ETH
Swiss Federal Institute of Technology Zurich

Department of Mathematics

Seminar for Statistics

Master Thesis Spring 2019

**Malte Londschien**

# Random Forests and Other Non-parametric Classifiers for Multivariate Change Point Detection

Submission Date: October 31st 2019

Co-Adviser Solt Kovács
Adviser: Prof. Dr. Peter Bühlmann

**Abstract**

We propose a novel view on non-parametric change point detection based on classifiers. We construct a log-likelihood like statistic that uses in-sample predictions for class probabilities and propose optimization methods which are feasible computationally, yet lead to very good empirical performance when paired with Random Forests. We provide theoretical results, including consistency, which motivate our choices. The performance of the proposed methodology is examined in a simulation study.

# 1 Introduction

Change (or break) point detection considers the localisation of abrupt distributional changes in (time) ordered observations. A lot of literature has developed around the statistical methods to optimally recover change points in different scenarios and their applications. We focus on offline (retrospective) detection problems, where all observations of a time series are available at the time point of analysis. Applications include biology (e.g. detection of changes in copy number variation (Olshen et al., 2004) or ion channel recordings (Hotz et al., 2013)), climate data (Reeves et al., 2007), environmental monitoring systems (Londschien et al., 2019) and financial time series (Kim et al., 2005), to name a few. Various methods for change point detection for these applications have been developed, usually adapted to the task at hand. In the following we consider setups with independent observations.

Parametric methods typically assume that the between change point distributions come from a certain finite-dimensional family of distributions, such that change points can be found by minimizing a regularized negative log-likelihood measure over an expanded parameter space, including a parameter for the locations of the change points. While this approach quickly becomes intractable, more efficient methods that find approximate solutions have been proposed and analyzed in much detail. The classical scenario of independent univariate Gaussian variables with constant variance and changes in mean has been studied recently by e.g. Frick et al. (2014), Fryzlewicz (2014) and Pein et al. (2017) consider a relaxed version, allowing for shifts in variance additional to the mean shifts. In general parametric method often perform quite well, as long as the parametric assumptions are at least approximately met.

Non-parametric methods forego such assumptions and try to use distribution independent measures. This usually comes with a loss of power whenever a parametric method would be applicable, however still yields acceptable results if this is not the case. There are quite a few proposals for univariate non-parametric change point detection methods, for example Zou et al. (2014) and Hernan Madrid Padilla et al. (2019). The availability of well performing non-parametric methods for multivariate problems is limited, with the only proposals that we are aware of from Lung-Yut-Fong et al. (2011), Liu et al. (2013), Matteson and James (2014) Chen and Zhang (2015) and Garreau and Arlot (2018). As these methods are either distance or rank based, they often fail in high-dimensional scenarios with much noise or complex distributions.

In the new era of machine learning, many methods that are able to learn complex (conditional) distributions, e.g. $k$-Nearest Neighbors, Random Forests (Breiman, 2001) or Neural Networks have been developed. Many of these methods are non-parametric and have proven to easily outperform simple rank or distance based methods. We propose to use the flexibility of such machine learning methods and apply them to change point detection to construct a novel class of non-parametric change point detection methods.

## 1.1 Our Contribution

We propose a novel classification based non-parametric change point detection framework. Motivated by parametric change point detection (Section 2.1), we construct a log-likelihood like statistic that uses in-sample predictions for class probabilities to compare different change point setups in Section 2.2. We discuss optimization methods to find approximate minima of this statistic in Section 3.1 and consider the suitability of different classifiers in Section 3.2. We present concrete implementations of our methodology in Section 3.3 based on Random Forests and $k$-Nearest Neighbor algorithms, present approaches for model selection in Section 3.4 and provide some theoretical results in Section 3.5. Lastly,

we demonstrate improved empirical results of our methodology compared to existing multivariate non-parametric change point detection methods in Section 4.

# 2 Change Point Detection Methodologies

Consider a sequence of independent random variables $(X_i)_{i=1}^n \subset \mathbb{R}^p$ with distributions $\tilde{\mathbb{P}}_1, \dots, \tilde{\mathbb{P}}_n$ such that the map $i \mapsto \tilde{\mathbb{P}}_i$ is piecewise constant. Let

$$\alpha^0 := \{0, n\} \cup \{i : \tilde{\mathbb{P}}_i \neq \tilde{\mathbb{P}}_{i+1}\}$$

be the set of *segment boundaries* and denote with $K^0 := |\alpha^0| - 1$ the total number of segments. We label the elements of $\alpha^0$ by their natural order starting with zero. Then consecutive elements in $\alpha^0$ define segments $(\alpha_{k-1}^0, \alpha_k^0]$ for $k = 1, \dots, K^0$ within which the $X_i \sim \mathbb{P}_k := \tilde{\mathbb{P}}_{\alpha_k}$ are i.i.d. We call $\alpha_1^0, \dots, \alpha_{K^0-1}^0$ the *change points* of the time series $X_1, \dots, X_n$. We are interested in estimating the change points (or equivalently $\alpha^0$) of the time series $X_1, \dots, X_n$ upon observing a realisation $(x_i)_{i=1}^n$.

## 2.1 The Parametric Case

In a parametric setup, we assume that the distributions $\mathbb{P}_k$ belong to some family of distributions $\{\mathbb{P}_\vartheta \mid \vartheta \in A\}$ for some finite-dimensional parameter space $A$. The distribution of $X_1, \dots, X_n$ can then be parametrized with the set of segment boundaries and the parameters of the in segment distributions. Namely, for each $k = 1, \dots, K^0$ let $\vartheta_k^0 \in A$ be such that $\mathbb{P}_{\vartheta_k^0} = \mathbb{P}_k$ and write $\Theta^0 := (\vartheta_k^0)_{k=1}^{K^0}$. The tupel $(\alpha^0, \Theta^0)$ then parametrizes the distribution of the sequence $X_1, \dots, X_n$ and for all $k = 1, \dots, K^0$ and $i = \alpha_{k-1}^0 + 1, \dots, \alpha_k^0$ the $X_i$ are $\mathbb{P}_{\vartheta_k^0}$-distributed. For ease of notation let $\iota_\alpha : i \mapsto |\alpha \cap \{0, \dots, i-1\}|$ be the function that maps $i = 1, \dots, n$ to the index $k$ such that $i \in (\alpha_{k-1}, \alpha_k]$. If there exist densities $p_{\vartheta_k}$ of the distributions $\mathbb{P}_{\vartheta_k}$, the normalized negative log-likelihood of observing some $(x_i)_{i=1}^n \subset \mathbb{R}^p$ in a setup parametrized by some $(\alpha, \Theta)$ is

$$\begin{aligned} L_n\left((x_i)_{i=1}^n \mid (\alpha, \Theta)\right) &:= -\frac{1}{n} \sum_{i=1}^{n} \log\left(p_{\vartheta_{\iota_\alpha(i)}}(x_i)\right) \\ &= -\frac{1}{n} \sum_{k=1}^{K} \sum_{i=\alpha_{k-1}+1}^{\alpha_k} \log(p_{\vartheta_k}(x_i)). \end{aligned} \tag{2.1}$$

**Example.** *Let $\mathbb{P}_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $i \mapsto \mu_i$ constant on segments defined by $\alpha^0$. This scenario is the classical* change in mean *setup. Then $\Theta^0 = (\mu_{\alpha_1^0}, \dots, \mu_{\alpha_{K^0}^0})$ and $p_\mu(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. The corresponding normalized negative log-likelihood of observing a sequence $(x_i)_{i=1}^n \subset \mathbb{R}$ given some segment boundaries $\alpha$ and in segment parameters $\Theta$ is*

$$L_n\left((x_i)_{i=1}^n \mid (\alpha, \Theta)\right) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2n\sigma^2} \sum_{k=1}^{K} \sum_{i=\alpha_{k-1}+1}^{\alpha_k} (x_i - \mu_k)^2$$

Let $\mathcal{A}$ be some class of possible segment boundaries and for $K \geqslant 0$ set $\mathcal{A}_K := \{\alpha \in \mathcal{A} \mid |\alpha| = K + 1\}$. A popular estimator for $\alpha^0$ assuming the existence of $K$ change points in the time series $X_1, \dots, X_n$ is the maximum likelihood estimator

$$\begin{aligned} \hat{\alpha}_K &:= \operatorname*{arg\,min}_{\alpha \in \mathcal{A}_K} \min_{\Theta \subseteq A^K} L_n\left((X_i)_{i=1}^n \mid (\alpha, \Theta)\right) \\ &= \operatorname*{arg\,min}_{\alpha \in \mathcal{A}_K} \frac{1}{n} \sum_{k=1}^{K} \min_{\vartheta \in A} \sum_{i=\alpha_{k-1}+1}^{\alpha_k} -\log(p_\vartheta(X_i)). \end{aligned} \tag{2.2}$$

If we define

$$L_n((u, v]) := \frac{1}{n} \min_{\vartheta \in A} \sum_{i=u+1}^{v} -\log(p_\vartheta(X_i))$$

to be the normalized minimal negative log-likelihood of the observations in the segment $(u, v]$, Equation (2.2) reduces to

$$\hat{\alpha}_K = \arg\min_{\alpha \in \mathcal{A}_K} \sum_{k=1}^{K} L_n((\alpha_{k-1}, \alpha_k]).$$

Note that the minimum normalized negative likelihood in Equation (2.2) is decreasing in the number of segments $K$. Thus, if we are interested in estimating $\alpha^0$ without prior knowledge of $K^0$, we need to introduce a regularization parameter $\gamma > 0$ to avoid overfitting. Thus define the estimator

$$\begin{aligned}\hat{\alpha}_\gamma &:= \arg\min_{\alpha \in \mathcal{A}} \sum_{k=1}^{|\alpha|-1} \left( \frac{1}{n} \min_{\vartheta \in A} \sum_{i=\alpha_{k-1}+1}^{\alpha_k} -\log(p_\vartheta(X_i)) + \gamma \right) \\ &= \arg\min_{\alpha \in \mathcal{A}} \sum_{k=1}^{|\alpha|-1} \left( L_n((\alpha_{k-1}, \alpha_k]) + \gamma \right), \end{aligned} \tag{2.3}$$

which penalizes each additional change point. A choice for $\gamma$ could be made using some BIC like criterion (see e.g. Zhang and Siegmund (2007)) or using cross validation.

