Big Data Analysis
(MA60306)

Bibhas Adhikari

Spring 2022-23, IIT Kharagpur

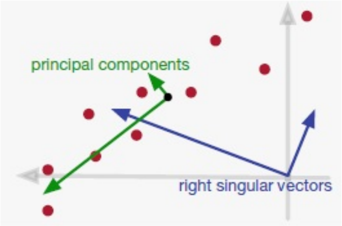
Lecture 11
February 2, 2023

Principal component analysis

Observation the best fitting subspace is a subspace!! i.e. it is a plane which passes through origin

Principal component analysis

Observation the best fitting subspace is a subspace!! i.e. it is a plane which passes through origin



Principal component analysis(PCA) - an extension of SVD when the desired subspace V does not pass through origin but it goes through the mean of all the data points! So use SVD after a prepossessing step, called centering to shift the data matrix to its mean at the origin!

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n \times d}$ such that each column has mean value 0.

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n \times d}$ such that each column has mean value 0.

$\rightarrow$ Define $\overline{\boldsymbol{a}}_j = \frac{1}{n}\sum_{i=1}^{n} A_{ij}$. the average of each column

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n \times d}$ such that each column has mean value 0.

- $\rightarrow$ Define $\overline{\boldsymbol{a}}_j = \frac{1}{n} \sum_{i=1}^{n} A_{ij}$. the average of each column
- $\rightarrow$ Define $\widetilde{A}$ with $\widetilde{A}_{ij} = A_{ij} - \overline{\boldsymbol{a}}_j$, the ij-th entry of the centered matrix $\widetilde{A}$

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n\times d}$ such that each column has mean value 0.

- $\rightarrow$ Define $\overline{\boldsymbol{a}}_j = \frac{1}{n}\sum_{i=1}^{n} A_{ij}$. the average of each column
- $\rightarrow$ Define $\widetilde{A}$ with $\widetilde{A}_{ij} = A_{ij} - \overline{\boldsymbol{a}}_j$, the ij-th entry of the centered matrix $\widetilde{A}$
- $\rightarrow$ Another way: define the centering matrix $C_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is the all-one vector. Then

$$\widetilde{A} = C_n A$$

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n\times d}$ such that each column has mean value 0.

- $\rightarrow$ Define $\overline{\boldsymbol{a}}_j = \frac{1}{n}\sum_{i=1}^{n} A_{ij}$. the average of each column
- $\rightarrow$ Define $\widetilde{A}$ with $\widetilde{A}_{ij} = A_{ij} - \overline{\boldsymbol{a}}_j$, the ij-th entry of the centered matrix $\widetilde{A}$
- $\rightarrow$ Another way: define the centering matrix $C_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is the all-one vector. Then

$$\widetilde{A} = C_n A$$

- $\rightarrow$ The matrix C_n is a projection matrix!! Where does it project?

PCA

Centering - adjusting the given data matrix $A \in \mathbb{R}^{n \times d}$ such that each column has mean value 0.

- $\rightarrow$ Define $\bar{\boldsymbol{a}}_j = \frac{1}{n}\sum_{i=1}^{n} A_{ij}$. the average of each column
- $\rightarrow$ Define $\widetilde{A}$ with $\widetilde{A}_{ij} = A_{ij} - \bar{\boldsymbol{a}}_j$, the ij-th entry of the centered matrix $\widetilde{A}$
- $\rightarrow$ Another way: define the centering matrix $C_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is the all-one vector. Then
$$\widetilde{A} = C_n A$$
- $\rightarrow$ The matrix C_n is a projection matrix!! Where does it project?
- $\rightarrow$ Let SVD of $\widetilde{A} = C_n A = U\Sigma V^T$. Then the singular values of $\widetilde{A}$ are called the *principal values*, and the k singular vectors corresponding to the k largest singular values are called *top-k principal directions/vectors*

