

# Segmentation of binary sequence via minimizing least square error with total variation regularization

Jeungju Kim<sup>a</sup>, Johan Lim<sup>1,a</sup>

<sup>a</sup>Department of Statistics, Seoul National University, Korea

---

## Abstract

In this paper, we propose a data-driven procedure to segment a binary sequence as an alternative to the popular hidden Markov model (HMM) based procedure. Unlike the HMM, our procedure does not make any distributional or model assumption to the data. To segment the sequence, we suggest to minimize the least square distance from the observations under total variation regularization to the solution, and develop a polynomial time algorithm for it. Finally, we illustrate the algorithm using a toy example and apply it to the Gemini boat race data between Oxford and Cambridge University. Further, we numerically compare the performance of our procedure to the HMM based segmentation through these examples.

**Keywords:** binary sequence, gemini boat race data, least square error, run length code, segmentation, total variation regularization

---

## 1. Introduction

The segmentation of a binary sequence has been discussed with several different names depending on its context, where the binary sequence comes from and what are the purpose of the segmentation. In signal processing, it is often called the de-noising of a noisy signal or the compression of binary signal (Gray, 1984; Lelewer and Hirschberg, 1987; Donoho *et al.*, 1998; Selesnick *et al.*, 2013). In statistics and their related areas, it is often read as the change point detection of a sequence of binary observations (Rojas and Wahlberg, 2014; Son *et al.*, 2023).

In all the above, the hidden Markov model (HMM) is the most popular method to segment the binary sequence/achieve their goal (Zucchini *et al.*, 2017). It assumes the existence of a true sequence that follows a Markov Chain (MC) and finds the most likely sequence having the highest posterior probability. To estimate the HMM we have 60 years old veteran algorithms, the Viterbi algorithm (Viterbi, 1967; Forney, 1973) and the Baum-Welch algorithm (Baum and Petrie, 1966), which are still beloved up to now. However, HMM makes a specific distributional assumption: the Markov Chain assumption on the true signal includes a specific order and the conditionally independent observational distribution given the true sequence.

In this paper, as an alternative to HMM, we propose a procedure for the segmentation of a binary sequence that is purely data-driven and does not make any distributional or model assumptions. We suggest finding a sequence that minimizes the least square distance to the observations, which we call the least square error in this paper, under the assumption that the number of segments is given a priori. Here, in a binary sequence, the least square distance is equal to the Hamming distance without the

---

<sup>1</sup>Corresponding author: Department of Statistics, Seoul National University, Kwanak Ro 1, Seoul 08826, Korea. E-mail: [johanlim@snu.ac.kr](mailto:johanlim@snu.ac.kr)

constant, and the number of segments of a binary sequence is equal to its total variation. We develop a polynomial time algorithm to solve the problem (a formal definition of the problem is given in (2.1) in the next section). In addition, we adapt the GAP statistic (Tibshirani *et al.*, 2001) as an outer procedure to decide the number of segments, which compares the total within-cluster sum of squares of clusters with its expected value under the assumption of no segment.

This article is organized as follows. In Section 2, we propose an algorithm to solve the problem, the minimization of the least square error under total variation regularization. We illustrate it with a toy example to help the understanding of the algorithm and show its computational complexity. In addition, we review and discuss the GAP statistic and its variants to decide the number of optimal segments. In Section 3, we apply it to segmenting two real data examples, the Gemini boat race data between Oxford and Cambridge University <https://www.theboatrace.org> and the Tiger Woods tournament data between September, 1999 and June, 2001.

## 2. Data driven binary segmentation

Suppose we observe a binary sequence of binary observations  $\mathbf{x}_n = (X_1, \dots, X_n)$  and want to find its segmented or de-noised sequence  $\mathbf{y}_n = (Y_1, \dots, Y_n)$ .

