Supervised change-point detection with dimension reduction

Charles Truong
Université Paris Saclay, Université Paris Cité,
ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli
F-91190 Gif-sur-Yvette, France
charles.truong@ens-paris-saclay.fr

Laurent Oudre
Université Paris Saclay, Université Paris Cité,
ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli
F-91190 Gif-sur-Yvette, France
laurent.oudre@ens-paris-saclay.fr

***Abstract*—This paper presents an automated approach for calibrating change point detection algorithms for high-dimensional time series. Our method leverages partial annotations provided by experts to learn a diagonal Mahalanobis metric, combined with a detection algorithm to replicate the expert's segmentation strategy on new signals. Our approach includes sparsity-inducing regularization to improve accuracy, which performs dimension selection and adapts to partial annotations. Our experiments on audio signals and physiological time series signals demonstrate that supervised learning improves detection accuracy significantly.**

***Index Terms*—change-point detection, metric learning, dimension reduction**

I. Introduction

Change point detection, or signal segmentation, is crucial in many machine learning pipelines that process time series. It consists in finding the temporal limits of successive homogeneous regimes of a multivariate signal. There are a large number of applications, from sleep monitoring [9], DNA sequences [4], study of neurological disorders [2], etc. In practice, the expert (e.g., a medical researcher or biologist in the context of healthcare time series) must choose the most appropriate change point detection procedure from the extensive associated literature [13]. One critical parameter to select is the kind of change to detect, related to the signal representation or the metric to measure the distance between samples. This calibration step is complex, time-consuming and often achieved by trial and error. However, the expert can often manually segment a few signals, at least partially (i.e. give approximate change locations). For instance, Fig. 1 shows the partial annotation of an expert: on a signal collected by monitoring, with an inertial sensor, a subject performing a sequence of simple activities (stand, walk, turn around, walk, stop) [2], a medical researcher has indicated a rough estimation for the activity changes. This work aims to formulate a procedure to automatically learn from segmentation examples (i.e. signals and their partial annotations) an appropriate metric. Combining the learned metric with a change-point detection algorithm could then replicate the expert's segmentation strategy.

In this article, we propose a procedure to learn from expert labels an appropriate norm that can replicate the expert's segmentation strategy and select relevant signal dimensions for this task, thanks to a sparsity regularization.

The article is organized as follows. Section II presents the change-point detection problem and an overview of the metric learning topic in the context of change-point detection. Section III describes the learning procedure and the associated optimization problem. Finally, Section IV presents the results obtained on synthetic and real data.

II. Background

A. The change point detection problem.

Consider a $\mathbb{R}^d$-valued signal $y = [y_1, y_2, \ldots, y_T]$ with T samples. Formally, change-point detection with an fixed number K of changes consists in solving the following discrete optimization problem

$$\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{\hat{K}}\} := \underset{\{t_1, t_2, \ldots, t_K\}}{\arg\min} \left[\sum_{k=0}^{K} \sum_{t=t_k}^{t_{k+1}-1} \left\| y_t - \bar{y}_{t_k..t_{k+1}} \right\|^2 \right] \quad (1)$$

where $y_{a..b}$ is the empirical mean of the sub-signal $\{y_t\}_{t=a}^{b-1}$ and $t_0 := 1$ and $t_{K+1} = T+1$ are dummy indexes and $\|\cdot\|$ is a user-defined norm on $\mathbb{R}^d$ (e.g. the Euclidean norm). The indexes $\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{\hat{K}}\}$ are the instants when the signal has the most significant mean-shifts. Several methods have been developed to optimize this sum of residuals (see [13] for a review). Algorithms based on dynamic programming solve Problem (1) exactly with a complexity of $\mathcal{O}(dKT^2)$. This is the method that will be adopted. Any faster but approximate methods, such as window-based procedures and binary segmentation, could be used instead, depending on the operational constraints.

B. Metric learning for change-point detection

Calibrating a change-point detection algorithm necessitates specifying the norm $\|\cdot\|$ used in Problem 1, which is related to the type of change that can be detected. While there are many articles focused on calibrating segmentation methods in a unsupervised way [13], only a few works have tackled

Charles Truong is funded by the PhLAMES chair of ENS Paris-Saclay. Part of this work has been funded by the Industrial Data Analytics and Machine Learning (IDAML) chair of ENS Paris-Saclay. Part of the computations has been executed on Atos Edge computer.