## 2.2 The Non-parametric Case

We will construct a measure similar to the in-sample normalized negative log-likelihood from Equation (2.1) for non-parametric change point detection using the probabilities returned by classification algorithms. For $0 \leqslant u < v \leqslant n$ write

$$\mathbb{P}_{(u,v]} := \frac{1}{v-u} \sum_{i=u+1}^{v} \tilde{\mathbb{P}}_i = \frac{1}{v-u} \sum_{k=1}^{K^0} \max\{0, \max\{v, \alpha^0_{k-1}\} - \min\{u, \alpha^0_k\}\} \mathbb{P}_k$$

for the mixture distribution of the segment $(u, v]$ and denote with $p_{(u,v]}$ its density. Recall that for any segmentation $\alpha = \{\alpha_k\}_{k=0}^K$ the function $\iota_\alpha$ maps $i = 1, \ldots, n$ to the index $k \in \{1, \ldots, K\}$ such that $i \in (\alpha_{k-1}, \alpha_k]$. We attach labels $Y = (Y_i)_{i=1}^n := (\iota_\alpha(i))_{i=1}^n$ to the time series $X = (X_i)_{i=1}^n$ and write $\mathbf{Z}_\alpha := (X_i, \iota_\alpha(i))_{i=1}^n$.

Consider any classification algorithm $\hat{p}$ that consistently predicts in-sample class probabilities.[1] Then, if learned on the labeled data set $\mathbf{Z}_\alpha$, the classifier $\hat{p}_{\mathbf{Z}_\alpha}$ will predict for any $i = 1, \ldots, n$ and $k = 1, \ldots, K$

$$\begin{aligned}\hat{p}_{\mathbf{Z}_\alpha}(x_i)_k \approx \mathbb{P}(Y = k \mid X = x_i) &= \frac{\mathrm{d}\mathbb{P}(X = x_i \mid Y = k)}{\mathrm{d}\mathbb{P}(X = x_i)} \mathbb{P}(Y = k) \\ &= \frac{p_{(\alpha_{k-1}, \alpha_k]}(x_i)}{p_{(0,n]}(x_i)} \frac{\alpha_k - \alpha_{k-1}}{n},\end{aligned}$$

allowing us to estimate

$$\frac{n}{\alpha_k - \alpha_{k-1}} \hat{p}_{\mathbf{Z}_\alpha}(x_i)_k \approx \frac{p_{(\alpha_{k-1}, \alpha_k]}(x_i)}{p_{(0,n]}(x_i)}. \tag{2.4}$$

Thus, motivated by the parametric log-likelihood (2.1), we construct a non-parametric estimate of the normalized relative log-likelihood given a particular segmentation $\alpha$ and a classifier $\hat{p}$ as

$$G_n\left((x_i)_{i=1}^n \mid (\alpha, \hat{p})\right) := \frac{1}{n} \sum_{k=1}^{K} \sum_{i=\alpha_{k-1}+1}^{\alpha_k} \log\left( \frac{n}{\alpha_k - \alpha_{k-1}} \hat{p}_{\mathbf{Z}_\alpha}(x_i)_k \right). \tag{2.5}$$

While this is similar to Equation (2.1), this compares the likelihood of the segmentation $\alpha$ to the trivial $\{0, n\}$ and thus can take values greater than zero. The following Lemma motivates the usage of a maximum relative log-likelihood estimate.

[1] Note that these might not be the algorithms that are typically used to minimize misclassification rate, see Malley et al. (2012).

**Lemma 1.** *Let*

$$\hat{p}_\alpha \coloneqq \mathbb{P}\left(Y = k \mid X\right) = \left(\frac{\alpha_{k\text{-}1} - \alpha_k}{n} \frac{p_{(\alpha_{k\text{-}1}, \alpha_k]}}{p_{(0,n]}}\right)_{k=1}^{|\alpha|-1}$$

*be the Bayes classifier corresponding to the segmentation $\alpha$. If $\alpha^0 \in \mathcal{A}$,*

$$\begin{aligned}\max_{\alpha \in \mathcal{A}} \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha, \hat{p}_\alpha)\right)\right] &= \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha^0, \hat{p}_{\alpha^0})\right)\right] \\ &= \sum_{k=1}^{K} \frac{\alpha_k^0 - \alpha_{k\text{-}1}^0}{n} D_{KL}(\mathbb{P}_k \parallel \mathbb{P}_{(0,n]}) \end{aligned} \tag{2.6}$$

*and any maximizer of (2.6) is a subsegmentation of $\alpha^0$.*

Since the predictions of any consistent classifier $\hat{p}$ learned on $\mathbf{Z}_\alpha$ will converge in distribution to those of $\hat{p}_\alpha$, it is sensible to use (2.5) together with any (non-parametric) consistent classifier to obtain a (non-parametric) estimate for change points similar as in Equations (2.2) and (2.3) for the parametric case. In particular, for the case where the number of change points is known (or assumed) to be $K$, define

$$\hat{\alpha}(\hat{p})_K \coloneqq \operatorname*{arg\,max}_{\alpha \in \mathcal{A}_K} G_n\left((X_i)_{i=1}^n \mid (\alpha, \hat{p})\right) \tag{2.7}$$

and otherwise, for some tuning parameter $\gamma$, set

$$\hat{\alpha}(\hat{p}) \coloneqq \operatorname*{arg\,max}_{\alpha \in \mathcal{A}} G_n\left((X_i)_{i=1}^n \mid (\alpha, \hat{p})\right) - \gamma|\alpha|. \tag{2.8}$$

In practice, distributions might not be absolutely continuous with respect to each other and classifiers might predict probabilities in $\{0, 1\}$. For this reason, instead of using the natural logarithm, we use $\log_\eta$, where for some $\eta > 0$ we define $\log_\eta \colon x \mapsto \log((1-\eta)x + \eta)$.

# 3 Implementation in Practice

A solution of the parametric maximum log-likelihood estimator (2.3) can be found with dynamic programming in $\mathcal{O}(n^2)$ evaluations of $L_n$ by calculating $L_n((u, v])$ for $1 \leqslant u < v \leqslant n$. In many traditional parametric setups (such as change in mean) in-sample likelihoods of neighboring segments can be recovered using cheap $\mathcal{O}(1)$ updates, making dynamic programming a viable optimization method. However, in modern high-dimensional settings where such cheap updates are not available and evaluations of $L_n$ are complex, the computational cost of dynamic programming is restrictive. In such settings, faster, but approximate algorithms to find optima of (2.3) are often applied.

Recovering the log-likelihood of a specific segmentation by stitching together different in-segment losses as required for dynamic programming is not possible with our classifier based approach, where we can only estimate relative log-likelihoods. As classifier fits tend to be costly, we need to make use of approximate algorithms to find optima of the classifier based maximum relative log-likelihood estimator (2.8).

We believe that binary segmentation style algorithms, where relative log-likelihoods are utilized, are tailor made for our setup. We give an overview of some variants and present how they can be applied to our setting of non-parametric change point detection. Furthermore, we discuss desirable properties of classifiers for change point detection, present our method of choice together with model selection approaches and lastly present some theoretical results.

## 3.1 Binary Segmentation Based Optimization Approaches

Binary segmentation (BS, Vostrikova (1981)) is a popular greedy algorithm to obtain an approximate solution to (2.3). Define the gains function of some segment $(u, v]$ at some split point $s$ to be

$$G_n^{(u,v]}(s) := L_n((u, v]) - L_n((u, s]) - L_n((s, v]), \tag{3.1}$$

the increase in log-likelihood when splitting the segment $(u, v]$ at $s$. Then, set

$$\hat{\alpha}_{(u,v]} := \operatorname*{arg\,max}_{s \in \{u+1, \ldots, v\}} G_n^{(u,v]}(s). \tag{3.2}$$

Searching for a single change point in $(X_i)_{i=1}^n$ is equivalent to finding $\hat{\alpha}_{(0,n]}$ and checking whether $G_n^{(0,n]}(\hat{\alpha}_{(0,n]}) > \gamma$. In the case of multiple change points BS finds an approximate solution to (2.3) by recursively splitting previously found segments using (3.2) until the maximal gain is not bigger than $\gamma$, the minimally required gain to split. BS typically requires $\mathcal{O}(n \log(n))$ evaluations of $G_n$.

As $G_n\left((X_i)_{i=u+1}^v \mid (\{u, s, v\}, \hat{p})\right)$ compares the likelihood of the segmentation $\{u, s, v\}$ to that of $\{u, v\}$, it provides an estimate of the gain $G_n^{(u,v]}(s)$. We can thus use BS and some of its alterations presented below for maximizing the classifier based relative log-likelihood (2.5) replacing evaluations of $G_n^{(u,v]}(s)$ with $G_n\left((X_i)_{i=u+1}^v \mid (\{u, s, v\}, \hat{p})\right)$.

**Optimistic Binary Segmentation** The number of evaluations of $G_n$ necessary for BS can still be prohibitive in many settings. Instead of evaluating $G_n^{(u,v]}(s)$ at every possible split $s$ in a full grid search to find its maximum, Haubner (2018) presents a technique to find one of its local maxima with $\log(n)$ smartly chosen evaluations. He notes that the expected gain curve for common parametric distributions is piece wise convex between the true underlying segment boundaries. Hence, splitting at a local maximum instead of the global one does not induce a false discovery and the missed global maximum can still be found in a later step. They call this adaptive search optimistic, and hence when using it instead of the grid search in BS, the resulting method is called Optimistic Binary Segmentation (OBS). This typically requires $\mathcal{O}(K^0 \log(n))$ evaluations of $G_n$.

The following Lemma shows that in the non-parametric case with a perfect classifier the expected gain curve is also piece wise convex, suggesting good performance when coupling OBS with our non-parametric estimation approach.