### 2.1. Main problem

The problem we suggest to solve is, for a given constant  $R$ ,

$$\begin{aligned} & \text{minimize}_{\mathbf{y}_n} \quad \mathcal{E}(\mathbf{y}_n, \mathbf{x}_n) := \sum_{i=1}^n (Y_i - X_i)^2 = \sum_{i=1}^n I(Y_i \neq X_i) \\ & \text{subject to} \quad \sum_{i=2}^n I(Y_{i-1} \neq Y_i) \leq R, \end{aligned} \quad (2.1)$$

where  $I(A)$  is the indicator function for the event  $A$ . In the above,  $\sum_{i=2}^n I(Y_{i-1} \neq Y_i)$  equals to the total variation of the sequence  $\mathbf{y}_n$  that is  $V(\mathbf{y}_n) := \sum_{i=2}^n |Y_i - Y_{i-1}|$ .

### 2.2. Algorithm to solve the main problem

We start with some notations and preliminary results related with the optimal solution  $\mathbf{y}_n^* = (Y_1^*, Y_2^*, \dots, Y_n^*)$  to (2.1).

**Lemma 1.** *Let  $\mathcal{A}$  be the set of indices  $i = 2, 3, \dots, n$  that satisfies  $X_i = X_{i-1}$ . For every  $i \in \mathcal{A}$ ,  $Y_i^* = Y_{i-1}^*$ .*

**Proof:** Suppose, for some  $k \in \mathcal{A}$ ,  $Y_k^* \neq Y_{k-1}^*$ . We assume  $X_{k-1} = X_k = 0$  and  $Y_{k-1} = 0, Y_k = 1$  without loss of generality. Let  $\mathbf{z}_n = (Z_1, Z_2, \dots, Z_n)$

$$Z_i = Y_i, \quad i \neq k, \quad \text{and} \quad Z_i = 1 - Y_i, \quad i = k,$$

i.e.,  $Z_i$  differs from  $Y_i$  only when  $i = k$ . We then have

$$\mathcal{E}(\mathbf{z}_n, \mathbf{x}_n) < \mathcal{E}(\mathbf{y}_n^*, \mathbf{x}_n),$$

while

$$V(\mathbf{z}_n) \leq V(\mathbf{y}_n^*),$$

It contradicts to  $\mathbf{y}_n^*$  is the optimal solution to (2.1). □

We use the following notations and terminologies in explaining our algorithm.

- (T1)  $\forall i = 2, 3, \dots, n$ , if  $X_i \neq X_{i-1}$ , we call  $i$  as a change point of  $\mathbf{x}_n$ .
- (T2) Suppose  $\mathbf{x}_n = (X_1, \dots, X_n)$  has  $p$  change points  $(i_1, \dots, i_p)$ , where  $i_1 < \dots < i_p$ .  $\forall k = 1, \dots, p+1$ , we define the  $k^{\text{th}}$  chunk of  $\mathbf{x}_n$  as the set

$$C_{\mathbf{x}}(k) := \{X_j \mid i_{k-1} \leq j < i_k\},$$

where  $i_0 = 0$  and  $i_{p+1} = n$ . The length of  $k^{\text{th}}$  chunk is notated as  $|C_{\mathbf{x}}(k)|$ .

- (T3) We define the act of flipping or flip as changing all the elements of a given chunk from 0 to 1 or 1 to 0.
- (T4) We define the cost vector  $d = (d_1, \dots, d_p)$  where  $d_k$  is the additional increase in error (the least square error) when the  $k^{\text{th}}$ -chunk is flipped.
- (T5) In the iteration of the algorithm, we call the  $\mathcal{E}(\mathbf{y}_{n,\ell}, \mathbf{x}_n)$  as the  $\ell^{\text{th}}$  loss where  $\mathbf{y}_{n,\ell}$  is the resulting vector after  $\ell$  flips.

