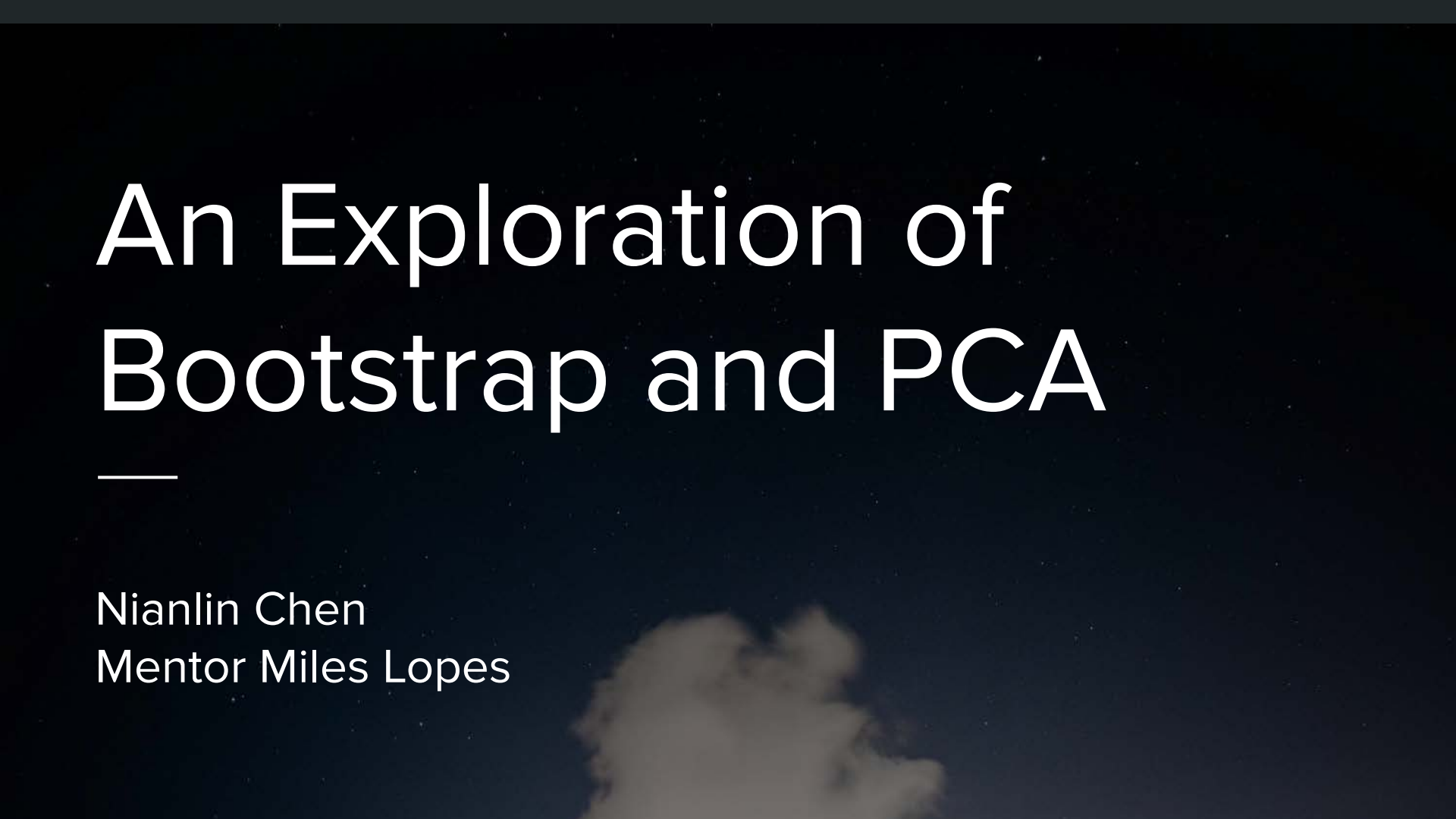
An Exploration of Bootstrap and PCA

Nianlin Chen
Mentor Miles Lopes



What is the research about?

PCA method helps to extract important information using the eigenvalues and eigenvectors of a sample covariance matrix.

It is important to construct a confidence interval for the unknown eigenvalues of the true covariance matrix.

A possible way to construct such an interval is via bootstrap methods.

In this project, we explore how well bootstrap CI works for solving this problem when the data are high-dimensional.

What is Principal Component Analysis?

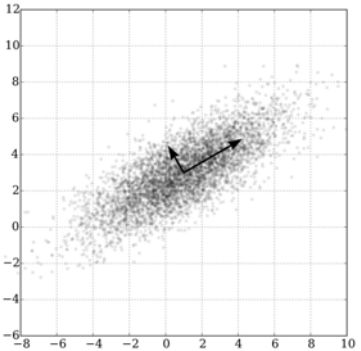
Principal Component Analysis (PCA) is a technique to reduce the dimensionality of a dataset, by projecting the data into special directions called “principal components”.

