UNIVERSIDAD DE ANTIOQUIA

1803

Dimensionality Reduction (PCA, Multidimensional Scaling)

Daniel Escobar
Catalina Bustamante

Content

- PCA
- Multidimensional Scaling
- Application

Dimensionality Reduction

- Feature selection (original features)
- Feature extraction (New feature space) : Data compression maintaining most of the relevant information.

- When faced with situations involving high-dimensional data, it is natural to consider the possibility of projecting those data onto a lower-dimensional subspace without losing important information regarding some characteristic of the original variables.

INTRODUCTION

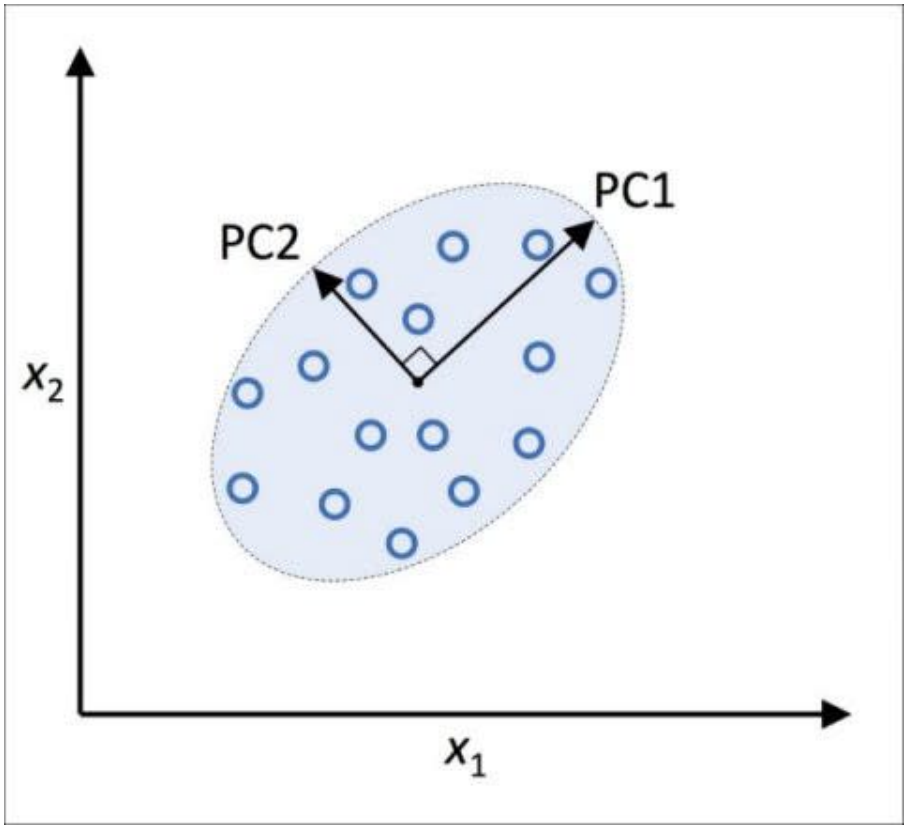
- Improve
  - Storage space
  - Computational efficiency
  - Predictive performance by reducing the curse of dimensionality

PCA

- Unsupervised linear transformation
- Reducing dimensionality
  - Lossy data compression
  - pattern recognition
  - Image analysis

PCA

- Discover important features of the data
  - Exploratory data analyses and de-noising of signals in stock market trading
  - Analysis of genome data and gene expression levels
  - Identify patterns in data based on the correlation between features
  - The first few principal component scores can reveal whether most of the data actually live on a linear subspace of $\mathbb{R}^k$ and can be used to identify outliers, distributional peculiarities, and clusters of points.
- Aims to find the direction of maximum variance in high dimensional data projecting onto a new subspace with equal or fewer dimensions
- Ortogonal axes (Principal components) of new subspace are the direction of maximum variance. New features are orthogonal to each others (uncorrelated)