We make three remarks as preliminaries for the algorithm. First, a sequence  $\mathbf{x}_n = (X_1, X_2, \dots, X_n)$  uniquely defines the initial sequence  $X_1$  and the set of change points and vice versa. This introduces the run-length code (RLC) for data compression in signal processing (Golomb, 1966). Second, from Lemma 1, to solve (2.1), it suffices to choose  $R - 1$  change points of the change points of  $\mathbf{x}_n$ . If  $\mathbf{x}_n$  has  $R - 1$  change points or less, the solution  $\mathbf{y}_n^*$  equals to  $\mathbf{x}_n$ . Third, there is a one-to-one correspondence between ‘picking change points of  $\mathbf{x}_n$ ’ and ‘flipping the chunks of  $\mathbf{x}_n$ ’. Thus, we could obtain the solution  $\mathbf{y}_n^*$  to (2.1) by flipping some chunks of  $\mathbf{x}_n$ .

Now let us explain our algorithm to solve (2.1).

- (S1) Change the encoding of the binary sequence  $\mathbf{x}_n = (X_1, X_2, \dots, X_n)$  into the run-length code (RLC) form as  $(|C_{\mathbf{x}_n}(1)|, |C_{\mathbf{x}_n}(2)|, \dots, |C_{\mathbf{x}_n}(p+1)|)$  where  $p$  equals to the number of change points of  $\mathbf{x}_n$ . Additionally, set the initial values of the cost vector to match the RLC.
- (S2) Divide the problem into 4 sub-problems according to our first move, named as ‘0 to 0 (I-00)’, ‘0 to 1 (I-01)’, ‘1 to 0 (I-10)’, ‘1 to 1 (I-11)’. Here, ‘0 to 0 (I-00)’ implies our first move is to make both the first and last chunk’s signals 0, ‘0 to 1 (I-01)’ implies our first move is to make the first chunk’s signal is 0 and the last chunk’s signal is 1, ‘1 to 0 (I-10)’ implies our first move is to make the first chunk’s signal is 1 and the last chunk’s signal is 0, and ‘1 to 1 (I-11)’ implies our first move is to make both the first and last chunk’s signals 1.
- (S3) For each of sub-problem (I-00, I-01, I-10, I-11), flip the chunk having the smallest cost except the first and the last chunk fixed at (S2). Update the corresponding RLC, cost, and loss. Repeat this until the required  $R$  is met.
- (S4) Among the solutions to sub-problems (I-00, I-01, I-10, I-11), find the solution with the smallest loss.

To understand the algorithm above better, let us consider the following simple toy example. Suppose we observe the sequence with a length of  $n = 19$  as

$$\mathbf{x} = (0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1).$$

We aim to segment it into three chunks ( $R = 2$ ), but want to minimally distort the original sequence in terms of the least square distance. To do it, we solve (2.1) with  $R = 2$ .

In (S1), the observation is encoded as RLC with  $X_1 = 0$  as

$$U_0 = (2, 4, 5, 4, 3, 1; 0), \quad (2.2)$$

and the corresponding cost vector is represented as

$$d_0 = (2, 4, 5, 4, 3, 1). \quad (2.3)$$

In this step, the cost vector and the RLC code is the same because flipping the chunk of length  $m$  directly increases the objective function by  $m$ .

In (S2), we consider four initial movements. Firstly, let us consider the (I-00) case. Since our original data  $\mathbf{x}_n$  starts from 0 and ends with 1, we should first flip the last chunk into 0. The last element of  $d_0$  equals to 1 so this increases the loss by 1. The resulting RLC code and cost vector is as follows.

$$U_1^{(1)} = (2, 4, 5, 4, 4; 0) \quad (2.4)$$

$$d_1^{(1)} = (2, 4, 5, 4, 2). \quad (2.5)$$

The last element of the RLC code is computed by  $U_{1,5}^{(1)} = U_{0,5}^{(1)} + U_{0,6}^{(1)} = 3 + 1 = 4$  and the last element of the cost is computed by  $d_{1,5}^{(1)} = d_{0,5}^{(1)} - d_{0,6}^{(1)} = 3 - 1 = 2$ . Here and throughout this example,  $U_{a,b}^{(i)}$  is the  $b^{\text{th}}$  component of  $U_a^{(i)}$ , where  $U_a^{(i)}$  represents the resulting RLC code after  $a$  iterations starting from the  $i^{\text{th}}$  initial movement. Similarly,  $d_{a,b}^{(i)}$  is defined for the cost vector.

