MLT Week-2

Sherry Thomas
21f3001449

Contents

Abstract

The week's discourse concentrates on the two primary challenges inherent in Principal Component Analysis (PCA) and endeavors to provide solutions. The solutions are achieved through the utilization of kernel functions and culminates in a more comprehensive and generalized algorithm for PCA.

Introduction

For a given dataset $D \in \mathbb{R}^{d\times n}$, the covariance matrix is $C \in \mathbb{R}^{d\times d}$. PCA for this dataset can have the following two problems:

- Time Complexity: The algorithimic complexity of finding the eigenvalues and eigenvectors of C is $O(d^3)$. Hence, as d grows, the time taken becomes very large.
- Non-Linear Dataset: The dataset may lie in a non-linear subspace. As PCA tries to get linear combination of Principal Components, non-linear datasets may result in non-optimal outputs.

Reducing the Time Complexity to find Eigenvalues and Eigenvectors

Let's take a dataset X with a large number of features(d) i.e. $[d >> n]$ where \ d: no. of features \ n: no. of datapoints

$$X = \begin{bmatrix} | & | & | & & | \\ x_1 & x_2 & x_3 & \dots & x_n \\ | & | & | & & | \end{bmatrix} X \in \mathbb{R}^{d \times n}$$

The covariance matrix C of X is given by,

$$C = \frac{1}{n}(XX^T) \qquad \text{where } C \in \mathbb{R}^{d \times d} \qquad \dots [1]$$

Let w_k be the the eigenvector corresponding to the k^{th} largest eigenvalue λ_k of C. \ We know

$$Cw_k = \lambda_k w_k$$

Substituting C from [1] in the above equation, and solving for w_k, we get

$$w_k = \sum_{i=1}^{n} (\frac{x_i^T w_k}{n\lambda_k}) \cdot x_i$$

$\therefore$ w_k is a linear combination of datapoints! Hence, we can say

$$w_k = X\alpha_k \qquad \text{for some } \alpha_k \qquad \dots [2]$$

Let $X^T X = K$ where $K \in \mathbb{R}^{n \times n}$. We shall use this to solve for α_k. After some algebra (Refer to lectures), we get:

$$K\alpha_k = (n\lambda_k)\alpha_k$$

We know that the non-zero eigenvalues of XX^T and $X^T X$ are the same according to the Spectral Theorm.

Explanation for the above: If $\lambda \neq 0$ is an eigenvalue of XX^T with eigenvector w, then $XX^T w = \lambda w$.

Multiply both sides by X^T to get

$$X^T X(X^T w) = \lambda X^T w$$

But then we see that λ is still an eigenvalue for $X^T X$ and the corresponding eigenvector is simply $X^T w$.

Let β_k be the the eigenvector corresponding to the k^{th} largest eigenvalue $n\lambda_k$ of K.

On solving the eigen equation for K, we get

$$\alpha_k = \frac{\beta_k}{\sqrt{n\lambda_k}} \qquad \dots [3]$$

Using equations [2] and [3], we can get the eigenvalues and eigenvectors of C using K and hence, bringing the time complexity from $O(d^3)$ to $O(n^3)$.

Finding PCA for Non-Linear Relationships

Transforming Features

We solve the problem of non-linear relationships by mapping them to higher dimensions.

$$x \rightarrow \phi(x) \qquad \mathbb{R}^d \rightarrow \mathbb{R}^D \qquad \text{where } [D >> d]$$

To compute D:

Let $x = \begin{bmatrix} f_1 & f_2 \end{bmatrix}$ be features of a dataset containing datapoints lying on a curve of degree two in a two-dimensional space.

To make it linear from quadratic, we map the features to $\phi(x) = \begin{bmatrix} 1 & f_1^2 & f_2^2 & f_1 f_2 & f_1 & f_2 \end{bmatrix}$

Mapping d features to the polygonial power p gives ${}^{d+p}C_d$ new features.

Issue: Finding $\phi(x)$ may be very hard.

Solution for this issue is in the next point.

Kernel Functions

A function that maps $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, and is a "valid", is called a Kernel Function.\ Proof of a "Valid" Kernel:

- Method 1: Exhibit the map to ϕ explicitly. [may be hard]
- Method 2: Using Mercer's Theorem:
  - $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a "valid" kernel if and only if:
    - k is symmetric i.e $k(x, x') = k(x', x)$
    - For any dataset $\{x_1, x_2, \dots, x_n\}$, the matrix $K \in \mathbb{R}^{n \times n}$, where $K_{ij} = k(i, j)$, is Positive Semi-Definite.

Two Popular Kernel Functions: * Polynomial Kernel: $k(x, x') = (x^T x + 1)^p$ * Radial Basis Function Kernel or Gaussian Kernel: $exp(-\frac{||x-x'||^2}{2\sigma^2})$

Kernel PCA

Let's take a dataset X with a large number of features(d) i.e. $[d >> n]$ where

- d: no. of features
- n: no. of datapoints

$$X = \begin{bmatrix} | & | & | & & | \\ x_1 & x_2 & x_3 & \dots & x_4 \\ | & | & | & & | \end{bmatrix}$$

- Step 1: Calculate $K \in \mathbb{R}^{n \times n}$ using a kernel function where $K_{ij} = k(x_i, x_j)$.
- Step 2: Center the kernel using the following formula.

  $$K^C = K - IK - KI + IKI$$

  where K^C is the centered kernel, and $I \in \mathbb{R}^{n \times n}$ where all the elements are $\frac{1}{n}$.

- Step 3: Compute the eigenvectors $\{\beta_1, \beta_2, \ldots, \beta_l\}$ and eigenvalues $\{n\lambda_1, n\lambda_2, \ldots, n\lambda_l\}$ of K^C and normalize to get

$$\forall u \qquad \alpha_u = \frac{\beta_u}{\sqrt{n\lambda_u}}$$

- Step 4: Compute $\sum_{j=1}^{n} \alpha_{kj} K_{ij}^C \quad \forall k$

$$x_i \in \mathbb{R}^d \rightarrow \left[\begin{array}{cccc} \sum_{j=1}^{n} \alpha_{1j} K_{ij}^C & \sum_{j=1}^{n} \alpha_{2j} K_{ij}^C & \ldots & \sum_{j=1}^{n} \alpha_{nj} K_{ij}^C \end{array} \right]$$

Credits

- Professor Arun Rajkumar: The content as well as the notations are from his slides and lecture.