(k, h)-segmentation

Aristides Gionis and Heikki Mannila

Basic Research Unit

Helsinki Institute for Information Technology

Department of Computer Science, University of Helsinki, Finland

Recurrent sources

Movivation

(k, h)-segmentation

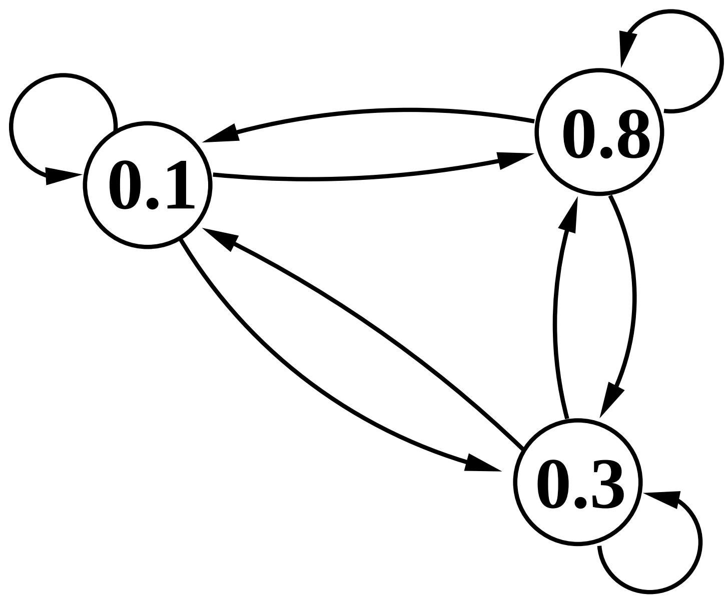
HMMs

Information theoretic

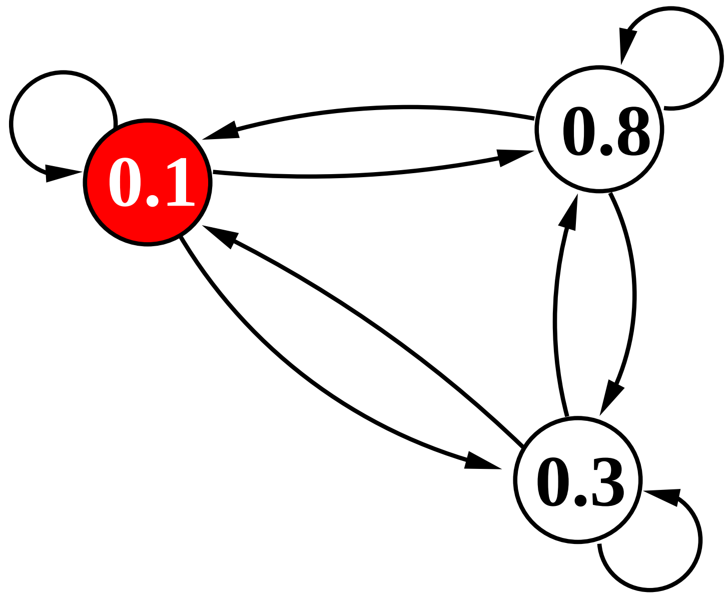
Statistical testing

MCMC

Motivation for finding recurrent sources

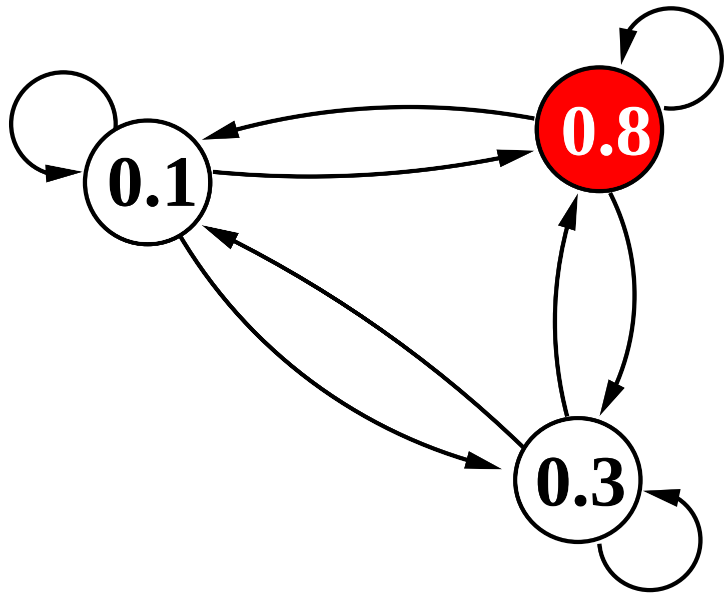


Motivation for finding recurrent sources



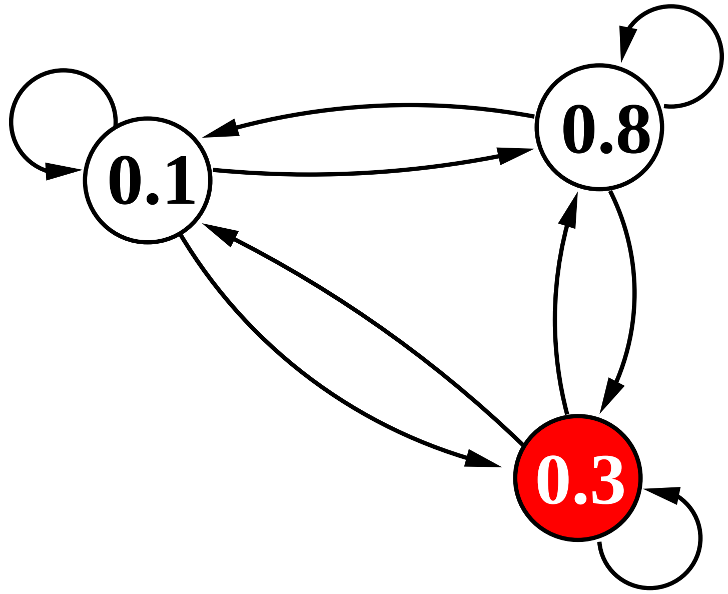
0000100000

Motivation for finding recurrent sources



