Mathematical Charcateristics of High-Dimensional Data

Yongwen Su

Department of Computer Science, Shanghai Jiao Tong University

Abstract

This article will review the characteristics of high-dimensional data and analyze some methods of reducing dimensions. In this article, I will introduce some important mathematical tools and implement Principle Components Analysis to process images. Finally, I will propose a hypothesis of minimum dimension and verify it by experiment. All my codes for experiment are on Implement-PCA-reconstruct-image(https://github.com/susufancy/Implement-PCA-reconstruct-image).

1 INTRODUCTION

Many data in real life have multiple dimensions, we call data with multiple dimensions high-dimensional data. High-dimensional data is widely used in data mining, data visualization and neural network training. Considering the direct use of high-dimensional data consumes a lot of resources, people usually reduce the dimensions of high-dimensional data. There already exists many related work on dimensionality reduction [2, 3, 6, 7]. Some researchers make some important work on nonlinear dimensionality reduction [6, 7], which inspires the research on dimensionality reduction. Some mathematical tools used for dimensionality reduction [1, 2, 4] are also constantly developing. In this article, I will introduce some important mathematical tools and implement PCA to process images.

1.1 Law of large numbers

In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times.

According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trial are performed.

Let $x_1, x_2, \cdots, x_n$ be n independent samples of a random variable x, then

$$Pr(|\frac{x_1 + x_2 + \cdots x_n}{n} - E(x)| \geq \epsilon) \leq \frac{Var(x)}{n\epsilon^2}$$

Proof. (by Chebychev's Inequality)

$$Pr(|\frac{x_1 + x_2 + \cdots x_n}{n} - E(x)| \geq \epsilon) \leq \frac{Var(\frac{x_1+x_2+\cdots x_n}{n})}{\epsilon^2}$$

$$= \frac{Var(x_1 + x_2 + \cdots x_n)}{n^2\epsilon^2} = \frac{Var(x)}{n\epsilon^2}$$

Application. $x, y : [z_1, z_2, \cdots, z_d]$ with $z_i \in N(0, 1)$
$|x|^2 \approx d, |y|^2 \approx d, |x - y|^2 = \sum_{i=1}^{d}(x_i - y_i)^2 = 2d$
$|x - y|^2 \approx |x|^2 + |y|^2$

Pythagorean theorem: random d-dimensional x, y are approximately orthogonal.If we scale these random points to be unit length and call x the North Pole, much of the surface area of the unit ball must lie near the equator.

1.2 Unit ball in d-dimensions

Most of the volume of a unit ball in high dimensions is concentrated near its equator no matter which direction is defined to be the North Pole.

Theorem: For $c \geq 1$ and $d \geq 3$, at least a $1 - \frac{2e^{\frac{-c^2}{2}}}{c}$ fraction of the volume of the d-dimensional unit ball has $|x_1| \leq \frac{c}{\sqrt{d-1}}$

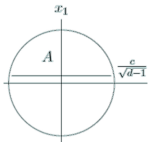


Figure 1:Unit ball in d-dimensions

This theorem tells us that the distribution of high-dimensional data in a certain direction is very dense or that the range is very narrow.

1.3 Gaussian Annulus Theorem

d-dimensional spherical Gaussian with 0 means and variance σ^2 in each coordinate has density function:

$$p(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} exp(-\frac{|x|^2}{2\sigma^2})$$

The radius of the ball need to be nearly $\sqrt{d}$ before there is a significant volume and hence significant probability mass.

Theorem: For a d-dimensional spherical Gaussian with unit variance in each direction, for any $\beta \leq \sqrt{d}$, all but at most $3e^{-c\beta^2}$ of the probability mass lies within the annulus

$$\sqrt{d} - \beta \leq |x| \leq \sqrt{d} + \beta$$

where c is a fixed positive constant.

