SUPERVISED KERNEL CHANGE POINT DETECTION WITH PARTIAL ANNOTATIONS

Charles Truong[⋆†] *Laurent Oudre*[‡†] *Nicolas Vayatis*[⋆†]

[⋆] CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235, Cachan, France
[†] COGNAC G, University Paris Descartes, CNRS, 75006 Paris, France
[‡] L2TI, University Paris 13, 93430 Villetaneuse, France

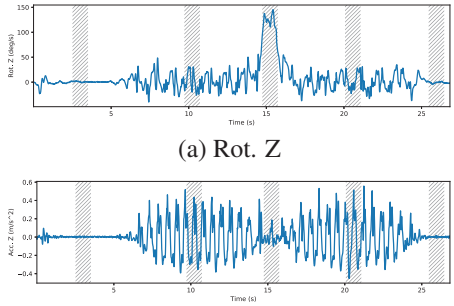
ABSTRACT

In this article, we propose an automatic procedure to calibrate change point detection algorithms. Our approach expands on the ability of an expert to provide very rough segmentation estimates, called partial annotations, for a few signal examples. Our contribution consists in a supervised strategy to learn a kernel Mahalanobis metric, which, once combined with a detection algorithm, can replicate the expert's segmentation strategy on new signals. Contrary to previous works, our approach is non-parametric, supervised and naturally accommodates partial annotations. Experiments on real-world data show that supervision significantly improves detection performance.

***Index Terms*—** Change point detection, kernel methods, kernel metric learning, biomedical signals.

1. INTRODUCTION

Change point detection or signal segmentation, which consists in finding the temporal boundaries of the successive regimes of a multivariate signal, is a central problem in signal processing. Applications are numerous and range from DNA sequences [1, 2] to industrial system monitoring [3, 4]. In practice, it is left to the expert (an economist, biologist, etc.) to choose, from the rich detection literature, an appropriate segmentation method. One particularly important parameter is the type of change to detect, which is encoded by the signal representation, or equivalently the chosen metric. This difficult and time-consuming process, often done by trial and error, could be made automatic by expanding on the expert's ability to manually segment a few signals, or at least provide partial annotations [2, 5], i.e. the start and end indexes of a portion of each regime. For instance, on Figure 1, the expert has annotated one-second-long portions of a signal collected by monitoring, with an inertial sensor, a subject performing a sequence of simple activities (stand, walk, turn around, walk, stop) [6, 7]. The objective of this work is to design a mechanism to automatically infer from segmentation examples (i.e. signals and their partial annotations) an appropriate metric. A change point detection algorithm that uses the inferred metric should be able to reproduce the expert's segmentation strategy.



(a) Rot. Z

(b) Acc. Z

Fig. 1. Example of a partially annotated signal, from the *Gait* data set (see Section 3.1). The annotations (hatched areas) denote portions of the signal that are considered homogeneous (i.e. not containing any change point).

Related work. Given a $\mathbb{R}^d$-valued signal $y := \{y_t\}_1^T$ with T samples, the task of change point detection with a fixed number K of changes consists in finding the indexes $\hat{t}_k$ ($k = 1, \dots, K$) such that

$$\hat{t}_1, \dots, \hat{t}_K := \underset{t_1 < \dots < t_K}{\arg\min} \sum_{k=0}^{K} \sum_{t=t_k+1}^{t_{k+1}} \left\| y_t - \bar{\mu}_{t_k..t_{k+1}} \right\|^2 \quad (1)$$

where $\bar{\mu}_{a..b}$ is the mean value of the sub-signal $\{y_t\}_{t=a+1}^b$, $t_0 := 0$ and $t_{K+1} := T$ are dummy indexes and $\|\cdot\|$ is a user-defined norm on $\mathbb{R}^d$ (typically, the Euclidean norm). Numerous algorithms can be found in the literature to minimize this sum of residuals, under various settings. Optimal methods find the exact change points that optimize the criterion (1). The most widely used procedure is based on dynamic programming [2, 8]. Faster but approximate methods have also been developed; well-known examples include window-based procedures [8, 4] and binary segmentation [3]. This article focuses on the calibration of the norm $\|\cdot\|$. To the best of our knowledge, there is only one work on su-