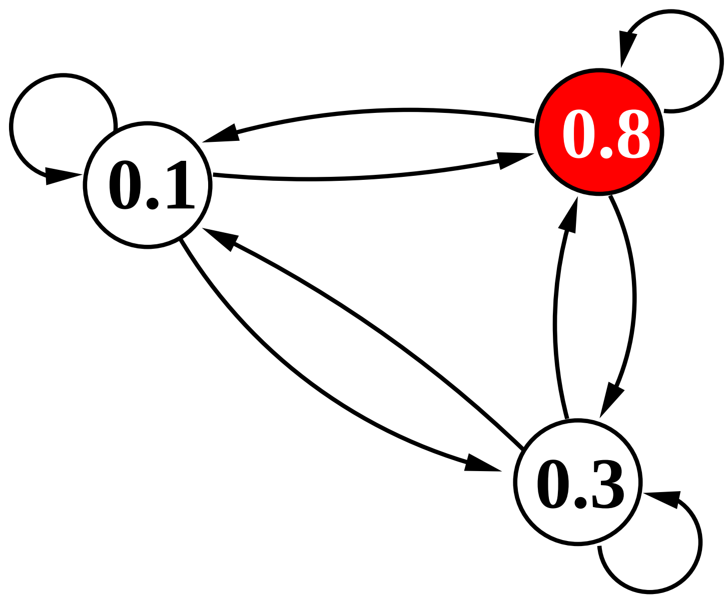
000010000011101111

Motivation for finding recurrent sources



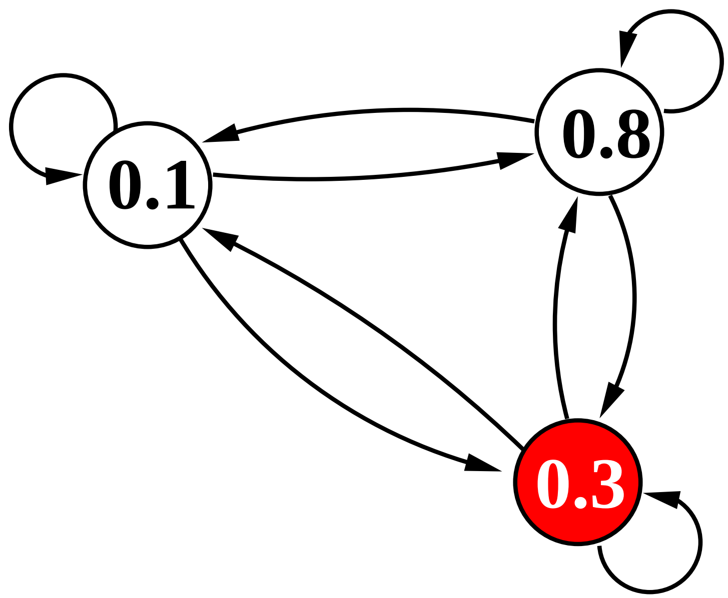
0000100000111011111001000100

Motivation for finding recurrent sources



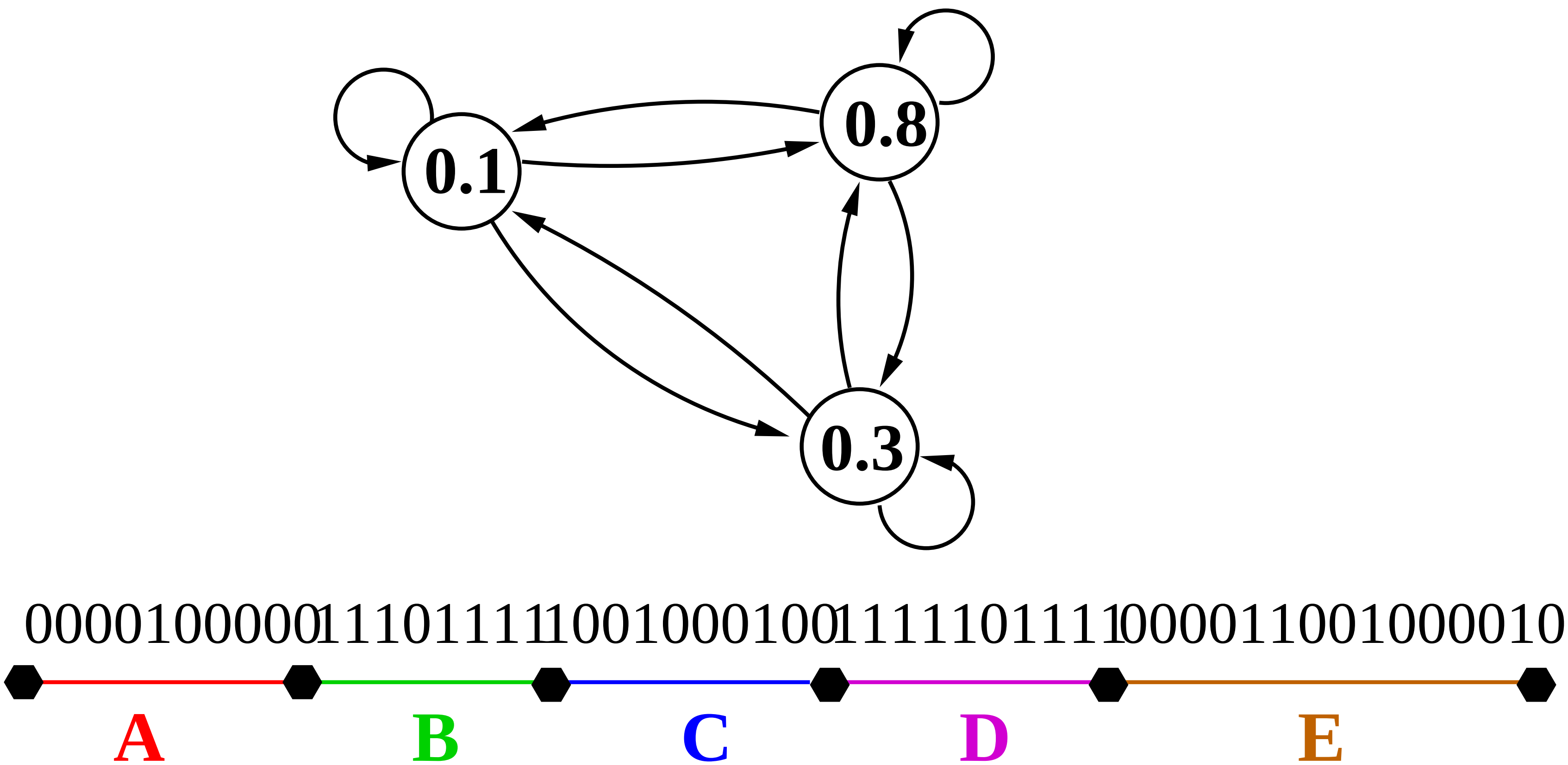
000010000011101111100100010011111101111

Motivation for finding recurrent sources

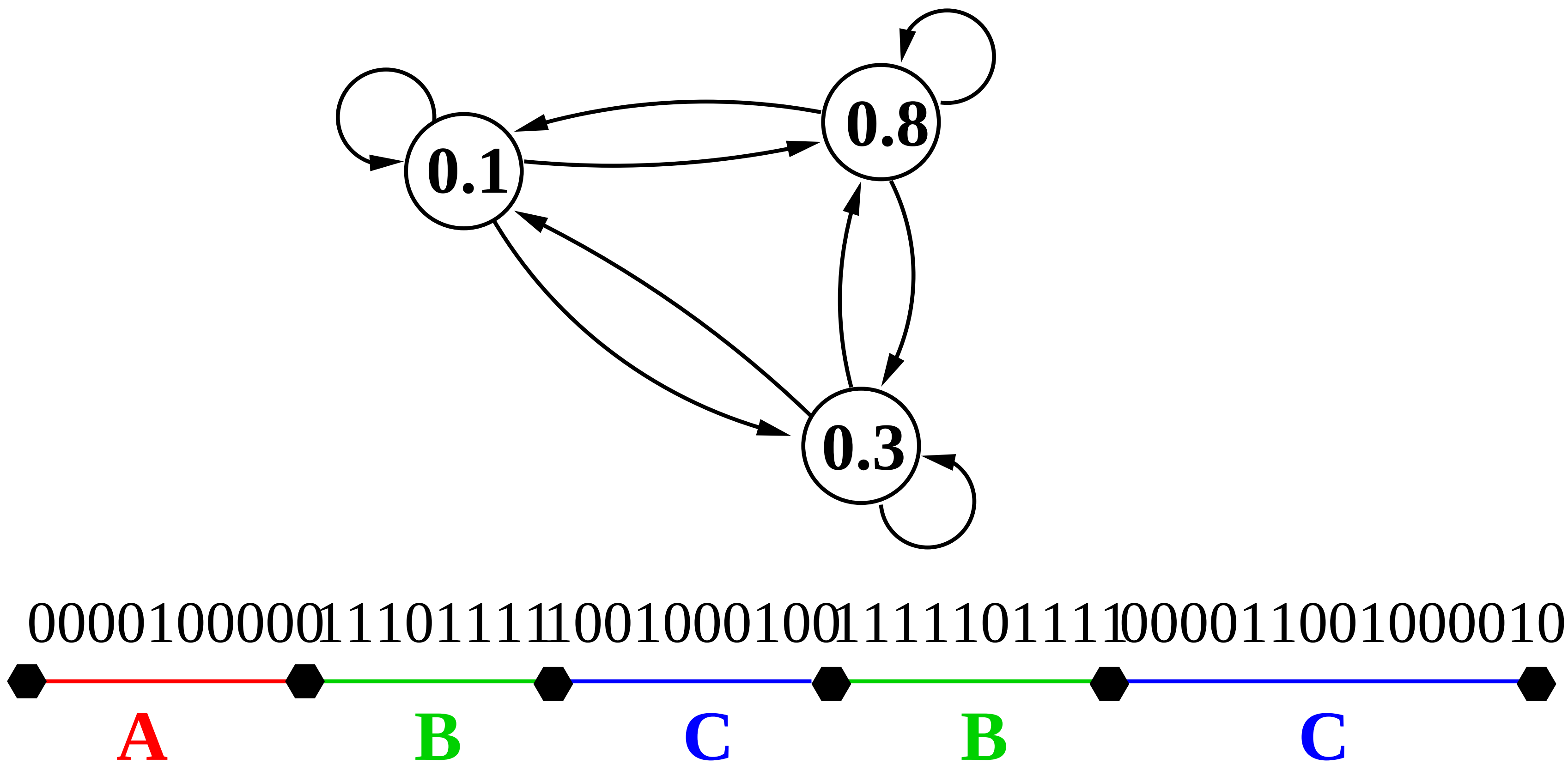


00001000001110111100100010011111011110000110010000010

Motivation for finding recurrent sources



Motivation for finding recurrent sources



(k,h)-segmentation

- Given sequence $S = a_1, a_2, \ldots, a_n$
