# Principal Component Analysis variants for Parkinson datasets

**Chen Cheng**

College of Science, University of Shanghai for Science and Technology, Shanghai, 200093, China

1114010209@qq.com

**Abstract.** Principal Component Analysis (PCA) is one of the most fundamental dimension reduction methods that need further research. With the widespread popularity of machine learning and the arrival of the era of big data, dimension reduction has become a hot topic and principal component analysis is a hot topic. However, although there are a lot of researchers who focus on the methods of the PCA, few researches on Parkinson Datasets have been made. As a result, the aim of our work is to discuss the PCA variants for Parkinson Datasets. This paper first introduces the three most commonly used PCA methods: PCA, Sparse PCA and Kernel PCA, and then introduces the Support Vector Machine (SVM) used to measure the dimension reduction effect. After that, we introduced the Parkinson's dataset and the meanings of root mean square error (RMSE), overall accuracy, Cohen's kappa (Kappa) and computational time, the indicators that are used to measure the dimensionality reduction effect. Finally, we identified the variants among different PCA methods on the Parkinson dataset by comparing the indicators of the data obtained after dimensionality reduction using different methods.

**Keywords:** PCA, Sparse PCA, Kernel PCA, SVM, Parkinson Datasets, RMSE, Kappa

## 1. Introduction

With the rapid development of science and technology, the volume of data in production and life has become increasingly complex. These complex data typically have high dimensions, and obtaining labels for the data can also be very expensive. In multivariable problems, information overlap often exists, which requires dimensionality reduction methods to simplify the problem and extract effective information. Therefore, studying the differences in various dimensionality reduction methods is very crucial because proper use of these methods can reduce the complexity of data, improve the efficiency of processing data, and reduce the cost of labeling data. This is an article studying the effectiveness of different dimensionality reduction methods in principal component analysis on a specific dataset. We will use principal component analysis, sparse principal component analysis, and kernel principal component analysis on the Parkinson's dataset, and compare the effectiveness of the three methods after dimensionality reduction. The Parkinson's dataset is a typical problem with multiple variables. To diagnose Parkinson's patients, we need to analyze 22 features for diagnosis. The dimensions of 22 features are too large, resulting in a significant workload when processing these data. Therefore, it is necessary to consider using dimensionality reduction to reduce the number of features. Two methods will be used to measure the effectiveness of dimensionality reduction. The first method is RMSE, which maps the dataset from the dimensionality-reduced space back to the original space and compares the errors between the datasets. The second method is to use SVM to process the dimensionality-reduced data and calculate the parameters of overall accuracy and Cohen's kappa for comparison. Support vector machine is currently one of the most commonly used and effective classifiers. It can achieve better results than other algorithms on small sample training sets due to its excellent generalization ability. Since the Parkinson's dataset only has 195 instances, it will not cause the running time of the support vector machine to be too long. Therefore, applying the support vector machine is a wise choice. Cohen's kappa can be used to evaluate the difference in evaluation results between two evaluators. Because there are two evaluators' evaluation results in our data, one is the label that comes with the Parkinson's dataset, and the other is the label obtained after support vector machine classification, we can use this statistic to evaluate the dimensionality reduction effect. After that, a model for predicting Parkinson's disease will be provided.

Firstly, this article introduces three different dimensionality reduction methods and their principles. Secondly, the classification method of the support vector machine has been introduced. The SVM will be used to estimate the efficiency of the dimensionality reduction indirectly. The following paragraph introduces the Parkinson's dataset and four indicators for measuring dimensionality reduction results. Afterward, the dimensionality reduction data was visualized and the results of the dimensionality reduction were discussed.

## 2. Methods

### *2.1. Principal Component Analysis*

Principal component analysis (PCA) [1] is one kind of classic dimensionality reduction method. It converts data represented by multiple linearly related variables into data represented by several linearly independent variables through orthogonal transformation. These linearly independent variables are the principal components, and these principal components retain the most of information in the original dataset.