This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

pervised change point detection which aims at finding an appropriate metric [9]. The authors use, in the criterion (1), a linear Mahalanobis-type (pseudo-)norm $\|\cdot\|_M$ given by $\|u\|_M := u'Mu$ ($\forall u \in \mathbb{R}^d$) where the metric matrix M is positive semi-definite (psd). The optimal metric matrix is learned by minimizing a convex loss between signal partitions. Initially, this algorithm requires full labels (i.e. the change point positions). A non-convex strategy is proposed to accommodate partial labels; however, it relies on the ability of the user to properly initialize the metric. Also, the learned metric is only sensitive to mean-shifts, which can be a drawback for complex signals. Conversely, kernel methods have emerged because they are able to detect changes in higher-order moments of probability distributions [10, 11]. Formally, let $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ denote a kernel function and $\mathcal{H}$, the associated reproducing kernel Hilbert space (rkhs). The related mapping function $\phi : \mathbb{R}^d \to \mathcal{H}$ is implicitly defined by $\phi(y_t) = k(y_t,\cdot) \in \mathcal{H}$, with $\langle \phi(y_s)|\phi(y_t)\rangle_{\mathcal{H}} = k(y_s,y_t)$ and $\|\phi(y_t)\|^2_{\mathcal{H}} = k(y_t,y_t)$. Kernel change point detection amounts to minimizing a criterion of the form (1) where the signal y is replaced by its high-dimensional mapping $\{\phi(y_t)\}_t$ and the norm is $\|\cdot\|_{\mathcal{H}}$. Those methods have been extensively studied both from an algorithmic [12, 13] and theoretical [10, 14] standpoint. Nevertheless, they are unsupervised by nature.

We propose to extend the criterion (1) to the general class of non-parametric kernel Mahalanobis-type norm $\|\cdot\|_{\mathcal{H},M}$ given by $\|\phi(u)\|_{\mathcal{H},M} := \phi(u)'M\phi(u)$ ($\forall u \in \mathbb{R}^d$) where M is a psd matrix. The sum of residuals (1) can be rewritten as follows, for any set of K change points $\mathcal{T} = \{t_1,\dots,t_K\}$:

$$V(\mathcal{T}) := \sum_{k=0}^{K} \sum_{t=t_k+1}^{t_{k+1}} \left\|\phi(y_t) - \bar{\mu}_{t_k..t_{k+1}}\right\|^2_{\mathcal{H},M}. \qquad (2)$$

The norm $\|\cdot\|_{\mathcal{H},M}$ controls the type of change point that can be detected, and is to be algorithmically calibrated using the available partial annotations. Our approach builds upon kernel metric learning methods, the work of [15] in particular, which have been successfully applied in classification, ranking, clustering [16, 17], but have yet to be put into practice for change point detection.

Contributions. The contribution of this article is a scheme to learn a kernel Mahalanobis-type norm using a set of training examples (i.e. signals and their partial annotations). The method that is described offers a new perspective on supervised signal segmentation. Compared to previous works, our approach is non-linear, non-parametric and can accommodate an arbitrary kernel. Once learned, the metric can be combined with any detection algorithm based on the minimization (exact or approximate) of the criterion (2). Experiments on real-world time-series show that the learning step improves the detection accuracy of several segmentation algorithms.

2. METHOD

Our approach consists in a learning step, during which an optimal metric matrix $\widehat{M}$ is estimated and a predicting step, during which change point detection is performed on new signals, using the criterion V. This section describes the successive steps to learn a metric and apply it on samples from new signals.

From annotations to constraints. We propose a scheme to construct similarity and dissimilarity constraints from partial annotations: *similarity* constraints are pairs of samples that should be close according to the learned distance, while *dissimilarity* constraints are pairs of samples that should be far according to the learned distance. Precisely, let $y^{\text{train}} := [y_1^{\text{train}}, y_2^{\text{train}}, \dots]$ denote the concatenation of all training samples, i.e. samples that belong to an annotated portion of a training signal. The two vectors y_s^{train} and y_t^{train} are considered "similar" if they belong to the same regime and "dissimilar" if they belong to two consecutive regimes of the same signal. Only pairs of samples that are from the same regime or two consecutive regimes create a similarity/dissimilarity constraint. Samples that do not belong to a homogeneous portion of the signal (according to the annotations) do not create any constraint. This scheme is illustrated on Figure 2.