It is often possible to capture most of the information in the data using only a small number of these directions.

Basic idea of PCA

The principal components of a dataset are the eigenvectors of the sample covariance matrix S = (1/n)(Y1Y1'+...+YnYn'), where Yi are the centered data vectors.

In the figure below, the arrows display the top two principal components for the dataset displayed in the scatterplot.

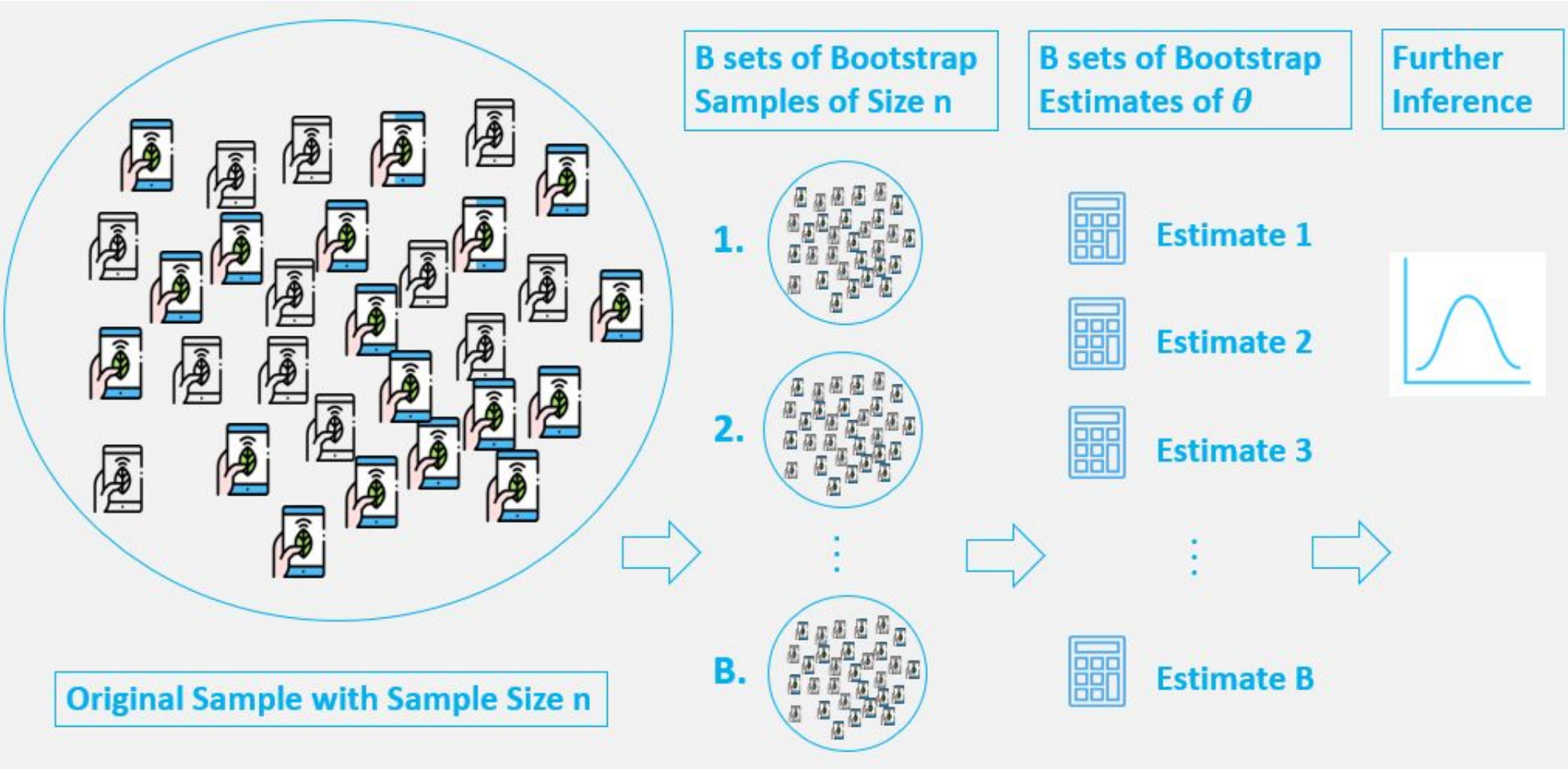


What is bootstrap method?

Bootstrap is a statistical method that can be used to construct confidence intervals by random sampling with replacement.

In order to construct a confidence interval for an unknown parameter, it is necessary to know how far the estimator is likely to deviate from the parameter.