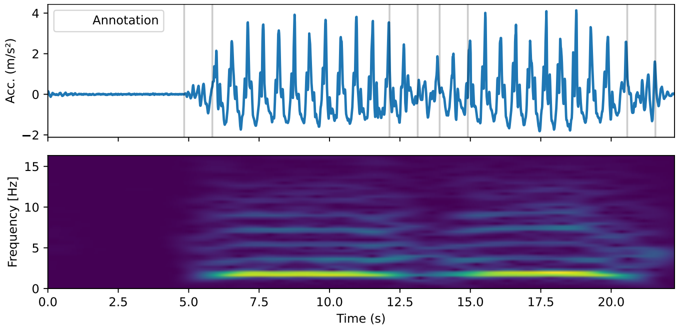


Fig. 1. Signal example with partial annotation. One (out of 6) dimension of a gait signal (acceleration along one axis) is shown along with its short-term Fourier transform (see Section IV for details). The annotations indicate that there is one change-point in each hatched area. The annotation here is partial (the exact location of the change is not provided); annotated portions are 1-second long.

this problem from a supervised standpoint. In [8], the authors learn a linear data transformation using a structured learning approach. The resulting optimization problem is computationally intensive as each gradient step requires an exact change-point estimation. Similarly, [6] proposes to minimize a well-chosen differentiable loss which still requires performing change detection many times; nevertheless, the authors are able to learn a more complex transformation of the data (a neural network). In [12], a kernel-based norm is fitted to the signals and change-point labels using a metric learning approach. The learned non-parametric transformation is not interpretable and still has a heavy memory footprint (quadratic in the number of samples). Contrary to existing approaches, our method learns a simpler data transformation. As a result, the associated loss minimization problem is easier to solve with standard convex optimization tools. In addition, important dimensions (for the change detection task) are selected while other are discarded.

III. Method

The procedure consists of two steps: (i) a learning step during which a norm is learned using labelled signals, and (ii) a prediction step during which a change-point detection procedure is applied to out-of-sample signals to segment them. This section formally introduces the nature of the labels provided by the expert and the metric learning approach.

A. *Sparse Mahalanobis-type norm*

To learn an adequate norm $\|\cdot\|$ from annotated examples, we restrict ourselves to a parametrized Mahalanobis-type (pseudo-)norm $\|x\|_w^2 := x^\intercal \operatorname{diag}(w) x$ where $\operatorname{diag}(w)$ is a diagonal matrix for a vector $w \in \mathbb{R}_+^d$ of positive weights.

Considering this norm, calibration reduces to finding an appropriate w, which can be seen as a scaling of each dimension p by w_p. In the context of high-dimensional signals, likely, all dimensions are not relevant for the change-point detection task. For instance, the signal might contain noisy components that could alter change detection algorithms.

To address this issue, we propose to enforce a sparsity constraint on the vector w. This constraint will force the metric learning step to select only the most relevant dimensions for the change-point detection problem, removing all noisy or misleading dimensions. By replacing in (1), the norm $\|\cdot\|$ by $\|\cdot\|_w$ with a properly calibrated and sparse w, the change-point detection procedure will only use the relevant dimensions and thus detect the change-point in an adequate representation space.

B. *Annotations and labels*

Annotations are provided by an expert and transformed into triplet constraints, which are then fed to the sparse metric learning algorithm. This article will consider two types of labels: full or partial.

For each training signal $y^{(l)}$, a **full label** consists in the set of change points $\mathcal{T}^{(l)} = \{t_1^{(l)}, t_2^{(l)}, \dots\}$. The set $\mathcal{T}^{(l)}$ includes all the changes contained in the signal $y^{(l)}$, according to the expert.