$$x = [x_1, x_2, x_3, \ldots . x_d], \quad x \in \mathbb{R}^{d}$$

$$W, \quad W \in \mathbb{R}^{dxk} \qquad k << d$$

$$z = [z_1, z_2, \ldots . . , z_k], \quad z \in \mathbb{R}^{k}$$

Steps

1. Standarize the d-dimensional dataset
2. Construct the covariance matrix
3. Decompose the covariance matrix in eigenvectors and eigenvalues
4. Sort the k largest eigenvalues
5. Select the k largest eigenvalue,
6. Construct a projection matrix **W** from the top k eigenvectors
7. Transform the d-dimensional input dataset **X** using **W**

1. Standarize the d-dimensional dataset

- PCA direction are highly sensitive to data scaling, standardize there features prior to PCA to assign equal importances to all features

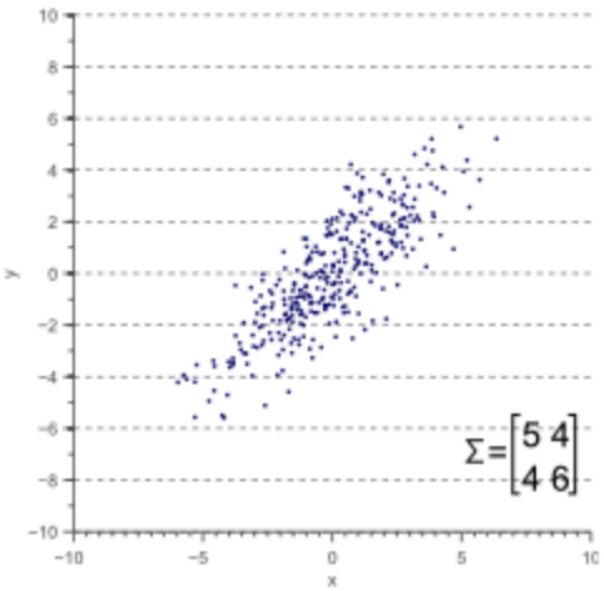
- $Z = (x - u)/\sigma$

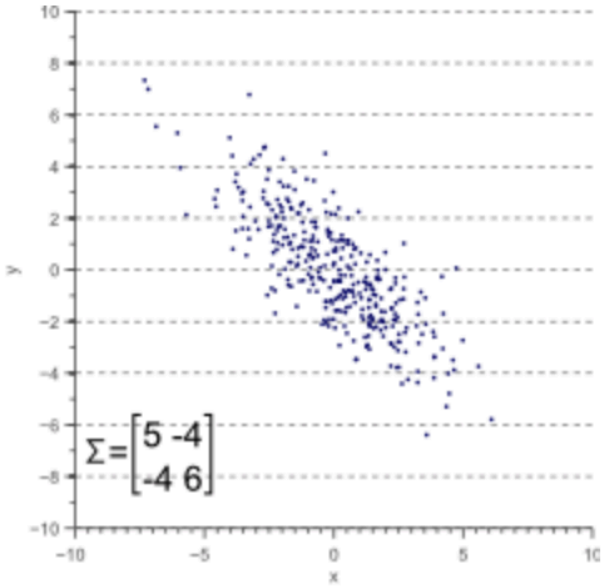
- 0 mean and unit variance

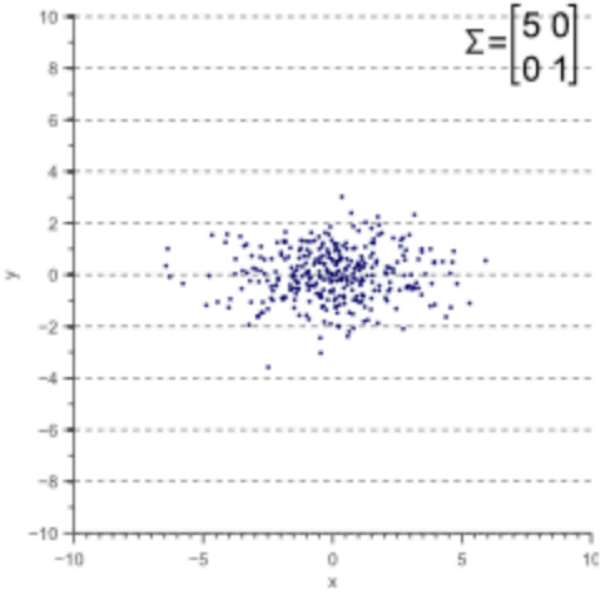
2. Construct the covariance matrix

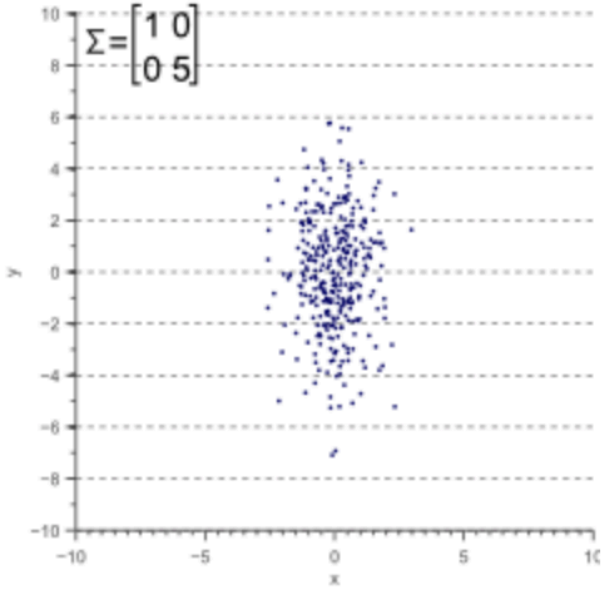
- $\sigma_{jk} = \frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$
- Positive covariance indicates than the feature increase or decrease together

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$









https://pathmind.com/wiki/eigenvector

3. Decompose the covariance matrix in eigenvectors and eigenvalues

$$\Sigma\nu = \lambda\nu \qquad \lambda = eigenvalue$$
$$v = eigenvalue$$

1. $P(\lambda) = det(A - \lambda I)$
2. $\lambda_1, \lambda_2, \ldots \lambda_d \qquad P(\lambda) = 0$
3. $(A - \lambda_i I)v = 0$

Example with a 2x2 Matrix

$$A = \begin{bmatrix} -5 & 2 \\ 2 & -2 \end{bmatrix}$$

$$det(A - \lambda I) = \begin{vmatrix} -5-\lambda & 2 \\ 2 & -2-\lambda \end{vmatrix}$$

$$(-5-\lambda)(-2-\lambda) - 4 = \lambda^2 + 7\lambda + 6 = 0$$

$$\lambda 1 = -1 \quad and \quad \lambda 2 = -6$$

$$\lambda 1 : (A - \lambda 1 I)v = 0 \Rightarrow \quad \left\{ \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \qquad v1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$$

$$\lambda 2 : (A - \lambda 2 I)v = 0 \Rightarrow \quad \left\{ \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \qquad v1 = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \right\}$$

