Machine Learning

8장 차원 축소(Dimension reduction)

고려대학교 통계학과

박유성

Contents

01 차원 축소(Dimension Reduction)

- 이미지 자료나 DNA 자료들은 입력변수의 차수(dimension)이 높은 경우가 대부분임.

- 해결 방안

  1) 불필요한 특성 변수 제거

  2) 특성변수 $X_{(n \times d)}$ 의 정보 대부분을 가진 새로운 변수로 차원 축소

- 본 장에서는 2)에 대해서 다루며 이는 아래와 같이 표현될 수 있음.

$$\boldsymbol{x}_i \longrightarrow \boldsymbol{z}_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

이때 $\boldsymbol{x}_i$는 $d \times 1$인 벡터, $\boldsymbol{w}$는 $d \times l$ $(l << d)$인 벡터, $i = 1, \ldots n$임.

SVD (Singular Value Decomposition)

- (8.1)과 같이 표현되는 것을 행렬 X의 SVD라고 함.

$$X_{n\times d} = U_{n\times n}\Lambda_{n\times d}V^T_{d\times d} \tag{8.1}$$

이때 U와 V는 직교 행렬(즉, $U^TU=\boldsymbol{I}$, $V^TV=I$)이며

Λ는 최대 $r=\min(n,d)$개의 singular 값들이 대각원소인 대각행렬임.

- $n \geq d$라고 가정하면 (8.1)식은

$$X_{n\times d} = \widetilde{U}_{n\times d}\widetilde{\Lambda}_{d\times d}V^T_{d\times d} \tag{8.2}$$

이때 $\widetilde{\Lambda}$는 Λ에서 singular 값이 0인 행을 제거한 행렬,

$\widetilde{U}$는 U에서 $\widetilde{\Lambda}$에 대응하는 부분행렬임.

SVD (Singular Value Decomposition)

- 그러면 (8.2)는 $\tilde{U}\tilde{U}^T = I$가 되어 $X^T X = V\tilde{U}^T\tilde{U}\tilde{\Lambda}\tilde{V}T = V\tilde{\Lambda}^2 V^T$이므로

$$(X^T X)V = V\tilde{\Lambda}^2 \tag{8.3}$$

이는 고유값 $\tilde{\Lambda}^2$와 이에 대응하는 고유벡터 행렬 V를 구하는 방정식임.

- 마찬가지로 $XX^T = \tilde{U}\tilde{\Lambda}V^T V\tilde{\Lambda}\tilde{U}^T = \tilde{U}\tilde{\Lambda}^2\tilde{U}^T$이므로

$$(XX^T)\tilde{U} = \tilde{U}\tilde{\Lambda}^2 \tag{8.4}$$

그러므로 (8.3)과 (8.4)에서의 $\tilde{\Lambda}^2$는 같다는 것을 알 수 있음.

- 그러므로 $S^2 = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T = \frac{1}{n}X^T X$에 대한 고유벡터 행렬은 V, 고유값은 $\frac{1}{n}\tilde{\Lambda}^2$

02 PCA (Principal Component Analysis)

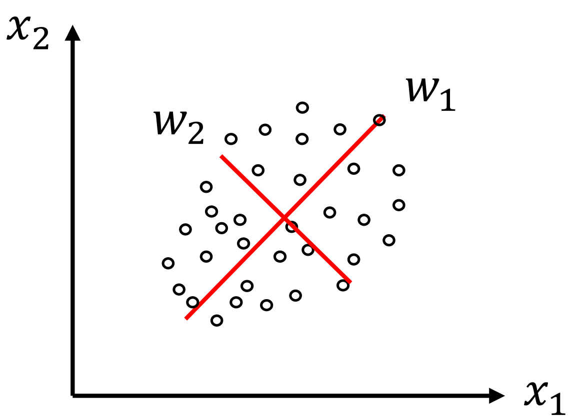
- PCA는 특성변수를 사용하여 설명력이 높은 벡터를 찾아 차원을 축소함.
- 즉 (8.2)에서 가장 큰 l $(l < d)$개의 고유값에 대응되는 고유벡터를 이용함.
- (8.2) 식에서 가장 큰 l개의 고유값을 이용하므로

$$X_{n\times d} \approx U^*_{n\times l} \Lambda^*_{l\times l} V^{*T}_{l\times d} \tag{8.5}$$

