A Short Introduction to Recent Works on Sparse PCA with Applications

Yang Xu *

March 18, 2019

Abstract

This article mainly discussed the recent advances in sparse principle component analysis, including drawbacks of standard PCA, sparse PCA, algorithm and consistency. Then I conducted some experiments using SPCA to handwritten digit dataset, decreased the loss over 20% compared to the setting in standard PCA.

1 Introduction

Principal components analysis (PCA) (Jolliffe 1986) is a classical method for the dimension reduction of data in the form of n observations of a vector with p variables [2]. In practical, PCA is a popular data-processing and dimension-reduction technique, with numerous applications in engineering, biology, and social science. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

2 The Standard PCA

2.1 Methods

Before we move onto the variances of PCA, let's firstly review the standard PCA method. The main idea of PCA is that it seeks the linear combinations of the original variables such that the derived variables capture maximal variance. Suppose we have a design matrix $X \in \mathbb{R}^{n\times p}$, where $X = (x_1, \ldots, x_p)$ with covariance matrix $C \in \mathbb{R}^{p\times p}$, p denotes to the dimension of $x_i., i = 1, \ldots, n$ and n is the number of observations. For each x_i, it has mean μ_i. Then by the spectral decomposition, we can diagonalize C that

$$C = Q\Lambda Q^{-1} \tag{1}$$

*University of California, San Diego

where Λ is diagonal matrix that $\Lambda = diag(\lambda_1, \dots \lambda_p)$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$. $Q \in \mathbb{R}^{p \times p}$ is orthogonal, with Q_i eigenvector of C for eigenvalue λ_i.

The principle component for X is

$$Q_j^T(x - \mu) \tag{2}$$

and it have zero mean and are uncorrelated, since

$$cov(Q_j^T(x-\mu), Q_k^T(x-\mu)) = Q_j^T cov(x-\mu)Q_k = \lambda_k Q_j^T Q_k \tag{3}$$

Another approach to implement PCA is SVD. Given $X = (x_1, \dots, x_p)$, then

$$X = UDV^T \tag{4}$$

where U, V are eigenvector matrix for XX^T, X^TX respectively, D is the singular matrix. $Z = UD$ is the principal components and columns of V are the corresponding loadings of the principal components.

The principle components can be considered as the maximization along a one dimensional subspace of from X. As is shown in plot, Sum of squared perpendicular distances of data

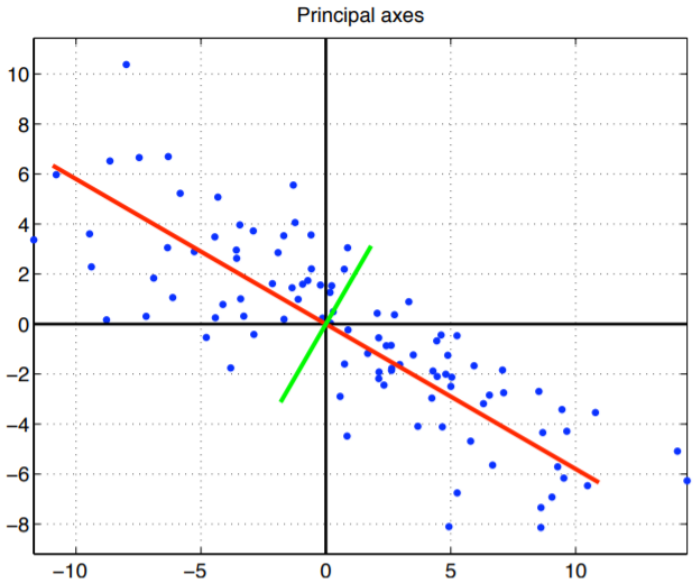


Figure 1 Variance explanation

points to first principle component (PC) line (red) is minimum among all lines through origin and i^{th} principle component is orthogonal to the previous one. Note that $\Lambda = Q^TCQ$, so that the total variance can be decomposed as follows:

$$Var = tr(C) = tr(\Lambda) = \sum_{i=1}^{p} \lambda_i \tag{5}$$

Therefore,

$$\frac{\lambda_1 + \cdots + \lambda_q}{\lambda_1 + \cdots + \lambda_p} \tag{6}$$

is the proportion of total variance explained by the first q largest principal components, where $1 \leq q \leq p$. Now we can choose the first q principle component to realize the dimension reduction that

$$PC_q = Q_{1:q}^T(x - \mu) \tag{7}$$

is the vector of the first q principal components. These principal directions explain most of the variance in the data.

There is another terminology to describe variance called loadings. Loadings also called component loadings in PCA, are the correlation coefficients between the variables and factors. The squared factor loading is the percent of variance in that variable explained by the factor.

2.2 Drawbacks

Although the standard PCA is very popular among data science and machine learning, it may face the bottleneck when encountering more generalized datasets. In many datasets, the number of dimensions p is very large, sometimes exceeding the number of observations n, so that the sample covariance matrix is not even consistent, and the representation of original variable can also results in interpretation difficulty [4]. We summarize them as the following:

- When p is comparable or dominates n, standard PCA is not consistent
- Each principal component is a linear combination of all the original variables and the loadings are typically nonzero, thus it is often difficult to interpret the results (derived PCs)

Hence, a more generalized version of PCA needs to be implemented, which will be discussed in the following sections.

3 Sparse Principle Component Analysis (SPCA)

Sparse Principle Component Analysis was first proposed by Zou, Hastie and Tibshirani (2006). They first converted the PCA to a regression-type problem then add the elastic-net penalty to the regression that PCA can be formulated as a optimization problem.

3.1 Sparse Approximations

Recall the lasso (l_1 penalty)

$$\hat{\beta}_{\text{lasso}} = \arg\min \|Y - \hat{Y}\|^2 + \lambda \sum_{i=1}^{p} |\beta_i| \tag{8}$$

where $\lambda \geq 0$. It is a very good, since lasso is strict that it forces some coefficients to 0 (produces sparsity in coefficients).

However, one major limitation of lasso is that the number of variables selected by lasso is limited by the number of observations. This implies $n \geq k$ where k is desired number of

PCs [3]. One way to dissolve this is combining both lasso and ridge penalty which is elastic net. Elastic net generalizes lasso to overcome the limitation.

In order to convert PCA to a regression-type problem, we note that the most important product we want is principle components. Zou and Hastie made the following regression approach.

For each i, denote by $Z_i = U_i D_{ii}$ the ith principal component. Consider a positive λ and the elastic net estimates $\hat{\beta}_{\text{en}}$ given by

$$\hat{\beta}_{\text{en}} = \arg\min \|Z_i - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1 \tag{9}$$

where $\hat{V}_i = \frac{\hat{\beta}_{\text{en}}}{\|\hat{\beta}_{\text{en}}\|}$ is an approximation to the loading for ith PC. $X\hat{V}_i$ is the ith approximated PC.

The reason to use Elastic-net here is that, we need to introduce lasso to produce the sparsity. However, in Lasso penalty we must set $n \geq k$ where k is a given amount of PCs. Therefore, we can add ridge penalty to relax this constrain, when $p > n$, we can choose some positive λ_2 that includes all variables in the fitted model.

There is one problem need to handle that we still need the result of PCA, since we need the value of principal component $Z = UD$ in this equation. Therefore it is not a genuine alternative. However, we can generalize this model to make it a two-stage exploratory analysis: perform PCA, then find the sparse approximation using Equation 9.

Theorem 1 *Let x_i denote the ith row vector of matrix X, consider the leading principal component. For ant $\lambda > 0$, let*

$$\begin{aligned}(\hat{\alpha}, \hat{\beta}) = &\arg\min_{\alpha,\beta} \sum_{i=1}^{n} \|x_i - \alpha\beta^T x_i\|^2 + \lambda\|\beta\|^2 \\ &s.t. \|\alpha\|^2 = 1\end{aligned} \tag{10}$$

then $\hat{\beta} \propto V_1$.

