Dimesionality Reduction using PCA, ICA and generalized PCA

Lokesh Surana (ES20BTECH11017)
Samar Singhai (BM20BTECH11012)

April 28, 2023

Outline I

1. Dimensionality reduction
2. Principal Component Analysis
    - Concept
    - Mathematics involved in PCA
    - Pseduo Code
    - Applications
    - Limitations
3. Independent Component Analysis (ICA)
    - Concept
    - How ICA works?
    - Pseduo Code
    - Applications
    - Limitations
4. Generalized PCA
    - Concept

Outline II

- How Generalized PCA works?
- Pseduo Code
- Applications
- Limitations

What is dimensionality reduction?

1. A technique in machine learning and data analysis. It allows us to simplify high-dimensional data sets by identifying the most important features and reducing the number of dimensions.
2. This can improve the efficiency and accuracy of algorithms, as well as help us gain a better understanding of the underlying patterns and relationships in the data.
3. **What are benifits of dimensional reduction?**

What is dimensionality reduction?

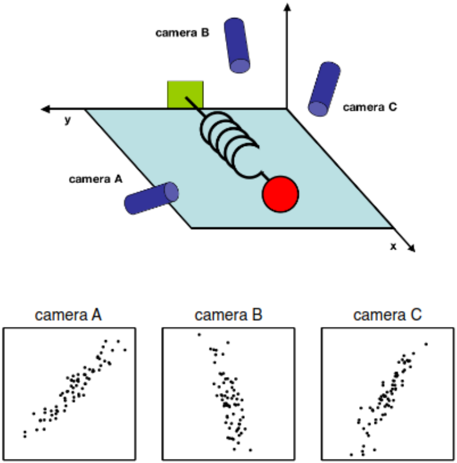

FIG. 1 A toy example. The position of a ball attached to an oscillating spring is recorded using three cameras A, B and C. The position of the ball tracked by each camera is depicted in each panel below.

Why dimensionality reduction is important?

1. **Speed up computations:** High-dimensional data requires a lot of computational resources to analyze, and working with such data can be very time-consuming. By reducing the dimensions of the data, we can speed up the computation and make it more efficient.
2. **Visualization:** High-dimensional data is difficult to visualize, but by reducing the dimensions of the data, we can plot the data in 2D or 3D and visualize the relationships between different variables.
3. **Improves model performance:** In many machine learning tasks, high-dimensional data can lead to overfitting and other problems. By reducing the dimensions of the data, we can improve the performance of our models and reduce the risk of overfitting.

Signal to Noise Ratio

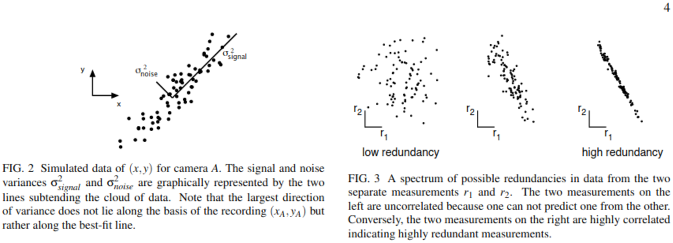
4
y
x
σ^2_{noise}
σ^2_{signal}
r_2
r_1
r_2
r_1
r_2
r_1
low redundancy
high redundancy

FIG. 2 Simulated data of (x, y) for camera A. The signal and noise variances σ^2_{signal} and σ^2_{noise} are graphically represented by the two lines subtending the cloud of data. Note that the largest direction of variance does not lie along the basis of the recording (x_A, y_A) but rather along the best-fit line.

FIG. 3 A spectrum of possible redundancies in data from the two separate measurements r_1 and r_2. The two measurements on the left are uncorrelated because one can not predict one from the other. Conversely, the two measurements on the right are highly correlated indicating highly redundant measurements.

