GSOC PROJECT:

Positive Semi-definite Procrustes Problem

ALAPAN CHAUDHURI

September 26, 2022

CONTENTS

1 Abstract 1
2 The PSDP Problem 2
 2.1 Linear constraints on the PSDP problem 2
3 Woodgate's Algorithm 2
 3.1 The Algorithm 2
 3.2 Results 3
4 Peng's Algorithm 3
 4.1 Results 4
5 OptPSDP 4

1 ABSTRACT

Procrustes is a free, open-source, and cross-platform Python library for (generalized) Procrustes problems with the goal of finding the optimal transformation(s) that makes two matrices as close as possible to each other. As a part of my GSoC project I extended this library by implementing algorithm(s) to solve one specific form of the Procrustes problem, namely, the positive semi-definite Procrustes (PSDP) problem.

In PSDP, the optimal transformations are constrained to be a positive semi-definite matrix. The original motivation for the PSDP problem (as laid down in [2]) was that the optimal transformations can be interpreted as inverse Hessian estimates to be used in Quasi-Newton methods for unconstrained function minimization.

Throughout the duration of my GSoC project, I contributed to the development of the following algorithms:

- Woodgate's algorithm [2] (#170)
- Peng's algorithm [3] (#187)
- OptPSDP [4] (#195)
- Testing for the PSDP module (#173)

2 THE PSDP PROBLEM

Given $n \times m$ matrices A and $B \in \mathbb{R}^{n \times m}$ the corresponding positive semi-definite Procrustes (PSDP) problem is defined by

$$\min_{P \in \mathcal{S}^n_{\geqslant}} ||PA - B||^2_F = \min_{P \in \mathcal{S}^n_{\geqslant}} \text{Tr}[(PA - B)^{\dagger}(PA - B)]$$

where $||.||_F$ denotes the Frobenius norm of a matrix and $\mathcal{S}^n_{\geqslant}$ denotes the set of symmetric positive semi-definite matrices of size n.

In this project, we were interested in algorithms which can yield a set of matrices, each of which can serve as a minimizer of the expression $||PA - B||^2_F$. However, how to obtain an explicit expression of P is still an open problem.

2.1 Linear constraints on the PSDP problem

Now, adding any set of linear constraint(s) on the matrix P is same as applying a set of matrices $\{Q_i\}$ such that $\forall\ Q_i,\ \text{Tr}[Q_i P] = q_i \in \mathbb{R}$.

3 WOODGATE'S ALGORITHM

Woodgate's algorithm uses an unconstrained non-convex approach towards solving the PSDP problem. It proposes a modified Newtown algorithm to tackle a modified version of the PSDP problem, referred to as PSDP*.

$$\text{PSDP*: } min_{\mathbf{E} \in \mathbb{R}^{\ n \times n}} ||\mathbf{F} - \mathbf{E}^T \mathbf{E} \mathbf{G}||$$

Now, all local minimizers of PSDP* are also global minimizers, leading us to the fact that $\hat{\mathbf{P}} = \hat{\mathbf{E}}^T \mathbf{E}$ where $\hat{\mathbf{E}}$ is any local minimizer of PSDP* and $\hat{\mathbf{P}}$ is the required minimizer for our original PSDP problem.

3.1 The Algorithm

The basic structure of the algorithm is as follows.

1. $\mathbf{E}_0$ is chosen randomly, $i = 0$.
2. Compute $L(\mathbf{E}_i)$.
3. If $L(\mathbf{E}_i) \geqslant 0$ then we stop and $\mathbf{E}_i$ is the answer.
4. Compute $\mathbf{D}_i$.
5. Minimize $f(\mathbf{E}_i - w_i \mathbf{D}_i)$.
6. $\mathbf{E}_{i+1} = \mathbf{E}_i - w_{i\,min} \mathbf{D}_i$
7. $i = i + 1$, start from 2 again.

Initially, E_0 was being chosen randomly and we realised that the final error in precision depends on it. Furthermore, we also realised that effectively scaling E_{i+1}, every time we redefine it, reduces the error we encounter by a significant margin.

Thus, we redefined step 6 as follows.

$$\mathbf{E}_{i+1} = \mathcal{N}(\mathbf{E}_i - w_{i\,min} \mathbf{D}_i)$$

Here, scaling is performed as follows.

$$\mathcal{N}(E_i) = \tilde{\alpha} E_i,$$

$$\tilde{\alpha} = \max\{0, \frac{tr(E_i' E_i Q)}{\sqrt{2tr(E_i' E_i E_i' E_i G G')}}\}$$

3.2 Results

The final code pertaining to the implementation of this algorithm was pushed as a part of pull request #170 PSDP: Woodgate Algorithm.

