Covariance Change Point Detection for Graph Signals

Even Matencio[1,2], Charles Truong[1], Laurent Oudre[1]
[1]*Université Paris Saclay, Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli,* F-91190, Gif-sur-Yvette, France
[2] *RTE, Direction R&D* F-92073, Paris La Défense, France

***Abstract*—We propose a new approach for covariance change point detection applied to graph signals. Specifically, our method draws on the notion of graph stationarity to derive a relevant parameterization of the covariance matrix that can be used in a cost function. This parameterization allows prior graph knowledge to be incorporated into the detection process and reduces the number of coefficients to be estimated. We have experimentally validated this method against relevant baselines, on synthetic and real data, and showed the influence of several parameters. These experiments demonstrated very low computational complexity, improved robustness against certain adverse effects and competitive performance in more general contexts.**

***Index Terms*—change point detection, graph signal processing, covariance matrix estimation.**

I. Introduction

Sensor networks are ubiquitous, generating a vast amount of structured multivariate time series data. Examples include electroencephalograms (EEG) capturing brain activity or meteorological sensors recording environmental variables. Non-stationarity - the statistical properties of data can change over time - is a common challenge with this type of data. Detecting abrupt changes is of significant interest, as it can reveal critical events, such as the onset of a neurological disorder in EEG data or a sudden weather change in meteorological records.

Change Point Detection (CPD) methods [1] are designed to solve this problem and can be customised to specify the nature of the changes to be detected. In multivariate time series, a popular choice is to analyze changes in the covariance structure between variables [2]. Most change point techniques in the literature focus on detecting these covariance shifts by repeatedly computing the empirical covariance matrix on sub-signals, such as the popular CUSUM algorithm [3]–[6]. However, these methods face several challenges. Estimating the covariance matrix, especially in high-dimensional settings, is difficult because it requires a large number of samples. In addition, these estimators are sensitive to noise (especially impulsive noise, which can be caused for instance by sensor defect), and can lead to incorrect matrix computation and misdetection of changes. The Graph Lasso method [7] attempts to address these issues by promoting sparsity on the precision matrix (inverse of covariance matrix). However, this approach is also computationally expensive, especially for change point detection approaches that require repeated estimations.

This work was funded by the Industrial Chair PhLAMES.

To address these challenges, we propose to leverage on the Graph Signal Processing (GSP) framework [8], which has gained popularity for its ability to analyze structured data. The assumption that the data lies on a predefined graph structure (which encodes the relationships between the different dimensions of the time series) has proven helpful in several tasks, such as interpolating missing data [9] or causality discovery [10]. In particular, we focus on the concept of *graph stationarity* introduced in the GSP community [11]–[13]. Indeed, detecting a change in graph stationarity is equivalent to using a parameterized covariance estimator that is "guided" by the underlying graph structure of the data. As will be seen, this approach provides a more efficient and potentially more robust solution to the problem of detecting covariance changes in structured multivariate time series data where a graph structure can be defined.

II. Background and notations

In this section, we define some basic notions about GSP (and graph stationarity) and change point detection.

In the following, for two integers $a < b$, $a..b$ denotes the integer interval $\{a+1, \dots, b\}$. For a sequence $(x_1, \dots, x_n)$, $x_{a..b}$ is the sub-sequence $(x_{a+1}, \dots, x_b)$.

A. Background on Graph Signal Processing (GSP)

Let us consider a weighted and undirected graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ is a set of edges and $\mathbf{W}$ a weight matrix. The Laplacian matrix of this graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix. Since G is an undirected graph, $\mathbf{L}$ is a symmetric and positive semi-definite matrix verifying $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with $\mathbf{\Lambda}$ the diagonal matrix of non-negative eigenvalues of $\mathbf{L}$ and $\mathbf{U}$ the matrix containing eigenvectors as columns, also called the Graph Fourier basis.

