Learning from Time Series Data
Using Information-Theoretic Methods

Archimedes Workshop on the Foundations of Modern AI
NTUA, Athens
July 2024

Ioannis Kontoyiannis
U of Cambridge

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

Outline

Background: **Variable-memory Markov chains**

Outline

Background: Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Outline

Background: Variable-memory Markov chains

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms

CTW, BCT, k-BCT, MCMC

Outline

Background: **Variable-memory Markov chains**

Bayesian Context Trees

Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms

CTW, BCT, k-BCT, MCMC

Applications

Model selection	**Estimation**	Content recognition
Segmentation	Anomaly detection	**Markov order estimation**
Filtering	**Prediction**	**Entropy estimation**
Causality testing	Compression	**Change-point detection**

Outline

Background: Variable-memory Markov chains

Bayesian Context Trees
Prior structure, marginal likelihood, the posterior

Exact Inference Algorithms
CTW, BCT, k-BCT, MCMC

Applications

Model selection	**Estimation**	Content recognition
Segmentation	Anomaly detection	**Markov order estimation**
Filtering	**Prediction**	**Entropy estimation**
Causality testing	Compression	**Change-point detection**

$\rightsquigarrow$ **BCT-X**: Hierachical models and exact inference for **continuous** time series

Some history

⇝ **1983-86**: Tree sources introduced by **Risannen** along with the `context` algorithm for model selection

Some history

⇝ **1983-86**: Tree sources introduced by **Risannen**
along with the `context` algorithm for model selection

⇝ **1995**: The Context-Tree Weighting algorithm (CTW)
is introduced by **Willems et al** and used for data compression

Some history

⇝ **1983-86**: Tree sources introduced by **Risannen**
along with the `context` algorithm for model selection

⇝ **1995**: The Context-Tree Weighting algorithm (CTW)
is introduced by **Willems et al** and used for data compression

⇝ **1996-2006**: Numerous papers explore the CTW in a Bayesian setting
introducing the Context Tree Maximising (CTM) algorithm
and examining statistical applications [**Willems et al**]

Some history

⇝ **1983-86**: Tree sources introduced by **Risannen**
along with the `context` algorithm for model selection

⇝ **1995**: The Context-Tree Weighting algorithm (CTW)
is introduced by **Willems et al** and used for data compression

⇝ **1996-2006**: Numerous papers explore the CTW in a Bayesian setting
introducing the Context Tree Maximising (CTM) algorithm
and examining statistical applications [**Willems et al**]

⇝ **1999**: **Bühlmann et al** use tree sources and `context`
for model selection in a frequentist/classical setting

Some history

$\rightsquigarrow$ **1983-86**: Tree sources introduced by **Risannen**
along with the `context` algorithm for model selection

$\rightsquigarrow$ **1995**: The Context-Tree Weighting algorithm (CTW)
is introduced by **Willems et al** and used for data compression

$\rightsquigarrow$ **1996-2006**: Numerous papers explore the CTW in a Bayesian setting
introducing the Context Tree Maximising (CTM) algorithm
and examining statistical applications [**Willems et al**]

$\rightsquigarrow$ **1999**: **Bühlmann et al** use tree sources and `context`
for model selection in a frequentist/classical setting

$\rightsquigarrow$ **Today's talk**:
Presents **BCT-X**: a principled, unified Bayesian framework
for general inference and learning tasks on time series,
centered around CTW and generalizations of tree-source models

Motivation

$\rightsquigarrow$ **Discrete time series are often hard**

Inference	Signal processing
Machine learning	Communications

Motivation

$\rightsquigarrow$ **Discrete time series are often hard**

Inference | Signal processing
Machine learning | Communications

$\rightsquigarrow$ **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

$\rightsquigarrow$ **Need astronomical amounts of data**

$\rightsquigarrow$ **Need smarter, parsimonious models**

$\rightsquigarrow$ **Variable-memory Markov chains**

Variable-memory Markov chain models

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

Variable-memory Markov chain models

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

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

Variable-memory Markov chain models

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

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

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

Variable-memory Markov chain models

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

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

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

Problem $\boldsymbol{m^d}$ grows very fast!

Variable-memory Markov chain models

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

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

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

Problem $\boldsymbol{m^d}$ grows very fast!

