

Learning discrete time series with Bayesian Context Trees

Statistics & Operations Research Seminar

Department of Mathematics, NKUA

May 2022

Ioannis Kontoyiannis

Statistical Laboratory, U of Cambridge

Joint work with **I. Papageorgiou** and **V. Lungu**, also based on earlier joint work with *A. Panotopoulou, L. Mertzanis, M. Skoularidou*



Outline

Background

Discrete time series

Variable-memory Markov chains

Outline

Background

Discrete time series

Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Outline

Background

Discrete time series

Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms

CTW, BCT, k -BCT, MCMC

Outline

Background

Discrete time series

Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms

CTW, BCT, k -BCT, MCMC

Applications

Model selection

Segmentation

Filtering

Causality testing

Estimation

Anomaly detection

Prediction

Compression

Content recognition

Markov order estimation

Entropy estimation

Change-point detection

Outline

Background

Discrete time series

Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms

CTW, BCT, k -BCT, MCMC

Applications

Model selection

Segmentation

Filtering

Causality testing

Estimation

Anomaly detection

Prediction

Compression

Content recognition

Markov order estimation

Entropy estimation

Change-point detection

~> **Large-alphabets** and **continuous** time series!

Motivation

~> **Discrete time series are often hard**

Inference

Machine learning

Signal processing

Communications

Motivation

~> **Discrete time series are often hard**

Inference

Machine learning

Signal processing

Communications

~> **Difficulty: Memory modelling**

E.g. for a binary time series with memory length of only 20 bits

2^{20} parameters must be estimated before even getting started

~> **Need astronomical amounts of data**

~> **Need smarter, parsimonious models**

~> **Variable-memory Markov chains**

Variable-memory Markov chain models

Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $A = \{0, 1, \dots, m - 1\}$
of size m

Variable-memory Markov chain models

Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $A = \{0, 1, \dots, m - 1\}$
of size m

Memory length d $P(X_n | X_{n-1}, X_{n-2}, \dots) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-d})$

Variable-memory Markov chain models

Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $A = \{0, 1, \dots, m - 1\}$
of size m

Memory length d $P(X_n | X_{n-1}, X_{n-2}, \dots) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-d})$

Distribution To fully describe it, we need to specify
 m^d conditional distributions $P(X_n | X_{n-1}, \dots, X_{n-d})$

Variable-memory Markov chain models

Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $\mathbf{A} = \{0, 1, \dots, m - 1\}$
of size m

Memory length d $P(X_n | X_{n-1}, X_{n-2}, \dots) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-d})$

Distribution To fully describe it, we need to specify
 m^d conditional distributions $P(X_n | X_{n-1}, \dots, X_{n-d})$

Problem m^d grows very fast!

Variable-memory Markov chain models

Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $\mathbf{A} = \{0, 1, \dots, m - 1\}$
of size m

Memory length d $P(X_n | X_{n-1}, X_{n-2}, \dots) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-d})$

Distribution To fully describe it, we need to specify
 m^d conditional distributions $P(X_n | X_{n-1}, \dots, X_{n-d})$

Problem m^d grows very fast!

Idea Use *variable length contexts* described by a **context tree** T

Variable-memory Markov chain models

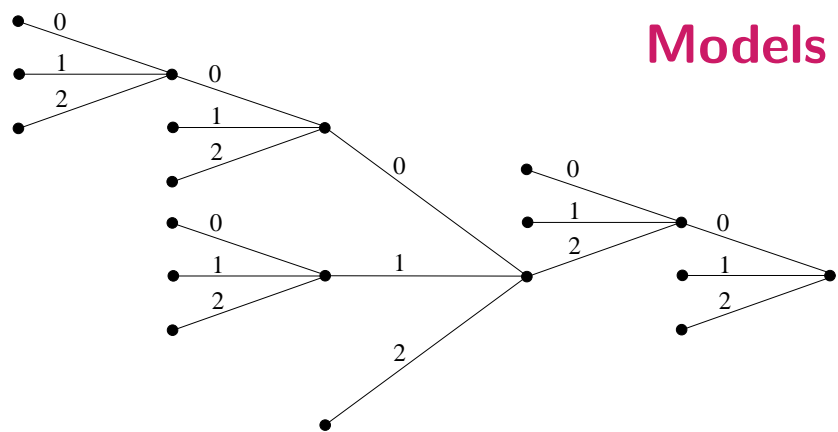
Markov chain $\{\dots, X_0, X_1, \dots\}$ with **alphabet** $\mathbf{A} = \{0, 1, \dots, m - 1\}$ of size m

Memory length d $P(X_n | X_{n-1}, X_{n-2}, \dots) = P(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-d})$

Distribution To fully describe it, we need to specify m^d conditional distributions $P(X_n | X_{n-1}, \dots, X_{n-d})$

Problem m^d grows very fast!

Idea Use *variable length contexts* described by a **context tree** T

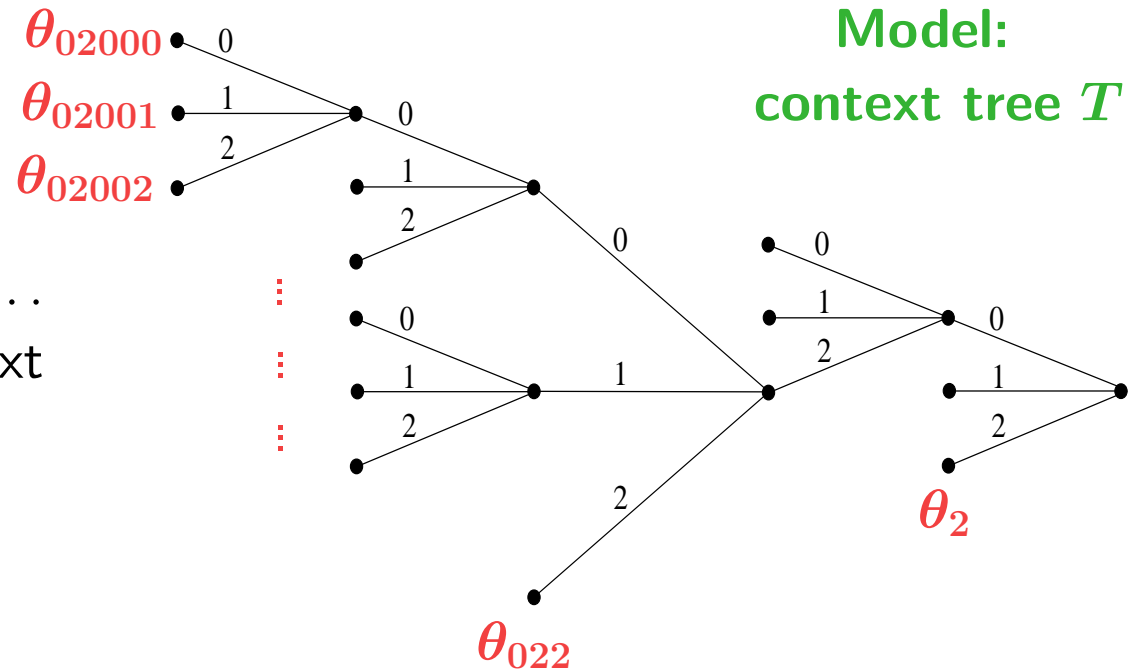


Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols

Memory length $d = 5$

Each past string X_{n-1}, X_{n-2}, \dots
corresponds to a unique context
on a leaf of the tree



Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols

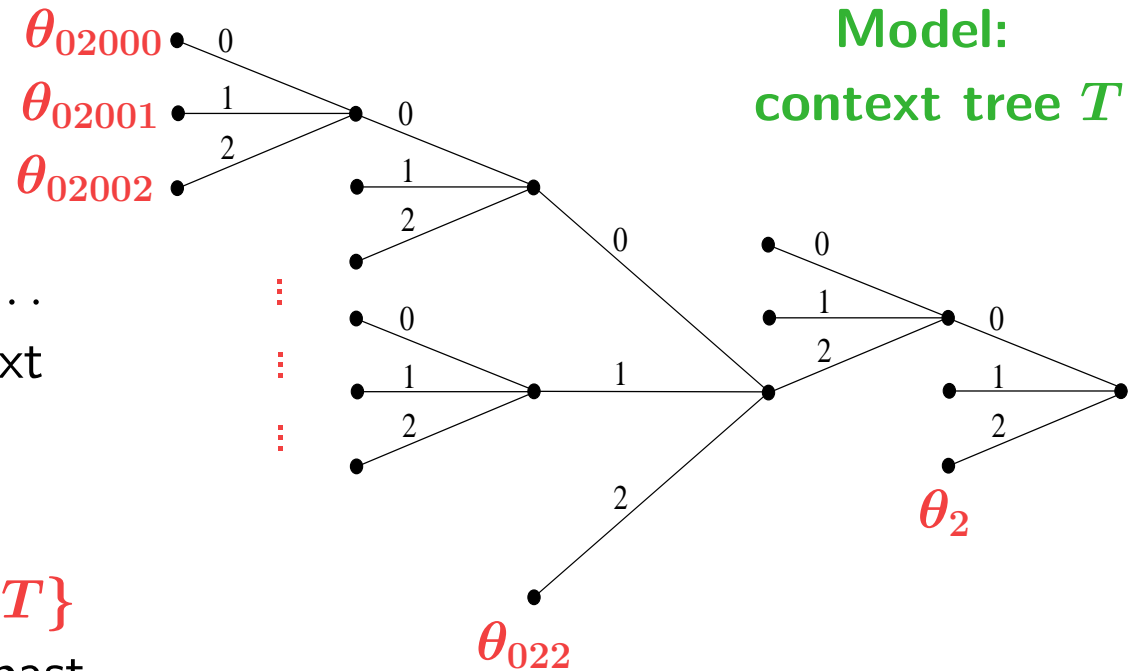
Memory length $d = 5$

Each past string X_{n-1}, X_{n-2}, \dots corresponds to a unique context on a leaf of the tree

Parameters: $\theta = \{\theta_s ; s \in T\}$

The distr of X_n given the past is given by the distr on that leaf

E.g. $P(X_n = 1 | X_{n-1} = 0, X_{n-2} = 2, X_{n-2} = 2, X_{n-3} = 1, \dots) = \theta_{022}(1)$



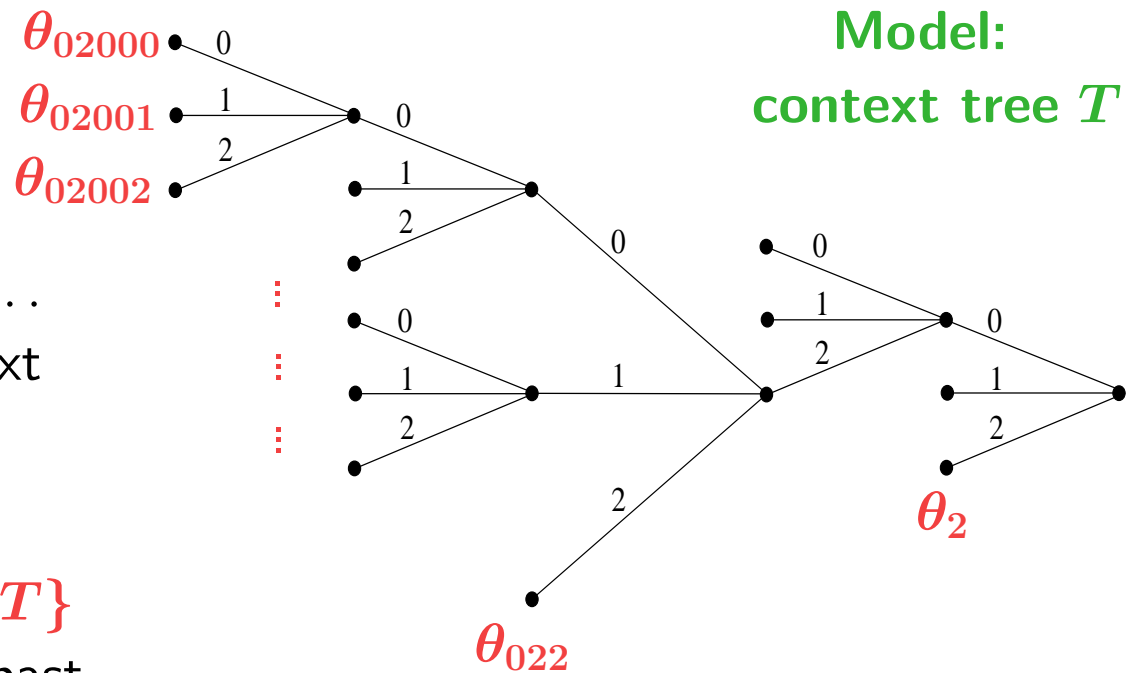
Model:
context tree T

Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols

Memory length $d = 5$

Each past string X_{n-1}, X_{n-2}, \dots corresponds to a unique context on a leaf of the tree



Model:
context tree T

Parameters: $\theta = \{\theta_s ; s \in T\}$

The distr of X_n given the past is given by the distr on that leaf

E.g. $P(X_n = 1 | X_{n-1} = 0, X_{n-2} = 2, X_{n-2} = 2, X_{n-3} = 1, \dots) = \theta_{022}(1)$

