EOF analysis

연구 내용
고유값, 고유벡터
주성분 분석
시공간 데이터 실습 (Python)

HANYANG UNIVERSITY

Contents

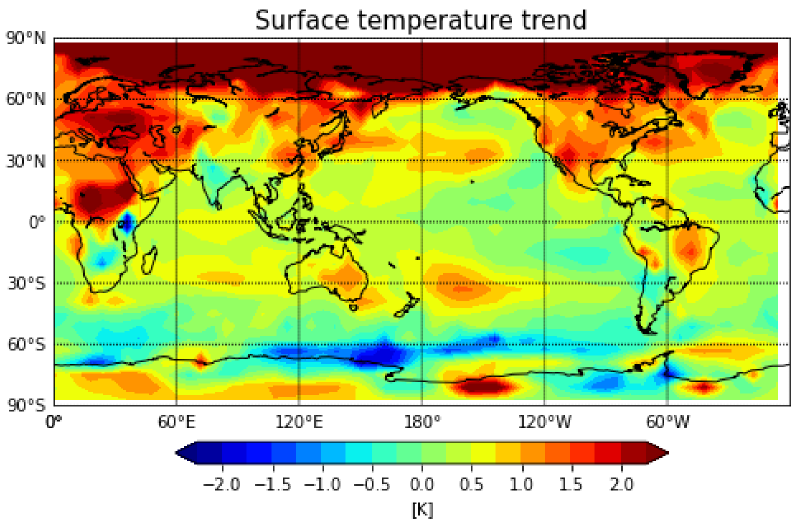
- 연구 내용
- 고유값, 고유벡터
- 주성분 분석
- 시공간 데이터 실습 (Python)

Contents

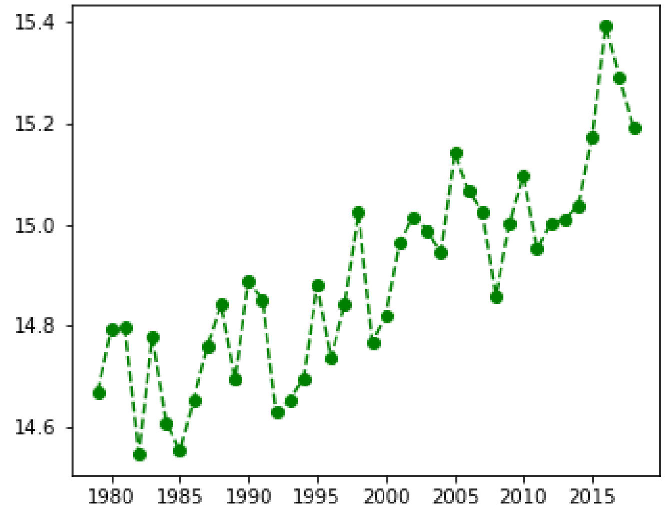
- **연구 내용**
- 고유값, 고유벡터
- 주성분 분석
- 시공간 데이터 실습 (Python)

연구 내용

- **지구 온난화 패턴 연구**



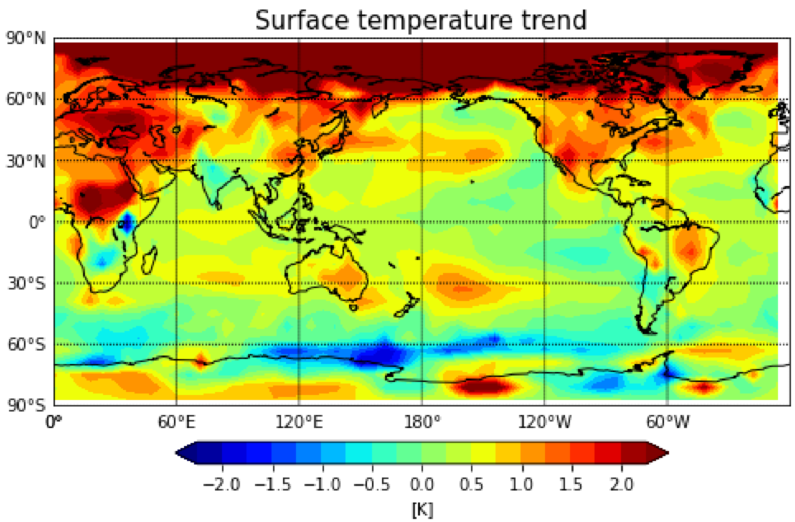
전 지구 평균 온도 변화 (1979~2018)



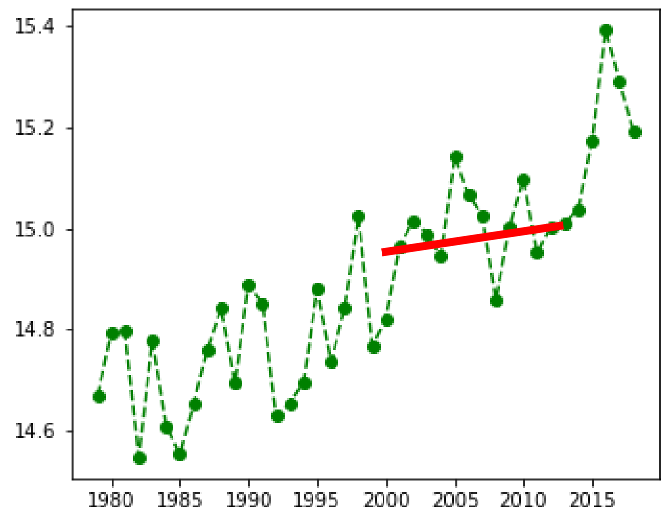
– 온도가 증가하는 지구온난화 경향을 뚜렷하게 보이는 한편, 자연변동성 (e.g. 엘니뇨/라니냐)의 영향이 섞여 있음.