We can extend this theorem to consider all principal components.

Theorem 2 *For the first k principal components, let $A_{p\times k} = [\alpha_1, \ldots, \alpha_k]$ orthogonal, and $B_{p\times k} = [\beta_1, \ldots, \beta_k]$. For any $\lambda > 0$,let*

$$\begin{aligned}(\hat{A}, \hat{B}) = &\arg\min_{A,B} \sum_{i=1}^{n} \|x_i - AB^T x_i\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1, \\ &s.t. A^T A = I_{k\times k}.\end{aligned} \tag{11}$$

Then $\hat{\beta}_j = cV_j, j = 1, 2, \ldots, k$ where c is a constant.

It effectively transform the PCA into a regression-type problem. The equation in Theorem 2 is called SPCA criterion, so that it is feasible to use some algorithm to do this convex optimization.

3.2 An Algorithm to Minimize SPCA Criterion

We can see that there are to outputs need to be estimated, by setting the constrain $A^TA = I_{k\times k}$, it turns out to be a convex optimization problem.

Assume A is fixed, then for each $j = 1, \ldots, k$, let $Y_j^* = X\alpha_j$ which is the generated principle component. By Equation 9, we know that

$$\begin{aligned}\hat{\beta}_j &= \arg\min_{\beta} \sum_{i=1}^{n} \|Y_j^* - X\beta_j\|^2 + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1 \\ &= \arg\min_{\beta} (\alpha_j - \beta)^T X^T X(\alpha_j - \beta) + \lambda\|\beta\|^2 + \lambda_{1,j}\|\beta\|_1\end{aligned} \tag{12}$$

Now we have $B = [\beta_1, \ldots, \beta_k]$, we can calculate A based on the fixed B. According to reduced rank Procrustes rotation, we can get

$$(X^TX)B = VDV^T \tag{13}$$

then $\hat{A} = UV^T$.

Now we can conclude a general SPCA algorithm described below.

Algorithm 1 General SPCA Algorithm

1: Let A start at V[,1:k] which are loadings of the first k ordinary PCs.
2: Given a fixed $A = [\alpha_1, \ldots, \alpha_k]$, solve the following elastic-net for $j = 1, 2, \ldots, k$

$$\beta_j = \arg\min_{\beta} (\alpha_j - \beta)^T X^T X(\alpha_j - \beta) + \lambda\|\beta\|^2 + \lambda_{1,j}\|\beta\|_1$$

3: For a fixed $B = [\beta_1, \ldots, \beta_k]$, compute the SVD of X^TXB, then update $A = UV^T$.
4: Repeat Steps 2-3, until convergence or the maximum number of iterations.
5: Normalization: $\hat{V}_j = \beta_j/\|\beta_j\|, j = 1, \ldots, k$.

Figure 2 SPCA algorithm

In practical, the output does not change much as λ is varied. For $n > p$, default choice of λ can be 0. Meanwhile, we can use a range of values to tune parameters $\lambda_{1,j}$.

3.3 Semidefinite Programming Approximation

Since SPCA is an optimization problem, we can also define the semidefinete programming of SPCA. d'Aspremont et al.(2009) [1] gave the formula for SPCA using semiedfinte programming. $A \in S^n$ be a covariance matrix, $x \in R^n, X = xx^T$, $k \in 1, 2, \ldots, n$, then we have the semidefinite relaxation:

$$\begin{aligned}&\max tr(AX) \\ &s.t. tr(X) = 1, card(X) \leq k^2, X \geq 0, rank(X) = 1\end{aligned} \tag{14}$$

They also discussed that the method has complexity $O(n^4\sqrt{log(n)}/\epsilon)$, where n is the size of the underlying covariance matrix and ϵ is the desired absolute accuracy on the optimal value of the problem.

