ISSN: 2249 - 7862 (Print)
Aureole, Vol. XII, December 2020 PP 157-165
www.aureoleonline.in

MACHINE LEARNING ALGORITHMS-DIMENSIONALITY REDUCTION USING PRINCIPAL COMPONENT ANALYSIS

*** Akhil S**

Abstract

Machine learning is the latest in a long line of attempts to distill human knowledge and reasoning into a form that is suitable for constructing machines and engineering automated systems. Machine Learning revolutionized the world of computers by teaching them to learn as they progress forward with large data, thus mitigating many previous programming pitfalls and impasses. Machine Learning builds algorithms, which when exposed to large data, can self teach and take decisions.When this technology is powered by Artificial Intelligence (AI) applications, the combination is powerful. In this paper, we will discuss principal component analysis (PCA), one of the most commonly used techniques for data compression and data visualization. It is also used for the identification of simple patterns, latent factors, and structures of high-dimensional data algorithm for linear dimensionality reduction.

Keywords **:** Machine Learning, Principal Component Analysis

* *II M.Sc Mathematics, PG and Research Department of Mathematics, NSS Hindu College, Changanacherry, Mail: akhilsnairhpd@gmail.com*

1. Introduction

Machine Learning (ML) revolutionized the world of computers by teaching them to learn as they progress forward with large data,thus mitigating many programming pitfalls and impasses. ML build algorithms which when exposed to large data, can self teach and take decisions. When this technology is powered by Artificial Intelligence (AI) applications, the combination is powerful. Machine Learning is steadily moving away from abstractions and engaging more in Business Problem Solving with support from AI and Deep Learning. It is expected that theoretical research in ML will pave the way for business probelm solving and business intelligence. AI + ML comnination will deliver systems that "Understand, Learn, Predict, Adapt and Operate autonomously". Deep Learning is one of the latest reserach trends in ML Algorithms, right now in this direction. It is a form of the Artificial Neural networks with many neurons/layers.

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (i.e., accounts for as much of the variability in the data as possible), and each succeeding component in turn hasthe highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric.

2.Principal Component Analysis(PCA)

Given data {x_1, x_2, ...x_n }where each element x_j has m attributes {a_{j1}, a_{j2}, ...a_{jm} } and j = 1, 2,.....n. While applying machine learning classificaction or clustering algorithm, it is not expected from all the features to have valuable contributions to the resulting classificaction model.The attributes with the least contributions do act as noise, and reducing them from the data will result in a more effective classificaction or clustering model.

The PCA is a multivariate statistical technique aiming at extracting the features that represent most of the information in the given data and eliminating the least features with least information. Therefore, the main purpose of using the PCA is to reduce the dimensionality of the data without seriously affecting the structure of the data. When collecting real data, usually the random variables that represent the data attributes are expected to be highly correlated. The correlations between these random variables can always be seen in the covariance matrix. The variances of the

random variables are found in the diagonal of the covariance matrix. The sum of the variances (diagonal elements of the covariance matrix) gives the overall variability.

The PCA works to replace the original random variables with other sets of orthonormalset of vectors called the principal components. The first principal component is desired to pass as much closer as possible to data points, and the projection of the data into the space spanned by the first component is the best projection over spaces spanned by other vectors in one dimension. The second principal component is orthogonal to the first principal component and the plain spanned by it together with the first component is the closest plane to the data. The third,fourth, and other such components feature at the same logic.

Given an m-dimensional data D of the form: The data D can be visualized by Figure 1.Hence, each data element x_j is represented by an m-dimensional row vector.

$$D = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

The data D can be visualized by Figure 1.1 Hence, each data element x_j is represented by an m-dimensional row vector

The main purpose of using the PCA is to reduce the high-dimensional data with a dimension m into a lower dimensional data of dimension k, where $k \leq m$.(see figure 2) The basis elements of the reduced space V constitute an orthonormal set. That is,if $\{v_1, v_2 \dots v_k\}$ is the basis of V, then:

		Attributes			
		a_1	a_2	$\cdots$	a_m
Instances	x_1	a_{11}	a_{12}	$\cdots$	a_{1m}
	x_2	a_{21}	a_{22}	$\cdots$	a_{2m}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	x_N	a_{N1}	a_{N2}	$\cdots$	a_{Nm}

Figure 1: The shape of a dataset consisting of N feature vectors, each has m attributes.