~> **Parsimony**

Instead of $3^5 = 243$ conditional distributions only need 13

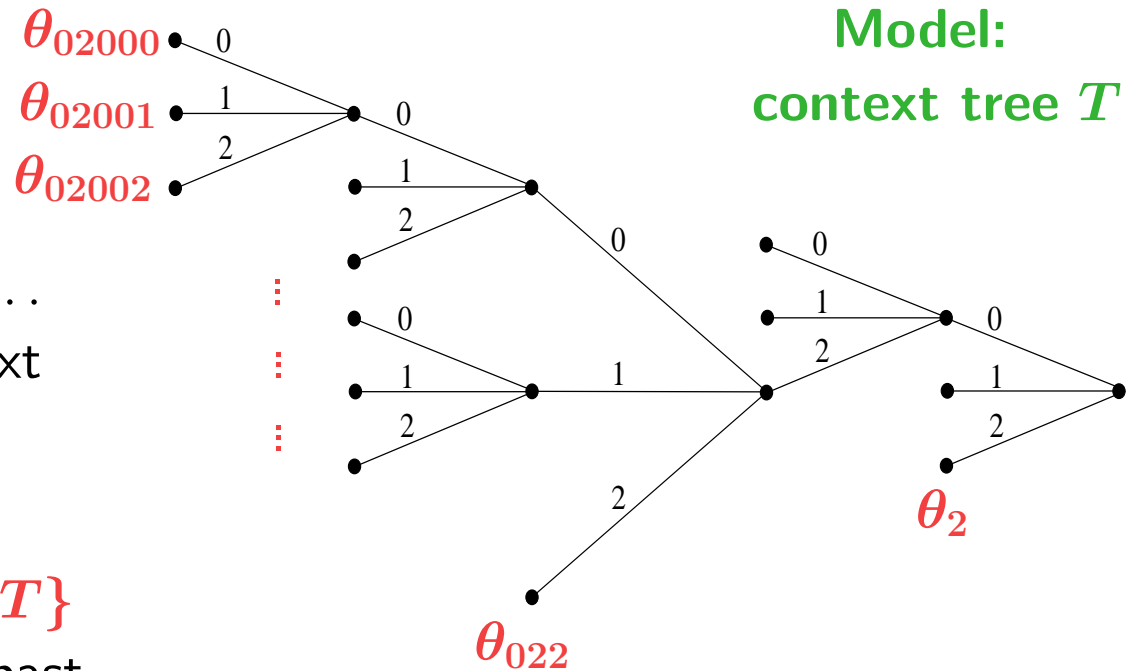
\Rightarrow potentially huge savings in general

Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols

Memory length $d = 5$

Each past string X_{n-1}, X_{n-2}, \dots corresponds to a unique context on a leaf of the tree



Parameters: $\theta = \{\theta_s ; s \in T\}$

The distr of X_n given the past is given by the distr on that leaf

E.g. $P(X_n = 1 | X_{n-1} = 0, X_{n-2} = 2, X_{n-2} = 2, X_{n-3} = 1, \dots) = \theta_{022}(1)$

~> Parsimony

Instead of $3^5 = 243$ conditional distributions only need 13

\Rightarrow potentially huge savings in general

~> Determining the underlying context tree of an empirical time series is of great scientific and engineering interest

Bayesian modelling for VMMCs:

The Bayesian Context Trees (BCT) framework

Prior on models Indexed family of priors on trees T of depth $\leq D$

Given m, D , for each $\beta \in (0, 1)$:

$$\pi(T) = \pi_D(T; \beta) = \alpha^{|T|-1} \beta^{|T|-L_D(T)}$$

with $\alpha = (1 - \beta)^{1/(m-1)}$; $|T| = \#$ leaves of T ; $L_D(T) = \#$ leaves at D

Bayesian modelling for VMMCs:

The Bayesian Context Trees (BCT) framework

Prior on models Indexed family of priors on trees T of depth $\leq D$

Given m, D , for each $\beta \in (0, 1)$:

$$\pi(T) = \pi_D(T; \beta) = \alpha^{|T|-1} \beta^{|T|-L_D(T)}$$

with $\alpha = (1 - \beta)^{1/(m-1)}$; $|T| = \#$ leaves of T ; $L_D(T) = \#$ leaves at D

Prior on parameters Given model T , the parameters $\theta = \{\theta_s; s \in T\}$ are taken independent with each $\pi(\theta_s | T) \sim \text{Dirichlet}(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$

Bayesian modelling for VMMCs:

The Bayesian Context Trees (BCT) framework

Prior on models Indexed family of priors on trees T of depth $\leq D$

Given m, D , for each $\beta \in (0, 1)$:

$$\pi(T) = \pi_D(T; \beta) = \alpha^{|T|-1} \beta^{|T|-L_D(T)}$$

with $\alpha = (1 - \beta)^{1/(m-1)}$; $|T| = \#$ leaves of T ; $L_D(T) = \#$ leaves at D

Prior on parameters Given model T , the parameters $\theta = \{\theta_s; s \in T\}$ are taken independent with each $\pi(\theta_s | T) \sim \text{Dirichlet}(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$

Observations $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$

Write $X_i^j = (X_i, X_{i+1}, \dots, X_j)$ and suppress initial context X_{-D+1}^0

Bayesian modelling for VMMCs:

The Bayesian Context Trees (BCT) framework

Prior on models Indexed family of priors on trees T of depth $\leq D$

Given m, D , for each $\beta \in (0, 1)$:

$$\pi(T) = \pi_D(T; \beta) = \alpha^{|T|-1} \beta^{|T|-L_D(T)}$$

with $\alpha = (1 - \beta)^{1/(m-1)}$; $|T| = \#$ leaves of T ; $L_D(T) = \#$ leaves at D

Prior on parameters Given model T , the parameters $\theta = \{\theta_s; s \in T\}$ are taken independent with each $\pi(\theta_s | T) \sim \text{Dirichlet}(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$

Observations $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$

Write $X_i^j = (X_i, X_{i+1}, \dots, X_j)$ and suppress initial context X_{-D+1}^0

Likelihood Given model T and parameters $\theta = \{\theta_s; s \in T\}$:

$$f(X_1^n | X_{-D+1}^0, \theta, T) = \prod_{s \in T} \prod_{j \in A} \theta_s(j)^{a_s(j)}$$

where $a_s(j) = \#$ times letter j follows context s in X_1^n

Bayesian inference for VMMCs

Given. Data $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Max model depth D

“The” goal of Bayesian inference

Determination of the **posterior distributions**:

$$\pi(\theta, T|X) = \frac{\pi(T)\pi(\theta|T)f(X|\theta, T)}{f(X)}$$

and

$$\pi(T|X) = \frac{\int_{\theta} f(X|\theta, T)\pi(\theta|T) d\theta \pi(T)}{f(X)}$$

Bayesian inference for VMMCs

Given. Data $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
 Max model depth D

“The” goal of Bayesian inference

Determination of the **posterior distributions**:

$$\pi(\theta, T|X) = \frac{\pi(T)\pi(\theta|T)f(X|\theta, T)}{f(X)}$$

and

$$\pi(T|X) = \frac{\int_{\theta} f(X|\theta, T)\pi(\theta|T) d\theta \pi(T)}{f(X)}$$

Main obstacle

Determination of the **prior predictive likelihood**:

$$f(X) = \sum_T \pi(T) \int_{\theta} f(X|\theta, T)\pi(\theta|T) d\theta$$

↪ the number of models in the sum grows *doubly exponentially* in D

Computation of the marginal likelihood

Given the structure of the model
the **marginal likelihoods** $f(X|T)$ can be computed explicitly

Lemma The *marginal likelihood* $f(X|T)$ can be computed as

$$f(X|T) = \prod_{s \in T} P_e(a_s)$$

where $P_e(a_s) = \frac{\prod_{j=0}^{m-1} [(1/2)(3/2) \cdots (a_s(j) - 1/2)]}{(m/2)(m/2 + 1) \cdots (m/2 + M_s - 1)}$

with the count vectors $a_s = (a_s(0), \dots, a_s(m-1))$ as before
and $M_s = a_s(0) + \cdots + a_s(m-1)$

Computation of the marginal likelihood

Given the structure of the model
the **marginal likelihoods** $f(X|T)$ can be computed explicitly

Lemma The *marginal likelihood* $f(X|T)$ can be computed as

$$f(X|T) = \prod_{s \in T} P_e(a_s)$$

where $P_e(a_s) = \frac{\prod_{j=0}^{m-1} [(1/2)(3/2) \cdots (a_s(j) - 1/2)]}{(m/2)(m/2 + 1) \cdots (m/2 + M_s - 1)}$

with the count vectors $a_s = (a_s(0), \dots, a_s(m-1))$ as before
and $M_s = a_s(0) + \cdots + a_s(m-1)$

What is quite surprising is that the entire
prior predictive likelihood $f(X) = \sum_T \pi(T) f(X|T)$
can also be computed effectively

The Context Tree Weighting algorithm (CTW)

Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, X_2, \dots, X_n$
Alphabet size m **Maximum depth** D
Prior parameter β

- \triangle **1.** [*Tree.*] Construct a tree with nodes corresponding to all contexts of length $1, 2, \dots, D$ contained in X
- \triangle **2.** [*Estimated probabilities.*] At each node s compute the vectors a_s
[$a_s(j) = \#$ times letter j follows context s in X_1^n]
and the probabilities $P_{e,s} = P_e(a_s)$ as in the Lemma
- \triangle **3.** [*Weighted probabilities.*] At each node s compute

$$P_{w,s} = \begin{cases} P_{e,s}, & \text{if } s \text{ is a leaf} \\ \beta P_{e,s} + (1 - \beta) \prod_{j \in A} P_{w,sj}, & \text{o/w} \end{cases}$$

The CTW computes the prior predictive likelihood

Theorem

The weighted probability $P_{w,\lambda}$ given by the CTW at the root λ is exactly equal to the prior predictive likelihood of the data X :

$$P_{w,\lambda} = f(X) = \sum_T \pi(T) \int_{\theta} f(X|\theta, T) \pi(\theta|T) d\theta$$

The CTW computes the prior predictive likelihood

Theorem

The weighted probability $P_{w,\lambda}$ given by the CTW at the root λ is exactly equal to the prior predictive likelihood of the data X :

$$P_{w,\lambda} = f(X) = \sum_T \pi(T) \int_{\theta} f(X|\theta, T) \pi(\theta|T) d\theta$$

Note

The CTW computes a “doubly exponentially hard” quantity in $O(nmD)$ time

The CTW can be updated *sequentially*

This is one of the very few examples of nontrivial Bayesian models for which the prior predictive likelihood is explicitly computable probably the most complex/interesting one

Some CTW pre-history

~> **1995**: The original CTW paper appears
Special case of CTW algorithm, that computes
an effective, adaptive probability distribution
that can be used with arithmetic coding
for **universal DATA COMPRESSION**

Tanya Ignatenko
Ali Nowbakht
Yuri Shtarkov
Tjalling Tjalkens
Paul Volf
Frans Willems

~> **1996-1998**: IEEE IT Soc best paper award
Extensions in various directions
with primary focus still on **data compression**

~> **1996-2006**: Parallel activity connects CTW with **STATISTICS**
Numerous papers (going back to original 1993 preprint)
explore the CTW in a Bayesian setting
introduce the Context Tree Maximising (CTM) algorithm
and examine applications to model selection [**Peter Bühlmann** et al]

~> **Today's talk**: Presents a principled, unified Bayesian framework
for general inference and learning tasks on discrete time series
centered around CTW

Bayesian Context Tree algorithm (BCT)

[The algorithm formerly known as
Context Tree Maximizing (CTM)]

Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, X_2, \dots, X_n$
Alphabet size m **Maximum depth** D
Prior parameter β

△ 1. [Tree.] and △ 2. [Estimated probabilities.]

Construct the tree and compute a_s and $P_{e,s}$ as before

△ 3. [Maximal probabilities.]

At each node s compute

$$P_{m,s} = \begin{cases} P_{e,s}, & \text{if } s \text{ is a leaf} \\ \max\{\beta P_{e,s}, (1 - \beta) \prod_{j \in A} P_{m,sj}\}, & \text{o/w} \end{cases}$$

△ 4. [Pruning.]