**Lemma 2.** *For the Bayes classifier $\hat{p}_{\{u,s,v\}}$ the expected gains curve*

$$s \mapsto \mathbb{E}\left[G_n\left((X_i)_{i=u+1}^v \mid (\{u, s, v\}, \hat{p}_{\{u,s,v\}})\right)\right]$$

*is piece wise convex between the underlying change points, with strict convexity in the $k$-th segment if $\mathbb{P}_{(u,\alpha_{k-1}]} \neq \mathbb{P}_k$ or $\mathbb{P}_k \neq \mathbb{P}_{(\alpha_k,v]}$.*

**Wild Binary Segmentation** BS and OBS might fail if the shifts in distribution of consecutive segments offset each other. Fryzlewicz (2014) proposes to sample $M$ random intervals $((u_i, v_i])_{i=1}^M$ by drawing interval boundaries uniformly from $\{0, \ldots, n\}$. When analyzing a segment $(u, v]$, Fryzlewicz suggests to select the split

$$\hat{\alpha}_{(u,v]} \coloneqq \underset{s \in \{u+1,\ldots,v\}}{\arg\max} \max_{i\,:\, u \leqslant u_i < v_i \leqslant v} G_n^{(u_i,v_i]}(s)$$

corresponding to a maximal gain over all intervals $(u_i, v_i]$ contained in $(u, v]$. This method is called Wild Binary Segmentation (WBS) and is consistent over a larger class of signals than BS. The performance of WBS depends heavily on the number of intervals $M$, which should be adapted to the (unknown) number of change points. In frequent change point scenarios $M$ might need to scale linearly with $n$, such that the overall number $\mathcal{O}(Mn)$ of evaluations of $G_n$ can be quadratic.

**Seeded Binary Segmentation** Kovács et al. (2019) note that the random sampling of search intervals in WBS is inefficient, as it results in too few intervals of small size. To alleviate this, they propose to deterministically construct overlapping search intervals that decrease exponentially in length. They call this method Seeded Binary Segmentation (SBS) and show comparable estimation performance to WBS. However, SBS only requires $\mathcal{O}(n \log(n))$ evaluations of $G_n$, independently of the number of change points.

**Two Step Algorithm** The Two Step Algorithm, first mentioned by Kaul et al. (2019) in a high-dimensional regression setting, is an EM-style approach for finding the maximum of the gain curve in a single change point scenario. The algorithm starts with an initial guess for a split point and fits models (here Lasso regression fits) separately for observations lying left and right to the split point. Keeping these fits fixed, they then evaluate (an approximation of) the gain curve for all possible splits. The difference to the grid search in standard BS is that they do not refit the model for each split point. They then repeat this procedure with the newly found optimal split as a second guess. They show a consistency result for the high-dimensional regression setting given a single change point.

The following Lemma suggests good performance when coupling the idea of the Two Step Algorithm with our classifier based approach in a single change point scenario.

**Lemma 3.** *Consider a change point setup with a single change point $\alpha_1^0$ in the segment $(u, v]$. Then, for any initial split $s \in (u+1, v-1]$ and the Bayes classifier $\hat{p} = \frac{s-u}{v-u}\frac{p_{(u,s]}}{p_{(u,v]}}$, if finite, the function*

$$j \mapsto \mathbb{E}\left[\sum_{i=u+1}^{j} \log\left(\frac{v-u}{s-u}\hat{p}(X_i)\right) + \sum_{i=j+1}^{v} \log\left(\frac{v-u}{v-s}(1-\hat{p}(X_i))\right)\right]$$

*is piece wise linear with the maximum at $j = \alpha_1^0$.*

## 3.2 Choice of Classifier

Even though the methodology of Section 2.2 is developed to be applied for any classifier that can make consistent predictions, this consistency is neither crucial for nor guarantees good empirical performance. More important are the following properties:

- Good in-sample prediction performance to obtain accurate relative log-likelihood estimates.
- Minimal dependence on tuning parameters, especially for regularization.
- Low variance to reduce the noise induced through different fits at neighboring split points.
- Low computational cost to allow for possibly many fits necessary in change point detection.

The need for non-parametric change point detection arises in cases where little is known about the structure of the analyzed data, as else parametric methods as described in Section 2.1 might be better suited. In such situations, methods with a broad applicability are optimal. Classifiers that we expect to result in good estimation performance include Random Forests (Breiman, 2001) and $k$-Nearest Neighbor methods.

## 3.3 Our Proposals

Random Forests are known to be easily applicable to most data sets without much fine tuning (see Hediger et al. (2019) in a related two-sample testing context), are scale invariant and give the possibility to retrieve unbiased in-sample Out-Of-Bag (OOB) predictions. However, fits are comparably costly and the randomness inherent in building the underlying trees increases the noisiness of the gain curve. The latter is especially problematic when evaluating at neighboring splits, as the variance of the fits might dominate the underlying change in signal. The Two Step Algorithm does not suffer from this problem, as it uses single fits for estimation of change point locations, also drastically decreasing the number of costly fits. In general, for methods that tend to overfit even slightly, one needs to take care with the Two Step Algorithm, due to its tendency to select refined splits close to the initial guess. This is not a concern for Random Forests due to the availability of OOB predictions. For better performance in complex multiple change point scenarios, we propose to combine the Two Step Algorithm with the interval drawing approach SBS. By estimating the gain curve in two steps for many seeded intervals, we expect that there is at least one interval where a change point is clearly identifiable and corresponds to a high maximum gain. Additionally, as the Two Step Algorithm method might fail if the initial split is chosen too far away from the true change point, we evaluate the gains curve using the $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{3}{4}$-quantiles of the possible split candidates as initial guesses. We then choose the split corresponding to the highest gain of the three curves as a second guess. For completeness, we present the Two Step Algorithm as used with our Random Forest approach in Algorithm 1.

The $k$-Nearest Neighbor algorithm is a simple, consistent non-parametric classification method. For $k$ large enough, it will produce similar fits for neighboring splits, resulting in a comparably smooth gain curve. It is considered to perform well in low-dimensional setups and has a computational cost of around $\mathcal{O}(n^2 p)$ if all pairwise distances are computed. These distances only need to be computed once and can then be recycled such that it is feasible to apply SBS with a full grid search on each segment with little additional cost. We expect $k$-NN to fail to produce sensible results in high-dimensional setups, e.g. when a high number of noise covariates are present, due to the curse of dimensionality. We present a visualization of the different gain curves on seeded intervals in Figure 3.

**Algorithm 1:** The Two Step Algorithm

**Input :** Observations $x_1, \ldots, x_n$, initial guesses $(s_j)_{j=1}^k \subset \{1, \ldots, n\}$, $\eta > 0$ and a classifier $\hat{p}$.

**Output:** Estimates $(\hat{\alpha}, \hat{G}_n^{(0,n]}(\hat{\alpha}))$ for the single change point and the corresponding gain.

**Step 1:** Learn classifiers $(\hat{p}_{\mathbf{Z}_{s_j}})_{j=1}^k$ and let

$$\hat{\alpha}^{(1)} \coloneqq \operatorname*{arg\,max}_{\alpha=1,\ldots,n} \max_{j=1,\ldots,k} \sum_{i=1}^{\alpha} \log_\eta \left( \frac{n}{s} \hat{p}_{\mathbf{Z}_{s_j}}(x_i)_1 \right) + \sum_{i=\alpha+1}^{n} \log_\eta \left( \frac{n}{n-s} \hat{p}_{\mathbf{Z}_{s_j}}(x_i)_2 \right)$$

**Step 2:** Learn a classifier $\hat{p}_{\mathbf{Z}_{\hat{\alpha}^{(1)}}}$ and let

$$\hat{\alpha}^{(2)} \coloneqq \operatorname*{arg\,max}_{\alpha=1,\ldots,n} \sum_{i=1}^{\alpha} \log_\eta \left( \frac{n}{s} \hat{p}_{\mathbf{Z}_{\hat{\alpha}^{(1)}}}(x_i)_1 \right) + \sum_{i=\alpha+1}^{n} \log_\eta \left( \frac{n}{n-s} \hat{p}_{\mathbf{Z}_{\hat{\alpha}^{(1)}}}(x_i)_2 \right)$$

**Return:** $\left( \hat{\alpha}^{(2)}, \sum_{i=1}^{\hat{\alpha}^{(2)}} \log_\eta(\frac{n}{s} \hat{p}_{\mathbf{Z}_{\hat{\alpha}^{(1)}}}(x_i)_1) + \sum_{i=\hat{\alpha}^{(2)}+1}^{n} \log_\eta(\frac{n}{n-s} \hat{p}_{\mathbf{Z}_{\hat{\alpha}^{(1)}}}(x_i)_2) \right)$

### 3.4 Model Selection

For change point detection with BS style optimization, model selection is typically done via thresholding each step of BS at some predefined $\gamma > 0$. In other words, when analysing a segment $(u, v]$, one splits whenever $\max_s G_n^{(u,v]}(s) > \gamma$. In case of parametric change point detection, where the gain is computed as the difference of in-segment losses, this is equivalent to minimizing Equation (2.1) over all change point scenarios $\alpha \in \mathcal{A}_{\text{BS}}$ that can be recovered from the BS tree using pruning. For non-parametric change point detection, this equivalence is no longer valid. An option is to select

$$\begin{aligned} \hat{\alpha}_\gamma &\coloneqq \operatorname*{arg\,max}_{\alpha \in \mathcal{A}_{\text{BS}}} G_n\left((x_i)_{i=1}^n \mid (\alpha, \hat{p})\right) - \gamma|\alpha| \\ &= \operatorname*{arg\,max}_{\alpha \in \mathcal{A}_{\text{BS}}} \frac{1}{n} \sum_{k=1}^{|\alpha|-1} \sum_{i=\alpha_{k\text{-}1}+1}^{\alpha_k} \log \left( \frac{n}{\alpha_k - \alpha_{k\text{-}1}} \hat{p}_{\mathbf{Z}_\alpha}(x_i)_k \right) - \gamma|\alpha|. \end{aligned}$$

To avoid oversegmentation with classifiers that tend to overfit, an optimal value for $\gamma$ could be obtained using cross validation (with equispaced folds), maximising the out-of-fold relative log-likelihood estimate. This approach performs reasonably in many settings, but proved to be a bottleneck in low-dimensional settings with repeated distributions in non-neighboring segments. Adapted to our preferred classifiers, we propose an alternative faster and better performing approach based on permutation tests. This can be applied when using the Two Step Algorithm together with a classifier that produces unbiased in-sample predictions (as Random Forests) or when fits are cheap (as with kNN, where the pairwise distances can be recycled).

In the latter case (e.g. kNN), we directly calculate the maximal gains of each search interval using different permutations. For the former case, i.e. classifiers that produce unbiased in-sample predictions together with the Two Step Algorithm, we permute the log-likelihood estimates for each fitted model to get estimates of the maximal gains given permuted observations. Note that this is not a proper permutation test as we do not repeat the second step of the Two Step Algorithm with the split found for the permuted log-likelihoods. When coupled with SBS, we compare the observed maximal gain to a chosen quantile of the maxima of the permuted values over all search intervals contained in the analyzed segment.