A *graph signal* $\mathbf{x} \in \mathbb{R}^N$ on $\mathcal{G}$, is defined as a mapping $\mathbf{x} : \mathcal{V} \to \mathbb{R}$. The Graph Fourier Transform (GFT) of graph signal $\mathbf{x}$ is defined as $\tilde{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$. A graph signal is said to be *wide-sense graph stationary (WSGS)* if and only if there exists $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N) \in \mathbb{R}_+^N$ such that its covariance matrix (in the statistical sense) $\mathbf{\Sigma} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$ can be written $\mathbf{\Sigma} = \mathbf{U}\mathrm{diag}(\boldsymbol{\gamma})\mathbf{U}^\top$ [11]–[13]. In other terms, a graph signal is WSGS if and only if its covariance matrix is diagonalizable in the Graph Fourier basis.

B. Background on change point Detection (CPD)

We here reuse the notations and formalism from [1]. Let us consider a multivariate time series (or equivalently the random process generating it) $\mathbf{y} = \{\mathbf{y}_t\}_{t=1}^T$ in $\mathbb{R}^N$. We denote the set of ground-truth change points of $\mathbf{y}$ by $\mathcal{T}^* = \{t_1^* < \ldots < t_K^*\} \subset [\![1, T]\!]$. Implicitly, we consider $t_0^* = 0$ and $t_{K+1}^* = T$ to be included in any set of change points.

When assuming that the number of ground-truth change points is known, change point detection consists in solving the following optimization problem

$$\min_{|\mathcal{T}|=K} \mathcal{V}_{\mathcal{M}}(\mathcal{T}, \mathbf{y}) = \sum_{k=0}^{K} c_{\mathcal{M}}(\mathbf{y}_{t_k..t_{k+1}}), \tag{1}$$

where $c_{\mathcal{M}}$ denotes a cost function that takes as input the sub-signal over a segment $a..b$ and computes the goodness-of-fit of this sub-signal with respect to a model $\mathcal{M}$. Most cost functions are based on probabilistic parametric models and defined as the negative log-likelihood of the Maximum Likelihood Estimator (MLE) over the considered segment. If one denotes by f the density function associated to the model $\mathcal{M}$ parametrized by θ, the resulting generic cost function $c_{\mathcal{M}}$ is given by

$$c_{\mathcal{M}}(\mathbf{y}_{a..b}) = -\sup_{\theta} \sum_{t=a+1}^{b} \log f(\mathbf{y}_t|\theta). \tag{2}$$

With such cost functions, Problem (1) can be solved exactly with dynamic programming techniques based on recursive computations on sub-signals. In the context of exact covariance matrix CPD, most methods in the literature use the cost function

$$c_{\text{MLE}}(\mathbf{y}_{a..b}) = (b-a) \log\det\left(\mathbf{\Sigma}_{a..b}\right), \tag{3}$$

where $\mathbf{\Sigma}_{a..b}$ is the empirical estimator of the covariance matrix on segment $a..b$ [2].

Alternatively, Problem (1) can be solved approximately by relying on test statistics, and more specifically CUSUM-like statistics with several variants [3]–[6]. As those tests only allow to detect a single change point, these methods are often wrapped within a suitable searching algorithm like Binary Segmentation (BS) for multiple CPD [1].

Many of the above methods rely on the computation of the empirical covariance matrix, i.e. require the estimation of $N(N+1)/2$ parameters which is known to be a tedious task [14], [15]. Thus, some methods were specifically designed to deal with this estimation issue, in the so-called "high-dimensional" setting. In [16], Avanesov and Buzun introduce a new statistic based on a refined Graph Lasso estimation of the covariance matrix [7] that they use within a BS procedure. Another alternative is to modify the cost function (3) by replacing the empirical covariance matrix by the inverse of the precision matrix estimated using the Graph Lasso algorithm [7].

III. Method

This section describes our proposed cost function and how it relates to the GSP framework.

A. Signal model and problem formulation

We observe a sequence $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$ of length T of $\mathbb{R}^N$-valued random variables such that $\forall t \geq 1, \mathbb{E}[\boldsymbol{y}_t]/* = \mathbf{0}$ and the signal of covariance matrices $\mathbb{E}\left[\boldsymbol{y}_t \boldsymbol{y}_t^\top\right]$ is piecewise constant with only a few changes, meaning that there exists K change point indices $t_1^\star < \cdots < t_K^\star$ and $K+1$ matrices $\mathbf{\Sigma}_0^\star, \ldots, \mathbf{\Sigma}_K^\star$ such that,

$$\mathbb{E}\left[\boldsymbol{y}_t \boldsymbol{y}_t^\top\right] = \sum_{k=0}^{K} \mathbf{1}_{t_k^\star..t_{k+1}^\star}\ \mathbf{\Sigma}_k^\star. \tag{4}$$

