Fall 2023 Final Project Guideline

Yeichan Kim
Expanded Statistics Club, Yonsei University

October 31, 2023

1 Introduction

Greetings esteemed members of Expanded Statistics Club!

It brings us immense pride to witness the dedication and enthusiasm with which you approach the field of statistics and data science. As we approach the semester's culmination, it's time to demonstrate our understanding, application, and critical analysis, particularly in multivariate analysis, which has been the topic of our interest.

Our final projects aim to assess your knowledge while also sharpening your research and statistical discourse skills. This endeavor allows you to dive deep into academic papers and data analysis projects, emphasizing understanding nuances, asking pertinent questions, and crafting narratives rooted in collective intelligence.

In this guideline, you will find a detailed breakdown of the four distinct project categories. **Two of these will involve reviewing intricate academic papers on estimating the number of factors in factor analysis**, where your skills in critical analysis, interpretation, and statistical computation will be put to the test. **The other two projects will focus on advanced methods that are not (or only partially) covered in lecture our sessions: t-SNE and Spectral Clustering**; you will be challenged to explain one of these methods and implement it to interpret, visualize, and draw conclusions from a given dataset.

Embrace this project as an opportunity to showcase your prowess, dedication, and communication skills. **And most importantly, engage with your teammates and share your insights openly.**

Wishing you all the best in this intellectual journey!

Best,
Yeichan

2 Paper Review: Estimating Number of Factors by Adjusted Eigenvalues Thresholding

A number of literatures discuss estimating the number of factors, including, but not limited to, the following: Bai and Ng (2002) [1]; Onatski (2010) [2]; Lam and Yao (2012) [3]; Ahn and Horenstein (2013) [4]. These methods are all based on the eigenvalues of the covariance matrix; however, it does not take into account the scales of the observed variables. In fact, we can show that under some mild conditions imposed on the population covariance matrix, these methods consistently overestimate K, the true number of factors (i.e., $P(\hat{K} \geq K + 1) \to 1$).

In order to address this issue, Fan et al. (2019) [5] exploit the structure of the correlation matrix to estimate the number of common factors, which contribute (factor loading) to more than one observed variable y_j. Under the conditions present in this paper, its first theorem gives a sufficient condition to ensure that the number of the eigenvalues of $\mathbf{R}$, the population correlation matrix, greater than 1 is equal to the number of common factors. That is, 1 can play a role as a "threshold" for estimating K using the sample correlation matrix.

With 1 being a baseline threshold, there is an additional term $\sqrt{\rho}$ such that $p/n \to \rho \in (0, \infty)$ for the bias correction of sample eigenvalues. Thus, we have $1 + \sqrt{\rho}$ as the adjusted correlation thresholding (ACT), and its consistency for estimating K is proved. Additionally, some case study results based on Monte Carlo simulations follow.

The above is the basic sketch of this paper as I understand it so far. You can access this paper at `https://arxiv.org/abs/1909.10710`. Here are some bits of advice for your project:

- This paper estimates the number of factors under a high-dimensional factor model, which is new to most of us. It might be beneficial to begin by highlighting the discrepancies between low-dimensional and high-dimensional statistics. Notably, in the former, $p/n \to 0$ as $n \to \infty$, whereas the convergence to zero does not hold in the latter since p is comparable to n in a high-dimensional setup.

- Don't be obsessed with every single proof and mathematical detail. Rather, focus on the flow and big picture of this paper and try to convey the message when presenting. While the above sketch may not be entirely accurate, it should serve as a guide to help you navigate the main paragraphs.

- **I strongly recommend you reproduce the simulation results.** The R source code is available for download on the aforementioned website. To fully understand this paper, it's crucial to comprehend the algorithm as presented in the code. Additionally, it would be fascinating to observe how the estimation performance evolves with your unique data-generating process. Should you have any questions or need guidance on this, please feel free to consult with me. You may ignore the section "Empirical Studies."

Good luck, and enjoy the challenge with your teammates!

