grpca: A Package for Fast Principal Component Analysis with GPU Acceleration

Rafael S. de Souza^{a,*}, Xu Quanfeng^a, Shiyin Shen^{a,b}, Chen Peng^{a,c}, Zihao Mu^a

Abstract

We present qrpca, a fast and scalable QR-decomposition principal component analysis prompers. In both R and python languages, makes use of torch for internal matrix computations, and enables GPU acceleration, when available, qrpca provides similar functionalities to prcomp (R) and sklearn (python) packages respectively. A benchmark test shows that qrpca can achieve computational speeds 10-20 × faster for large dimensional matrices than default implementations, and is at least twice as fast for a standard decomposition of spectral data cubes. The qrpca source code is made freely available to the community.

Keywords: Principal component analysis; Astroinformatics; GPU computing**

1. Introduction

Principal component analysis (PCA; Pearson, 1901) stands out as a prime method for dimensionality reduction and data exploration (see Jolliffe and Cadima, 2016, for a review). It compresses a dataset while preserving as much variability as possible. Given the original matrix comprising orthogonal variables, which are linear combinations of the original variables, which are linear combinations. The best of our knowledge, we present the first public package for QR-decomposition We present qrpca, a fast and scalable QR-decomposition principal component analysis package. The software, written

Despite its broad applicability, SVD PCA implementations are computationally costly for high dimensional matrices¹. This limitation triggered the development of

such as proomp in R, and sklearn in python.

The remainder of the paper is organized as follows. In Section 2 we provide an overview of the SVD PCA and QR-decomposition PCA. Section 3 shows a speed performance test alongside a practical application example to integral field unit (IFU) spectroscopy. Finally, in Section 4 we present our concluding remarks.

^aKey Laboratory for Research in Galaxies and Cosmology Shanghai Astronomical Observatory Chinese Academy of Sciences 80 Nandan Rd. Shanghai 200030 China

^bKey Lab for Astrophysics Shanghai 200034 People's Republic of China ^cShanghai Institute of Technology 100 Haiquan Rd. Shanghai 201418 China

^{*}Corresponding author

Email address: drsouza@shao.ac.cn (Rafael S. de Souza)

¹Throughout the paper, we will refer to data size as the number of rows, and dimension as the number of columns.

2. Methodology

Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be a rectangular matrix composed by n rows by m columns. The SVD decomposition of \mathbf{X} is given by:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}.\tag{1}$$

Here, $\mathbf{S} \in \mathbb{R}^{m \times m}$ is a diagonal matrix, and $\Phi = \mathbf{US}$ gives the PCA projection. We show at Algorithm 1 a pseudocode for a SVD PCA. If the dimensionality of \mathbf{X} is large, then the computation of the eigenvalues will be time consuming, and may cause memory overflow (e.g. Adnan et al., 2021).

A faster alternative is to use an intermediate step as suggested by Sharma et al. (2013). The procedure consists in first factorize \mathbf{X} into an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times m}$, and an upper triangular matrix $\mathbf{R} \in \mathbb{R}^{m \times m}$. The SVD decomposition of $\mathbf{R}^{\mathsf{T}} = \mathbf{U}_{\star} \mathbf{S}_{\star} \mathbf{V}_{\star}^{\mathsf{T}}$ then yields the same diagonal matrix $\mathbf{S}_{\star} \equiv \mathbf{S}$ of \mathbf{X} , and an equivalent PCA transform $\mathbf{QVS} \equiv \mathbf{US}$. The computational advantage comes from the intermediate \mathbf{QR} decomposition, which enables running SVD factorization on the upper triangular matrix \mathbf{R} to compute eigenvectors instead of running SVD directly on \mathbf{X} . See Algorithm 2 for a pseudo-code for the case of QR-decomposition PCA.

