Riemannian classification of EEG signals with missing values

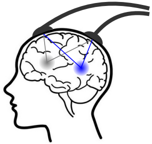
Alexandre Hippert-Ferrer[1], Ammar Mian[2], Florent Bouchard[1], Frédéric Pascal[1]

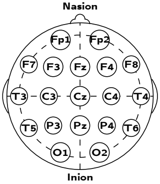
[1]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
[2]Laboratoire d'Informatique, Systèmes, Traitement de l'Information et la Connaissance (LISTIC), Annecy, France

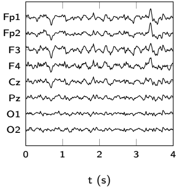
July 2022

What is electroencephalography?

- Electroencephalography (EEG): recording the brain activity
- Non-invasive neuroimagery modality
- Used in Brain Computer Interface (BCI) – *e.g.*, exoskeleton control, help mechanical ventilation, games, *etc.*







- BCI boils down to a classification task of brain signals at hand – *e.g.*, recognize a mental task, exposition to a stimulus, *etc.*

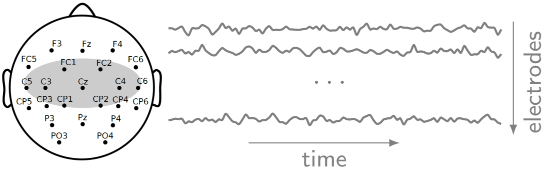
$\Rightarrow$ Here we focus on the minimum distance to Riemannian mean [Barachant et al., 2011]

Motivation of the study

Advantages	**Drawbacks**
low-cost	low spatial resolution
high temporal resolution	low SNR

- EEG data also contains faulty measurements due to impedence change, electrode popping or artifacts...

$\Rightarrow$ Leads to missing data $\Leftarrow$

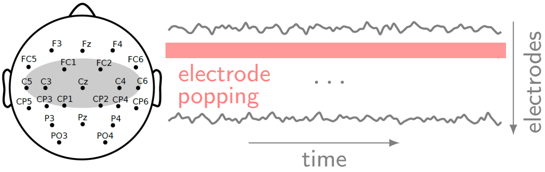


Motivation of the study

Advantages	Drawbacks
low-cost	low spatial resolution
high temporal resolution	low SNR

- EEG data also contains faulty measurements due to impedence change, electrode popping or artifacts...

$\Rightarrow$ Leads to missing data $\Leftarrow$

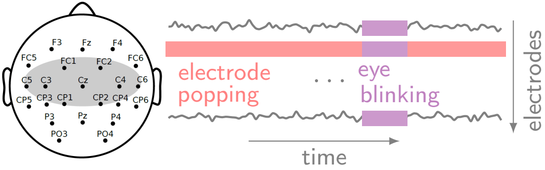


Motivation of the study

Advantages	**Drawbacks**
low-cost	low spatial resolution
high temporal resolution	low SNR

- EEG data also contains faulty measurements due to impedence change, electrode popping or artifacts...

$\Rightarrow$ Leads to missing data $\Leftarrow$

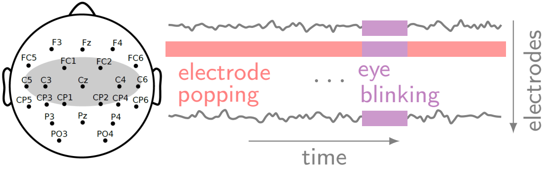


Motivation of the study

Advantages	**Drawbacks**
low-cost	low spatial resolution
high temporal resolution	low SNR

- EEG data also contains faulty measurements due to impedence change, electrode popping or artifacts...

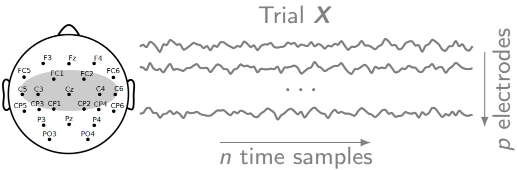
$\Rightarrow$ Leads to missing data $\Leftarrow$



$\Rightarrow$ Need to propose new classification methods that handle missing data for BCI applications.

