4 Dimensionality reduction

Date	@October 20, 2024
Topic	Theory

Remarks

The variance is referred to a single attribute and tells how much spread x is in the data along the only axis. It is the power of two of the standard deviation.

$$s^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})^2$$

The covariance is referred to two attributes and it measures the correlation between x and y. If it is equal to 0 the attributes are independent, if it is higher than 0 they move in the same direction, otherwise they move in opposite directions.

$$cov(\mathbf{x},\mathbf{y}) = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})$$

A covariance matrix contains covariance values between all possible dimensions (n attributes).

$$S = \begin{bmatrix} cov(\mathbf{x},\mathbf{x}) & cov(\mathbf{x},\mathbf{y}) & cov(\mathbf{x},\mathbf{z}) \\ cov(\mathbf{y},\mathbf{x}) & cov(\mathbf{y},\mathbf{y}) & cov(\mathbf{y},\mathbf{z}) \\ cov(\mathbf{z},\mathbf{x}) & cov(\mathbf{z},\mathbf{y}) & cov(\mathbf{z},\mathbf{z}) \end{bmatrix}$$

$$S = \frac{1}{m-1}(X - \overline{X})^t(X - \overline{X})$$

Regarding orthogonality and orthonormality, two vectors u1 and u2 for which their product (transposing the second one) is equal to 0, are said to be

orthogonal. If the product is 1 they are said to be orthonormal. the inverse of an orthogonal matrix is its transpose.

Vectors u having same directions as Au are called eigenvectors of A (A is an n by n matrix). In the equation Au = λu, λ is called an eigenvalue of A. In order to calculate u and λ:

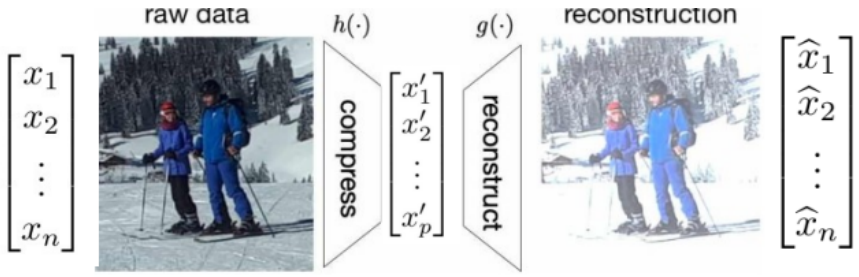
- calculate det(A-λI), yelds a polynomial
- determine roots to det(A-λI) = 0, and these are the eigenvalues λ
- solving (A-λI)u = 0 for each λ you obtain the eigenvectors u

Dimensionality reduction

The visualization problem is a crucial matter and it is not easy to extract useful information from multivariate data.

Dimensionality reduction means solving problems by changing the viewpoint and finding good features automatically. It creates a compressed representation of data and develops an hypothesis map that reads representation of data point and transforms it to a set of features.

The purpose is to find a map h which maximally compresses the raw data while still allowing to accurately reconstruct the original datapoint from a small number of features. From n raw features, we obtain p features, with p < n.

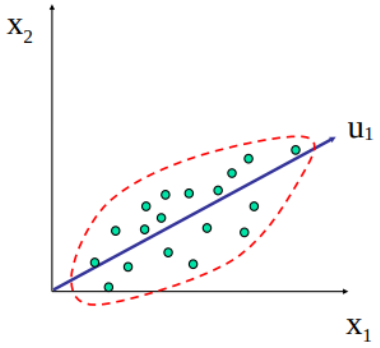


There are several techniques:

- singular value decomposition
- principal component analysis (PCA)
- linear discriminant analysis (LDA)
- non linear techniques (tSNE)

PCA

PCA can be use to reduce the number of dimensions in data, find a patterns in high-dimensional data and visualize data of high dimensionality. The idea is that we tipically have a data matrix X of m observations on n correlated variables, and we ant to find a new basis such that the first component maximizes information of data. For PCA it means to capture variance in the data, looking for a transformation of the xi into p (up to n) new variables x'i that are uncorrelated. This means finding a projection that captures the largest amount of variation in data



PCA can be viewed as a rotation of the existing axes to new positions in the space defined by original variables. New axes are orthogonal and represent the directions with maximum variability.

PCA is useful if there is some redundancy in variables or the data do not span the whole of n dimensional space. Redundancy means that some of the variables are correlated with another and because of it or space not covered, it is possible to reduce the observed variables into a smaller number of principal components. The components account for most of the variance in the observed variables.

Two definitions of PCA

Principal component analysis seeks a space of lower dimensionality, known as the principal subspace, such that the orthogonal projection of the data points onto this subspace maximizes the variance of the projected points. An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors, indicated by the blue lines.