Algorithm 1 SVD PCA

Require: Input matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$

1: Compute SVD on **X**

2: $\mathbf{U} \in \mathbb{R}^{n \times m}$ Orthogonal matrix

3: $\mathbf{S} \in \mathbb{R}^{m \times m}$ Rectangular diagonal matrix

4: $\mathbf{V} \in \mathbb{R}^{m \times m}$ Orthogonal matrix

5: Compute PCA transform $\Phi = \mathbf{US}$

Algorithm 2 QR-decomposition PCA

Require: Input matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$

1: Compute **QR** decomposition on **X**

2: $\mathbf{Q} \in \mathbb{R}^{n \times m}$ Orthogonal matrix

3: $\mathbf{R} \in \mathbb{R}^{m \times m}$ Upper triangular matrix

4: Compute SVD on $\mathbf{R}^{\mathsf{T}} \in \mathbb{R}^{m \times m}$

5: $\mathbf{S} \in \mathbb{R}^{m \times m}$

6: $\mathbf{V} \in \mathbb{R}^{m \times m}$

7: Compute PCA transform $\Phi = \mathbf{QVS}$

3. Performance evaluation

In this section, we provide a simple test of the qrpca computational performance on simulated data and an ex-

ample application to the analysis of Astronomical IFU spectra.

3.1. Simulated data

To evaluate the performance of grpca, we create a collection of random matrices varying the rows and column sizes. For a fixed dimension of m = 1000, we vary the data size from $n = 10^2 - 10^6$, and for a fixed data size of n = 1,000, we varied the dimensions from $m = 10^2 - 10^5$. The mock data is sampled from a normal zero mean and unity variance distribution. Fig. 1 shows the speedup gain for a range of dataset sizes and dimensions. Left panels show the comparison between the python version of grpca and default PCA implementation of sklearn, while the right panels show the comparison between the R implementation of grpca and prcomp. Visual inspection on Fig. 1 shows that grpca consistently perform faster than their related counterparts for datasets with more than 1,000 dimensions, with a top performance at least 15× faster for matrices with 10,000 dimensions and GPU enabled. sklearn shows competitive performance against the python qrpca for high-dimensional data, while prcomp performs well on moderately large datasets, and the main advantage of using grpca comes from the GPU on these cases. Overall, grpca can be particularly beneficial for analyzing spectral data-cubes and hyperspectral images, and the next section shows one practical application in analyzing Astronomical spectra.

3.2. MaNGA data

Here we show one astronomical application using the IFU spectral data from the Mapping Nearby Galaxies at Apache Point Observatory (MaNGA; Bundy et al., 2015) survey. MaNGA is a program of the Sloan Digital Sky Survey IV (SDSS-IV; Blanton et al., 2017), and also is the largest IFU survey of nearby galaxies to date. MaNGA obtains integrated field spectroscopy of galaxies using customdesigned fiber bundles, where the buffered fibers have a core diameter of 2 arcsec. Depending on galaxies' size, MaNGA uses different IFUs, where the IFU size range is from 19 to 127 fibers, and the corresponding field of view is from 12 to 32 arcsec in diameter. The wavelength coverage of MaNGA spectra is 3622-10354 Å, and the resolution is about R \sim 2000. For each galaxy target, MaNGA takes three dithered exposures. By stacking the spectra from dithered exposures, MaNGA builds a data cube $(N_X * N_Y * N_{wave})$ for each target galaxy, where the spatial pixel scale is designed as 0.5 arcsec per pixel (Law et al., 2016). For the largest MaNGA IFU, the data cube

