기계 학습

2021210088 허지혜

목차

6.8 매니폴드 학습

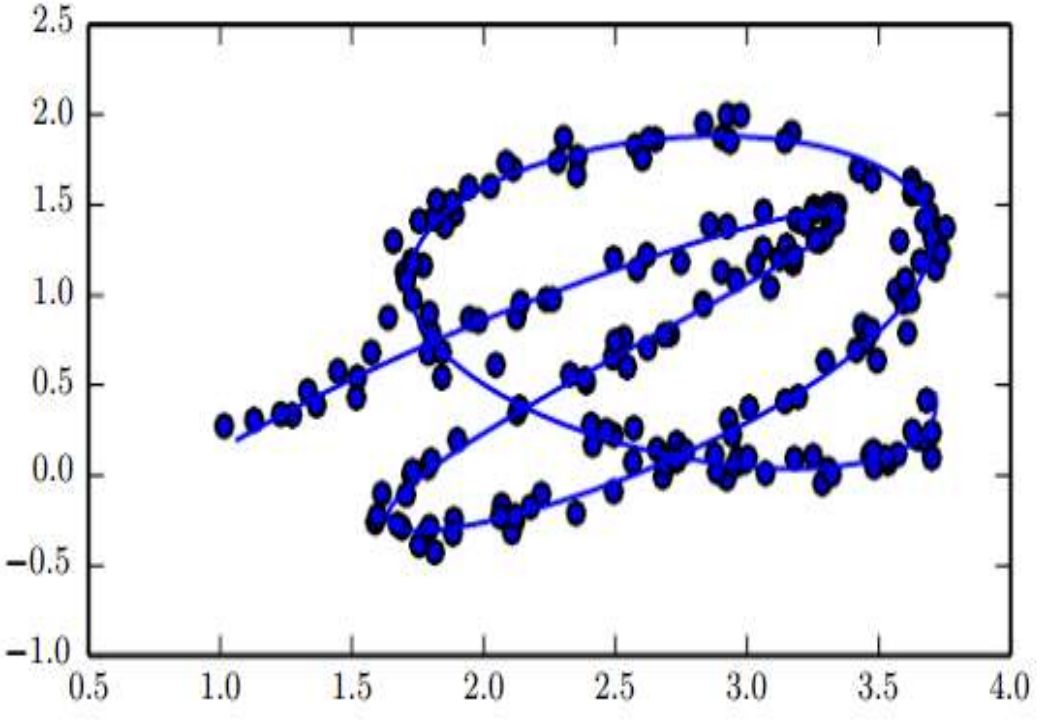
- 지금까지 학습한 공간 변환은 거의 선형 변환에 국한되어 있다.
또한 데이터 분포가 가지는 구조를 간접적으로 표현한다는 한계도 있다.
직접적으로 고려하는 기법은 각각의 점에 대해 이웃과는 기하학적 관련성을 명시적으로
표현하고 의사결정에 반영해야 한다.

- 이 절에서는 학습하는 매니폴드 학습할 데이터 분포의 비선형 구조를 직접적으로 고려한다.

6.8.1 매니폴드란 ?

- 매니폴드 ?

매니폴드는 고차원 공간에 내재한 저차원 공간이다.



위 그림에서 선이 없고 점만 있다고 가정하면
단순한 데이터 분포로 보일 수 있지만,
실제로는 꼬여있는 실처럼 실선은
매니폴드를 나타낸다.
그 선이 모델이 학습해야 할 매니폴드이다.
이 선을 고차원으로 projection하면 꼬인 실이
펴지게 된다.

이와 비슷하게 차원을 높임으로써 매니폴드
학습을 쉽게할 수도 있다.

6.8.1 매니폴드란 ?

그림 6-30 매니폴드

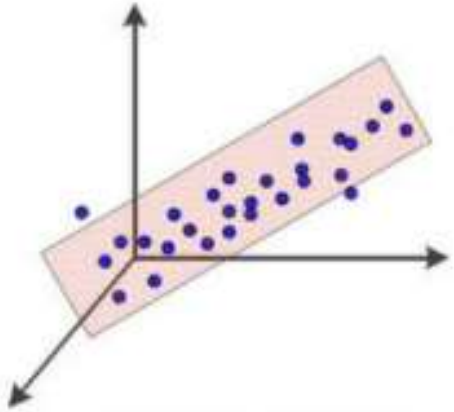
- 위치 데이터를 X = (위도, 경도, 고도)^{T}와 같이
삼차원으로 표현할 수 있다.
하지만 데이터 분포의 구조를 살펴보면,
X = (기준점으로부터 거리)^T 의 일차원에 표현할 수도 있다.

