

Comparación de métodos para el cálculo de valores propios de matrices simétricas y aplicaciones en eigenfaces

Brayan Romero

Carlos Nosa

Universidad Nacional de Colombia

brromeroc@unal.edu.co

cnosa@unal.edu.co

Octubre de 2023

Motivación

En el estudio de PCA (principal component analysis) se requiere hallar valores y vectores y propios de una matriz simétrica. El presente proyecto estudia la efectividad de algunos algoritmos conocidos para el cálculo de este tipo de valores y vectores propios.

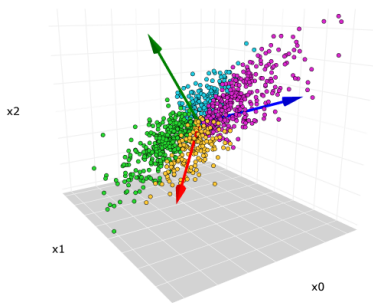


Figura: Análisis de componentes principales. <https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d>

Meta

Mejorar la efectividad en términos de precisión en el cálculo de valores y vectores propios de matrices simétricas y ver las consecuencias que esto implica en el estudio de PCA.

Ruta de acción

- Investigar e implementar algoritmos de cálculo de valores y vectores propios de rutinas (HEEVD, SYEVD de LAPACK Routines) conocidas de ciertas librerías (se mencionaran en la bibliografía).
- Hacer el estudio de PCA a un conjunto de datos(rostros) aplicando cada uno de los métodos de cálculo de valores y vectores propios mencionados previamente.
- Comparar en términos de precisión la efectividad de cada uno de los métodos en el análisis de componentes principales de un conjunto de datos.

Referencias

A continuación se muestran las referencias usadas para conocer e implementar el estudio de PCA en Eigenfaces.

- 1 Eigenfaces — Face Classification in Python:
<https://towardsdatascience.com/eigenfaces-face-classification-in-python-7b8d2af3d3ea>
- 2 Github-Face Classification in Python:
<https://github.com/daradecic/Python-Eigenfaces/blob/master/EigenFaces.ipynb>
- 3 OpenCV Eigenfaces for Face Recognition:
<https://pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/>

Referencias

Las siguientes referencias se usan para saber la implementación de ciertos métodos en Python.

- 4 Ordenar valores propios y vectores propios en python
<https://stackoverflow.com/questions/8092920/sort-eigenvalues-and-associated-eigenvectors-after-using-numpy-linalg-eig-in-pyt>
- 5 Eigenvalues grandes: <https://towardsdatascience.com/eigenvalues-and-eigenvectors-378e851bf372>
- 6 Importing Image Data into NumPy Arrays:
<https://www.pluralsight.com/guides/importing-image-data-into-numpy-arrays>

Referencias

Estar referencias aluden a las rutinas utilizadas en algunos paquetes.

- 7 LAPACK Routines—oneAPI Specification 1.2-rev-1 documentation. (s. f.). Recuperado 8 de octubre de 2023, de <https://spec.oneapi.io/versions/latest/elements/oneMKL/source/domain/lapack-singular-value-eigenvalue-routines>
- 8 Linalg.py. (s. f.). GitHub—Numpy/Numpy at v1.26.0. (s. f.). GitHub. <https://github.com/numpy/numpy/blob/v1.26.0/numpy/linalg/linalg.py#L1490>
- 9 Cuppen, J. J. M. (1980). A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36(2), 177-195. <https://doi.org/10.1007/BF01396757>

Referencias

Algunas referencias teóricas son

- 10 Golub, G. H., & Loan, C. F. V. (1996). Matrix computations. JHU Press.