Data representation and feature extraction

$\Rightarrow$ EEG recordings are usually represented by trials.



- Data: $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{p \times n}$
- Common hypothesis: $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$
- Spatial covariance matrices are used as features

$$\boldsymbol{\Sigma} = \frac{1}{n} \boldsymbol{X}\boldsymbol{X}^\top$$

1 trial = 1 covariance matrix to classify !

Minimum distance to Riemannian mean [Barachant et al., 2011]

- Affine invariant distance based on the geometry of the space of SPD matrices:

$$\delta_{\mathcal{AF}}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2) = \left\| \log \left(\mathbf{\Sigma}_1^{-1/2} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{-1/2} \right) \right\|_{\mathrm{F}} .$$

Minimum distance to Riemannian mean [Barachant et al., 2011]

- Affine invariant distance based on the geometry of the space of SPD matrices:

$$\delta_{\mathcal{AF}}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \left\| \log \left(\boldsymbol{\Sigma}_1^{-1/2} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-1/2} \right) \right\|_{\mathrm{F}} .$$

Given a Z-classes classification problem, with training data $(\boldsymbol{\Sigma}_\ell, y_\ell)_{1 \leq \ell \leq L}$:

Training

Compute the mean of each class $\{\overline{\boldsymbol{\Sigma}}^{(z)} : z \in \{1, \dots, Z\}\}$:

$$\overline{\boldsymbol{\Sigma}}^{(z)} = \underset{\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}}{\arg\min} \sum_{\ell=1}^{L} \delta^2_{\mathcal{AF}}(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_\ell)$$

Minimum distance to Riemannian mean [Barachant et al., 2011]

- Affine invariant distance based on the geometry of the space of SPD matrices:

$$\delta_{\mathcal{AF}}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2) = \left\| \log\left(\mathbf{\Sigma}_1^{-1/2} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{-1/2} \right) \right\|_{\mathrm{F}} .$$

Given a Z-classes classification problem, with training data $(\mathbf{\Sigma}_\ell, y_\ell)_{1 \leq \ell \leq L}$:

Training

Compute the mean of each class $\{\overline{\mathbf{\Sigma}}^{(z)} : z \in \{1, \ldots, Z\}\}$:

$$\overline{\mathbf{\Sigma}}^{(z)} = \underset{\mathbf{\Sigma} \in \mathcal{S}_p^{++}}{\arg\min} \sum_{\ell=1}^{L} \delta^2_{\mathcal{AF}}(\mathbf{\Sigma}, \mathbf{\Sigma}_\ell)$$

Testing

Prediction of a label $y^\star$ on a new sample $\mathbf{\Sigma}$:

$$y^\star = \underset{z \in \{1, \ldots, Z\}}{\arg\min} \; \delta_{\mathcal{AF}}(\mathbf{\Sigma}, \overline{\mathbf{\Sigma}}^{(z)})$$

$\Rightarrow$ assignation to the closest mean

EEG classification with missing values

Strategies to classify incomplete data using the covariance matrix

- Impute the data, then classify – *e.g.*, k-nearest neighbors (KNN) [Troyanskaya et al., 2001]
- MDRM with incomplete data: masked minimum to Riemannian mean [Yger et al., 2020]
- Expectation-Maximization (EM) algorithm $\Rightarrow$ handy iterative procedure to find $\widehat{\boldsymbol{\Sigma}}_{ML}$. [Hippert-Ferrer et al., 2022]