- 매니폴드는 보통 비선형 구조를 가지는데,
특정 점을 중심으로 인근만 살피면 선형 구조에 가깝다.
예로, 지구의 외피는 구 지만 한반도의 지도를 그릴 땐
평면으로 간주하는 이치와 비슷하다.

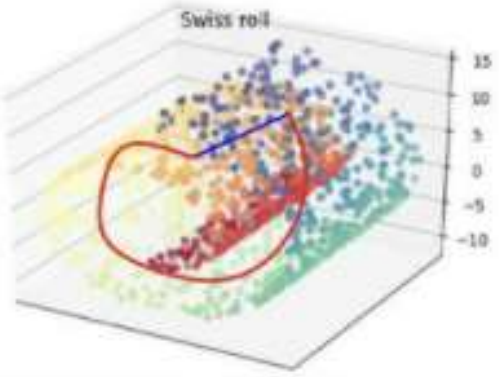
6.8.1 매니폴드란 ?

- 매니폴드 가정

"real-world data presented in high-dimensional spaces are expected to concentrate in the vicinity of a manifold M of much lower dimensionality d_M, embedded in high-dimensional input space R^d. 고차원 공간에 주어진 실제 세계의 데이터는 고차원 입력 공간 R^d에 내재한 훨씬 저차원인 d_M차원 매니폴드의 인근에 집중되어 있다."



(a) PCA로 찾은 매니폴드가 적절한 상황



(b) 비선형 매니폴드가 필요한 상황

그림 6-31 매니폴드 가정

매니폴드 가정을 설명하기 위해 인위적으로 만든 데이터이다.

매니폴드를 어떻게 찾고, 어떻게 표현할 것인가 ?

6.8.2 IsoMap

• 알고리즘

최근접 이웃 그래프를 구축한다.

1) 각 점에 대한 k-최근접 이웃을 구하여 유클리디안 거리를 n * n 행렬 M에 채운다.
이때 각 행은 k개 요소만 0이 아닌 값을 가진다. 나머지 요소는 빈 곳이 된다.
2) 빈 곳은 최단 경로의 길이를 계산하여 채운다.

• IsoMap
M의 고유 벡터를 계산하고, 큰 순서대로 d_{row}의 고유 벡터를 선택한다.
이들 고유 벡터가 새로운 저차원 공간 형성한다.
(고유 벡터가 d차원이 아니라 n차원이므로 PCA처럼 투영을 이용해 저차원으로 변화 불가)
I번째 샘플의 k번째 좌표는 $\sqrt{\lambda_k}v_k^i$ 로 변환(v_k^i 는 λ_k 에 해당하는 고유 벡터의 i번째 요소)

예) 2차원으로 변환하는 경우, $\chi = \{x_1, x_2, \dots, x_n\}$의 x_i는 $(\sqrt{\lambda_1}v_1^i, \sqrt{\lambda_2}v_2^i)^T$ 로 변환된다.