For each node s , if the above max is achieved
by the first term, then prune all its descendants

The BCT computes the MAP model

Theorem

The (pruned) tree T_1^* resulting from the BCT procedure has maximal *a posteriori* probability among all trees:

$$\pi(T_1^*|X) = \max_T \pi(T|X) = \max_T \left\{ \frac{\int_{\theta} f(X|\theta, T) \pi(\theta|T) d\theta \pi(T)}{f(X)} \right\}$$

The BCT computes the MAP model

Theorem

The (pruned) tree T_1^* resulting from the BCT procedure has maximal *a posteriori* probability among all trees:

$$\pi(T_1^*|X) = \max_T \pi(T|X) = \max_T \left\{ \frac{\int_{\theta} f(X|\theta, T) \pi(\theta|T) d\theta \pi(T)}{f(X)} \right\}$$

Note – as with the CTW

The BCT also computes a doubly exponentially hard quantity in $O(nmD)$ time

Again, one of the very few examples of nontrivial Bayesian models for which the mode of the posterior is explicitly computable probably the most complex/interesting one

A first empirical result: Simulated data

5th order VMMC data $X_{-D+1}, \dots, X_0, X_1, X_2, \dots, X_n$

Alphabet size $m = 3$

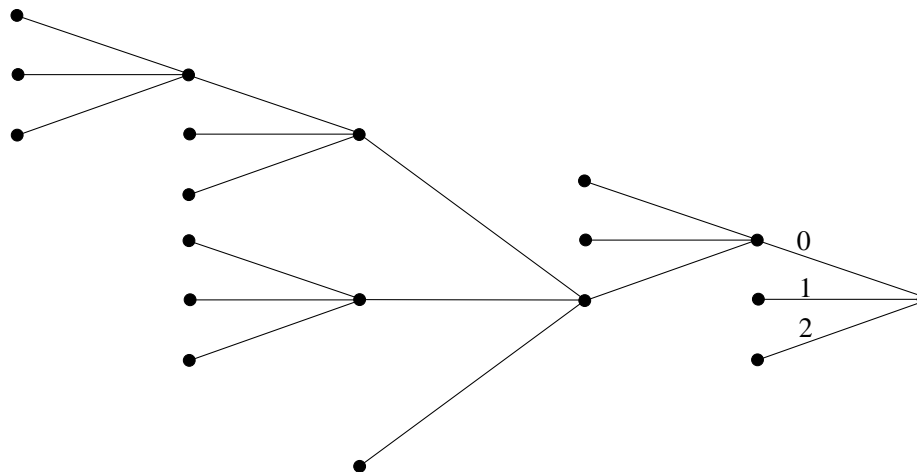
VMMC with $d = 5$ as in the example

Data length $n = 10000$ samples

BCT

MAP model with max depth $D = 5$, $\beta = 3/4$?

$\rightsquigarrow D = 5$: space of more than 10^{24} models



Finding the k a posteriori most likely models: The k -BCT algorithm

- △ 1. [*Construct full tree.*] △ 2. [*Compute a_s and $P_{e,s}$.*]
- △ 3. [*Matrix representation.*] Each node s contains a $k \times m$ matrix B_s
Line i represents the i th best subtree starting at s
Either entire line consists of * meaning “prune at s ”
Or j th element describes which line of the j child of s to follow
Line i also contains the “maximal probab” $P_{m,s}^{(i)}$ associated with i th subtree
- △ 4. [*At each leaf s .*] Entire matrix B_s contains *’s and all $P_{m,s}^{(i)}$ are = $P_{e,s}$
- △ 5. [*At each internal node s .*]
Consider all k^m combinations of subtrees of the children of s
For each combination compute the associated maximal prob as in BCT
Order the results by prob, keep the top k , describe them in the matrix B_s
- △ 6. [*Bottom-to-top-to-bottom.*] Repeat (5.) recursively until the root
Starting at the root, read the top k trees

The k -BCT identifies the k a posteriori most likely models

Theorem

The k trees $T_1^*, T_2^*, \dots, T_k^*$ described recursively at the root after the k -BCT procedure are the k a posteriori most likely models w.r.t.:

$$\pi(T|X) = \frac{\int_{\theta} f(X|\theta, T) \pi(\theta|T) d\theta \pi(T)}{f(X)}$$

Note

The complexity of k -BCT is $O(nmD \times k^m)$ in both time and space
Lower complexity implementations are possible

Additional results

(i) *Model posterior probabilities* $\pi(T|X) = \frac{\pi(T) \prod_{s \in T} P_e(a_s)}{P_{w,\lambda}}$

for ANY model T , where $P_{w,\lambda}$ = prior predictive likelihood and $P_e(a_s) = P_{e,s}$ are the estimated probabilities in CTW

(ii) *Posterior odds* $\frac{\pi(T|X)}{\pi(T'|X)} = \frac{\pi(T) \prod_{s \in T, s \notin T'} P_e(a_s)}{\pi(T') \prod_{s \in T', s \notin T} P_e(a_s)}$.

for ANY pair of models T, T'

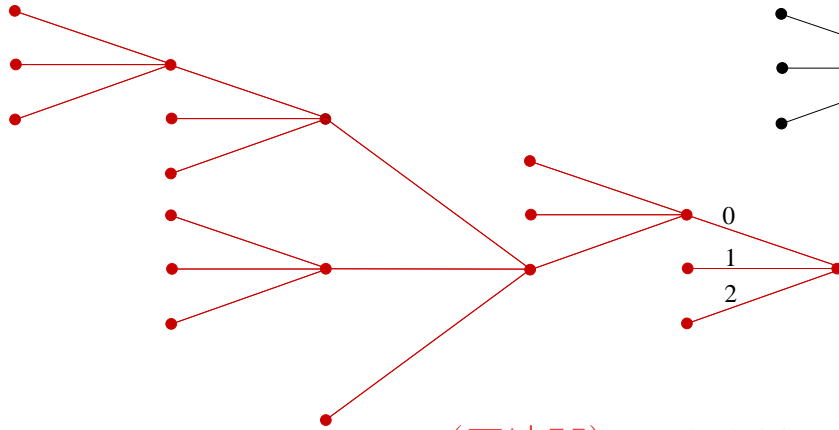
(iii) *Full conditional density of θ*

$$\pi(\theta|T, X) \sim \prod_{s \in T} \text{Dirichlet}(a_s(0) + 1/2, a_s(1) + 1/2, \dots, a_s(m-1) + 1/2)$$

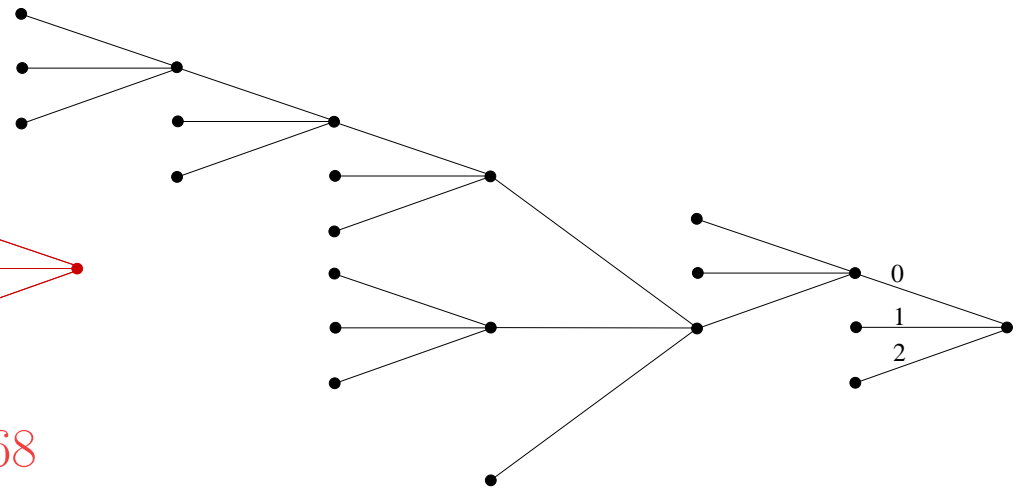
k -BCT models for the same 5th order chain

$D = 10 \rightsquigarrow$ more than 10^{5900} models

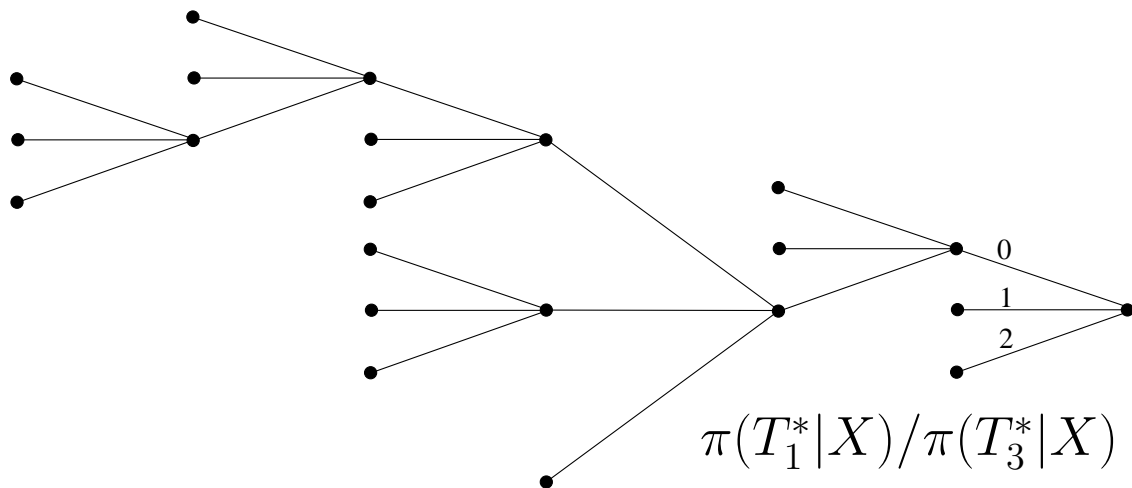
$n = 10000, k = 3, \beta = 3/4$



$$\pi(T_1^*|X) \approx 0.368$$
$$\pi(T_1^*) \approx 3.8 \times 10^{-6}$$



$$\pi(T_1^*|X)/\pi(T_2^*|X) \approx 6.29$$



$$\pi(T_1^*|X)/\pi(T_3^*|X) \approx 8.82$$

MCMC exploration of the posterior

Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
 Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

MCMC exploration of the posterior

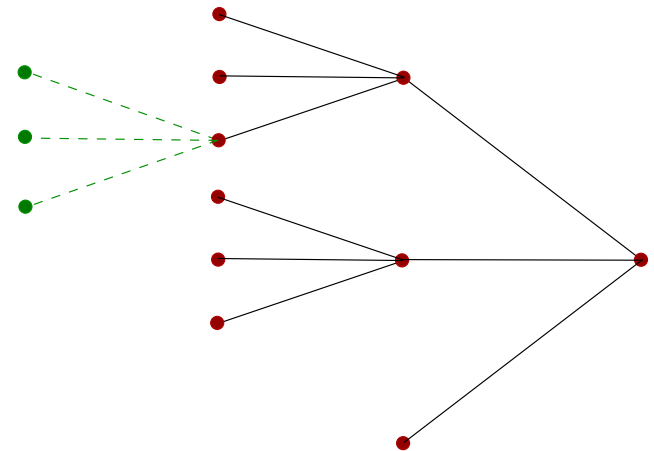
Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

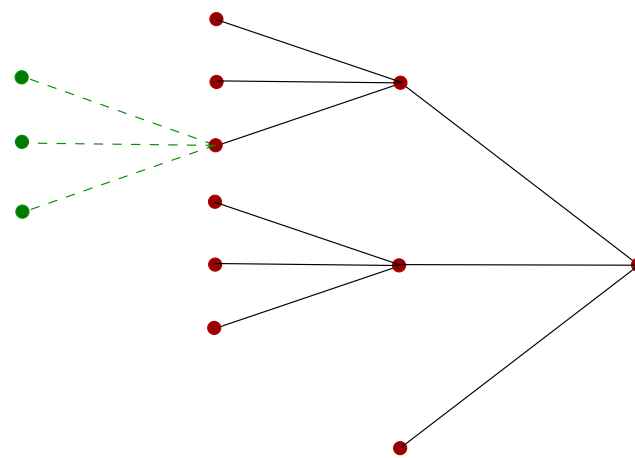
Iterate: At each t :

\triangle [Metropolis proposal] Given $T(t)$ propose T'
by randomly adding or removing m sibling leaves



MCMC exploration of the posterior

Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β



Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Iterate: At each t :

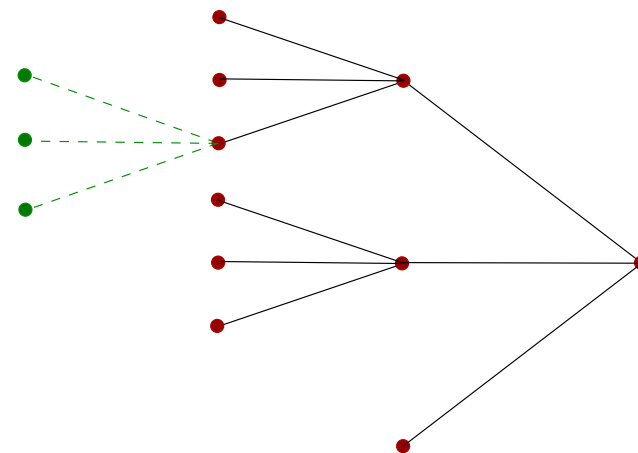
△ [Metropolis proposal] Given $T(t)$ propose T'
 by randomly adding or removing m sibling leaves

△ [Metropolis step] Define $T(t+1)$ by accepting or rejecting T'

$$\frac{\pi(T'|X)}{\pi(T(t)|X)} = \frac{\pi(T') \prod_{s \in T', s \notin T(t)} P_e(a_s)}{\pi(T(t)) \prod_{s \in T(t), s \notin T'} P_e(a_s)}$$

MCMC exploration of the posterior

Given. **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β



Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Iterate: At each t :

△ [Metropolis proposal] Given $T(t)$ propose T'
 by randomly adding or removing m sibling leaves

△ [Metropolis step] Define $T(t+1)$ by accepting or rejecting T'

$$\frac{\pi(T'|X)}{\pi(T(t)|X)} = \frac{\pi(T') \prod_{s \in T', s \notin T(t)} P_e(a_s)}{\pi(T(t)) \prod_{s \in T(t), s \notin T'} P_e(a_s)}$$