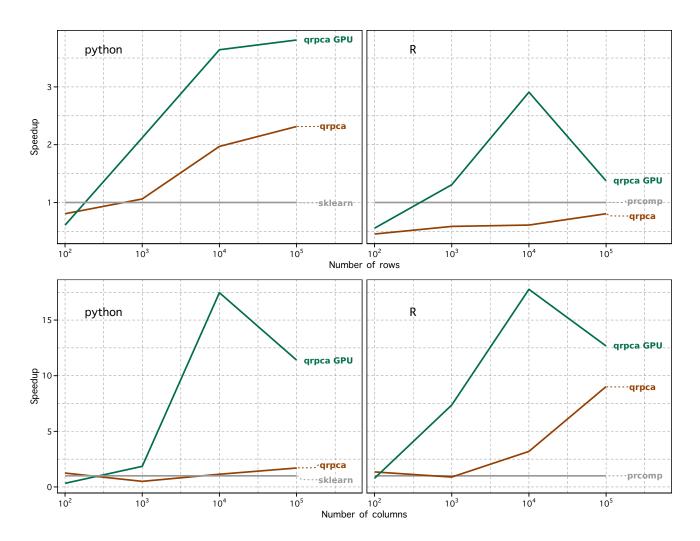


Figure 1: A speedup performance evaluation between <code>qrpca</code> and major SVD PCA implementations, namely <code>sklearn</code> and <code>prcomp</code>. Each line represents the speedup time normalized by the running time of <code>sklearn</code> (left panels), and <code>prcomp</code> (right panels). The benchmark was executed in a machine with the following specifications: CPU – 2.2GHz Intel Xeon Silver 4210; GPU – NVIDIA A40 (48GB); OS – Ubuntu Linux 18.04 64 bits; RAM – 188 GB.

has $N_X * N_Y = 74 * 74$ spaxels. For the wavelength channel, the MaNGA data cube has $N_{wave} = 4573$ for logarithmically sampled data and $N_{wave} = 6732$ for the linear sampling (Law et al., 2016).

We now show a simple task of computing the first eigenmaps of the MaNGA IFU data. PCA decomposition of data-cubes is particularly interesting to disentangle uncorrelated physical phenomena in the galaxy. For example, it has been used to identify a broad line region of a previously unknown active galactic nucleus in the galaxy NGC 4736 (Steiner et al., 2009). We showcase our approach by decomposing the IFU data of the galaxy merger Mrk 848 (MaNGA ID:12-193481). Mrk 848 is a major merger with strong interaction between the two galaxies with estimated stellar masses $\log(M_{\star}/M_{\odot}) = 10.44$ and 10.30 (Yang et al., 2007) at z = 0.041. The MaNGA cube

consists of an array of 74 * 74 * 4563 array.

To decompose the data cube, we follow three basic steps. The first step involves transforming the tensor into a 5476*4563 matrix where each row represents one spectrum and a wavelength for each column. Then we apply a PCA transform to this matrix, which yields a matrix of a similar dimension, where each column now represents a PC. The final step is to transform back to the original format, and each PC will now represent an eigenmap. We show code snippets to read, process, and visualize the first eigenmaps with R and python versions of qrpca at Appendix A. The computation time with qrpca is at least 2-3 times faster than standard implementations.

Figure 2 shows the first four PCs, where we can see different aspects of the merger structure. The first PC correlates with the overall merger structure, including the

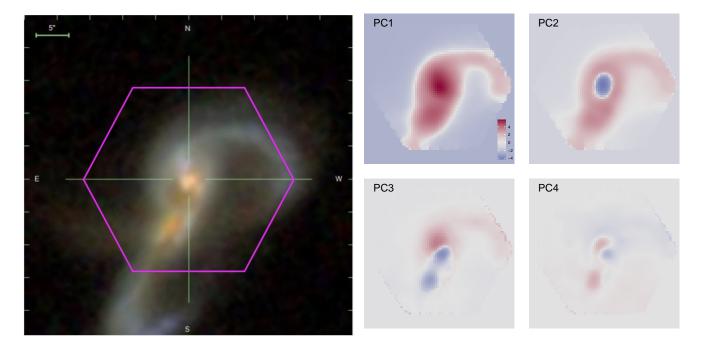


Figure 2: Lef panel: The SDSS cutout of the galaxy Mrk 848 (MaNGA ID:12-193481) with a purple hexagon denoting the approximate spatial grasp of the IFU. Right panel display the first 4 eigenmaps, where different aspects of the merger structure can be seen.

core and tail regions. The second PC isolates the core region of the central galaxy, while the third and fourth PCs discriminate the star-forming regions triggered by the merging process (Yuan et al., 2018). The galactic structures illuminated by this simple decomposition are broadly consistent with the different aged stellar populations revealed by detailed spectral energy distribution fitting (Yuan et al., 2018), but further scrutiny of this object is beyond the scope of this work.

