기계학습 차원축소 정리노트

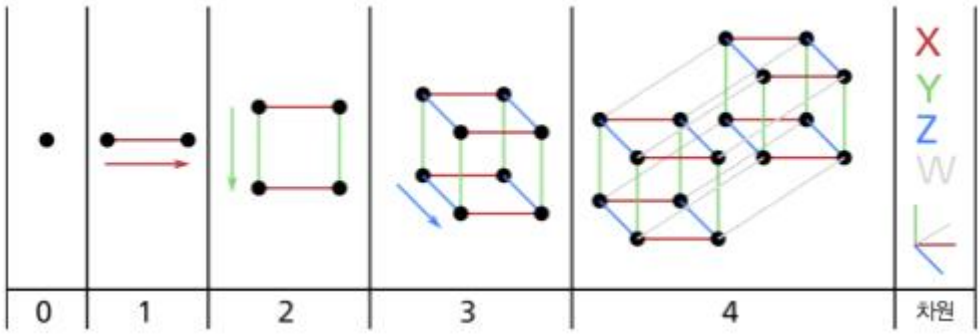
4조 - (이희구, 유제우)

※ 4조 의견

Q. 차원 축소란 뭘까?

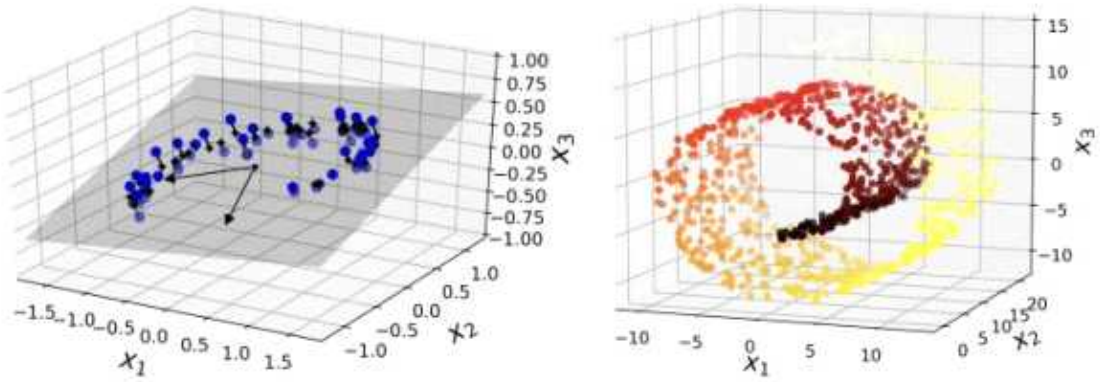
차원 축소는 특성 수를 줄여서 학습 불가능한 문제를 학습 가능한 문제로 만드는 기법이다. 훈련 속도가 빨라지지만, 일부 정보가 유실되어 성능이 저하 될 수 있다. 훈련 속도를 높이는 것 외에 데이터 시각화에도 유용하다.

Q. 차원의 저주?



차원의 저주는 샘플의 특성이 많으면 학습이 매우 어려워진다는 것을 의미한다.
3차원을 초과하는 고차원의 공간을 직관적으로 상상하기 어렵다. 차원이 커질수록 두 지점 사이의 거리가 매우 커지는 문제가 있다. 고차원은 많은 공간을 가지고 있어서, 고차원 데이터셋은 매우 희박할 위험이 있다. 특성수가 아주 많은 경우, 훈련 샘플 사이의 거리가 매우 커서 과대적합 위험도가 커진다. 그 이유는 두 샘플 사이의 거리가 멀어서 기존 값들을 이용한 추정이 매우 불안정해지기 때문이다. 이에 대한 해결책으로는 샘플 수를 늘려서 해결할 수 있다는 것이 있다. 그러나 고차원의 경우 충분히 많은 샘플 수를 준비하는 일은 사실상 불가능하다.

