9기_ML팀_이시현_요약본

6.1 차원 축소 개요

1. 차원 축소 정의: 매우 많은 피처(feature)로 구성된 다차원 데이터 세트의 차원을 새로운 차원의 데이터 세트로 줄이는 과정이다.
2. 차원 증가 시 문제점:
    1. 데이터 포인트 간 거리가 기하급수적으로 멀어져 데이터가 희소(sparse)해진다.
    2. 적은 차원에서 학습된 모델보다 신뢰도가 떨어진다.
    3. 피처 간 상관관계가 높을 경우 다중 공선성(multicollinearity) 문제로 모델 예측 성능이 저하된다.
3. 차원 축소 방법:
    1. 피처 선택(Feature Selection): 데이터 특징을 나타내는 주요 피처만 선택하고 불필요한 피처를 제거한다.
    2. 피처 추출(Feature Extraction): 기존 피처를 압축하여 저차원의 중요 피처로 추출하며, 추출된 특성은 기존 피처와 완전히 새로운 값이 된다.
4. 주요 차원 축소 알고리즘: PCA, SVD, NMF 등이 있으며, 잠재적인 요소를 추출하여 데이터를 더 잘 설명할 수 있게 한다.
5. 차원 축소 활용 분야:
    1. 이미지: 픽셀 수를 줄여 이미지 압축 및 과적합(overfitting) 방지에 사용된다.
    2. 문서: 단어들의 잠재 요소를 시맨틱(Semantic) 토픽으로 찾아내어 토픽 모델링에 사용된다 (SVD, NMF 기반).
6. 차원의 저주 (Curse of Dimensionality): 데이터의 변수 수(차원)가 늘어날수록 데이터 분석 및 학습이 어려워지는 현상이다. 고차원 데이터셋에서는 훈련 샘플들이 서로 멀리 떨어져 있어 새로운 샘플이 훈련 샘플과 멀리 떨어져 있을 가능성이 높다.

7. 차원 축소 접근 방법:
    1. 투영(Projection): 고차원 공간의 훈련 샘플을 저차원 공간에 수직으로 투영하여 새로운 특성 축에 대응시킨다.
    2. 매니폴드 학습(Manifold Learning): 훈련 샘플이 놓여있는 매니폴드(국부적으로 d차원 초평면으로 보일 수 있는 n차원 공간의 일부)를 모델링하는 방식이다.

6.2 PCA (Principal Component Analysis)

1. PCA 개요: 여러 변수 간의 상관관계를 이용해 이를 대표하는 주성분(Principal Component)을 추출하여 차원을 축소하는 기법이다.
    1. 데이터의 변동성(Variance)을 가장 높게 가지는 축(주성분)을 찾아 데이터를 이 축으로 투영한다.
    2. 변동성이 가장 큰 방향을 기준으로 첫 번째 벡터 축을 생성하고, 이후 축은 이전 축과 직각이 되는 벡터를 순차적으로 생성한다.
2. PCA의 선형대수 관점: 입력 공분산 행렬(Covariance Matrix)을 분해하여 고유벡터와 고유값을 구한다.
3. PCA 스텝:
    1. 입력 데이터 세트의 공분산 행렬을 생성한다.
    2. 공분산 행렬의 고유벡터와 고유값을 계산한다.
    3. 고유값이 가장 큰 순으로 K개(변환 차수)의 고유벡터를 추출한다.
    4. 추출된 고유벡터를 이용해 입력 데이터를 선형 변환한다.
4. 붓꽃(Iris) 데이터셋 적용 예시: 4개 속성을 2개 차원으로 압축하여 시각화했다.
    1. PCA는 분산 기반이므로, 사이킷런의 StandardScaler를 이용해 평균 0, 표준편차 1인 정규 분포로 스케일링해야 한다.
    2. PCA(n_components=2) 클래스를 사용하여 4차원 데이터를 2차원으로 변환했다.
    3. 2개의 주성분(pca_component_1, pca_component_2)으로 변환 후에도 품종(Setosa, Versicolor, Virginica) 구분이 명확하게 가능했다.