4 Numerical Experiment

Sparse PCA can be applied to many industries. We begin with a illustration on a simple constructed example. We play with some experiment on a single component model firstly, then extend this result to applications.

Begin with column vectors in p dimensions, for $i = 1, 2, \ldots, n$, define

$$x_i = v_i \rho + \sigma z_i \tag{15}$$

where $x_i \in \mathbb{R}^p$ a data vector, $v_i \sim N(0, 1)$, $\rho \in \mathbb{R}^p$ is a single component to be calculated, $z_i \sim N_p(0, I)$ are independent p-dimensional random variables.

Define $\rho_l = f(l/n)$, where

$$f(t) = C(0.7B(1500, 3000)(t) + 0.5B(1200, 900)(t) + 0.5B(600, 160)(t)) \tag{16}$$

denotes the distribution of ρ and

$$B(a, b)(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \tag{17}$$

denotes the Beta probability density function defined on $[0, 1]$.

Set $p = 2048, n = 1024, \sigma = 1, k = 372, C = 0.1$, we first draw the plot of ρ, as is shown in Figure 3(i), it has three peaks spectrum. This is because ρ is determined by a mixture of Beta functions which gives this property. C is a scale parameter which controls $\|\rho\|_2$.

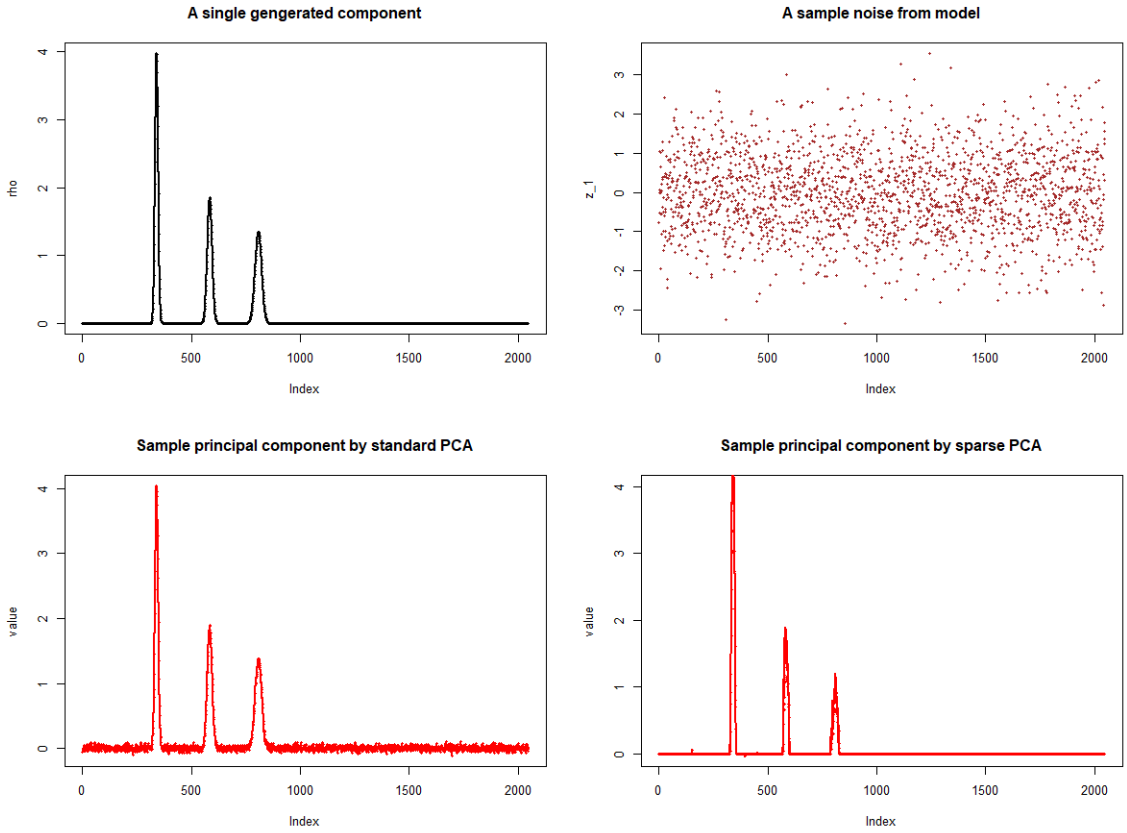


Figure 3 Principle components

Figure 3(ii) demonstrates sample paths drawn from the model above with $n = 1024$ samples in total. Figure 3(iii) and (iv) show the two approaches to implement PCA, standard and sparse respectively.