- We want to find k segments
- But only $h < k$ different *segment types* are allowed
- Each of the k segments should be assigned to one of the h types
- Find the best segmentation into k segments, the h types, and the assignment of each segment to one type

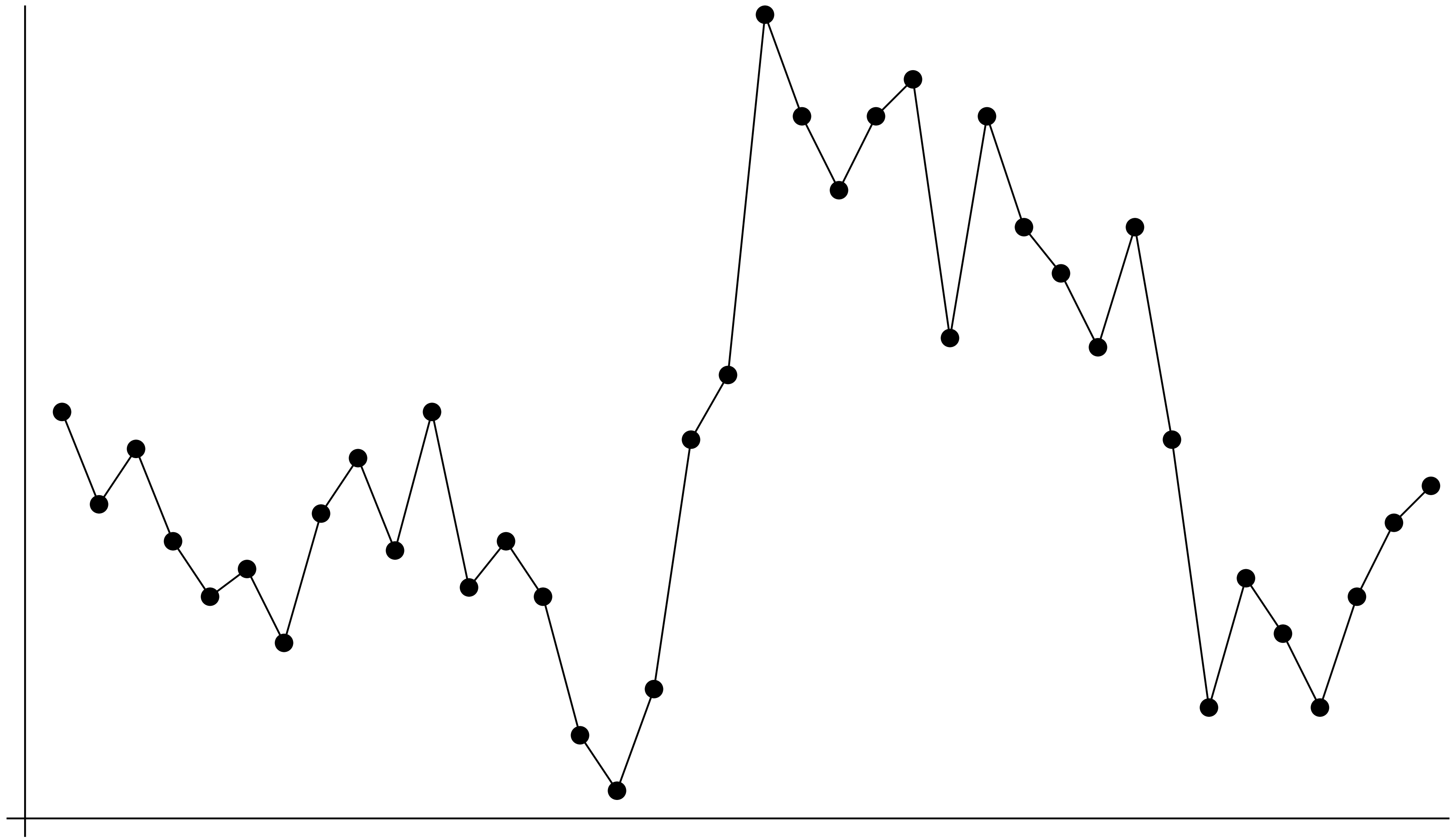
(k,h)-segmentation: problem definition

- Assume piecewise constant representation, and L_2^2
- Given sequence $S = a_1, a_2, \ldots, a_n$
- We want to find
  - partition of S into k segments $S_1, \ldots, S_k$,
  - h levels $l_1, \ldots, l_h$
  - assignment of segment j to level $l_j \in \{l_1, \ldots, l_h\}$

