AskPython

Singular Value Decomposition (SVD) in Python

Singular Value Decomposition (SVD) is one of the widely used methods for dimensionality reduction. SVD decomposes a matrix into three other matrices.

If we see matrices as something that causes a linear transformation in the space then with Singular Value Decomposition we decompose a single transformation in
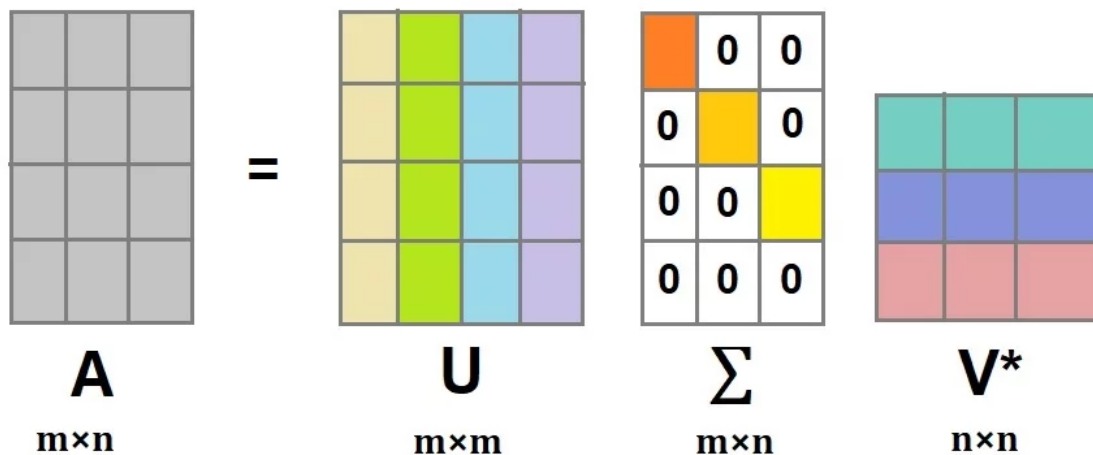
three movements.

in this article we'll see different methods to implement SVD.

Singular Value Decomposition Basics

SVD factors a single matrix into matrix U, D and V* respectively.

$$A = U \Sigma V^*$$

A
m×n

U
m×m

Σ
m×n

V*
n×n

SVD

where,

U and V* are orthogonal matrices.

D is a diagonal matrix of singular values.

The SVD can also be seen as the decomposition of one complex transformation in 3 simpler transformations (rotation, scaling, and rotation).

In terms of transformations

Matrices U and V* causes rotation

Diagonal matrix D causes scaling.

So basically it allows us to express our original matrix as a linear combination of low-rank matrices. Only the first few, singular values are large.

The terms other than the first few can be ignored without losing much information and this is why SVD is referred to as a dimensionality reduction technique.

Implementation of SVD in Python

Let's begin with the implementation of SVD in Python. We'll work with multiple libraries to demonstrate how the implementation will go ahead.

1. Using Numpy

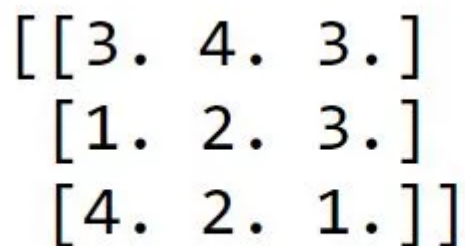Python Numpy having capabilities to implement most Linear Algebra methods offers easy implementation of SVD.

We will use `numpy.linalg` module which has `svd` class to perform SVD on a matrix.

```python
import numpy as np

#Creating a matrix A
A = np.array([[3,4,3],[1,2,3],[4,2,1]])

#Performing SVD
U, D, VT = np.linalg.svd(A)

#Checking if we can remake the original matrix using U,D,VT
A_remake = (U @ np.diag(D) @ VT)
print(A_remake)
```

[[3. 4. 3.]
 [1. 2. 3.]
 [4. 2. 1.]]

Restored Matrix

D is a 1D array instead of a 2D array. D is a diagonal matrix with most of the values ends up being zero, such a matrix is called a **sparse matrix**. to save space it is returned as a 1D array.

2. Using scikit-learn

We will use `TruncatedSVD` class from `sklearn.decomposition` module.

In `TruncatedSVD` we need to specify the number of components we need in our output, so instead of calculating whole decompositions we just calculate the required singular values and trim the rest.

```python
#Importing required modules
import numpy as np
from sklearn.decomposition import TruncatedSVD

#Creating array
A = np.array([[3,4,3],[1,2,3],[4,2,1]])

#Fitting the SVD class
trun_svd =  TruncatedSVD(n_components = 2)
A_transformed = svd.fit_transform(A)

#Printing the transformed matrix
print("Transformed Matrix:")
print(A_transf)
```

```
Transformed Matrix:
[[ 0.31622777  2.21359436]
 [ 2.52982213 -1.26491106]
 [-1.58113883 -1.58113883]]
```

Transformed Matrix

Conclusion

In this article, we saw how we can implement Singular Value Decomposition (SVD) using libraries like Numpy and scikit-learn.

Happy Learning!

Search …   🔍

Recent Posts

5 Easy Ways to Add Rows to a Pandas Dataframe

DataFrame.query() function: How to query pandas DataFrame?

8 Methods to Drop Multiple Columns of a Pandas Dataframe

8 Common Django Commands That You Must Know

Get Head and Tail of a Pandas Dataframe or Series

Turn Index to Column in a Pandas Dataframe

Python Yagmail Module – An easy way to have emails sent!

Memoization in Python – A Brief Introduction

Plot Mathematical Functions – How to Plot Math Functions in Python?

Deploy a Django App on Heroku – An Easy Step-by-Step Guide

Favorite Sites

JournalDev

GoLangDocs

VM-Help

LinuxForDevices

MySQLCode

CodeForGeek