Handwritten Digital Image Classification Based on PCA Dimensionality Reduction

Xingxing Li\textsuperscript{a}, Chao Duan\textsuperscript{b}, Yan Zhi\textsuperscript{c} and Panpan Yin\textsuperscript{d}

Department of electronics and information engineering, Guangzhou College of Technology and Business, FoShan, 528138, China

\textsuperscript{a}lixingxing_Edooy@163.com, \textsuperscript{b}MDL0857a@126.com, \textsuperscript{c}sparks666888@163.com, \textsuperscript{d}abc0604@126.com

Abstract. Dimensionality reduction is an important idea in machine learning. In the machine learning, some high-dimensional data sets are often encountered. In the case of high-dimensional data, data samples are sparse and distance calculations are difficult. Such problems are serious problems faced by all machine learning methods. "Dimensional disaster." In addition, linear correlation between features is easy to occur in high-dimensional features, which means that some features are redundant.

1. Introduction
In many fields of research and application, it is usually necessary to observe data containing multiple variables, collect large amounts of data, and analyze to find the law. Multivariate big data sets will undoubtedly provide a wealth of information for research and application, but also increase the workload of data collection to some extent [1]. More importantly, in many cases, there may be correlations between many variables, which increases the complexity of problem analysis [2]. If each indicator is analyzed separately, the analysis is often isolated and the information in the data cannot be fully utilized. Therefore, blindly reducing the indicator will lose a lot of useful information and lead to erroneous conclusions.

Therefore, it is necessary to find a reasonable method to reduce the loss of information contained in the original indicator while reducing the indicators that need to be analyzed, so as to achieve a comprehensive analysis of the collected data. Since there is a certain correlation between variables, it can be considered to change the closely related variables into as few new variables as possible, so that these new variables are irrelevant, so they can be represented by fewer comprehensive indicators [3]. Various types of information that exist in each variable. Principal component analysis and factor analysis belong to this type of dimensionality reduction algorithm. Dimensionality has the following advantages: 1) making the data set easier to use; 2) reducing the computational overhead of the algorithm; 3) removing noise; 4) making the results easy to understand.

2. Principle

2.1. Principal component analysis concept
PCA (Principal Component Analysis), the principal component analysis method, is one of the most widely used data dimensionality reduction algorithms. The main idea of PCA is to map n-dimensional
features to k-dimension. This k-dimensional is a new orthogonal feature, also called principal component, which is a k-dimensional feature reconstructed on the basis of the original n-dimensional features. The job of PCA is to sequentially find a set of mutually orthogonal coordinate axes from the original space. The selection of the new coordinate axes is closely related to the data itself. Wherein, the first new coordinate axis selection is the direction with the largest variance in the original data, and the second new coordinate axis selection is the plane orthogonal to the first coordinate axis, so that the variance is the largest, and the third axis is the first one. The variance of the plane orthogonal to the two axes is the largest. By analogy, n such coordinate axes can be obtained. With the new axes obtained in this way, we found that most of the variances are contained in the first k axes, and the subsequent axes contain almost zero variance. Therefore, we can ignore the remaining axes, leaving only the first k axes with most of the variance [4]. In fact, this is equivalent to retaining only the dimensional features that contain most of the variance, while ignoring the feature dimensions containing the variance of almost zero, to achieve dimensionality reduction of the data features.

In fact, by calculating the covariance matrix of the data matrix, and then obtaining the eigenvalue eigenvectors of the covariance matrix, the matrix composed of the eigenvectors corresponding to the k features with the largest eigenvalue (i.e., the largest variance) is selected. In this way, the data matrix can be transformed into a new space to achieve dimensionality reduction of data features.

Since there are two methods for obtaining the eigenvalue eigenvectors of the covariance matrix: eigenvalue decomposition covariance matrix and singular value decomposition covariance matrix, the PCA algorithm has two implementation methods: PCA algorithm based on eigenvalue decomposition covariance matrix, based on the SVD decomposition covariance matrix implements the PCA algorithm.

2.2. Singular value decomposition
In linear algebra, we have learned the eigendecomposition of matrices (the matrix here must be a square matrix of n×n). The relationship between matrix A and eigenvalues and eigenvectors is as follows:

\[ Av = \nu \]  

The A matrix is characterized as a feature, and the eigenvector \( \nu \) is a set of orthogonal vectors. The specific expression is as follows:

\[ A = QQ^{-1} \]  

Here, since n eigenvectors in Q are standard orthogonal bases, \( QT = Q^{-1} \) is satisfied, that is, Q is a unitary matrix.