The basic idea of PCA for dimensionality reduction in high-dimensional data is as follows: First of all, standardize the variables of the original dataset to a dataset with a mean of 0 and a variance of 1. Then, perform the orthogonal transformation on the standardized data, transforming the original data into new data that are represented by several linearly independent vectors. The data represented by these new vectors not only needs to be linearly unrelated to each other, but also needs to retain the maximum amount of information. As long as these new data keep the maximum amount of information, can they be representative of replacing the original dataset?

To quantify the concept of how much information is retained, PCA uses variance to measure the size of the information in a new variable. The principal components are sorted according to the size of variance, with the first being called the first principal component, the second being called the second principal component, and so on.

Next, the simple idea of the PCA will be shown mathematically.

Assuming the raw data is an m-dimensional random variable:

$$\mathbf{x} = (x_1, x_2, \cdots, x_m)^T,$$

the mean vector is:

$$\boldsymbol{\mu} = E(\mathbf{x}) = (\mu_1, \mu_2, \cdots, \mu_m)^T$$

the covariance matrix is

$$cov(\mathbf{x}, \mathbf{x}) = E[(\mathbf{x} - \boldsymbol{\mu})(x - \boldsymbol{\mu})^T]$$

Linear transformation from m-dimensional random variable $\mathbf{x}$ to m-dimensional random variable $\mathbf{y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \cdots, \boldsymbol{y_m})$ is

$$y_i = \boldsymbol{\alpha}_i^T \mathbf{x} = \alpha_{1i}x_1 + \alpha_{2i}x_2 + \cdots + \alpha_{mi}x_m \ and\ \alpha_i^T = (\alpha_{1i}, \alpha_{2i}, \cdots \alpha_{mi}).$$

The mean, variance, and covariance statistics of random variables after linear transformation can be expressed as

$$E(y_i) = \boldsymbol{\alpha}_i^T \mu_i, i = 1,2,\cdots,m$$

$$var(y_i) = \boldsymbol{\alpha}_i^T cov(\boldsymbol{x},\boldsymbol{x})\boldsymbol{\alpha}_i, i = 1,2,\cdots,m$$

$$cov(\boldsymbol{y_i},\boldsymbol{y_j}) = \boldsymbol{\alpha}_i^T cov(\boldsymbol{x},\boldsymbol{x})\boldsymbol{\alpha}_j, i,j = 1,2,\cdots,m$$

The transformed random variables $\boldsymbol{y_1},\boldsymbol{y_2},\cdots,\boldsymbol{y_m}$ can be considered as the first principal component, second principal component, ……, and the m-th principal component when the linear transformation from random variable $\mathbf{x}$ to random variable $\mathbf{y}$ satisfies the following conditions:

(1) The coefficient vector after linear transformation is the unit vector:

$$\|\boldsymbol{\alpha}_i\| = 1$$

(2) The variables after linear transformation are linearly independent of each other:

$$cov(\boldsymbol{y_i},\boldsymbol{y_j}) = 0 \; when \; i \neq j$$

(3) The variable is the largest of all linear transformations of the original random variable

The above three conditions provide a general method for solving principal components. The method of conditional extreme value can be used to get the principal component, and the process of getting the first component will be shown as an example.

The mathematical expression for the optimization problem of the first principal component is

$$max \; \boldsymbol{\alpha}_1^T cov(\boldsymbol{x},\boldsymbol{x})\alpha_1$$

$$s.t. \alpha_1^T \alpha_1 = 1$$

Let the Lagrange function be

$$L = \boldsymbol{\alpha}_1^T cov(\boldsymbol{x},\boldsymbol{x})\alpha_1 - \boldsymbol{\lambda}(\alpha_1^T \alpha_1 - 1)$$

Calculating the derivative of Lagrange function and let it be zero

$$\frac{\partial L}{\partial \alpha_1} = cov(\boldsymbol{x},\boldsymbol{x})\alpha_1 - \boldsymbol{\lambda}\alpha_1 = 0$$

According to the relationship between eigenvalues and eigenvectors and the formula above, the $\boldsymbol{\lambda}$ is the maximum eigenvalue of the matrix $cov(\mathbf{x},\mathbf{x})$, $\alpha_1$ is the corresponding eigenvector. If $\alpha_1$ is the unit eigenvector corresponding to the maximum eigenvalue $\lambda_1$ of $cov(\mathbf{x},\mathbf{x})$, $\alpha_1$ and $\lambda_1$ are the solutions for the optimization problem.