6.8.2 IsoMap

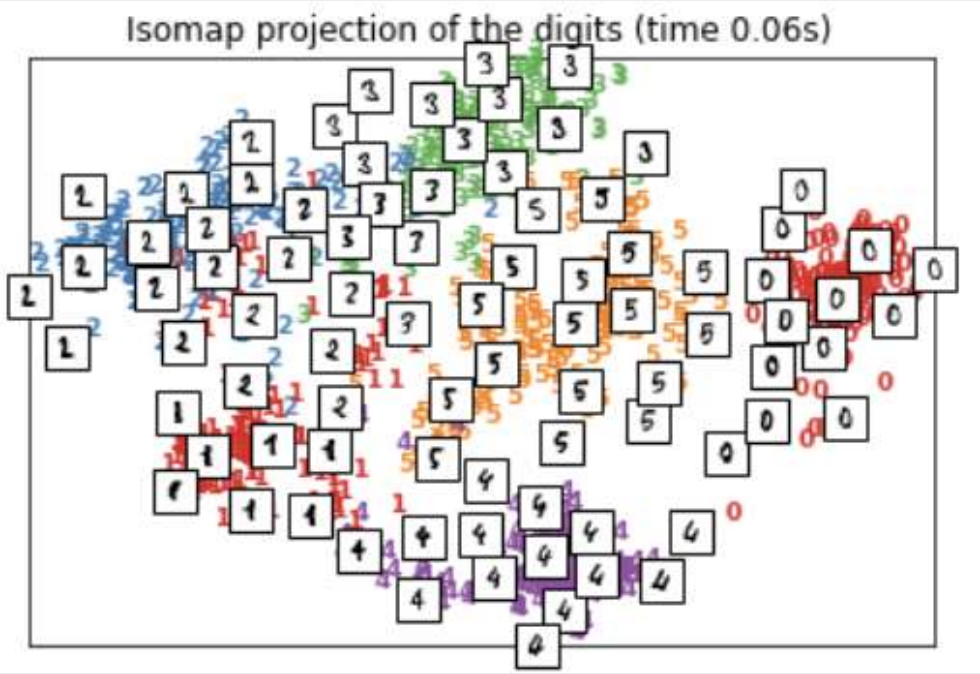
• IsoMap은 거리 행렬 M을 구할 때 k를 적절하게 설정해야 한다.

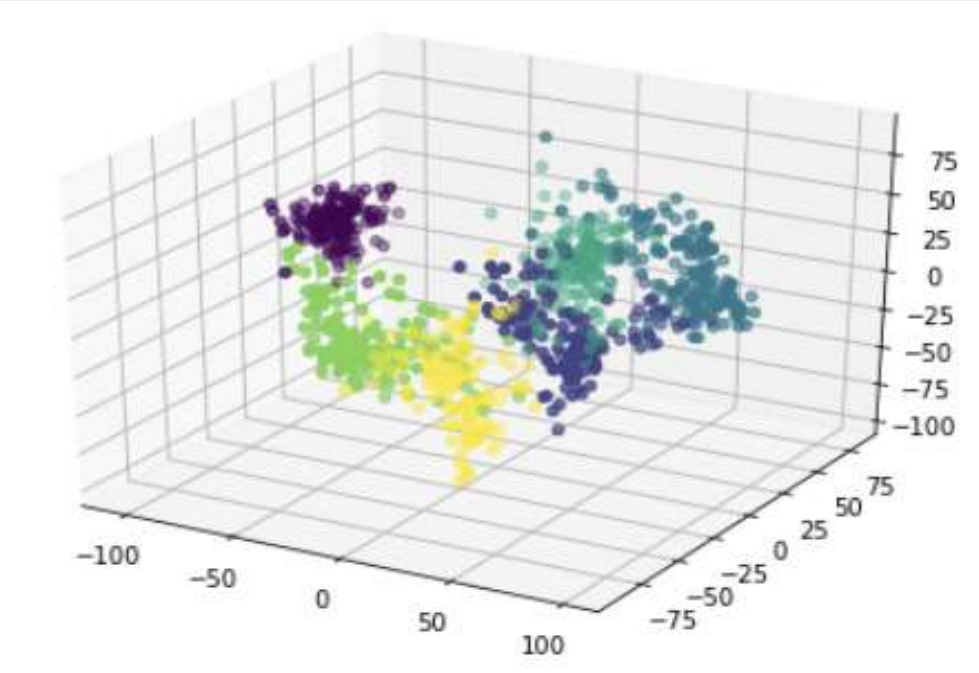
너무 크면 최단 경로를 사용해야 적절한 샘플 쌍이 유클리디언 거리를 사용하는 문제가 생기고
너무 작으면 멀리 있는 샘플 쌍 사이에 경로가 없어 불연속 공간이 되는 문제가 생긴다.

• 훈련집합이 커지면 M의 크기가 너무 커져 저장하는 문제 뿐 아니라,
고유벡터를 구하는 수치적인 문제가 발생한다.