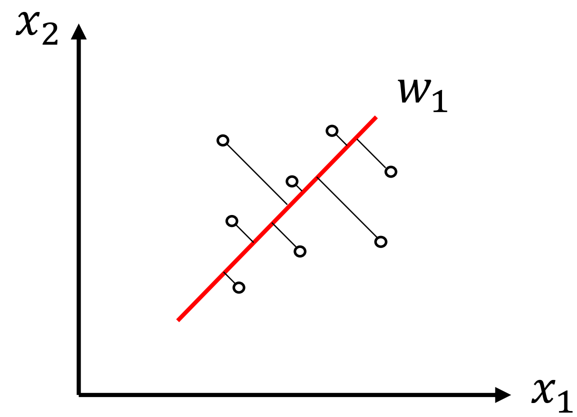
이때 $\Lambda^*_{l\times l}$는 고유값의 크기가 하위 $(d-l)$인 행을 제외한 $\widetilde{\Lambda}_{d\times d}$의 부분행렬, $V^{*T}_{l\times d}$는 $\Lambda^*_{l\times l}$에 대응되는 고유벡터 행렬임.

- (8.5)에 의해 주성분은 $\boldsymbol{w}_{d\times l} = V^*_{d\times l}$이며 새로운 특성변수 $\boldsymbol{z}_i = \boldsymbol{w}^T \boldsymbol{x}_i$가 됨.

특성변수 X가 2차원인 경우



<주성분 w_1과 w_2>



<$w_1{}^T x$: w_1에의 x의 projection>

- $\boldsymbol{w}_1$은 자료의 분산이 가장 큰 축임.
- $\boldsymbol{w}_2$은 $\boldsymbol{w}_1$과 직교하면서 두 번째로 자료의 분산이 큰 축임.
- $\boldsymbol{w}_1^T\boldsymbol{x}$ 은 $\boldsymbol{x}$를 $\boldsymbol{w}_1$에 projection한 것임.
- 즉 l개의 주성분을 이용할 때는 l개의 주성분에 projection한 것임.

Randomized PCA

- 자료의 크기 또는 특성변수의 크기가 매우 크면 주성분 w 를 구하기 위한 SVD 시 계산이 불가능하거나 시간이 많이 소요됨.
- 이 경우에 Randomized PCA가 유용함.
- Randomized PCA는 QR 분해를 이용하여 행렬의 SVD을 함.
- 행렬 $A_{(m \times n)}$ (rank는 k)에 대한 SVD는 다음 장과 같이 계산됨.

Randomized PCA

- 행렬 $A_{(m\times n)}$ (rank는 k)에 대한 SVD

  1) 우선 확률변수 행렬 $P_{(n\times(k+p))}\,(p\geq 0)$ 을 정의함. (이때 행은 서로 독립이며 열은 iid한 정규분포로부터 뽑은 확률변수임.)

  2) $Y=A\times P$를 정의한 후 Y에 대해 QR분해를 하면 $Y=Q\times R$이 됨. 이때 R은 상삼각행렬이며 QQ^TA로 A를 근사할 수 있음.

  3) $B=Q^TA$로 정의하면 이는 $(k+p)\times n$ 행렬이 되고 이를 SVD함. 그러면 $B=\hat{U}\Lambda V^T$가 되어 $QQ^TA=QB$이므로 $U=Q\hat{U}$라 하면 $QQ^TA=U\Lambda V^T$ 임.

  4) 그러므로 $A\approx U\Lambda V^T$로 분해됨.

Kernelized PCA

- PCA는 선형 변환이고 kernelized PCA는 비선형 변환임.
- 특성변수 $\boldsymbol{x}$ 를 비선형 $\boldsymbol{h}(\boldsymbol{x})$로 전환한 후 이에 대해 PCA를 하여 차원축소를 하는 방법임.
- 비선형 $\boldsymbol{h}(\boldsymbol{x})$로 전환하였으므로

$$S_h^2 = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{h}^T(\boldsymbol{x}_i) = \frac{1}{n}H^T(X)H(X)$$

, $\boldsymbol{x}_i$ 는 $d\times 1$ 행렬, $\boldsymbol{h}(\boldsymbol{x}_i)$ 는 $k\times 1$ ($k > d$) 행렬이며

$$H(X) = \begin{bmatrix} h_1(\boldsymbol{x}_1)\ h_2(\boldsymbol{x}_1) \cdots h_k(\boldsymbol{x}_1) \\ \vdots \qquad\qquad\qquad \vdots \\ h_1(\boldsymbol{x}_n)\ h_2(\boldsymbol{x}_n) \cdots h_k(\boldsymbol{x}_n) \end{bmatrix}$$ 임.