Q. 고차원 공간에서 데이터의 분포는 어때?



대부분의 문제는 훈련 샘플이 모든 차원에 걸쳐 균일하게 퍼져 있지 않다. 또한, 많은 특성은 거의 변화가 없는 반면, 다른 특성들은 서로 강하게 연관되어 있고, 모든 훈련 샘플이 고차원 공간 안의 저차원 부분 공간에 놓여 있다는 것을 확인할 수 있다.

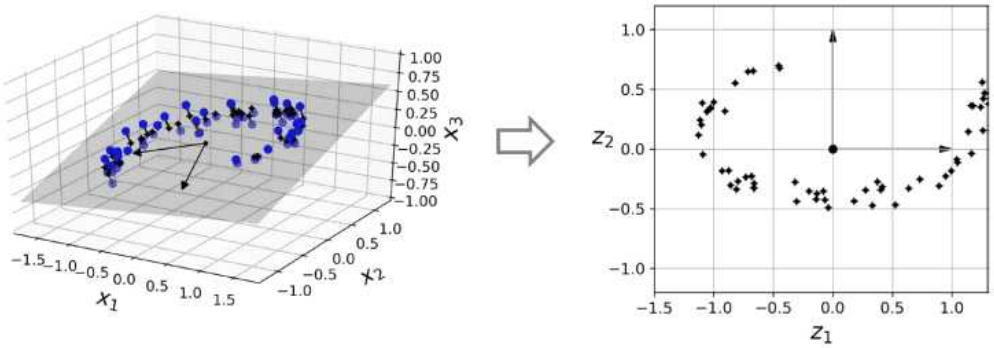
Q. 차원을 감소시키는 두 가지 주요한 접근법에는 어떤 것들이 있어?

대표적으로 투영과 매니폴드 학습이 있다.

Q. 투영? 투영이 뭐야?

투영이란 물체의 그림자를 어떤 물체 위에 비추는 일을 의미한다. n차원 공간에 존재하는 d차원 부분공간을 d차원 공간으로 투영한다.

Q. 투영이 정확히 어떤 것인지 감이 안와. 예를 들어 설명해줘



▲ 그림 8-2 2차원에 가깝게 배치된 3차원 데이터셋

▲ 그림 8-3 투영하여 만들어진 새로운 2D 데이터셋

모든 훈련 샘플을 이 부분 공간에 수직으로 (즉, 샘플과 평면 사이의 가장 짧은 직선을 따라) 투영하면 [그림 8-3]과 같은 2D 데이터셋을 얻는다. 또한, 데이터셋의 차원을 3D에서 2D로 줄일 수 있다. 각 축은 (평면에 투영된 좌표인) 새로운 특성 z1과 z2에 대응된다.