We also assume that for any t, $\boldsymbol{y}_t$ can be seen as a WSGS graph signal, i.e. that there exists a graph $\mathcal{G}$ and a Graph Fourier basis $\mathbf{U}$ such that

$$\forall k, \exists \boldsymbol{\gamma}_k \in \mathbb{R}_+^N, \quad \mathbf{\Sigma}_k^\star = \boldsymbol{U}\mathrm{diag}(\boldsymbol{\gamma}_k)\boldsymbol{U}^\top \tag{5}$$

The objective of change point detection is to estimate the change point locations $\mathcal{T}^\star$.

B. Proposed cost function

Assuming that on segment $a..b$, the $\mathbf{y}_t$ are independent and identically distributed (iid) and follow a zero-mean multivariate Gaussian distribution with fixed covariance matrix of the form (5), the expression of the MLE cost function as defined in (2) can be derived as

$$c_{\text{STATIO}}(\mathbf{y}_{a..b}) = (b-a) \sum_{n=1}^{N} \log \boldsymbol{\gamma}_{a..b}[n], \tag{6}$$

where $\boldsymbol{\gamma}_{a..b}[n]$ is the n^{th} coefficient of vector $\boldsymbol{\gamma}_{a..b}$. Interestingly, $\boldsymbol{\gamma}_{a..b}$ can easily be computed from the Graph Fourier Transform of $\mathbf{y}_{a..b}$ and more specifically as

$$\boldsymbol{\gamma}_{a..b} = \frac{1}{(b-a)} \sum_{t=a+1}^{b} \left(\tilde{\mathbf{y}}_t\right)^2 \tag{7}$$

where $\left(\tilde{\mathbf{y}}_t\right)^2$ is elementwise square of the vector $\tilde{\mathbf{y}}_t$.

Therefore, our cost function requires only the estimation of the diagonal coefficients of the covariance matrix in the Fourier basis of the graph, which is assumed to be known. Intuitively, this leads to two desirable consequences (which will be demonstrated empirically in the Results sections): firstly, the number of parameters to be estimated is smaller than that of the empirical covariance matrix (N vs. $\frac{N(N+1)}{2}$), enabling the cost function to be calculated more robustly for smaller sub-signals; secondly, the parameterization of the covariance matrix enables us to take advantage of the structure knowledge provided by the graph.

C. Efficient dynamic programming

The cost function (6) offers the nice property of being derived from (2) and can therefore be used to find an optimal solution of Problem (1) with dynamic programming. In order to speed up the process, several tricks can be used.

We first compute the Graph Fourier Transform of the whole signal and we then store the cumulative sum of the $(\tilde{\mathbf{y}}_t)^2$ bewteen 1 and T $\sum_{t=0}^{t'} (\tilde{\mathbf{y}}_t)^2$ in a matrix $\mathbf{C}$ in $\mathbb{R}^{N \times (T+1)}$,

whith first column being $\mathbf{0}$. This speeds up the computation of the estimator (7) as its computation over the segment $a..b$ simply amounts to subtracting the column $a+1$ from column $b+1$ of the matrix $\mathbf{C}$. Eventually, the cost over all the segments $a..b$, for any $1 \leq a < b \leq T$, is stored in a triangular matrix. This matrix is used to run the optimization algorithm from [17], that is accelerated thanks to the *jit* decorator of the `numba` library. The complexity of the dynamic programming is $\mathcal{O}(T^2N)$, where T is the number of samples and N is the number of dimensions/nodes.

IV. Synthetic experiments

In order to better understand the properties of the proposed approach, we start by providing experiments on synthetic data.[1]

A. Setup

We generate Erdős–Rényi (ER) graphs with $N = 20$ nodes and a target mean degree $d = 10$. The edge probability p is uniformly drawn in $[p_{\min}, p_{\max}]$ with $p_{\min} = (1 - \sigma_p) * \frac{d}{N-1}$ and $p_{\max} = (1 + \sigma_p) * \frac{d}{N-1}$. The hyper-parameter σ_p is a bandwidth coefficient that is set to $\sigma_p = 0.4$, allowing more diversity in the graph connectivity.