Figure 2: Signal to Noise Ratio

PCA

1. The motivation behind Principal Component Analysis (PCA) is to reduce the dimensionality of a dataset while retaining as much of the variation in the data as possible. This is useful when the original dataset has many features or variables, making it difficult to visualize or analyze the data.
2. By identifying the principal components of the data, linear combinations of the original variables, PCA can represent the data in a lower-dimensional space that still captures the essential patterns of variation in the original data.
3. PCA can also perform data compression since the principal components can be used to reconstruct the original data with some loss of information.
4. PCA can be used to remove noise from the data since the principal components correspond to the directions of maximal variance in the data and are, therefore, less likely to be affected by noise.

Variance

1. A statistical measure that quantifies the amount of spread or dispersion in a dataset. It measures how far a set of data points are spread out from the mean.
2. In PCA, the variance of a dataset is the amount of information or variation contained in that dataset. The first principal component captures the maximum amount of variation in the data. Subsequent principal components capture progressively smaller amounts of variation.
3. The eigenvalues measure the amount of information or variation in the data captured by each principal component.
4. The sum of all eigenvalues is equal to the total variance of the data. Therefore, the proportion of variance a principal component explains can be calculated as the ratio of its eigenvalue to the sum of all eigenvalues - "explained variance ratio."

The covariance matrix

1. A **symmetric square matrix** that summarizes the relationships between variables in a dataset.
2. Covariance matrix is **positive semi-definite** - all of its eigenvalues are non-negative.
3. The covariance matrix determine the degree of correlation between the variables in the data set. If the covariance between two variables is positive, the variables tend to move in the same direction. On the other hand, if the covariance is negative, the variables tend to move in opposite directions. If the covariance is zero - no correlation.
4. In the context of PCA, it is a measure of the variability in the data. The covariance matrix provides information about **how much the variables in the data set vary from the mean and how they are related?**

Eigenvectors and eigenvalues and their properties

1. An eigenvector is a non-zero vector that, when multiplied by a matrix, results in a scalar multiple of the original vector. This scalar multiple is called the eigenvalue.
2. **Orthogonality:** Eigenvectors corresponding to distinct eigenvalues are orthogonal to each other, meaning they are perpendicular in n-dimensional space. **This property is useful for separating different sources of variation in a dataset.**
3. **Normalization:** Eigenvectors are typically normalized to have unit length. **This ensures that the magnitude of the eigenvectors does not affect their contribution to the principal components.**

Principal Component

1. This components are new variables that are linear combinations of the original variables.
2. The first principal component is a linear combination of the original variables that captures the largest variation in the data. The second principal component is a linear combination of the original variables that captures the largest amount of variation that is orthogonal (uncorrelated) to the first principal component, and so on.
3. By using eigenvectors and eigenvalues of the covariance matrix, principal components are derived. The eigenvectors represent the directions of maximum variance in the data, and the eigenvalues indicate the amount of variance explained by each eigenvector.

How eigenvectors and eigenvalues are used to obtain principal components?

1. **The first principal component is a linear combination of the original variables that captures the largest amount of variation in the data**, and subsequent principal components capture orthogonal variation in decreasing order of importance.
2. Compute the covariance matrix of the dataset → Calculate the eigenvectors and eigenvalues of the covariance matrix → Sort the eigenvectors by their corresponding eigenvalues in descending order → Select the top k eigenvectors with the highest eigenvalues to form the principal components → Project the original data onto the k principal components to obtain the lower-dimensional representation.

Pseduo Code for PCA

1. Input:
   - X: matrix of size (n, m), where n is the number of samples and m is the number of features
   - k: number of principal components to keep
2. 1. Standardize the data - Calculate the mean of each column of X, subtract it from the corresponding column, and divide by the standard deviation of that column. This produces a standardized version of X with mean 0 and variance 1 for each feature.
3. 2. Compute the covariance matrix - Calculate the covariance matrix of the standardized data using the formula $cov(X) = \frac{XX^T}{n-1}$
4. 3. Compute the eigenvectors and eigenvalues of the covariance matrix: Use an eigendecomposition method, such as the power iteration method or the QR algorithm

Pseduo Code for PCA

1. 4. Sort the eigenvectors in decreasing order of eigenvalue: Select the k eigenvectors with the largest eigenvalues, and store them in a matrix V.
2. 5. Project the data onto the new k-dimensional space: Compute the matrix product X' = X * V, where X' is the new matrix of size (n, k) representing the data projected onto the k-dimensional space defined by the eigenvectors.
3. output:
   - X': matrix of size (n, k) representing the data projected onto the k-dimensional space defined by the eigenvectors.