2 DIMENSIONALITY REDUCTION

High-dimensional data dimensionality reduction is very practical and widely used in big data analysis and neural network training. The theorem introduced in the first part reveals that the distribution of high-dimensional data has certain characteristics. Therefore, the principle of dimensionality reduction of high-dimensional data is to keep the original characteristics of the dimensionality-reduced data as much as possible.

2.1 Random Projection

If we define the nearest neighbor of point x is the point whose distance from x is minimal. Then finding the nearest neighbor search n points from R_d will cost more than expected.

$$\begin{bmatrix} v_{11} & v_{21} & \cdots & v_{n1} \\ v_{12} & v_{22} & \cdots & v_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ v_{1d} & v_{2d} & \cdots & v_{nd} \end{bmatrix} \tag{1}$$

We need do dimension reduction: project the database points to a k dimensional space with $k << d$. And this projection must keep the relative distances between points as more as possible.

The projection fuction: Pick k vectors $u_1, u_2, \cdots, u_k$ independently from the Gaussian distribution $\frac{1}{(2\pi)^{d/2}\sigma^d} exp(-\frac{|x|^2}{2\sigma^2})$, for any vector v, the projection $f : R^d \to R^k$ is:

$$f(v) = (u_1 \cdot v, u_2 \cdot v, \cdots, u_k \cdot v)$$

This projection satisfies linearity which means that

$$f(v_1 - v_2) = f(v_1) - f(v_2)$$

$$|f(v)| \approx \sqrt{k}|v|$$

Random Projection Theorem: Let v be a fixed vector in R^d and let f be defined as above. Then there exists constant $c > 0$ such that for $\epsilon \in (0, 1)$

$$Pr(||f(v) - \sqrt{k}|v|| \geq \epsilon\sqrt{k}|v||) \leq 3e^{-ck\epsilon^2}$$

This theorem guarantees that the error of the relative distance between the data after the dimensionality reduction operation is within a certain range, thus illustrating the practical significance of the random projection method. A more in-depth study of the random projection method requires the use of JL theorem (Johnson-Lindenstrauss Lemma) to ensure that the data error is within an acceptable range [3].

2.2 Johnson-Lindenstrauss Lemma

In mathematics, the Johnson–Lindenstrauss lemma is a result concerning low-distortion embeddings of points from high-dimensional into low-dimensional Euclidean space[1]. The lemma states that a set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. The theorem is described as follows:

For any $0 < \epsilon < 1$ and a set X of m points in R^N, let $n \geq \frac{8lnm}{\epsilon^2}$, there is a linear map $f : R^N \rightarrow R^n$ such that for any set of m points in R^N, the random projection f defined above has the property that for all pairs of points v_i and v_j, with

$$\frac{||f(v_i) - f(v_j)||^2}{(1+\epsilon)} \leq ||v_i - v_j||^2 \leq \frac{||f(v_i) - f(v_j)||^2}{(1-\epsilon)}$$

Johnson-Lindenstrauss lemma has uses in compressed sensing, manifold learning, dimensionality reduction, and graph embedding. Much of the data stored and manipulated on computers, including text and images, can be represented as points in a high-dimensional space (see vector space model for the case of text). However, the essential algorithms for working with such data tend to become bogged down very quickly as dimension increases. It is therefore desirable to reduce the dimensionality of the data in a way that preserves its relevant structure.

2.3 Singular Value Decomposition

In linear algebra, the singular value decomposition (SVD) [4] is a factorization of a real or complex matrix that generalizes the eigendecomposition of a square normal matrix to any $m \times n$ matrix via an extension of the polar decomposition.
For an $m \times n$ matrix A of rank r there exists a factorization (Singular Value Decomposition) as follows:

$$A = U\Sigma V^T$$

The columns of $m \times m$ matrix U are orthogonal eigenvectors of AA^T. The columns of $n \times n$ matrix V are orthogonal eigenvectors of A^TA. Every elements in diagonal of the $m \times n$ matrix Σ is named singular value. Every element except the ones in diagonal is zero. Because AA^T and A^TA are both phalanx, so it's easy to compute their orthogonal eigenvectors. For example, compute the orthogonal value of phalanx A is equal to solve $|A - \lambda E| = 0$. The eigenvalue matrix is equal to the square of the singular value matrix. So $\sigma_i = \sqrt{\lambda_i}$.