In (S3), we flip the second chunk because it has the smallest value in  $d_1$  except the first and the last ones. In fact, tie occurs between the second and the fourth element and we break the tie in favor of the lowest index. This results in the increase in the loss by  $d_{1,2} = 4$  and the loss accumulated is 5. After this flip, the RLC code and cost vector become

$$U_2^{(1)} = (11, 4, 4; 0) \quad (2.6)$$

$$d_2^{(1)} = (3, 4, 2). \quad (2.7)$$

In this iteration,  $U_{2,1}^{(1)} = U_{1,1}^{(1)} + U_{1,2}^{(1)} + U_{1,3}^{(1)} = 2 + 4 + 5 = 11$  and  $d_{2,1}^{(1)} = d_{1,1}^{(1)} + d_{1,3}^{(1)} - d_{1,2}^{(1)} = 2 + 5 - 4 = 3$ . Since the total number of chunks is 3, which is equal our target, we stop here.

For subproblem (I-01), (S2) is skipped since our data  $\mathbf{x}$  already starts from 0 and ends with 1. In (S3), the fifth element of  $d_0$  has the smallest value except the first and the last element so we flip the fifth chunk. This results in

$$U_1^{(2)} = (2, 4, 5, 8; 0) \quad (2.8)$$

$$d_1^{(2)} = (2, 4, 5, 2), \quad (2.9)$$

and the loss becomes 3. The chunk with the smallest cost is the second so flipping the second one yields

$$U_2^{(2)} = (11, 8; 0) \quad (2.10)$$

$$d_2^{(2)} = (3, 2), \quad (2.11)$$

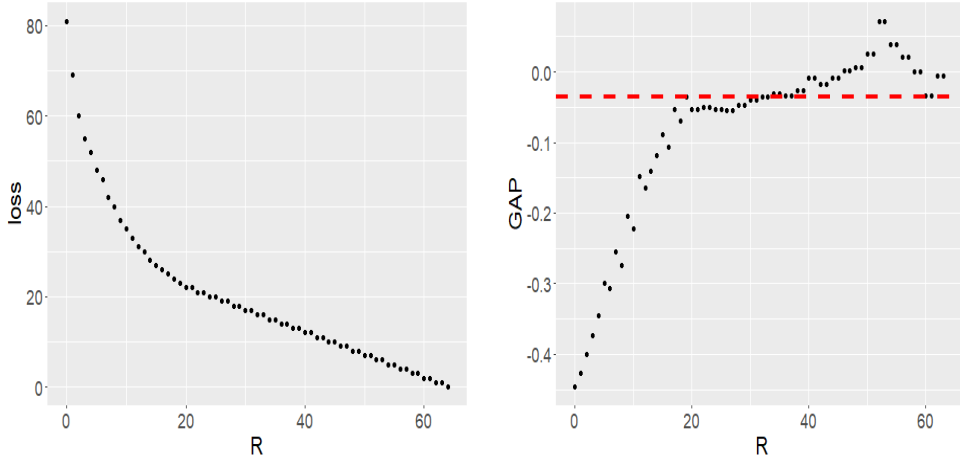


Figure 1: Loss and GAP statistics versus  $R$ : (a) the left panel plots  $R$  vs the optimal error  $\mathcal{E}$  of  $\mathbf{y}_n^*$ , (b) the right panel plots  $R$  vs GAP statistics.

and the loss increases to 7. Now that the total number of chunk is less than our target, we stop here.

We iterate the same procedure for subproblem (I-10) and (I-11), and both end up with

$$\begin{aligned} U_2^{(3)} &= (6, 13; 1), & d_2^{(3)} &= (2, 3), & \text{loss} &= 7, \\ U_2^{(4)} &= (6, 5, 8; 1), & d_2^{(4)} &= (2, 5, 2), & \text{loss} &= 5, \end{aligned}$$

respectively.