### 3.5 Theoretical Results

For competing methods, theoretical results are provided by Lung-Yut-Fong et al. (2011) and Chen and Zhang (2015) for single change point scenarios and Matteson and James (2014) and Garreau and Arlot (2018) in fixed number of change point scenarios. For our approach Lemma 1 gives intuition that the estimators of Equations (2.7) and (2.8) might provide reasonable results if the classifier can produce good probability predictions. However, as noted in Section 3.1, due to computational constraints it is near

impossible to compute these estimators exactly. Lemma 2 and Lemma 3 provided motivation for the use of OBS and the Two Step Algorithm as optimization methods to find approximate solutions. Showing consistency for OBS for this complex setup is beyond the scope of this paper. However, we are able to provide a consistency result in case of a single change point for the Two Step Algorithm 1 together with a consistent classifier.

**Theorem 4.** *Consider a triangular array of random variables* $X_{i,n},\ n \geqslant 0,\ i = 1, \ldots, n$ *such that* $X_{1,n}, \ldots, X_{\lfloor \tau n \rfloor, n} \sim \mathbb{P}_1$ *and* $X_{\lfloor \tau n \rfloor + 1, n}, \ldots, X_{n,n} \sim \mathbb{P}_2$ *for some* $\tau \in (0,1)$ *and distributions* $\mathbb{P}_1 \neq \mathbb{P}_2$ *with densities* $p_1, p_2$*. Let* $s \in (0,1),\ \eta > 0$ *and* $\hat{p}$ *be a binary classifier that for training observations* $\mathbf{Z}_n = (X_{i,n}, \iota_{\{0, \lfloor sn \rfloor, n\}}(i))_{i=1}^n \subset \mathbb{R}^p \times \{1,2\}$ *consistently estimates* $\mathbb{P}(Y = 1 \mid X)$*. Then*

$$\hat{\alpha}_n \coloneqq \frac{1}{n} \operatorname*{arg\,max}_{j=1,\ldots,n} \sum_{i=1}^{j} \log_\eta \left( \frac{1}{s} \hat{p}_{\mathbf{Z}_n}(X_{i,n}) \right) + \sum_{i=j+1}^{n} \log_\eta \left( \frac{1}{1-s} (1 - \hat{p}_{\mathbf{Z}_n}(X_{i,n})) \right)$$

*is a consistent estimate of* $\tau$*, i.e.*

$$\mathbb{P}(|\hat{\alpha}_n - \tau| > \varepsilon) \overset{n \to \infty}{\to} 0.$$

*Proof.* See Appendix. □

Theorem 4 shows that, independent of the initial split point, Algorithm 1 consistently recovers the change point $\lfloor \tau n \rfloor$ already in the first step. However, the second step improves the estimation performance in practice and might be necessary to obtain a better estimate for the true maximal gain.

In practice we combine the Two Step Algorithm with SBS and a greedy selection of maximal gains. Lemma 3 can be generalized to multiple change point scenarios. The resulting expected curve is still piece wise linear, but might be almost flat in some segments. See the left of Figure 3 for an example. If the maximal gain is in such a flat region, the estimated split might lie anywhere between the two underlying change points. For consistency results in this scenario, one would need a smarter selection approach, which guarantees selection based on segments with a single change point. Baranowski et al. (2019) proposes to select the split based on the narrowest segment with a maximal gain over a predefined threshold. This cannot be easily expanded to our setting, as such a threshold is difficult to obtain.

# 4 Simulations

We present results from a simulation study comparing our classifier based methods to available non-parametric competitors.

## 4.1 Competing Methods

The methods for non-parametric change point detection we are aware of are *ECP* of Matteson and James (2014), the *MultiRank* procedure of Lung-Yut-Fong et al. (2011), the Graph-based Segmentation method *gSeg* of Chen and Zhang (2015) and Kernel change point analysis methods such as *KernSeg* of Celisse et al. (2018).

ECP searches for a single change point based on energy distances and checks for its significance with a permutation test. This generalizes to multiple change point scenarios through binary segmentation. An implementation of ECP is available through the R package **ecp** (James and Matteson, 2015).

MultiRank utilizes a rank based multivariate homogeneity statistic combined with dynamic programming. Significance of found change points are assessed using asymptotic theory. We used the code available in the supplementary material of Matteson and James (2014) to run simulations with MultiRank.

Kernel change point methods minimize a within segment average dissimilarity measure, where dissimilarities are computed using a (Gaussian) kernel. Celisse et al. (2018) propose the KernSeg algorithm, a fast dynamic programming approach. While the authors discuss asymptotic model selection, they do not provide a method in their R package **KernSeg**.

The Graph-based change point detection method gSeg constructs a graph with the observations as nodes and finds a single change point based on edges between segments. The authors do not present

a method for multiple change point detection. An implementation of gSeg is available through the R package **gSeg**. We use the minimal spanning tree as a graph for our simulations.

Lastly, we compare our method to a parametric one that was proposed by Londschien et al. (2019) for (high-dimensional) Gaussian graphical models based on the graphical Lasso (glasso, Friedman et al. (2008)). This can also be used to find changes in mean or covariance in low-dimensional Gaussian data.

Our kNN and Random Forest based methods *kNNcd* and *RFcd* are implemented in the R-package **hdcd** for high-dimensional change point detection that also implements the methods of Londschien et al. (2019), to be made available on CRAN. We restrict the algorithms to only consider segments of length at least $\lceil \delta n \rceil$, choosing $\delta = 0.1$ in all except the Wine 2 scenario, where we select $\delta = 0.05$. We restrict all other methods in a similar fashion where possible. We select a value of $\eta = e^{-6}$. In SBS, we use a decay rate of $\frac{1}{\sqrt{2}}$ for the construction of seeded intervals, resulting in 31 intervals for the scenarios with $\delta = 0.1$ and 69 intervals with $\delta = 0.05$. We furthermore add the current analyzed segment to the predrawn seeded intervals in each step of BS. We select the number of neighbors for kNN to be $\lceil (v-u)^{\frac{1}{2}} \rceil$ when analyzing segments $(u, v]$. For Random Forests we use the fast implementation **ranger** by Wright and Ziegler (2017) with parameters chosen as proposed by Malley et al. (2012), optimized to estimate probabilities. For the estimate of class proportions when using OOB predictions we correct for the effective number of observations, i.e. use $\frac{s-u-1}{v-u-1}$ and $\frac{v-s}{v-u-1}$ for observations in the segment $(u, s]$ and $\frac{s-u}{v-u-1}$ and $\frac{v-s-1}{v-u-1}$ for observations in the segment $(s, v]$.

For the permutation based model selection we perform 400 and 2000 permutations with a significance level of 0.05 and 0.02 for kNNcd and RFcd respectively, as additional permutations for kNNcd are relatively expensive and as we assume RFcd to underestimate $p$-values. All other methods were used with their default settings.

## 4.2 Setups

We analyze the performance of our classifier based methods and competitors in a variety of multivariate setups. We include some of the setups analyzed in Matteson and James (2014), one setup motivated by a simulation from Chen and Zhang (2015) and a high-dimensional setup from Londschien et al. (2019). All of these simulation setups draw from multivariate Gaussians with changes in mean or covariance. To simulate truly non-parametric scenarios, we use publicly available labeled real data, namely the Iris (Fisher, 1936) and Wine Quality (Cortez et al., 2009) data sets.

All setups we include from Matteson and James (2014) consider three equally sized segments with 100 observations drawn from multivariate Gaussian distributions. The observations in the outer (first and third) segments are always drawn from $\mathcal{N}(0, I_d)$ while those of the middle (second) come from a Gaussian $G$ with a shift in the mean or the variance. In the *Change in Mean (CIM)* setup $d = 2$ and $G = \mathcal{N}((1, 1)^T, I_2)$. In the *Change in Covariance (CIC)* setup again $d = 2$ but $G = \mathcal{N}(0, \Sigma)$ with $\Sigma_{1,1} = \Sigma_{2,2} = 1$ and $\Sigma_{1,2} = 0.5$. Lastly, in the challenging *Noise* setup $d = 5$ and $G = \mathcal{N}(0, I_5 + N)$ with $N_{1,2} = N_{2,1} = 0.9$ and zero for all other entries.

For the *Random Network (RN)* setup of Londschien et al. (2019), we create data sets of 500 observations in total, with segments of lengths $70, 120, 120$ and $190$. We randomly select the order of the segments and for each segment draw the inverse covariance matrix from a 100-dimensional Random Network structure, see Londschien et al. (2019) for details. Note that this is a truly high-dimensional setup, as the number of covariates (100) is higher than the number of observations in the smallest segment (70).

For the *Iris* setup we use the Iris data set containing measurements of three species of the flower iris with each 50 observations of four characteristics. To increase difficulty we centere the observations from each group separately, randomly arrange the three species as segments and shuffle the in segment observations. For added complexity we additionally add two noise variables, one from a Gaussian and one from a univariate distribution. Lastly we center and standardize the resulting data set, as this matters for distance based competitors.

The Wine Quality data set contains results of 11 chemical analyses of red and white wines from northern Portugal and a sensory quality variable. Except for the color (which we encode binary), each variable is continuous. In the *Wine 1* setup we randomly arrange 4 segments of lengths $70, 120, 120$ and $190$ and sample wine measurement from one of the qualities 4 - 8 for each segment without replacement. In the *Wine 2* setup, we increase the number of segments to 7, randomly arranging segments of lengths $70, 70, 120, 120, 120, 190$ and $310$ for a total of 1000 observations. To each segment we randomly assign

a quality of 5 - 7 (the most prevalent), not allowing for equal qualities in neighboring segments. This setup is considerably more difficult than Wine 1, as the minimal relative segment length is 0.07, some non-neighboring segments correspond to the same distribution and the shifts in distribution are smaller. We again standardize to obtain the final Wine 1 and Wine 2 data sets.

Multiple change point scenarios represent realistic settings similar to what could be expected in reality. However, the performance of the analyzed methods depends highly on good model selection. Since the KernSeg method does not include a model selection method and as the gSeg method can only be applied to scenarios with a single change point, these methods cannot be compared in the former simulation setups. To disentangle the model selection performance from the change point location estimation and to compare KernSeg and gSeg to our and other competing methods, we provide two setups with a single change point.

Both setups comprise of two segments with 80 and 120 observations each. In the $G_1$ setup, motivated by the setup used by Chen and Zhang (2015) for power comparisons, observations in the first segment are drawn from $\mathcal{N}(0, I_{20})$ and observations of the second segment have their mean shifted by 0.2 and the variance of the first component is increased to $20^{\frac{1}{3}} \approx 2.7$. In the $W_1$ setup, we randomly draw measurements of white wine for the first and of red wine for the second segment (irrespective of quality). We center and standardize each segment separately.