In what follows, **let us consider:**

- Complete data $\{\boldsymbol{y}_i\}_{i=1}^n \in \mathbb{R}^p$ and $\boldsymbol{y}_i = \{\boldsymbol{y}_i^o, \boldsymbol{y}_i^m\}$
- Unknown parameter of interest $\boldsymbol{\theta} \in \Omega$: in our case $\boldsymbol{\Sigma} \in S_p^{++}$
- A probabilistic model of the data $p(\boldsymbol{y}|\boldsymbol{\theta})$
- Maximum likelihood (ML): $\hat{\boldsymbol{\theta}}_{ML} = \underset{\boldsymbol{\theta} \in \Omega}{\arg\max}\, p(\{\boldsymbol{y}_i^o\}|\boldsymbol{\theta})$

Data incompleteness model

- Data transformation:

$$\widetilde{\boldsymbol{y}}_i = \boldsymbol{P}_i \boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i^o \\ \boldsymbol{y}_i^m \end{pmatrix}, \quad i \in [1, n]$$

$\{\boldsymbol{P}_i\}_{i=1}^n \in \mathbb{R}^{p \times p}$: set of n permutation matrix.

- Example: $n = p = 3$

$$\begin{array}{ccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \\ \hline \boxed{y_{11}} & \boxed{y_{21}} & y_{31} \\ y_{12} & \boxed{y_{22}} & \boxed{y_{32}} \\ y_{13} & y_{23} & y_{33} \end{array} \xrightarrow{\tilde{\boldsymbol{y}}_i = \boldsymbol{P}_i \boldsymbol{y}_i} \begin{array}{ccc} \tilde{\boldsymbol{y}}_1 & \tilde{\boldsymbol{y}}_2 & \tilde{\boldsymbol{y}}_3 \\ \hline y_{12} & y_{23} & y_{31} \\ y_{13} & \boxed{y_{21}} & y_{33} \\ \boxed{y_{11}} & \boxed{y_{22}} & \boxed{y_{32}} \end{array}$$

Data incompleteness model

- Data transformation:

$$\widetilde{\boldsymbol{y}}_i = \boldsymbol{P}_i \boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i^o \\ \boldsymbol{y}_i^m \end{pmatrix}, \quad i \in [1, n]$$

$\{\boldsymbol{P}_i\}_{i=1}^n \in \mathbb{R}^{p \times p}$: set of n permutation matrix.

- Example: $n = p = 3$

$$\begin{array}{ccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \\ \hline y_{11} & y_{21} & y_{31} \\ y_{12} & y_{22} & y_{32} \\ y_{13} & y_{23} & y_{33} \end{array} \xrightarrow{\tilde{\boldsymbol{y}}_i = \boldsymbol{P}_i \boldsymbol{y}_i} \begin{array}{ccc} \tilde{\boldsymbol{y}}_1 & \tilde{\boldsymbol{y}}_2 & \tilde{\boldsymbol{y}}_3 \\ \hline y_{12} & y_{23} & y_{31} \\ y_{13} & y_{21} & y_{33} \\ y_{11} & y_{22} & y_{32} \end{array}$$

- Covariance matrix:

$$\widetilde{\boldsymbol{\Sigma}}_i = \begin{pmatrix} \widetilde{\boldsymbol{\Sigma}}_{i,oo} & \widetilde{\boldsymbol{\Sigma}}_{i,mo} \\ \widetilde{\boldsymbol{\Sigma}}_{i,om} & \widetilde{\boldsymbol{\Sigma}}_{i,mm} \end{pmatrix} = \boldsymbol{P}_i \boldsymbol{\Sigma} \boldsymbol{P}_i^\top$$

$\widetilde{\boldsymbol{\Sigma}}_{i,mm}$, $\widetilde{\boldsymbol{\Sigma}}_{i,mo}$, $\widetilde{\boldsymbol{\Sigma}}_{i,oo}$ are the block CM of $\boldsymbol{y}_i^m$, $\boldsymbol{y}_i^m$ and $\boldsymbol{y}_i^o$, and $\boldsymbol{y}_i^o$.