Q. 그럼 차원 축소에서 투영을 계속 사용하면 되는 건가?

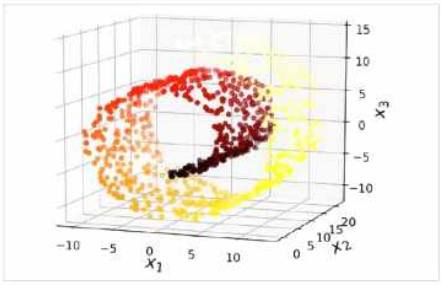


그림 8-4 스위스 롤 데이터셋

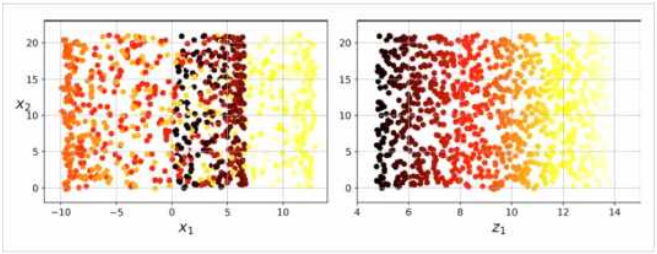


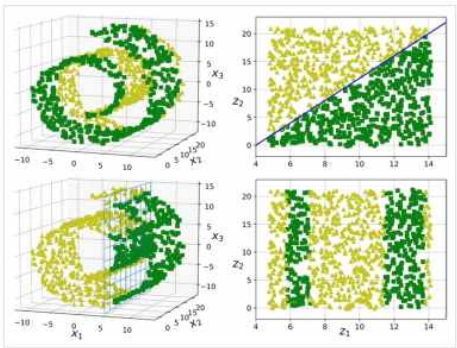
그림 8-5 평면에 그냥 투영시켜서 뭉개진 것(왼쪽)과 스위스 롤을 펼쳐놓은 것(오른쪽)

차원 축소에 있어서 투영이 언제나 최선의 방법은 아니다. 많은 경우 스위스 롤 데이터 셋처럼 부분 공간이 뒤틀리거나 휘어 있기도 한다. 그냥 평명에 투영시키면 [그림 8-5]의 왼쪽처럼 스위스 롤의 층이 서로 뭉개진다. 스위스 롤을 펼쳐서 [그림 8-5]의 오른쪽처럼 보다 적절한 2차원 데이터 셋을 얻을 수 있다.

Q. 매니폴드 학습은 뭐야?

스위스 롤은 2D 매니폴드의 한 예이다. 2D 매니폴드는 고차원 공간에서 휘어지거나 뒤틀린 2D 모양을 형성한다. d 차원 매니폴드는 국부적으로 d 차원 초평면으로 보일 수 있는 n 차원 공간의 일부이다. 대부분 실제 고차원 데이터 셋이 더 낮은 저차원 매니폴드에 가깝게 놓여 있다는 매니폴드 가정에 근거한다.

Q. 이 학습법에는 문제가 없어?



▲ 그림 8-6 저차원에서 항상 간단하지 않은 결정 경계

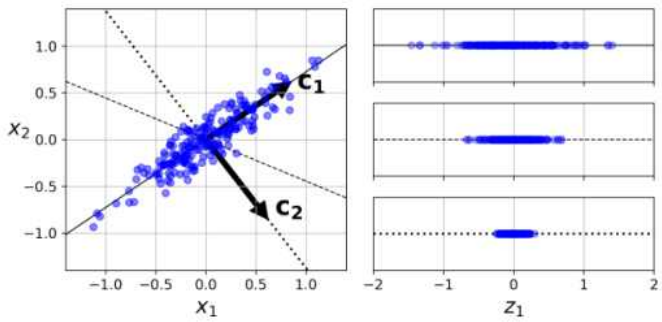
저차원의 매니폴드 공간으로 차원축소를 진행하면 보다 간단한 매니폴드가 된다는 가정과 함께 사용된다. 그러나 암묵적인 가정이 항상 유효하지는 않는다. 차원 감소는 훈련 속도는 향상되지만 항상 더 낫은 솔루션이 아니다.

Q. PCA에 대해 설명해줘

PCA는 차원축소 기법 활용 학습 알고리즘으로 투영 기법 알고리즘이다. PCA는 주성분분석이고, 커널 PCA는 비선형 투영이다. 또한, 매니폴드 기법 알고리즘은 LLE 지역 선형 임베딩이다.

Q. 분산 보존에 대해 설명해줘

저차원으로 투영할 때 훈련 세트의 분산이 최대로 보존되는 축을 선택해야 한다. 분산이 최대로 보존되어야 정보가 가장 적게 손실된다. 원본 데이터 셋과 투영된 데이터 셋 사이의 평균제곱거리 최소화하는 축을 선택한다.



▲ 그림 8-7 투영할 부분 공간 선택하기

저차원의 초평면에 훈련 세트를 투영하기 전에 먼저 올바른 초평면을 선택한다. c 1 벡터가 위치한 실선 축으로 투영하는 경우가 분산을 최대한 보존한다. c 2 벡터가 위치한 점선 축으로 투영된 것은 분산을 매우 적게 보존한다. 가운데 파선의 투영된 것은 분산을 중간 정도로 보존한다.

