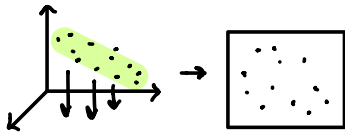


차원 축소

- 훈련 속도 향상
- 경우에 따라 성능 향상
- 시각화에 유리

8.2.1 투영



연관성이 높은 특성들 → 고차원 공간 내 저차원 부분 공간에 위치
즉, 3D → 2D로 투영 가능

8.2.2 매니폴드 학습 (위와 달리 두터운 경우)

ex) 2차원 매니폴드 (스위스 롤, 3D에 있으나 2D로 표현 가능)

자유도 높은 데이터셋 (무작위)을 압축할 수 있도록 도와줌

8.3 PCA 주성분 분석 Principal Component Analysis

1) 분산을 최대한 보존하는 축을 선택한다
= 평균제곱근 최소화

2) 첫번째 축에 직교하고 남은 분산을 최대한 보존하는 두번째 축을 찾는다. ... 반복 ⇒ i번째 주성분 (PC)
numpy - `Svd()`: 주성분 구하는 함수 / sklearn에서는 PCA할 때 자동으로 `mean → 0`만든다

3) 첫번째부터 d번째까지 주성분으로 정의한 초평면에 투영 → d차원으로 축소

sklearn - `PCA()` (`components_` 주성분들
`explained_variance_ratio_` 각 주성분이 설명하는 분산 비율)

* 적절한 차원 수 선택하기

충분한 분산이 될 때까지 더하기 (시각화 시에는 2, 3개)

① `PCA(n_components = 0.95)`
보존하려는 분산 값 비율 (0 ~ 1 사이 값)

② `cumsum`을 그래프로 나타낸 뒤 결정

* 랜덤 PCA : 확률적 알고리즘으로 빠르게

`PCA(svd_solver="randomized")`

* 점진적 PCA (IPCA) : 훈련 세트를 미니배치로 나누고 하사색 주입 (큰 훈련세트에서 유용)
`IncrementalPCA()`
fit 알고 partial_fit 반복문

8.4 커널 PCA 커널트릭으로 비선형 특영
KPCA

`KernelPCA(kernel="rbf")`

매핑 → 재구성 → 재구성 원상 최소화하는 커널선택

8.5 LLE 지역 선형 임베딩
비선형 차원 축소 각 훈련 샘플이 이웃에 얼마나 선형 연관되었는가. (강음 적은 경우 유용)

8.6 랜덤 특영

다차원 스케일링 (MDS) : 거리 보존

Isomap : 가까운 샘플 연결하고 간격 유지하면서 축소

t-SNE : 시각화

LDA → 클래스 멀리 떨어지게 : 분류 전제 유용