연구 내용

- **라니냐 현상**



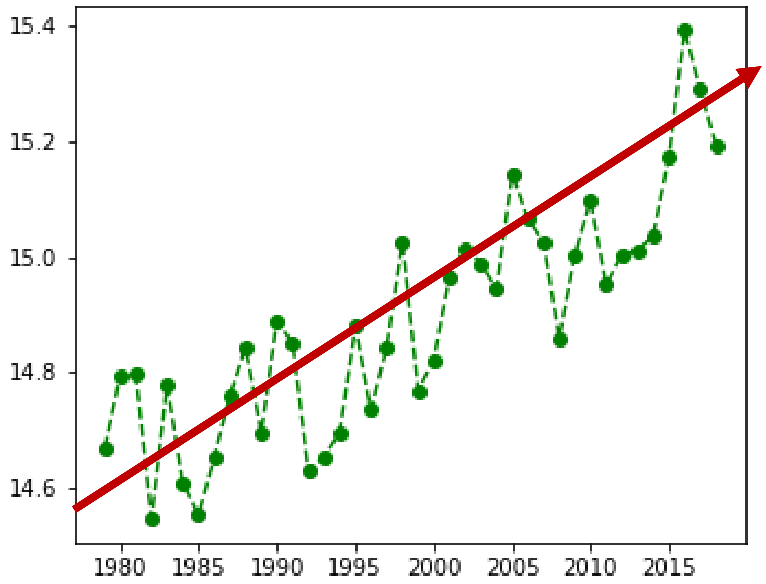
전 지구 평균 온도 변화 (1979~2018)



  - 90년대 후반부터 2012년까지 라니냐가 자주 일어나면서 온난화를 지연 시켰던 것으로 보임.
  - 이를 **"global warming hiatus"** 라고 부르기도 함.

연구 내용

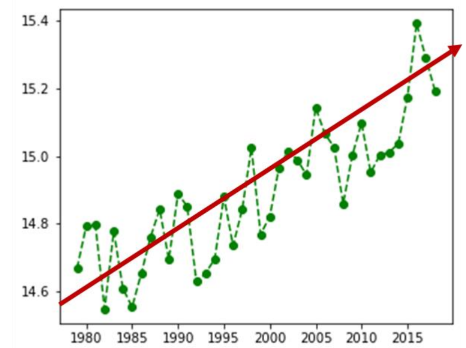
- **단순 선형 회귀모형**
  - 40개(**1979~2018, Year**)의 데이터를 제일 잘 설명하는 회귀직선을 찾기 위해 최소제곱법(Least Squared Method)을 이용해 회귀직선의 계수들을 추정함.



전 지구 평균 온도 $= \beta_0 + \boldsymbol{\beta_1} \times$ 연도 + 오차

연구 내용

- **주성분 분석 (Principal Component Analysis, PCA)**
  - 앞선 단순 선형 회귀모형에서 추정한 선형 회귀식이 '전지구 평균 온도변화'를 제일 잘 설명하는 주성분 분석(PCA)에서의 PC_1과 비슷한 개념임.



$\text{전 지구 평균 온도} = \beta_0 + \boldsymbol{\beta_1} \times \text{연도} + \text{오차}$

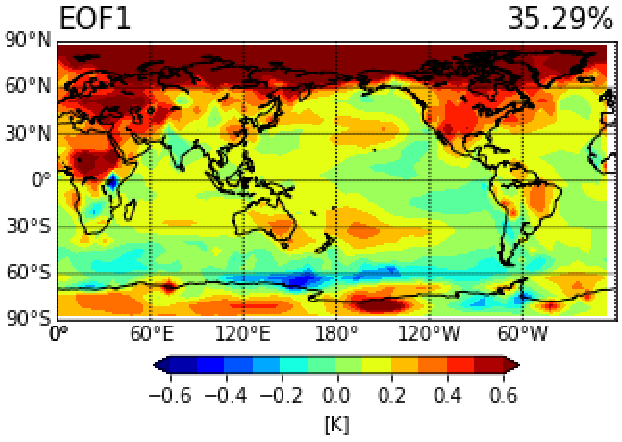
  - 하지만, 위의 선형모형은 전 지구 평균온도 변화에 대해 연도별 변화 추이를 살펴본 분석모형임.
    - 각 한 점들은 해당 연도의 전 지구 평균 온도임.

  - 시간적 특성과 공간적 특성, 이 두 가지 요인을 동시에 고려한 분석을 위해서는 주성분 분석을 수행해야 함.