2.4 Principal Components Analysis

As the name implies, PCA finds the most important aspect of data and replaces the original data with the most important aspect of data [2]. Specifically, if our data set is n-dimensional, there are m data in total $(x_1, x_2, \cdots, x_m)$. We hope to reduce the dimensions of these m data from n to n' dimensions, and hope that these m n' dimensional data sets can represent the original data set. We know that there will definitely be a loss of data from n-dimensional to n'-dimensional, but we hope that the loss will be reset. So how to make this n'-dimensional data represent the original data?

Let's take a look at the simplest case first, that is $n = 2$, $n' = 1$, which is to reduce the data from two dimensions to one dimension. The data is shown below. We hope to find a certain dimension direction, which can represent the data of these two dimensions. There are two vector directions listed in the figure, μ_1 and μ_2, so which vector can better represent the original data set? It can also be seen intuitively that μ_1 is better than μ_2.

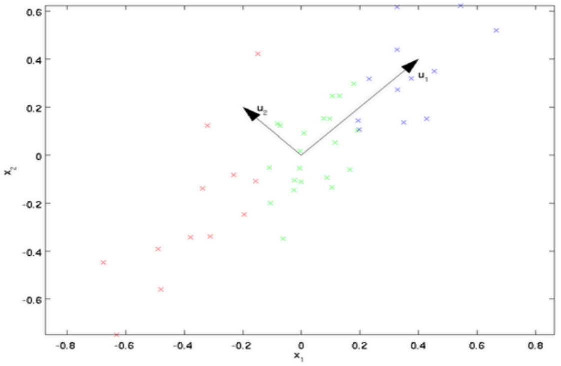


Figure 2: two-dimensional data

Why is μ_1 better than μ_2? There can be two explanations. The first explanation is that the distance between the sample point and this line is close enough, and the second explanation is that the projection of the sample point on this line can be separated as much as possible.

If we generalize n' dimension to any dimension, our criterion for reducing the dimension is: the distance between the sample point and this hyperplane is close enough, or the projection of the sample point on this hyperplane can be separated as much as possible. Based on the above two standards, we can get two equivalent derivations of PCA: Based on minimum projection distance and Based on maximum projection variance. Take based on maximum projection variance for example to find the PCA algorithm:

After m n-dimensional data $(x^{(1)}, x^{(2)}, \cdots, x^{(m)})$ has been centralized, which means $\sum_{i=1}^{m} x^{(i)} = 0$. The new coordinate system obtained after projection transformation is $\{w_1, w_2, \cdots, w_n\}$ and $||w||_2 = 1, w_i^T w_j = 0$. If we reduce the data from n dimensions to n' dimensions, that is, discard some of the coordinates in the new coordinate system, the new coordinate system is $\{w_1, w_2, \cdots, w_{n'}\}$. The projection of $x^{(1)}$ in n' coordinate system is $z^{(i)} = (z_1^{(i)}, z_2^{(i)}, \cdots, z_{n'}^{(i)})^T$, and the $z_j^{(i)} = w_j^T x^{(i)}$ is the coordinate of $x^{(i}$ in the low dimensional coordinate system.

For any $x^{(i}$, the projection variance in new coordinate system is $x^{(i)T} W W^T x^{(i)}$, we want to maximize $\sum_{i=1}^{m} x^{(i)T} W W^T x^{(i)}$ which equals to maximize $\sum_{i=1}^{m} tr(W^T X X^T W)$. The Lagrange function is

$$J(W) = tr(W^T X X^T W + \lambda(W^T W - I))$$

. After differentiating W, we get

$$X X^T W = (-\lambda) W$$

. As can be seen from the above, W is a matrix of n eigenvectors of XX^T, and λ is a matrix of eigenvalues value of XX^T, the eigenvalues are on the main diagonal, and the remaining positions are 0. When we reduce the data set from n dimensions to n' dimensions, we need to find the feature vector corresponding to the largest n' feature values. The matrix W composed of these n feature vectors is the matrix we need. For the original data set, we only need to use $z^{(i)} = W^T x^{(i)}$ to reduce the original data set to the n'-dimensional data set with the smallest projection distance.