Ingredients for our EM

1. Transformed data: $\widetilde{\boldsymbol{y}}_i = \boldsymbol{P}_i \boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i^o \\ \boldsymbol{y}_i^m \end{pmatrix}$
2. Transformed CM: $\widetilde{\boldsymbol{\Sigma}}_i = \boldsymbol{P}_i \boldsymbol{\Sigma} \boldsymbol{P}_i^\top$
3. The complete loglikelihood $\mathcal{L}_c$:

$$\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{Y}) \propto -n \log |\boldsymbol{\Sigma}| - p \sum_{i=1}^{n} \log \tau_i - \sum_{i=1}^{n} \widetilde{\boldsymbol{y}}_i^\top \widetilde{\boldsymbol{\Sigma}}_i^{-1} \widetilde{\boldsymbol{y}}_i$$

The EM algorithm

Parameter: $\boldsymbol{\theta} = \{\boldsymbol{\Sigma}\}$.

E-step: compute the expectation of $\mathcal{L}_c$

$$Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{y}_i^m|\boldsymbol{y}_i^o,\boldsymbol{\theta}^{(t)}}\left[\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{y}_i^o,\boldsymbol{y}_i^m)\right] = \mathrm{tr}\left\{\boldsymbol{B}_i^{(t)}\widetilde{\boldsymbol{\Sigma}}_i^{-1}\right\}$$

need few manipulations

with $\boldsymbol{B}_i^{(t)} = \begin{pmatrix} \boldsymbol{y}_i^o\boldsymbol{y}_i^{o\top} & \boldsymbol{y}_i^o\mathbb{E}\left[\boldsymbol{y}_i^{m\top}\right] \\ \mathbb{E}\left[\boldsymbol{y}_i^m\right]\boldsymbol{y}_i^{o\top} & \mathbb{E}\left[\boldsymbol{y}_i^m\boldsymbol{y}_i^{m\top}\right] \end{pmatrix}$.

The EM algorithm

Parameter: $\boldsymbol{\theta} = \{\boldsymbol{\Sigma}\}$.

E-step: compute the expectation of $\mathcal{L}_c$

$$Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{y}_i^m|\boldsymbol{y}_i^o,\boldsymbol{\theta}^{(t)}}\left[\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{y}_i^o,\boldsymbol{y}_i^m)\right] = \operatorname{tr}\left\{\boldsymbol{B}_i^{(t)}\widetilde{\boldsymbol{\Sigma}}_i^{-1}\right\}$$

need few manipulations

with $\boldsymbol{B}_i^{(t)} = \begin{pmatrix} \boldsymbol{y}_i^o\boldsymbol{y}_i^{o\top} & \boldsymbol{y}_i^o\mathbb{E}\left[\boldsymbol{y}_i^{m\top}\right] \\ \mathbb{E}\left[\boldsymbol{y}_i^m\right]\boldsymbol{y}_i^{o\top} & \mathbb{E}\left[\boldsymbol{y}_i^m\boldsymbol{y}_i^{m\top}\right] \end{pmatrix}$.

M-step: obtain $\boldsymbol{\Sigma}^{(t+1)}$ as the solution of the following max. problem

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in S_{++}^p}{\arg\max} \sum_{i=1}^{n} Q_i\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right)$$

The EM algorithm

Parameter: $\boldsymbol{\theta} = \{\boldsymbol{\Sigma}\}$.