PCA

Let

$$A = \begin{bmatrix} 1 & 5 \\ 2 & 3 \\ 3 & 10 \end{bmatrix}.$$

Then the center vector is $\overline{a} = [2, \; 6]$
The centered matrix is

$$\widetilde{A} = \begin{bmatrix} -1 & -1 \\ 0 & -3 \\ 1 & 4 \end{bmatrix}$$

PCA

Another interpretation of PCA

→ We introduce a complete orthonormal set of d-dimensional vectors $\boldsymbol{v}_j, 1 \leq j \leq d$ that satisfy $\boldsymbol{v}_i^T \boldsymbol{v}_j = \delta_{i,j}$

PCA

Another interpretation of PCA

→ We introduce a complete orthonormal set of d-dimensional vectors $\boldsymbol{v}_j, 1 \leq j \leq d$ that satisfy $\boldsymbol{v}_i^T \boldsymbol{v}_j = \delta_{i,j}$

→ Then any data point $\boldsymbol{x}_i$ can be written as

$$\boldsymbol{x}_i = \sum_{j=1}^{d} \alpha_{ij} \boldsymbol{v}_j$$

i.e. this corresponds to a rotation of the coordinate system to a new system defined by the v_j, and the original d components $\{x_{i1}, \ldots, x_{id}\}$ are replaced by an equivalent set $\{\alpha_{i1}, \ldots, \alpha_{id}\}$

PCA

Another interpretation of PCA

→ We introduce a complete orthonormal set of d-dimensional vectors $\boldsymbol{v}_j, 1 \leq j \leq d$ that satisfy $\boldsymbol{v}_i^T \boldsymbol{v}_j = \delta_{i,j}$

→ Then any data point $\boldsymbol{x}_i$ can be written as

$$\boldsymbol{x}_i = \sum_{j=1}^{d} \alpha_{ij} \boldsymbol{v}_j$$

i.e. this corresponds to a rotation of the coordinate system to a new system defined by the v_j, and the original d components $\{x_{i1}, \ldots, x_{id}\}$ are replaced by an equivalent set $\{\alpha_{i1}, \ldots, \alpha_{id}\}$

→ Obviously, $\alpha_{ij} = \boldsymbol{x}_i^T \boldsymbol{v}_j$ and hence

$$\boldsymbol{x}_i = \sum_{j=1}^{d} (\boldsymbol{x}_i^T \boldsymbol{v}_j) \boldsymbol{v}_j$$

PCA

→ Goal: to approximate the data points using a representation involving a restricted number $k < d$ of variables corresponding to a projection onto a lower-dimensional subspace

PCA

- → Goal: to approximate the data points using a representation involving a restricted number $k < d$ of variables corresponding to a projection onto a lower-dimensional subspace
- → The k-dimensional subspace can be represented WLOG by the first k vectors, and so we approximate each data point $\boldsymbol{x}_i$ by

$$\widetilde{\boldsymbol{x}}_i = \sum_{j=1}^{k} z_{ij}\boldsymbol{v}_j + \sum_{j=k+1}^{d} \boldsymbol{b}_j\boldsymbol{v}_j$$

  where $\{z_{ij}\}$ depend on the particular data point, and $\{\boldsymbol{b}_j\}$ are constants that are the same for all data points
- → We are free to choose the $\{\boldsymbol{v}_j\}$, $\{z_{ij}\}$, and $\{\boldsymbol{b}_j\}$ so as to minimize the distortion introduced by the reduction in dimensionality

PCA

→ The distortion measure that we consider is the squared distance between the original data point $\boldsymbol{x}_i$, and its approximation $\widetilde{\boldsymbol{x}}_i$, averaged over the data set i.e. to minimize

$$J = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i\|^2$$

PCA

→ The distortion measure that we consider is the squared distance between the original data point $\boldsymbol{x}_i$, and its approximation $\widetilde{\boldsymbol{x}}_i$, averaged over the data set i.e. to minimize

$$J = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i\|^2$$

→ First, consider this minimization wrt $\{z_{ij}\}$:
Homework Substituting $\widetilde{\boldsymbol{x}}_i$, setting the derivative with respect to z_{ij} to zero, and making use of the orthonormality conditions, one can obtain