Pseduo Code for PCA using SVD

1. Input:
   X: m x n matrix, where m is the number of features and n is the number of samples in the dataset
2. 1)Standardize: Substract the mean from each feature of X
3. 2)Calculate the singular value decomposition of X: X = USV'
4. 3)Calculate the principal components by selecting the top k columns of V' (where k is the desired dimensionality reduction)
5. 4)Project the original data onto the principal components to obtain the reduced dataset: Y = V(:,1:k)' * X

Applications

1. **Few of the applications of PCA are:**
2. In genetics, PCA can be used to identify the most important genes and reduce the dimensions of the data, making it easier to analyze and understand the relationships between different genes.
3. In finance, PCA can be used to identify the most important trends and patterns in stock prices and reduce the dimensions of the data, making it easier to analyze and predict future trends.
4. In computer vision, PCA can be used to identify an image's most important features and reduce the data's dimensions, making it easier to analyze and process.

Limitations of PCA

1. PCA assumes that the relationship between variables is linear. This means that PCA may not be appropriate for datasets that contain non-linear relationships between variables, such as those that exhibit complex or nonlinear patterns.
2. Assumes that the principal components are orthogonal, meaning they are uncorrelated. This assumption may not hold in some cases, such as when dependencies or interactions between variables violate the orthogonality assumption.
3. PCA can also be sensitive to outliers in the data. Outliers can skew the results of PCA, as they can disproportionately influence the principal components.

Other techniques?

1. Other dimensionality reduction techniques include Independent Component Analysis (ICA), which aims to identify underlying independent sources of variation in the data, and Kernel PCA, which extends PCA to non-linearly separable data by projecting the data onto a higher-dimensional space. These techniques can be useful when dealing with complex or non-linear data sets.

Concept

1. Independent Component Analysis (ICA) is a dimensionality reduction technique that is motivated by the desire to separate a multivariate signal into its independent components or sources. In many real-world applications, such as speech processing or image analysis, signals are often mixed together, making it difficult to separate individual sources.
2. ICA aims to overcome this challenge by identifying the underlying sources and their corresponding mixing coefficients, even if the signals are mixed in an unknown and nonlinear way.
3. ICA has various applications in signal processing, such as speech separation, image processing, and data compression. By separating the independent sources, ICA can reveal hidden information and improve the quality of the processed signals

How ICA works?

1. ICA is based on the following assumptions
   The main assumptions of ICA are statistical independence and non-Gaussianity of the sources. ICA uses a maximum likelihood approach to estimate the independent components.
2. Let X be a random vector of observed signals, and A be an unknown matrix representing the mixing process that transforms the original sources S into the observed signals X, such that X = AS. The goal of ICA is to estimate the matrix A and the independent sources S, given only the observed signals X.
3. The first assumption of ICA is that the sources S are statistically independent. This means that the joint probability distribution of the sources can be factorized as P(S) $= \pi P(S_i)$, where Si is the i-th component of the source vector S. In other words, the sources are not correlated with each other and contain different information.

How ICA works?