Kernel metric learning. Once the constraints have been generated, the optimal metric matrix $\widehat{M}$ can be estimated. To that end, we define $\widehat{M}$ to be the solution of the following constrained optimization problem:

$$\begin{aligned} &\min_{M\succeq 0} \quad D_{LD}(M,I) \quad \text{s.t.} \\ &\left\|\phi(y_s^{\text{train}}) - \phi(y_t^{\text{train}})\right\|^2_{\mathcal{H},M} \le u,\ y_s^{\text{train}} \text{ and } y_t^{\text{train}} \text{ similar} \\ &\left\|\phi(y_s^{\text{train}}) - \phi(y_t^{\text{train}})\right\|^2_{\mathcal{H},M} \ge v,\ y_s^{\text{train}} \text{ and } y_t^{\text{train}} \text{ dissimilar} \end{aligned} \qquad (3)$$

where $D_{LD}(M,M_0) := \operatorname{tr}(MM_0^{-1}) - \log\det(MM_0^{-1})$ is the LogDet divergence which acts as a distance on the set of psd matrices and $u > 0$ (resp. $v > 0$) is an upper (resp. lower) bound on the intra-regime (resp. inter-regime) pairwise distances. The divergence between M and the identity matrix is akin to a regularization.

The resolution of problem (3), which is often referred to as Information-Theoretic Metric Learning (ITML), has been extensively studied theoretically [15, 18] and algorithmically [17]. We present here the key aspects of the optimization algorithm. First, the optimization (3) is performed over the space of psd matrices on the feature space $\mathcal{H}$, which is possibly infinite dimensional and only implicitly defined through the kernel $k(\cdot,\cdot)$. An equivalent but finite-dimensional and more efficient formulation is proposed, where the kernel matrix is learned instead of the metric matrix. Under this setting, the output of the kernel metric learning algorithm is *not the*

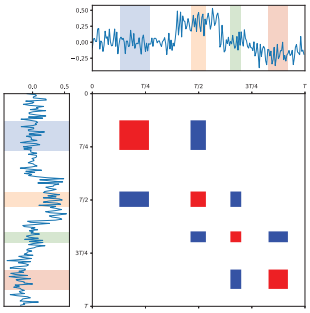


Fig. 2. Illustration of the scheme to transform annotations into constraints. Annotations are highlighted in coloured areas on the (dummy) signal. Similarity/dissimilarity constraints can be stored in a matrix A such that $A_{st} = 1$ (in red) if y_s and y_t are similar, $A_{st} = -1$ (in blue) if y_s and y_t are dissimilar, 0 otherwise (in white).

optimal metric matrix $\widehat{M}$ but rather the Gram matrix $\widehat{G}$ such that $Gst := \phi(y_s^{\text{train}})'\widehat{M}\phi(y_t^{\text{train}})$ for all training samples y_s^{train} and y_t^{train}. Second, the equivalent formulation is solved using the iterative Bregman's method [18], with the following update rule:

$$\widehat{G} \longleftarrow \widehat{G} + \beta\widehat{G}(e_s - e_t)(e_s - e_t)'\widehat{G} \tag{4}$$

where e_t is the t-th canonical basis vector, and $\beta \in \mathbb{R}$ depends on $\phi(y_s^{\text{train}})$, $\phi(y_t^{\text{train}})$ and whether they are similar or dissimilar. Each update has a complexity of the order of $\mathcal{O}(T_{\text{train}}^2)$, where T_{train} is the number of training samples.

Computing the learned metric on new samples. After the metric learning step, all that is left is to combine the associated Mahalanobis metric $\|\cdot\|_{\widehat{M},\mathcal{H}}$ with a change point detection algorithm. To that end, one must be able to compute the inner-products $\phi(z_s)'\widehat{M}\phi(z_t)$ for any samples z_s and z_t from a new $\mathbb{R}^d$-valued signal z. A difficulty lies in the fact that the metric matrix $\widehat{M}$ is not explicitly available. Thanks to a representer type of theorem [15, Theorem 1], the knowledge of $\widehat{G}$ proves to be enough. Precisely, the following expression can be used for any samples z_s and z_t:

$$\phi(z_s)'\widehat{M}\phi(z_t) = k(z_s, z_t) + k_s' \, G^{-1}(\widehat{G} - G)G^{-1} \, k_t \tag{5}$$

where $k_\bullet := [k(z_\bullet, y_1^{\text{train}}), k(z_\bullet, y_2^{\text{train}}), \dots]'$ and G is the matrix of inner-products of the training samples in the untransformed space, i.e. $G_{st} := k(y_s^{\text{train}}, y_t^{\text{train}})$.