4. Sort the k largest eigenvalues

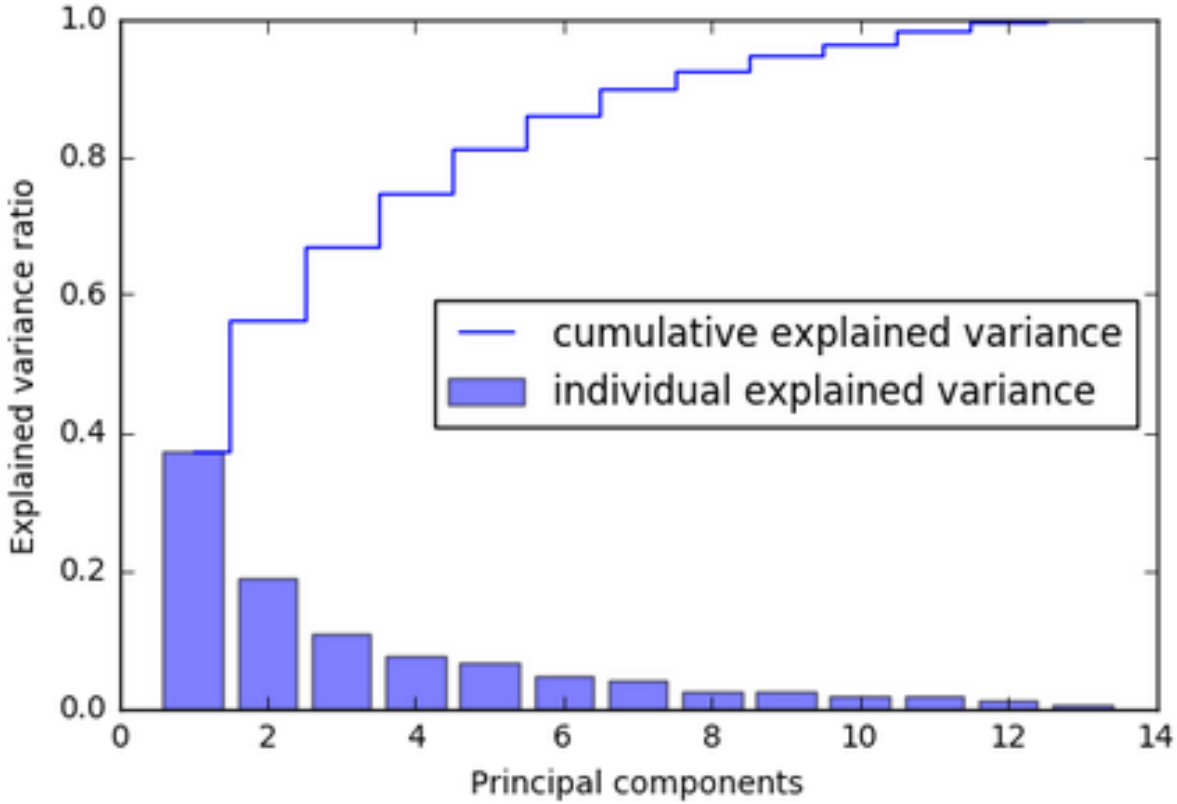
- $$\frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j}$$
- The eigenvector with the highest eigenvalue is therefore the principal component

5. Select the k largest eigenvalue

k is the dimensionality of new feature subspace $k \leq d$

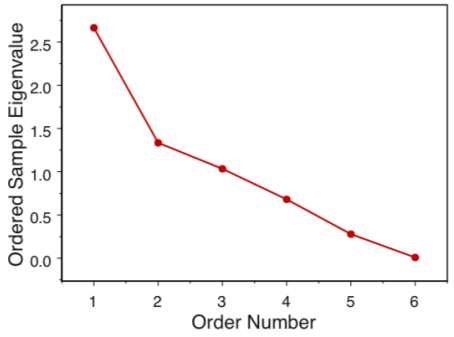
Trade-off between computational efficiency and the performance of the classifier

Total an explained variance

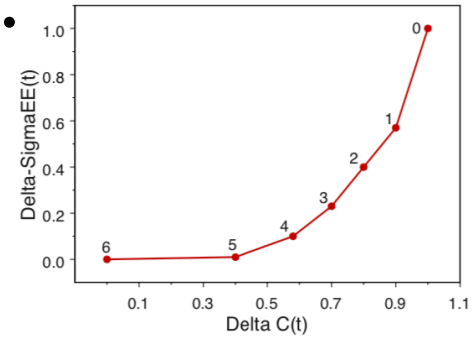


How Many Principal Components to Retain?