For each training signal $y^{(l)}$, a **partial label** consists in the set of intervals $\mathcal{S}^{(l)} = \{[s_1^{(l)}, e_1^{(l)}], [s_2^{(l)}, e_2^{(l)}], \dots\}$ that contain a change point. Instead of giving the exact position of a change $t_k^{(l)}$, the expert only provides an approximate position $[s_k^{(l)}, e_k^{(l)}]$ such that $t_k^{(l)} \in [s_k^{(l)}, e_k^{(l)}]$. All the changes contained in the signal $y^{(l)}$, according to the expert, are in one of the intervals of the set $\mathcal{S}^{(l)}$. Each interval $[s_k^{(l)}, e_k^{(l)}]$ contains only one change and the intervals do not overlap. An example of partial annotations is shown on Fig. 1.

C. *Construction of the triplets of samples*

The proposed metric learning procedure relies on triplets of samples (anchor sample, positive sample, negative sample) that will be used to construct some constraints that will be used in the metric learning procedure. Intuitively, two samples that belong to the same homogeneous segment (i.e. without change) are from the same class, while two samples that belong to two consecutive segments (i.e. separated by a change point) are from different classes. The procedure to construct these triplets varies according to the labels type (full or partial).

Using a full label $\mathcal{T}^{(l)}$, a triplet can be created as follows: for any anchor sample y_t in a certain segment $[t_k^{(l)}, t_{k+1}^{(l)}[$, a positive sample is any element of the same segment y_{t^+} of $[t_k^{(l)}, t_{k+1}^{(l)}[$ (except the anchor sample) and a negative sample y_{t^-} is any element of the previous segment $[t_{k-1}^{(l)}, t_k^{(l)}[$ or the following segment $[t_{k+1}^{(l)}, t_{k+2}^{(l)}[$.

Using a partial label $\mathcal{S}^{(l)}$, a triplet can be created similarly: for any anchor sample y_t in a certain segment $[e_k^{(l)}, s_{k+1}^{(l)}]$, a positive sample is any element of the same segment y_{t^+} of $[e_k^{(l)}, s_{k+1}^{(l)}]$ (except the anchor sample) and a negative sample y_{t^-} is any element of the previous segment $[e_{k-1}^{(l)}, s_k^{(l)}]$ or the following segment $[e_{k+1}^{(l)}, s_{k+2}^{(l)}]$.

D. *Sparse metric learning*

Let $\mathcal{D}^{(l)}$ be the set of triplets generated from the labels (full $\mathcal{T}^{(l)}$ or partial $\mathcal{S}^{(l)}$). The sparse metric learning procedure for change-point detection consists in solving the following optimization problem

$$\min_{w\in\mathbb{R}_+^d}\left[\left(\sum_l \frac{1}{|\mathcal{D}^{(l)}|}\sum_{(y_t,y_{t^+},y_{t^-})\in\mathcal{D}^{(l)}} \ell_w\left(y_t,y_{t^+},y_{t^-}\right)\right) + \lambda\left\|w\right\|_1\right] \quad (2)$$

with

$$\ell_w\left(y_t,y_{t^+},y_{t^-}\right) := \left[1+\left\|y_t-y_{t^+}\right\|_w^2-\left\|y_t-y_{t^-}\right\|_w^2\right]_+ \quad (3)$$

where $[\cdot]_+ := \max(0,\cdot)$ and $\lambda > 0$ controls the trade-off between the sparsity of w and the triplet constraints. This is simply the sum over the training set of the margin-based hinge loss and a sparsity inducing regularization. The learned $\hat{w}$, which is the solution of Problem 2, is then such that the distance between samples from the same segment is smaller than the distance between samples from consecutive regimes (separated by a change-point). Because there can be a large number of possible triplets in $\mathcal{D}^{(l)}$, learning a weight vector w can be computationally costly. A sampling strategy is frequently used to focus to reduce the computational burden; such a strategy is often called triplet mining [7]. In this work, a fixed number of triplets is simply sampled at random from each set $\mathcal{D}^{(l)}$. Also, Problem 2 includes a non-smooth regularization and large number of triplet constraints, stochastic composite optimization has been proposed [10]. This is an iterative minimization algorithm where each step is a stochastic gradient step followed by the application of a proximal operator (for the ℓ_1 norm). This work uses the implementation of [3].