데이터 구조

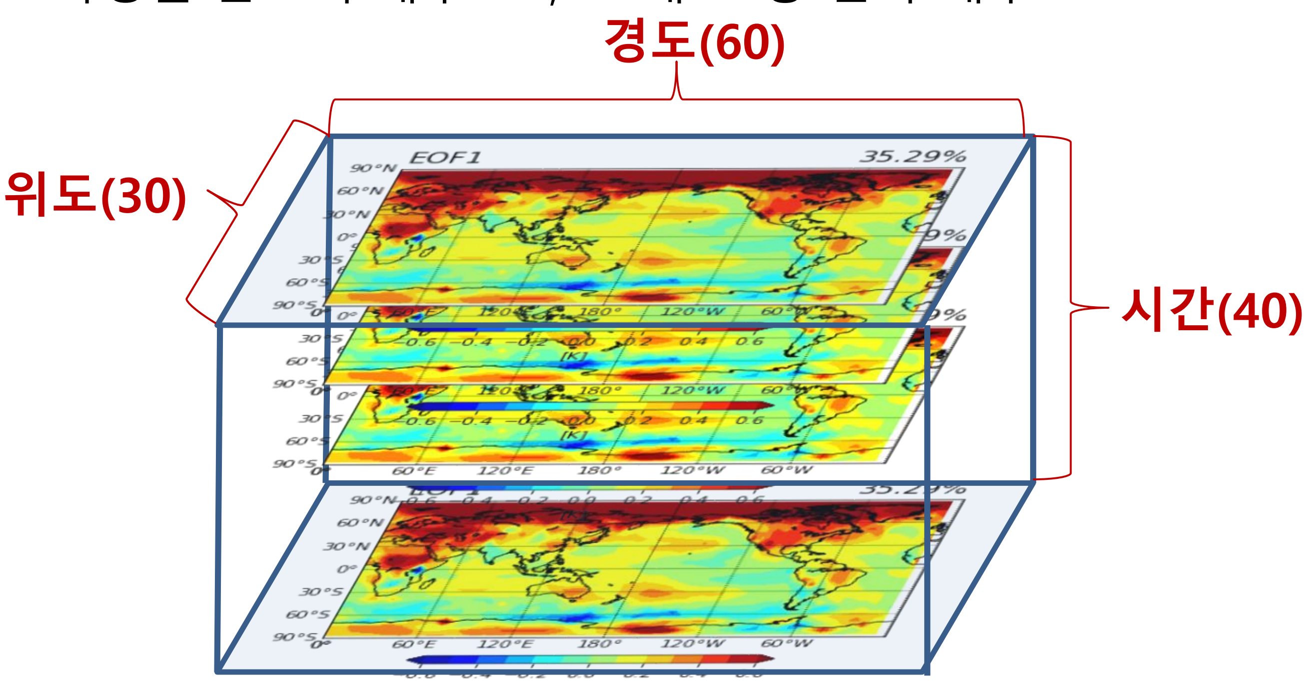
- **예제 데이터: T2m_ERA5_1979_2018_lowR.nc**
  - Python code

```
data = nc.Dataset("T2m_ERA5_1979_2018_lowR.nc",'r')
lon = data.variables['lon'][:] ## Longitude (경도)
Lat = data.variables['lat'][:] ## Latitude (위도)
Time = data.variables['time'][:] ## Time (연도)
T2 = data.variables['t2m'][:,:,:] ## T2m data (온도)
```



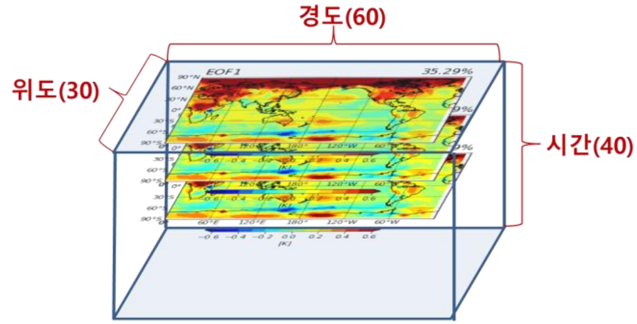
데이터 구조

- **예제 데이터: T2m_ERA5_1979_2018_lowR.nc**
  - 경도의 개수 : 60개 → 열의 개념
  - 위도의 개수 : 30개 → 행의 개념
  - 시간: 40개 (1979~2018, Year) → 층의 개념
  - 측정된 온도의 개수: 72,000개 → 총 관측 개수



데이터 구조

- **예제 데이터: T2m_ERA5_1979_2018_lowR.nc**
  - 데이터 행렬(matrix), 배열(array) 구조
    - Array = 1 (1979년)
      - ➢ $\begin{bmatrix} x_{1,1,1} & \cdots & x_{1,1,60} \\ \vdots & \ddots & \vdots \\ x_{1,30,1} & \cdots & x_{1,30,60} \end{bmatrix}$ 1,800 개 관측값
    - Array = 2 (1980년)
      - ➢ $\begin{bmatrix} x_{2,1,1} & \cdots & x_{2,1,60} \\ \vdots & \ddots & \vdots \\ x_{2,30,1} & \cdots & x_{2,30,60} \end{bmatrix}$
    - Array = 40 (2018년)
      - ➢ $\begin{bmatrix} x_{40,1,1} & \cdots & x_{40,1,60} \\ \vdots & \ddots & \vdots \\ x_{40,30,1} & \cdots & x_{40,30,60} \end{bmatrix}$



Contents

HANYANG UNIVERSITY

고유값, 고유벡터

- **고유값과 고유벡터**
  - 정방행렬 A에 대해 다음 식을 만족하는 영벡터가 아닌 벡터 v, 실수 λ를 찾을 수 있다고 가정하자.