3 RESULTS ON IMAGE DATA

In this section, I'll first provide implementation details. Then, I'll show qualitative results on challenging examples. Finally, I'll compare my method to previous work. You can find all my code files on .

3.1 Implementation Details

All my experiments are performed on CPU. You need install python($\geq$ 3.6). You also need install opencv, numpy, PIL, pandas packages to make sure the programs work.
All images data this project used are from CIFAR-10 dataset(CIFAR-10). The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. In this experiment, I used only a very small amount of CIFAR-10 dataset to display the application of PCA method.

3.2 Qualitative Results

In this section, I will introduce some results of images data process by PCA method. The results include two similar original images and reconstructed images after PCA dimensionality reduction, as well as reconstructed images after PCA dimensionality reduction of different types of images with obvious differences.

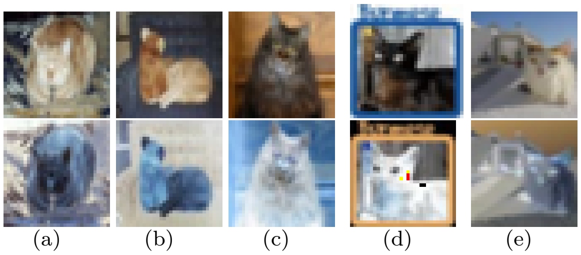


Figure 1: Similar cat images. The five pictures above are the original pictures obtained from CIFAR-10, and the five pictures below are the new pictures obtained by data reconstruction after PCA dimension reduction.

We can find some patterns. The bright part in the original image becomes the dark part in the reconstructed image. This means that some coordinate axes in the reconstructed image are opposite to those in the original image. There is no change in the degree of contrast between light

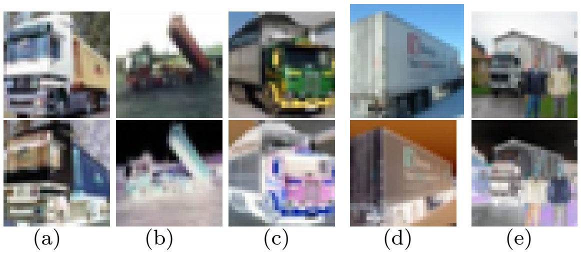


Figure 2: Similar truck images. The five pictures above are the original pictures obtained from CIFAR-10, and the five pictures below are the new pictures obtained by data reconstruction after PCA dimension reduction.
We can find that the experiment results for cat and truck using PCA dimensionality reduction are very similar. The possible reason for this phenomenon is that PCA dimensionality reduction has similarities to the processing mode of image data.

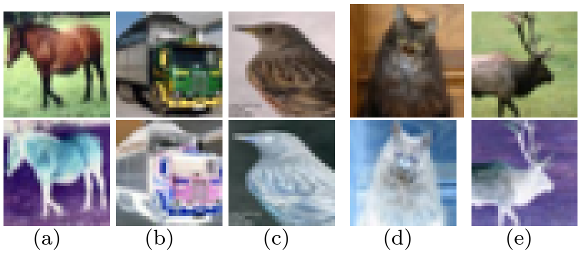


Figure 3: Five different images. The five pictures above are the original pictures obtained from CIFAR-10, and the five pictures below are the new pictures obtained by data reconstruction after PCA dimension reduction.
We can find that the experiment results for different images using PCA dimensionality reduction are very similar with trucks' or cats' which confirms that the possible reason for this phenomenon is that PCA dimensionality reduction has similarities to the processing mode of image data.

and dark, indicating that the PCA dimension reduction process retains the characteristics of the data itself, so that it can be distinguished from other data.