IV. Experiments

In the following, our method is denoted *SML-CPD* for Sparse Metric Learning for Change-Point Detection.

a) Detection algorithms: Our method *SML-CPD* is compared to two common change-point detection algorithms: *EUC-CPD* which is equivalent to *SML-CPD* without the sparse metric learning step (i.e. the norm $\|\cdot\|_w$ reduces to the Euclidean norm $\|\cdot\|$) and *RBF-CPD* which is a kernel-based segmentation procedure that can detect general changes in the distribution of the samples [1]. The chosen kernel is the radial basis function (RBF). Note that both *SML-CPD* and *EUC-CPD* are applied on the time-frequency representation of the signal, while *RBF-CPD* is applied on the original data[1].

b) Evaluation metrics: The detection power is evaluated with the accuracy which is the proportion of correctly detected changes. For a given margin $M > 0$, a true change t is considered detected if the estimated change-point $\hat{t}$ is such that $|t-\hat{t}| < M$. All scores are computed with a 5-fold cross-validation.

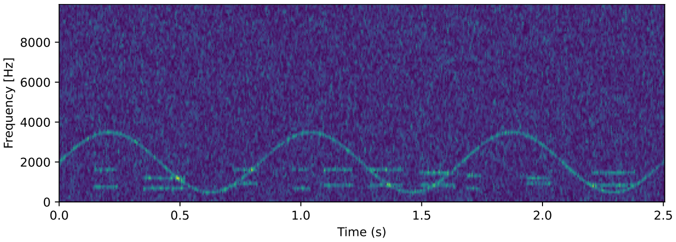


Fig. 2. STFT of a synthetic signal (SNR=-10 dB).

A. Simulated data

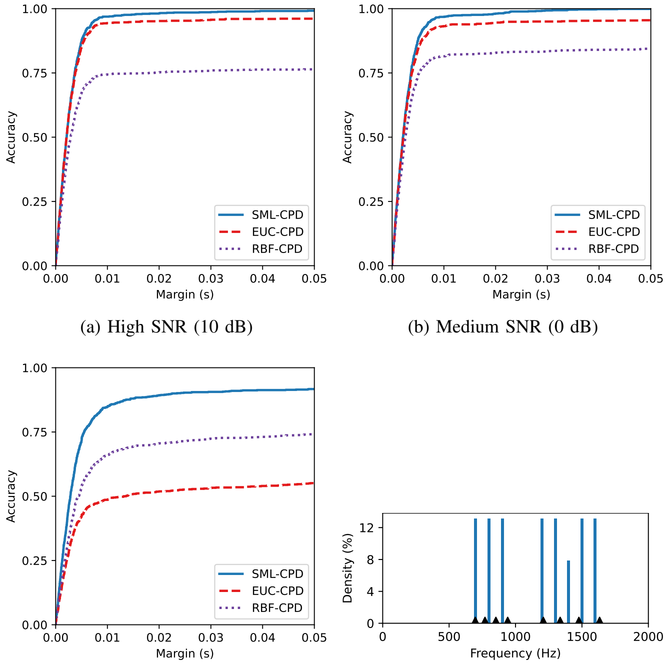
a) Data: The simulated data set consists in 50 audio time series created following the dual-tone multi-frequency (DTMF) system. The DTMF is the signal produced when dialling a phone number. This system encodes a sequence of digits as a sequence of sounds and each sound follows a dual-tone model $y[t] = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$ with $f_1 \in \{697, 770, 852, 941\}$ Hz and $f_2 \in \{1209, 1336, 1477, 1633\}$ Hz; each one of the 16 possible combinations of frequencies (f_1, f_2) is associated with a symbol in "0123456789ABCD#⋆". Here, the signals are sequences of 10 symbols separated by silences; segments (sound or silence) have a random duration between 50 and 200 ms (sampling frequency: 44.1 kHz). The time series have been corrupted by two types of noise: a sound with smoothly varying instantaneous frequency and an additive Gaussian white noise of variance chosen such that the signal-to-noise ratio (SNR) is equal to 10 dB ("High SNR" scenario), 0 dB ("Medium SNR" scenario) and -10 dB ("Low SNR" scenario). The input to our method is the short-term Fourier transform (STFT) of the raw signals (window of 10 ms and 50% overlap). See Fig. 2 for an example. The objective is to estimate the start and end of each sound.

b) Results: For all noise scenarios (high, medium and low SNR), the supervised approach *SML-CPD* is more accurate, uniformly on all error margins, as shown on Fig. 3. Unsurprisingly, when the SNR decreases, performance diminishes but the gap between *SML-CPD* and the other methods is wider, meaning the supervision adapts to the noise. A very interesting feature of our method is the ability to select dimensions that are important for the change-point detection task. The selected dimensions (here, frequencies) are shown on Fig. 3-d: they coincide with the true DTMF frequencies (there is a small discrepancy due to the discretization of the frequency domain by the STFT) while the frequencies associated with the noisy part are discarded.