The eigenvalue decomposition of the matrix has a relatively large limitation. It is required that the matrix A must be a square matrix (that is, a matrix in which rows and columns must be equal). Singular value decomposition can handle these general matrices. Assuming that matrix A is now an m×n matrix, we can write its singular value decomposition into the following form:

\[ A = UV^T \]  

Here U is a matrix of m×m, \( \Sigma \) is a matrix of m×n (except for values on the diagonal, the rest of the values are all 0), and V is a matrix of n×n. Here, the matrices U and V are both unitary matrices, that is to say \( UTU = I \), \( VTV = I \) (where I is the unit matrix). The definition of SVD is as shown in the following figure 1.
The matrix $A^TA$ is an $n \times n$ square matrix, we do eigenvalue decomposition on the square matrix, and we can get the following expression:

$$(A^TA)\mathbf{v}_i = \lambda_i \mathbf{v}_i$$  \hspace{1cm} (4)

The vector $\mathbf{V}_I$ obtained here is referred to as a right singular vector. From this value we can get $u_i$ (left singular vector, vector in square matrix $U$) and $\sigma_i$ (value on the diagonal in matrix, which is our singular value).

$$\sigma_i = \sqrt{\lambda_i}$$  \hspace{1cm} (5)

$$u_i = \frac{1}{\sigma_i} A \mathbf{v}_i$$  \hspace{1cm} (6)

For singular values, it is similar to the eigenvalues in our eigendecomposition, and is also arranged in the singular value matrix $\Sigma$ from large to small, and the singular value is reduced particularly fast. In many cases, the sum of the singular values of the top 10% or even the top 1% accounts for more than 99% of the sum. In other words, we can use the largest $K$ singular values and the corresponding left and right singular vectors to approximate the description matrix, as shown in the following figure2.

**Figure 1.** SVD decomposition process.

The specific expression is as follows:

$$A_{mn} = U_{mkxk} \Sigma_{kxk} V^T_{kn} \approx U_{mkxk} \Sigma_{kxk} V^T_{kn}$$  \hspace{1cm} (7)
In parallel computing, singular value decomposition is very fast. However, the matrix decomposed by the singular value is not very interpretative, and it feels like a black box, but it is still used in many fields.

2.3. Principal component analysis

The PCA algorithm can be said to be the most commonly used algorithm, and it is widely used in the fields of data compression and redundancy elimination.

In fact, for a straight line we can choose any two linearly independent unit vectors (the unit vector is chosen because it is easy to calculate, refer to the above reminder) to indicate its basis. Two unit diagonals are used as the basis (here we select the basis is orthogonal, because orthogonal groups have better properties, so orthogonal bases are generally chosen). With these understandings, we can think that the meaning of multiplying two matrices is to transform each column of the vector in the right matrix into the space represented by each row vector in the left matrix (Reminder: the conclusion will be combined with the following Discussion from high dimensional to low dimensional mapping in PCA).

First, suppose that there is such a hyperplane in low-dimensional space. Mapping data from high-dimensional to hyperplane will maximize the variance between samples (the variance between samples is the largest, which means that samples are still mapped after mapping to low-dimensional it can better preserve the difference between samples. If there are a large number of samples overlapping after mapping, there will be a large number of useless samples in the mapped samples. It is assumed that the sample is centralized, that is, the mean of the samples is made. 0, then the variance of all samples can be expressed as:

\[ \text{Var}(a) = \frac{1}{m} \sum_{i=1}^{m} a_i^2 \]  

(8)

So as long as a set of bases is found, the variance of the sample onto each base is the largest, but there is a problem because the variance is the largest when looking for each base, which may result in the following base and the previous base. This makes no sense. Therefore, it is obviously inappropriate to simply choose the direction of the largest variance. We introduce a constraint. When looking for such a set of orthogonal bases, we must not only maximize the variance after mapping, but also make the covariance between the features zero (covariance indicates the correlation between features, making the features linearly independent) The variance is expressed as follows:

\[ \text{Cov}(a, b) = \frac{1}{m} \sum_{i=1}^{m} a_i b_i \]  

(9)

The covariance matrix can express both the variance and the covariance. The covariance matrix is a symmetric matrix. The diagonal of the matrix is the variance. The rest of the values are covariances. The specific representations are as follows:

\[ \frac{1}{m} XX^T = \begin{bmatrix} \frac{1}{m} \sum_{i=1}^{m} a_i^2 & \frac{1}{m} \sum_{i=1}^{m} a_i b_i \\ \frac{1}{m} \sum_{i=1}^{m} a_i b_i & \frac{1}{m} \sum_{i=1}^{m} b_i^2 \end{bmatrix} \]  

(10)

Then the problem is solved, as long as the covariance matrix is diagonalized, the diagonalized matrix except the value on the diagonal, the rest of the value is 0.