4. explained_variance_ratio_ 속성을 통해 각 컴포넌트가 설명하는 전체 변동성 비율을 확인할 수 있다.

5. PCA 분류 성능 비교 (붓꽃 데이터):
    1. 원본 데이터(4차원): 랜덤 포레스트 교차 검증 평균 정확도 0.96을 기록했다.
    2. PCA 변환 데이터(2차원): 평균 정확도 0.88로, 4개 속성이 2개로 감소하면서 예측 성능이 약 8% 하락했다.

    -예측 성능은 감소했지만, 속성 개수가 50% 감소한 것을 고려하면 원본 데이터 특성을 상당 부분 유지하고 있다.

6. 신용카드 데이터셋 적용: 24개 속성 중 BILL_AMT 관련 6개 속성 간 상관도가 매우 높았다.
    1. 변동성 확인: 6개 속성을 2개 컴포넌트로 변환했을 때, 첫 번째 축이 90% 이상의 변동성을 수용할 정도로 상관도가 높았다.
    2. 분류 예측 비교: 원본 23개 속성 기반 분류 예측 정확도 대비, 6개 컴포넌트 변환 후 예측 정확도는 약 2% 수준의 저하만 발생하여 뛰어난 압축 성능을 보였다.

7. 적절한 차원 수 선택:
    1. 설명된 분산 비율을 차원 수에 대한 함수로 그려 95% 등 원하는 분산 비율을 유지하는 데 필요한 최소 차원 수를 계산한다.
    2. 설명된 분산 비율의 성장이 멈추는 변곡점(Elbow) 지점에서 차원을 축소한다.

6.3 LDA (Linear Discriminant Analysis)

1. LDA 개요: PCA와 유사하게 저차원 공간에 데이터를 투영하여 차원을 축소하는 지도학습 기법이다.
    1. 차이점: PCA가 데이터의 변동성(분산)을 최대화하는 축을 찾는 반면, LDA는 개별 클래스를 최대한 분별할 수 있는 축을 찾는다.
    2. 클래스 간 분산(between-class scatter)을 최대화하고 클래스 내부 분산(within-class scatter)을 최소화하는 비율을 최대화한다.

2. LDA 스텝:
    1. 클래스별 평균 벡터를 구하고, 클래스 내부 분산 행렬(S_w)과 클래스 간 분산 행렬(S_b)을 구한다.
    2. $S_w^{-1}S_b$행렬을 고유벡터로 분해하여 고유값을 추출한다.
    3. 고유값이 가장 큰 순으로 K개 고유벡터를 추출한다.
    4. 추출된 고유벡터를 이용해 입력 데이터를 변환한다.

3. 붓꽃 데이터셋 적용:
    1. LDA는 지도학습이므로 변환 시 클래스 결정 값(target)이 필요하다.
    2. LinearDiscriminantAnalysis(n_components=2) 클래스를 사용해 4차원 데이터를 2차원으로 변환했다.
    3. 시각화 결과: 2차원 평면에 시각화했을 때, Setosa, Versicolor, Virginica 세 품종이 가장 명확하게 분리되었다.

6.4 SVD (Singular Value Decomposition)

1. SVD: 특이값 분해(Singular Value Decomposition)는 PCA와 유사한 행렬 분해 기법으로, m x n 행렬뿐만 아니라 정방행렬이 아닌 행렬도 분해할 수 있다.
    - 분해 형태: $A = U\Sigma V^T$
        - U: m x m 직교 행렬 (좌 특이벡터)
        - Σ: m x n 대각 행렬 (특이값 포함)
        - V^T: n x n 직교 행렬의 전치 (우 특이벡터)

2. Truncated SVD (절단된 SVD): Σ행렬에서 0이 아닌 대각 원소에 대응하는 U와 V^T의 일부만 추출하여 차원을 줄인 형태로 분해하는 방식이다.
    - 원본 행렬을 근사적으로 복원할 수 있으며, 사이파이에서 희소 행렬에 대해 지원된다.
3. SVD 예제: 4x4 랜덤 행렬을 분해하고 복원하는 과정을 통해 SVD의 작동 원리를

확인했다.

