

## Tobigs19\_Dimensionality\_Reduction\_Week7\_Assignment

### ■ Dimension: Features 개수

#### ■ 차원이 많아질수록 문제 발생

1. 연산량이 기하급수적으로 증가
2. 빈공간이 발생하여 정보의 밀도가 감소
3. 공간을 설명하기 위한 데이터 부족으로 과적합 및 성능 감소

### ■ Dimension Reduction: Features 수를 줄이는 것

#### ■ 관측 데이터를 잘 설명할 수 있는 Latent Space를 찾기 -> 위 문제를 해결하면서 시각화로 활용 가능

#### ■ 축소 방법

##### ◆ Feature selection: Lasso

##### ◆ Feature extraction: PCA, LDA

### ■ PCA: 최적 표현을 위해 최대분산을 찾아 차원을 축소

#### ■ 기존 Features에 선형결합을 통해 새로운 변수들로 변환(변환에 Eigen vector를 이용)

### ■ LDA: 최적 분류를 위해 분별 정보를 최대한 유지하면서 차원을 축소

#### ■ 클래스 내의 분산, 클래스 간의 분산을 동시에 고려

### ■ Manifold Learning: 선형 구조(PCA, LDA)로 설명할 수 없는 비선형 구조를 고려

#### ■ Isomap: 국소적으로 eigen-decompose하는 방법

#### ■ LLE: 국소적인 평면의 데이터들이 축소된 차원에서도 인접하도록 반영

#### ■ SNE: LLE에서 인접이 확률적으로 되게 계산

#### ⇒ T-SNE: 고차원의 데이터를 저차원으로 거리 관계를 유지하면서 임베딩시키는 기법

#### ■ 직관적으로 데이터 구조를 시각화