Lastly, we create setups to analyze the behavior of our methods in the absence of change points. For the $G_0$ setup 200 observations are randomly sampled from $\mathcal{N}(0, I_{20})$ and for the $W_0$ setup we randomly sample 200 observations from the Wine data set.

## 4.3 Performance Measures

Our main performance measure is the adjusted Rand Index (Hubert and Arabie, 1985), a common measure to compare clusterings. Given two partitions of $n$ observations, the Rand Index (Rand, 1971) is the number of agreements (pairs of observations that are either in the same subset for both partitions or are in different subsets for both partitions) divided by the total number of pairs $\binom{n}{k}$. The adjusted Rand Index is the normalized difference between the Rand Index and its expectation when choosing partitions randomly. The adjusted Rand Index is bounded by one and expected to be zero when choosing partitions randomly. Note that Matteson and James (2014) reported Rand Indices rather than the adjusted version in their simulation results.

We additionally use the Hausdorff distance (see e.g. Truong et al. (2019, Section 3.2.2)). For two non-trivial segmentations $\alpha, \alpha'$, set $d(\alpha, \alpha') \coloneqq \max_{1 \leqslant k \leqslant |\alpha|-1} \min_{1 \leqslant k' \leqslant |\alpha'|-1} |\alpha_k - \alpha'_k|/n$ and furthermore define $d(\alpha, \{0, n\}) \coloneqq 1$. Thus $d(\alpha^0, \hat{\alpha})$ measures the biggest relative distance of a true change point of $\alpha^0$ to the closest in $\hat{\alpha}$ and $d(\hat{\alpha}, \alpha^0)$ the biggest relative distance of a found change point in $\hat{\alpha}$ to the closest true change point. The former penalizes undersegmentation while the latter penalizes oversegmentation. Define the Hausdorff distance $d_H(\alpha, \alpha') \coloneqq \max(d(\alpha, \alpha'), d(\alpha', \alpha))$.

## 4.4 Results

We first present simulation results of multiple change point setups for methods that allow for detection of multiple change points including model selection. Tables 1 and 2 show the average adjusted Rand indices and Hausdorff distances between true and estimated change points over 500 simulation runs.

| method | CIM | CIC | Noise | RN | Iris | Wine 1 | Wine 2 |
|---|---|---|---|---|---|---|---|
| ECP | 0.96 | 0.04 | 0.03 | 0.30 | 0.01 | 0.85 | 0.82 |
| MultiRank | 0.97 | 0.33 | 0.34 | 0.30 | 0.28 | 0.75 | 0.72 |
| glasso | 0.83 | 0.77 | 0.83 | 0.96 | 0.72 | 0.54 | 0.35 |
| RFcd | 0.93 | 0.30 | 0.64 | 0.91 | 0.87 | 0.94 | 0.92 |
| kNNcd | 0.92 | 0.52 | 0.55 | 0.74 | 0.51 | 0.86 | 0.85 |

Table 1: Average adjusted Rand Indices

Our RFcd method outperforms all other methods in all non-parametric setups (Iris, Wine 1 and Wine 2) and performs well in the high-dimensional RN setup, where only the glasso method that is

| method | CIM | CIC | Noise | RN | Iris | Wine 1 | Wine 2 |
|---|---|---|---|---|---|---|---|
| ECP | 0.01 | 0.94 | 0.94 | 0.61 | 0.98 | 0.09 | 0.12 |
| MultiRank | 0.01 | 0.34 | 0.34 | 0.40 | 0.36 | 0.20 | 0.19 |
| glasso | 0.13 | 0.14 | 0.13 | 0.03 | 0.16 | 0.34 | 0.50 |
| RFcd | 0.03 | 0.62 | 0.26 | 0.05 | 0.07 | 0.04 | 0.04 |
| kNNcd | 0.05 | 0.39 | 0.36 | 0.17 | 0.32 | 0.10 | 0.09 |

Table 2: Average Hausdorff distances between found and true underlying change points

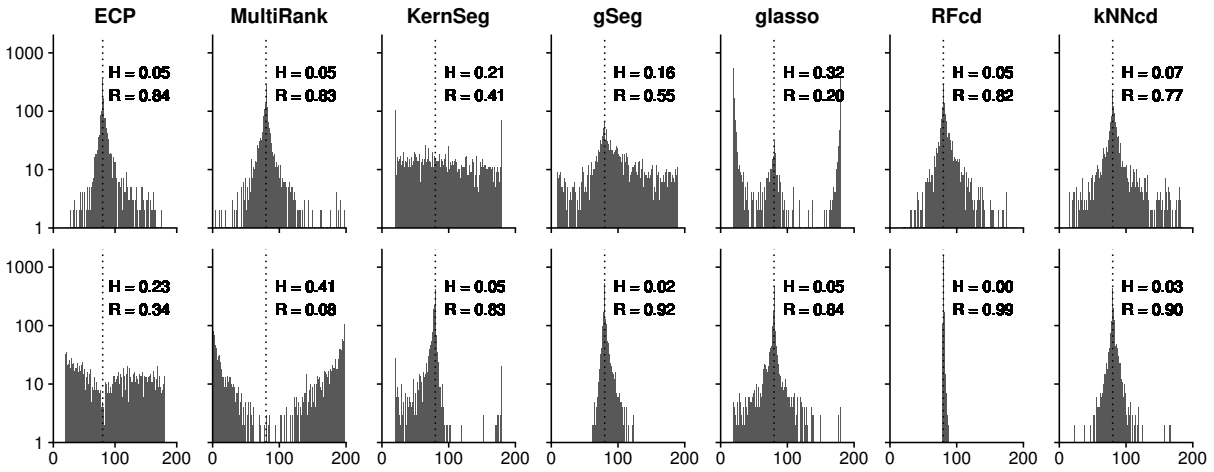


Figure 1: Histograms of single change point estimates for $G_1$ (top) and $W_1$ (bottom) setups for 2000 simulations each. Note that the $y$-axis is on a log-scale. In each setup, the resulting average Hausdorff distance (H) and average adjusted Rand Index (R) are displayed. The vertical dashed lines mark the location of the true underlying change point.

custom built for this scenario yields slightly better results. The MultiRank and ECP methods slightly outperform our method in the low-dimensional CIM setup, which we expect is due to the high overlap between the distributions. While no non-parametric method achieves good performance for the CIC and Noise setups, none of our methods is significantly outperformed by any of the competing non-parametric methods. Only glasso returns reasonable results, which is not surprising, as changes lie only in the covariance between two covariates.

Figure 1 shows histograms of the locations of the first found change points in the $G_1$ and $W_1$ setups. For the $G_1$ setup both the ECP and MultiRank methods perform well, similar to our RFcd method and outperforming the kNNcd method. The KernSeg and gSeg methods fail to produce any sensible results. This is reversed for the $W_1$ setup, where ECP and MultiRank do not detect any signal in the data while both KernSeg and gSeg produce usable results. However, none of the methods perform as well as the RFcd method, which finds exactly the split point in 1627 out of 2000 simulation runs.

Lastly, we present a summary of the false positive rates of the methods capable of model selection in Table 3. We additionally display ROC curves comparing the the power of the methods in selecting the change points in the $W_1$ and $G_1$ setups to the respective false positive rates of $W_0$ and $G_0$, given different thresholds for $p$-values.

| method | $G_0$ | $W_0$ |
|---|---|---|
| ECP | 0.050 | 0.048 |
| MultiRank | 1.000 | 1.000 |
| glasso | 0.000 | 0.181 |
| RFcd | 0.046 | 0.103 |
| kNNcd | 0.056 | 0.052 |

Table 3: Percentage of simulations where the respective methods falsely found a change point

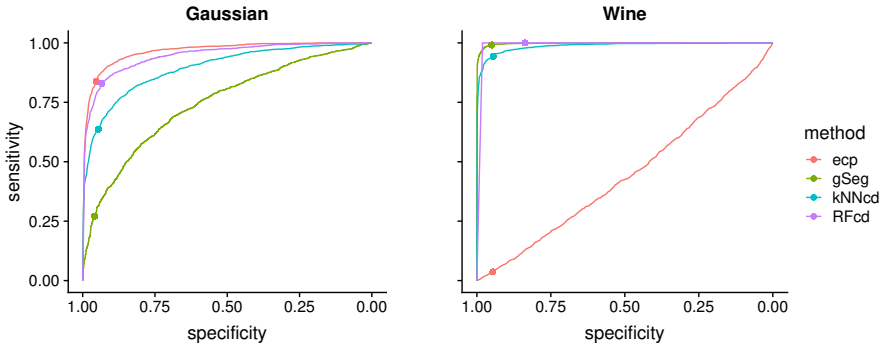


Figure 2: ROC curves comparing specificity against sensitivity between the $G_0$ and $G_1$ (left) and $W_0$ and $W_1$ (right) setups. The threshold used in simulations is marked with a dot.

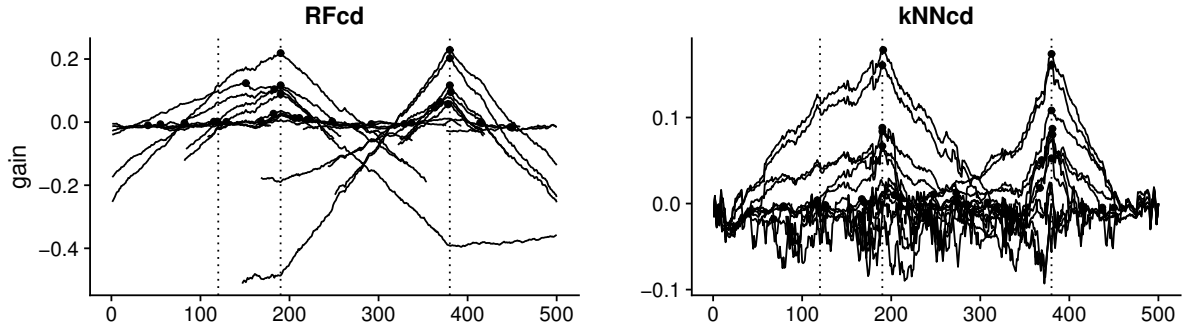


Figure 3: Gain curves of the RFcd and kNNcd methods for all seeded intervals when applied to a simulated Wine 1 dataset. The locations of the true change points are marked with dotted vertical lines and points mark the maxima of each gain curve.