$$Av = \lambda v$$

    - 위 식을 만족하는 **실수 λ를 고유값 (eigenvalue), 벡터 v를 고유벡터 (eigenvector)라고 함**.
      - 행렬 A의 고유벡터 (v)는 행렬 A를 곱해서 변환을 해도 방향이 바뀌지 않는 벡터임. ⇒ **방향은 바뀌지 않고 크기만 바뀌는 벡터 (고유벡터)**
      - 고유값 (λ)은 변환된 고유벡터와 원래 고유벡터의 크기 비율이다.
    - 고유값과 고유벡터를 찾는 작업을 **고유값분해 (eigenvalue decomposition)**이라고 함.

$$Av - \lambda v = (A - \lambda I)v = 0$$

고유값, 고유벡터

- **예제**
  - 행렬 A

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 2 & -3 \end{bmatrix}$$

에 대해 다음 스칼라값과 벡터는 각각 고유값과 고유벡터가 됨.

$$\lambda = -1, v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$Av = \begin{bmatrix} 1 & -2 \\ 2 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-2 \\ 2-3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} = (-1) \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \lambda v$$

고유값, 고유벡터

- **특성 방정식**
  - 행렬이 주어졌을 때, 고유값-고유벡터를 어떻게 구할 수 있을까?

$$Av - \lambda v = (A - \lambda I)v = 0$$

  - 행렬 A의 고유값은 $A - \lambda I$의 행렬식이 0이 되도록 하는 **특성방정식 (characteristic equation)**의 해를 구하면 됨.

$$det(A - \lambda I) = 0$$

    - 이 조건은 행렬 $A - \lambda I$가 역행렬이 존재하지 않는다는 의미임.
    - $det(A - \lambda I) = det\left(\begin{bmatrix} 1 & -2 \\ 2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = det\left(\begin{bmatrix} 1-\lambda & -2 \\ 2 & -3-\lambda \end{bmatrix}\right)$

$$= (1 - \lambda)(-3 - \lambda) - (-2)(2) = (\lambda - 1)(\lambda + 3) + 4$$

$$= \lambda^2 + 2\lambda + 1 = (\lambda + 1)^2 = 0$$

$$\therefore \lambda = -1$$

고유값, 고유벡터

- **특성 방정식**
  - $(A - \lambda I)v = \left(\begin{bmatrix} 1 & -2 \\ 2 & -3 \end{bmatrix} - \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\right) v$

    $= \begin{bmatrix} 1+1 & -2 \\ 2 & -3+1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

    $= \begin{bmatrix} 2 & -2 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$
  - $2v_1 - 2v_2 = 0$

    $\therefore v_1 = v_2$
  - 즉, $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 또는 단위벡터인 $v = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$

고유값, 고유벡터

- **파이썬 실습**

Eigen value, Eigen vecter

- 넘파이(**numpy**) 라이브러리의 **array** 함수를 이용하여 행렬(matrix)를 생성함.

```
In [85]: A= np.array([[1,-2],[2,-3]])
```

```
In [83]: A
```

```
Out[83]: array([[ 1, -2],
                [ 2, -3]])
```

- 넘파이(**numpy**) 라이브러리의 **linalg** 서브패키지에서 고유값과 고유벡터를 구하는 **eig** 명령을 이용

```
In [84]: np.linalg.eig(A)
```

```
Out[84]: (array([-0.99999998, -1.00000002]),
          array([[0.70710678, 0.70710678],
                 [0.70710678, 0.70710678]]))
```

```
In [86]: lambda_1, v_1 = np.linalg.eig(A)
```

```
In [87]: lambda_1
```

```
Out[87]: array([-0.99999998, -1.00000002])
```

```
In [88]: v_1
```

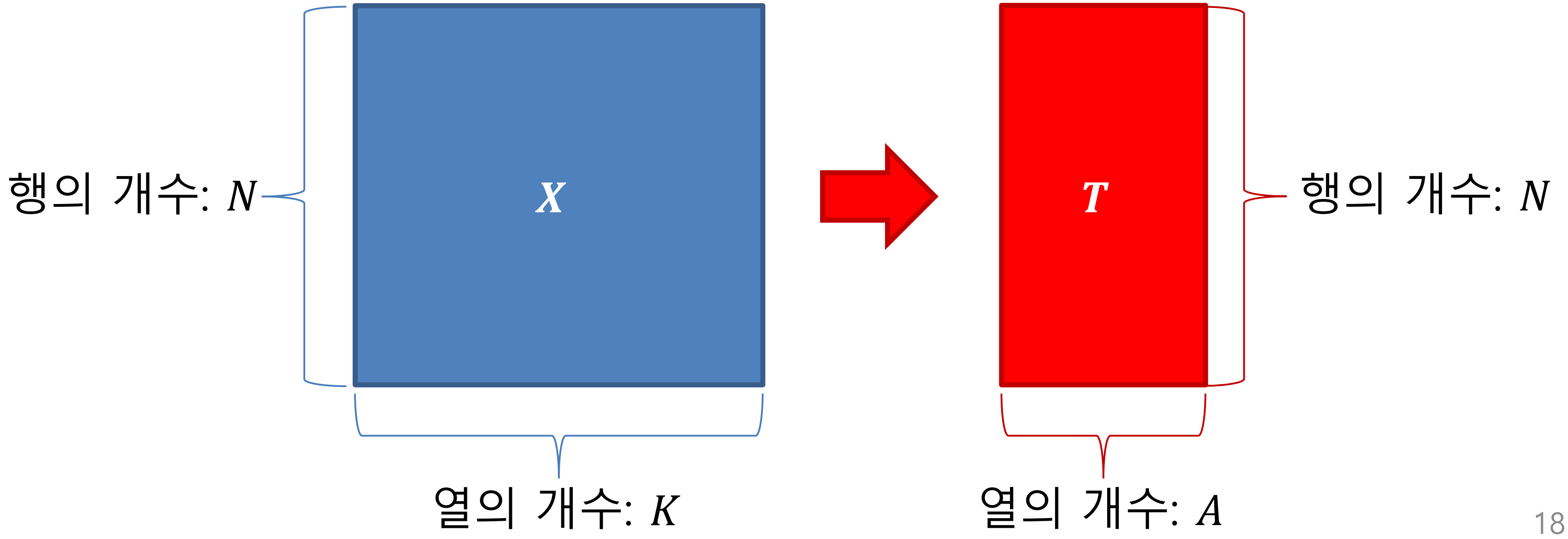
```
Out[88]: array([[0.70710678, 0.70710678],
                [0.70710678, 0.70710678]])
```