The basic idea of the bootstrap is to look at how much the estimator fluctuates when in it is computed on resampled versions of the original data.



https://towardsdatascience.com/an-introduction-to-the-bootstrap-method-58bcb51b4d60

Research: Approaches using bootstrap

Bootstrap sample is randomly taking from original sample with size N with replacement. I.e. resample 5 times from [1,2,3,4,5] to get [2,2,3,4,5].

Construct bootstrap confidence interval

- Resample dataset a number of times (B) and find maximum eigenvalue λi of each covariance matrix of resampled dataset i.

```
for i = 1:B
    bootindices = randsample(n,n,true); % resample original dataset
    Xboot = X(bootindices,:);
    Xbarboot = mean(Xboot); % find the mean of the resample dataset
    sigmahatboot = 1/n*Xboot'*Xboot-Xbarboot*Xbarboot';% find sample coverance matrix
    eigenvalues = eig(sigmahatboot); % get the eigenvalues
    lamdahatboot(i) = max(abs(eigenvalues)); % draw the maximun eigenvalue
end
```

- Calculate the difference of the true eigenvalue and bootstrap eigenvalue (d)
- Find the confidence interval. i.e. for 95% confidence interval, find a=0.025 percentile and b=0.975 percentile of d. The confidence interval should be (λtrue-b, λtrue+a).

My simulation

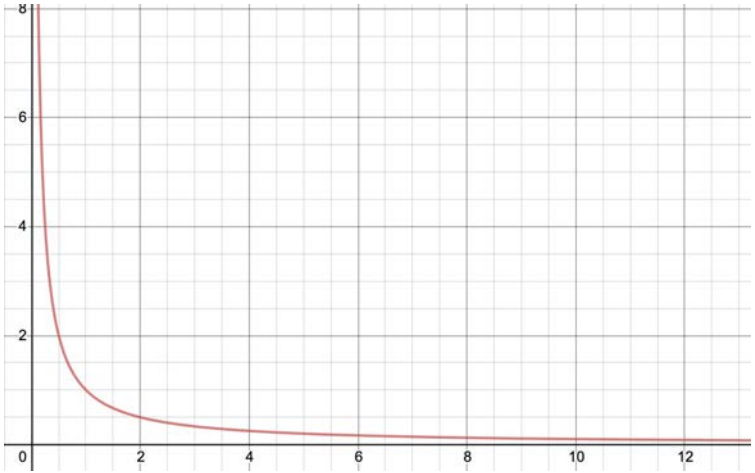
How to test the accuracy of the bootstrap confidence interval we got using this algorithm?

- Generate a collection of M different datasets.
- For each of the M datasets, run the bootstrap method to construct a 95% confidence interval for the true eigenvalue.
- Return 1 in if the true eigenvalue is included in the interval.
- Otherwise, return 0.
- Then check if the proportion of 1’s in the M trials is around 0.95.

Research: Decay of eigenvalues

To further understand the bootstrap confidence interval, we try different covariance matrix of a dataset. Hence, we consider the case of decay of eigenvalues, e.g. $\lambda_j = j^{(-\beta)}$, for $\beta>0$. For example, when $\beta=1$, we have $\lambda_1 = 1$; $\lambda_2 = ½$; $\lambda_3 = ⅓$... and so on.

We are interested to understand how the performance of the bootstrap is related to β.



Decay of eigenvalue for β=1

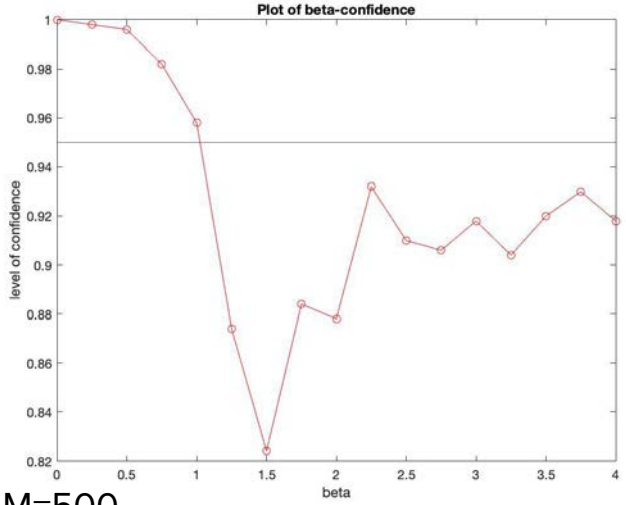
Research: Decay of eigenvalues

To investigate the role of β, we keep other parameters fixed and consider a grid of β values, 0, .25, 0.5,....,

