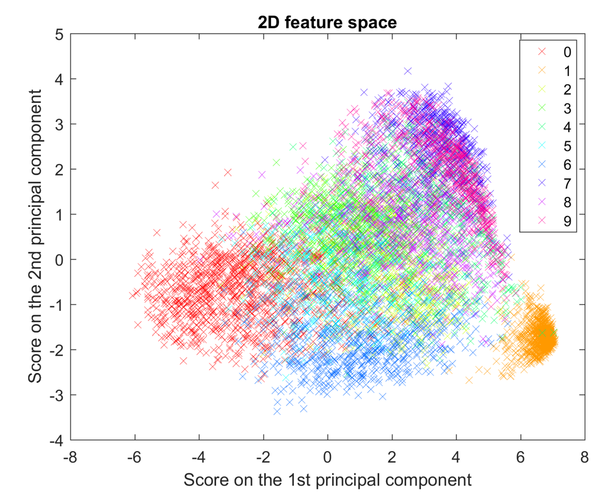
SPCA with sparsity requirement 64





KPCA with gaussian kernel





Performance

# Prin. comp. / Scheme	8	16	32
(a) **PCA**	0.55 / 11.31%	0.37 / 5.63%	0.50 / 4.88%
(b) **SPCA (64)**	3.08 / 11.96%	7.87 / 6.38%	23.56/ 4.58%
(c) **SPCA (8)**	2.19 / 14.95%	6.38 / 6.43%	14.67 / 4.98%
(d) **KPCA (p)**	399.13 / 17.09%	386.64 / 11.61%	381.45 / 6.83%
(e) **KPCA (g)**	447.76 / 11.61%	446.80 / 5.93%	447.40 / 4.93%

Table 2: Comparison on elapsed time (the first number which measured in sec) and test error rate (the percentage in red color).

- The elapsed time of SPCA grows as the number of principal components increases since SPCA is a recursive process.
- KPCA spends much more time, it is probably because it performs PCA in a higher-dimensional space.
- The classification is more accurate if we use more principal components. However, the effect is decreasing.
- The highest accuracy is achieved with SPCA. But overall, all three methods are effective dimension reduction approaches.

Error samples

Here are some test samples which were wrongly classified under the use of 32 principal components.