Finally, in (S4), we choose both (I-00) and (I-11) which yielded the smallest loss, 5. Thus our algorithm finds two optimal solutions  $U_2^{(1)}$  and  $U_2^{(4)}$ .

### 2.3. Two remarks

We make two remarks on our proposed algorithm. First, the solution found by the algorithm is at least sub-optimal by its nature. The algorithm searches for the best flip of the current segmentation among all possibilities to make the smallest increase in the cost. Thus, by its nature, the solution has the smallest cost among its neighbor segmentations made by a flip of the current segmentation. Second, the proposed algorithm is a polynomial time algorithm. It is because, in (S3), each step decreases the quantity  $\sum_{i=2}^n I(Y_{i-1} \neq Y_i)$  by 2. Since the number of chunks is less than  $n$ , this shows that the algorithm is  $O(n)$ .

### 2.4. Selection of $R$

The parameter  $R$  decides the number of segments of  $\mathbf{y}_n^*$  as  $R + 1$ , which is an outer parameter that is given before formulating the main problem (2.1). The parameter represents the regularity or smoothness of the optimal signal, and it is important to decide it appropriately. We find many methods to do it in the literature, although there exists no best one over all others (Casella *et al.*, 2014).

In this paper, we read our segmentation problem as a clustering of binary sequences on a line graph and adapt the original GAP statistic by (Tibshirani *et al.*, 2001), which is designed to find the optimal number of clusters in cluster analysis. We recall that there exist several variants of the GAP



Figure 2: The original  $y_n$  and optimal sequence of HMM and HTV  $y_n^*$  when  $R = 14$ .

(e.g. Yan and Ye (2007) and Mohajer *et al.* (2010)). However, we do not consider them here because all measures have pros and cons, and have steps we may subjectively involve in.

A brief introduction to the original GAP is as follows. Suppose the data are segmented into  $K$  exclusive and exhaustive clusters,  $C_1, \dots, C_K$ , where  $C_k = \{a_k, a_k + 1, \dots, b_k\}$  and  $a_{k+1} = b_k + 1$  and  $a_k \leq b_k$  for  $k = 1, 2, \dots, n - 1$ . In connection with our total variation penalty in (2.1),  $K = R$ . Given the clusters, the sum of the pairwise distances for all points in cluster  $k$  is defined as

$$D_k = \sum_{i, i' \in C_k} d(i, i'),$$

where  $d(i, i')$  is any distance measure between  $Y_i$  and  $Y_{i'}$ . Here, we use the Euclidean distance and  $D_k$  is simply reduced to the product of the number of 1s and the number of 0s in cluster  $k$ . Once  $D_k$  is computed, the pooled within-cluster sum of squares (PWSS) is defined as

$$W_K = \sum_{k=1}^K \frac{1}{2|C_k|} D_k,$$

where  $|C_k|$  is the number of observations in cluster  $k$ . Finally, the GAP statistics is defined as

$$\text{Gap}_n(k) = \mathbb{E}_n^* [\log(W_k)] - \log(W_k), \tag{2.12}$$

where  $\mathbb{E}_n^*[\log(W_k)]$  is the expectation of  $\log(W_k)$  for the observations from one cluster. We approximate the expectation in (2.12) using Monte Carlo samples independently from Bernoulli distribution with probability  $p$ , where  $p$  is set as the empirical proportion of 1 of  $\mathbf{x}_n$ .

To be specific, let  $\mathbf{x}_n^b$  be the Monte Carlo replications of  $\mathbf{x}_n$  under one cluster assumption, and  $W_k^{(b)}$  their PWSS values for  $b = 1, 2, \dots, B$ . Then,