△ [Gibbs step] Take $\theta(t+1) \sim$ sample from the full cond'al density

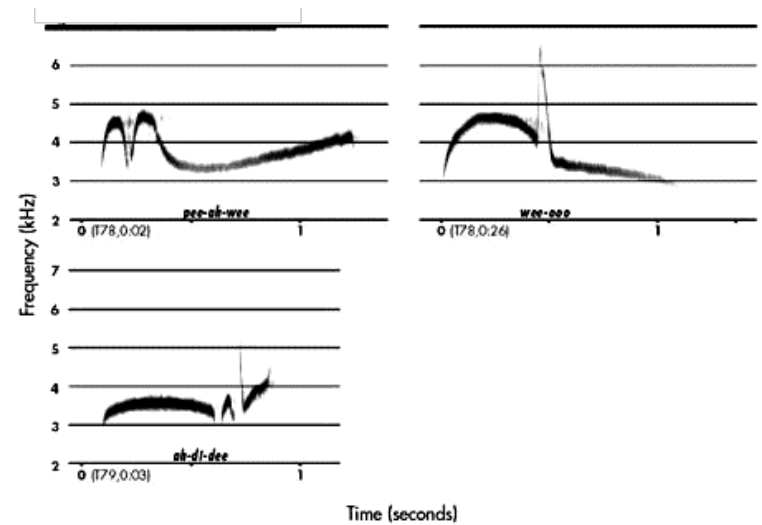
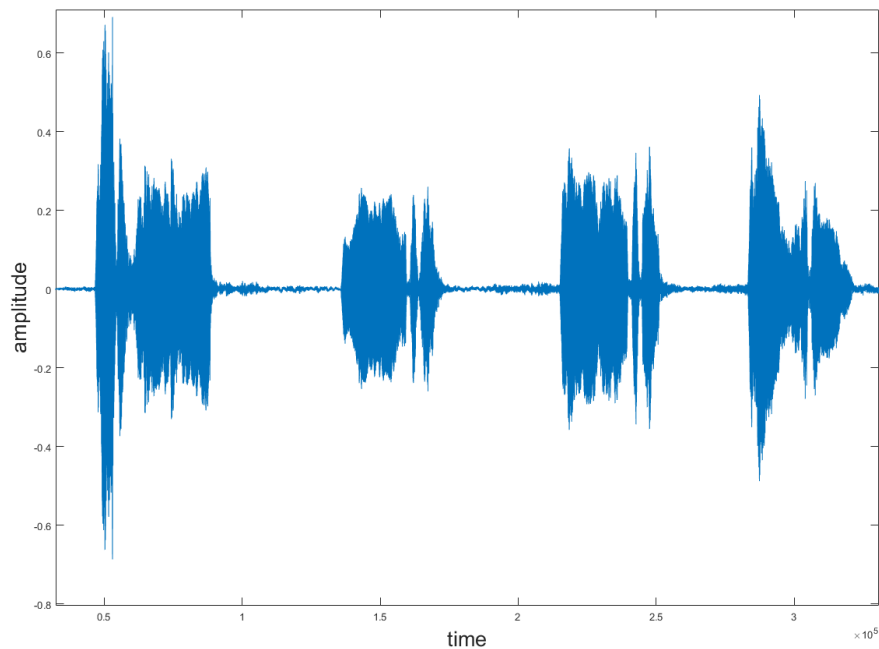
$$\prod_{s \in T(t+1)} \text{Dirichlet}(a_s(0) + 1/2, a_s(1) + 1/2, \dots, a_s(m-1) + 1/2)$$

A fun data set: Wood Pewee bird song

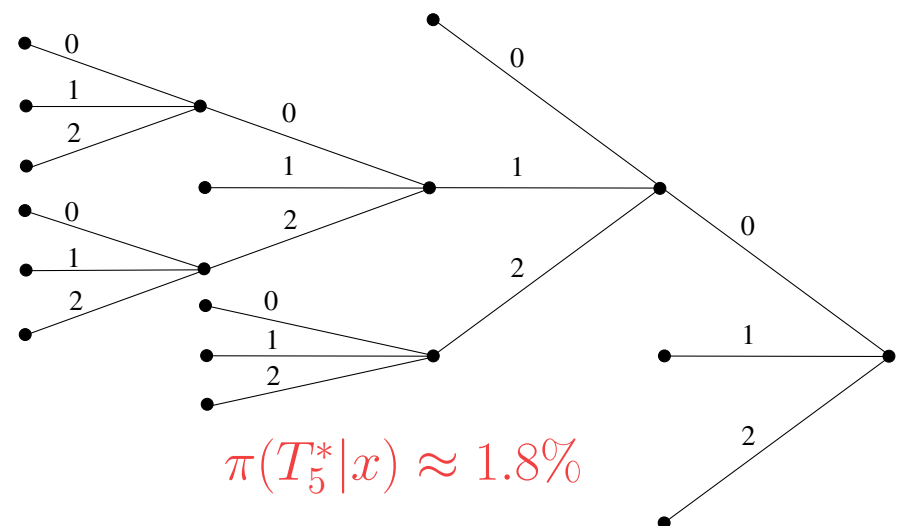
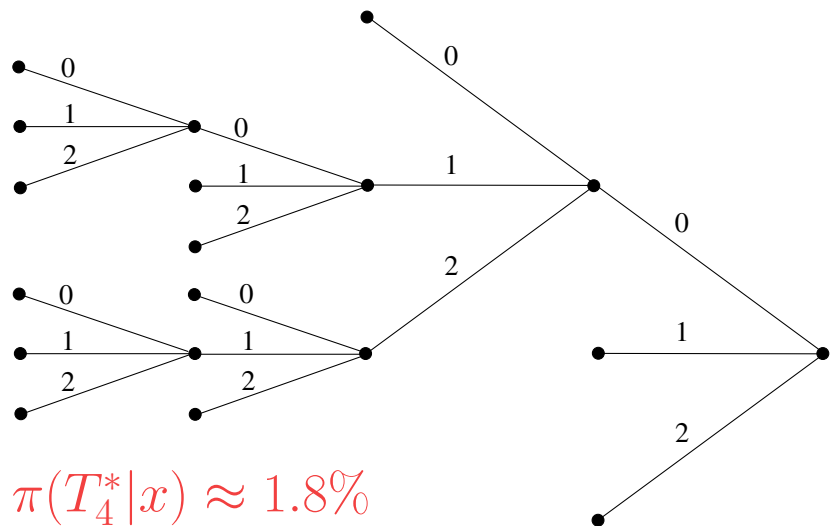
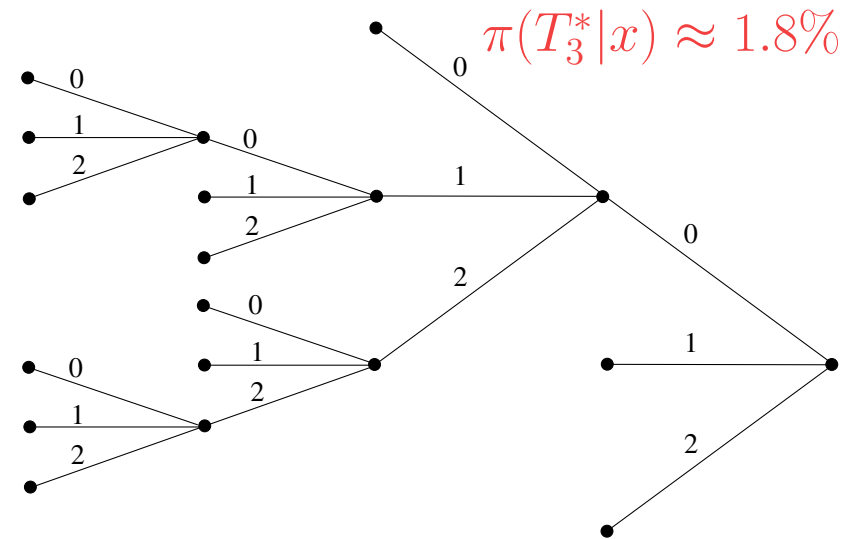
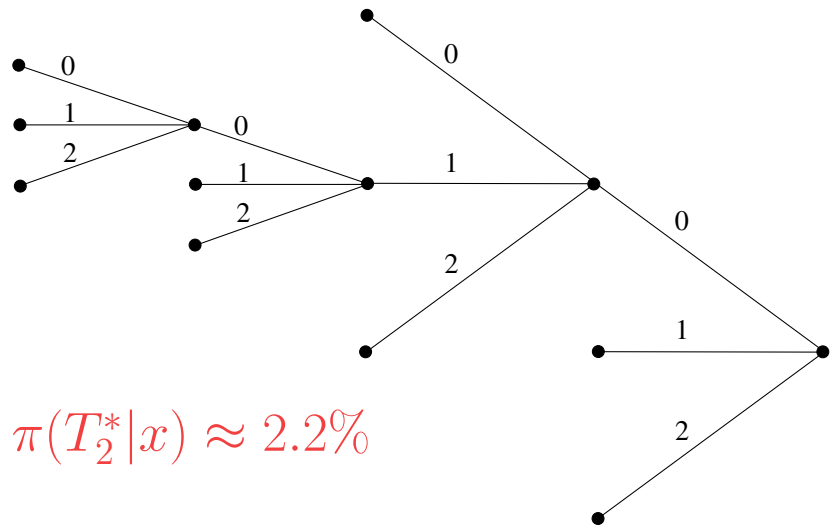
Data Recorded bird song data, transcribed as a sequence of (mono-)phthongs
Goal: Understand structure, complexity, variation and function

[Craig (1943) “The song of the wood pewee”]

[Berchtold-Raftery (2002) “The MTD model”]

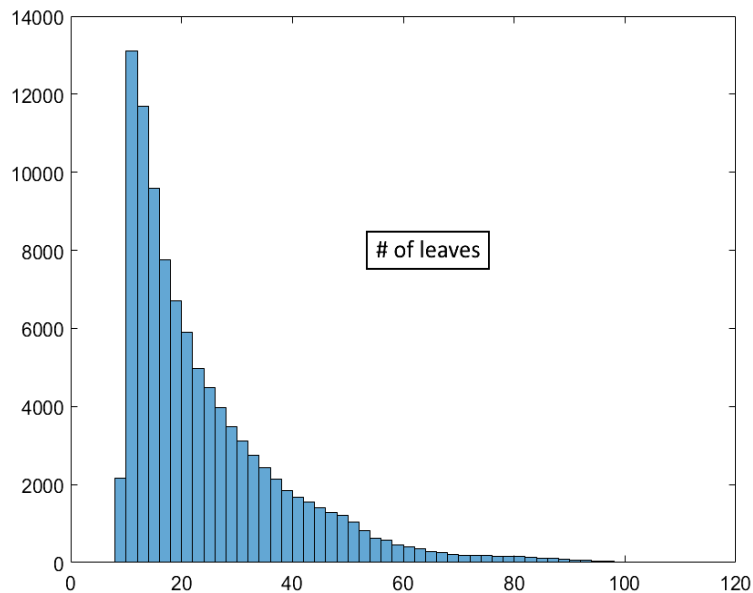
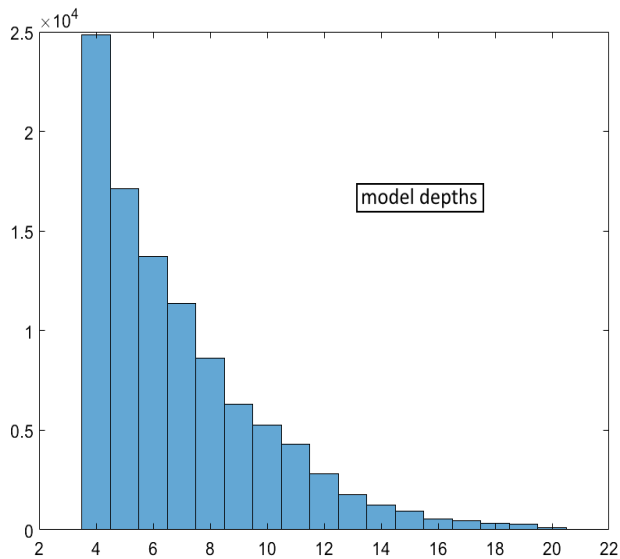


Wood Peewee bird song: Next 4 models



Total posterior of top 5 models $\approx 20.2\%$

MCMC results on Peewee data

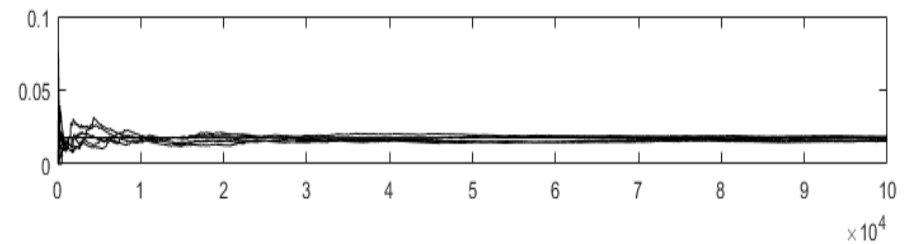
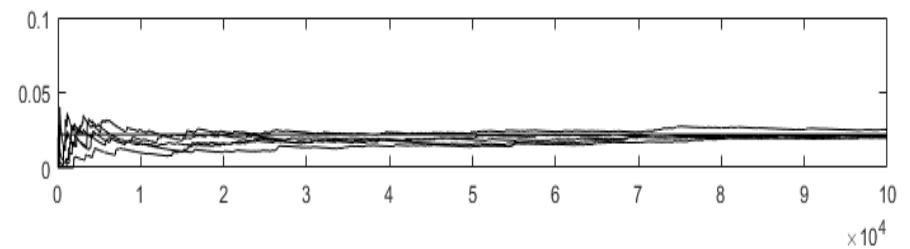
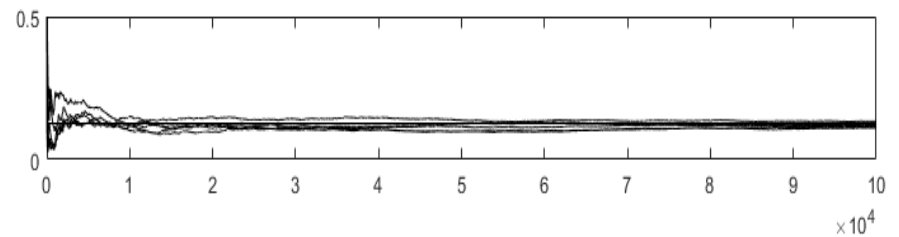