As a result, $\alpha_1^T x$ is the first component and its variance is the maximum eigenvalue of the corresponding covariance matrix:

$$var(\boldsymbol{\alpha}_1^T x) = \boldsymbol{\alpha}_1^T cov(\boldsymbol{x},\boldsymbol{x})\alpha_1 = \lambda_1$$

From what is done above, the process of finding the first component is over, and the same method can be used to find the k-th component.

In summary, PCA uses dimension reduction technology to reduce a large number of variables to a small number of principle components. The majority of the data in the original dataset may be covered by these primary components.

## *2.2. Sparse PCA*

Principal Component Analysis (PCA) is widely recognized as an effective feature extraction algorithm that can perform feature extraction and data dimensionality reduction. However, due to the fact that the PCA ultimately produces a linear combination of the original data variables, sometimes we may find it hard to explain what the corresponding features of each principal component are when we need to analyze and explain the principal components. In order to overcome this difficulty, Sparse Principal Analysis (Sparse PCA) [2] is introduced to solve this problem. The so-called sparsity in the name refers to the sparsity of the principal component coefficients, which means that most of the coefficients in the principal component are 0. This method will make the principal component coefficients (coefficients in front of each variable when forming the principal component) sparse, which means that most of the coefficients will become zero. In this way, we can highlight the main parts of the principal component, making it easier to explain.

In practical applications, the elements in sparse vectors are usually directly ignored to obtain the so-called sparse solution based on principal component analysis. For example, by artificially setting some constraints, when the coefficient in the coefficient vector $\boldsymbol{\alpha}_{\mathrm{i}}$ is smaller than a given constant, the value of the coefficient will be directly reduced to 0. As a result, there will be many zero elements in the coefficient vector, and we can obtain a sparse vector, which is called sparse principal component analysis.

The method mentioned above is the simplest and most crude method for solving sparse principal component analysis. For sparse principal component analysis problems, there are other formulas and methods. Still, the core idea is to introduce an additional constraint condition in the original principal component analysis optimization problem:

$$max\ \boldsymbol{\alpha}_1^T cov(\boldsymbol{x},\boldsymbol{x})\alpha_1$$

$$s.t.\ \alpha_1^T\alpha_1 = 1$$

to obtain sparse solutions.

In summary, Sparse PCA adds some constraints to the original PCA. The addition of restrictive conditions can result in more zeros in the coefficients before the principal component, making the principal component simpler.

### *2.3. Kernel PCA*

Kernel PCA (KPCA) [3] is an extension of Principal Component Analysis, a widely-used dimensionality reduction technique. Unlike PCA, which assumes linear separability of the data, Kernel PCA is designed to handle nonlinear complicated datasets by mapping them to a higher-dimensional space using a kernel function.

Several kernel functions [4] can be used in KPCA, including:

Polynomial kernel function:$K(x,y) = (x \cdot z + 1)^p$

Gaussian kernel function:$e^{(-\frac{\|x-y\|^2}{2\sigma^2})}$

These kernel functions enable KPCA to map the data into a higher-dimensional feature space, where it becomes linearly separable and then projects it onto the principal components.

### *2.4. SVM*

Support vector machines [5] are algorithms that analyze data in classification and regression and the data using this method are labeled. It is a strong and complete model, which are able to carry out linear classification, regression, and identify outliers. In addition, the case of nonlinearity also applies.

The support vector machine learning method includes three models. One of these is the linearly separable support vector machine, which needs a linearly separable training set and can produce a hyperplane by maximizing the hard interval.

The hyperplane can be obtained by maximizing the soft interval in the second method, which uses a linear support vector machine and requires that the training set be roughly linearly separable.