Kernelized PCA

- S_h^2 의 고유벡터, 고유값 방정식은 $S_h^2 V = V\Lambda$ 라 하면 새로운 변수는

$$Z(h) = H(X)V \tag{8.6}$$

가 됨.

- $V = \frac{1}{n}H^T(X)Z(h)\Lambda^{-1}$, $S_h^2 = \frac{1}{n}H^T(X)H(X)$ 이므로 (8.6)에 대입하면

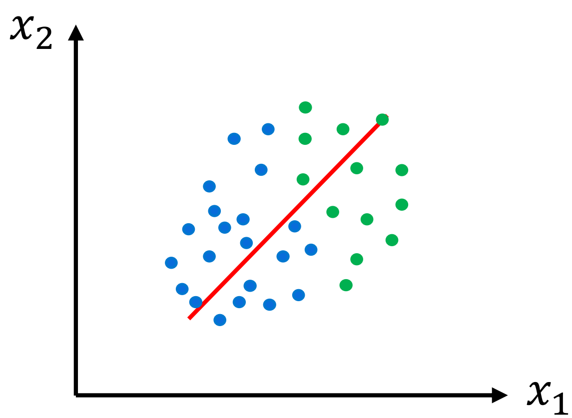
$$\frac{1}{n}\boldsymbol{K}Z(h) = Z(h)\Lambda \tag{8.7}$$

가 되어 행렬 $\boldsymbol{K}$ 의 고유벡터는 $Z(h)$, 고유값은 Λ 됨.

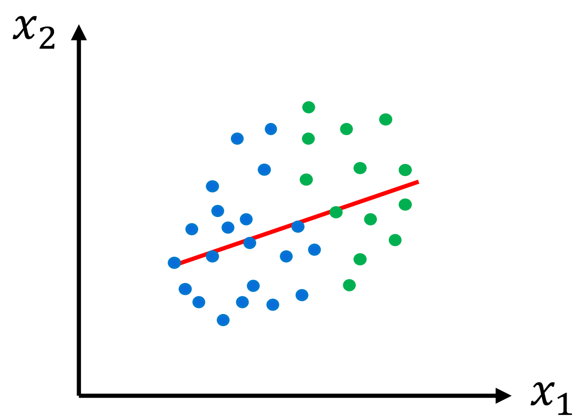
- 이 $\boldsymbol{K}_{ij} = \boldsymbol{h}^T(\boldsymbol{x}_i)\boldsymbol{h}(\boldsymbol{x}_j)$를 커널함수 $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$로 대체하는 커널속임수로 구하면 $Z(h)$의 값 또한 구할 수 있음. (8.6)에서 가장 큰 l 개의 고유값에 대응하는 고유벡터를 이용하여 차원축소를 함.

03 Fisher's LDA (Linear Discriminant Analysis)

- 차원 축소에서 이용되는 방법 중 하나는 Fisher's LDA (Linear Discriminant Analysis)임.
- PCA는 특성변수 X만을 이용해 설명력이 높은 벡터를 찾는 반면 Fisher's LDA는 y의 class를 잘 분류해주는 벡터를 찾는 것이 목적임.