We can find some observation that in the left bottom plot, which is a principle component using standard PCA. It captures the spectrum correctly, however, in the rest intervals, its value oscillates around 0, which results from noise in the model. Therefore, the effect of the noise remains clearly visible in the estimated principal eigenvector.

On the contrary, Figure 3(iv) is the principle component using SPCA. We can see there is a significant difference between them. Intervals between spectrum peaks are flat, almost the same as the true principle component ρ. Therefore, SPCA has a better performance for estimating principle components with noise.

5 Applications Using SPCA

Sparse principle components analysis has been applied to various fields. It is a popular method for data preprocessing if the desired p is grater than n. Signals and images, in which the number of sampling points, or pixels, is often comparable with or larger than n. For example, electrocardiogram (ECG) signal analysis. People apply SPCA to ECG signal to study the character of beat variation better, since local features to play a significant role in the principal component eigenvectors.

Another example is image classification, such as facial expressions and handwritten digits. These kinds of images involve a considerable amount of features, usually in thousands. We can do SPCA process then improve the classification result, given limit computation complexity.

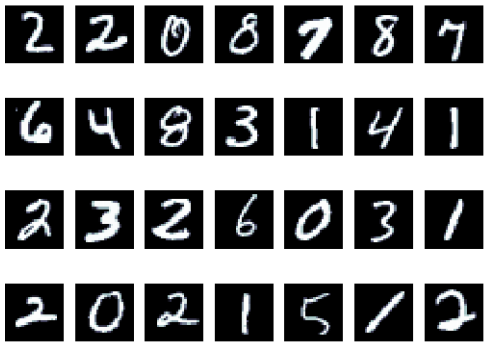
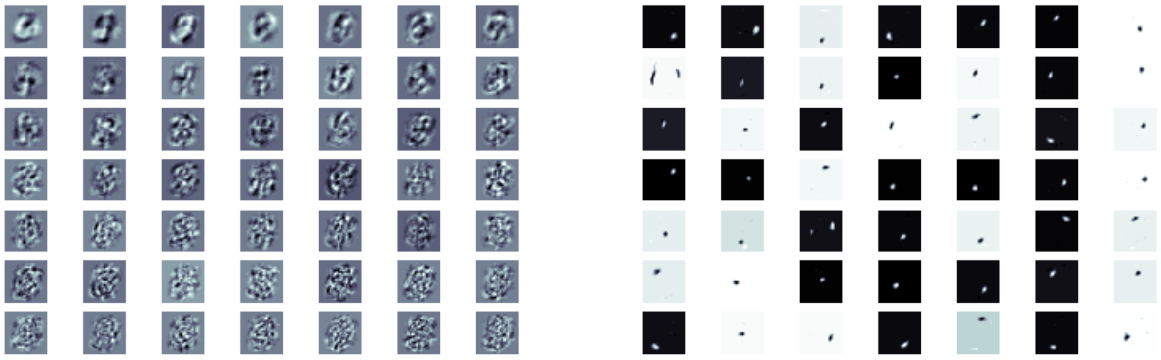