B. Physiological data

a) Data: The *Gait* data set contains 42 labelled time series (sampling frequency: 100 Hz) from an inertial sensor placed at the lower back of a subject performing a fixed sequence of simple activities: "Stand", "Walk", "Turnaround", "Walk", "Stop". The objective is to detect the time indexes at which the activity of the subject changes (each signal has 4 change-points). The time series have $d = 6$ dimensions: the accelerations (m/s^2) along three axes (X, Y and Z) and

[1] We use the Python package "ruptures" [13] for the segmentation algorithms.



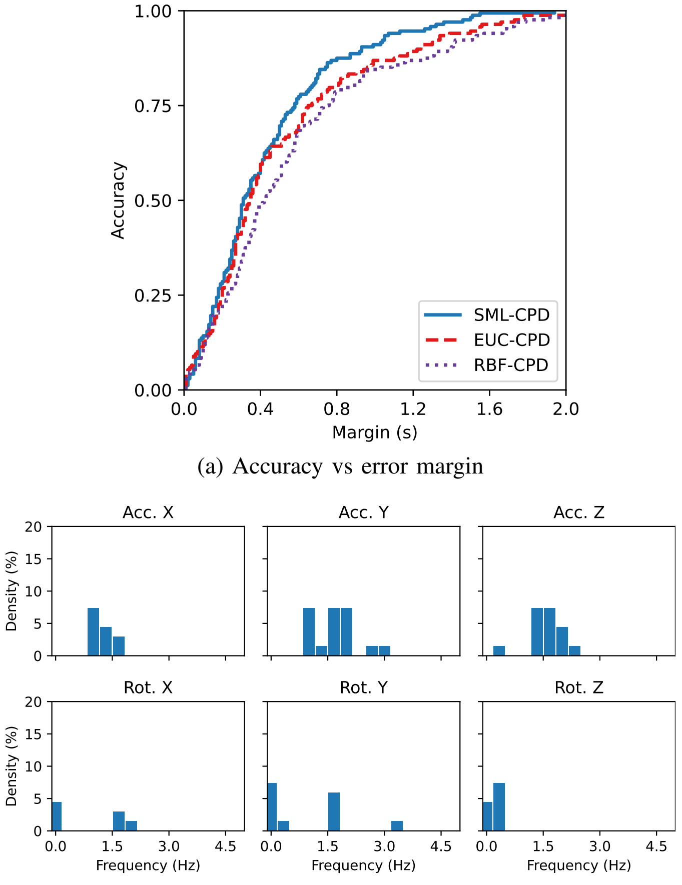
(a) High SNR (10 dB)

(b) Medium SNR (0 dB)

(c) Low SNR (-10 dB)

(d) Selected frequencies (blue bars)

Fig. 3. Results on the simulated data. (a-c) Accuracy is plotted versus the allowed error margin (in seconds). The top curve (*SML-CPD*) has the best accuracy for all margin levels. (d) Selected frequencies by *SML-CPD* for the low SNR scenario. True frequencies are marked with a small black triangle.



(a) Accuracy vs error margin

(b) Distribution of selected frequencies

Fig. 4. Results on the physiological data. (a) Accuracy is plotted versus the allowed error margin (in seconds). The top curve (SML-CPD) has the best accuracy for all margin levels. (b) Selected frequencies by SML-CPD for each dimension of the signal.

the angular velocities (deg/s) around the same three axes. Fig. 1 shows an example (only on dimension is displayed). The time-frequency representation is the STFT, computed with 300 samples per segment and an overlap of 299 samples, of each dimension; the concatenation of all STFT yields a $d = 906$-dimensional signal. As for the partial annotations, a medical researcher used an annotation tool to provide portions of 50 samples (0.5 s) around activity's changes.

b) Results: Supervision improves detection accuracy. The cross-validated accuracy is shown in Fig. 4-a. The accuracy curve can be read like a ROC curve: here, *SML-CPD* has the highest curve and outperforms other methods, meaning supervision markedly improves the detection at all margins. For a reasonable margin $M = 1$ s, accuracies are 91.1% for *SML-CPD*, 86.9% for *EUC-CPD*, 83.9% for *RBF-CPD*.