<PCA>



<Fisher's LDA>

파란색은 class 1, 초록색은 class 2

특성변수가 d차원이고 class 수는 c인 경우

- n_c : class c 의 관측치 수 , $\boldsymbol{x}_i^c$: class c에서 i번째 관측치
  각 그룹 내의 특성변수의 평균 $\boldsymbol{m}_c = \frac{1}{n}\sum_{i=1}^{n_c}\boldsymbol{x}_i^c, \quad c = 1, 2, \ldots, C$ 임.
- 그러면 그룹 내 분산(S_w)과 그룹 간 분산(S_B)은

$$S_w = \sum_{c=1}^{C} S_c = \sum_{c=1}^{C}(\boldsymbol{x}_i^c - \boldsymbol{m}_c)(\boldsymbol{x}_i^c - \boldsymbol{m}_c)^T \ , \ S_B = \sum_{c=1}^{C} n_c(\boldsymbol{m}_c - \boldsymbol{m})(\boldsymbol{m}_c - \boldsymbol{m})^T$$

  임.
- $J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_w \boldsymbol{w}}$ 라고 정의하면 이 값이 크다는 것은 그룹 간 분산이
  그룹 내 분산보다 크다는 것을 의미함

특성변수가 d차원이고 class 수는 c인 경우

- 그러므로 $J(\boldsymbol{w})$를 최대로 해주는 $\boldsymbol{w}$를 찾으면 됨.

  즉, $\frac{d}{d\boldsymbol{w}}[J(\boldsymbol{w})] = 0$인 $\boldsymbol{w}$이므로 이를 정리하면 $S_w^{-1} S_B \boldsymbol{w} = \lambda \boldsymbol{w}$ 임.

- $S_w^{-1} S_B$에 대한 고유벡터와 고유값을 구한 후 가장 큰 l개의 고유값에 대응하는 고유벡터를 $\boldsymbol{w}_{d \times l}\ (l < d)$ 라고 정의함.

- 마지막으로 이 $\boldsymbol{w}$를 이용하여 l 차원인 새로운 특성변수 $Z = X\boldsymbol{w}$를 생성함.

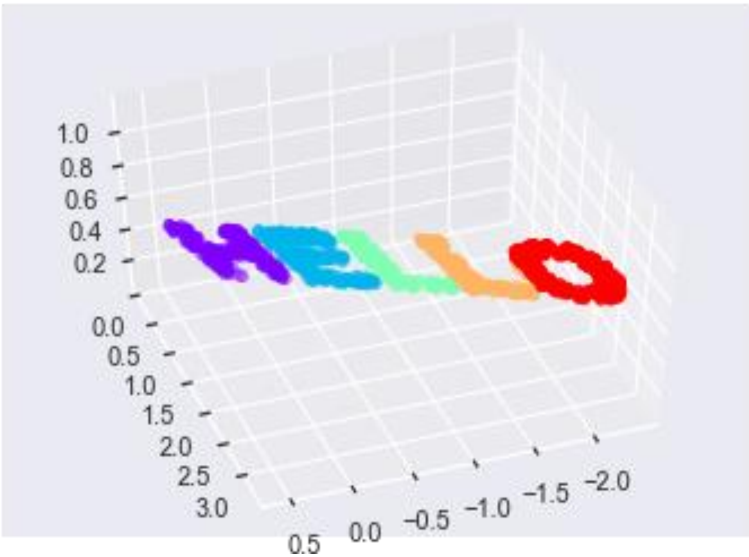
04 MDS (Multidimensional Scaling)

- 고차원에서 측정한 유클리디안(Euclidean) 거리뿐만 아니라 심리적, 감성적인 거리를 시각화가 가능한 1~3차원으로 재 표현하는 기법임.
- PCA의 경우 거리를 왜곡하지만 MDS는 거리를 그대로 옮겨옴.
- $x_1, x_2, \ldots, x_n$을 p차원의 n개의 자료, d_{ij}를 x_i와 x_j의 거리라 하면 MDS는

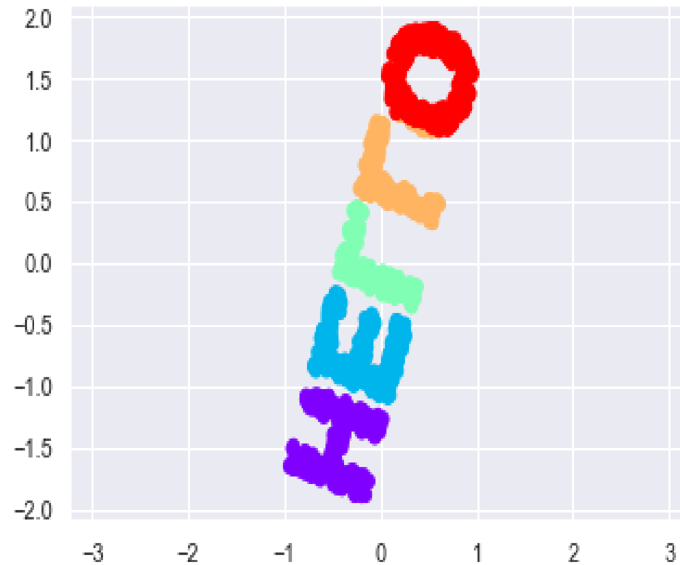
$$\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij} - \|z_i - z_j\|)^2 \tag{8.8}$$