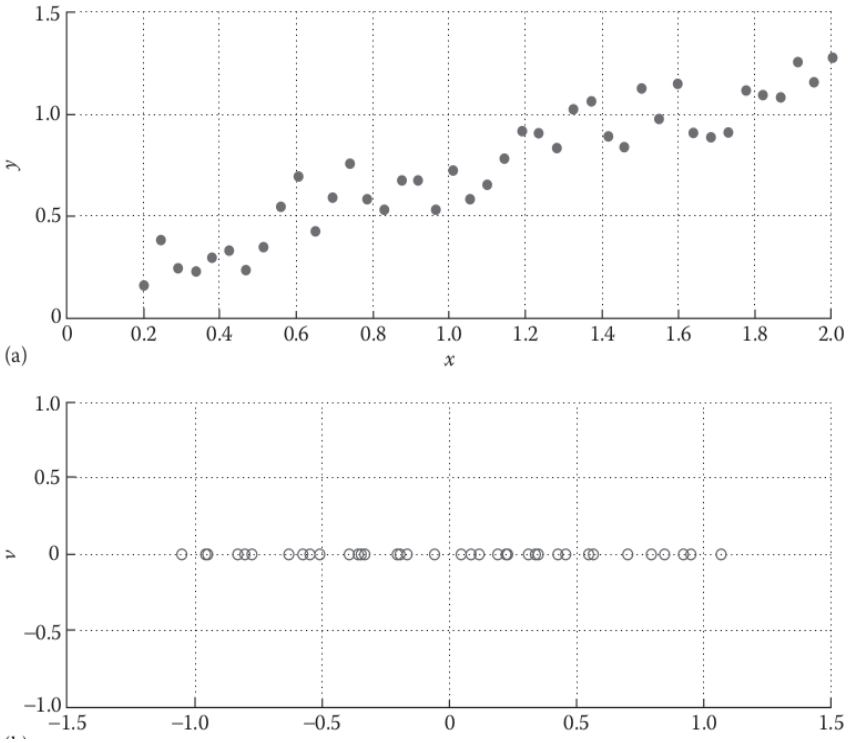

Figure 2: (a) The original two-dimensional data in the feature space and (b) the reduced data in one-dimensional space.

$$v_i^T . v_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$$

The singular value decomposition (SVD) plays the main role in computing the basis elements $\{v_1, v_2 \ldots v_k\}$.

2.1 The SVD and Dimensionality Reduction

Suppose that C is the covariance matrix that is extracted from a given dataset D. The elements of the covariance matrix are the covariances between the random variables representing the data attributes (or features). The variance of the random variables lies in the diagonal of C, and their sum is the total variability. Generally, the SVD works to express matrix *C* as a product of three matrices: *U*,Σ,and *V* . Because matrix *C* is an *m x m* matrix, the SVD components of matrix *C* are given by

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

where:

U & V are orthogonal *m x m* matrix

Σ is an *m x m* diagonal matrix.

The diagonal elements of Σ are the eigenvalues of the covariance matrix C , ordered according to their magnitudes from bigger to smaller. The columns of U (or V) are the eigenvectors of matrix C that correspond to the eigenvalues of C .

By factoring matrix C into its SVD components, we would be rewarded with three benefits:

1. The SVD identifies the dimensions along which data points exhibit the most variation, and order the new dimensions accordingly. The total variation exhibited by the data is equal to the sum of all eigenvalues, which are the diagonal elements of the matrix Σ and the variance of the jth principal component is the jth eigenvalue.
2. Replace the correlated variables of the original data by a set of uncorrelated ones that are better exposed to the various relationships among the original data items. The columns of matrix U, which are the eigenvectors of C, define the principal components, which act as the new axes for the new space.
3. As a result of benefit (2), the SVD finds the best approximation of the original data points using fewer dimensions. Reducing the dimension is done through selecting the first k principal components, which are the columns of matrix U.

3. Implimentation of PCA

The steps to impliment PCA as follows

Step 1: Compute the mean of the data as follows:

$$\bar{x} = \begin{bmatrix} \bar{a_1} & \bar{a_2} & \dots & \bar{a_n} \end{bmatrix} = \frac{1}{N} \sum_{j=1}^{N} x_j \tag{3}$$

Step 2: Normalize the data by subtracting the mean value $\bar{x}$ from all the instances x_i giving the adjusted mean vectors $x_i - \bar{x}$

$$\bar{D} = \begin{bmatrix} x_1 - \bar{x} \\ x_2 - \bar{x} \\ \vdots \\ x_n - \bar{x} \end{bmatrix}$$

Step 3: Construct the covariance matrix $Cov(\bar{D}) \in R^{m \times m}$ from D as follows:

$$Cov(\bar{D})_{ij} = \frac{1}{N} \sum (x_{ni} - \bar{a_i})(x_{nj} - \bar{a_j} \tag{4}$$

Step 4: Let $\{\lambda_1, \lambda_2, \dots \lambda_m\}$ be the set of eigen values of the covariance matrix $Cov(\bar{D})$ in a way such that $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_m|$ and $\{v_1, v_2, \dots v_m\}$ are the corresponding set of eigen vectors. The k vectors $v_1, v_2, \dots v_k$ represent the first k principal components.