Nonlinear Support Vector Machine is the final one [6]. The training set does not meet the requirements of the first two situations in this case. Using a kernel function, the poor training set can be transformed into a linearly separable data set, and the hyperplane may be found by maximizing the soft interval. Simple models are particular instances of complex models in the three models. Here is a brief introduction to the idea of linear support vector machines.

First of all, a clear definition of linear separable support vector machines should be given. When the data meets the first condition mentioned earlier, the optimal linearly separable hyperplane $w^*x + b^* = 0$ and the corresponding classification decision function $f(x) = sign(\, w^*x + b^*)$ can be obtained by maximizing the hard margin to solve the corresponding convex quadratic programming problem. This kind of circumstance is called linearly separable support vector machine.

In order to maximize the margin, it is necessary to represent the margin first. For support vector machine, the distance from a given point to the linearly separated hyperplane can be expressed as the reliability of classification prediction. When the linearly separated hyperplane of classification is determined, $|w \cdot x + b|$ can be expressed as the distance from the point $x$ to the hyperplane. At the same time, we can also use the consistency between the symbol of $w \cdot x + b$ and the symbol of classification marker $y$ to judge whether the classification is correct. Therefore, for a given training sample and a linearly separated hyperplane $w \cdot x + b = 0$, the functional margin of the linearly separated hyperplane with respect to any sample point $(x_i, y_i)$ can be expressed as $\widehat{d_\iota} = y_i(w \cdot x_i + b)$

Then the margin between the training set and the linearly separated hyperplane can be determined by the minimum functional margin between the hyperplane and all sample points, that is: $\hat{d} = min\widehat{d_\iota}(i = 1,2, \dots, N)$

In order to make the margin not affected by the changes of the linearly separated hyperplane parameters $w$ and $b$, we also need to add a normalization constraint $||w||$ to $w$ so as to fix the margin. In this way, the functional margin is converted into the geometric margin.

At this time, the geometric margin of the linearly separated hyperplane with respect to any sample point $(x_i, y_i)$ can be expressed as: $d_i = y_i(\frac{w}{||w||} \cdot x_i + \frac{b}{||w||})$

The interval between the training set and the linearly separated hyperplane can also be expressed by $d = mind_i(i = 1,2, \dots, N)$.

The maximum margin obtained based on linearly separable support vector machines is also known as hard margin maximization. Maximizing hard margin can be intuitively interpreted as classifying training data with sufficiently high reliability.

Next, we formalize the maximization of hard margins into a constrained optimization problem:

$$\max_{w,b} \; d$$

$$s.t. \;\; y_i\left(\frac{w}{||w||} \cdot x_i + \frac{b}{||w||}\right) \geq d, i = 1,2, \dots, N$$

According to the relationship between function margin and geometric margin, the above expression can be rewritten as:

$$\max_{w,b} \frac{\hat{d}}{||w||}$$

$$s.t. y_i(w \cdot x_i + b) \geq \hat{d}, i = 1,2, \dots, N$$

In the above mathematical expression, it can be seen that the value of $\hat{d}$ have no relation with the solution of the optimization problem. To make the problem simple, let it be 1, the above equation can be rewritten as an equivalent optimization problem:

$$\min_{w,b} \frac{1}{2} ||w||^2$$

$$s.t. y_i(w \cdot x_i + b) - 1 \geq 0, i = 1,2, \dots, N$$

So far, the hard margin maximization problem has been transformed into a convex quadratic programming problem, and its Lagrange function can be constructed:

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{N}\alpha_i$$

The equation above can be solved directly.

## 3. Data and Experiments

The Parkinson's Disease Dataset [7] is used to evaluate principal component analysis variants for dimensionality reduction. This dataset contains biomedical voice measurements from 31 people, including 23 with Parkinson's disease. The dataset has 195 instances with 22 features each, and the goal is to predict the presence of Parkinson's disease in the patient. Three dimensionality reduction methods will be used in turn, and the two-dimensional data after dimensionality reduction will be visualized. Afterwards, the dimensionality reduced data will be transmitted back to the original space to obtain a reconstructed dataset. The reconstructed dataset will be compared with the original dataset, and RMSE will be used as an indicator to measure the dimensionality reduction effect. In addition, the SVM will be used on the dimensionality reduced dataset. When using support vector machines, the dimensionality reduced data will be divided into two parts, while 80% of the data being used as the training set and the remaining data as the test set. After executing this SVM, the overall accuracy and Cohen's kappa will be calculated.