Intuition behind the learned metric. Using the Mahalanobis-type norm $\|\cdot\|_{\mathcal{H},M}$ can be seen as performing the following operations: the signal samples are first mapped to a high-dimensional feature space (through ϕ) then they are linearly transformed, then mean-shifts are detected. Indeed, decomposing the symmetric matrix $M = U'U$ yields $\|\phi(y_t)\|_{\mathcal{H},M} = \|U\phi(y_t)\|_{\mathcal{H}}$. Therefore, measuring distances (in the feature space) with the pseudo-norm $\|\cdot\|_{\mathcal{H},M}$ is equivalent to applying a transformation $U\phi(\cdot)$ on the data. The resulting sum of residuals V, defined in (2), measures the error of approximating the transformed signal $\{U\phi(y_t)\}_t$ by a piecewise constant function. The first mapping ϕ is unsupervised (i.e. not task-specific) and extracts a great number (possibly infinite) of features while the second mapping U is linear and task-specific. Note that if the kernel k implicitly defines an infinite-dimensional rkhs, the transformation, determined by M (also infinite dimensional) is non-parametric.

3. RESULTS

We combine the kernel metric learning procedure to improve the performances of four unsupervised change point detection methods on a real-word data set called the *Gait* data set.

3.1. Experimental setting

Data set. The *Gait* data set consists of 262 annotated recordings (sampling frequency: 100 Hz) from an inertial sensor placed at the lower back of a subject performing a sequence of simple activities [6, 7]. The successive regimes are "Stand", "Walk", "Turnaround", "Walk", "Stop". The task is to detect the time indexes at which subject's activity changes. For this study, two dimensions are used: the angular velocity around the vertical axis ("Rot .Z") and the vertical acceleration ("Acc. Z"). An example is displayed on Figure 1. Both dimensions are scaled to have zero mean and unit variance. In the following, the time-frequency representation of signals from *Gait* is defined as the short-term Fourier transform (STFT), computed with 300 samples per segment and an overlap of 299 samples (see Figure 3). Partial annotations consist of 50 samples (0.5 s) taken from the middle of each regime.

Detection algorithms. The four unsupervised detection algorithm are `OptLin` [1] (based on dynamic programming, with the Euclidean norm), `OptGau` [12] (based on dynamic programming, with the rkhs norm induced by the Gaussian kernel), `WinLin` [3] (based on a window-sliding procedure, with the Euclidean norm) and `WinGau` [8] (based on a window-sliding procedure, with the rkhs norm induced by the Gaussian kernel)[1]. Window-based methods use a 100-sample long window. The related supervised algorithms ($M = \widehat{M}$) are transparently denoted ♡`OptLin`, ♡`WinLin`, ♡`OptGau`, and ♡`WinGau`. Algorithms that use the linear kernel (end in `Lin`) take as input the time-frequency representation of the signals. Algorithms that use the Gaussian kernel (end in `Gau`) take as input the (scaled) original signals. Supervised methods (those with a ♡) are evaluated with a 10-fold cross-validation.

Evaluation metrics. To evaluate segmentation accuracy, two metrics are introduced: HAUSDORFF and F1 SCORE. The

[1] All algorithms are implemented in the Python package "ruptures" [19].

Table 1. Segmentation results (mean and standard deviation)

	HAUSDORFF	F1 SCORE
WinLin	2.92 (±3.21)	0.81 (±0.17)
♡WinLin	2.04 (±2.54)	0.82 (±0.18)
OptLin	1.80 (±2.35)	0.84 (±0.17)
♡OptLin	1.10 (±0.72)	0.85 (±0.13)
WinGau	5.79 (±2.86)	0.64 (±0.17)
♡WinGau	3.77 (±3.29)	0.78 (±0.19)
OptGau	1.44 (±2.12)	0.90 (±0.15)
♡OptGau	0.99 (±1.59)	0.94 (±0.13)