1. The second assumption of ICA is that the sources S are non-Gaussian. This is because Gaussian sources can be linearly mixed to form a Gaussian distribution, making it impossible to recover the original sources. Non-Gaussian sources, on the other hand, retain their independence even when mixed, making it possible to separate them.
2. The maximum likelihood estimation approach used in ICA aims to estimate the independent components by maximizing the likelihood function of the observed signals X. The likelihood function is defined as the joint probability density function of the observed signals, given the mixing matrix A and the independent sources S, i.e., L(A,S—X) = P(X—A,S). Using the assumptions of statistical independence and non-Gaussianity, the likelihood function can be maximized using methods such as gradient ascent or fixed-point iteration.

Pseduo Code

1. Input:
   - X: Observed signals (matrix of size n X m)
   - n-components: Number of independent components to estimate
2. 1.Center the observed signals by subtracting the mean of each column of X from the corresponding column.
3. 2.Whiten the observed signals by applying a whitening transform to X-centered. Whitening the data transforms the data into a new space where the covariance matrix is diagonal, and each feature's variance equals one. This makes the estimation of the independent components easier.
4. 3.Initializing the unmixing matrix randomly (starts the estimation process).

Pseduo Code

1. 4.The main estimation loop updates the unmixing matrix A by performing gradient ascent on the likelihood function. The nonlinearity function is applied to the estimated sources to make the problem non-Gaussian. The step size controls the learning rate of the algorithm, and the orthonormalization step ensures that the unmixing matrix is orthonormal.
2. 5.The estimated independent sources are computed by applying the inverse transformation of whitening and centering to the unmixing matrix.
3. 6.The estimated independent sources are returned as output.
4. Output:
   S: Estimated independent sources (matrix of size n x n-components)

Applications

1. Speech Separation: ICA can be used to separate different speakers' voices from a single audio recording. The assumption of statistical independence allows ICA to separate the sources by identifying the independent components in the mixed signal. The non-Gaussianity assumption helps ICA to identify the non-Gaussian speech components, which can be used to separate the sources.
2. EEG Analysis: EEG signals are generated by the electrical activity of the brain, and ICA can be used to separate the sources of these signals. The statistical independence assumption allows ICA to identify the independent components of the EEG signals, such as alpha, beta, and gamma waves. The non-Gaussianity assumption helps to identify the non-Gaussian components, which may represent artifact or noise components.

Limitations of ICA

1. **Non-Gaussianity assumption:** ICA assumes that the independent components are non-Gaussian, which may not hold in all cases. Some independent components may be Gaussian, making separating the sources difficult or impossible.
2. ICA assumes that the observed signals are linear mixtures of the independent components. If the mixing process is nonlinear, ICA may not be able to separate the sources. Moreover, ICA assumes that the mixing process is instantaneous and does not take into account any delays or temporal dependencies between the sources.
3. ICA is sensitive to the initial conditions of the estimation process. Depending on the initial unmixing matrix, the algorithm may converge to different solutions, which can lead to different estimates of the independent components. So different initializations have to be tried and compare the results to ensure the robustness of the estimates.

Limitations

1. ICA requires the specification of the number of independent components to estimate, which can be difficult to determine in practice. If the number of estimated components is too high, some components may be redundant or contain noise. Some important sources may be missed if the number of estimated components is too low. So, **the choice of the number of independent components** may require trial and error or prior knowledge about the problem.
2. ICA can be **computationally expensive**, especially for large datasets or high-dimensional signals. The whitening and unmixing steps require matrix decompositions and multiplications, which can be time-consuming. Moreover, the optimization process may require a large number of iterations to converge, which can increase the computational cost.

What and why?

1. The motivation behind generalized PCA is to address the limitations of traditional PCA methods, which assume that the data follows a Gaussian distribution and that the relationships between variables are linear.
2. Generalized PCA aims to extend PCA to handle non-Gaussian distributions and non-linear relationships between variables. This can improve the accuracy and usefulness of PCA in various applications, such as image and signal processing, where non-linear and non-Gaussian relationships are common. By incorporating non-linear relationships and non-Gaussian distributions, generalized PCA can capture more complex and meaningful patterns in the data.

How Generalized PCA works?