을 최소화하는 z_i, $i = 1,2,\ldots,n$을 찾는 것으로 요약됨. 이때 $\|z_i\| = \sqrt{z_{i1}^2 + \ldots + z_{id}^2}$로 유클리디안 거리이며 z_i는 기울기 하강법으로 구함.

3차원의 HELLO에 대한 적용



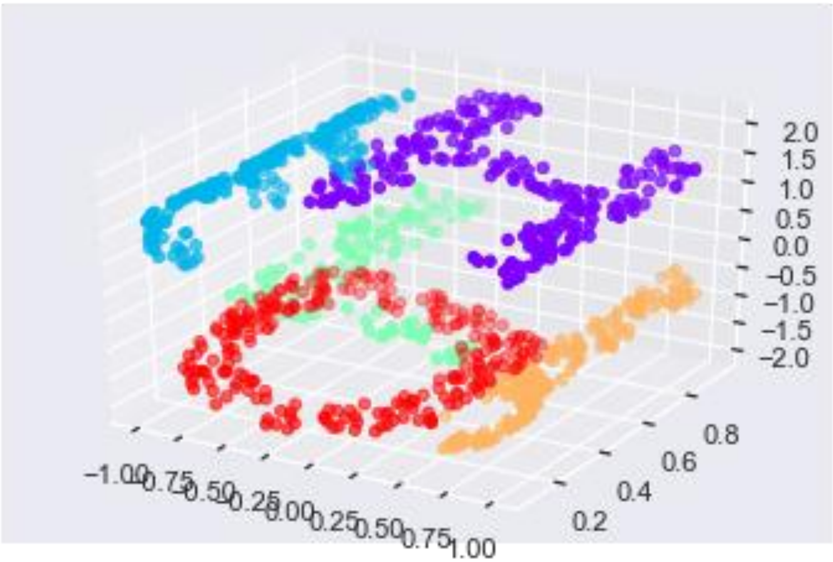
<3차원의 HELLO>



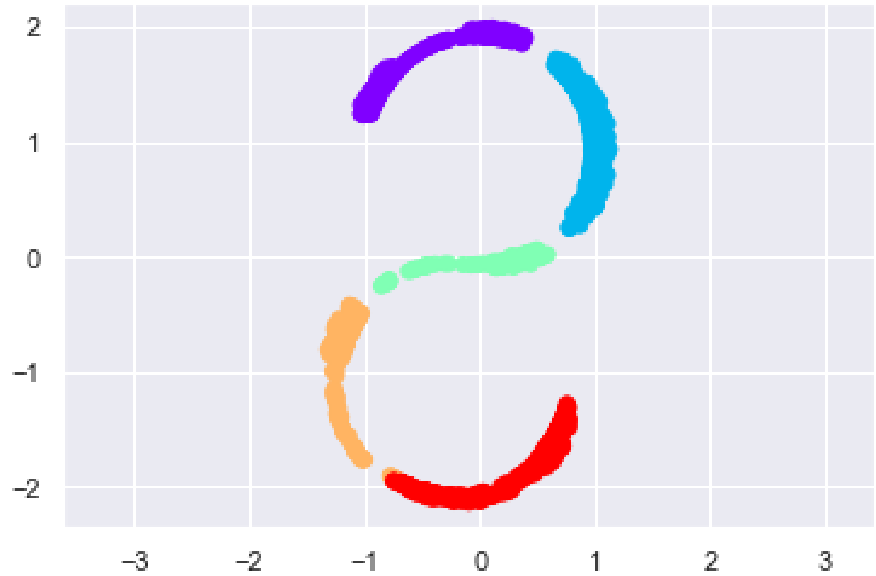
<3차원의 HELLO를 MDS에 의해 차원 축소>

- 약간의 왜곡은 있지만 의미 전달에는 문제가 없음.