Our method projects the signals into a low-dimension space. The number of non-zero coefficients in the learned w of *SML-CPD* is around 15 in the different folds of the cross-validation, meaning that only 15 dimensions are kept to perform the segmentation, compared to the 906 dimensions of the original STFT.

SML-CPD *provides useful insights on the segmentation.* The learned weight vector $\hat{w}$ in *SML-CPD* helps the expert understand the important dimensions to segment their signals. Fig. 4-b displays the selected dimensions/frequency distribution of the STFT. First, even though possible frequencies range from 0 Hz to 50 Hz, no frequency above 4 Hz was ever chosen. Second, for the accelerations, most frequencies are picked from the [1 Hz - 2.5 Hz] band; this corresponds to the frequency of the prominent phenomenon during the walk: the repetitions of footsteps. A footstep lasts about 0.8 seconds for healthy subjects and less for neurological impaired patients (both are present in the *Gait* data set). Third, for the angular velocity around the Z axis (close to vertical), selected frequencies are below 0.5 Hz. This is consistent with the behaviour of the signal during the turnaround: there is a relatively smooth peak in the angular velocity, which is visible at frequencies below 0.5 Hz.

V. Conclusion and Future Work

To sum up, this paper introduced an approach incorporating expert annotations to enhance change-point detection algorithms, avoiding time-consuming trial-and-error calibration. Furthermore, our approach includes an informative dimension selection mechanism that improves the performance of the learned metric. In future work, we plan to address scenarios where the number of changes is unknown by combining our approach with existing methods [5], [11]. Additionally, we will explore more advanced signal transformations, such as neural networks.

References

[1] S. Arlot, A. Celisse, and Z. Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 20(162):1–56, 2019.

[2] R. Barrois, Th. Gregory, L. Oudre, Th. Moreau, Ch. Truong, A.A. Pulini, A. Vienne, Ch. Labourdette, N. Vayatis, S. Buffat, A. Yelnik, C. De Waele, S. Laporte, P.P. Vidal, and D. Ricard. An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis. *PLoS ONE*, 11(10), 2016.

[3] W. De Vazelhes, C. J. Carey, Y. Tang, N. Vauquier, and A. Bellet. metric-learn: metric mearning algorithms in Python. *Journal of Machine Learning Research (JMLR)*, 21:1–6, 2020.

[4] T. D. Hocking, G. Rigaill, P. Fearnhead, and G. Bourque. Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data. *Journal of Machine Learning Research (JMLR)*, 21:1–40, 2020.

[5] T. D. Hocking, G Rigaill, J.-P. Vert, and F. Bach. Learning sparse penalties for change-point detection using max margin interval regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 172–180, Atlanta, USA, 2013.

[6] C. Jones and Z. Harchaoui. End-to-End Learning for Retrospective Change-Point Estimation. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.

[7] M. Kaya and H. S. Bilge. Deep metric learning : a survey. *Symmetry*, 11(9):1066, 2019.

[8] R. Lajugie, F. Bach, and S. Arlot. Large-margin metric learning for constrained partitioning problems. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 297–395, Beijing, China, 2014.

[9] Mathias Perslev, Sune Darkner, Lykke Kempfner, Miki Nikolic, Poul Jørgen Jennum, and Christian Igel. U-Sleep: resilient high-frequency sleep staging. *npj Digital Medicine*, 4(1):1–12, 2021.

[10] Y. Shi, A. Bellet, and F. Sha. Sparse compositional local metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 28, pages 2078–2084, Québec City, Québec, Canada, 2014.

[11] C. Truong, L. Oudre, and N. Vayatis. Penalty learning for changepoint detection. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1569–1573, 2017.

[12] C. Truong, L. Oudre, and N. Vayatis. Supervised kernel change point detection with partial annotations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Brighton, UK, 2019.

[13] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167, 2020.