- 선형 독립인 로우(행) 개수가 2개일 경우, 분해된 Σ값 중 2개만 0이 아닌 값으로 나타났다.

4. SVD와 PCA의 관계:
    - PCA: 입력 데이터의 공분산 행렬에 대한 고유값 분해를 기반으로 한다.
    - SVD: 원본 데이터 행렬에 직접 적용하여 분해한다.
    - 구현: 사이킷런에서 StandardScaler로 데이터를 중심화(평균 0)하면, PCA는 SVD와 동일한 변환을 수행하게 된다.
    - PCA는 밀집 행렬에 대한 변환만 가능하지만, SVD는 희소 행렬 변환도 가능하다.
    - SVD는 LSA(Latent Semantic Analysis)의 기반 알고리즘으로, 이미지 압축 및 텍스트 토픽 모델링 분야에 사용된다.
5. Truncated SVD 예제: 6x6 임의 행렬을 4개 차원으로 Truncated SVD 분해 후 복원하여 원본 행렬과 근사적으로 복원됨을 확인했다.
6. Truncated SVD와 PCA 비교 (붓꽃 데이터):
    - 스케일링된 붓꽃 데이터에 Truncated SVD와 PCA를 적용하여 2차원으로 변환한 결과, 두 기법의 변환 행렬의 평균 값이 거의 동일하게 나타났다.

6.5 NMF (Non-Negative Matrix Factorization)

1. NMF 개요: 음수가 아닌 행렬 근사(Low-Rank Approximation) 방식의 행렬 분해 기법이다.
    - 특징: 원본 행렬 X를 모두 양수인 두 행렬 W와 H의 곱(으로 분해하는 것이 보장된다.
    - 잠재 특성: 분해된 행렬 W와 H의 요소들은 잠재 특성을 가지며, 원본 행의 요소가 잠재 특성에 얼마나 가중치를 갖는지(W), 잠재 특성이 열(속성)로 어떻게 구성되었는지(H)를 나타낸다.
2. NMF 학습 방식:

    :증배 갱신 규칙(Multiplicative Update Rules)을 사용하여 $W$$W$와 $H$$H$행렬을 반복적으로 갱신한다.

3. NMF와 PCA 비교:
    - PCA: 공분산 행렬을 고유 벡터로 분해하며, 고유 벡터들이 직교하므로 실제 데이터 구조를 잘 반영하지 못할 수 있다.
    - NMF: 대량의 정보를 의미 있는 특징과 변수로 효율적으로 표현할 수 있다.
4. NMF 활용:
    - 토픽 모델링: 뉴스 기사 등에서 단어 기반으로 잠재 특성(토픽)을 추출하는 데 사용된다.
    - 추천 시스템: 사용자-평가 순위 데이터셋을 분해하여 사용자가 평가하지 않은 상품에 대한 잠재적인 요소를 예측하고 추천하는 데 사용된다.
5. 붓꽃 데이터셋 적용: NMF를 사용하여 2개 컴포넌트로 변환 후 시각화했을 때, 품종별로 클러스터링이 가능할 정도로 뛰어난 고유성을 가지고 있음을 확인했다.