6.8.2 IsoMap

- 예) MNIST 7개의 손글씨 Dataset을 Isomap에 활용하여 차원 축소





출처 : https://woosikyang.github.io/first-post.html

6.8.3 LLE

- **LLE**

거리 행렬 M 대신에 식 (6.42)의 함수를 최소로 하는 가중치 행렬 W를 사용한다.

$$\epsilon(\mathbf{W}) = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in \{\mathbf{x}_i \text{의 이웃}\}} w_{ij}\mathbf{x}_j \right\|_2^2 \qquad (6.42)$$

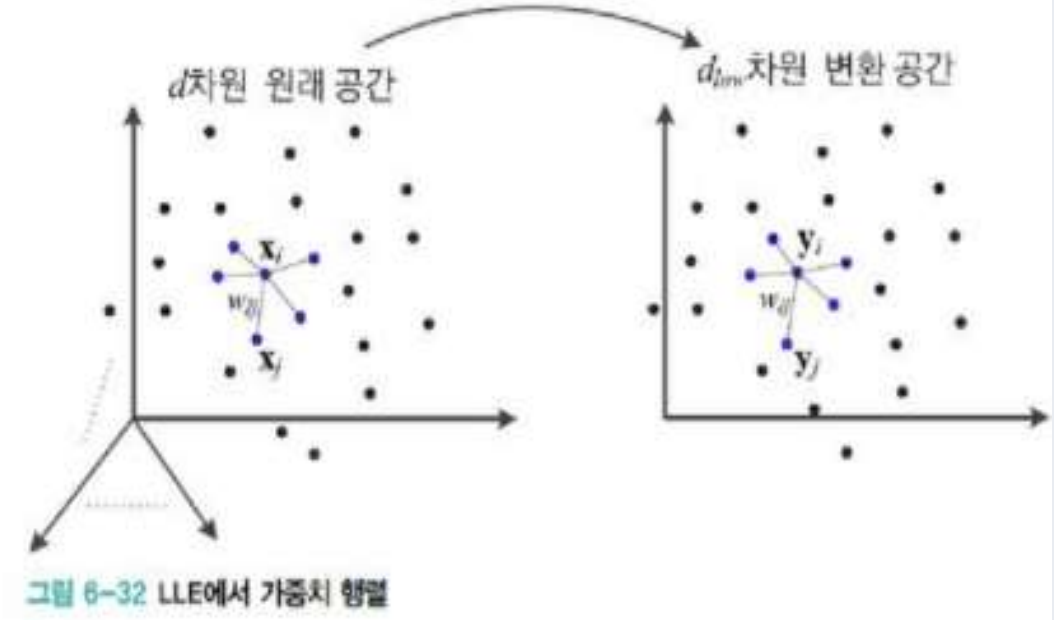


그림 6-32 LLE에서 가중치 행렬

- 첫번째 단계에서 k-근접 이웃을 구한다.
아래 사진은 k = 5 로 설정한 상황이다.

x_i를 k-최근접 이웃의 선형결합 $\sum_{x_j \in \{x_j \text{의 이웃}\}} w_{ij}x_j$ 으로 근사화하는 셈이다.

여기서 식 (6.42)를 최소화하는 행렬 W를 찾아내면 된다.

6.8.3 LLE

- **LLE**

저차원 공간에서는,
변환된 저차원 공간의 점을 $\mathbb{X}' = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$ 라고 하면
식 (6.43)을 최소화 하는 X' 를 찾아야한다.

$$\phi(\mathbb{X}') = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{\mathbf{y}_j \in \{\mathbf{y}_i\text{의 이웃}\}} w_{ij}\mathbf{y}_j \right\|_2^2 \qquad (6.43)$$

고차원 원래 공간에서의 식 (6.42)와 저차원 변환 공간에서 식 (6.43)을 비슷하게 유지함으로써 원래 데이터와 변환된 데이터가 비슷한 구조를 가진다.