$$z_{ij} = \boldsymbol{x}_i^T \boldsymbol{v}_j, j = 1, \ldots, d$$

PCA

→ The distortion measure that we consider is the squared distance between the original data point $\boldsymbol{x}_i$, and its approximation $\widetilde{\boldsymbol{x}}_i$, averaged over the data set i.e. to minimize

$$J = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i\|^2$$

→ First, consider this minimization wrt $\{z_{ij}\}$:
Homework Substituting $\widetilde{\boldsymbol{x}}_i$, setting the derivative with respect to z_{ij} to zero, and making use of the orthonormality conditions, one can obtain

$$z_{ij} = \boldsymbol{x}_i^T \boldsymbol{v}_j, j = 1, \ldots, d$$

Homework Similarly, setting the derivative of J wrt $\boldsymbol{b}_j$ to zero gives

$$\boldsymbol{b}_j = \overline{\boldsymbol{x}}^T \boldsymbol{v}_j, j = k+1, \ldots, d$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$

PCA

→ Then we have $x_i - \widetilde{x}_i = \sum_{j=k+1}^{d} \left\{ (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \boldsymbol{v}_j \right\} \boldsymbol{v}_j$

→ Thus

$$J = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=k+1}^{d} \left(\boldsymbol{x}_i^T \boldsymbol{v}_j - \overline{\boldsymbol{x}}^T \boldsymbol{v}_j \right)^2 = \sum_{j=k+1}^{d} \boldsymbol{v}_j^T S \boldsymbol{v}_j$$

where S is the covariance matrix defined by

$$S = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$$

[1] Chapter 12, C. M.Bishop, Pattern Recognition and Machine Learning, Springer, 2009

PCA

→ Then we have $x_i - \widetilde{x}_i = \sum_{j=k+1}^{d} \left\{ (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \boldsymbol{v}_j \right\} \boldsymbol{v}_j$

→ Thus

$$J = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=k+1}^{d} \left(\boldsymbol{x}_i^T \boldsymbol{v}_j - \overline{\boldsymbol{x}}^T \boldsymbol{v}_j \right)^2 = \sum_{j=k+1}^{d} \boldsymbol{v}_j^T S \boldsymbol{v}_j$$

where S is the covariance matrix defined by

$$S = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$$

Homework Then show that the general solution to the minimization for J for arbitrary d and $k < d$ is obtained by choosing the $\{\boldsymbol{v}_j\}$ as the eigenvectors of the the covariance matrix S[1]

[1] Chapter 12, C. M.Bishop, Pattern Recognition and Machine Learning, Springer, 2009

Multidimensional scaling

Multidimensional scaling (MDS) - is a data analysis technique which translates distances and dissimilarities into a visual representation through a ‘geometric’ picture

Multidimensional scaling

Multidimensional scaling (MDS) - is a data analysis technique which translates distances and dissimilarities into a visual representation through a ‘geometric’ picture

- → The input for an MDS algorithm usually is not an object data set but similarities of a set of objects
- → Here we will use the term ‘distance’ in a generic sense, meaning it reflects a dissimilarity/similarity between pairs of the objects

Multidimensional scaling

Multidimensional scaling (MDS) - is a data analysis technique which translates distances and dissimilarities into a visual representation through a 'geometric' picture

→ The input for an MDS algorithm usually is not an object data set but similarities of a set of objects

→ Here we will use the term 'distance' in a generic sense, meaning it reflects a dissimilarity/similarity between pairs of the objects

→ Suppose there are n objects in a set and the similarities between all pairs are measured. Then we can define an $n \times n$ distance matrix $D = [d_{ij}]$, where d_{ij} represents the similarity/distance between the objects i and j

Multidimensional scaling

Multidimensional scaling (MDS) - is a data analysis technique which translates distances and dissimilarities into a visual representation through a 'geometric' picture

- → The input for an MDS algorithm usually is not an object data set but similarities of a set of objects
- → Here we will use the term 'distance' in a generic sense, meaning it reflects a dissimilarity/similarity between pairs of the objects
- → Suppose there are n objects in a set and the similarities between all pairs are measured. Then we can define an $n \times n$ distance matrix $D = [d_{ij}]$, where d_{ij} represents the similarity/distance between the objects i and j
- → The objects are configured as virtual points in a low dimensional linear Euclidean space, and this point set is called *configuration* such that the Euclidean distances between the points have closest relation to the similarities