The ECP and kNNcd methods return the expected false positive rate of 5% for both setups, as they are based on exact permutation tests. The RFcd method performs differently for the $G_0$ versus the $W_0$ setup, selecting change points in 5 respectively 10% of the simulations. The glasso method does not select any change point in the Gaussian setup $G_0$, where its parametric assumptions are met, but overfits in the non-parametric setup $W_0$. Lastly, the MultiRank methods seems to overfit significantly, selecting a change point in each of the simulations.

Lastly, we would like to comment on the computational times. In the Iris data set the MultiRank and ECP methods required about 0.1 - 0.2 seconds for estimation, while our classifier based methods took around 5 seconds. In the Wine 2 data set ECP took around 15, MultiRank and RFcd around 40 and kNNcd around 200 seconds on single Intel Xenon 3.0 GHz processor cores. These two setups were the fastest and slowest across our simulations.

# 5 Conclusion

We presented a novel methodology that is able to transfer the flexibility of modern classification algorithms to the setting of change point detection. The flexibility of such algorithms often comes at the cost of being resource intensive, making specific optimization approaches necessary. We provided feasible algorithms, motivated by theoretical results, which perform very well in practice when paired with Random Forest classifiers.

Our algorithms are available through the R-package **hdcd** available on CRAN, which we hope will enable its applicability for practitioners.

# A Proofs

**Lemma 1.** *Let*

$$\hat{p}_\alpha \coloneqq \mathbb{P}\left(Y = k \mid X\right) = \left(\frac{\alpha_{k\text{-}1} - \alpha_k}{n} \frac{p_{(\alpha_{k\text{-}1},\alpha_k]}}{p_{(0,n]}}\right)_{k=1}^{|\alpha|-1}$$

*be the Bayes classifier corresponding to the segmentation $\alpha$. If $\alpha^0 \in \mathcal{A}$,*

$$\begin{aligned}\max_{\alpha\in\mathcal{A}} \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha, \hat{p}_\alpha)\right)\right] &= \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha^0, \hat{p}_{\alpha^0})\right)\right] \\ &= \sum_{k=1}^{K} \frac{\alpha_k^0 - \alpha_{k\text{-}1}^0}{n} D_{KL}(\mathbb{P}_k \parallel \mathbb{P}_{(0,n]}) \end{aligned} \tag{2.6}$$

*and any maximizer of (2.6) is a subsegmentation of $\alpha^0$.*

*Proof.* This follows from the fact that the continuous extension of $x \mapsto x\log(x)$ is strictly convex on $[0,\infty)$. In particular, for any $u < v$ such that that $\kappa \coloneqq \{k : (\alpha_{k\text{-}1}^0, \alpha_k^0] \cap (u,v] \neq \varnothing\}$ contains more than one element, setting $\pi_k \coloneqq \frac{|(\alpha_{k\text{-}1}^0, \alpha_k^0] \cap (u,v]|}{|(u,v]|}$ for $k \in \kappa$:

$$\begin{aligned}\mathbb{E}\left[\sum_{i=u+1}^{v} \log\left(\frac{p_{(u,v]}(X_i)}{p_{(0,n]}(X_i)}\right)\right] &= (v-u)\mathbb{E}_{X\sim\mathbb{P}_{(u,v]}}\left[\log\left(\frac{p_{(u,v]}(X)}{p_{(0,n]}(X)}\right)\right] \\ &= (v-u)\mathbb{E}_{X\sim(\sum_{k\in\kappa}\pi_k\mathbb{P}_k)}\left[\log\left(\frac{\sum_{k\in\kappa}\pi_k p_k(X)}{p_{(0,n]}(X)}\right)\right] \\ &< (v-u)\sum_{k\in\kappa}\pi_k\mathbb{E}_{X\sim\mathbb{P}_k}\left[\log\left(\frac{p_k(X)}{p_{(0,n]}(X)}\right)\right] \\ &= (v-u)\sum_{k\in\kappa}\pi_k D_{KL}(\mathbb{P}_k \parallel \mathbb{P}_{(0,n]}),\end{aligned}$$

where we applied the strict convexity in the third line.

Let $\alpha$ be any segmentation of $\{0,\dots,n\}$ not containing $\alpha^0$. Then, applying the above to all relevant segments $(\alpha_{k\text{-}1}, \alpha_k]$ we get that

$$\begin{aligned}\mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha, \hat{p})\right)\right] &< \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha\cup\alpha^0, \hat{p})\right)\right] \\ &= \sum_{k=1}^{K} \frac{\alpha_k^0 - \alpha_{k\text{-}1}^0}{n} D_{KL}(\mathbb{P}_k \parallel \mathbb{P}_{(0,n]}) \\ &= \mathbb{E}\left[G_n\left((X_i)_{i=1}^n \mid (\alpha^0, \hat{p})\right)\right].\end{aligned}$$

□

**Lemma 2.** *For the Bayes classifier $\hat{p}_{\{u,s,v\}}$ the expected gains curve*

$$s \mapsto \mathbb{E}\left[G_n\left((X_i)_{i=u+1}^v \mid (\{u,s,v\}, \hat{p}_{\{u,s,v\}})\right)\right]$$

*is piece wise convex between the underlying change points, with strict convexity in the $k$-th segment if $\mathbb{P}_{(u,\alpha_{k-1}]} \neq \mathbb{P}_k$ or $\mathbb{P}_k \neq \mathbb{P}_{(\alpha_k,v]}$.*

*Proof.* Let $1 \leqslant k \leqslant K$ and $\alpha_{k-1} + 1 < s < \alpha_k$. We show that $\mathbb{E}[G(s+1) - 2G(s) + G(s-1)] \geqslant 0$ with equality if and only if $\mathbb{P}_{(0,\alpha_k-1]} = \mathbb{P}_k = \mathbb{P}_{(\alpha_k,n]}$.

For this rewrite

$$G(s) \coloneqq G_n^{(u,v]}(s) = \frac{1}{n}\left(\sum_{i=1}^{s}\log\left(\frac{p_{(0,s]}(X_i)}{p_{(0,n]}(X_i)}\right) + \sum_{i=s+1}^{n}\log\left(\frac{p_{(s,n]}(X_i)}{p_{(0,n]}(X_i)}\right)\right) \tag{A.1}$$

to obtain

$$n\left(G(s+1)-2G(s)+G(s-1)\right)=\sum_{i=1}^{s+1}\log\left(\frac{p_{(0,s+1]}(X_i)}{p_{(0,s]}(X_i)}\right)+\sum_{i=1}^{s-1}\log\left(\frac{p_{(0,s-1]}(X_i)}{p_{(0,s]}(X_i)}\right)+\sum_{i=s+2}^{n}\log\left(\frac{p_{(s+1,n]}(X_i)}{p_{(s,n]}(X_i)}\right)+\sum_{i=s}^{n}\log\left(\frac{p_{(s-1,n]}(X_i)}{p_{(s,n]}(X_i)}\right)+\log\left(\frac{p_{(s,n]}(X_{s-1})}{p_{(s,n]}(X_{s+1})}\right)+\log\left(\frac{p_{(s,n]}(X_{s+1})}{p_{(s,n]}(X_{s-1})}\right)$$

Since $X_{s-1}, X_s$ and $X_{s+1}$ are i.i.d. the last two terms have expectation zero. Thus taking expectations, it follows directly that

$$\mathbb{E}\left[n\left(G(s+1)-2G(s)+G(s-1)\right)\right]=(s+1)D_{KL}(\mathbb{P}_{(0,s+1]}\parallel\mathbb{P}_{(0,s]})+(s-1)D_{KL}(\mathbb{P}_{(0,s-1]}\parallel\mathbb{P}_{(0,s]})+(n-s-1)D_{KL}(\mathbb{P}_{(s+1,n]}\parallel\mathbb{P}_{(s,n]})+(n-s+1)D_{KL}(\mathbb{P}_{(s-1,n]}\parallel\mathbb{P}_{(s,n]})\geqslant 0,$$

with equality if and only if $\mathbb{P}_{(0,\alpha_k-1]}=\mathbb{P}_k=\mathbb{P}_{(\alpha_k,n]}$. □

**Lemma 3.** *Consider a change point setup with a single change point $\alpha_1^0$ in the segment $(u,v]$. Then, for any initial split $s\in(u+1,v-1]$ and the Bayes classifier $\hat{p}=\frac{s-u}{v-u}\frac{p_{(u,s]}}{p_{(u,v]}}$, if finite, the function*

$$j\mapsto\mathbb{E}\left[\sum_{i=u+1}^{j}\log\left(\frac{v-u}{s-u}\hat{p}(X_i)\right)+\sum_{i=j+1}^{v}\log\left(\frac{v-u}{v-s}(1-\hat{p}(X_i))\right)\right]$$

*is piece wise linear with the maximum at $j=\alpha_1^0$.*

*Proof.* Write

$$G(j)\coloneqq\sum_{i=u+1}^{j}\log\left(\frac{v-u}{s-u}\hat{p}(X_i)\right)+\sum_{i=j+1}^{v}\log\left(\frac{v-u}{v-s}(1-\hat{p}(X_i))\right),\tag{A.2}$$

and define

$$U_j\coloneqq G(j)-G(j-1)=\log\left(\frac{p_{(u,s]}(X_j)}{p_{(s,v]}(X_j)}\right),$$

such that $G(j)=G(u)+\sum_{i=u+1}^{j}U_i$. Assume wlog. that $s\geqslant\alpha_1^0$. Then $p_{(u,s]}=\frac{\alpha_1^0-u}{s-u}p_1+\frac{s-\alpha_1^0}{s-u}p_2$ and $p_{(s,v]}=p_2$ and for $i\leqslant\alpha_1^0$:

$$\begin{aligned}\mathbb{E}[U_i]&=\mathbb{E}_{\mathbb{P}_1}\left[\log\left(\frac{p_{(u,s]}(X)}{p_{(s,v]}(X)}\right)\right]\\&=\frac{s-u}{\alpha_1^0-u}\mathbb{E}_{\mathbb{P}_{(u,s]}}\left[\log\left(\frac{p_{(u,s]}(X)}{p_{(s,v]}(X)}\right)\right]-\frac{s-\alpha_1^0}{s-u}\mathbb{E}_{\mathbb{P}_2}\left[\log\left(\frac{p_{(u,s]}(X)}{p_{(s,v]}(X)}\right)\right]\\&=\frac{s-u}{\alpha_1^0-u}D_{KL}(\mathbb{P}_{(u,s]}\parallel\mathbb{P}_2)+\frac{s-\alpha_1^0}{s-u}D_{KL}(\mathbb{P}_2\parallel\mathbb{P}_{(u,s]})>0.\end{aligned}$$