After 10^6 iterations:

Acceptance rate $\approx 59 - 61\%$

$\approx 306,000$ models visited

posterior of models visited $\approx 61.3\%$



[\rightsquigarrow Markov order estimation]

Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
 Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β

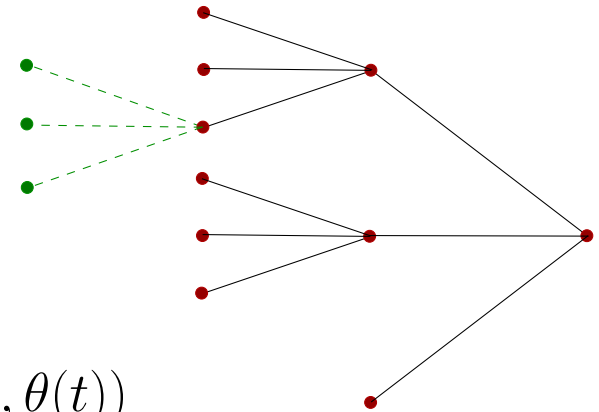
Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Iterate: At each t :

△ [Metropolis-Gibbs step]

Obtain model and parameters sample $(T(t), \theta(t))$



Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

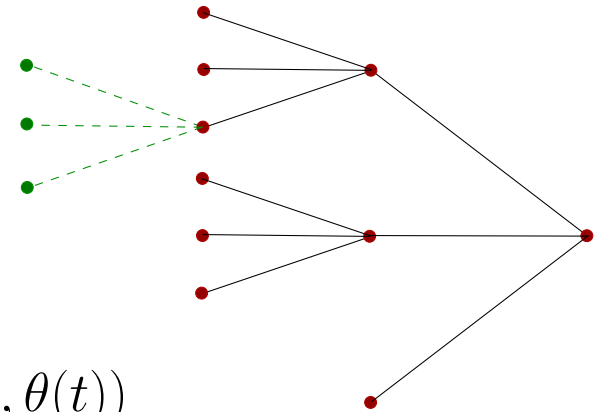
Iterate: At each t :

△ [Metropolis-Gibbs step]

Obtain model and parameters sample $(T(t), \theta(t))$

△ [Entropy step] Obtain entropy sample $H(t)$:

↪ either compute $H(t) = H(T(t), \theta(t))$ directly



Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Iterate: At each t :

△ [Metropolis-Gibbs step]

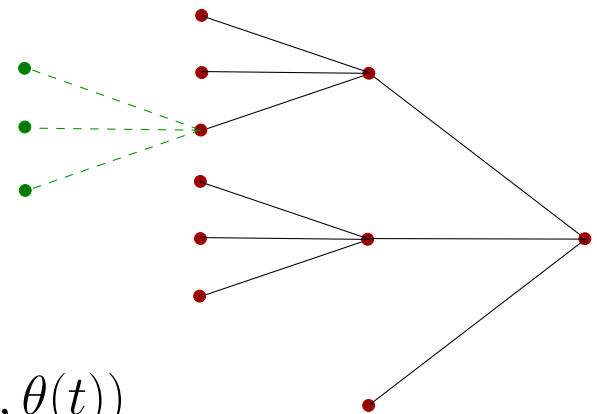
Obtain model and parameters sample $(T(t), \theta(t))$

△ [Entropy step] Obtain entropy sample $H(t)$:

↪ either compute $H(t) = H(T(t), \theta(t))$ directly

↪ or generate $Y_1^L \sim (T(t), \theta(t))$

and estimate $H(t) = -\frac{1}{L} \log P(Y_1^L | T(t), \theta(t))$



Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \dots, X_0, X_1, \dots, X_n$
Parameters m, D, β

Run BCT algorithm

Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)} \text{Unif}$

Iterate: At each t :

△ [Metropolis-Gibbs step]

Obtain model and parameters sample $(T(t), \theta(t))$

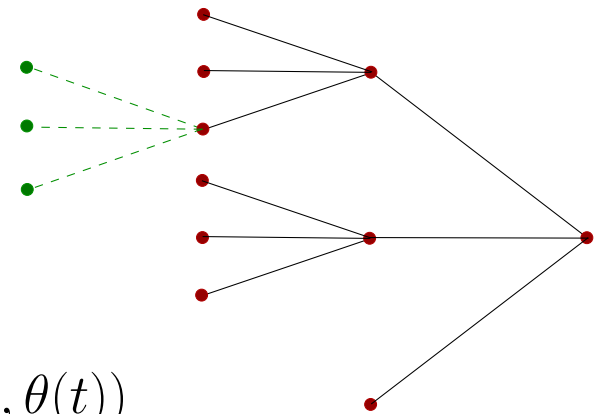
△ [Entropy step] Obtain entropy sample $H(t)$:

↪ either compute $H(t) = H(T(t), \theta(t))$ directly

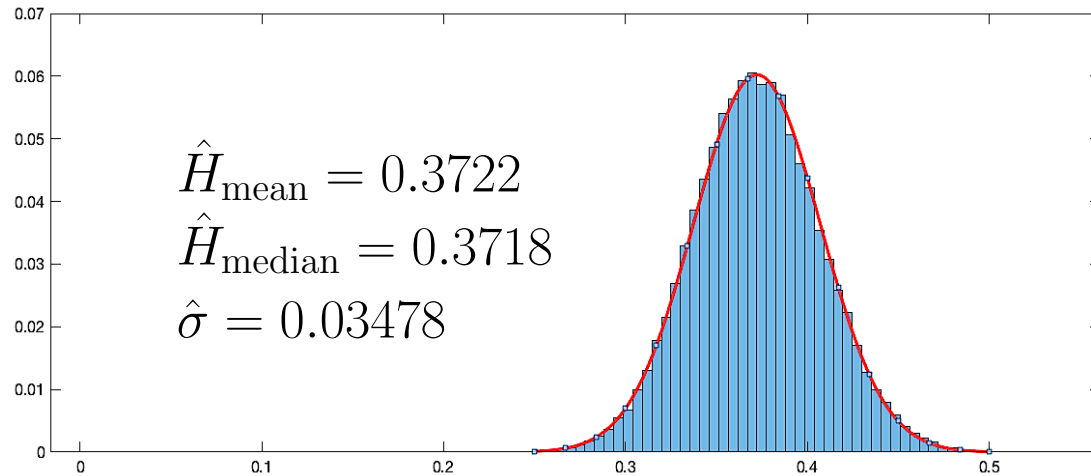
↪ or generate $Y_1^L \sim (T(t), \theta(t))$
and estimate $H(t) = -\frac{1}{L} \log P(Y_1^L | T(t), \theta(t))$

Estimate $\pi(H|X)$

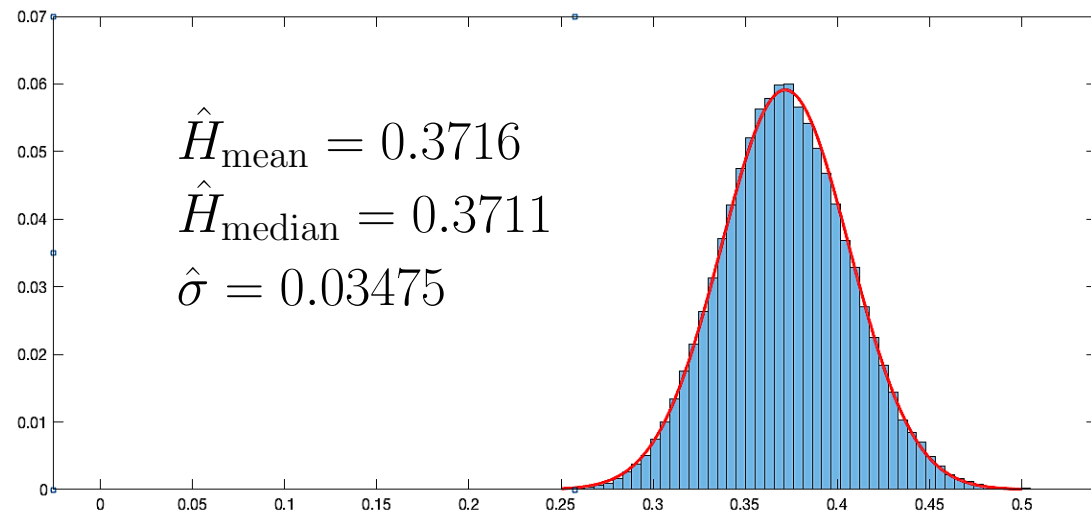
Compute posterior of H from $(H(1), H(2), \dots, H(N))$



Entropy of Peewee song



After 10^5 MCMC iterations



$\hat{H} = 0.372$ bits/sample
 $= 0.217$ bits/second

Prediction

Observe

The **posterior predictive distribution**

$$f(x_{n+1}|X_{-D+1}^n) = \sum_T \int_{\theta} \underbrace{f(x_{n+1}|X_{-D+1}^n, \theta, T)}_{\text{likelihood}} \underbrace{\pi(\theta, T|X_{-D+1}^n)}_{\text{posterior}} d\theta$$

Prediction

Observe

The **posterior predictive distribution**

$$\begin{aligned} f(x_{n+1}|X_{-D+1}^n) &= \sum_T \int_{\theta} \underbrace{f(x_{n+1}|X_{-D+1}^n, \theta, T)}_{\text{likelihood}} \underbrace{\pi(\theta, T|X_{-D+1}^n)}_{\text{posterior}} d\theta \\ &= \frac{f(x_{n+1}, X_1^n | X_{-D+1}^0)}{f(X_1^n | X_{-D+1}^0)} \\ &= \frac{\text{prior predictive likelihood up to } n+1}{\text{prior predictive likelihood up to } n} \end{aligned}$$

Prediction

Observe

The **posterior predictive distribution**

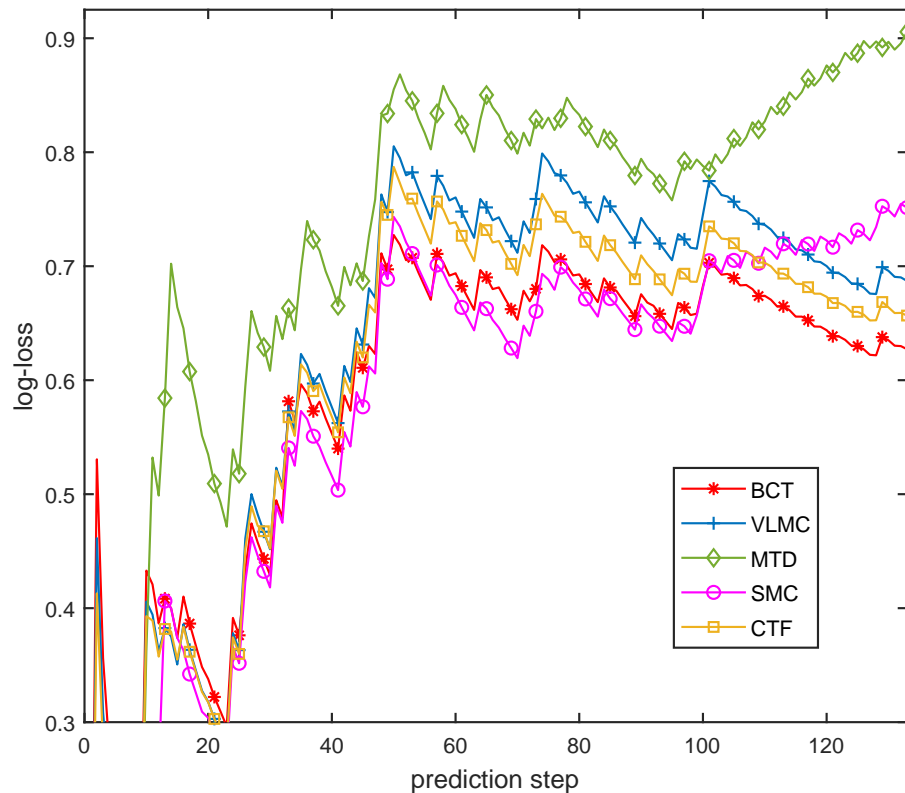
$$\begin{aligned} f(x_{n+1}|X_{-D+1}^n) &= \sum_T \int_{\theta} \underbrace{f(x_{n+1}|X_{-D+1}^n, \theta, T)}_{\text{likelihood}} \underbrace{\pi(\theta, T|X_{-D+1}^n)}_{\text{posterior}} d\theta \\ &= \frac{f(x_{n+1}, X_1^n | X_{-D+1}^0)}{f(X_1^n | X_{-D+1}^0)} \\ &= \frac{\text{prior predictive likelihood up to } n+1}{\text{prior predictive likelihood up to } n} \end{aligned}$$