At the same time, a small amount of data (eg. Figure1 (d)) appear red and black noise, which shows that the coordinate system selected by the PCA dimension reduction process will make some data have unconventional projections.

Overall, the data after PCA dimensionality reduction retains most of the characteristics of the original data, but also has some changes, which confirms our previous mathematical derivation.

3.3 Compare Results With Different Number Of Eigenvectors

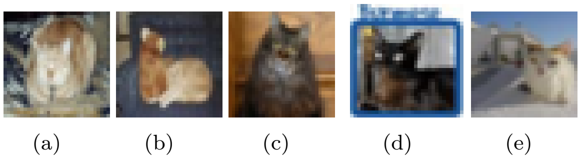

(a) (b) (c) (d) (e)

Figure 4: Original images.

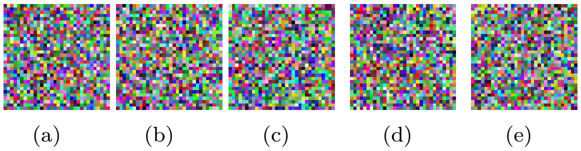

(a) (b) (c) (d) (e)

Figure 5: Reconstructed images using 3 eigenvectors.The images reconstructed using the 3 eigenvectors are basically noise, but the outline edges of the original image can be seen vaguely.

Observed from the experimental results: when the number of feature vectors is less than 4, the images reconstructed from these vectors are basically noise, and cannot reflect the original image. When the number of feature vectors is greater than or equal to 4, the reconstructed image is basically very close to the original image.

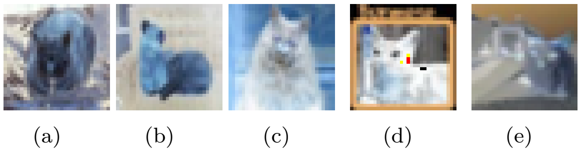
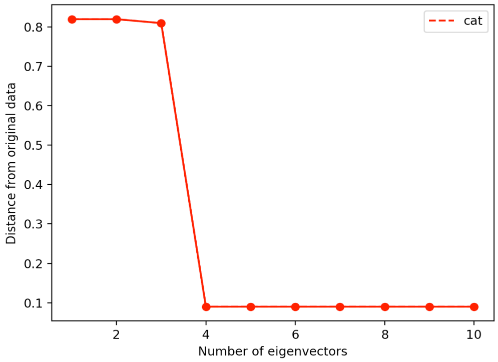

(a) (b) (c) (d) (e)

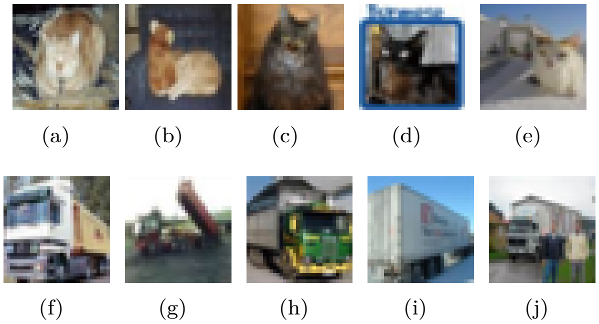
Figure 6: Reconstructed images using 4 eigenvectors.The images reconstructed using the 4 eigenvectors are very closed to fully constructed images.

This observation confirms our previous mathematical reasoning: high-dimensional data can be reduced to low-dimensional space through PCA and maintain its own nature. The data reconstructed by a certain number of eigenvectors is enough to characterize the original data. After exceeding this specific number, the features of the image will not increase with the number of eigenvectors.



By observing the distance between the reconstructed data and the original data, we can know that to express the features of the original data well, we need to use enough eigenvectors to reconstruct the data, and too many eigenvectors are meaningless. Therefore, it is very practical to find the smallest suitable number of eigenvectors, because this is the lower limit of PCA dimension reduction.