6.8.4 t – SNE(Stochastic Neighbor Embedding)

그림 6-33 MNIST를 이용한 매니폴드 학습 기법의 성능 비교

• 원래 공간에서 유사도를 측정한다.
x_i와 x_j 의 유사도를 식 (6.45)의 확률로 측정한다.

$$p_{j|i} = \frac{\exp\left(\frac{-\|x_i - x_j\|_2^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\|x_i - x_k\|_2^2}{2\sigma_i^2}\right)} \quad (6.44)$$

$$p_{ij} = p_{ji} = \frac{p_{j|i} + p_{i|j}}{2n} \quad (6.45)$$

6.8.4 t – SNE(Stochastic Neighbor Embedding)

• 변환된 공간에서의 유사도는 스튜던트 t 분포로 측정한다.
y_i와 y_j는 변환된 공간에서의 점이다.

$$q_{ij} = \frac{\left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2\right)^{-1}}{\sum_{k \neq i}(1 + \|\mathbf{y}_i - \mathbf{y}_k\|_2^2)^{-1}} \qquad (6.46)$$

원래 데이터와 변환된 데이터의 구조가 비슷해야 하기 때문에
확률 분포 P와 Q가 비슷할수록 좋다.
비슷한 정도를 측정하기 위해 식 (6.47)의 KL 다이버전스를 목적함수로 사용한다.

$$J(\mathbb{X}') = KL(P \parallel Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\log\left(\frac{p_{ij}}{q_{ij}}\right) \qquad (6.47)$$

6.8.4 t – SNE(Stochastic Neighbor Embedding)

• 학습 알고리즘

목적함수 J를 최소로 하는, 즉 P와 Q의 KL 다이버전스를 최소하는 X′를 찾는 문제이다.

밑의 식인 경사 하강법을 이용한다.

$$\frac{\partial J}{\partial \mathbf{y}_i} = 4\sum_{j=1}^{n}(p_{ij}-q_{ij})(\mathbf{y}_i-\mathbf{y}_j)\left(1+\|\mathbf{y}_i-\mathbf{y}_j\|_2^2\right)^{-1} \quad (6.48)$$

알고리즘 6–5 t-SNE

입력: $\mathbb{X}=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\}$, 반복 횟수 T, 학습률 ρ, 모멘텀 계수 α

출력: $\mathbb{X}'=\{\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_n\}$

1 $\mathbb{X}$의 모든 샘플 쌍에 대해 식 (6.45)로 p_{ij}를 계산한다.

2 $N(0,10^{-4}\mathbf{I})$ 가우시안 분포로부터 초기해 $\mathbb{X}'^{(0)}=\{\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)},\cdots,\mathbf{y}_n^{(0)}\}$을 샘플링한다.

3 for (t=1 to T)

4 식 (6.46)으로 $\mathbb{X}'^{(t-1)}$의 모든 쌍에 대해 q_{ij}를 계산한다.

5 for (i=1 to n)

6 식 (6.48)로 그레이디언트 $\frac{\partial J}{\partial \mathbf{y}_i}$를 계산한다.

7 if (t>1) $\mathbf{y}_i^{(t)}=\mathbf{y}_i^{(t-1)}+\eta\frac{\partial J}{\partial \mathbf{y}_i}+\alpha(\mathbf{y}_i^{(t-1)}-\mathbf{y}_i^{(t-2)})$ // 학습률과 모멘텀 적용

8 else $\mathbf{y}_i^{(t)}=\mathbf{y}_i^{(t-1)}+\eta\frac{\partial J}{\partial \mathbf{y}_i}$

6.8.5 귀납적 학습 모델과 트랜스덕티브 학습 모델

• 트랜스덕티브 학습 모델
훈련집합 이외의 새로운 샘플을 처리할 능력이 없는 모델이다.
IsoMap, LLE, t-SNE 는 모두 트랜스덕티브 모델이다.
데이터 가시화라는 목적에 관한 한 PCA나 오토인코더와 같은 귀납적 모델보다 성능이 뛰어나다.

> "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate problem. 어떤 문제를 풀 때 데이터가 제한되어 있으면, 그 문제를 직접 풀어야 한다. 중간 문제로서 좀 더 일반적인 문제를 풀 필요가 전혀 없다."

• 귀납적 모델
훈련집합 이외의 새로운 샘플을 처리할 능력이 있는 모델
앞에서 배운 IsoMap, LLE, t-SNE 를 제외한 지금까지 공부한 모든 모델

• 주어진 문제에 따라 둘 중 적절한 것을 선택하는 지혜가 필요하다!

끝