We tested our implementation against tests provided as a part of the paper [2]. For example, below you can see a comparison between the algorithm proposed in the paper (Algorithm 3) and our implementation. The matrix to the transformed and the matrix to transform to are also provided below as G and F respectively.

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 3 \\ 0 & 2 & 4 \end{bmatrix} G = \begin{bmatrix} 1 & 6 & 0 \\ 4 & 3 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}$$

i	Algorithm 1	Algorithm 2	Algorithm 3
0	5.782402435	5.782402435	5.782402435
1	5.638582564	5.638582564	5.652754417
2	5.618082458	5.618082458	5.601178079
3	5.604185664	5.604185664	5.600999072
4	5.602903385	5.602903385	5.600999069
5	5.601618578	5.601618578	

```
Woodgate's algorithm took 4 iterations.
Error = 5.600999068630568.
Required P = [[ 0.2235149   -0.1105954    0.24342429]
 [-0.1105954    0.05472271 -0.12044658]
 [ 0.24342429 -0.12044658   0.26510708]]
```

4 PENG'S ALGORITHM

Peng's algorithm is presented in [3] provides an exact solution to the symmetric PSDP problem using matrix inner product and matrix decomposition theory.

The symmetric positive semi-definite Procrustes problem $\min_{P \in S_n} \|F - PG\|$, where $F, G \in R_{n \times m}$ is often appeared in many fields such as structural analysis, system parameter identification, non-linear programming, signal processing and so on.

The algorithm is constructive in nature and can be described as follows:

- Decompose the matrix $\mathbf{G}$ into its singular value decomposition.
- Construct $\hat{\mathbf{S}} = \Phi * (\mathbf{U}_1^T \mathbf{F} \mathbf{V}_1 \Sigma + \Sigma \mathbf{V}_1^T \mathbf{F}^T \mathbf{U}_1)$.
- Perform spectral decomposition of $\hat{\mathbf{S}}$.

$$\widehat{S} = \Phi * (U_1^T F V_1 \Sigma + \Sigma V_1^T F^T U_1) = N \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix} N^T,$$

- Computing intermediate matrices $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$.

$$\widehat{P}_{11} = N \begin{pmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_r \end{pmatrix}_+ N^T.$$

- Check if solution exists (if $\text{rank}(\hat{P}_{11}) = \text{rank}([\hat{P}_{11}\hat{P}_{12}])$).
- Compute $\hat{\mathbf{P}}$ (required minimizer) using $\mathbf{P}_{11}$ and $\mathbf{P}_{12}$.

$$\widehat{P} = U \begin{pmatrix} \widehat{P}_{11} & \widehat{P}_{12} \\ \widehat{P}_{12}^T & \widehat{P}_{12}^T \widehat{P}_{11}^+ \widehat{P}_{12} + P_{22} \end{pmatrix} U^T$$

Here, $*$ denotes Hadamard product and $^+$ denotes Moore-Penrose generalized inverse.

4.1 Results

The final code pertaining to the implementation of this algorithm was pushed as a part of pull request #187 Peng's Algorithm (updated PR).

5 OPTPSDP

OptPSDP in [4] proposed by H. F. Ovideo is an algorithm for solving the PSDP problem using the spectral gradient projection method. This numerical solution uses the Zhang and Hager's non-monotone technique in combination with the Barzilai and Borwein's step size to accelerate convergence. The final code pertaining to the implementation of this algorithm was pushed as a part of pull request #195 Peng's Algorithm (updated PR).

REFERENCES

1. Procrustes: A Python Library to Find Transformations that Maximiz e the Similarity Between Matrices, F. Meng, M. Richer, A. Tehrani, J. La, T. D. Kim, P. W. Ayers, F. Heidar-Zadeh, Computer Physics Communications, 276(108334), 2022..
2. Woodgate, K. G. (1993, December). A new algorithm for the positive semi-definite Procrustes problem. In Proceedings of 32nd IEEE Conference on Decision and Control (pp. 3596-3601). IEEE.
3. Jingjing, P., Qingwen, W., Zhenyun, P., Zhencheng, C. (2019). Solution of symmetric positive semidefinite Procrustes problem. The Electronic Journal of Linear Algebra, 35, 543-554.
4. Oviedo, H. F. (2019). A Spectral Gradient Projection Method for the Positive Semi-definite Procrustes Problem. arXiv preprint arXiv:1908.06497.