$$\hat{\text{Gap}}(k) = (1/B) \sum_{b=1}^B \log(W_k^{(b)}) - \log(W_k),$$

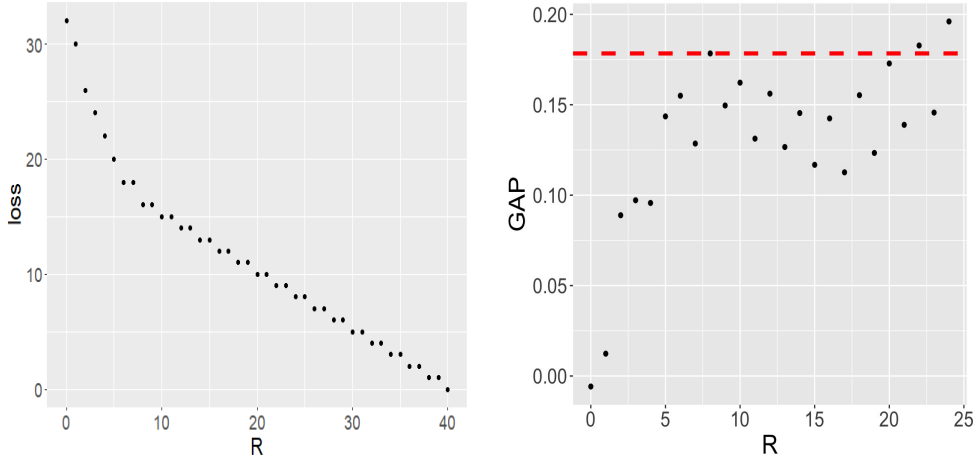


Figure 3: Loss and GAP statistics versus  $R$ : (a) the left panel plots  $R$  vs the optimal error  $\mathcal{E}$  of  $\mathbf{y}_n^*$ , (b) the right panel plots  $R$  vs GAP statistics.

and find the optimal  $k$  as

$$k^* = \operatorname{argmin} \left\{ k \mid \hat{\text{GAP}}(k) > \hat{\text{GAP}}(k+1) - \epsilon s_k \right\}, \quad (2.13)$$

where

$$\begin{aligned} \bar{W} &= \frac{1}{B} \sum_{b=1}^B \log(W_k^{(b)}) \\ \operatorname{sd}^2(\hat{\text{GAP}}(k)) &= \frac{1}{B} \sum_{b=1}^B (\log(W_k^{(b)}) - \bar{W})^2 \\ s_k^2 &= \operatorname{sd}^2(\hat{\text{GAP}}(k)) \left(1 + \frac{1}{B}\right), \end{aligned}$$

and  $\epsilon$  can be empirically chosen. The one-standard error rule in the literature (Tibshirani *et al.*, 2001) sets  $\epsilon = 1$  in (2.13).

### 3. Data example

#### 3.1. The gemini boat race data

In this section, we apply our algorithm (named as HTV) to the Gemini boat race data from <https://theboatrace.org/results>. It records the boat race results between Oxford and Cambridge University from year 1829 to now. The data are in the form of  $167 \times 5$  table each column denoting race, year, winner, winning distance, and winning time. In our analysis, we only used the winner column, where we arbitrarily assigned 1 if Oxford won, 0 if Cambridge won.

We tried  $R$  from 0 to 64 and found the optimal solution to (2.1),  $\mathbf{y}_n^*(R)$ , for each  $R$ . The minimum error achieved for each  $R$  is plotted in the left panel of Figure 1. In addition, we computed the GAP statistic for every  $R$  with the method suggested in Section 2.4. The result is shown in Figure 1. Applying the process in Section 2.4 resulted in  $\hat{R} = 0$  meaning that there's no clear evidence of clusters