Contents

- 연구 내용
- 고유값, 고유벡터
- **주성분 분석**
- 시공간 데이터 실습 (Python)

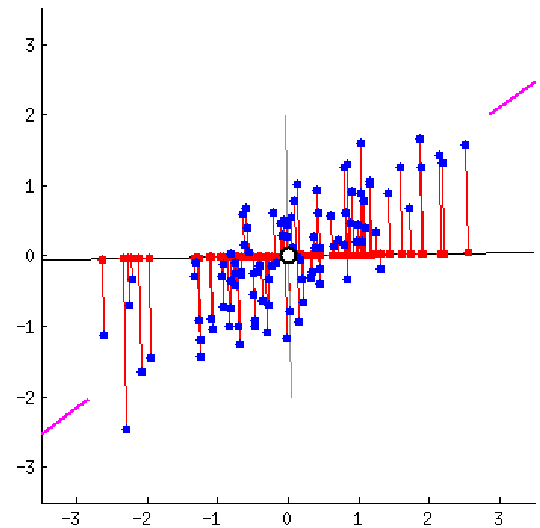
주성분 분석

- **주성분 분석 이란?**
  - **주성분 분석 (Principal Component Analysis, PCA)**은 고차원 데이터 집합이 주어졌을 때, 원래의 고차원 데이터와 가장 비슷하면서 더 낮은 차원의 데이터를 찾아내는 방법임.
  - 대표적인 **차원축소 (dimension reduction) 방법** 중 하나임.



주성분 분석

- **주성분 분석의 개요**
  - 즉, PCA는 데이터의 분산을 **최대한 보존**하면서 서로 직교하는 새로운 기저(축)를 찾아, 고차원 공간의 샘플들을 저차원의 공간으로 변환하는 기법임.
  - 데이터의 분산을 **최대한 보존**한다는 의미는?
    - 데이터들의 흩어진 정도가 가장 큰 경우인 방향벡터(v)를 주성분으로 찾는다 의미임.



[출처] https://builtin.com/data-science/step-step-explanation-principal-component-analysis

주성분 분석

- **PCA의 정의**
  - PCA는 K개의 독립변수로 구성된 $X = [X_1 \quad \cdots \quad X_K]$를 선형결합을 이용해 축소하는 방법임. (i.e., weighted averages)

$$Y_1 = a_{11}X_1 + a_{21}X_2 + \cdots + a_{K1}X_K$$

    - 각 독립변수들의 n개의 unit들을 가지고 있으므로,

$$X_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{n1} \end{bmatrix}, X_2 = \begin{bmatrix} x_{12} \\ \vdots \\ x_{n2} \end{bmatrix}, \cdots, X_K = \begin{bmatrix} x_{1K} \\ \vdots \\ x_{nK} \end{bmatrix}$$

$$\begin{bmatrix} y_{11} \\ \vdots \\ y_{n1} \end{bmatrix} = a_{11}\begin{bmatrix} x_{11} \\ \vdots \\ x_{n1} \end{bmatrix} + a_{21}\begin{bmatrix} x_{12} \\ \vdots \\ x_{n2} \end{bmatrix} + \cdots + a_{K1}\begin{bmatrix} x_{1K} \\ \vdots \\ x_{nK} \end{bmatrix}$$

주성분 분석

- **PCA의 정의**

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1K} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nK} \end{bmatrix}_{n \times K}$$

- 따라서, 선형결합을 matrix notation 으로 표기하면,
  - $a_1 = [a_{11} \quad \cdots \quad a_{K1}]^T$

$$\boldsymbol{y_1} = \boldsymbol{X a_1}$$

$$\begin{bmatrix} y_{11} \\ \vdots \\ y_{n1} \end{bmatrix}_{n \times 1} = \begin{bmatrix} x_{11} & \cdots & x_{1K} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nK} \end{bmatrix}_{n \times K} \begin{bmatrix} a_{11} \\ \vdots \\ a_{K1} \end{bmatrix}_{K \times 1}$$

- 그러면, $a_1 = [a_{11} \quad \cdots \quad a_{K1}]^T$를 어떻게 구할 수 있나요?

