Homework of Singular Value Decomposition

Computational Linear Algebra for Large Scale Problems
Politecnico di Torino

Hajali Bayramov - S288874
Hafez Ghaemi - S289963

June 5, 2021

Abstract

In this homework, we explored two real-world applications of singular value decomposition (SVD). First, we applied it on a clustering benchmark to perform spectral clustering. Second, we used SVD for video background extraction. A file named "INSTRUCTIONS.txt" containing the explanations on how to run the codes and reproduce the results presented in this report, is attached.

1 Introduction

In the first part of this project, a clustering benchmark dataset from *scikit learn* library is used to explain one of the main applications of Singular Value Decomposition (SVD) in data science. In the second part, a real-world video clip with constant background is fed to a SVD-based algorithm for video background extraction.

2 Problem Overview

General clustering algorithms (i.e. K-means) are not the best choice to perform clustering when it comes to clusters with irregular shapes. We can achieve a better clustering performance on these types of dataset by using spectral clustering which utilizes SVD. Also, having a powerful too such as SVD gives us the possibility to extract the backgrounds from surveillance cameras, which, in return can be useful in more advanced tasks such as object detection and semantic segmentation.

3 Methodology

3.1 Background

SVD is a factorization technique that decomposes the matrix into three different matrices:

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n},$$

where $V^T V = I_{n\times n}$ and is composed of:

$$V^T = eiginvectors(A^T A)^T = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix},$$

$U^T U = I_{m\times m}$ and is composed of:

$$U = [u_1 \quad u_2 \quad \dots \quad u_m], u_m = \begin{cases} \frac{1}{\sigma_m} A v_m & \text{if } \sigma_m \text{ exists;} \\ \frac{NS(A^T)}{\|NS(A^T)\|} & \text{otherwise.} \end{cases}$$

U is the left singular vectors and V is the right singular vectors. Whereas Σ has singular values and is diagonal matrix:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & & & \end{bmatrix}_{m\times n}, \quad \sigma_n = \sqrt{\lambda_n}.$$

3.2 Datasets

Two types of data are used in this project. First, a clustering benchmark from *scikit learn* is exploited to show the importance of using SVD in clustering applications. Then, a video clip is used to show the extraction of the background.

3.3 Spectral clustering

Clustering is the task of grouping similar objects together. It is an unsupervised machine learning technique. The most common approach for clustering is KMeans. However, this approach only works for globular and convex cluster shapes. If we consider a dataset such as figure 1, KMeans will perform poorly. The result of applying KMeans on this dataset is given in figure 2.

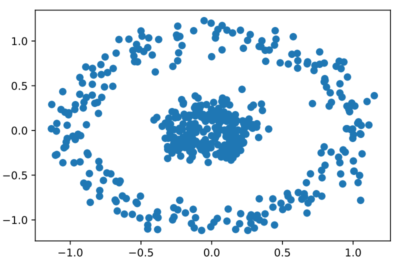

Figure 1: The raw data of the clustering benchmark

To improve the results of pure KMeans, we can use a method named spectral clustering. In practice, spectral clustering is very useful when the structure of the individual clusters is highly non-convex, or more generally when a measure of

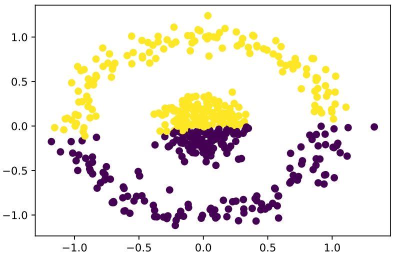

Figure 2: The clusters formed using KMeans algorithm

the center and spread of the cluster is not a suitable description of the complete cluster, such as when clusters are nested circles on the 2D plane. This algorithm is described below:

1. Start with the Affinity matrix (A) or the Adjacency matrix of the data. This represents how similar one object is to another. In a graph, this would represent if an edge existed between the points or not.

2. Find the Degree matrix (D) of each object. This is a diagonal matrix with entry (i,i) equal to the number of objects object i is similar to

3. Find the Laplacian (L) of the Affinity Matrix: L = A − D

4. Find the highest k eigenvectors of the Laplacian Matrix depending on their eigenvalues

5. Run k-means on these eigenvectors to cluster the objects into k classes

As can be seen from this description, before applying KMeans, we extract the **K** eigenvectors from the Laplacian Matrix of the Affinity Matrix, and then perform KMeans. This will improve the clustering performance on the non-globular and non-convex datasets. In order to extract the K most dominant eigenvectors from the Laplacian, we will utilize singular value decomposition (SVD). The results of spectral clustering on the considered benchmark is given in figure 3.

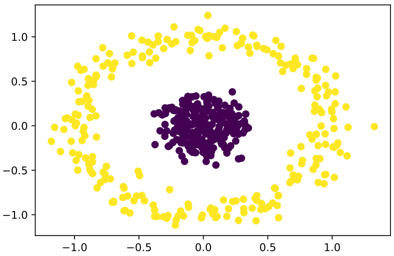

Figure 3: The clusters formed using spectral clustering

3.4 Background removal

Low-rank matrices formed by the help of SVD are actually the approximation of the original matrix. Higher the rank, closer we get to the original matrix. We can obtain a k-rank approximation of A. To do this, select the first k singular values and truncate the 3 matrices accordingly.

Sudden changes usually do not occur in the natural images. Most pixel columns do depend on each other. In a sense, we can use this to our advantage and extract and potentially remove a static background from a video using a low-rank SVD approximation.

The following steps can be implemented to extract the video background: Create matrix M from video – This is done by sampling image snapshots from the video at regular intervals, flattening these image matrices to arrays, and storing them as the columns of matrix M

We can think of M as being the sum of two matrices – one representing the background and other the foreground. The background matrix does not see a variation in pixels and is thus redundant i.e. it does not have a lot of unique information. So, it is a low-rank matrix So, a low-rank approximation of M is the background matrix. We can use SVD to obtain the background matrix. And after that, if we want to obtain the foreground, we can subtract M by this background matrix.

We will use a rank '1' approximation to approximate the backgorund of the video. In order to get rank '1' approximation of the matrix M we need to take u_1, multiply it with v_1, and then increase it by σ_1 (all of which are the outputs of SVD). The final result is the construction of the rank '1' matrix with the highest singular value.

A video, along with its attached background are attached to this report. Also, in figure 4 you can see a frame of the original video, and in figure 5, you can see the extracted background using the rank '1' SVD approximation.

Figure 4: A frame of the original video

Figure 5: The extracted background

4 Conclusion

To sum up, this work proved the usefulness of SVD in practical applications. These applications are mainly due to the SVD's ability of low-rank approximation. The first application was background extraction in video processing. Also, SVD is a great tool to use in unsupervised learning, i.e. clustering. It helps us to separate the clusters using spectral clustering in non-globular, non-convex datasets.