The signals are generated according to the model described in Section III-A with length $T = 1000$. The minimum segment length is set to $\ell = 0.4\frac{N(N+1)}{2}$ and we add to the signals a Gaussian additive white noise of SNR = 20 dB. For each simulation $N_{exp} = 80$ graphs/signals are generated.

B. Baselines

We compare our approach STATIO to three baseline methods:

- MLE: solving of Problem (1) with dynamic programming and the cost function c_{MLE} (3) (we used the ruptures Python package [1])
- GLASSO: solving of Problem (1) with dynamic programming and the cost function c_{MLE} (3) where the raw empirical covariance is replaced by the inverse of the precision matrix estimated using the Graph Lasso algorithm from [7] (we used the R package provided by the authors and set the sparsity pernalty to $4\sqrt{(\log N/(b-a))}$)
- COVCP: two-sample test statistic plugged with a binary segmentation procedure [16]. We used the R library provided by the authors, and set both the window and bootstrat set size to 80. Details can be found on the github.

C. Evaluation metrics

The chosen metric to evaluate the covariance CPD task is the F1-score F_1 defined as the harmonic mean of the *precision* and *recall* of a prediction. In the framework of CPD, the definition of true positives relies on the choice of a margin $0 < \lambda < \min_{0 \leq k \leq K} |t^*_{k+1} - t^*_k|$ [1]. In the following experiments, we choose λ = 5 samples.

[1]All the material described below can be found at https://github.com/evenmatencio/graph-signals-change-point-detection.

TABLE I: Performances of the different methods

Method	F1-score	Computation time (in seconds)
MLE	0.65 ± 0.20	12.1 ± 0.07
GLASSO	0.30 ± 0.14	537 ± 43
COVCP	0.34 ± 0.15	56 ± 7.2
STATIO	1.0 ± 0	0.5 ± 0.03

D. Results

a) Experiment 1: Benchmark performances: Table I shows the results of the four methods. It appears that the proposed method obtains the best results in terms of F1 score. This result was expected, as the data is generated according to our model, but it shows that the proposed algorithm is capable of solving the formulated problem. The following experiments will focus on how it can deal with deviations from this model. It is interesting to note that, as the number of parameters to be estimated is smaller than that of the other methods, the computation time of our method is very low. In contrast, the GLASSO method is particularly time-consuming and performs poorly in this configuration.

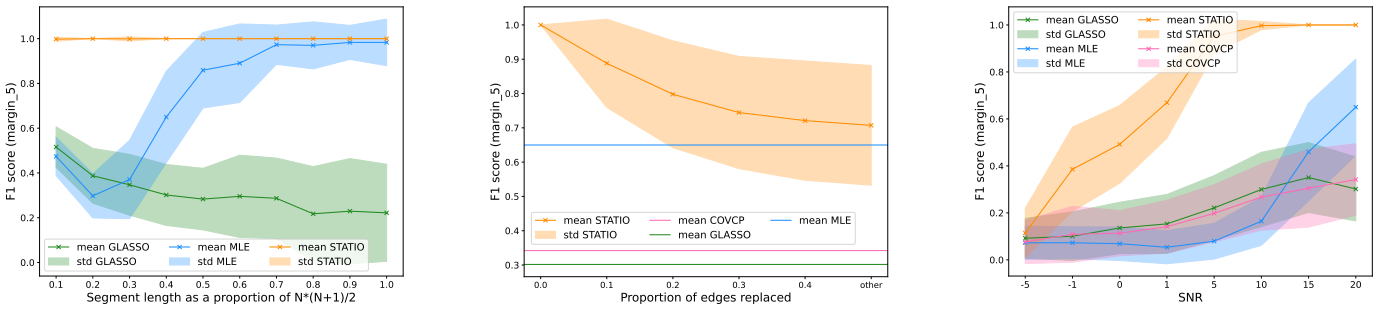
b) Experiment 2: Influence of ℓ: As discussed in Section III-B, one of the advantages of the proposed cost function is that the number of parameters to be estimated is smaller than for the standard covariance matrix. To investigate this question, we have run experiments where the minimum segment size ℓ in the data varies as a proportion of $\frac{N(N+1)}{2}$ (which is the total number of coefficients to be estimated in the full covariance matrix) - see Figure 1a.