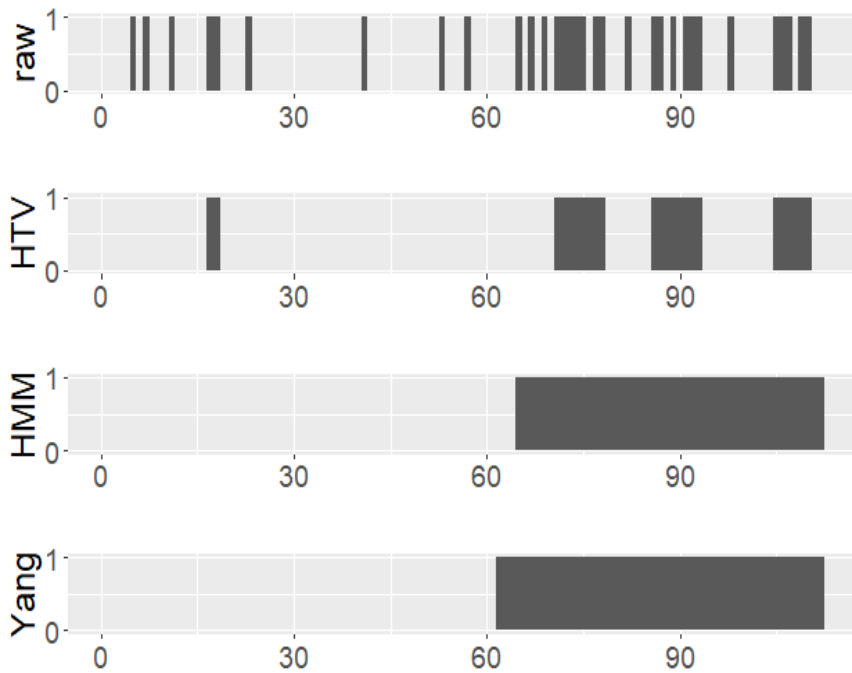


Figure 4: Segmentation of Tiger Woods' tournament results.

in the data. However, we conclude it is quite reasonable to select  $R = 19$ , which is a local maximum as shown in the right panel of Figure 1. The total variation rate for  $R = 19$  for was  $24/167 \approx 14.4\%$ , and the detected change points are as, with a total of 19 in all, 1845, 1866, 1873, 1892, 1898, 1903, 1905, 1907, 1926, 1928, 1952, 1957, 1967, 1976, 1982, 1988, 1996, 2013, 2022. The results are plotted in Figure 2. In the figure, for the comparison, we applied the HMM with the first-order hidden Markov chain for the hidden states for the comparison. The HMM-based approach identified 14 distinct changepoints corresponding to  $R = 13$  in our method. We find that the major segments by three results, our HTVs with  $R = 14$  and  $R = 19$  and HMM, are quite similar to each other.

### 3.2. The Tiger Woods tournament data

As our second data example, we analyzed the tournament data of Tiger Woods, which includes the results of 112 tournaments from September 1999 to June 2001. The data are reported in Yang (2004) and the author proposed a Bayesian binary segmentation procedure, inherently similar to HMM, to understand the phenomenon of 'streakiness'.

Figure 4 displayed the results by three methods considered here, HTV, HMM, and the Bayesian method by Yang (2004). Our HTV identified eight distinct change points following the procedure outlined in Section 2.4, where the GAP statistics and losses are plotted in Figure 3. On the other hand, the HMM and the Bayesian method by Yang (2004) pinpointed only a single changepoint, which provides much coarser segments. The clustered data elucidate the career trajectory of Tiger Woods during the period. We claim our HTV shows a more detailed move of his career than the other two methods. The detected 8 change points are the 17th, 19th, 71th, 79th, 86th, 94th, 105th, and 111th and



Table 1: The detected change points and their relevant events

Changepoint	Tournament name	Relevant event
17	Masters tournament, 1997	First major championship win in Woods' career
19	MasterCard Colonial, 1997	Woods started to undertake his first swing change under the guidance of Butch Harmon
71	WGC-NEC Invitational, 1999	Woods underwent laser eye surgery to correct his myopic.
86	Memorial, 2000	Woods perfectly adapted to his changed swing.
94	National Car Rental Golf Classic Disney, 2000	Woods signed a blockbuster deal with NIKE.
111	US Open, 2001	Woods claimed a historic achievement that has since been known as the 'Tiger Slam' (winning all four men's major championships).

their possibly relevant events are reported in Table 1.

#### 4. Conclusion

In this paper, we propose to minimize the least square errors to segment a binary sequence as an alternative to the HMM-based method. We develop a polynomial time algorithm to find the solution for a given number of segments,  $R$ , and discuss the procedure to select  $R$ , the regularization parameter. We illustrate our procedure by segmenting two real data examples, the boat race data between Oxford and Cambridge University and the tournament results of Tiger Woods.