MDS

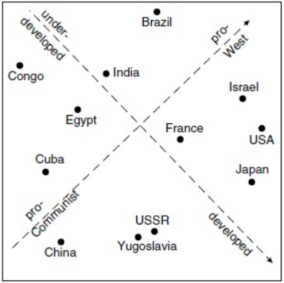
Example Similarity ratings for 12 nations (Wish, 1971)

Nation		1	2	3	4	5	6	7	8	9	10	11	12
Brazil	1	–											
Congo	2	4.83	–										
Cuba	3	5.28	4.56	–									
Egypt	4	3.44	5.00	5.17	–								
France	5	4.72	4.00	4.11	4.78	–							
India	6	4.50	4.83	4.00	5.83	3.44	–						
Israel	7	3.83	3.33	3.61	4.67	4.00	4.11	–					
Japan	8	3.50	3.39	2.94	3.83	4.22	4.50	4.83	–				
China	9	2.39	4.00	5.50	4.39	3.67	4.11	3.00	4.17	–			
USSR	10	3.06	3.39	5.44	4.39	5.06	4.50	4.17	4.61	5.72	–		
U.S.A.	11	5.39	2.39	3.17	3.33	5.94	4.28	5.94	6.06	2.56	5.00	–	
Yugoslavia	12	3.17	3.50	5.11	4.28	4.72	4.00	4.44	4.28	5.06	6.67	3.56	–

MDS

Example Similarity ratings for 12 nations (Wish, 1971)

Nation		1	2	3	4	5	6	7	8	9	10	11	12
Brazil	1	–											
Congo	2	4.83	–										
Cuba	3	5.28	4.56	–									
Egypt	4	3.44	5.00	5.17	–								
France	5	4.72	4.00	4.11	4.78	–							
India	6	4.50	4.83	4.00	5.83	3.44	–						
Israel	7	3.83	3.33	3.61	4.67	4.00	4.11	–					
Japan	8	3.50	3.39	2.94	3.83	4.22	4.50	4.83	–				
China	9	2.39	4.00	5.50	4.39	3.67	4.11	3.00	4.17	–			
USSR	10	3.06	3.39	5.44	4.39	5.06	4.50	4.17	4.61	5.72	–		
U.S.A.	11	5.39	2.39	3.17	3.33	5.94	4.28	5.94	6.06	2.56	5.00	–	
Yugoslavia	12	3.17	3.50	5.11	4.28	4.72	4.00	4.44	4.28	5.06	6.67	3.56	–



MDS

Example Distances between ten cities

	1	2	3	4	5	6	7	8	9	10
1	0	569	667	530	141	140	357	396	570	190
2	569	0	1212	1043	617	446	325	423	787	648
3	667	1212	0	201	596	768	923	882	714	714
4	530	1043	201	0	431	608	740	690	516	622
5	141	617	596	431	0	177	340	337	436	320
6	140	446	768	608	177	0	218	272	519	302
7	357	325	923	740	340	218	0	114	472	514
8	396	423	882	690	337	272	114	0	364	573
9	569	787	714	516	436	519	472	364	0	755
10	190	648	714	622	320	302	514	573	755	0

MDS

Example Distances between ten cities

	1	2	3	4	5	6	7	8	9	10
1	0	569	667	530	141	140	357	396	570	190
2	569	0	1212	1043	617	446	325	423	787	648
3	667	1212	0	201	596	768	923	882	714	714
4	530	1043	201	0	431	608	740	690	516	622
5	141	617	596	431	0	177	340	337	436	320
6	140	446	768	608	177	0	218	272	519	302
7	357	325	923	740	340	218	0	114	472	514
8	396	423	882	690	337	272	114	0	364	573
9	569	787	714	516	436	519	472	364	0	755
10	190	648	714	622	320	302	514	573	755	0