Step 5: Construct the following matrix

$$F = \begin{bmatrix} v_1 & v_2 & \dots v_k \end{bmatrix} \in R^{m \times k}$$

The reduced data i k dimension will be

$$\bar{D}_{reduced} = \bar{D}.F \in R^{N \times k}$$

3.1 Number of Principal Component to Choose

When applying the PCA, the variances of the random variables are represented by the eigenvalues of the covariance matrix that are located at the main diagonal of matrix Σ. In matrix Σ the eigenvalues are sorted according to their magnitudes, from bigger to smaller. That means, the principal component with bigger variance comes first.

Let $S \in R^n$ be the vector of the diagonal elements of Σ. The variance covered by choosing the first k principal components is given by

$$\frac{\sum_{j=1}^{k} S_j}{\sum_{j=1}^{m} S_j}$$

	Mcg	*Gvh*	*Lip*	*Chg*	*Aac*	*Alm1*	*Alm2*
Mcg	0.0378	0.0131	0.0025	0.0004	0.0052	0.0166	0.0068
Gvh	0.0131	0.0219	0.0006	0.0001	0.0013	0.0055	–0.0037
Lip	0.0025	0.0006	0.0078	0.0008	0.0008	0.0018	–0.0011
Chg	0.0004	0.0001	0.0008	0.0007	–0.0001	0	–0.0003
Aac	0.0052	0.0013	0.0008	–0.0001	0.0149	0.0074	0.0065
Alm1	0.0166	0.0055	0.0018	0	0.0074	0.0464	0.0365
Alm2	0.0068	–0.0037	–0.0011	–0.0003	0.0065	0.0365	0.0437

Figure 3: The Covariance Matrix of the Ecoli Dataset, with Seven Random Variables Representing the Seven Data Attributes

We applied the PCA algorithm to the “ecoli" data, and the variance retained by the first k principal components are explained in figure 4. This means that if we select only the first principal component, then we can retain only 51.6% of the variance. If we select the first two principal components, then 76% of the variance will be retained, and so on.. By knowing this, we can determine how many principal components to select. For example, if we need to retain 90at least of the variance, then we shall choose the first four principal components, if we need 99% of the variance, then we shall select the first six principal components

K	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Var(k)	0.5162	0.7604	0.8446	0.9187	0.9678	0.9962	1

Figure 4: Variances of the Seven Principal Components of Ecoli Data

3.2 Data Reconstruction Error

If $\tilde{x_i} = x_i - \bar{x} \in \bar{D}$, the $z_i = \tilde{x_i}.F \in R^k$ R k is the projection of $\tilde{x_i}$ in the new linear space spanned by the orthonormal basis vectors $\{v_1, v_2, \ldots, v_k\}$. Moving from R^m to R^k is reversible and one can reconstruct $\hat{x_i}$, which is an approximation to $\tilde{x_i}$ where:

$$\hat{x_i} = z_i.F^T \in R^m$$

The error associated with the data reconstruction $Error_{Rec}$ is given by

$$Error_{Rec} = \sum_{i=1}^{N} \|\tilde{x_i} - \hat{x_i}\|^2 \tag{5}$$

The reconstruction errors obtained by applying the PCA with k principal components are explained in the Figure 5

K	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Error(*k*)	14.2171	4.9019	4.316	2.8588	1.6533	0.2207	0