***E-step*: compute the expectation of $\mathcal{L}_c$**

$$Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{y}_i^m|\boldsymbol{y}_i^o,\boldsymbol{\theta}^{(t)}}\left[\mathcal{L}_c(\boldsymbol{\theta}|\boldsymbol{y}_i^o, \boldsymbol{y}_i^m)\right] = \operatorname{tr}\left\{\boldsymbol{B}_i^{(t)}\widetilde{\boldsymbol{\Sigma}}_i^{-1}\right\}$$

need few manipulations

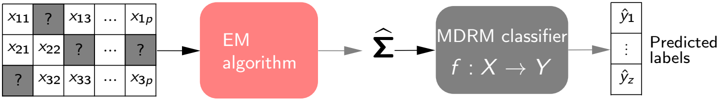
with $\boldsymbol{B}_i^{(t)} = \begin{pmatrix} \boldsymbol{y}_i^o\boldsymbol{y}_i^{o\top} & \boldsymbol{y}_i^o\mathbb{E}\left[\boldsymbol{y}_i^{m\top}\right] \\ \mathbb{E}\left[\boldsymbol{y}_i^m\right]\boldsymbol{y}_i^{o\top} & \mathbb{E}\left[\boldsymbol{y}_i^m\boldsymbol{y}_i^{m\top}\right] \end{pmatrix}$.

***M-step*: obtain $\boldsymbol{\Sigma}^{(t+1)}$ as the solution of the following max. problem**

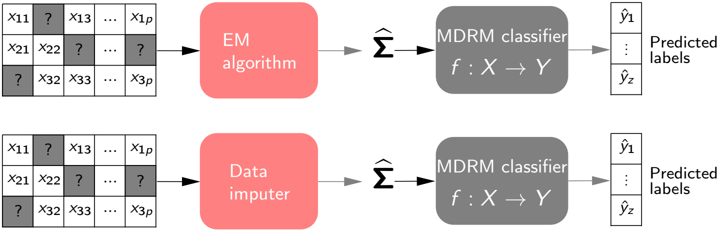
$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in S_{++}^p}{\arg\max} \sum_{i=1}^n Q_i\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right)$$

Closed-form expression: $\boldsymbol{\Sigma}^{(t+1)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{P}_i\boldsymbol{B}_i^{(t)\top}\boldsymbol{P}_i^\top$

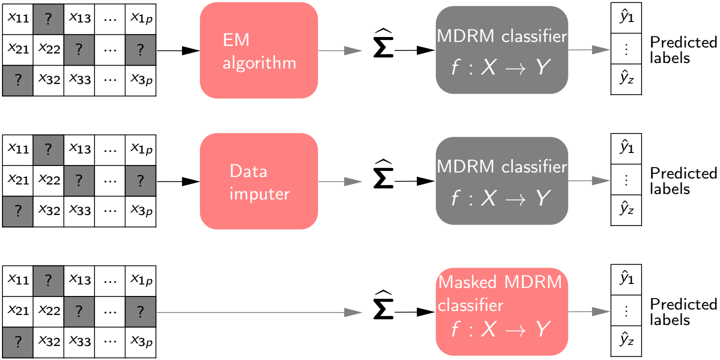
Classification pipelines



Classification pipelines



Classification pipelines



Moabb datasets

Considered EEG recordings are from the moabb database:

Dataset 1 – P300 Event related potentials

Potentials are triggered from target stimuli consisting of flashes.

- 10 subjects, 16 electrodes
- Binary classification problem (flash stimulus or no flash)
- $L = 1728$ covariance matrices to classify with $p = 16$ and $n = 103$

Moabb datasets

Considered EEG recordings are from the moabb database:

Dataset 1 – P300 Event related potentials

Potentials are triggered from target stimuli consisting of flashes.

- 10 subjects, 16 electrodes
- Binary classification problem (flash stimulus or no flash)
- $L = 1728$ covariance matrices to classify with $p = 16$ and $n = 103$

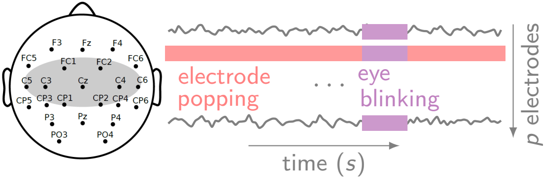
Dataset 2 – Motor Imagery

Imagination of movement of left hand (class 1), right hand (class 2), both feet (class 3) and tongue (class 4):