### *3.1. Root Mean Square Error*

The root means square error (RMSE) [8] measures the average difference between a statistical model's predicted values and the actual values. Mathematically, it is the standard deviation of the residuals.

### *3.2. Overall Accuracy*

In classification algorithms, this parameter is used to measure the accuracy of classification. Specifically, it examines the difference between the true and predicted values of labels.

### *3.3. Cohen's Kappa*

A statistic used to gauge inter-annotator agreement is Cohen's kappa. It is a number that indicates how well-agreed about a categorization problem two annotators are. It is defined [9] as

$$k=\frac{p_0-p_e}{1-p_e}$$

Where $p_0$ is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and $p_e$ is the expected agreement when two annotators assign labels randomly. $p_e$ is estimated using a per-annotator empirical prior [10] over the class labels.

### *3.4. Computational Time*

Computational time is an indicator that records the time spent by the algorithm from input data to output data.

### 3.5. *Visualization of dimensionality reduction data*

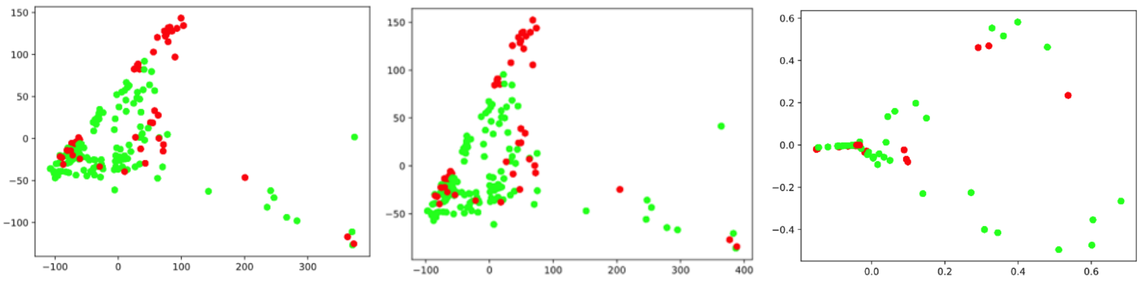
(a) (b) (c)

**Figure 1.** Visualizations of the dimensionality reduced data

The figure 1 above are visualizations of the dimensionality reduced data. Among them, (a) represents the data after PCA dimensionality reduction, (b) represents the data after Sparse PCA dimensionality reduction, and (c) represents the data after Kernel PCA dimensionality reduction. In each scatter plot, green dots represent objects without Parkinson's disease, while red dots represent objects with Parkinson's disease.

### 3.6. *Outcomes*

In the Table 1, the RMSE of the data after dimensionality reduction using different methods, as well as the Overall accuracy and Cohen's kappa obtained from the data after dimensionality reduction using different methods processed by SVM algorithm.

**Table 1.** Result of the data after dimensionality reduction using different methods

| Parameter / Method | RMSE | Overall accuracy | Cohen's kappa |
|---|---|---|---|
| PCA | 5.351 | 0.744 | 0.217 |
| Sparse PCA | 5.355 | 0.744 | 0.217 |
| Kernel PCA | 22.765 | 0.769 | 0.107 |

## 4. Conclusion

When using linear dimensionality reduction methods on the Parkinson's dataset, the visualization effect of dimensionality reduction into 2D data is good, and it can be clearly seen on the graph that the sample distribution holds two classes, and the RMSE values of these two linear dimensionality reduction methods are also very small. When using non-linear dimensionality reduction methods, the value of RMSE will become relatively large. The classification effect of 2D data visualization is not as good as that of the first two linear dimensionality reduction methods.