Similarly, for $i>\alpha_1^0$

$$\mathbb{E}[U_i]=\mathbb{E}_{\mathbb{P}_2}\left[\log\left(\frac{p_{(u,s]}(X)}{p_{(s,v]}(X)}\right)\right]=-D_{KL}(\mathbb{P}_2\parallel\mathbb{P}_{(u,s]})<0,$$

Which implies the statement. □

**Lemma 5.** *Let $\mathbb{P}_1, \mathbb{P}_2$ be probability measures with corresponding densities $p_1, p_2$. Let $\delta_1, \delta_2 \in [0,1]$ and $\delta_1 \neq \delta_2$, write $\mathbb{P}_1(\delta_1) \coloneqq (1-\delta_1)\mathbb{P}_1 + \delta_1 \mathbb{P}2$, $\mathbb{P}_2(\delta_2) \coloneqq (1-\delta_2)\mathbb{P}_1 + \delta_2\mathbb{P}_2$ and let $p_1(\delta_1), p_2(\delta_2)$ be their respective p.d.f.'s. Then*

$$\mathbb{E}_{\mathbb{P}_1}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right] = \frac{\delta_2}{\delta_2-\delta_1} D_{KL}(\mathbb{P}_1(\delta_1) \parallel \mathbb{P}_2(\delta_2)) + \tag{A.3}$$

$$\frac{\delta_1}{\delta_2-\delta_1} D_{KL}(\mathbb{P}_2(\delta_2) \parallel \mathbb{P}_1(\delta_1)). \tag{A.4}$$

*In particular, when $\delta_1 < \delta_2$, this expectation is greater than zero.*

*Proof.* The Kullback Leibler divergence between distribution $Q_1, Q_2$ with densities $q_1, q_2$ is $D_{KL}(Q_1 \parallel Q_2) = \mathbb{E}_{Q_1}[\log\left(\frac{q_1(X)}{q_2(X)}\right)]$. Thus

$$D_{KL}(\mathbb{P}_1(\delta_1) \parallel \mathbb{P}_2(\delta_2)) = (1-\delta_1)\mathbb{E}_{P_1}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right] + \delta_1 \mathbb{E}_{P_2}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right]$$

and

$$D_{KL}(\mathbb{P}_2(\delta_2) \parallel \mathbb{P}_1(\delta_1)) = (1-\delta_2)\mathbb{E}_{P_1}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right] + \delta_2 \mathbb{E}_{P_2}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right]$$

such that

$$\begin{aligned}
&\frac{\delta_2}{\delta_2-\delta_1} D_{KL}(\mathbb{P}_1(\delta_1) \parallel \mathbb{P}_2(\delta_2)) + \frac{\delta_1}{\delta_2-\delta_1} D_{KL}(\mathbb{P}_2(\delta_2) \parallel \mathbb{P}_1(\delta_1)) \\
&\quad = \left(\frac{\delta_2}{\delta_2-\delta_1}(1-\delta_1) - \frac{\delta_1}{\delta_2-\delta_1}(1-\delta_2)\right)\mathbb{E}_{P_1}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right] + \\
&\qquad \left(\frac{\delta_2}{\delta_2-\delta_1}\delta_1 - \frac{\delta_1}{\delta_2-\delta_1}\delta_2\right)\mathbb{E}_{P_2}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right] = \mathbb{E}_{\mathbb{P}_1}\left[\log\left(\frac{p_1(\delta_1)}{p_2(\delta_2)}\right)\right].
\end{aligned}$$

□

**Lemma 6.** *Let $X_{i,n}, n \geqslant 0, i = 1, \ldots, n$ be a triangular array of centered random variables. Let $\tau \in [0,1]$ and assume that for $n \geqslant 0$ the sequences $(X_{i,n})_{i=1}^{\lfloor \tau n \rfloor}$ and $(X_{i,n})_{i=\lfloor \tau n \rfloor + 1}^{n}$ are i.i.d. with variances $\sigma_1^2$ and $\sigma_2^2$ respectively. Define $\sigma \coloneqq \sqrt{\tau\sigma_1^2 + (1-\tau)\sigma_2^2}$ and let $S_{i,n} \coloneqq X_{1,n} + \cdots + X_{i,n}$. Then, for any $\alpha > 0$ :*

$$\mathbb{P}\left(\max S_{1,n}, \ldots, S_{n,n} < \alpha\sqrt{n}\sigma\right) \overset{n\to\infty}{\to} 2 - 2\Phi(\alpha) \coloneqq \sqrt{\frac{2}{\pi}}\int_0^\alpha \exp\left\{\frac{u^2}{2}\right\}\mathrm{d}u$$

*Proof.* This proof is a slight alteration of the one presented by Erdős and Kac (1946), allowing for random variables of differing variances. The idea of the proof is to show that the limiting distribution exists and is independent of the individual distributions of the random variables $X_{i,n}$.

Let $(G_i)_{i=1,2,\ldots}$ be a sequence of independent standard Gaussian random variables, let $R_k \coloneqq G_1 + \ldots + G_k$ and define $P_n(\alpha) \coloneqq \mathbb{P}(\max S_{1,n}, \ldots, S_{n,n} < \alpha\sqrt{n}\sigma_n)$. We would like to show that for any integer $k > 0$ and real numbers $\varepsilon, \alpha > 0$ we have that

$$\begin{aligned}
&\mathbb{P}\left(\max R_1, \ldots, R_k > (\alpha-\varepsilon)\sqrt{k}\right) - \frac{1}{\varepsilon^2 k} \leqslant \liminf_{n\to\infty} P_n(\alpha) \\
&\quad \leqslant \limsup_{n\to\infty} P_n(\alpha) \leqslant \mathbb{P}\left(\max R_1, \ldots, R_k > \alpha\sqrt{k}\right).
\end{aligned} \tag{A.5}$$

Define $t \coloneqq \frac{\tau\sigma_1^2}{\sigma^2}$ and let $f\colon [0,1] \to [0,1],\ x \mapsto \mathbb{1}_{[0,t]}(x)\frac{\sigma^2}{\sigma_1^2}x + \mathbb{1}_{(t,1]}(x)(\tau + \frac{\sigma^2}{\sigma_2^2}(x-t))$. Then $f$ is piece wise linear with $f(0) = 0, f(t) = \tau, f(1) = 1$ and for any $0 < x_1 < t < x_2 < 1 : \sigma_1^2 f'(x_1) = \sigma_2^2 f'(x_2) = \sigma^2$. For $j = 1, \ldots, k$ let $i_{j,n} \coloneqq \lfloor nf(\frac{j}{k})\rfloor$ such that $\mathrm{Var}(S_{i_{j,n}} - S_{i_{j-1,n}}) \approx \frac{n}{k}\sigma^2$. Then, by the Central Limit Theorem

$$P_{n,k}(\alpha) \coloneqq \mathbb{P}(\max S_{i_{1,n},n}, \ldots, S_{i_{k,n},n} < \alpha\sqrt{n}\sigma_n) \overset{n\to\infty}{\to} \mathbb{P}(\max R_1, \ldots, R_k > \alpha\sqrt{k}).$$

Define for $i = 1, \ldots, n$ the events

$$E_{i,n}(\alpha) \coloneqq \{S_{i,n} \geqslant \alpha\sqrt{n}\sigma,\ S_{i-1,n} < \alpha\sqrt{n}\sigma, \ldots, S_{1,n} < \alpha\sqrt{n}\sigma\}$$

such that $\sum_{i=1}^{n} \mathbb{P}(E_{i,n}(\alpha)) = 1 - P_n(\alpha) \leqslant 1$. Note that for $i_{j,n} \leqslant i < i_{j+1,n}$ the event $\{|S_{i_{j+1,n}} - S_i| > \varepsilon\sigma\sqrt{n}\}$ is independent of $E_{i,n}$ and by Tchebyshev's inequality has probability not exceeding $\frac{1}{k\varepsilon^2}$. Furthermore observe that each of the events $E_{i,n}(\alpha) \cap \{|S_{i_{j+1,n}} - S_i| \leqslant \varepsilon\sigma\sqrt{n}\}$ is contained in the event that at least one of the $S_{i_{j,n},n}$ is greater than $(\alpha\ -\varepsilon)\sqrt{n}\sigma$. Thus

$$1 - \mathbb{P}_n(\alpha) = \sum_{i=1}^{n} \mathbb{P}(E_{i,n}(\alpha)) \leqslant \frac{1}{k\varepsilon^2} + 1 - P_{n,k}(\alpha - \varepsilon).$$

Since $P_{k,n}(\alpha) \leqslant P_n(\alpha)$ it follows that $P_{n,k}(\alpha - \varepsilon) - \frac{1}{k\varepsilon^2} \leqslant P_n(\alpha) \leqslant P_{n,k}(\alpha)$ implying Equation (A.5) by letting $n \to \infty$.

Let $(B_t)_{t\geqslant 0}$ be a Brownian Motion. Then one can show for $a > 0$ using the reflection principle that $\mathbb{P}(\max_{0\leqslant s\leqslant t} B_s > a) = 2\mathbb{P}(B_t > 0)$. Viewing $(R_i)_{i\geqslant 1}$ as a discretisation of $(B_t)_{t\geqslant 0}$, it follows that

$$\mathbb{P}(\max R_1, \ldots, R_k > \alpha\sqrt{k}) \overset{k\to\infty}{\to} 2\mathbb{P}(R_k > \alpha\sqrt{k}) = 2 - 2\Phi(\alpha).$$

The result then follows by letting $k \to \infty$ and using the fact that $\Phi$ is continuous. □

**Theorem 4.** *Consider a triangular array of random variables $X_{i,n}$, $n \geqslant 0$, $i = 1, \ldots, n$ such that $X_{1,n}, \ldots, X_{\lfloor \tau n \rfloor, n} \sim \mathbb{P}_1$ and $X_{\lfloor \tau n \rfloor + 1, n}, \ldots, X_{n,n} \sim \mathbb{P}_2$ for some $\tau \in (0,1)$ and distributions $\mathbb{P}_1 \neq \mathbb{P}_2$ with densities $p_1, p_2$. Let $s \in (0,1)$, $\eta > 0$ and $\hat{p}$ be a binary classifier that for training observations $\mathbf{Z}_n = (X_{i,n}, \iota_{\{0,\lfloor sn \rfloor, n\}}(i))_{i=1}^{n} \subset \mathbb{R}^p \times \{1,2\}$ consistently estimates $\mathbb{P}(Y = 1 \mid X)$. Then*

$$\hat{\alpha}_n \coloneqq \frac{1}{n} \operatorname*{arg\,max}_{j=1,\ldots,n} \sum_{i=1}^{j} \log_\eta\left(\frac{1}{s}\hat{p}_{\mathbf{Z}_n}(X_{i,n})\right) + \sum_{i=j+1}^{n} \log_\eta\left(\frac{1}{1-s}(1 - \hat{p}_{\mathbf{Z}_n}(X_{i,n}))\right)$$

*is a consistent estimate of $\tau$, i.e.*

$$\mathbb{P}(|\hat{\alpha}_n - \tau| > \varepsilon) \overset{n\to\infty}{\to} 0.$$

*Proof.* For the following for $0 \leqslant u < v \leqslant 1$ write $p^n_{(u,v]} \coloneqq p_{(\lfloor un \rfloor, \lfloor vn \rfloor]}$ for the density of the mixture distribution of the variables $X_{\lfloor un \rfloor + 1, n}, \ldots, X_{\lfloor vn \rfloor, n}$.

