Ch. 12 Continuous Latent Variables

김채형 이상욱

2020-1 ESC 세션 6주차

2020.05.14

Curse of Dimensionality ; 차원의 저주

차원의 저주는 데이터의 차원이 높아질수록 발생하는 여러 문제들을 통틀어 일컫는 말입니다. 데이터의 차원이 증가할수록 해당 공간의 크기(부피)가 기하급수적으로 증가합니다. 그렇기 때문에 동일한 개수의 데이터의 밀도는 차원이 증가할수록 급속도로 희박(sparse)해집니다. 따라서, 차원이 증가할수록 데이터의 분석 또는 모델링에 필요한 데이터의 샘플 수가 기하급수적으로 증가하게 됩니다. 이러한 어려움을 표현한 용어가 바로 '차원의 저주'입니다. 이와 같은 문제를 해결하기 위해 우리는 '차원 축소'를 실시하게 됩니다.

Dimensionality Reduction ; 차원 축소

차원 축소에는 크게 두가지 방법이 존재합니다.

Feature Selection

기존 변수 중 일부 중요 변수만을 사용하는 기법입니다. 다시 말해 d개의 차원을 가지는 데이터로부터 m개만을 선택하여 사용하는 방법입니다. 이때 어떠한 방법을 통해 d개를 선택할지에는 여러 방법들이 존재합니다.

Feature Extraction

기존 변수를 조합하여 새로운 변수를 만들어내는 기법입니다. 기존 변수 가운데 일부 혹은 전부를 활용할 수 있습니다.

Principal Component Analysis ; PCA ; 주성분 분석

- 오늘 배울 PCA는 차원 축소의 대표적인 방법입니다.
- PCA는 Feature Extraction에 속하는 방법으로,
  기존 변수를 선형결합(linear combination)하여 새로운 변수를 만들어냅니다.
- 즉 PCA는 linear mapping을 찾는 알고리즘으로,
  unsupervised learning에 해당됩니다.
- 먼저 이러한 PCA 알고리즘을 계산하는 데에 쓰이는 고유값 분해와 특이값 분해에 대해 알아보고자 합니다.

Eigen Decomposition ; 고유값 분해

- 행렬 A가 $n \times n$ 행렬이고, 선형독립인 eigenvector를 최대 n 개까지 갖는다고 합시다.

$$A\mathbf{x_1} = \lambda_1\mathbf{x_1},\ A\mathbf{x_2} = \lambda_2\mathbf{x_2},\ \cdots,\ A\mathbf{x_n} = \lambda_n\mathbf{x_n}$$

- 이는 다음과 같이 표현될 수 있습니다.

$$\begin{aligned} A\begin{pmatrix}\mathbf{x_1} & \mathbf{x_2} & \cdots & \mathbf{x_n}\end{pmatrix} &= \begin{pmatrix}A\mathbf{x_1} & A\mathbf{x_2} & \cdots & A\mathbf{x_n}\end{pmatrix} \\ &= \begin{pmatrix}\lambda_1\mathbf{x_1} & \lambda_2\mathbf{x_2} & \cdots & \lambda_n\mathbf{x_n}\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{x_1} & \mathbf{x_2} & \cdots & \mathbf{x_n}\end{pmatrix}\begin{pmatrix}\lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n\end{pmatrix}\end{aligned}$$

Eigen Decomposition ; 고유값 분해

- 위 식은 아래와 같이 표현될 수 있습니다.

$$AX = XD$$

X : eigenvector $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ n개를 세로로 쌓은 $n \times n$ 행렬
D : eigenvalue $\lambda_1, \ldots, \lambda_n$를 대각성분으로 갖고 나머지 성분은 모두 0인 행렬

- 행렬 A에 대한 Eigen Decomposition은 다음과 같습니다.

$$A = XDX^{-1}$$

Singular Value Decomposition ; SVD ; 특이값 분해

- 특이값 분해는 고유값 분해와 마찬가지로 Matrix Decomposition의 한 방법입니다.
- 하지만 특이값 분해는 고유값 분해와 달리 정사각행렬이 아니더라도 적용이 가능합니다.
- SVD는 한개의 행렬을 여러개의 layer의 합으로 나누는 것을 말합니다.

Singular Value Decomposition ; SVD ; 특이값 분해

- 한개의 행렬을 여러개의 layer의 합으로 나눈다는 것은 한개의 행렬을 여러개의 행렬의 합으로 나타낸다는 것을 말합니다.

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T$$

X : $n \times p$ 행렬
$\sigma_1, \ldots, \sigma_n$: $X^T X$의 eigenvalue들
$u_1, \ldots, u_n$: $X X^T$의 eigenvector들 : $n \times 1$ 벡터
$v_1^T, \ldots, v_n^T$: $X^T X$의 eigenvector들의 transpose : $1 \times p$ 벡터

Singular Value Decomposition ; SVD ; 특이값 분해

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T = U\Sigma V^T$$

U : XX^T의 eigenvector들
즉 $u_1, \ldots, u_n$을 세로로 쌓은 $n \times n$ 행렬

V : X^TX의 eigenvalue들의 transpose
즉 $v_1^T, \ldots, v_p^T$을 가로로 쌓은 $p \times p$ 행렬

Σ : XX^T와 X^TX의 eigenvalue들을 크기 내림차순으로 대각원소에 넣은 $n \times p$ 행렬

- 행렬 X에 대한 Singular Value Decomposition은 다음과 같습니다.

$$X = U\Sigma V^T$$

Truncated SVD

- 이제 $n < p$일 때 한 행렬을 n개의 layer로 나눌 수 있게 되었습니다.

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T$$

- 이 중 가장 중요한 d개의 layer만 더해 행렬 X를 요약하려고 합니다.
- 이를 위해 singular value의 크기 순으로 상위 d개만 남깁니다.
- 이것이 바로 Truncated SVD의 idea입니다.

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_d u_d v_d^T = U_d \Sigma_d V_d^T$$

Truncated SVD : Application

- 픽셀들로 이루어진 이미지 역시 행렬로 볼 수 있습니다.
- 각 픽셀은 Red, Green, Blue 색깔을 나타내는 세개의 숫자값을 담고 있습니다.
- 하나의 사진을 R, G, B 값을 갖는 같은 크기의 행렬 3개로 나누고, 각각의 행렬에 SVD를 수행합니다.
- singular value의 크기 순으로 상위 d개만 더함으로써 이미지를 요약할 수 있습니다. => Truncated SVD

Truncated SVD : Application

(a) d=3 (b) d=45 (c) d=297

Figure 1: d의 값에 따른 이미지 변화