3차원의 HELLO에 대한 적용



<S자 커브 형태로 배열된 HELLO>



<S자 커브 형태로 배열된 HELLO를 MDS에 의해 차원 축소>

- 비선형일 경우 앞에서 설명한 MDS는 작동하지 않음.

비선형 자료에 대한 MDS

- 해결 방법

1) K-nearest neighbors를 직선으로 가정하여 이들 자료에만 MDS를 적용하거나 선형 결합을 함.

2) 지도면 상의 최소 거리를 거리 측정함. 국지적으로 평면인 공간을 manifold라고 하기 때문에 이를 manifold 학습이라고 함.

K-nearest neighbors 이용

- Local MDS

- K-nearest neighbors를 이용하는 방법임.
- MDS에 사용한 자료점 $\boldsymbol{x}_i$ 와 차원 축소 자료점 $\boldsymbol{z}_i$ 를 이용하며 Local MDS의 손실함수는

$$L = \sum_i \sum_j (1-\lambda)(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|)^2 I_{[\|\boldsymbol{z}_i - \boldsymbol{z}_j\| < \sigma_i]} + \lambda(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|)^2 I_{[\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \sigma_i]} \quad (8.9)$$

이 됨. 이때 $0 \le \lambda \le 1$이고 만약 $t < \sigma$이면 $I_{[t<\sigma]} = 1$이고 그렇지 않으면 $I_{[t<\sigma]} = 0$

$\lambda = 0$에 가까우면 기존 특성변수, $\lambda = 1$에 가까우면 새로운 특성변수에 비중 둠.

K-nearest neighbors 이용

- (8.9)식은 $I_{[\|z_i - z_j\| < \sigma_i]}$ 는 z_i 를 중심으로 반지름이 σ_i 인 원 안에 포함된 자료만 최소화 하고 $I_{[\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \sigma_i]}$ 역시 $\boldsymbol{x}_i$를 중심으로 일정 범위 안(궁극적으로 k-nearest neighbors)에 있는 $\boldsymbol{x}_j$들과의 거리를 최소화 한다는 의미임.
- 확률적 기울기 강하법으로 손실함수를 최소화 하는 z_i 를 구하며 σ_i의 초기치는 모든 자료를 포함하도록 충분히 크게 부여하고 최종적으로 k-nearest neighbors가 되도록 설정함.
- 특히 (8.9)에서 $\lambda = 0$ 이면 곡선성분분석(curvilinear component analysis, CCA)라고 함.

K-nearest neighbors 이용

- Locally linear embedding (LLE)

- 모든 자료점을 k개의 주변 자료의 선형결합으로 근사할 수 있다고 가정에서 출발하며

$$\sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j=1}^{k} w_{ij}\boldsymbol{x}_j\|^2 \tag{8.10}$$

를 최소화 하는 w_{ij}를 OLS로 구한 후(단, $\sum_{j=1}^{k} w_{ij} = 1$) 이 를 이용해

$$\sum_{i=1}^{n} \|\boldsymbol{z}_i - \sum_{j=1}^{k} w_{ij}\boldsymbol{z}_j\|^2 \tag{8.11}$$

를 최소화 하는 z_i 를 구함. $w_{ij} = 0,\ k < j \leq n$라고 하면 w_{ij}을 행렬 W로 표현할 수 있으며 (8.11)을 최소화 하는 z_i는 $(I - W)^T(I - W)$의 두 번째로 작은 고유벡터가 됨.

Manifold 학습법

- Isomap

- 지도면 상의 최소 거리를 이용한 manifold 학습법임.

- 자료점 각각의 k-nearest neighbors를 구한 후 이들을 선으로 연결한 후 이 선의 길이를 유클리디안 거리로 계산함. $\boldsymbol{x}_i$ 와 $\boldsymbol{x}_j$ 의 거리는 이들 선들의 경로를 따라 최단 거리로 정의한 후 이 거리에 MDS를 적용하여 차원 축소함.

- 이외에도 확률적 거리를 기반으로 한 stochastic neighbor embedding (SNE) 등이 있으나 계속 속도가 느리고 LLE보다 성능이 떨어지는 것으로 나타나 자세한 내용은 생략함.

Q & A