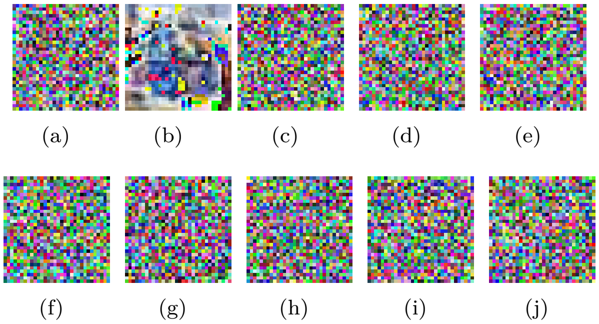
According to the experimental results, I speculate that the threshold of the number of eigenvectors is the number of samples minus one. Next, we will do some experimental verification.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 7: Original images



(a) (b) (c) (d) (e)

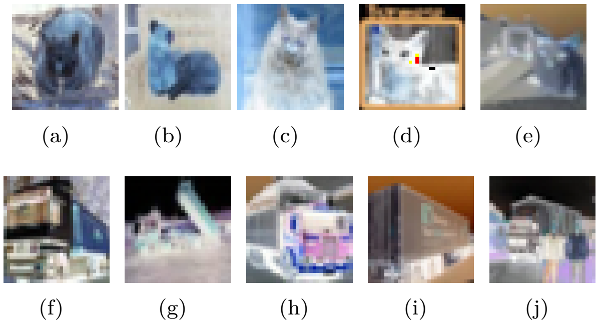
(f) (g) (h) (i) (j)

Figure 8: Reconstructed images using 8 eigenvectors. The images reconstructed using the 8 eigenvectors are basically noise, but the outline edges of the original image can be seen vaguely.

The PCA dimension reduction experiment results of ten sample data verify our hypothesis: the number of necessary eigenvectors is the number of samples minus one.

3.4 Disadvantages

Although PCA can retain most of the features of the original data, the meaning of each feature dimension of PCA is vague and not as interpretable as the original data. This can be seen from the experimental results. When the number of dependent dimensions is less than the number of samples minus one, basically the reconstructed data is noisy and cannot explain the meaning of each dimension of the data. In addition, PCA may discard components with small variances, but these components may also contain important information about sample differences, which may affect subsequent data processing.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 9: Reconstructed images using 9 eigenvectors.The images reconstructed using the 9 eigenvectors are very closed to fully constructed images.

4 Conclusion

We analyzed some characteristics of high-dimensional data, introduced two dimensionality reduction methods of random projection and PCA, and implemented the PCA algorithm. We use it for dimensionality reduction and reconstruction of image data, and propose a new method for processing image data. Finally, according to the experimental results, a speculation was made: the smallest dimension should be the number of samples minus one, and this reasoning was verified through multiple experiments.

References

[1] Sanjoy Dasgupta, Anupam Gupta, An elementary proof of the Johnson-Lindenstrauss Lemma, International Computer Science Institute, 1999: TR-99-006.

[2] S Wold, K Esbensen, P Geladi, Principal component analysis, Chemometrics and Intelligent Laboratory Systems, 1987: 37-52.

[3] E Bingham, H Mannila, Random projection in dimensionality reduction: applications to image and text data, Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining,August 2001: 245–250.

[4] Michael E. Wall, Andreas Rechtsteiner, Luis M. Rocha, Singular Value Decomposition and Least Squares Solutions, A Practical Approach to Microarray Data Analysis, 2003: 91-109.

[5] Laurens van der, Maaten Eric Postma, Jaap van den Herik, Dimensionality Reduction: A Comparative Review, J Mach Learn Res, 2009: TiCC TR 2009–005.

[6] Sam T. Roweis, Lawrence K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science 22 Dec 2000.

[7] D DeMers, GW Cottrell, None-Linear Dimensionality Reduction, Advances in Neural Information Processing Systems, 1992.