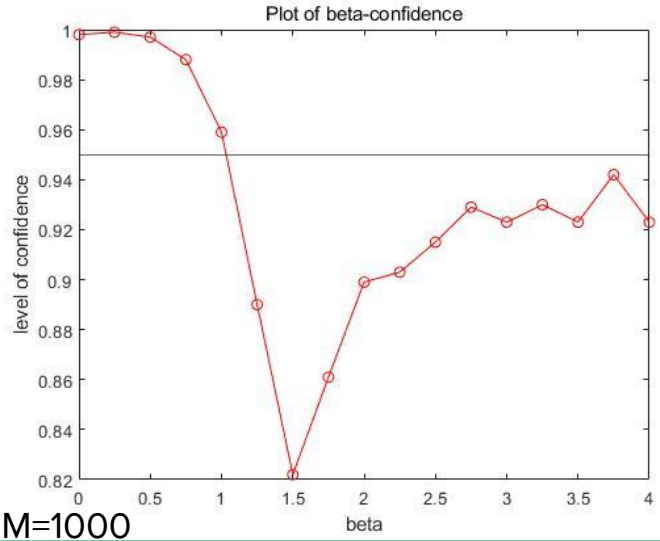
For each β, we run a simulation to find the actual coverage probability of the bootstrap intervals.

Our intuition is that large β should make things easier for the bootstrap, and small β should make things harder.

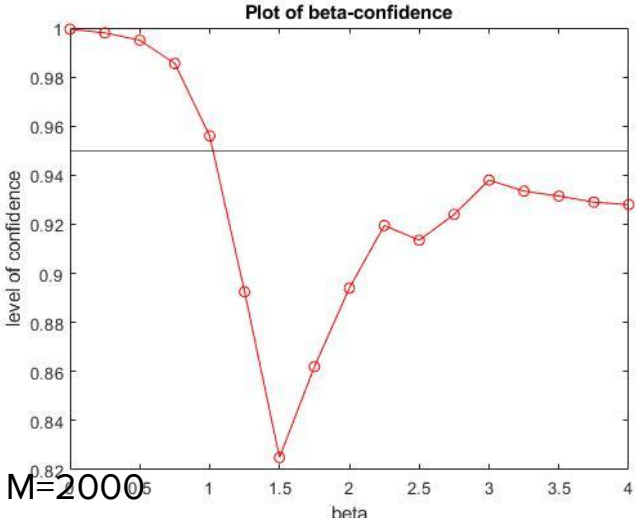
Let’s see what happened...



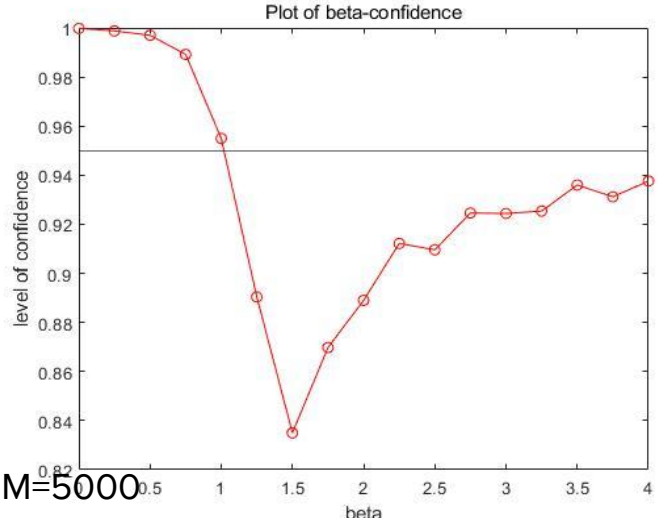
M=500



M=1000



M=2000



M=5000

Each experiments are with n = p = 100, $\beta = 0 : 0.25 : 4$

Conclusion

From the results graphs, a few things are noticed:

- The curves become nicer and smoother as # of simulations increasing.
- In general, the curve exhibits an overall growth as β growthing. As the β is larger, the performance of the bootstrap become nicer.
- The curves show very high coverage of bootstrap confidence interval (about 100%) at the beginning (β around 0, λ=1), then drops rapidly at β=1.5 and then rises slowly as β increasing.

$$\sum_{j=1}^{\infty} \frac{1}{j^{0.5}} \text{ (sum diverges) } \sum_{j=1}^{\infty} \frac{1}{j^{1.5}} = 2.61238$$

Problems and challenges encountered during the investigating

To get a more precise results, it is better to choose a larger n, or try an even larger M. However, it is difficult to achieve by using my computer because of a huge consumption of time.

- Cloud computing

Why cloud computing is related to Bootstrap

Using cloud computer can help save lots of time. It work to analysis bootstrap confidence interval because the workload can be broken into several small jobs and work simultaneously, and the aggregate of those results of small jobs is the same as running the entire work. And by using cloud computing, each small pieces will be assigned to different computers.

Figure 1: Cloud Computing

- Galileo, which is a free online cloud computing station I tried. It works to run R code.

Thanks for watching