The results of this paper address the segmentation problem for the simplest form of data, but we conjecture they could be extended to more complex data. For example, our HTV can be directly applied to the segmentation of binomial sequences, a notable instance being baseball hit data. To be specific, the HTV for binomial sequences is: First, we initialize chunks based on their predominant binary outcomes and we randomly break the tie. Specifically, a chunk is assigned a value of 1 if the frequency of 1s exceeds that of 0s, and 0 otherwise. This step results in an RLC form. Second, calculate the cost vector of chunks but with extra care as our initial chunks could contain both 0s and 1s. Now that we obtained the RLC code and the cost vector, we can proceed as specified in Section 2. Some other examples we are of interest are the sequence of data with a higher base more than two, or the sequence of two- or multi-dimensional binary data. In both extensions, the form of total variation regularization becomes more complex, posing a challenge in how to effectively control it during the iterative procedures. We leave these for future work.

#### Acknowledgement

The authors are grateful to the associate editor and two reviewers for several variable comments. The R code of this paper is available from <https://github.com/z0o0/bseg>. This paper is supported by the National Research Foundation of Korea (No. NRF-2021R1A2C1010786).

#### References

- Baum LE and Petrie T (1966). Statistical inference for probabilistic functions of finite state Markov chains, *The Annals of Mathematical Statistics*, **37**, 1554–1563.
- Casella G, Moreno E, and Girón FJ (2014). Cluster analysis, model selection, and prior distributions on models, *Bayesian Analysis*, **9**, 613–658.
- Donoho DL, Vetterli M, DeVore RA, and Daubechies I (1998). Data compression and harmonic anal-

- ysis, *IEEE Transactions on Information Theory*, **44**, 2435–2476.
- Forney GD (1973). The Viterbi algorithm, *Proceedings of the IEEE*, **61**, 268–278.
- Gray RM (1984). Vector quantization, *IEEE ASSP Magazine*, **1**, 4–29.
- Golomb S (1966). Run-length encodings (corresp.), *IEEE Transactions on Information Theory*, **12**, 399–401.
- Kehagias A (2004). A hidden Markov model segmentation procedure for hydrological and environmental time series, *Stochastic Environmental Research and Risk Assessment*, **18**, 117–130.
- Lelewer DA and Hirschberg DS (1987). Data compression, *ACM Computing Surveys*, **19**, 261–296.
- Mohajer M, Englmeier K-H, and Schmid VJ (2010). A comparison of gap statistic definitions with and without logarithm function (Technical Report Number 096, 2010), Department of Statistics, University of Munich, München.
- Rojas CR and Wahlberg B (2014). On change point detection using the fused lasso method, Available from: arXiv preprint, arXiv:1401.5408
- Selesnick IW and Chen PY (2013). Total variation denoising with overlapping group sparsity. In *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 5696–5700.
- Son W, Lim J, and Yu D (2023). Path algorithms for fused lasso signal approximator with application to COVID-19 spread in Korea, *International Statistical Review*, **91**, 218–242.
- Tibshirani R, Walther G, and Hastie T (2001). Estimating the number of clusters in a data set via the gap statistic, *Journal of the Royal Statistical Society, Series-B*, **63**, 411–423.
- Viterbi AJ (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*, **13**, 260–269.
- Yan M and Ye K (2007). Determining the number of clusters using the weighted gap statistic, *Biometrics*, **63**, 1031–1037.
- Yang T(2004). Bayesian binary segmentation procedure for detecting streakiness in sports, *Journal of the Royal Statistical Society, Series-A*, **167**, 627–637.
- Zucchini W, MacDonald IL, and Langrock R (2017). *Hidden Markov Models for Time Series: An Introduction Using R.*, CRC Press, Boca Raton, Florida.

Received December 24, 2023; Revised May 02, 2024; Accepted May 20, 2024