추가 자료: 다른 차원 축소 기법들

1. 랜덤 투영 (Random Projection): 무작위 행렬을 이용해 고차원 데이터를 저차원으로 변환하며, 계산이 단순하고 빠르다. 존슨-린덴스트라우스 정리에 기반하여 거리 정보가 보존된다.
2. 다차원 스케일링 (MDS, Multidimensional Scaling): 샘플 간의 거리(distance)를 최대한 보존하면서 차원을 축소하며, 시각화에 적합하다. 대규모 데이터에는 계산 복잡도로 인해 비효율적이다.
3. Isomap: 샘플 간의 지오데식 거리(geodesic distance)를 유지하며 차원을 축소하여 데이터의 비선형 구조(매니폴드 구조)를 보존하는 데 강하다.
4. t-SNE (t-distributed Stochastic Neighbor Embedding): 지역적 구조(Local structure) 보존에 집중하며, 고차원 데이터의 군집 구조 시각화에 매우 강력하다. 계산량이 많고 하이퍼파라미터에 민감하다.
5. LDA (Linear Discriminant Analysis): 분류를 위한 알고리즘이지만, 차원 축소에도 사용되며 클래스 간 분리도를 높이는 투영면을 생성한다.

캐글 노트북: Dimensionality Reduction for Beginners

1. [1단계] 데이터 준비 및 로드: pandas, matplotlib, seaborn, sklearn 라이브러리를 준비하고 유방암 데이터셋을 로드

2. [2단계] 데이터 탐색 및 전처리:
    1. 타겟 변수(악성/양성)를 라벨 인코딩하여 숫자로 변환했다.
    2. StandardScaler를 사용하여 30개 특징(차원)의 스케일을 통일하여 평균 0, 표준편차 1로 변환했다.
3. [3단계] 차원 축소 및 시각화:

#PCA 적용 및 분석

30개 Feature를 시각화가 용이한 2차원으로 압축했다.

1. StandardScaler로 표준화된 데이터에 PCA(n_components=2)를 적용했다.
2. 심층 분석: pca.components_ 속성을 통해 찾아낸 주성분과 원본 특징 간의 관계를 히트맵으로 확인했다. PC1은 대부분의 특징과 양의 상관관계를 보였다.

-결론: PCA는 초기 탐색에 좋고 빠르지만, 복잡한 비선형 구조를 찾는 데는 한계가 있다.

MDS 적용 및 분석

고차원 데이터 내에 숨어있는 저차원 구조(Manifold)를 찾아 시각화하며, 샘플 간의 거리를 보존하는 데 중점을 둔다.

1. 스케일링된 데이터로 MDS(n_components=2)를 실행했다.
2. PCA와 마찬가지로 악성/양성 그룹이 잘 분리되었으며, 2개 차원만으로도 30차원 공간에서의 상대적 거리 관계를 성공적으로 시각화했다.

t-SNE 적용 및 분석

지역적 구조(Local structure) 보존에 집중하여 군집(클러스터)을 찾는 데 최적화된 기법이다.

1. TSNE(n_components=2)를 사용하여 스케일링된 데이터에 적용했다.
2. 유방암 데이터의 군집을 가장 명확하게 분리하는 성능을 보였다.
3. 한계: PCA나 MDS에 비해 계산 속도가 느리며, 시각화된 군집 간의 크기나 거리는 실제 의미를 갖지 않는다 (오직 '누구와 뭉쳐 있는가'가 중요).
4. 기법별 요약 및 비교:
    - PCA: 분산 보존. 빠르고 전반적 분포 확인에 용이하나, 비선형 구조에 약

함.

- MDS: 상대적 거리 보존. 비선형 구조 가능하나, 해석이 어렵고 PCA보다 느릴 수 있음.
- t-SNE: 이웃(Neighbor) 보존. 군집 시각화에 강력하나, 계산량이 많고 축/거리 해석에 주의 필요.

5. 최적 기법 선택: 데이터 분석 목적에 따라 적합한 기법이 다르다.
    - PCA: 데이터의 전반적인 구조와 주요 성분 확인 시.
    - MDS: 샘플 간의 상대적인 거리 관계가 중요할 때.
    - t-SNE: 숨겨진 군집을 시각적으로 명확하게 확인하고 싶을 때.