- Scree Plot:



- PC Rank Trace:



- Kaiser's Rule: eigenvalues > 0.7

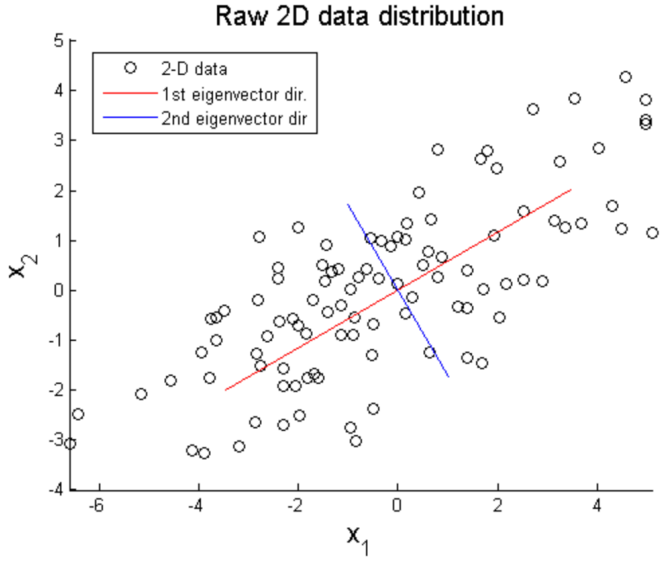
6. Construct a projection matrix **W** from the top k eigenvectors

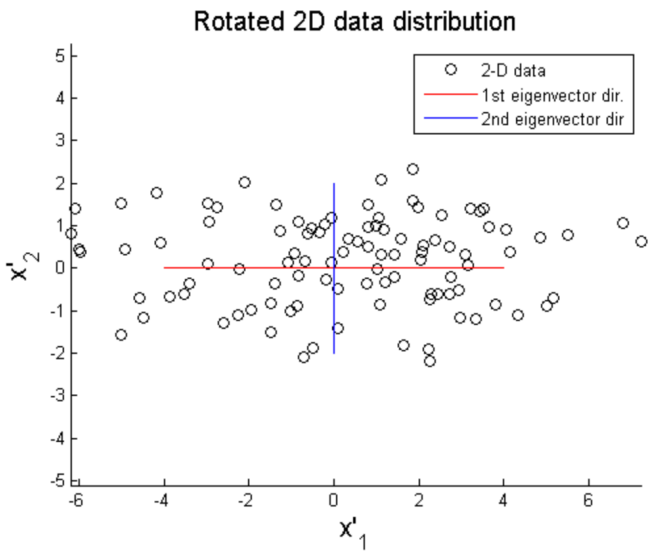
7. Transform the d-dimensional input dataset **X** using **W**

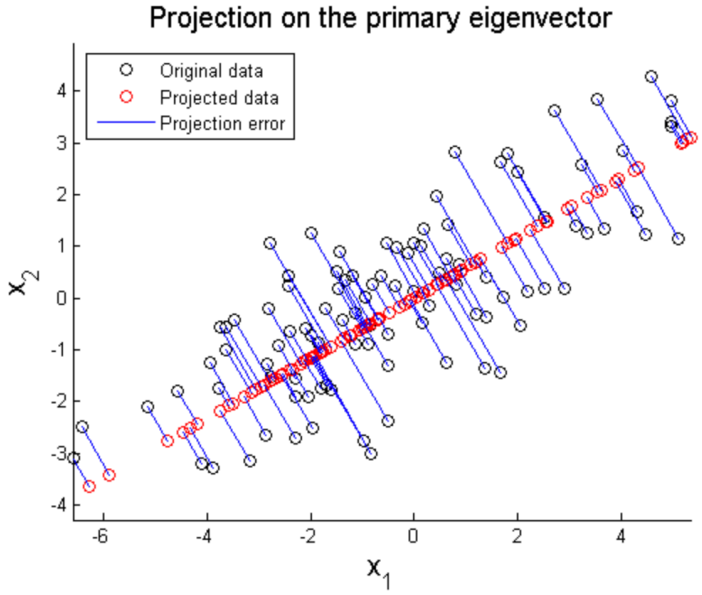
To obtain the new k-dimensional feature subspace