4. Conclusions

PCA is an essential tool for multivariate data analysis. However, its standard implementation does not scale well for high-dimensional datasets. In this paper, we present qrpca, a package for fast PCA computation based on QR decomposition. The code enables GPU acceleration when available. We showcase experiments on both simulated and real datasets of varying dimensions. Experimental results show that our package can perform more than $10 \times$ faster than conventional approaches, depending on the matrix dimensions.

qrpca is written in both R and Python and is freely available at $GitHub^2$, $Zenodo^3$ and listed in the Python Package $Index^4$.

Acknowledgments

We thank Ana L. Chies-Santos for her insightful suggestions while preparing this manuscript. We thank Emille E. O. Ishida for the final revision of this manuscript. This work was supported by the National Natural Science Foundation of China (No. 1201101284, 12073059), the National Key R&D Program of China (No. 2019YFA0405501), and the China Manned Space Project (No. CMS-CSST-2021-A04).

Appendix A. Code Snippets

Here, we show how to perform a PCA on MaNGA decomposition using the qrpca package. The IFU data used in our example is available at https://data.sdss.org/sas/dr17/manga/spectro/redux/v3_1_1/7443/stack/manga-7443-12703-LOGCUBE.fits.gz

R code for grpca computation on MaNGA

```
require(qrpca); require(reticulate)
require(FITSio); require(ggplot2);
require(dplyr); require(reshape2)
cube <- "manga-7443-12703-LOGCUBE.fits"
df <- readFITS(cube)
n_row <- dim(df$imDat)[1]
n_col <- dim(df$imDat)[2]
n_wave <- dim(df$imDat)[3]</pre>
```

²https://github.com/RafaelSdeSouza/qrpca

³https://doi.org/10.5281/zenodo.6556360

⁴https://pypi.org/project/qrpca/

```
9 data.2D <- array_reshape(df$imDat,</pre>
10 c(n_row*n_col,n_wave),order = c("F"))
pca <- qrpca(data.2D)</pre>
12 # Function to extract the k-th eigenmap
  eigenmap <- function(pcobj, k = 1) {
13
14
    x <- as.matrix(pcobj$x)
    out <- matrix(x[,k],nrow=n_row,ncol=n_col)</pre>
16
    out.
17 }
18
map1 <- eigenmap(pca) %>% melt()
20
 ggplot (map1, aes (x=Var1, y=Var2, z=value)) +
21
    geom_raster(aes(fill=value)) +
    scale_fill_viridis_c(option="C") +
23
    theme(legend.position = "none")
```

Python code for grpca computation on MaNGA

```
from astropy.io import fits
2 from astropy.wcs import WCS
3 import torch
4 import numpy as np
5 from qrpca.decomposition import qrpca
6 import matplotlib.pyplot as plt
  data = fits.open("manga-7443-12703-LOGCUBE.
      fits")
9 dat = data[1].data.transpose(1,2,0)
wcs = WCS(data[1].header)
  da = dat.reshape(-1, dat.shape[-1]).astype(np.
      float32)
  device = torch.device("cuda:0" if torch.cuda.
      is_available() else "cpu")
  pca = qrpca(n_component_ratio=1, device=device)
14 map1 = pca.fit_transform(da)
15 ax = plt.subplot(projection=wcs[0,:,:])
ax = plt.qca()
17 lon = ax.coords['ra']
18 lat = ax.coords['dec']
19 plt.imshow(map1.reshape(74,74))
20 plt.xlabel("RA [deg]")
21 plt.ylabel("DEC [deg]")
22 plt.show()
```