Figure 4 Examples of handwritten digits

As is shown in Figure 4, for each digit, it is a 28×28 pixels image. we first convert images to one dimensional vector and minus its mean, so that for a single vector, it has a length of 784 and mean 0. Let $n = 100, p = 50$, we can get loadings plots form both methods, shown in Figure 5. We can see that loadings have very difference demonstrations in two plots, loadings of SCPA are more sparse in the image. It is also worth highlighting that while PCA loadings that correspond to smaller singular values usually capture more high-frequency

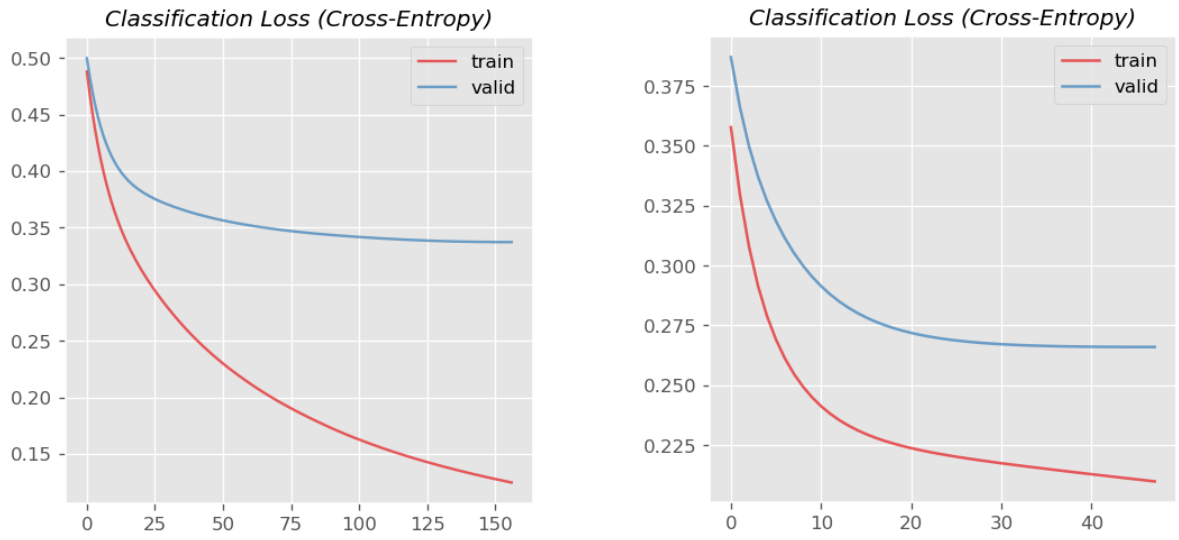
features, this phenomenon does not seem to appear in the SPCA loading due to their sparse nature. Meanwhile, due to the sparse nature, loadings in SPCA can identify features more locally. This means they can characterize different digits more clearly. Therefore, loadings can identify important features that distinguish handwritten digits.



(a) PCA loadings (b) SPCA loadings

Figure 5 Comparison between loadings in PCA and SPCA

After applying dimension reduction to the features, we trained the model through a simple neural network. After training, we can make a comparison between two classification errors.



(a) Classification loss using PCA (b) Classification loss using SPCA

Figure 6 Comparison between loss in PCA and SPCA

Table 1 Prediction performance

Preprocessing method	Classification error
PCA	0.337
SPCA	0.266

As we can see, using SPCA to preprocess data can get a smaller classification error compared to that in PCA. Performace improved 21% on validation, SPCA has the less loss. Hence, we can draw the conclusion that for $n < p$, SPCA can produce much more sparse loadings and it performs better than PCA in this classification task. Sparse features offer a more compact and interpretable representation of the data and thus make them much more distinguishable in classification.

References

[1] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[2] Iain M Johnstone and Arthur Yu Lu. Sparse principal components analysis. *Unpublished manuscript*, 7, 2004.

[3] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[4] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.