The idea of the proof is as follows: We show, similar as in Lemma 3, that given a perfect classifier $p_n \coloneqq s\frac{p^n_{(0,s]}}{p^n_{(0,1]}}$, the expected gain curve

$$G_n(j) \coloneqq \frac{1}{n}\sum_{i=1}^{j} \log_\eta\left(\frac{1}{s}p_n(X_{i,n})\right) + \sum_{i=j+1}^{n} \log_\eta\left(\frac{1}{1-s}(1 - p_n(X_{i,n}))\right) \tag{A.6}$$

is piece wise linear with the unique maximum at $\alpha^0_1 \coloneqq \lfloor \tau n \rfloor$. As $\hat{p}_{\mathbf{Z}_n}$ is a consistent estimate of $p_n$, the same is true with $\hat{p}_{\mathbf{Z}_n}$ for large enough $n$ with high probability. We then apply Lemma 6 to show that with high probability, the observed gain curve will only deviate on the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$ from its expectation, implying that the location of its maximum $n\hat{\alpha}_n$ will be at most of the order $\mathcal{O}(\sqrt{n})$ away from $\alpha^0_1 = \lfloor \tau n \rfloor$.

Recall that $\log_\eta(x) = \log((1-\eta)x + \eta)$. Set $U_{i,n} \coloneqq G_n(i) - G_n(i-1)$. Then

$$\begin{aligned} U_{i,n} &= \log_\eta\left(\frac{p_{(0,s]}(X_{i,n})}{p_{(0,1]}(X_i)}\right) - \log_\eta\left(\frac{p_{(s,1]}(X_i)}{p_{(0,1]}(X_{i,n})}\right) \\ &= \log\left(\frac{(1-\eta)p_{(0,s]}(X_{i,n}) + \eta p_{(0,1]}(X_i)}{(1-\eta)p_{(s,1]}(X_{i,n}) + \eta p_{(0,1]}(X_{i,n})}\right). \end{aligned}$$

Assume that $s \geqslant \tau$. Then $p_{(0,s]} = \frac{\tau}{s}p_1 + \frac{s-\tau}{s}p_2$, $p_{(s,1]} = p_2$ and $p_{(0,1]} = \tau p_1 + (1-\tau)p_2$. Thus

$$\mathbb{E}[U_{i,n}] = \mathbb{E}\left[\log\left(\frac{((1-\eta)\frac{\tau}{s} + \eta\frac{\tau}{1})p_1(X_i) + ((1-\eta)\frac{s-\tau}{s} + \eta\frac{1-\tau}{n})p_2(X_i)}{\eta\frac{\tau}{1}p_1(X_i) + (1-\eta\frac{\tau}{1})p_2(X_i)}\right)\right].$$

Since $\delta_1 \coloneqq (1-\eta)\frac{s-\tau}{s} + \eta\frac{1-\tau}{1} = \frac{s-\tau}{s} + \eta^{\tau}_{1}\frac{n-s}{s} < \delta_2 \coloneqq 1 - \eta^{\tau}_{1}$, by Lemma 5, this strictly greater than zero for $i \leqslant \lfloor \tau n \rfloor$ and strictly smaller otherwise.

Let $\hat{U}_i$ be the random variable corresponding to $U_i$ when replacing $p_n$ with $\hat{p}_{\mathbf{Z}_n}$. Then, due to the consistency of $\hat{p}$, for $n$ large enough, with high probability, the same result holds, i.e. for some $\delta > 0$, with high probability $\mathbb{E}[\hat{U}_{i,n}] > \delta$ for $i \leqslant \tau$ and $\mathbb{E}[\hat{U}_{i,n}] < \delta$ for $i > \tau$.

Note that $\log_\eta$ is bounded in absolute value by $|\log(\eta)|$, such that the variances of the $U_i$ and $\hat{U}_i$ are bounded by $2|\log(\eta)|$ and we can apply Lemma 6 to the triangular arrays $\hat{U}_{i,n} - \mathbb{E}[\hat{U}_{i,n}]$ and $-(\hat{U}_{i,n} - \mathbb{E}[\hat{U}_{i,n}])$. Thus, again with high probability, the sequence $(\hat{G}_n(i) - \hat{G}_n(i))_{i=1}^n$ varies at most $2\sqrt{n}\alpha|\log(\eta)|$ from its expectation and thus its the location of its maximum varies at most $2\frac{1}{\delta}\sqrt{n}\alpha|\log(\eta)|$ from $\tau n$. □

# References


Baranowski, R., Y. Chen, and P. Fryzlewicz (2019). Narrowest-over-threshold detection of multiple change points and change-point-like features. *Journal of the Royal Statistical Society, Series B 81*(3), 649–672.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Celisse, A., G. Marot, M. Pierre-Jean, and G. Rigaill (2018). New efficient algorithms for multiple change-point detection with reproducing kernels. *Computational Statistics & Data Analysis 128*, 200–220.

Chen, H. and N. Zhang (2015). Graph-based change-point detection. *The Annals of Statistics 43*(1), 139–176.

Cortez, P., A. Cerdeira, F. Almeida, T. Matos, and J. Reis (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems 47*(4), 547–553.

Erdős, P. and M. Kac (1946). On certain limit theorems of the theory of probability. *Bulletin of the American Mathematical Society 52*(4), 292–302.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics 7*(2), 179–188.

Frick, K., A. Munk, and H. Sieling (2014). Multiscale change point inference. *Journal of the Royal Statistical Society, Series B 76*, 495–580.

Friedman, J., T. Hastie, and R. Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*(3), 432–441.

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics 42*(6), 2243–2281.

Garreau, D. and S. Arlot (2018). Consistent change-point detection with kernels. *Electronic Journal of Statistics 12*(2), 4440–4486.

Haubner, L. (2018). Optimistic binary segmentation: A scalable approach to changepoint detection in high-dimensional graphical models. *Master Thesis, ETH Zurich*.

Hediger, S., L. Michel, and J. Näf (2019). On the Use of Random Forest for Two-Sample Testing. *arXiv:1903.06287*.

Hernan Madrid Padilla, O., Y. Yu, D. Wang, and A. Rinaldo (2019). Optimal nonparametric change point detection and localization. *arXiv:1905.10019*.

Hotz, T., O. M. Schütte, H. Sieling, T. Polupanow, U. Diederichsen, C. Steinem, and A. Munk (2013). Idealizing ion channel recordings by a jump segmentation multiresolution filter. *IEEE Transactions on NanoBioscience 12*(4), 376–386.

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of classification 2*(1), 193–218.

James, N. and D. Matteson (2015). ecp: An r package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software 62*(7), 1–25.

Kaul, A., V. K. Jandhyala, and S. B. Fotopoulos (2019). An efficient two step algorithm for high dimensional change point regression models without grid search. *Journal of Machine Learning Research 20*(111), 1–40.

Kim, C.-J., J. C. Morley, and C. R. Nelson (2005). The structural break in the equity premium. *Journal of Business & Economic Statistics 23*(2), 181–191.

Kovács, S., L. Housen, and P. Bühlmann (2019). Seeded binary segmentation. *working paper*.

Liu, S., M. Yamada, N. Collier, and M. Sugiyama (2013). Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks 43*, 72–83.

Londschien, M., S. Kovács, and P. Bühlmann (2019). Change point detection for graphical models in presence of missing values. *arXiv:1907.05409*.

Lung-Yut-Fong, A., C. Lévy-Leduc, and O. Cappé (2011). Homogeneity and change-point detection tests for multivariate data using rank statistics. *arXiv:1107.1971*.

Malley, J. D., J. Kruppa, A. Dasgupta, K. G. Malley, and A. Ziegler (2012). Probability machines. *Methods of information in medicine 51*(01), 74–81.

Matteson, D. S. and N. A. James (2014). A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association 109*(505), 334–345.

Olshen, A. B., E. S. Venkatraman, R. Lucito, and M. Wigler (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics 5*(4), 557–572.

Pein, F., H. Sieling, and A. Munk (2017). Heterogeneous change point inference. *Journal of the Royal Statistical Society, Series B 79*, 1207–1227.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*(336), 846–850.

Reeves, J., J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu (2007). A Review and Comparison of Changepoint Detection Techniques for Climate Data. *Journal of Applied Meteorology and Climatology 46*, 900–915.

Truong, C., L. Oudre, and N. Vayatis (2019). Selective review of offline change point detection methods. *Signal Processing*, ...

Vostrikova, L. Y. (1981). Detecting 'disorder' in multidimensional random processes. *Soviet Mathematics Doklady 24*, 270–274.

Wright, M. N. and A. Ziegler (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software 77*(1), 1–17.

Zhang, N. R. and D. O. Siegmund (2007). A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics 63*(1), 22–32.

Zou, C., G. Yin, L. Feng, and Z. Wang (2014). Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics 42*(3), 970–1002.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

---

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Random Forests and Other Non-parametric Classifiers for Multivariate Change Point Detection

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

| **Name(s):** | **First name(s):** |
|---|---|
| Londschien | Malte |
| | |
| | |
| | |
| | |

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

| **Place, date** | **Signature(s)** |
|---|---|
| Zürich, 31.10.2019 |  |
| | |
| | |
| | |
| | |

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*