- (i) can be computed *sequentially*, online
- (ii) converges to the true conditional distribution
- (iii) achieves the minimax optimal risk in terms of log-loss

Prediction results

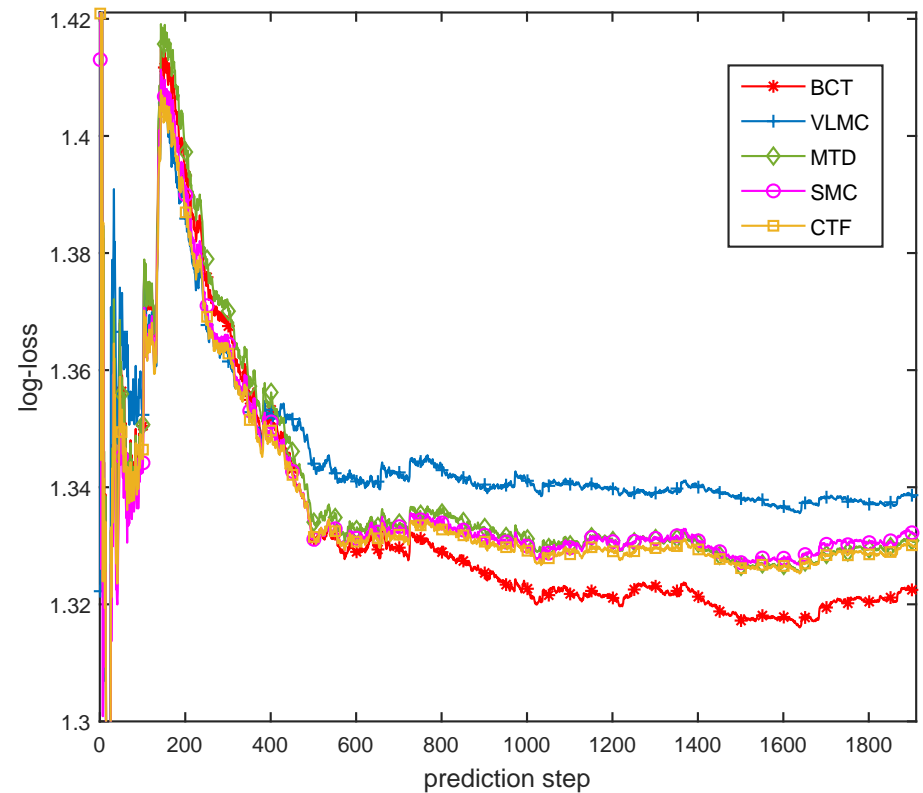
Pewee song data

training = 1185 (90%), testing = 132 (10%)



SARS-CoV-2 protein S gene

training = prediction = 1906 samples



Raftery's **MTD** (Markov Transition Distribution)

Dunson et al's **CTF** (Conditional Tensor Factorization)

Bühlmann et al's **VLMC** (Variable Memory Markov Chains)

Xiong et al's **SMC** (Sparse Markov Chains)

Theoretical justifications

“Theorem 1” [BIC/MDL connection]

For every data string X of arbitrary length n , any initial context X_{-D+1}^0 and any model T of depth no more than D with parameters θ the prior predictive likelihood $f(X) = f(X_1^n | X_{-D+1}^0)$ satisfies

$$\log f(X) \approx \log P(X|\theta, T) - \frac{|T|(m-1)}{2} \log n$$

and this is in a strong sense best possible

Theoretical justifications

“Theorem 1” [BIC/MDL connection]

For every data string X of arbitrary length n , any initial context X_{-D+1}^0 and any model T of depth no more than D with parameters θ the prior predictive likelihood $f(X) = f(X_1^n | X_{-D+1}^0)$ satisfies

$$\log f(X) \approx \log P(X|\theta, T) - \frac{|T|(m-1)}{2} \log n$$

and this is in a strong sense best possible

Theorem 2 The BCT predictive distribution

$$f(j|X_{-D+1}^n)$$

converges to the true underlying distribution as fast as possible:

(i) For data generated by any model T of depth no more than D with parameters θ , and for any $j \in A$:

$$f(j|X_{-D+1}^n) - P(j|X_{-D+1}^n, \theta, T) \rightarrow 0 \quad \text{with prob 1}$$

(ii) Theorem 1 \Rightarrow that it achieves the minimal log-loss

More theoretical justifications

Theorem 3 [Consistency]

For any ergodic VMMC $\{X_n\}$ of depth no more than D

$$\pi(\cdot, \cdot | X) \xrightarrow{\mathcal{D}} \delta_{(T^*, \theta^*)} \quad \text{with prob 1}$$

and

$$T_1^{(n)} = T^* \quad \text{eventually, with prob 1}$$

More theoretical justifications

Theorem 3 [Consistency]

For any ergodic VMMC $\{X_n\}$ of depth no more than D

$$\pi(\cdot, \cdot | X) \xrightarrow{\mathcal{D}} \delta_{(T^*, \theta^*)} \quad \text{with prob 1}$$

and

$$T_1^{(n)} = T^* \quad \text{eventually, with prob 1}$$

Theorem 4 [Asymptotic normality]

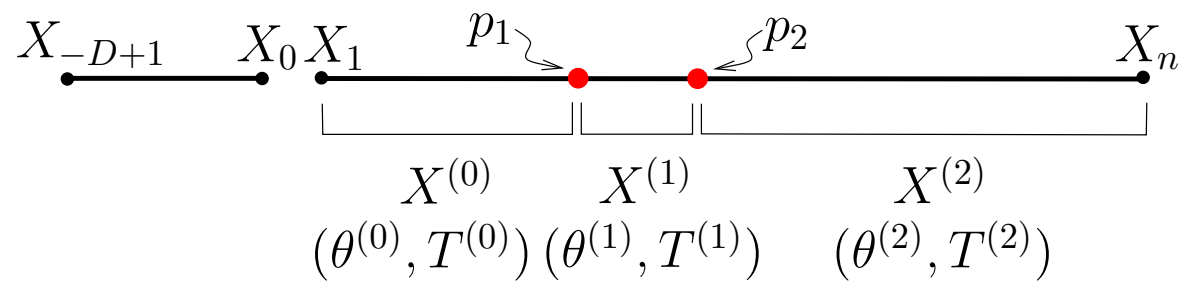
Let $\{X_n\}$ be an ergodic VMMC of depth $\leq D$, with stationary distr π

Suppose $\theta^{(n)} \sim \pi(\cdot | X_{-D+1}^n, T^*)$ and let $\bar{\theta}^{(n)}$ denote its mean. Then:

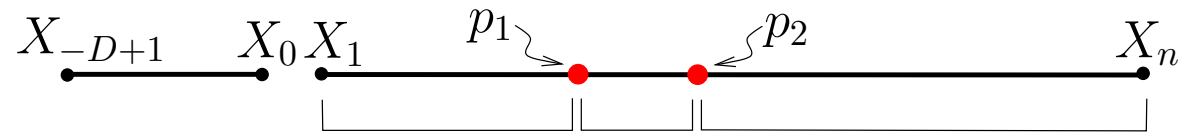
$$\sqrt{n} \left[\theta^{(n)} - \bar{\theta}^{(n)} \right] \xrightarrow{\mathcal{D}} N(0, J) \quad \text{with prob 1}$$

[Let Θ_s^* be the diagonal matrix with entries $\theta_s^*(j)$, $j \in A$, and let J_s denote the $m \times m$ matrix $J_s = \frac{1}{\pi(s)} [\Theta_s^* - (\theta_s^*)^t (\theta_s^*)]$. Then J is the $m|T^*| \times m|T^*|$ block-diagonal matrix consisting of all $m \times m$ blocks J_s]

Changepoint detection: A Bayesian setting



Changepoint detection: A Bayesian setting



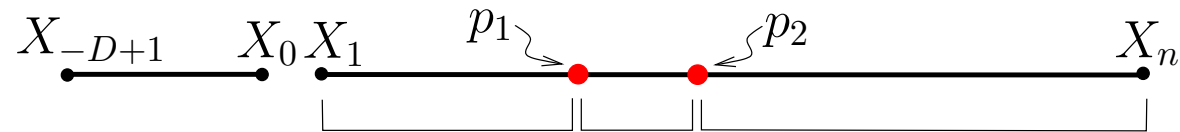
Number of changepoints

$$0 \leq \ell \leq \ell_{\max}$$

$$\text{Prior } \pi(\ell) \sim \text{Po}(\lambda) \mid \ell \leq \ell_{\max}$$

$$\begin{array}{ccc} X^{(0)} & X^{(1)} & X^{(2)} \\ (\theta^{(0)}, T^{(0)}) & (\theta^{(1)}, T^{(1)}) & (\theta^{(2)}, T^{(2)}) \end{array}$$

Changepoint detection: A Bayesian setting



Number of changepoints

$$0 \leq \ell \leq \ell_{\max}$$

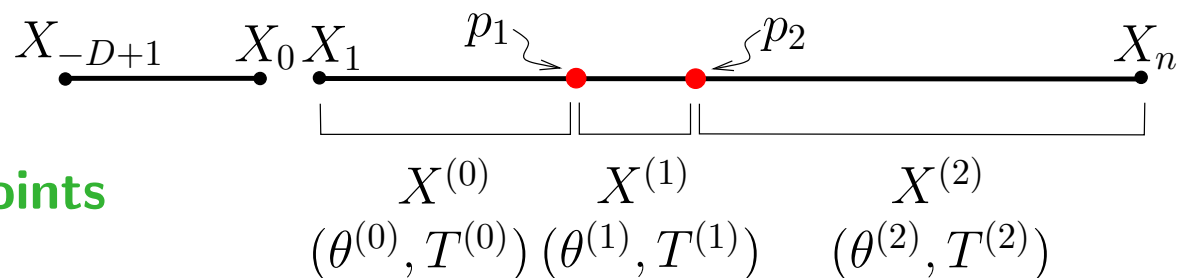
Prior $\pi(\ell) \sim \text{Po}(\lambda) \mid \ell \leq \ell_{\max}$

$$X^{(0)} \quad X^{(1)} \quad X^{(2)}$$

$$(\theta^{(0)}, T^{(0)}) \quad (\theta^{(1)}, T^{(1)}) \quad (\theta^{(2)}, T^{(2)})$$

Changepoint locations Given ℓ , the prior $\pi(p|\ell)$ of the locations $p = (p_1, \dots, p_\ell)$ is the distr of the even points in $2\ell + 1$ indep ordered draws from $\{1, 2, \dots, n\}$ without replacement

Changepoint detection: A Bayesian setting



Number of changepoints

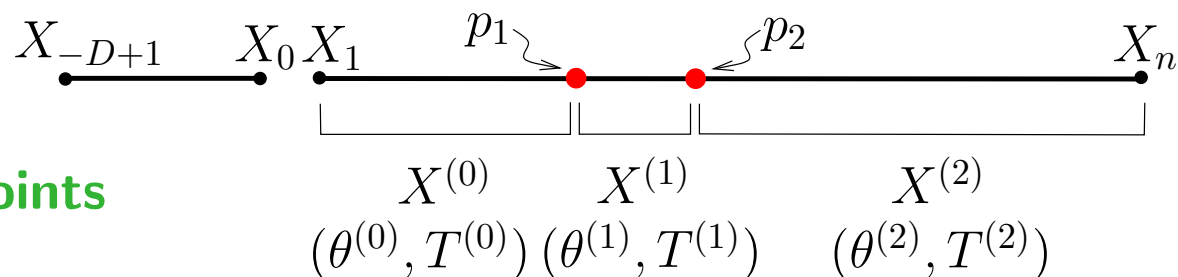
$$0 \leq \ell \leq \ell_{\max}$$

Prior $\pi(\ell) \sim \text{Po}(\lambda) \mid \ell \leq \ell_{\max}$

Changepoint locations Given ℓ , the prior $\pi(\mathbf{p} \mid \ell)$ of the locations $\mathbf{p} = (p_1, \dots, p_\ell)$ is the distr of the even points in $2\ell + 1$ indep ordered draws from $\{1, 2, \dots, n\}$ without replacement

Prior on models Given ℓ, \mathbf{p} , the $\ell + 1$ models $\{T^{(i)}\}$ are i.i.d. under $\pi(\{T^{(i)}\} \mid \ell, \mathbf{p})$ each with distr $\pi_D(T^{(i)})$ as before

Changepoint detection: A Bayesian setting



Number of changepoints

$$0 \leq \ell \leq \ell_{\max}$$

$$\text{Prior } \pi(\ell) \sim \text{Po}(\lambda) \mid \ell \leq \ell_{\max}$$

Changepoint locations Given ℓ , the prior $\pi(p|\ell)$ of the locations

$p = (p_1, \dots, p_\ell)$ is the distr of the even points in $2\ell + 1$ indep ordered draws from $\{1, 2, \dots, n\}$ without replacement