The overall accuracy of PCA and Sparse PCA is generally the same, both at 0.744. The same parameters obtained by these two methods may be due to their use of linear dimensionality reduction methods. However, the overall accuracy of KPCA is 0.769, which is 0.029 higher than the first two linear methods. Perhaps the use of kernel techniques has resulted in a very small improvement in accuracy. Therefore, when using overall accuracy as a measure of Parkinson's diagnostic accuracy, kernel principal component analysis can be considered.

When it comes to the Cohen's kappa, the result obtained by PCA and Sparse PCA is 0.217, while the result obtained by using Kernel PCA is 0.107. The parameter obtained through nonlinear dimensionality reduction is relatively small, the kernel techniques may reduce the consistency. As a result, when using Cohen's kappa to judge the Parkinson's diagnostic accuracy, linear dimensionality reduction methods should be prioritized.

In addition to what method is effective under specific situation, analyzing the reason for what cause the varying results is important as well when applying different PCA methods. The effective information in the original data set will inevitably be reduced after the data dimension is reduced by using the principal component analysis method, because we will lose some information when performing PCA operations. In addition, the computer does not retain the analytical solutions when performing algorithm operations. Instead, it adopts the numerical solution. Due to the use of numerical solutions, there may be rounding errors and the impact of large numbers covering small numbers. Therefore, when you reduce the dimension of the dataset and project the reduced dimension data back to the original dataset, errors occur. The nonlinear PCA dimensionality reduction algorithm is more complex because it involves the use of kernel functions to map the original data into high-dimensional space. Compared with the original data set, nonlinear operation and numerical error make the data set lose more information. As a result, the RMSE of nonlinear PCA methods is bigger than the liner one, and the Cohen's kappa of the nonlinear is smaller than the liner one.

PCA maps high-dimensional data to a new low-dimensional space by searching for principal components in the data. The principal component is a linear combination of original features, which maximizes the variance of the mapped data. Due to the fact that general principal component analysis is a linear dimensionality reduction method, which only considers the global structure of the data, it only performs well in handling linear relational data. By contrast, Kernel principal component analysis is a nonlinear dimensionality reduction method. The Nonlinear dimensionality reduction method maps high-dimensional data to low dimensional space through nonlinear transformation, preserving the local and global structure of data. There are no obvious linear features in the Parkinson's dataset, so KPCA has a relatively good effect on this dataset, which is reflected in the higher overall accuracy of KPCA compared to the other two methods.

To compare the differences between two linear dimensionality reduction methods, computational time was introduced. Through experiments, it was found that the PCA algorithm spent 0.004 seconds on dimensionality reduction, while the Sparse PCA took 0.612 seconds. Perhaps in order to make the principal component coefficients (coefficients in front of each variable when forming the principal component) sparse, the Sparse principal component introduced an extra penalty function [2]. It is this penalty function that increases computational time. Although more time has been spent, this method can better explain the meaning of principal components.

## References

[1] H. Hotelling. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24:417–441, 1933.

[2] "Structured Sparse Principal Component Analysis" R. Jenatton, G. Obozinski, F. Bach, 2009

[3] Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Kernel principal component analysis." International conference on artificial neural networks. Springer, Berlin, Heidelberg, 1997.

[4] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10(5):1299–1319, 1998.

[5] Noble, W. What is a support vector machine? *Nat. Biotechnol.* 24, 1565–1567 (2006).

[6] Hsieh, W.W.. Machine learning methods in the environmental sciences: Neural networks and kernels: Cambridge university press, 2009: Chapter 7, pp.157-169

[7] 'Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection', Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. BioMedical Engineering OnLine 2007, 6:23 (26 June 2007)

[8] J. Cohen (1960). "A coefficient of agreement for nominal scales". Educational and Psychological Measurement 20(1):37-46.

[9] *Hyndman, Rob J.; Koehler, Anne B. (2006). "Another look at measures of forecast accuracy". International Journal of Forecasting. 22 (4): 679–688.*

[10] R. Artstein and M. Poesio (2008). "Inter-coder agreement for computational linguistics". Computational Linguistics 34(4):555-596.