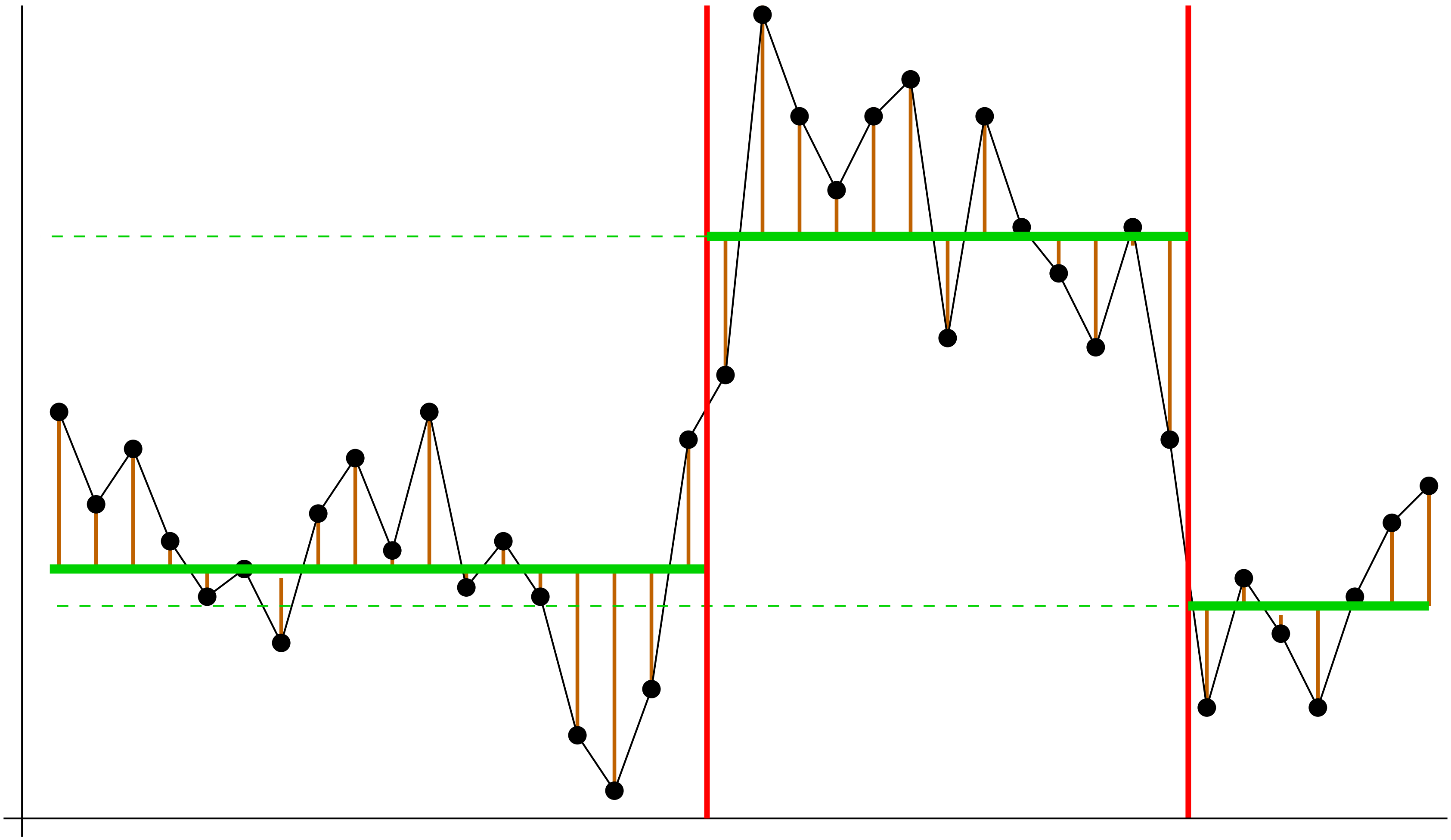
  in order to minimize the total error

$$R[n, k, h] = \sum_{j=1}^{k} \sum_{i=b_j}^{e_j} (a_i - l_j)^2$$

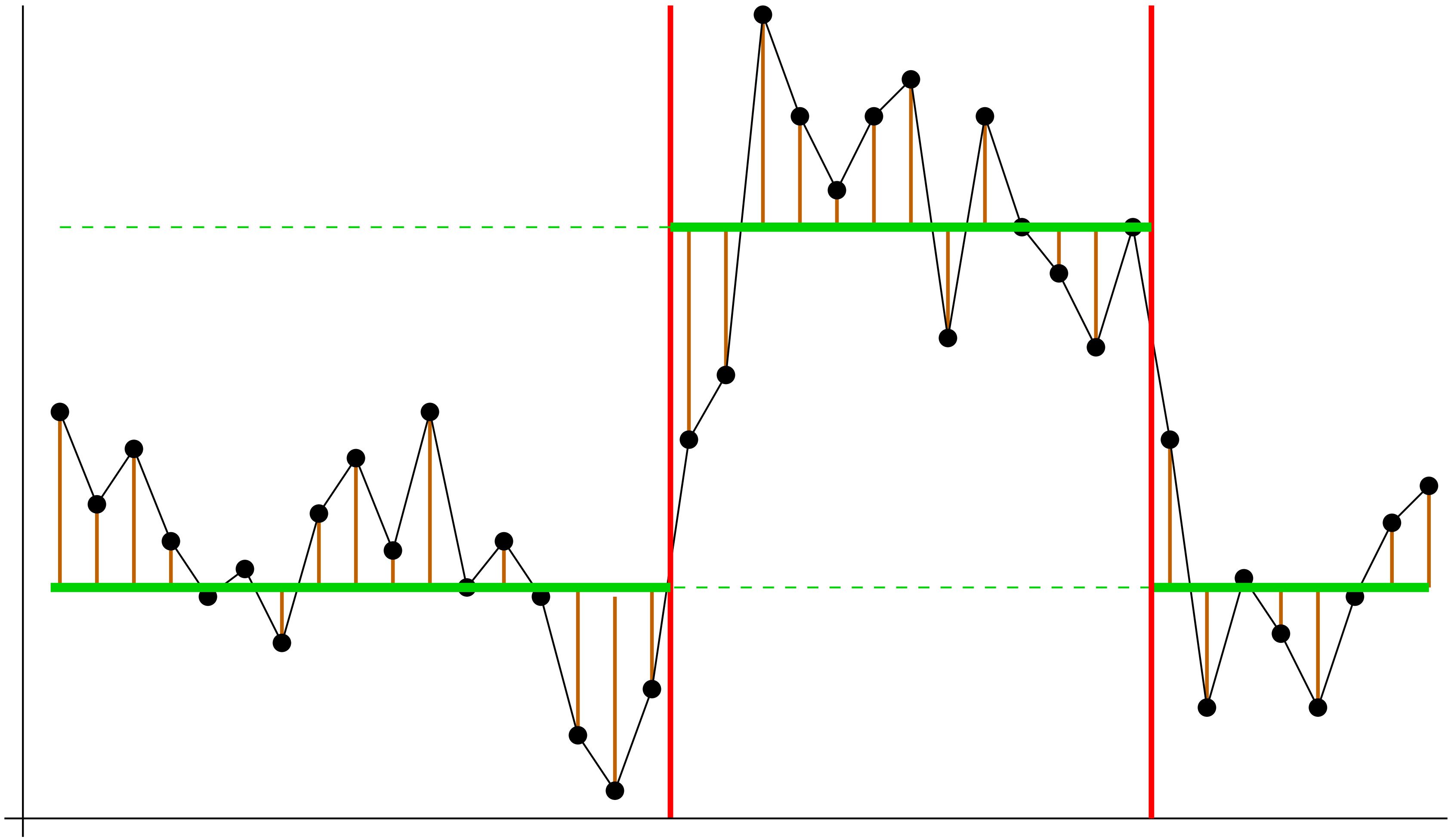
Example



Example: $k = 3$ and $h = 3$



Example: $k = 3$ and $h = 2$



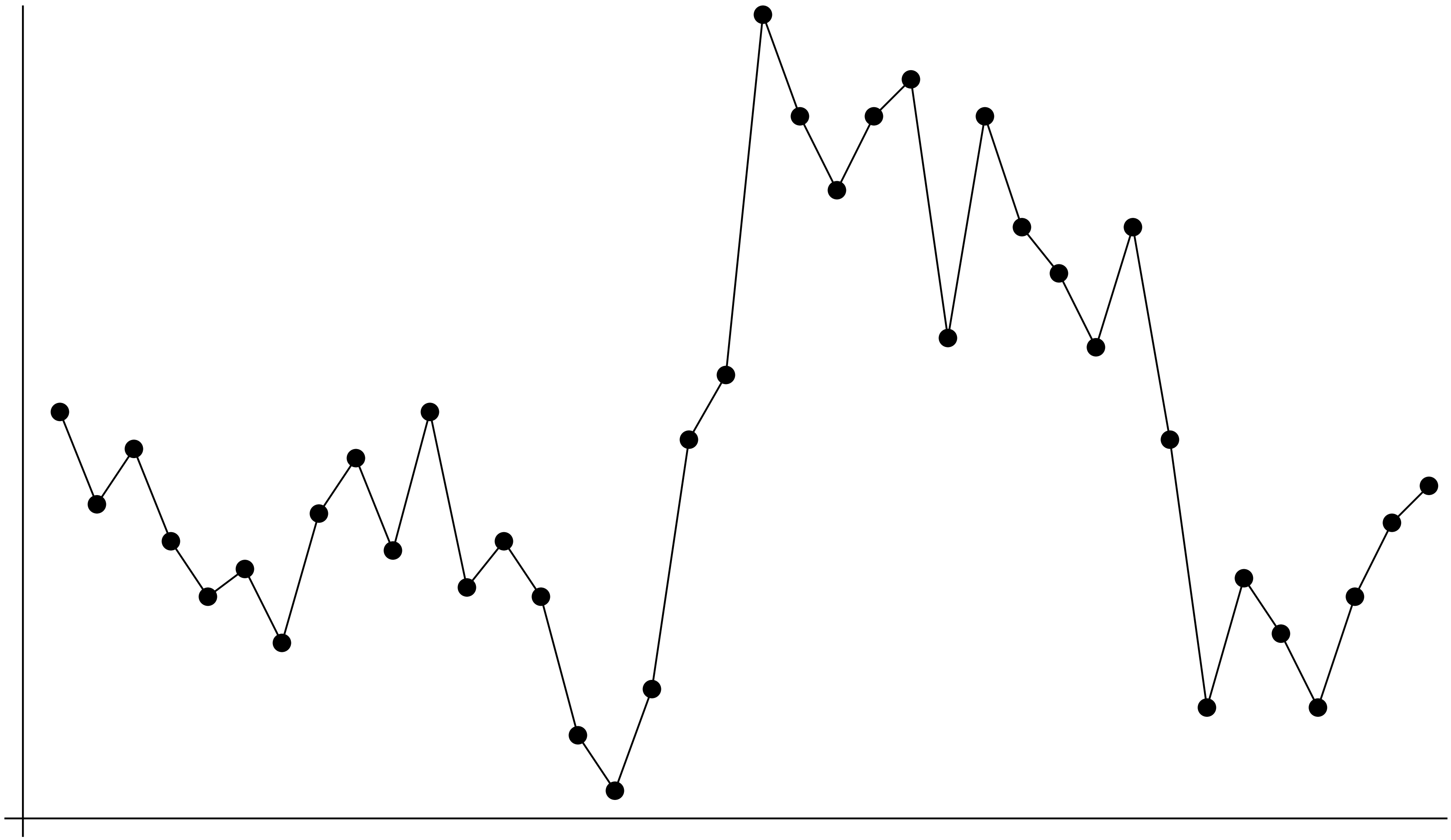
Some facts about the (k, h)-segmentation problem

- NP-Complete problem for multidimensional data ($d > 1$), w.r.t. L_1 and L_2 (contrast with k-segmentation, which is polynomial)
- Generalizes k-segmentation and clustering
  - k-segmentation: $h = k$
  - clustering: $k = n$
- Simple approximation algorithms that combine the above two subproblems