(a) Global results

	Stand/Walk	Walk/Turnaround	Turnaround/Walk	Walk/Stop
WinLin	2.00 (±2.81)	0.94 (±1.58)	1.28 (±2.11)	1.42 (±2.39)
♡WinLin	1.00 (±2.02)	0.85 (±1.09)	0.66 (±0.86)	1.33 (±2.03)
OptLin	0.60 (±0.92)	0.38 (±0.62)	0.38 (±0.43)	1.69 (±2.24)
♡OptLin	0.42 (±0.42)	0.66 (±0.42)	0.61 (±0.35)	0.93 (±0.73)
WinGau	5.21 (±3.05)	1.20 (±1.64)	2.27 (±2.75)	1.50 (±2.34)
♡WinGau	2.98 (±3.22)	1.25 (±1.91)	2.37 (±2.96)	1.20 (±2.23)
OptGau	0.51 (±0.38)	0.41 (±1.05)	0.42 (±0.78)	1.23 (±2.11)
♡OptGau	0.42 (±0.41)	0.42 (±1.17)	0.42 (±0.90)	0.74 (±1.50)

(b) Absolute error (in second) by change point type

HAUSDORFF metric measures the worst prediction error [20] between a set of change point indexes $\{t_1, t_2, \dots\}$ and their estimates $\{\hat{t}_1, \hat{t}_2, \dots\}$. Formally, HAUSDORFF$(\{t_k\}_k, \{\hat{t}_k\}_k)$ is expressed in second and is equal to

$$\max\{\max_k \min_l |t_k - \hat{t}_l|\,, \max_k \min_l |\hat{t}_k - t_l|\}. \qquad (6)$$

The F1 SCORE is the geometric mean of precision PR $:=$ $\#\text{TP}/\#\{\hat{t}_l\}_l$ and recall RE $:=$ $\#\text{TP}/\#\{t_k\}_k$ where the true positive set TP $:= \{t_k \,|\, \exists\, \hat{t}_l \text{ s.t. } |\hat{t}_l - t_k| < M\}$ contains detected change points, up to a margin $M = 1$ s.

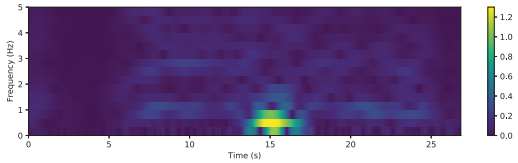
3.2. Results and discussion

Several observations can be made from the results reported in Table 1. The most important is that supervision significantly improves segmentation performances: for both metrics, the Wilcoxon signed-rank tests between the scores of a detection method and its supervised counterpart produce p-values well under 1%. The most accurate method, and the only one with an HAUSDORFF just below 1 second, is ♡OptGau. As a comparison, signals from the *Gait* data set last from 17 to 40 seconds. This is evidence that our approach is able to learn an appropriate metric from the raw signals, and replace a preprocessing step such as a STFT. Interestingly, for this algorithm, supervision mostly improves the detection of the last change point ("Walk/Stop"), by about 0.5 second on average (from 1.23 to 0.74, see Table 1-b), even though it is the least accurately detected when there are no supervision. This is due to the fact that (similarity or dissimilarity) constraints that are the most violated by the original norm determine the most the learned metric matrix [15]. As a result, the change point which is the least accurately detected by the unsupervised algorithm is likely to see the most improvement in detection accuracy. This can come at the cost of a decrease in estimation precision for certain change indexes. For instance, OptLin detects the change points between "Walk" and "Turnaround" better than ♡OptLin. Nevertheless, the two other change points are better estimated by ♡OptLin, and, according to global metrics, HAUSDORFF and F1 SCORE, detection is still significantly improved by our approach.

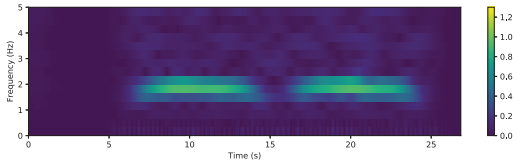
It should be noted that those results were obtained using annotations spanning 0.5 second in the middle of each regime. Table 2 shows the segmentation accuracy of ♡OptGau for different widths of partial annotation. When only 0.1 second of each regime is annotated, segmentation is still more precise than for OptGau, but not as good as when 0.5 second is annotated, since less information is provided. Conversely, there can also be too much annotation: for 1.5 seconds, performances decrease compared to 0.5 second. This is because the shortest regime ("Stop") is often less than 1.5 second, meaning that annotations include ambiguous samples that are close to the change points, and are likely to generate strong but unwanted constraints. In a nutshell, more annotations improve segmentation accuracy, but samples located around regime changes should be discarded.