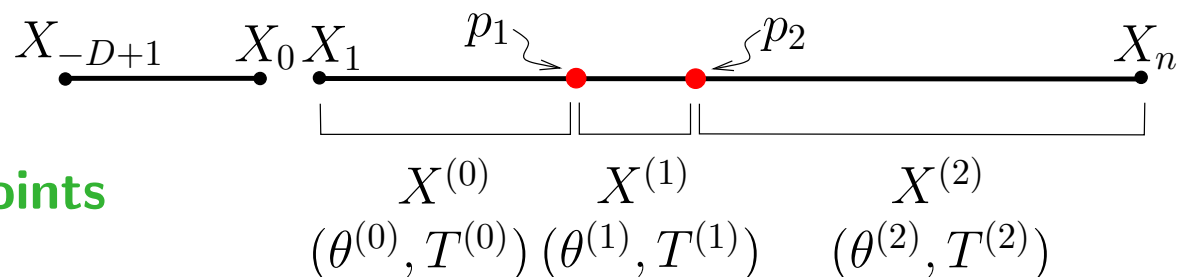
Prior on models Given ℓ, p , the $\ell + 1$ models $\{T^{(i)}\}$ are i.i.d.

under $\pi(\{T^{(i)}\}|\ell, p)$ each with distr $\pi_D(T^{(i)})$ as before

Prior on parameters Given $\ell, p, \{T^{(i)}\}$, the parameter vectors $\{\theta^{(i)}\}$ are i.i.d.

under $\pi(\{\theta^{(i)}\}|\ell, p, \{T^{(i)}\})$ each with a product Dirichlet distr as before

Changepoint detection: A Bayesian setting



Number of changepoints

$$0 \leq \ell \leq \ell_{\max}$$

Prior $\pi(\ell) \sim \text{Po}(\lambda) \mid \ell \leq \ell_{\max}$

Changepoint locations

Given ℓ , the prior $\pi(p|\ell)$ of the locations $p = (p_1, \dots, p_\ell)$ is the distr of the even points in $2\ell + 1$ indep ordered draws from $\{1, 2, \dots, n\}$ without replacement

Prior on models

Given ℓ, p , the $\ell + 1$ models $\{T^{(i)}\}$ are i.i.d. under $\pi(\{T^{(i)}\}|\ell, p)$ each with distr $\pi_D(T^{(i)})$ as before

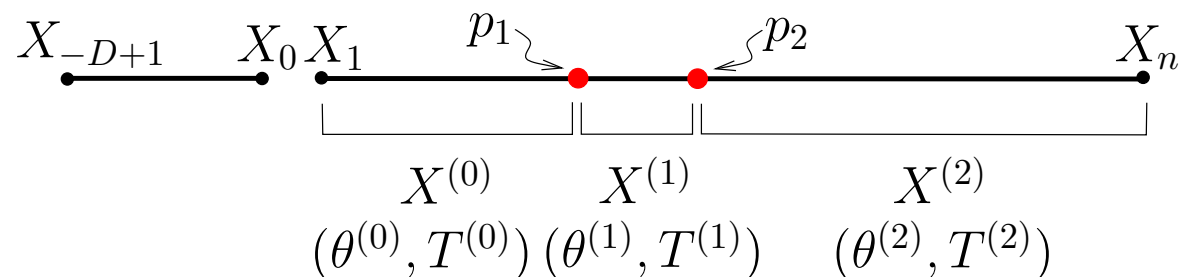
Prior on parameters

Given $\ell, p, \{T^{(i)}\}$, the parameter vectors $\{\theta^{(i)}\}$ are i.i.d. under $\pi(\{\theta^{(i)}\}|\ell, p, \{T^{(i)}\})$ each with a product Dirichlet distr as before

Likelihood $f(X_1^n | X_{-D+1}^0, \ell, p, \{\theta^{(i)}\}, \{T^{(i)}\}) = \prod_{0 \leq i \leq \ell} f(X^{(i)} | T^{(i)}, \theta^{(i)})$

where each term in the product is a VMCMC likelihood as before

BCT changepoint detection



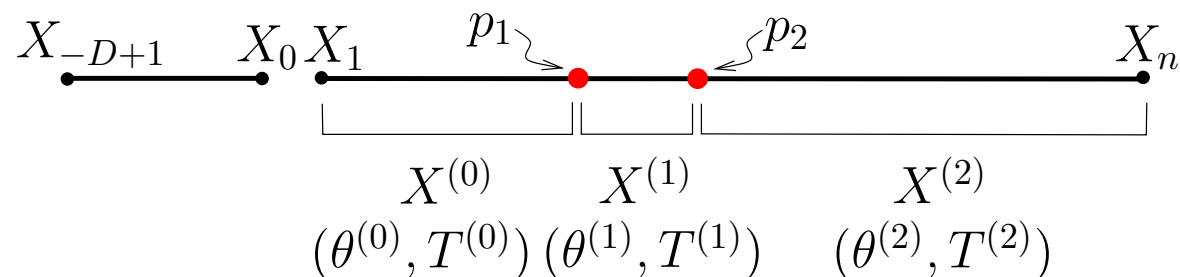
Goal From the likelihood

$$f(X_1^n | X_{-D+1}^0, \ell, p, \{\theta^{(i)}\}, \{T^{(i)}\})$$

and the priors, **determine the posterior $\pi(\ell, p | X)$**

Ordinarily MCMC would require sampling from all the parameters $(\ell, p, \{\theta^{(i)}\}, \{T^{(i)}\}) \rightsquigarrow$ a nearly impossible task

BCT changepoint detection



Goal From the likelihood

$$f(X_1^n | X_{-D+1}^0, \ell, p, \{\theta^{(i)}\}, \{T^{(i)}\})$$

and the priors, **determine the posterior $\pi(\ell, p | X)$**

Ordinarily MCMC would require sampling from all the parameters $(\ell, p, \{\theta^{(i)}\}, \{T^{(i)}\}) \rightsquigarrow$ a nearly impossible task

But here $\pi(\ell, p | X) \propto f(X | \ell, p) \pi(p | \ell) \pi(\ell) = \left[\prod_{0 \leq i \leq \ell} f(X^{(i)}) \right] \pi(p | \ell) \pi(\ell)$

which can be computed by $(\ell + 1)$ applications of the CTW!

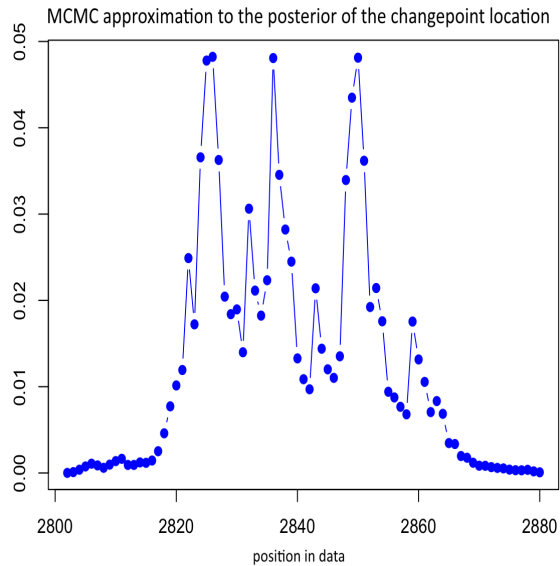
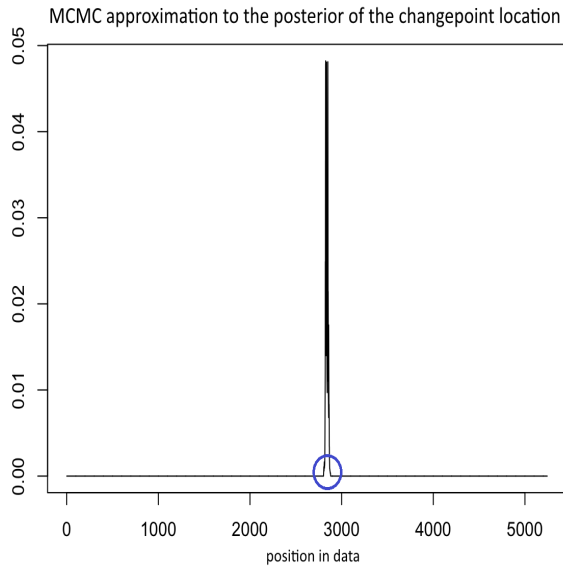
\rightsquigarrow **Effective MCMC** We can sample from the desired posterior $\pi(\ell, p | X)$ with a Metropolis-Hastings sampler that employs the CTW in each step

Changepoint detection results

Simian vacuolating virus 40:

DNA genome, $n = 5243$ bp

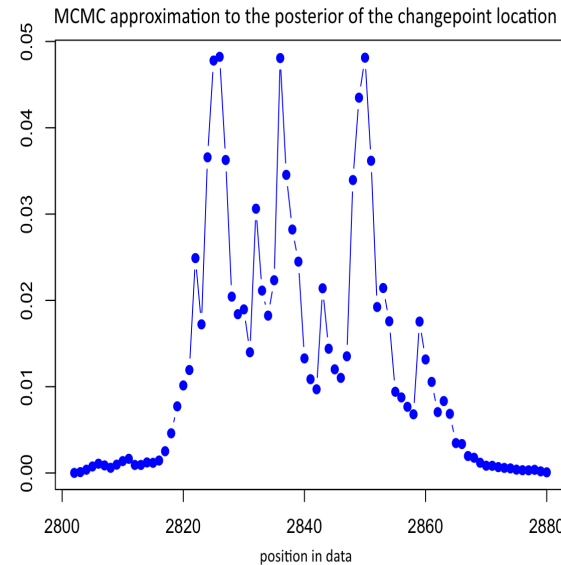
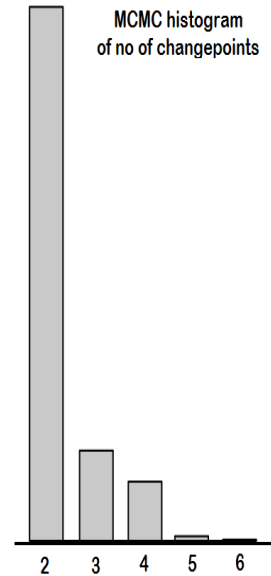
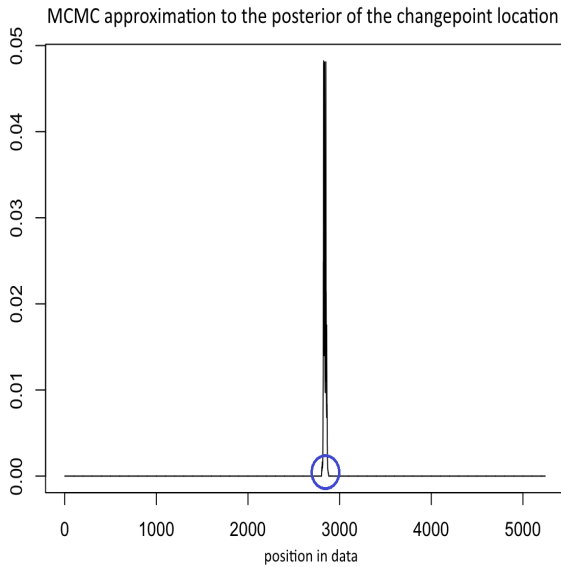
MCMC: $D = 5$, $\ell_{\max} = 1$



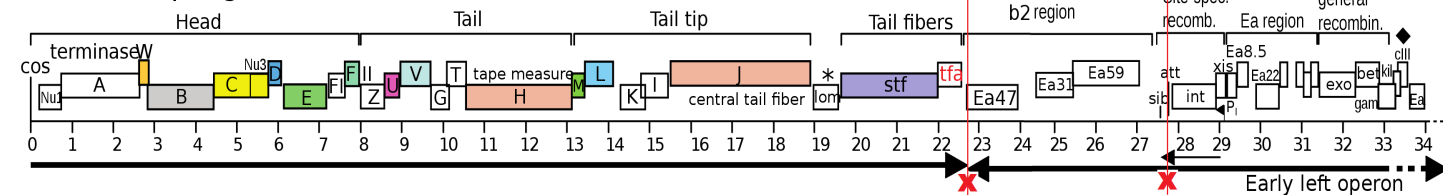
Changepoint detection results

Simian vacuolating virus 40:
 DNA genome, $n = 5243$ bp
 MCMC: $D = 5$, $\ell_{\max} = 1$

Enterophage λ bacterium: Examine $n = 35$ K bp
 of its DNA genome, with $D = 10$, $\ell_{\max} = 10$