  - $d = 1$: 3-approximation for L_1, 5-approximation for L_2^2
  - $d > 1$: $(3 + \epsilon)$-approximation for L_1, $(1 + 4\alpha^2)$-approximation for L_2^2, where α is the best approximation factor for k-means clustering problem

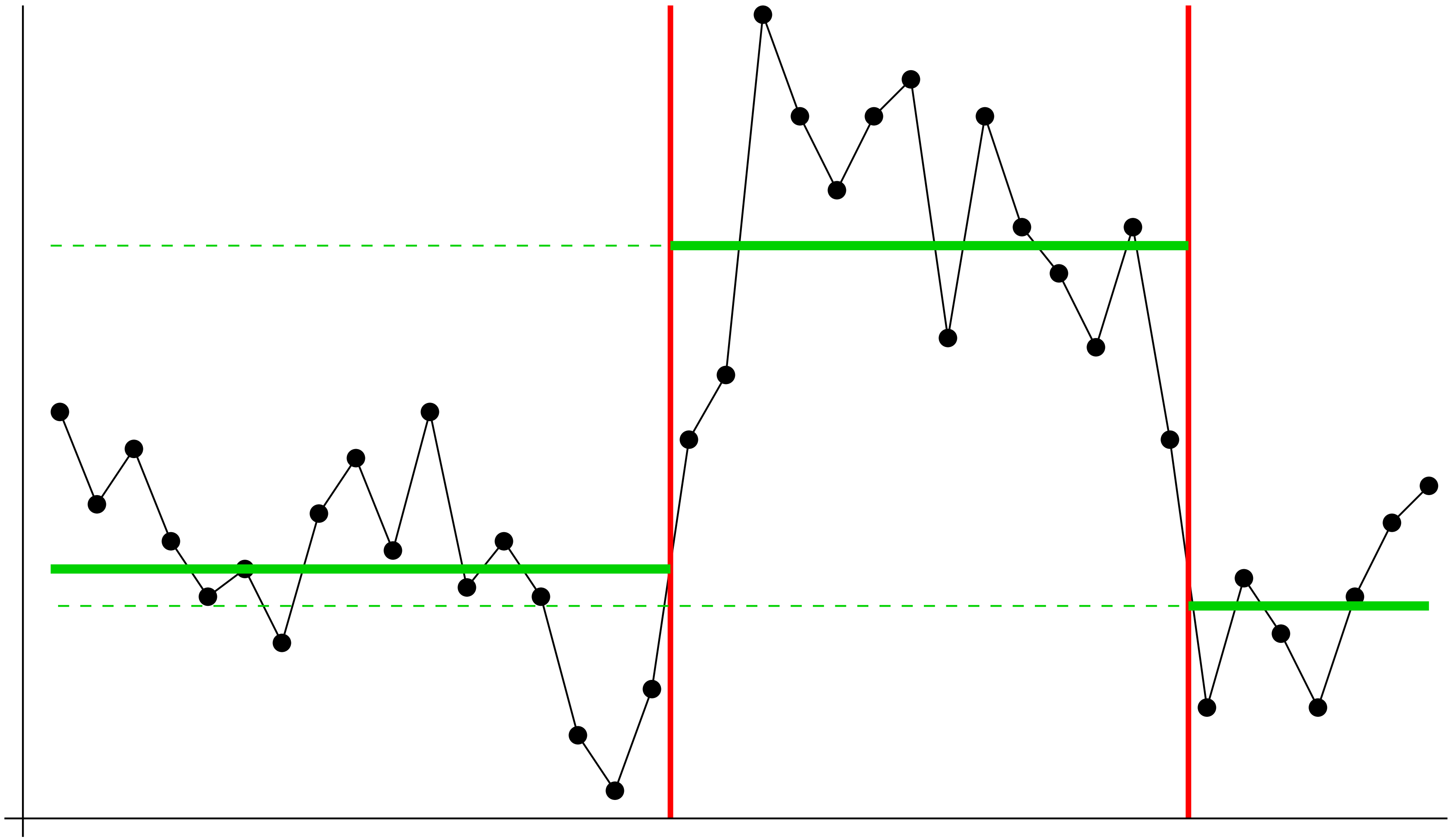
ClusterSegments algorithm

- Solve k-segmentation problem and obtain segments $S_1, \ldots, S_k$
- Consider the representative c_j for each segment S_j (mean, median, etc.)
- Map segment S_j to a weighted point with value c_j and weight $w_j = |S_j|$
- Cluster those k weighted points to h centers $L = \{l_1, \ldots, l_h\}$
- Assign each segment to its closer center in L
- Running time is $O(n^2 k)$ (from dynamic programming)