1. One approach is to use the kernel trick, which allows us to perform non-linear transformations of the data without explicitly computing them. Specifically, the kernel trick involves mapping the original data points into a higher-dimensional feature space using a non-linear function and then computing the principal components in this feature space.
2. Kernel PCA is a specific instance of the kernel trick, where the principal components are computed using the kernel matrix instead of the original data matrix. The kernel matrix is computed by applying the kernel function to all pairs of data points and can be efficiently computed using matrix operations.
3. Another approach is to use other non-linear techniques, such as manifold learning methods, which aim to discover the underlying low-dimensional structure of the data by preserving the local geometry of the data.

Pseduo Code

The generalized PCA method is implemented through the following steps:

1. Solve the quasi covariance matrix.
   $R' = \Sigma B(x_i - \bar{x}_i)(x_i - \bar{x}_i)^\top B^\top$
   $B = \sqrt{\frac{1}{2}D_j}$, where D_j is the weight coefficient in the jousselme evidence distance.
2. Find the eigenvalues of the quasi covariance matrix and arrange them in descending order.
3. Normalize the eigenvectors.
4. Select the first few feature vectors for dimensionality reduction.
5. Calculate the dimensionality reduction results.

Pseduo Code for Kernel PCA I

Input: Data matrix X, kernel function K,
number of principal components k

1. Compute the kernel matrix K-prime, which is a matrix that captures the similarity between pairs of data points in X using a kernel function K.
2. Center the kernel matrix K-prime by subtracting the mean of each row and column. This is done to ensure that the principal components are computed with respect to the covariance matrix of the data.
3. Compute the eigendecomposition of the centered kernel matrix K-centered to obtain the eigenvalues and eigenvectors of the covariance matrix of the data.
4. Select the top k eigenvectors corresponding to the largest eigenvalues, and normalize them to have unit length. These eigenvectors define the k-dimensional subspace that captures the most variance in the data.

Pseduo Code for Kernel PCA II

5. Project the original data matrix X onto the k-dimensional subspace by computing the dot product between each data point and the k eigenvectors. This results in a k-dimensional representation of the data that captures the most important information.
6. The output of the algorithm is the k-dimensional representation of the data (Z), the eigenvalues (eigvals), and the eigenvectors (eigvecs) of the covariance matrix.

Applications

1. **Image Processing:** Generalized PCA can be used for image compression and feature extraction. For instance, in face recognition, a kernel PCA can be applied to the face images to extract the principal components that capture the variations in the face images. These principal components can then represent the faces in a lower dimensional space, reducing the computational complexity of face recognition algorithms.
2. **Bioinformatics:** Generalized PCA can be used to analyze gene expression data and identify the genes most correlated with a particular disease or condition. For example, kernel PCA can be used in cancer research to identify the genes that are differentially expressed between cancerous and healthy tissues. These genes can then be used to develop diagnostic tests or therapeutic targets.

Limitations of Generalized PCA I

1. The **choice of kernel function** greatly affects the results of generalized PCA. Different kernels have different properties and may be more appropriate for certain types of data. Therefore, choosing the right kernel is crucial for obtaining accurate and meaningful results. However, there is no universally optimal kernel for all types of data, and the choice of kernel often requires prior knowledge or experimentation.
2. **Difficulty interpreting non-linear relationships:** Generalized PCA can capture non-linear relationships between variables, a major advantage over traditional PCA. However, non-linear relationships can be difficult to interpret and may not have a simple intuitive explanation. This can make it challenging to gain insights and draw meaningful conclusions from the results of generalized PCA.

Limitations of Generalized PCA II

3. Generalized PCA involves computing the kernel matrix, which can be **computationally expensive** for large datasets. In addition, computing the eigendecomposition of the kernel matrix can also be time-consuming.
4. **Overfitting:** Generalized PCA can overfit the data if the number of principal components is too high. This can result in the model capturing noise and other irrelevant information in the data, which can lead to poor performance on new data. Regularization techniques such as ridge regression can be used to address this issue.