Q. 주성분에 대해 설명해줘

주성분 축 중에서 첫 번째 주성분은 훈련 세트에서 분산을 최대한 보존하는 축이다. 두 번째 주성분은 첫 번째 주성분과 수직을 이루면서 분산을 최대한 보존하는 축이다. 세 번째 주성분은 첫 번째, 두 번째 주성분과 수직을 이루면서 분산을 최대한 보존하는 축이다. 이러한 축들을 데이터 셋에 있는 차원의 수만큼 찾는다.

Q. 파이썬 코드로 데이터 셋의 차원을 줄이는 방법을 알려줘

PCA 모델을 사용해 데이터셋의 차원을 2로 줄이기

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X2D = pca.fit_transform(X)
```

```
pca.components_

array([[-0.93636116, -0.29854881, -0.18465208],
       [ 0.34027485, -0.90119108, -0.2684542 ]])
```

Q. 적절한 차원수를 선택하는 방법을 알려줘

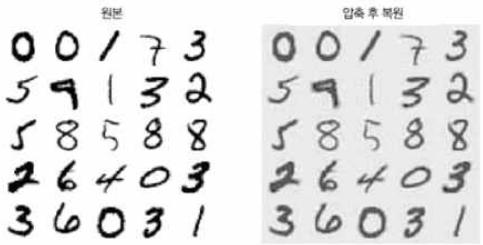
설명된 분산 비율의 합이 충분한 분산, 약 95% 정도가 되도록 하는 차원의 수를 선택한다. 보통 데이터 시각화 목적으로 차원을 2~3개로 줄인다.

```
pca = PCA()
pca.fit(X_train)
cumsum = np.cumsum(pca.explained_variance_ratio_)
d = np.argmax(cumsum >= 0.95) + 1
```

```
d
154
```

차원을 축소하지 않고 PCA를 계산한 뒤 훈련 세트의 분산을 95%로 유지하는 데 필요한 최소한의 차원 수를 계산한다.

Q. 압축을 위한 PCA?



차원을 축소하고 난 후에는 훈련 세트의 크기가 줄어든다. PCA를 MNIST 데이터셋의 차원 축소를 위해 사용할 수 있다.

Q. 랜덤 PCA?

```
rnd_pca = PCA(n_components=154, svd_solver="randomized", random_state=42)
X_reduced = rnd_pca.fit_transform(X_train)
```

랜덤 PCA라 부르는 확률적 알고리즘을 사용해 처음 d개의 주성분에 대한 근삿값을 빠르게 찾을 수 있다. d가 n보다 많이 작으면 완전 SVD보다 훨씬 빠르다. svd_solver 매개변수를 "randomized"로 지정한다.

Q. 점진적 PCA?

훈련세트를 미니배치로 나눈 후 IPCA(점진적 PCA)에 하나씩 주입이 가능하다. 온라인 학습에 적용 가능하다. 넘파이의 array_split() 함수, memmap() 클래스 활용이 가능하다. 이를 통해 전체 훈련 세트를 사용하는 fit() 메서드가 아니라 partial_fit() 메서드를 미니배치마다 호출하고, 바이너리 파일로 저장된 (매우 큰) 데이터셋을 마치 메모리에 들어있는 것처럼 취급할 수 있는 도구를 제공하여 미니배치/온라인 학습이 가능하다.

Q. 커널 PCA?