References

- Adnan, T.M.T., Tanjim, M.M., Adnan, M.A., 2021. Fast, scalable and geo-distributed pca for big data analytics. Information Systems 98, 101710. URL: https://www.sciencedirect.com/science/article/pii/S0306437920301526, doi:https://doi.org/10.1016/j.is.2020.101710.
- Amara, A., Quanz, S.P., 2012. PYNPOINT: an image processing package for finding exoplanets. MNRAS 427, 948–955. doi:10.1111/j.1365-2966.2012.21918.x, arXiv:1207.6637.
- Battulga, L., Lee, S.H., Nasridinov, A., Yoo, K.H., 2020. Hash-tree pca: Accelerating pca with hash-based grouping. J. Supercomput. 76, 8248-8264. URL: https://doi.org/10.1007/s11227-019-02947-x, doi:10.1007/s11227-019-02947-x.

- Blanton, M.R., Bershady, M.A., Abolfathi, B., Albareti, F.D., Allende Prieto, C., Almeida, A., Alonso-García, J., Anders, F., Anderson, S.F., Andrews, B., et al., 2017. Sloan Digital Sky Survey IV: Mapping the Milky Way, Nearby Galaxies, and the Distant Universe. AJ 154, 28. doi:10.3847/1538-3881/aa7567, arXiv:1703.00052.
- Bundy, K., Bershady, M.A., Law, D.R., Yan, R., Drory, N., MacDonald, N., Wake, D.A., Cherinka, B., Sánchez-Gallego, J.R., Weijmans, A.M., Thomas, D., Tremonti, C., Masters, K., Coccato, L., Diamond-Stanic, A.M., Aragón-Salamanca, A., Avila-Reese, V., Badenes, C., Falcón-Barroso, J., Belfiore, F., Bizyaev, D., Blanc, G.A., Bland-Hawthorn, J., Blanton, M.R., Brownstein, J.R., Byler, N., Cappellari, M., Conroy, C., Dutton, A.A., Emsellem, E., Etherington, J., Frinchaboy, P.M., Fu, H., Gunn, J.E., Harding, P., Johnston, E.J., Kauffmann, G., Kinemuchi, K., Klaene, M.A., Knapen, J.H., Leauthaud, A., Li, C., Lin, L., Maiolino, R., Malanushenko, V., Malanushenko, E., Mao, S., Maraston, C., McDermid, R.M., Merrifield, M.R., Nichol, R.C., Oravetz, D., Pan, K., Parejko, J.K., Sanchez, S.F., Schlegel, D., Simmons, A., Steele, O., Steinmetz, M., Thanjavur, K., Thompson, B.A., Tinker, J.L., van den Bosch, R.C.E., Westfall, K.B., Wilkinson, D., Wright, S., Xiao, T., Zhang, K., 2015. Overview of the SDSS-IV MaNGA Survey: Mapping nearby Galaxies at Apache Point Observatory. ApJ 798, 7. doi:10.1088/0004-637X/798/1/7, arXiv:1412.1482.
- Falbel, D., Luraschi, J., 2022. torch: Tensors and Neural Networks with 'GPU' Acceleration. Https://torch.mlverse.org/docs, https://github.com/mlverse/torch.
- Fan, J., Sun, Q., Zhou, W.X., Zhu, Z., 2018. Principal Component Analysis for Big Data. John Wiley & Sons, Ltd. pp. 1–13. doi:10.1002/9781118445112.stat08122.
- Ishida, E.E.O., de Souza, R.S., 2013. Kernel PCA for Type Ia supernovae photometric classification. MNRAS 430, 509–532. doi:10.1093/mnras/sts650, arXiv:1201.6676.
- Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: a review and recent developments. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 374, 20150202. doi:10.1098/rsta.2015.0202.
- Kiefer, S., Bohn, A.J., Quanz, S.P., Kenworthy, M., Stolker, T., 2021. Spectral and angular differential imaging with SPHERE/IFS. Assessing the performance of various PCA-based approaches to PSF subtraction. A&A 652, A33. doi:10.1051/0004-6361/202140285, arXiv:2106.05278.
- Law, D.R., Cherinka, B., Yan, R., Andrews, B.H., Bershady, M.A., Bizyaev, D., Blanc, G.A., Blanton, M.R., Bolton, A.S., Brownstein, J.R., Bundy, K., Chen, Y., Drory, N., D'Souza, R., Fu, H., Jones, A., Kauffmann, G., MacDonald, N., Masters, K.L., Newman, J.A., Parejko, J.K., Sánchez-Gallego, J.R., Sánchez, S.F., Schlegel, D.J., Thomas, D., Wake, D.A., Weijmans, A.M., Westfall, K.B., Zhang, K., 2016. THE DATA REDUCTION PIPELINE FOR THE SDSS-IV MaNGA IFU GALAXY SURVEY. The Astronomical Journal 152, 83. URL: https://doi.org/10.3847/0004-6256/152/4/83, doi:10.3847/0004-6256/152/4/83.
- Lazcano, R., Madroñal, D., Salvador, R., Desnos, K., Pelcat, M., Guerra, R., Fabelo, H., Ortega, S., Lopez, S., Callico, G., Juarez, E., Sanz, C., 2017. Porting a pca-based hyperspectral image dimensionality reduction algorithm for brain cancer detection on a many-core architecture. Journal of Systems Architecture 77, 101–111. URL: https://www.sciencedirect.com/science/article/pii/S1383762116302934, doi:https://doi.org/10.1016/j.sysarc.2017.05.001.
- Nie, L., Li, G., Peterson, J.R., Wei, C., 2021. The point spread function reconstruction II. The smooth PCA. MNRAS 503, 4436–4445. doi:10.1093/mnras/stab733.