3 Paper Review: Remarks on Parallel Analysis

Parallel Analysis (Horn (1965) [6], Buja and Eyuboglu (1992) [7]) is one of the most popular methods for selecting the number of factors. In the widely used permutation-based version proposed by Buja and Eyuboglu (1992), we start with the $n \times p$ data matrix $\mathbf{X}$ and generate a matrix $\mathbf{X}_\pi$ by permuting the entries in each column of $\mathbf{X}$ separately. We repeat this procedure several times, producing multiple permuted matrices. Then, we select the first factor if the top singular value of $\mathbf{X}$ is larger than a fixed percentile of the top singular values of the permuted matrices. A typical choice is 95th percentile. If the first factor is selected, then we repeat the same procedure for the second largest singular value of $\mathbf{X}$, comparing with the second singular values of permuted matrices, and so on. We stop when a factor is not selected.

Despite the empirical success of Parallel Analysis, there was essentially no theoretical justification for its effectiveness. However, this changed with Dobriban (2020) [8], which demonstrated that the permutation method consistently retains factors in certain high-dimensional models. The intuition is that permutations keep the noise invariant, while "destroying" the low-rank signal. This provides justification for permutation methods. His work also uncovers the drawbacks of permutation methods and paves the way for improvements.

Dobriban's work is pivotal as it provides a theoretical guarantee for Parallel Analysis. However, considering the mathematical rigor of both papers you're set to review, I'd prefer to set it aside for now. Instead, I'd recommend delving into Buja and Eyuboglu (1992) [7], which introduced the permutation-based method for the first time and is comparatively more straightforward. Their paper is accessible at `https://www.tandfonline.com/doi/epdf/10.1207/s15327906mbr2704_2?needAccess=true`, but please note that access is available if you are using the internet provided by Yonsei University.

Should you choose to undertake this project, please refer to the following guidelines:

- As mentioned previously, this paper, without any intensive mathematical description, is more straightforward compared to the one in the first project. Therefore, **I anticipate a more detailed and thorough explanation of this paper** from those who select this option, compared to expectations for the first project.
- As I've suggested earlier in the first project, **I strongly urge you to reproduce some of the results presented in this paper**. While there are multiple R packages available for the permutation method, programming it from scratch will certainly enhance your statistical computing skills. Besides using the data-generation process from this paper, you might also consider referring to the data-generation scheme from the first project to evaluate the effectiveness of the permutation method in that context. For details on this, feel free to consult with me or with colleagues tackling the first project.

Good luck, and enjoy the challenge with your teammates!

4 Data Analysis: Visualizing High-Dimensional MNIST Data using t-SNE

One of the most important unsupervised learning methods we have learned is Principal Component Analysis (PCA), which is a linear dimensionality reduction technique. PCA captures the directions of maximum variance in the data, ensuring that the first principal component explains the most variance, and each subsequent component explains the next highest amount. However, PCA is limited in some sense because it assumes linear correlations and relationships among features and may not capture complex, non-linear structures inherent in the data. To address this, we can take advantage of a non-linear dimensionality reduction method called **t-distributed Stochastic Neighbor Embedding (t-SNE)**, which aims to preserve local structures and has been particularly effective in visualizing clusters in high-dimensional data.

In this project, we aim to understand the principles behind t-SNE and apply it to visualize patterns, clusters, and structures within the MNIST dataset of handwritten digits. Here is a tentative outline of this project I suggest:

- **1. Introduction**: Explain a brief history and purpose of t-SNE and discuss the main principles and intuition behind it: how it converts similarities between data points to joint probabilities and minimizes the divergence between the distribution in the high-dimensional space and the low-dimensional space.
- **2. Mathematical Overview**: Explore the cost function used by t-SNE and its optimization and the role of perplexity and other hyperparameters.
- **3. Data Preprocessing**: Make sure that the MNIST dataset is correctly loaded and its structure is understood. Normalize pixel values between 0 and 1, ensuring that the data is suitably preprocessed for dimensionality reduction.
- **4. Implementation**: Choose an appropriate tool or library (e.g., Scikit-learn) and tune hyperparamters appropriately. You may refer to the library documentation.
- **5. Visualization**: Generate 2D (or optionally 3D) embeddings of the MNIST digits. Create scatter plots where data points are color-coded based on their true digit labels. Ensure that plots are clear, with a legend, axis labels, and a title. The plots should allow for easy identification of clusters and potential overlaps between different digit classes.
- **6. Analysis and Interpretation**: Observe and comment on the clusters formed by different digits. Are some digits more distinctly clustered than others? Note any overlaps or confusions between digit classes. Which digits tend to be closer to each other in the t-SNE space? Discuss any anomalies or outliers and their potential implications.
- **7. Comparison with PCA**: Implement PCA on the MNIST dataset and compare how the t-SNE visualization differs from the visualization obtained by PCA. What insights can be gained from these differences?
- **8. Conclusion**

Good luck, and enjoy the challenge with your teammates!