As expected, MLE suffers from segment length decrease because it fully relies on the estimation of the whole covariance matrix. However, for larger values of ℓ the standard method shows very good performance which stresses the importance of the parametrization deduced from (5). Precisely, our method benefits from the low number of estimated coefficients and is especially efficient when the segment sizes are small.

c) Experiment 3: Influence of the graph knowledge: Our method relies on prior graph knowledge that is used in the parametrization of the covariance matrices (5). One interesting question is related to the robustness of our method to this graph knowledge. To investigate this, we have run experiments where the Graph Fourier basis provided in (5) is not exactly the one of the ground-truth graph that generated the data, but of another noisy graph where a certain percentage of edges have been substituted. Figure 1b displays the F1-score as a function of the percentage of edges replaced.

Naturally, using noisy graph knowledge decreases the performance of our method. However, even in the worst cases, one should notice that our method still outperforms the other baselines, showing again the significant benefits brought by the statistical simplicity of our estimator. When 50% of the edges are changed, the performances of our method become similar to those of MLE, which is coherent.

d) Experiment 4: Influence of noise: Figure 1c shows the influence of the SNR on the performances of all four methods. Here, we can see the benefits brought by the sparsity assumption of the Graph Lasso model as both COVCP and



(a) Influence of ℓ (b) Influence of the graph (c) Influence of the noise

Fig. 1: Influence of the stationarity segment length ℓ, of the percentage of edges replaced and of the noise level

GLASSO outperform the raw standard MLE method for low SNR because they allow a more robust estimation of the covariance matrix, even if the observed signals do not respect the sparsity assumption. However, the robustness gains achieved with a sparse estimation are still less important than those enabled by the parametrization (5).

V. Real data experiments

We here show the performances of our method on a real world usecase on EEG data.

A. Data

We use the real-world data set PhysioNet EEG Motor Movement [18]. This data set consists of EEG signals obtained from 109 volunteers, that were recorded using a 64-channel EEG setup sampled at 160 Hz. Each subject went through 14 experimental runs: two baseline runs (that will not be used in our experiments) and three repetitions of four different tasks. During these four tasks, patterns are displayed over a white screen for a short time length. Whenever the pattern appears, the volunteers are asked to react accordingly, and then relax when the target disappears. Each of the four tasks corresponds to fist or foot movements, that can be either actually performed or simply imagined by the subjects. The experimental runs last for two minutes. Each run contains 29 change points evenly spaced that successively correspond to the display and removal of the target on the screen.

B. Experimental setup

We filtered the signals with a standard third-order band-pass Butterworth filter (0.5 - 40 Hz). We sub-sampled the signals 8 times to achieve reasonable running times and ended up with signals of lengths $T = 2500$ samples. The graph fed to the STATIO method was built based on the positions of the sensors of the 64-channel EEG setup. More precisely, we retrieved the 2D coordinates of the target channels stored in the *easycap-M1* montage of the `mne` python library. Based on these coordinates, we built a 4-Nearest Neighbor undirected graph. We did not include the GLASSO method because one single experimental run was about one month with our resources.

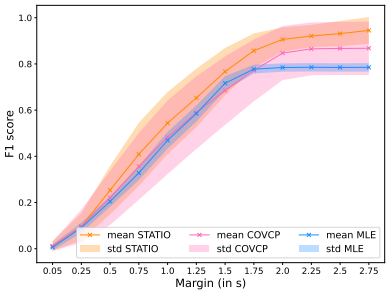


Fig. 2: Performances of the methods as a function of the margin of evaluation λ

C. Results

Figure 2 displays the performances of the methods as a function of the margin of evaluation λ (see Section IV-C). For all the margins considered, our method outperforms the other two, reaching almost 0.6 for a margin of 1 s, when just over 4 seconds separate each breakpoint. It is interesting to note that the relatively large value of the margin required to achieve satisfying performances are not only due to the latency of reaction time, which is commonly evaluated to 0.2 seconds approximately [19].