- 9 subjects, 22 electrodes
- 4-classes classification problem
- $L = 576$ covariance matrices to classify with $p = 22$ and $n = 1001$

Missing data scenario

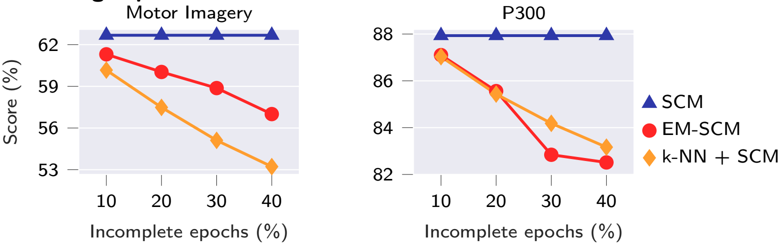
- Electrode popping: multiple electrodes are entirely removed during the experiment.
- Eye blinking: multiple electrodes are removed for a short period of time (∼200 ms).



	KNN [Troyanskaya et al., 2001]	Masked mean [Yger et al., 2020]	EM [Hippert-Ferrer et al., 2022]
Electrode popping	×	✓	✓
Eye blinking	✓	×	✓

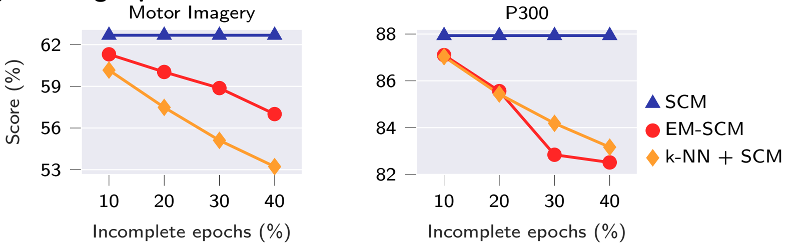
Results on moabb datasets

Eye blinking experiment

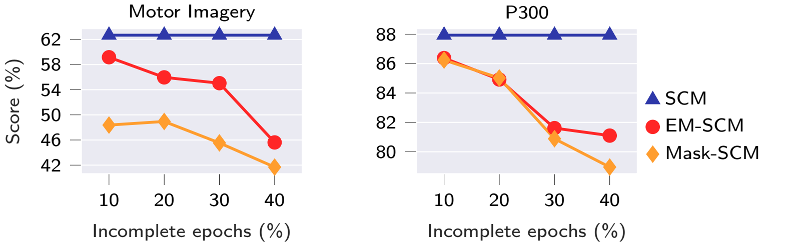


Results on moabb datasets

Eye blinking experiment



Electrode popping experiment



Conclusion and perspectives

- A new strategy to handle missing data in the context of EEG classification
- Relies on a simple yet powerful tool: the EM algorithm
- Works on a wider range of missing data scenario compared to KNN imputation and masked Riemannian means
- Competitive in terms of accuracy

Some interesting perspectives include...

- Consider robust estimation (M-estimators, t-distribution) instead of Gaussian estimation
- Adapt other classifiers to missing data: tangent space, common spatial filters, *etc.*

Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2011).
Multiclass brain–computer interface classification by Riemannian geometry.
IEEE Transactions on Biomedical Engineering, 59(4):920–928.

Hippert-Ferrer, A., El Korso, M., Breloy, A., and Ginolhac, G. (2022).
Robust low-rank covariance matrix estimation with a general pattern of missing values.
Signal Processing, 195:108460.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001).
Missing value estimation methods for DNA microarrays.
Bioinformatics, 17(6):520–525.

Yger, F., Chevallier, S., Barthélemy, Q., and Sra, S. (2020).
Geodesically-convex optimization for averaging partially observed covariance matrices.
In *Proceedings of The 12th Asian Conference on Machine Learning*, volume 129 of *Proceedings of Machine Learning Research*, pages 417–432.