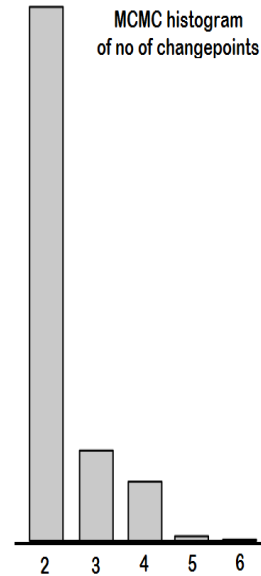
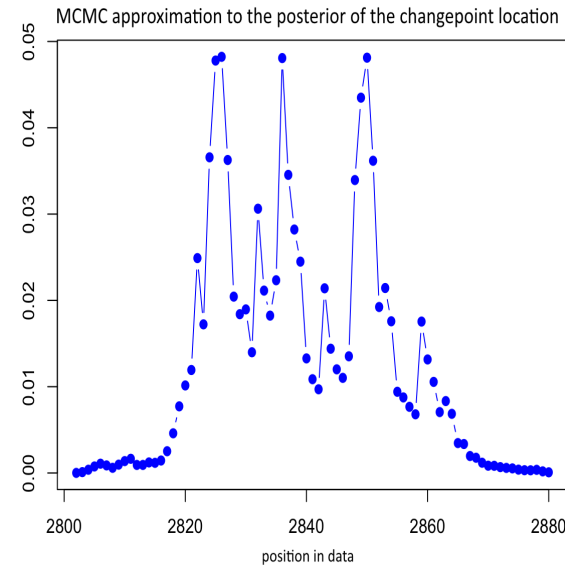
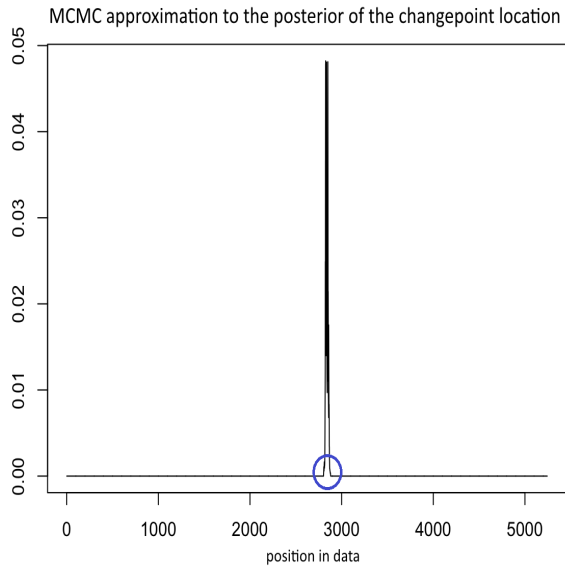


Genome of phage λ

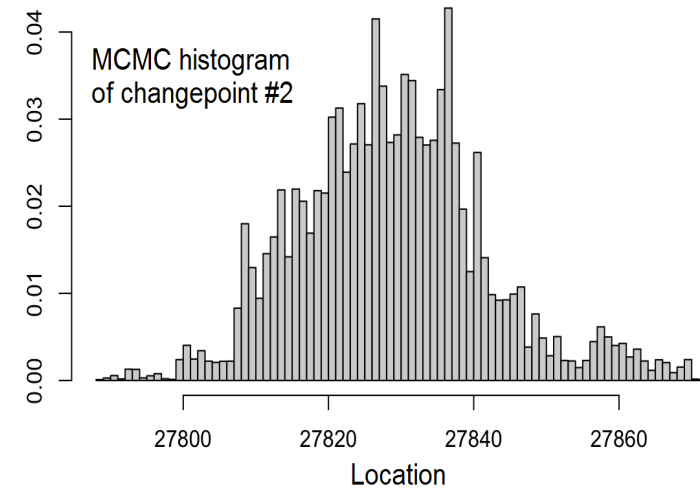
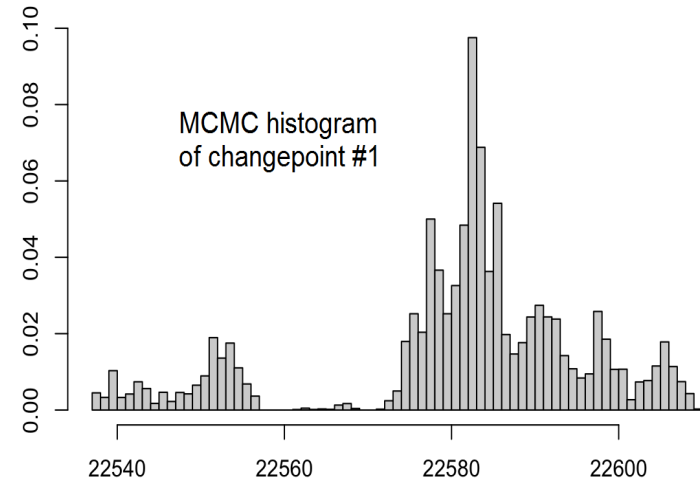


Changepoint detection results

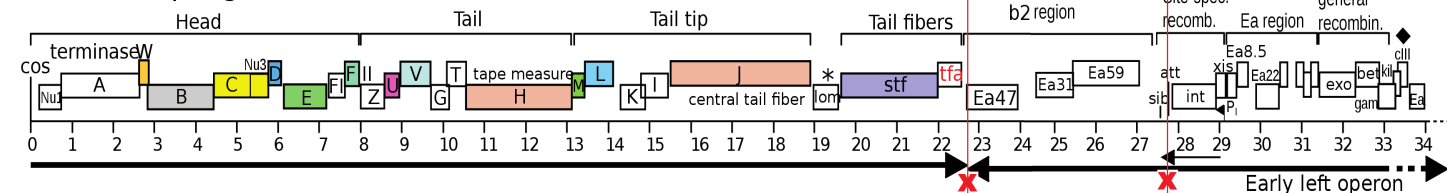
Simian vacuolating virus 40:
 DNA genome, $n = 5243$ bp
 MCMC: $D = 5$, $\ell_{\max} = 1$



Enterophage λ bacterium: Examine $n = 35$ K bp
 of its DNA genome, with $D = 10$, $\ell_{\max} = 10$



Genome of phage λ



Real-valued time series

Goal Perform effective Bayesian inference with real-valued time series

Real-valued time series

Goal Perform effective Bayesian inference with real-valued time series

First step: Define an appropriate model-class

Idea: Quantize and mix

- i. Start with an existing model class $\mathcal{M} = \{M\}$
e.g., $\text{AR}(p)$, $\text{MA}(p)$, $\text{ARMA}(p)$, $\text{ARIMA}(p)$, etc
- ii. Quantize the continuous observations X to a discrete time series Y on $A = \{0, 1, \dots, m - 1\}$ in a meaningful way
- iii. Select an m -ary context tree T of depth $\leq D$
- iv. Place a model M_s at each leaf $s \in T$
- v. Define the distribution of X_n given its past $(\dots, X_{n-2}, X_{n-1})$ as the distribution dictated by (M_s, ϕ_s) , where $s \in T$ is the context corresponding to Y_{n-D}, \dots, Y_{n-1}

Real-valued time series

Goal Perform effective Bayesian inference with real-valued time series

First step: Define an appropriate model-class

Idea: Quantize and mix

- i. Start with an existing model class $\mathcal{M} = \{M\}$
e.g., $AR(p)$, $MA(p)$, $ARMA(p)$, $ARIMA(p)$, etc
- ii. Quantize the continuous observations X to a discrete time series Y on $A = \{0, 1, \dots, m - 1\}$ in a meaningful way
- iii. Select an m -ary context tree T of depth $\leq D$
- iv. Place a model M_s at each leaf $s \in T$
- v. Define the distribution of X_n given its past $(\dots, X_{n-2}, X_{n-1})$ as the distribution dictated by (M_s, ϕ_s) , where $s \in T$ is the context corresponding to Y_{n-D}, \dots, Y_{n-1}

Main step: Inference

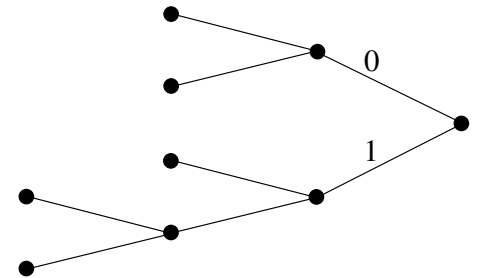
Idea: Generalize the CTW/BCT ideas, tools and algorithms

- i. Perform inference jointly on (M, ϕ) and on (T, θ)
- ii. Develop new methodological tools in applications

Example: The binary BCT-AR model class

Model class

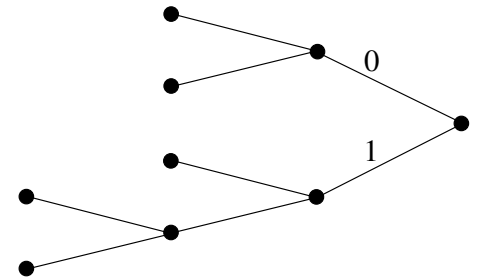
▷ Binary context trees T of depth $\leq D$



Example: The binary BCT-AR model class

Model class

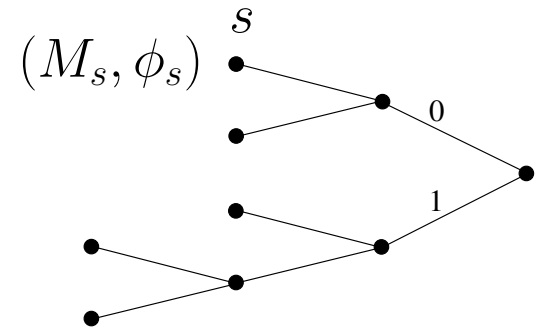
- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \rightarrow \{0, 1\}$



Example: The binary BCT-AR model class

Model class

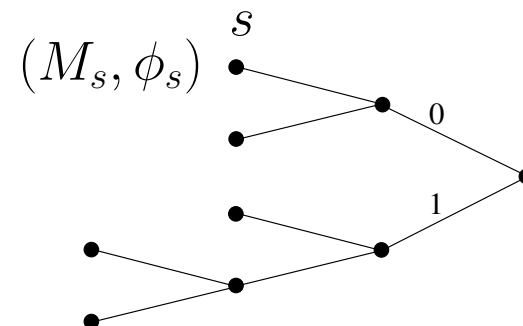
- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \rightarrow \{0, 1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$



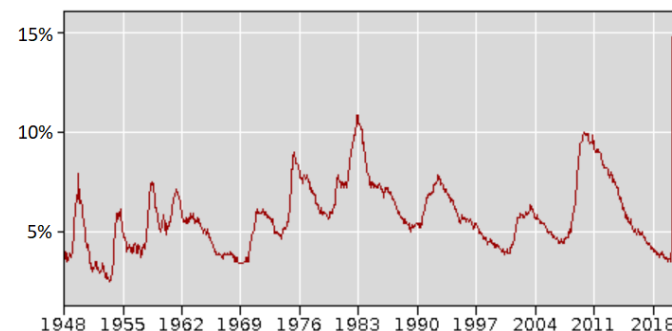
Example: The binary BCT-AR model class

Model class

- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \rightarrow \{0, 1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$



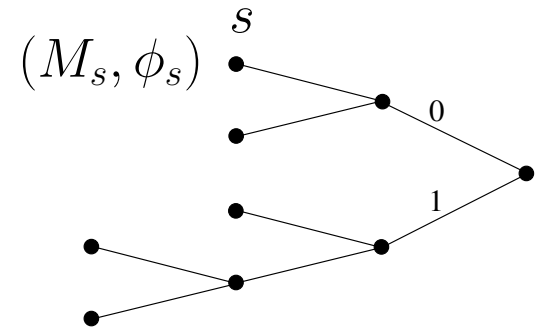
Application US unemployment data
Sequence X of differences of quarterly rates
1948-2019: $n = 288$ observations



Example: The binary BCT-AR model class

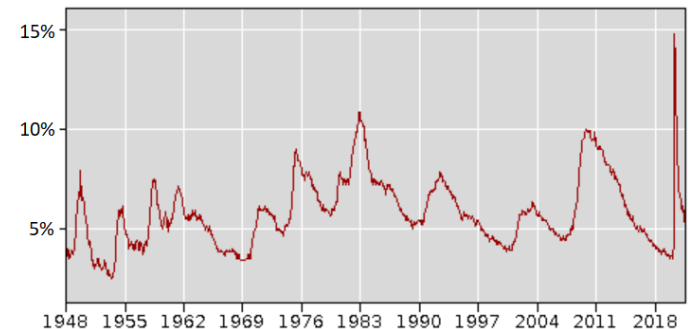
Model class

- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \rightarrow \{0, 1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$



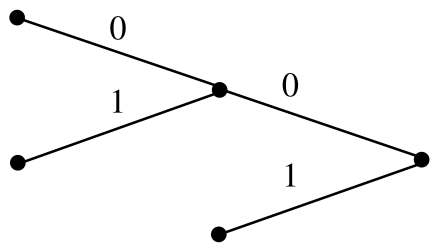
Application US unemployment data

Sequence X of differences of quarterly rates
1948-2019: $n = 288$ observations



Results With AR $p_{\max} = 5$, $D = 10$

and appropriate priors on p and on parameters:
Estimated quantizer threshold is $\approx 0.15\%$



Mean squared error (MSE) of forecasts

Model	Prediction step				
	1	2	3	4	5
Seas. ARIMA	5.40	7.71	10.1	11.6	11.0
SETAR	5.42	8.34	8.82	9.48	9.95
MAR	5.33	7.61	8.92	9.56	9.71
BCT-AR	4.90	7.33	8.44	9.08	9.48

Extensions and further results

~> Applications

Model selection

Segmentation

Filtering

Causality testing

Estimation

Anomaly detection

Prediction

Compression

Change-point detection

Markov order estimation

Entropy estimation

Content recognition

+ **Continuous** time series

~> Results on real data

▷ Satellite imaging ▷ genetics ▷ neuroscience ▷ finance

▷ meteorology (wind and rainfall prediction),

▷ animal communication (whale/dolphin/bird song data)

+ **R** package: **BCT**

~> Theoretical foundation

▶ Rich collection of asymptotics and non-asymptotic bounds