VI. Conclusion

In this article, we have introduced a novel method for change point detection in covariance matrices inspired by the notion of graph stationarity. The purpose of this method is to leverage a simple parametrization of the covariance matrix in order to bypass the crucial limitation imposed by the computation of the empirical covariance matrix. The intuitive motivations behind this method were experimentally validated in different synthetic scenarios, but also over a real use-case. Our approach also has a very low computational complexity which can be useful to process large multivariate time series. Exploring more generic scenarios, would be a nice improvement to our work. Furthermore, the sensitivity to the graph structure used to calculate our cost function naturally raises questions about the notion of graph stationarity. Further theoretical study would be a significant improvement in identifying the limits of our method.

REFERENCES

[1] Charles Truong, Laurent Oudre, and Nicolas Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, pp. 107299, Feb. 2020.

[2] M. Lavielle and G. Teyssière, "Detection of multiple change-points in multivariate time series," *Lithuanian Mathematical Journal*, vol. 46, no. 3, pp. 287–306, July 2006.

[3] Alexander Aue, Siegfried Hörmann, Lajos Horváth, and Matthew Reimherr, "Break detection in the covariance structure of multivariate time series models," *The Annals of Statistics*, vol. 37, no. 6B, pp. 4046–4087, Dec. 2009, Publisher: Institute of Mathematical Statistics.

[4] Haeran Cho and Piotr Fryzlewicz, "Multiple-Change-Point Detection for High Dimensional Time Series via Sparsified Binary Segmentation," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 77, no. 2, pp. 475–507, Mar. 2015.

[5] Daren Wang, Yi Yu, and Alessandro Rinaldo, "Optimal covariance change point localization in high dimensions," *Bernoulli*, vol. 27, no. 1, pp. 554–575, Feb. 2021, Publisher: Bernoulli Society for Mathematical Statistics and Probability.

[6] Holger Dette, Guangming Pan, and Qing Yang, "Estimating a Change Point in a Sequence of Very High-Dimensional Covariance Matrices," *Journal of the American Statistical Association*, vol. 117, no. 537, pp. 444–454, Jan. 2022.

[7] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, July 2008.

[8] David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013, Conference Name: IEEE Signal Processing Magazine.

[9] Antoine Mazarguil, Laurent Oudre, and Nicolas Vayatis, "Non-smooth interpolation of graph signals," *Signal processing*, vol. 196, pp. 108480, 2022.

[10] Lucas Zoroddu, Pierre Humbert, and Laurent Oudre, "Learning network granger causality using graph prior knowledge," *Transactions on Machine Learning Research*, 2024.

[11] Benjamin Girault, "Stationary graph signals using an isometric graph translation," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug. 2015, pp. 1516–1520, ISSN: 2076-1465.

[12] Nathanaël Perraudin and Pierre Vandergheynst, "Stationary Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, July 2017, Conference Name: IEEE Transactions on Signal Processing.

[13] Antonio G. Marques, Santiago Segarra, Geert Leus, and Alejandro Ribeiro, "Stationary Graph Processes and Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017, Conference Name: IEEE Transactions on Signal Processing.

[14] Roman Vershynin, "How Close is the Sample Covariance Matrix to the Actual Covariance Matrix?," *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, Sept. 2012.

[15] Jianqing Fan, Yuan Liao, and Han Liu, "An overview of the estimation of large covariance and precision matrices," *The Econometrics Journal*, vol. 19, no. 1, pp. C1–C32, Feb. 2016.

[16] Valeriy Avanesov and Nazar Buzun, "Change-point detection in high-dimensional covariance structure," *Electronic Journal of Statistics*, vol. 12, no. 2, pp. 3254–3294, Jan. 2018, Publisher: Institute of Mathematical Statistics and Bernoulli Society.

[17] Jushan Bai and Pierre Perron, "Computation and analysis of multiple structural change models," *Journal of Applied Econometrics*, vol. 18, no. 1, pp. 1–22, 2003.

[18] Gerwin Schalk, Dennis J. McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system," *IEEE transactions on bio-medical engineering*, vol. 51, no. 6, pp. 1034–1043, June 2004.

[19] David L. Woods, John M. Wyma, E. William Yund, Timothy J. Herron, and Bruce Reed, "Factors influencing the latency of simple reaction time," *Frontiers in Human Neuroscience*, vol. 9, Mar. 2015, Publisher: Frontiers.