Principal components

A principal component can be defined as a linear combination of optimally-weighted observed variables. The number of components that can be extracted in a principal component analysis is equal to the number of observed variables (n). Often only the first few components account for meaningful amount of variance. This means that the first component will be correlated with some of the observed variables. When the analysis is complete, te resulting components will display varying degrees of correlation with the observed variables, but are completely uncorrelated with one another.

Orthonormal change of basis P

The feature transformation of PCA is a change of the basis with orthonormal vectors. The aim is to find orthonormal P (n x n) such that $X' = XP^t$. With matrix S′ = cov(X′) diagonalized. The first p rows of P are the principal components of X.

In order to find P with cov(X′) diagonalized:

$$cov(X') = \frac{1}{m-1}(X')^t(X') \qquad \text{if X' has zero mean}$$

$$= \frac{1}{m-1}(XP^t)^t(XP^t)$$

$$= \frac{1}{m-1}PX^tXP^t$$

$$= \frac{1}{m-1}P(X^tX)P^t = \frac{1}{m-1}PAP^t$$

where $A = X^tX$ is symmetric (n x n).

Therefore there is a matrix E of eigenvectors of A and a diagonal matrix D such that $A = EDE^t$. Now we can define P to be the transpose of the matrix E of eigenvectors $P := E^t$ and we can write A as $A = P^tDP$.

The inverse of an orthogonal matrix is its transpose (due to its definition) $P^{-1} = P^t$.

$$
\begin{aligned}
cov(X') &= \frac{1}{m-1} PAP^t \\
&= \frac{1}{m-1} PP^t DPP^t \\
&= \frac{1}{m-1} PP^{-1} DPP^{-}1 = \frac{1}{m-1} D
\end{aligned}
$$

P now diagonalizes cov(X′) and it is the transpose of the matrix of eigenvectors of $A = XX^t$. The principal components of X are the eigenvectors of $A = X^t X$. The i^{th} diagonal value of cov(X′) is the variance of X′ along u_i (along the i^{th} principal component, the i^{th} row of P).

We need now to take the covariance matrix of the original matrix X and compute the eigenvalues and the eigenvectors, which are the new axis, thus the principal components.

The principal components 1 PC1 is the eigenvalue with the largest absolute value and it indicates that the data have the largest variance along its eigenvector, the direction along which there is the greatest variation.

The principal component 2 PC2 is the direction with maximum variation left in data, orthogonal to the PC1, i.e. the eigenvector corresponding to the second largest eigenvalue.

Steps of PCA

- assuming marked X as the mean matrix (all rows are equal, corresponding to the mean of all rows)
- adjust the original data by the mean X - Xmarcked
- compute the covariance matrix S of X
- find the eigenvectors and eigenvalues of S
- for matrix S, vectors u having the same direction as Su, with factor λ
- eigenvalues λ_i corresponds to variance on each component i=1,...,n
- sort by λ_i
- take the first p eigenvectors u_i where p is the number of top eigenvalues

- the project the original data $X' = XP^t$

Eigenvalues and variance

Eigenvalues λ_i are used for calculation of [% of total variance] (V_i) for each component i:

$$V_i = 100\frac{\lambda_i}{\sum_{j=1}^{n}\lambda_j} \qquad \sum_{i=1}^{n}\lambda_i = n \qquad \text{(if data standardized)}$$

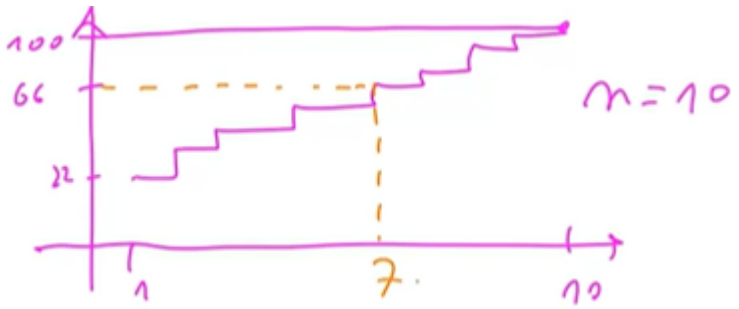
Properties of principal components

Principal components are a linear combination of the original variables and they are uncorrelated with each other. They also capture as much of the original variance as possible.