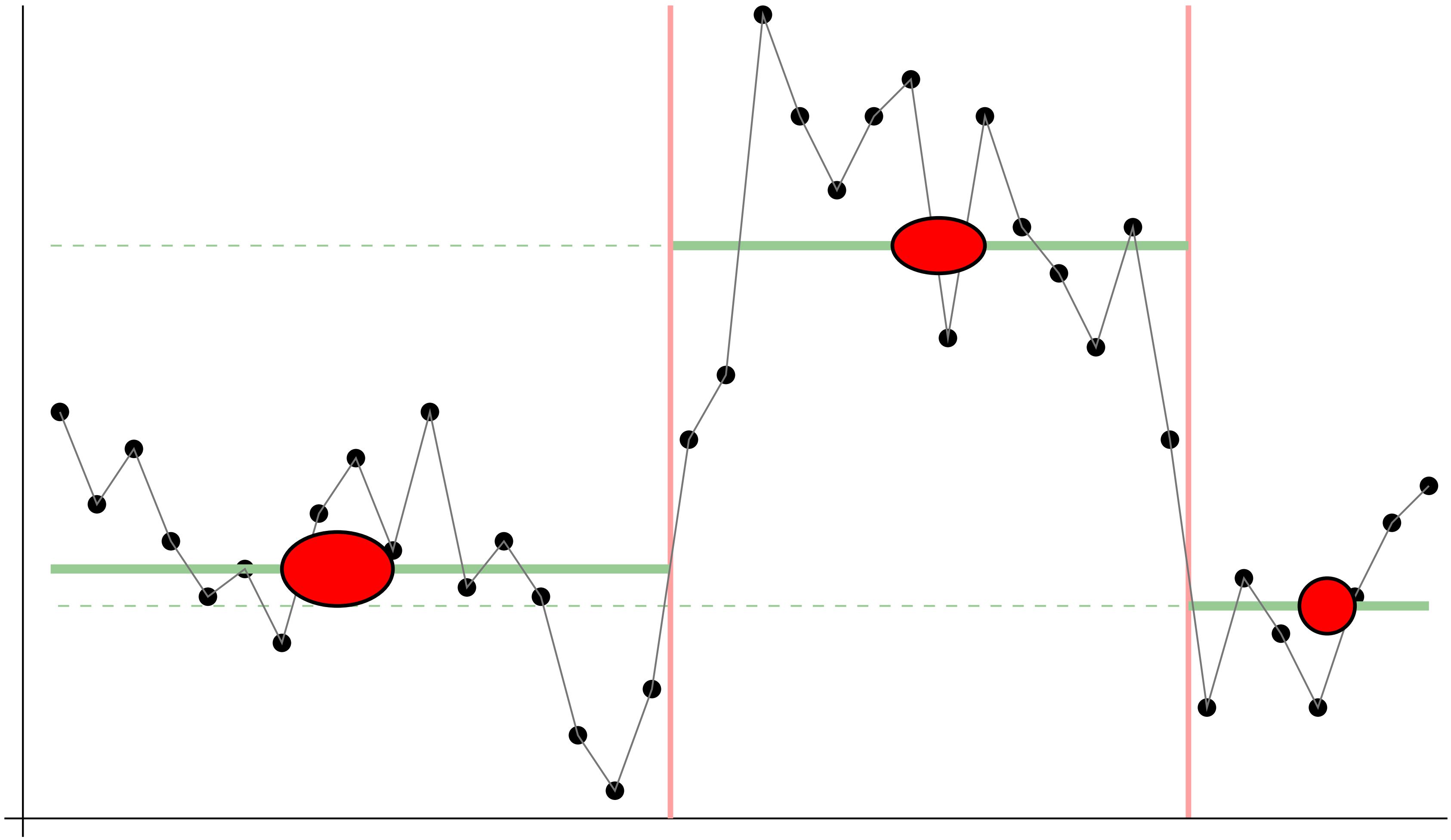
ClusterSegments example, $k = 3$, $h = 2$



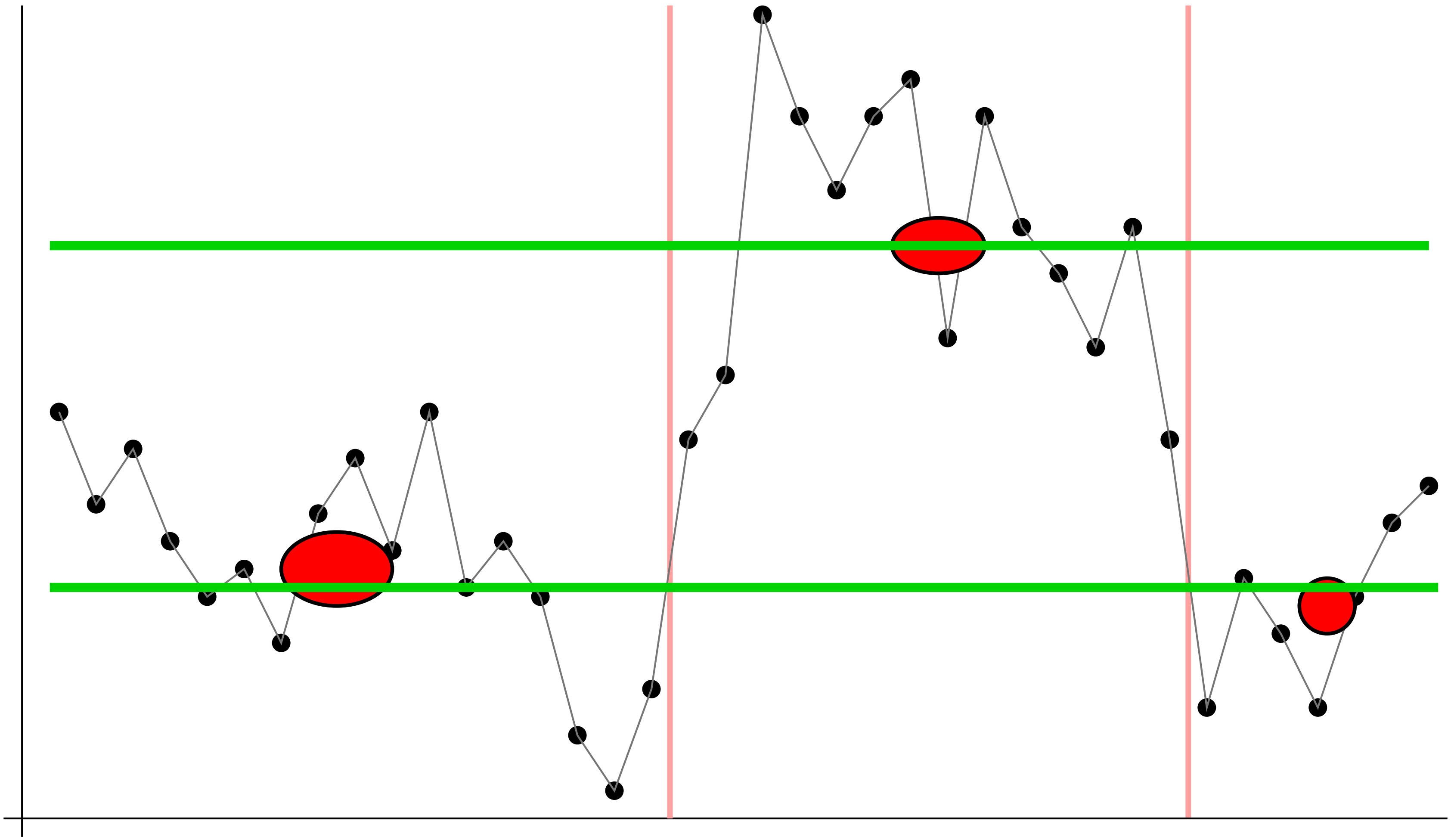
ClusterSegments example, $k = 3$, $h = 2$