주성분 분석

- **PCA의 정의**
  - 정규화(Normalization) & 평균 중심화(mean centering)
    - 선형 결합의 정규화

$$\sum_{j=1}^{K} a_{1j}^2 = 1$$

    - 행렬 X에 평균 중심화
      - 모든 변수들에 대한 열-평균값은 0이다. 즉, 평균이 0이 아닌 경우는 평균을 빼줌으로써, 열평균을 0으로 만들어야 함.

$$\begin{bmatrix} \boxed{\begin{matrix} x_{11} \\ \vdots \\ x_{n1} \end{matrix}} & \begin{matrix} \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} x_{1K} \\ \vdots \\ x_{nK} \end{matrix} \end{bmatrix}_{n \times K}$$

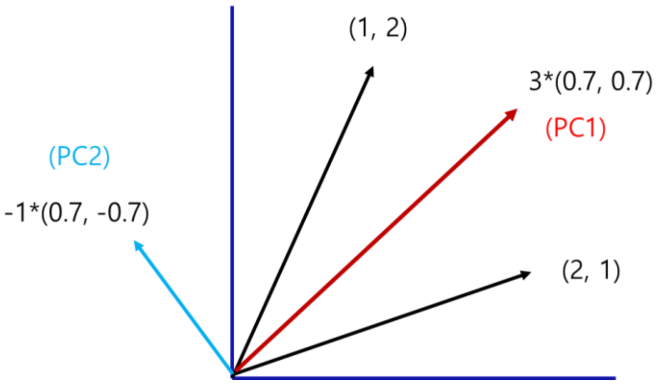
$$\boxed{\frac{1}{n}\sum_{i=1}^{n} x_{i1} = 0}$$

주성분 분석

- **PCA의 정의**
  – 제1주성분(PC_1)은 정규화된 선형결합으로, 가장 큰 분산을 가짐.

$$\boldsymbol{y_1} = \boldsymbol{X a_1}$$

  – PC_2는 정규화된 선형결합으로, 제한 조건은 PC_1과 서로 상관관계가 없는 orthogonal한 방향을 가져야 함.



주성분 분석

- **PCA의 정의**
  - Loadings
    - 벡터 a_j를 loading(로딩)이라고 함. **(⇒ 고유벡터)**

$$a_j = [a_{1j} \quad \cdots \quad a_{Kj}]^T$$

  - Score
    - y_{ij}를 score(스코어)라고 함.

$$y_j = [y_{1j} \quad \cdots \quad y_{nj}]^T$$

주성분 분석

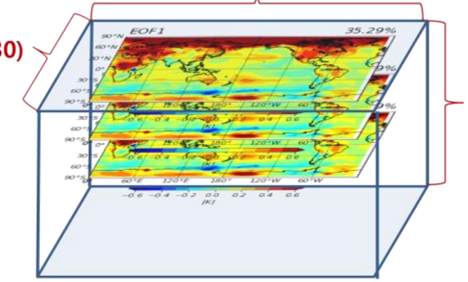
- **PCA의 계산**

① 공분산 행렬(Covariance matrix)

- X와 Y의 공분산 (covariance) 공식
  - $Var(X) = E\left[\left(X - E(X)\right)^2\right] = E(X^2) - [E(X)]^2$
  - $Cov(X,Y) = E\left[\left(X - E(X)\right)\left(Y - E(Y)\right)\right] = E(XY) - E(X)E(Y)$

- 예: 2차원 데이터
  - $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 총 n개의 순서쌍
  - $X = [x_1 \quad \cdots \quad x_n]^T, Y = [y_1 \quad \cdots \quad y_n]^T$
  - $A = [X, Y]_{n\times 2}$
  - $C = \begin{bmatrix} Var(X) & Cov(X,Y) \\ Cov(Y,X) & Var(Y) \end{bmatrix}_{2\times 2}$

주성분 분석



- **PCA의 계산**
  ① 공분산 행렬(Covariance matrix)
    - 예제 데이터: T2m_ERA5_1979_2018_lowR.nc
      - ➢ 1,800차원 데이터: 위도$(30)\times$경도$(60) = 1{,}800$
      - ➢ $(x_{1,1},\dots,x_{1,1800}),\dots,(x_{40,1},\dots,x_{40,1800})$: 총 40개의 순서쌍
      - ➢ $X_1 = \begin{bmatrix} x_{1,1} \\ \vdots \\ x_{40,1} \end{bmatrix}, X_2 = \begin{bmatrix} x_{1,2} \\ \vdots \\ x_{40,2} \end{bmatrix}, \dots, X_{1800} = \begin{bmatrix} x_{1,1800} \\ \vdots \\ x_{40,1800} \end{bmatrix}$
      - ➢ $X = [X_1, X_2, \cdots, X_{1800}]_{40\times 1800}$
      - ➢ $C = \begin{bmatrix} Var(X_1) & \cdots & Cov(X_1, X_{1800}) \\ \vdots & \ddots & \vdots \\ Cov(X_{1800}, X_1) & \cdots & Var(X_{1800}) \end{bmatrix}_{1800\times 1800}$