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library, Curran Associates, Inc.
- Pearson, K., 1901. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2, 559–572. doi:10.1080/14786440109462720.
- Sharma, A., Paliwal, K.K., Imoto, S., Miyano, S., 2013. Principal component analysis using qr decomposition. International Journal of Machine Learning and Cybernetics 4, 679–683. URL: https://doi.org/10.1007/s13042-012-0131-7, doi:10.1007/s13042-012-0131-7.
- de Souza, R.S., Maio, U., Biffi, V., Ciardi, B., 2014. Robust PCA and MIC statistics of baryons in early minihaloes. MNRAS 440, 240–248. doi:10.1093/mnras/stu274.
- Steiner, J.E., Menezes, R.B., Ricci, T.V., Oliveira, A.S., 2009. PCA Tomography: how to extract information from data cubes. Monthly Notices of the Royal Astronomical Society 395, 64–75. doi:10.1111/j.1365-2966.2009.14530.x.
- Vogt, F., Tacke, M., 2001. Fast principal component analysis of large data sets. Chemometrics and Intelligent Laboratory Systems 59, 1–18. URL: https://www.sciencedirect.com/science/article/pii/S0169743901001307, doi:https://doi.org/10.1016/S0169-7439(01)00130-7.
- Yang, X., Mo, H.J., van den Bosch, F.C., Pasquali, A., Li, C., Barden, M., 2007. Galaxy Groups in the SDSS DR4. I. The Catalog and Basic Properties. ApJ 671, 153–170. doi:10.1086/522027, arXiv:0707.4640.
- Yohana, E., Ma, Y.Z., Li, D., Chen, X., Dai, W.M., 2021. Recovering the 21-cm signal from simulated FAST intensity maps. MNRAS 504, 5231–5243. doi:10.1093/mnras/stab1197, arXiv:2104.10937.
- Yuan, F.T., Argudo-Fernández, M., Shen, S., Hao, L., Jiang, C., Yin, J., Boquien, M., Lin, L., 2018. Spatially resolved star formation and dust attenuation in Mrk 848: Comparison of the integral field spectra and the UV-to-IR SED. A&A 613, A13. doi:10.1051/0004-6361/201731865, arXiv:1801.04860.