Figure 5: Reconstruction Errors Obtained by Applying PCA with k Principal Components

```
function [Dred, Drec, Error] = PCAReduction(D, k)
% D is a dataset, consisting of N instances (rows)
and m features (columns)
% k is the new data dimension, where 0 <= k <=m
[N, m] = size(D) ; % Number of data instances N and
number of features is m
mu = mean(D) ;        % mean of data set -> mu

Db = zeros(size(D)) ; % Db is the mean adjusted
dataset
    for j = 1 :  N
        Db(j, :)  = D(j, :)  - mu ;
        % subtracting the mean from the N data
instances
end
```

```
C = zeros(m) ; % C is the covariance matrix
for i = 1 :  m
    for j = 1 :  m
        C(i, j) = 0 ;
        for n = 1 :  N
            C(i, j) = C(i, j) + Db(n, i)*Db(n, j)/N ;
        end
    end
end
```

```
[U,S,V] = svd(C) ;          % Applying the SVD to the
covariance matrix
F = V(:, 1:k) ;          % F is the matrix with the first k components
Dred = Db*F ; % Generating the reduced data
Drec = Dred*F' ; % Reconstructing the mean adjusted
data in m-dimensions

for j = 1 :  N
    Drec(j, :)  = Drec(j, :)  + mu ; % Reconstructing
the original data
end
Error = norm(Drec - D,2)^2 % Computing the error
```

4 MATLAB ® Code Applies the PCA

We applied the above algorithm to the Iris dataset with k = 2. In Figure 6, we show the plot of the reduced data (which is 2D)

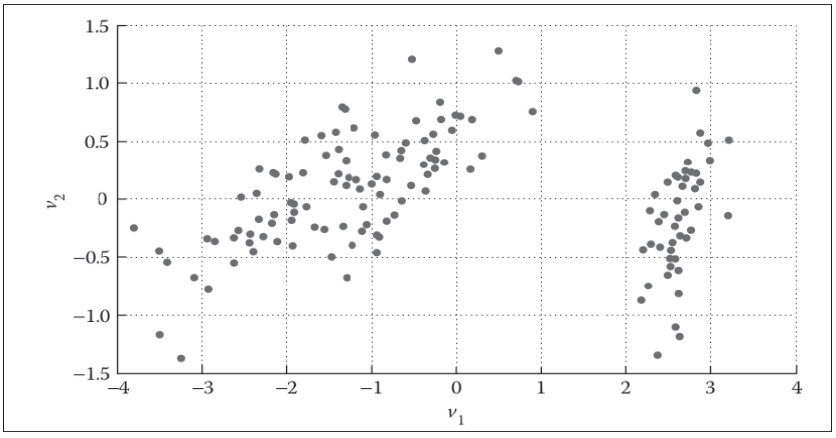

Figure 6: Two-dimensional reduced data

We also plotted the data in a three-dimensional reduced space in Figure 7.

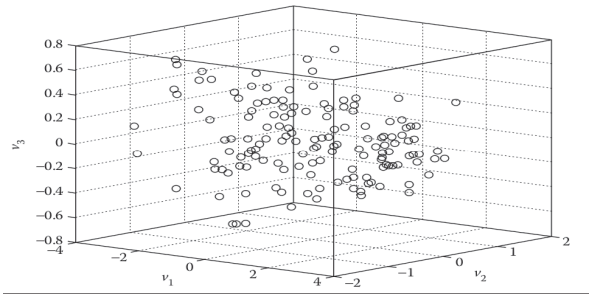

Figure 7: Three-dimensional reduced space.

Conclusion

PCA is sensitive to the relative scaling of the original variables. Working directly with high-dimensional data, such as images, comes with some di culties:It is hard to analyze, interpretation is dicult,visualization is nearly impossible, and (from a practical point of view) storage of the data vectors can be expensive. However, high-dimensional data often has properties that we can exploit. For example, high-dimensional data is often overcomplete, i.e., many dimensions are redundant and can be explained by a combination of other dimensions. Furthermore, dimensions in high-dimensional data are often correlated so that the data possesses an intrinsic lower-dimensional structure. Dimensionality reduction exploits structure and correlation and allows us to work with a more compact representation of the data, ideally without losing information. We can think of dimensionality reduction as a compression technique, similar to jpeg or mp3, which are compression algorithms for images and music.

References

1.Rene Vidal, Yi Ma, S. Shankar Sastry. *Generalized Principal Component Analysis.* Springer, 2016.

2.Mare Peter Deisneroth, A. Aldo Faisal, Cheng Soon Ong *Mathematics for Machine Learning* Cambridge University Press(to be published)

3.I.T Jolliffe *Principal Component Analysis* 2 ed. Springer

4.Mohssen Mohammed, Muhammad Badruddin Khan, Eihab Bashier Mohammed Bashier. *Machine Learning Algorithms and Applications.* CRC Press,2016

5.Aggarwal, C. C., Yu, P. S. *Outliner detection for high dimensional data,* ACM SIGMOD Record, vol. 30, issue 2, 37{46, 2001.