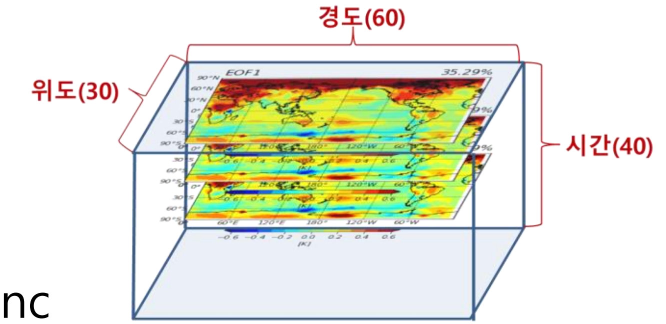
주성분 분석

- **PCA의 계산**
  - ② 고유값, 고유벡터 구하기
    - 고유벡터: 주성분 벡터로서 데이터의 분포에서 분산이 큰 방향을 나타냄.
    - 고유값: 그 분산의 크기를 나타냄.
    - $Cv_i = \lambda_i v_i$
      - v_i: 공분산행렬(C)의 고유벡터
      - λ_i: 고유값, v_i 방향으로의 분산
    - v_1: 가장 분산이 큰 방향
    - v_2: v_1에 수직이면서, 다음으로 가장 분산이 큰 방향
    - 전체 변동에 대한 공헌도

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_K}, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$$

주성분 분석

- **PCA의 계산**
  - ② 고유값, 고유벡터 구하기
    - 예제 데이터: T2m_ERA5_1979_2018_lowR.nc



➢ $C = \begin{bmatrix} Var(X_1) & \cdots & Cov(X_1, X_{1800}) \\ \vdots & \ddots & \vdots \\ Cov(X_{1800}, X_1) & \cdots & Var(X_{1800}) \end{bmatrix}_{1800\times1800}$

➢ $E = \begin{bmatrix} v_{1,1} & \cdots & v_{1,1800} \\ \vdots & \ddots & \vdots \\ v_{1800,1} & \cdots & v_{1800,1800} \end{bmatrix}_{1800\times1800}$

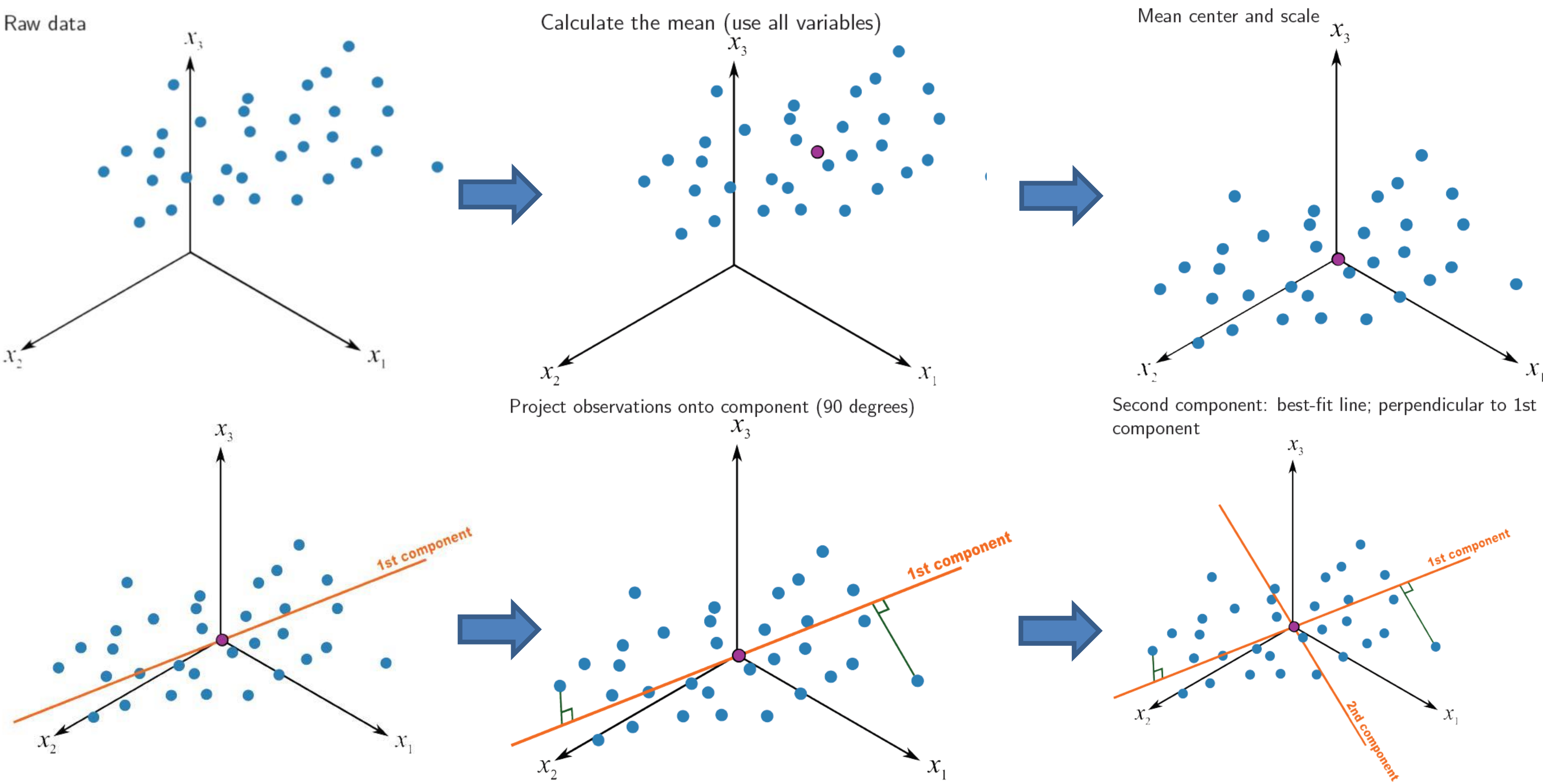
➢ 고유값의 크기 순으로 행렬 E를 재정렬함. ($\boldsymbol{\lambda_1} \geq \lambda_2 \geq \cdots \geq \lambda_K$)