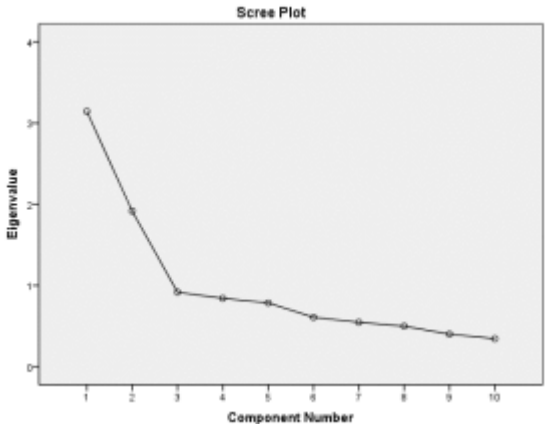
When to stop and choose P?

We can follow different methods:

- the Kaiser criterion: keep PCs with eigenvalues > 1, because if we standardize a dataset, we know that originally all variables had a variance of 1, so if we end up with some variable that has a eigenvalue < 1, it means that it is less important than any of the original feature.
- proportion of variance explained: enough PCs to have cumulative variance explained that is larger than a threshold (which is arbitrary)



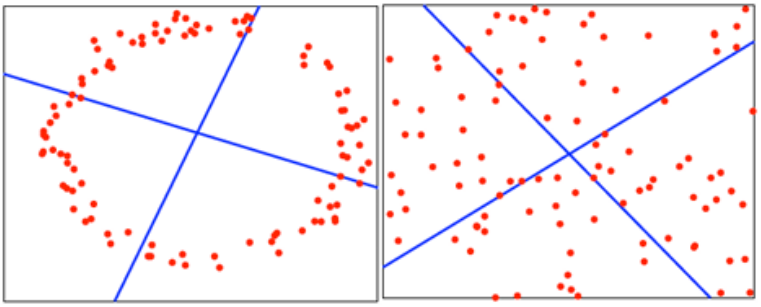
- Elbow method: start of the bend in the line (point of inflexion) indicate how many components are retained



Non-linear dimensionality reduction

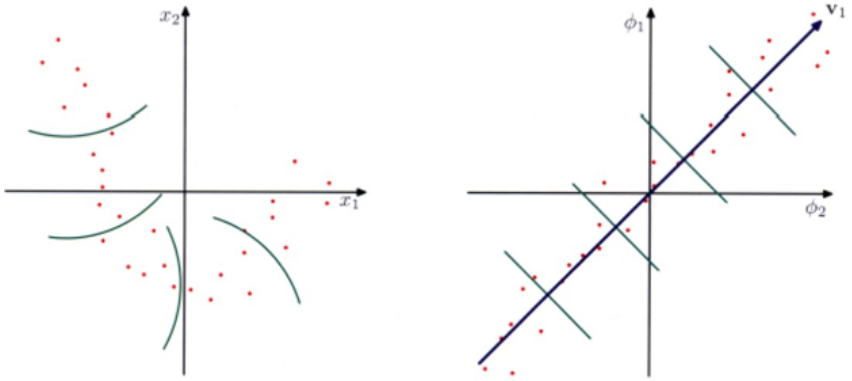
Extension of PCA - Kernel PCA

PCA will make no difference between these two examples because the variance is the same.



However, the two plots are very differente because there's a pattern on the left. To find that pattern we must use non-linear axes.

Instead of using directly points x as they are, we can go to some different feature space f(x), e.g. polar coordinates instead of cartesian coordinates. We can do PCA in this new space as well. The result will be non-linear in the original space.



Kernel PCA works on the projection f of the data in the new feature space. f is such that the kernel $K(x, y) = f(x)^t f(y)$ do not have negative eigenvalues. We can work directly with kernel matrix K instead of performing the projection.

t-SNE

t-distributed stochastic neighbor embedding is a nonlinear dimensionality reduction technique for embedding high-dimensional data for visualization. In this technique similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

It involves two iterative stage:

1. constructs a probability distribution over pairs of high dimensional objects in such a way that similar objects are assigned a higher probability while dissimilar points are assigned a lower probability
2. defines a similar probability distribution over the points in the lo-dimensional map.

Then it performs them iteratively until Kullback-Leibler divergence between the two distributions is minimized.

t-SNE reduces widespread data and expands densely packed data; it is often able to recover well-separated clusters, and with special parameter choices, approximates a simple form of spectral clustering. The visual cluster can be influenced strongly by the chosen parametrization. Some of the clusters might be false finding.

For a data set with m elements, t-SNE runs in $O(m^2)$ time and requires $O(m^2)$ space.

While t-SNE is concerned with preserving small pairwise distances whereas, PCA focuses on maintaining large pairwise distances to maximize variance. While PCA preserves the variance in the data, whereas t-SNE preserves the relationships between data points in a lower-dimensional space, making it quite a good algorithm for visualizing complex high-dimensional data.