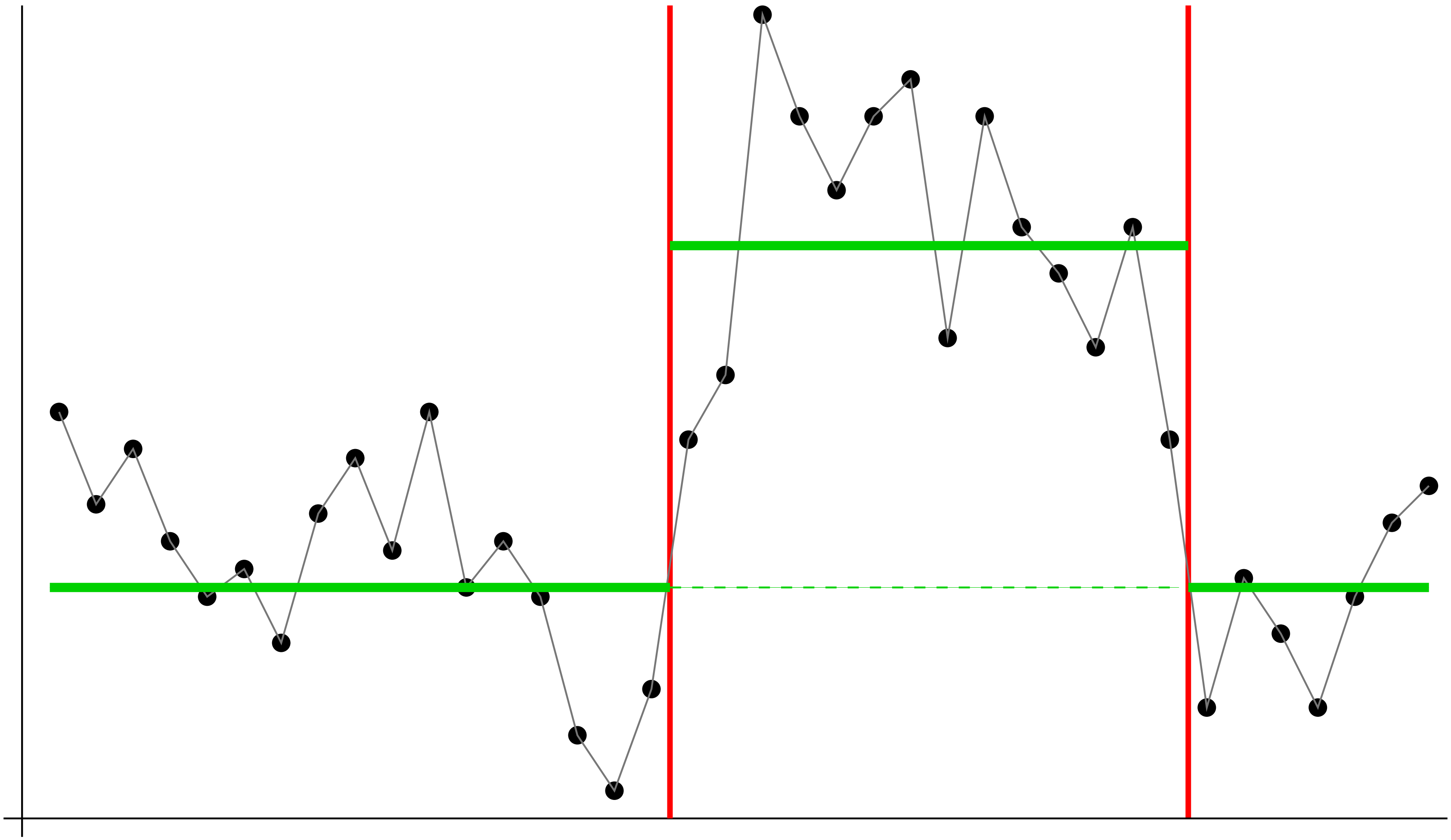
ClusterSegments example, $k = 3$, $h = 2$



ClusterSegments example, $k = 3$, $h = 2$



ClusterSegments example, $k = 3$, $h = 2$



Iterative algorithm

- If we know the k best segments, we can find the h best levels
- If we know the h best levels, we can find the k best segments
- Start from an initial solution,
  e.g., the one produced by the previous algorithm
- Iterate:
  - keep segment boundaries fixed, find levels
  - keep levels fixed, find boundaries
- EM-style, fast convergence, good results

Factor 3 approximation result for $\textsc{ClusterSegments}$

- d a metric
- Point i assigned to segment $p(i)$
- Segment j assigned to level $r(j)$
- Point i is assigned to level $l_{r(p(i))}$
- Minimize

$$\sum_{i=1}^{n} d(t_i, l_{r(p(i))})$$

- Claim: this is at most 3 times the optimum
- The segments are good segments $(k, k)-$segmentation
- The levels are good levels
- Some computation needed

Model section in (k, h)-segmentation

- Apply the BIC score function for model M_{kh}

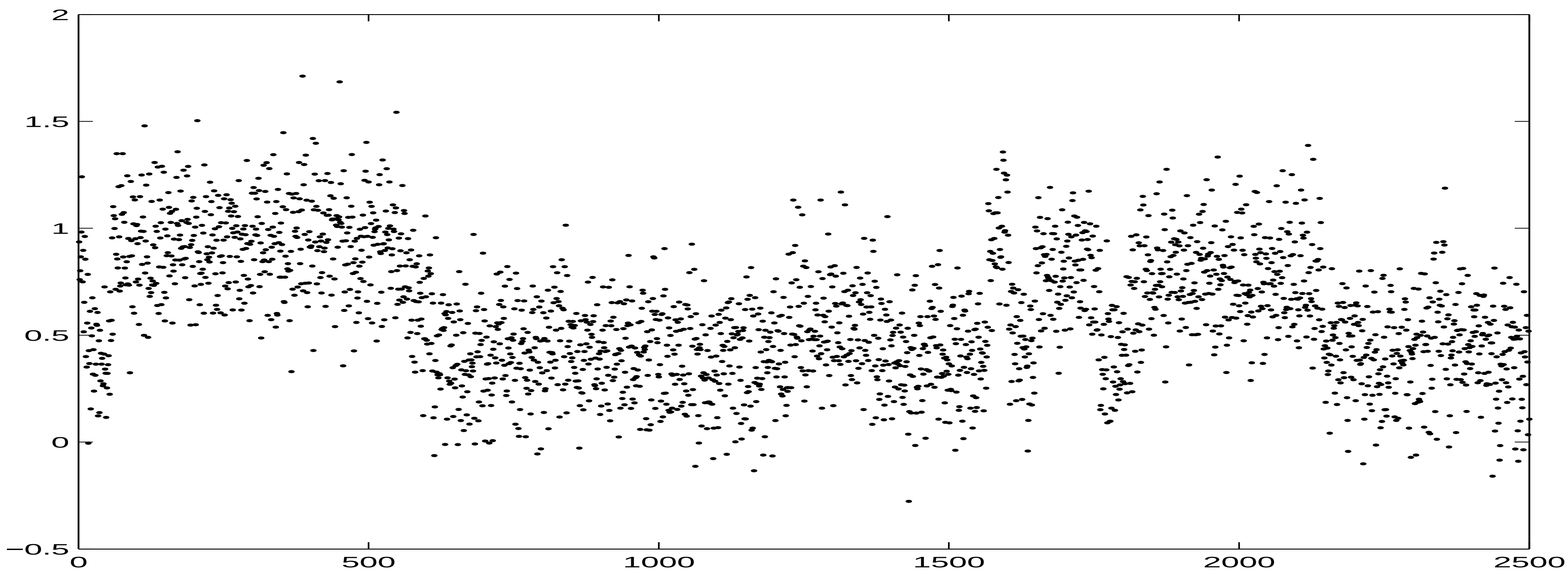
$$\begin{aligned} S_{BIC}(M_{kh}) &= 2LL(S \mid M_{kh}) + (\#\ \text{parameters}) \log n \\ &= \sum_{j=1}^{k} \sum_{i=b_j}^{e_j} (a_i - l_j)^2 + (k + hd) \log n \end{aligned}$$

  where $l_j \in L = \{l_1, \ldots, l_h\}$

- Normalize the sequence so that the total variance is equal to 1
- In case of multidimensional sequences do not normalize dimensions separately, so that dimensions with larger variability contribute more to the segmentation