➢ $v_i = [v_{i,1} \quad \cdots \quad v_{i,1800}]^T$와 X의 선형결합을 통해서 Y(score)를 구함.

$$Y_{40\times1800} = X_{40\times1800} \times E_{1800\times1800}$$

주성분 분석

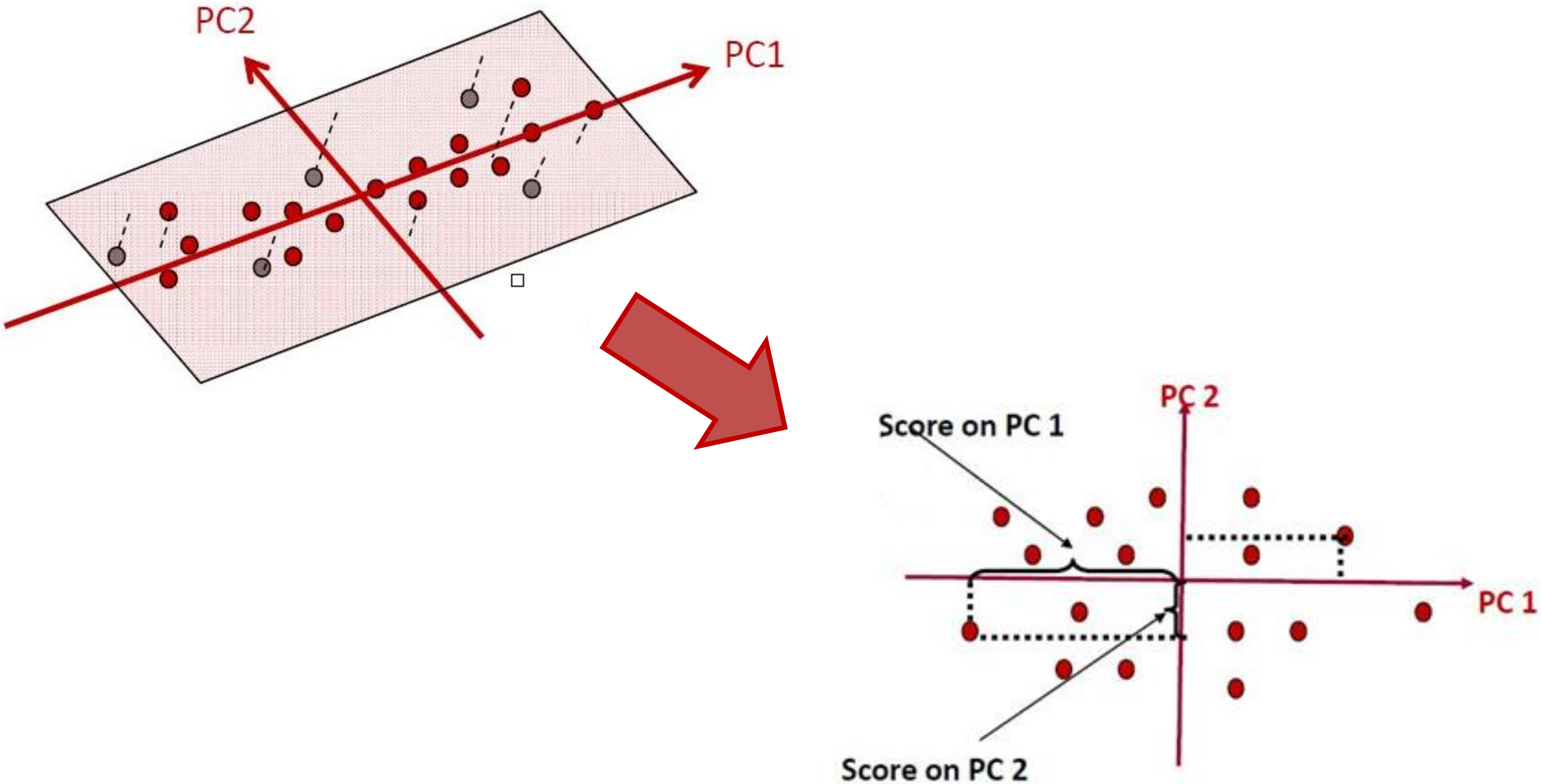
- **요약정리 (1)**



Raw data
Calculate the mean (use all variables)
Mean center and scale
Project observations onto component (90 degrees)
Second component: best-fit line; perpendicular to 1st component
1st component
2nd component
x_1
x_2
x_3

주성분 분석

- **요약정리 (2)**



Contents

- 연구 내용
- 고유값, 고유벡터
- 주성분 분석
- **시공간 데이터 실습 (Python)**

HANYANG UNIVERSITY

HANYANG
UNIVERSITY

한양대학교 ERICA 캠퍼스