Let the covariance matrix corresponding to the original data matrix X be C, and P be a matrix of bases in rows. Let \( Y = PX \), then \( Y \) is the data after \( X \) transforms to \( P \). Let \( Y \)'s covariance matrix be \( D \), and we derive the relationship between \( D \) and \( C \):

\[ D = YY^T = PXX^TP^T = PPC^T \]  

(11)
The optimization goal becomes to find a matrix $P$, which is an orthogonal base matrix, and the diagonal elements are arranged in order from largest to smallest, then the first $K$ rows of $P$ are the bases to be searched, and the matrix consisting of the first $K$ rows of $P$ Multiplying by $X$ reduces $X$ from $N$ to $K$ and satisfies the above optimization conditions.

Therefore, now we only need to diagonalize the covariance matrix $C$, and since $C$ is a real symmetric matrix, the eigenvectors under different eigenvalues are orthogonal to each other. Diagonalize the covariance matrix $C$.

$$E^TC^T=\Lambda=\begin{bmatrix} \lambda_1 & \cdots & \lambda_n \end{bmatrix} \quad (12)$$

3. Experiment

3.1. Introduction data

The data is a well-known handwritten digital image, each image is 28'28 in size, as shown in the following figure 3. The training set type is (3823, 65), where 3823 is the number of samples, each sample consists of 64 features, 1 tag; the training set has 1797 samples.

![Figure 3. Dataset.](image)

3.2. Evaluation criterion

Accuracy is the most common metric and is easy to understand. It is the number of samples that are paired divided by the number of samples. Generally speaking, the higher the correct rate, the better the classifier.

$$Acc = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

Precision rate is for the prediction result, which indicates how many of the samples with positive prediction true positive samples are. Then there are two possibilities for predicting positive. One is to predict the positive class as a positive class (TP), and the other is to predict the negative class as a positive class (FP).

$$P = \frac{TP}{TP+FP} \quad (14)$$
Recall rate is for the original sample, which indicates how many positive examples in the sample are predicted correctly. There are also two possibilities. One is to predict the original positive class as a positive class (TP), and the other is to predict the original positive class as a negative class (FN).

\[
R = \frac{TP}{TP + FN}
\]

(15)

3.3. Experiment results
The experiment was classified using a linear classifier, and Table 1 is the result of different feature dimensions. Among them, 64-dimensional is the original feature dimension without PCA dimension reduction, and the others are PCA dimension reduction in every 4 dimensions. The specific results are shown in Table 1.

Table 1. Different feature dimension classification results.

| Feature Dimension | 64  | 60  | 56  | 52  | 48  | 44  | 40  | 36  | 32  |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Precision         | 94.29% | 94.50% | 94.28% | 94.40% | 94.62% | 94.48% | 94.47% | 94.70% | 94.06% |
| Recall            | 94.16% | 94.44% | 94.21% | 94.32% | 94.55% | 94.38% | 94.38% | 94.60% | 93.99% |
| Accuracy          | 94.16% | 94.44% | 94.21% | 94.32% | 94.55% | 94.38% | 94.38% | 94.60% | 93.99% |

According to the results of the above table, different feature dimensions produce different results. So draw the line graph of the recall as follow Figure 4, and find that the recall value is always in a state of oscillation before the PCA falls to 36 dimensions; when the PCA drops to 36 dimensions, the recall value is the highest; after the PCA drops to the 36 dimension, the recall value is always is in a downward trend.

Figure 4. To format a figure caption use the Microsoft Word template style

4. Conclusion
Dimensionality reduction is a method of preprocessing data for high-dimensional feature data. Dimensionality reduction is the most important feature of retaining high-dimensional data, removing noise and unimportant features, thereby achieving the goal of improving data processing speed. In actual production and application, dimension reduction can save us a lot of time and cost within a certain information loss range. Dimensionality reduction has also become a very widely used data preprocessing method.
Acknowledgments
This work was financially supported by fund project, that is, Guangzhou Institute of industry and commerce college level research project in 2019"Research and design of S band radio frequency front-end" KA201937.

References
[1] Beam A L, Kohane I S. Big Data and Machine Learning in Health Care [J]. JAMA, 2018.
[2] Zhang J, Song W, Jiang B, et al. Measurement of lumber moisture content based on PCA and GS-SVM [J]. Journal of Forestry Research, 2018(2): 1-8. Reference to a chapter in an edited book:
[3] Nasution M Z F, Sitompul O S, Ramli M. PCA based feature reduction to improve the accuracy of decision tree c4.5 classification [J]. Journal of Physics: Conference Series, 2018, 978:012058-.
[4] Thierry B, Sajid J, Hongyang Z, et al. On the Applications of Robust PCA in Image and Video Processing [J]. Proceedings of the IEEE, 2018, 106(8): 1427-1457.