Table 2. Influence of the annotation width (for ♡OptGau)

	0.1 s	0.5 s	1.5 s
HAUSDORFF	1.22 (±1.90)	0.99 (±1.59)	1.15 (±1.80)
F1 SCORE	0.92 (±0.14)	0.94 (±0.13)	0.92 (±0.14)



(a) Rot. Z (STFT)



(b) Acc. Z (STFT)

Fig. 3. STFT of the signals of Figure 1.

4. CONCLUSION

We have extended a kernel metric learning procedure to the setting of change point detection. Thanks to a novel scheme, which offers a new perspective of supervised time-series segmentation, our approach learns a non-linear, non-parametric and task-specific signal transformation, using partial annotations provided by an expert. Experiments on real-world data have shown that supervision significantly improves detection performance, for several segmentation algorithms.

5. REFERENCES

[1] R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead, "On optimal multiple changepoint algorithms for large data," *Statistics and Computing*, vol. 27, no. 2, pp. 519–533, 2017.

[2] T. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, V. Boeva, J. Cappo, O. Delattre, F. Bach, and J.-P. Vert, "Learning smoothing models of copy number profiles using breakpoint annotations," *BMC Bioinformatics*, vol. 14, no. 1, pp. 164, 2013.

[3] M. Basseville and I. Nikiforov, *Detection of abrupt changes: theory and application*, vol. 104, Prentice Hall Englewood Cliffs, 1993.

[4] A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé, "Robust changepoint detection based on multivariate rank statistics," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 3608–3611.

[5] T. 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)*, Atlanta, USA, 2013, pp. 172–180.

[6] R. Barrois-Müller, L. Oudre, T. Moreau, C. Truong, N. Vayatis, S. Buffat, A. Yelnik, C. de Waele, T. Gregory, S. Laporte, P. P. Vidal, and D. Ricard, "Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 18 Suppl 1, pp. 1880–1881, 2015.

[7] R. Barrois-Müller, T. Gregory, L. Oudre, T. Moreau, C. Truong, A. Aram Pulini, A. Vienne, C. 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*, vol. 11, no. 10, pp. e0164975, 2016.

[8] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, and O. Cappé, "A regularized kernel-based approach to unsupervised audio segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1665–1668.

[9] 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)*, Beijing, China, 2014, pp. 297–395.

[10] S. Arlot, A. Celisse, and Z. Harchaoui, "Kernel change-point detection," *arXiv preprint arXiv:1202.3878*, pp. 1–26, 2012.

[11] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research (JMLR)*, vol. 13, pp. 723–773, 2012.

[12] Z. Harchaoui and O. Cappé, "Retrospective mutiple change-point estimation with kernels," in *Proceedings of the IEEE/SP Workshop on Statistical Signal Processing*, Madison, Wisconsin, USA, 2007, pp. 768–772.

[13] Z. Harchaoui, F. Bach, and É. Moulines, "Kernel change-point analysis," in *Advances in Neural Information Processing Systems 21 (NIPS 2008)*, Vancouver, Canada, 2008, pp. 609–616.

[14] D. Garreau and S. Arlot, "Consistent change-point detection with kernels," *arXiv preprint arXiv:1612.04740v3*, pp. 1–41, 2017.

[15] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *Journal of Machine Learning Research (JMLR)*, vol. 13, pp. 519–547, 2012.

[16] E. P. Xing, M. I. Jordan, and S. J. Russell, "Distance metric learning, with application to clustering with side-Information," in *Advances in Neural Information Processing Systems 21 (NIPS 2003)*, 2003, pp. 521–528.

[17] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, Corvalis, Oregon, USA, 2007, pp. 209–216.

[18] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *Journal of Machine Learning Research (JMLR)*, vol. 19, pp. 341–376, 2009.

[19] C. Truong, L. Oudre, and N. Vayatis, "ruptures, change point detection in Python," 2018, Available: github.com/deepcharles/ruptures.

[20] L. Boysen, A. Kempe, V. Liebscher, A. Munk, and O. Wittich, "Consistencies and rates of convergence of jump-penalized least squares estimators," *The Annals of Statistics*, vol. 37, no. 1, pp. 157–183, 2009.