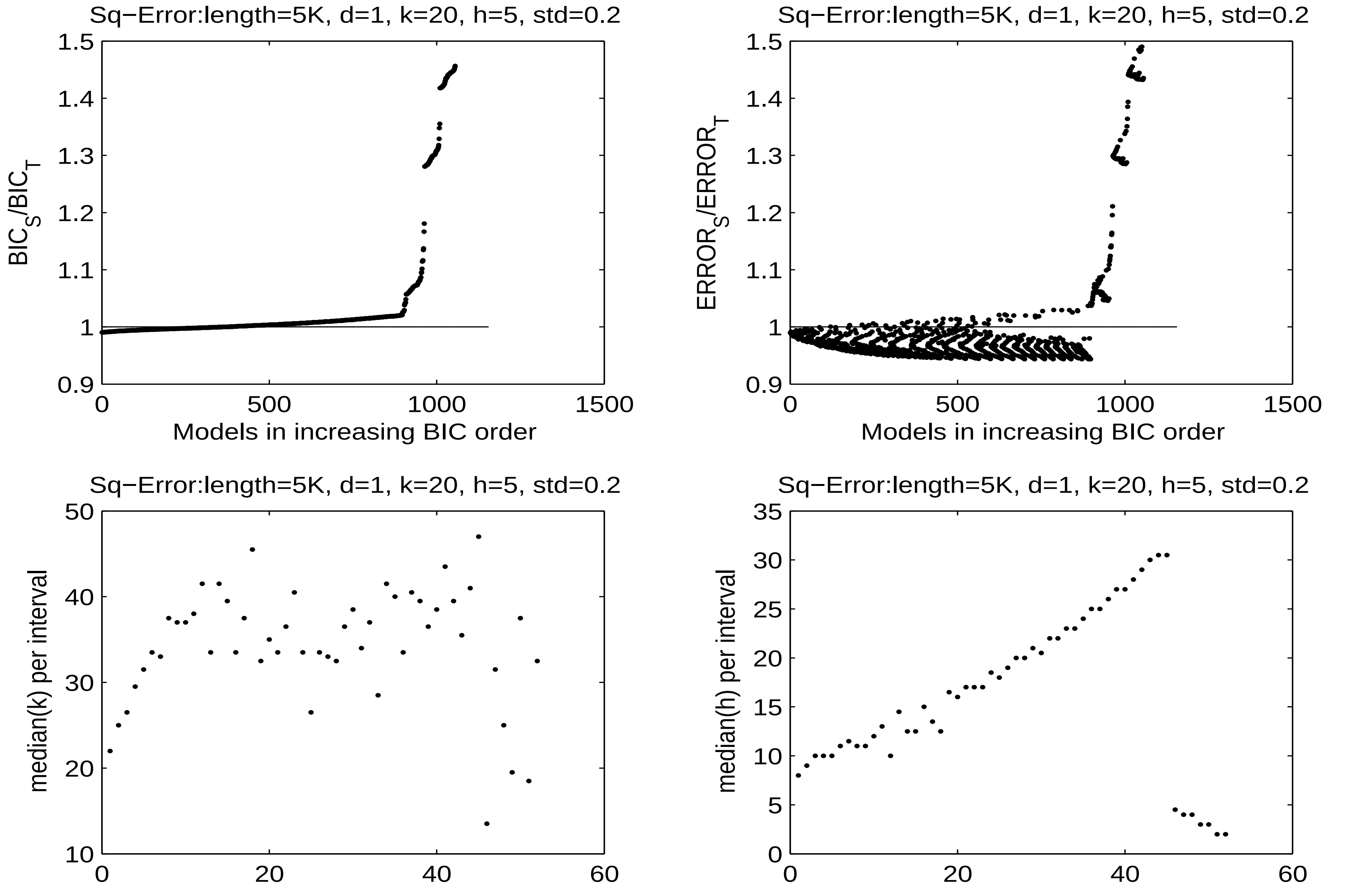
Experiments with model section in (k, h)-segmentation

- Synthetic datasets from a known model with k segments and h levels
- For segment S_j, points are generated from the normal distribution $\mathcal{N}(l_j, \sigma)$
- Verify that the model found by BIC agrees with the ground truth



$k = 20,\ h = 5$

Experiments with model section in (k, h)-segmentation



Hidden Markov models (HMMs)

- h states, m symbols, sequence $S = a_1 \ldots a_n$ of n observations, where $a_i \in \{\alpha_1, \ldots, \alpha_m\}$
- Sequence has been generated according to the following process
  - at state u, symbol α is emitted with probability $Q[u, \alpha]$
  - move from state u to state v with probability $P[u, v]$

  (only symbols are observed, states are hidden)
- Question: Given the sequence S, what is the most likely sequence of states that has generated S?
- Assume sequence of states $\Lambda = \lambda_1 \ldots \lambda_n$. Then

$$Pr[\Lambda \mid S] \propto Pr[S \mid \Lambda] = Pr[\lambda_1]Q[\lambda_1, a_1] \prod_{i=2}^{n} P[\lambda_{i-1}, \lambda_i]Q[\lambda_i, a_i]$$

HMMs (cont.)

- Denote by $\Theta = (P, Q)$ the parameters of HMM
- The two HMM problems:
  - if the HMM parameters Θ are known determine the sequence of states Λ that maximizes $Pr[S, \Lambda \mid \Theta]$ given the model
  - determine the HMM parameters Θ in order to maximize $Pr[S \mid \Theta]$
- The first problem can be solved by dynamic programming
- In this context it is also known as the "Viterby algorithm"

Estimating the parameters of an HMM

- Improve Θ iterative so that $Pr[S \mid \Theta]$ increases
- Stop when $Pr[S \mid \Theta]$ converges
- Iterative improving of $Pr[S \mid \Theta]$ is based on Baum-Welch equations
- Essentially are the emperical probabilities
  - $P[u, v]$ is the expected number of times of making a transition from u to v over the expected number of times being in u
  - $Q[u, \alpha]$ is the expected number of times of emitting symbol α while being in state u over the expected number of times being in u
- However, the emperical probabilties are estimated over all possible state sequences, which can be done (again) by dynamic programming

Connection between HMMs and (k,h)-segmentation

- Want to find Λ:

$$\arg\max_{\Lambda} Pr[\Lambda \mid S] = \arg\max_{\Lambda} Pr[\lambda_1]Q[\lambda_1, a_1]\prod_{i=2}^{n} P[\lambda_{i-1},\lambda_i]Q[\lambda_i, a_i]$$

- Assume k state changes, $\Lambda = \lambda_1 \ldots \lambda_k$ and corresponding segments of the sequence $S = S_1 \ldots S_k$, with durations $t_j = |S_j|$

$$\begin{aligned} Pr[S \mid \Lambda] &= Pr[\lambda_1]Q[\lambda_1, a_1]\prod_{i=2}^{n} P[\lambda_{i-1},\lambda_i]Q[\lambda_i, a_i] \\ &= Pr[\lambda_1]Pr[S_1 \mid \lambda_1]P[\lambda_1,\lambda_1]^{t_1-1} \\ &\quad \prod_{j=2}^{k} P[\lambda_{j-1},\lambda_j]Pr[S_j \mid \lambda_j]P[\lambda_j,\lambda_j]^{t_j-1} \end{aligned}$$

Connection between HMMs and (k, h)-segmentation

- Assume "uniform" transition probabilities, i.e.,

$$Pr[u,v] = \begin{cases} p & \text{if u=v} \\ 1 - p/h & \text{otherwise} \end{cases}$$

Then

$$Pr[\Lambda \mid S] = \prod_{j=1}^{k} Pr[S_j \mid \lambda_j])(p^{n-k}(1 - \frac{p}{h})^k)$$

- Taking minus logarithms, we want to minimize

$$-LL(S \mid \Lambda) + f(n, k, h)$$

very similar to (k, h)-segmentation with BIC!

Connection between HMMs and (k, h)-segmentation

- So, (k, h)-segmentation is more restricted than HMM (uniform case)
- More appropriate to numeric values, while HMM more common for descrete symbols
- Question: Can one generalize the (k, h)-segmentation techniques to the general HMM problem?