```
from sklearn.decomposition import KernelPCA

rbf_pca = KernelPCA(n_components=2, kernel="rbf", gamma=0.04)
X_reduced = rbf_pca.fit_transform(X)
```

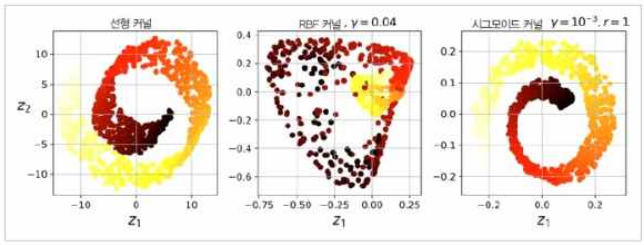
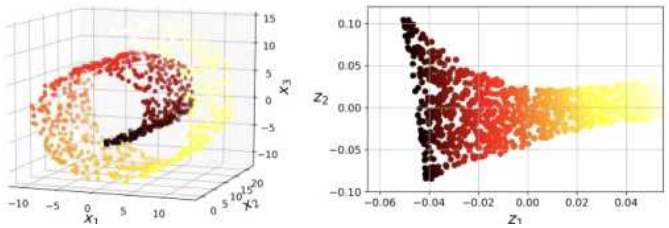


그림 8-10 여러 가지 커널의 kPCA를 사용해 2D로 축소시킨 스위스 롤

커널트릭을 PCA 적용해 차원 축소를 위한 복잡한 비선형 투영을 수행한다. 투영된 후에 샘플의 군집을 유지하거나 꼬인 매니폴드에 가까운 데이터 셋을 펼칠 때도 유용하다.

Q. LLE가 뭐지?



▲ 그림 LLE를 사용하여 펼쳐진 스위스 롤

지역선형임베딩(LLE)은 비선형 차원축소 기법으로 투영이 아닌 매이폴드 학습에 의존한다. LLE는 가장 가까운 이웃에 얼마나 선형적으로 연관되어 있는지 측정하고, 국부적인 관계가 가장 잘 보존되는 훈련 세트의 저차원 표현을 찾는다. 또한, 잡음이 너무 많지 않은 경우 꼬인 매니폴드를 펼치는 데 작동한다.

Q. 다른 차원 축소 기법에는 어떤 것들이 있어?

이전 학습랜덤 투영, 다차원 스케일링, Isomap, t-SNE(t-distributed stochastic neighbor embedding), 선형 판별 분석(LDA, linear discriminant analysis) 등이 있다.

다양한 PCA 기법 사용 방법에 대해 고민하고 조사해봤습니다.

[압축을 위한 PCA]
차원을 축소하여 데이터를 압축하고, 압축된 형태에서도 원래 데이터의 대부분의 정보를 보존하는 것이다. **이미지나 텍스트 데이터와 같이 크기가 큰 데이터의 경우, 차원을 줄여 저장 공간을 절약하거나 계산 속도를 향상시키는 데 사용**된다.

[랜덤 PCA]
SVD보다 빠르게 처음 몇 개의 주성분을 찾기 위한 확률적 알고리즘으로, **데이터가 매우 크고 첫 몇 개의 주성분에만 관심이 있는 경우**, 랜덤 PCA를 사용하여 계산 효율성을 향상시킬 수 있다.

[점진적 PCA]
대량의 데이터셋을 처리할 때 메모리에 한 번에 올릴 수 없는 경우 사용하는 방법으로, **온라인 학습에 적합**하다. 대용량 데이터셋에 대한 PCA를 수행할 때, 전체 데이터를 한 번에 처리하지 않고 미니배치로 나누어 점진적으로 학습할 때 사용된다.

[커널 PCA]
비선형 데이터에 대한 차원 축소를 수행하는 데 사용되며, 데이터를 고차원 공간으로 매핑하여 선형 분리 가능한 형태로 만든다. **고차원이지만 비선형 구조를 가지고 있는 데이터셋에서 선형 분류 또는 군집화를 수행할 때 사용**된다.

이러한 PCA의 다양한 변형은 각각의 특징에 따라 선택되며, 데이터의 특성과 목적에 맞게 적절한 기법을 선택하는 것이 중요하다. 압축을 위한 PCA는 저장 공간이나 계산 속도를 개선하려는 경우에 적합하며, 랜덤 PCA는 계산 효율성이 중요한 경우에 유용하다. 점진적 PCA는 대용량 데이터셋에 대한 온라인 학습에 적합하며, 커널 PCA는 비선형 데이터에 대한 효과적인 차원 축소를 제공한다.