5 Data Analysis: Segmenting the IRIS Dataset using Spectral Clustering

One of the unsupervised learning techniques we have discussed is K-means clustering, which aims to partition data into distinct, non-overlapping subsets based on their similarities. While K-means works well when clusters are isotropic and roughly of the same size, it struggles with more complex cluster shapes and varying densities due to its reliance on centroid-based distance metrics. Enter **Spectral Clustering**, an algorithm that addresses these challenges. It leverages the properties of a similarity graph and its spectral (eigen) properties to identify non-linearly separable groupings in the data. Instead of solely focusing on pointwise distances as in traditional clustering techniques, Spectral Clustering operates in a transformed space where complex cluster shapes become more separable, allowing for a more nuanced and flexible clustering approach.

In this project, we aim to delve into the mechanics of Spectral Clustering and harness its capabilities to segment and discern patterns within the renowned IRIS dataset of floral species. Below is a tentative outline for this endeavor that I propose:

- **1. Introduction**: Explain the mathematical foundations of Spectral Clustering. Referring to the lecture note created by Minju and Dongyoon might be beneficial. Some key terminologies to cover include, but are not limited to, similarity graph, Laplacian matrix, eigenvectors, and eigenvalues.
- **2. Data Preprocessing and Exploration**: Conduct an overview of the IRIS dataset, detailing its features, the number of data points, and classes. Your preliminary analysis should be complemented with basic statistics and visualization. In terms of preprocessing, ensure that you check for missing values and standardize the data when required. Conclude this phase by computing the similarity graph, choosing an appropriate measure such as the Gaussian kernel.
- **3. Implementation**: Choose an appropriate tool or library (e.g., Scikit-learn) and tune hyperparamters appropriately. You may refer to the library documentation. Dive deeper by computing the Laplacian matrix manually and understand the eigenvalue and eigenvector decomposition process. This will provide a hands-on grasp of the mathematics behind the library's implementation.
- **4. Visualization and Interpretation**: Visualize the clustered data in 2D or 3D using suitable techniques (e.g., scatter plot). Compare clusters with true labels from the IRIS dataset. To understand the distinctiveness of each cluster, examine the central tendency of each feature within the cluster. Finally, provide an overarching interpretation. What do these clusters signify in relation to the features of the dataset? Which features contribute most to the cluster separations?
- **5. Comparison with Other Clustering Techniques**: Implement another clustering technique (e.g., K-means) on the IRIS dataset and compare the results in terms of cluster coherence, separation, and alignment with the true labels.
- **6. Conclusion**

Good luck, and enjoy the challenge with your teammates!

References

[1] J. Bai and S. Ng, "Determining the number of factors in approximate factor models," *Econometrica*, vol. 70, no. 1, pp. 191–221, 2002.

[2] A. Onatski, "Determining the number of factors from empirical distribution of eigenvalues," *The Review of Economics and Statistics*, vol. 92, no. 4, pp. 1004–1016, 2010.

[3] C. Lam and Q. Yao, "Factor modeling for high-dimensional time series: Inference for the number of factors," *The Annals of Statistics*, vol. 40, no. 2, pp. 694 – 726, 2012.

[4] S. C. Ahn and A. R. Horenstein, "Eigenvalue ratio test for the number of factors," *Econometrica*, vol. 81, no. 3, pp. 1203–1227, 2013.

[5] J. Fan, J. Guo, and S. Zheng, "Estimating number of factors by adjusted eigenvalues thresholding," 2019.

[6] J. L. Horn, "A rationale and test for the number of factors in factor analysis," *Psychometrika*, vol. 30, pp. 179–185, June 1965.

[7] A. Buja and N. Eyuboglu, "Remarks on parallel analysis," *Multivariate Behavioral Research*, vol. 27, no. 4, pp. 509–540, 1992. PMID: 26811132.

[8] E. Dobriban, "Permutation methods for factor analysis and pca," *Annals of Statistics*, vol. 48, pp. 2824–2847, 10 2020.