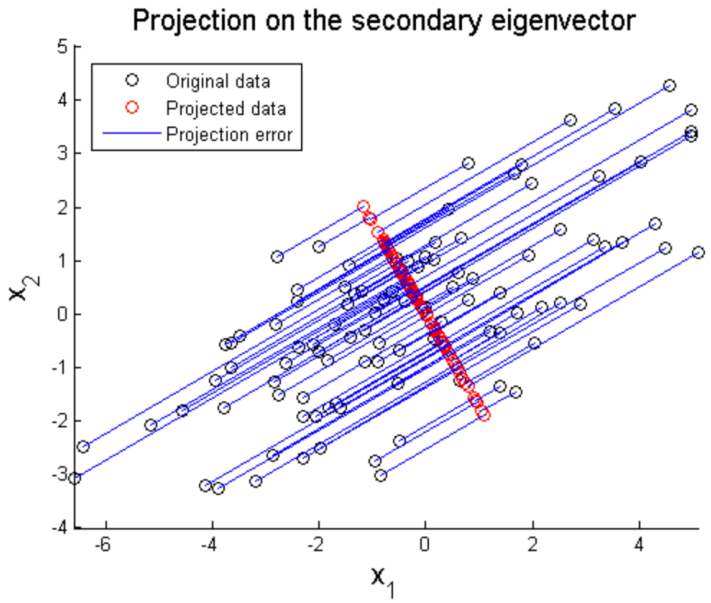
$$x' = xW \qquad X' = XW$$

$$x' = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p$$









https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

Adaptations of principal component analysis

- Functional PCA
- Simplified principal components:
  - Rotation
  - Adding a constraint
- Robust principal component analysis
- Symbolic data principal component analysis

Multidimensional Scaling

Designed to project high dimensional data down to small dimensions while preserving relative distances between observations.

It can be used to look at higher dimensional data and try to find patterns or grouping

MDS attempts to preserve pairwise distances.

MDS

- Proximities do not have to be distances, can be a more complicated concept.
- There should be a monotonic relationship between the closeness of two entities and the corresponding similarity value.

STEPS

1. Calculate $\mathscr{D}_{dxd}$ distance or affinity matrix
   $\mathscr{D}_{dxd} = [(xi - x_j)^T (x_i - x_j)]^{1/2}$
2. Calculate eigenvalues and eigenvectors
   $A : diag(\lambda_1, \lambda_2 \ldots . \lambda_d) \qquad V : v_1, v_2 \ldots . . v_p$
3. Construct the matrix
   $B : VAV^T = ZZ^T \qquad Z = VA^{1/2}$
4. $d_{ij} = [(Z_i - Z_j)^T (Z_i - Z_j)]^{1/2} \qquad d_{ij} : \delta_{ij} \quad from \quad \mathscr{D}$
5. The first k eigenvectors are used to get n points with
   $d_{ij} \approx \delta_{ij}$

Multidimensional scaling models

- Metric:
  - Classical scaling: Euclidean distances $d_{ij} = \delta_{ij}$
  - Metric least squares scaling: $d_{ij} = f(\delta_{ij})$ f is a continuous monotonic function
- Non metric
  - The transformation f can now be arbitrary, but must obey the monotonicity constraint

REFERENCES

- Izenman (2008). Modern multivariate statistical techniques

- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.
- http://www.cs.haifa.ac.il/~rita/uml_course/lectures/PCA_MDS.pdf

- Raschka, S. , Mirjalili V (2017) Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2nd Edition
- Cox, T. F., & Cox, M. A. (2000). Multidimensional scaling. Chapman and hall/CRC.
- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.
- Izenman (2008). Modern multivariate statistical techniques