Idea Use *variable length contexts* described by a **context tree** $\boldsymbol{T}$

Variable-memory Markov chain models

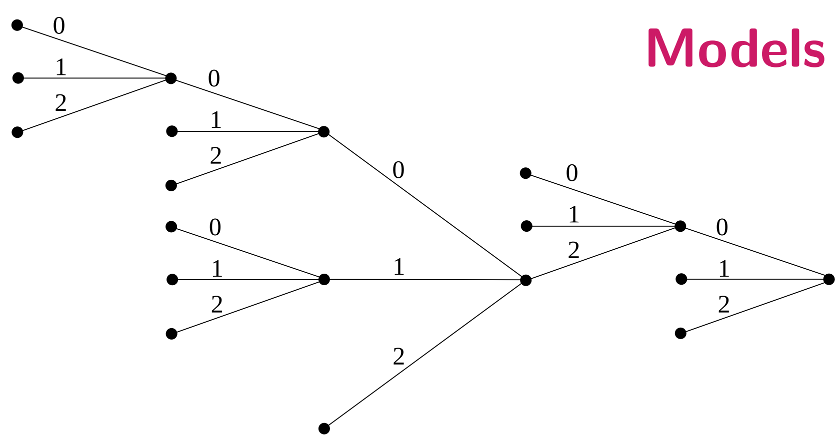
Markov chain $\{\ldots, X_0, X_1, \ldots\}$ with **alphabet A** $= \{0, 1, \ldots, m-1\}$ of size m

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

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

Problem m^d grows very fast!

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



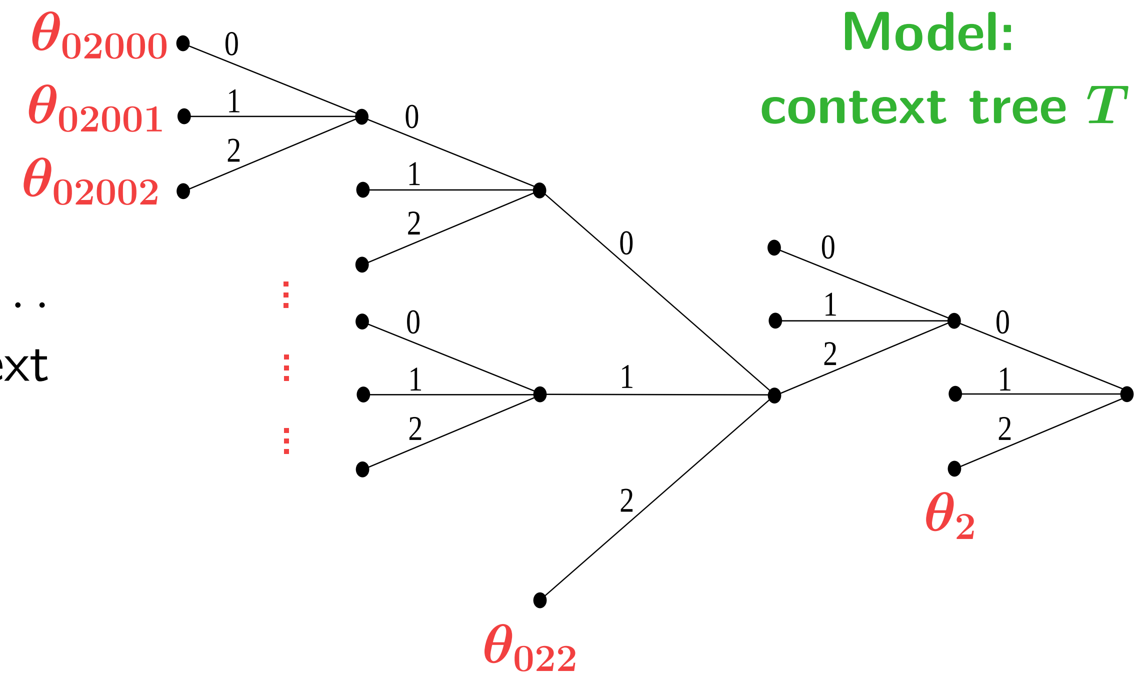
Models $\equiv$ Trees

Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols
Memory length $d = 5$

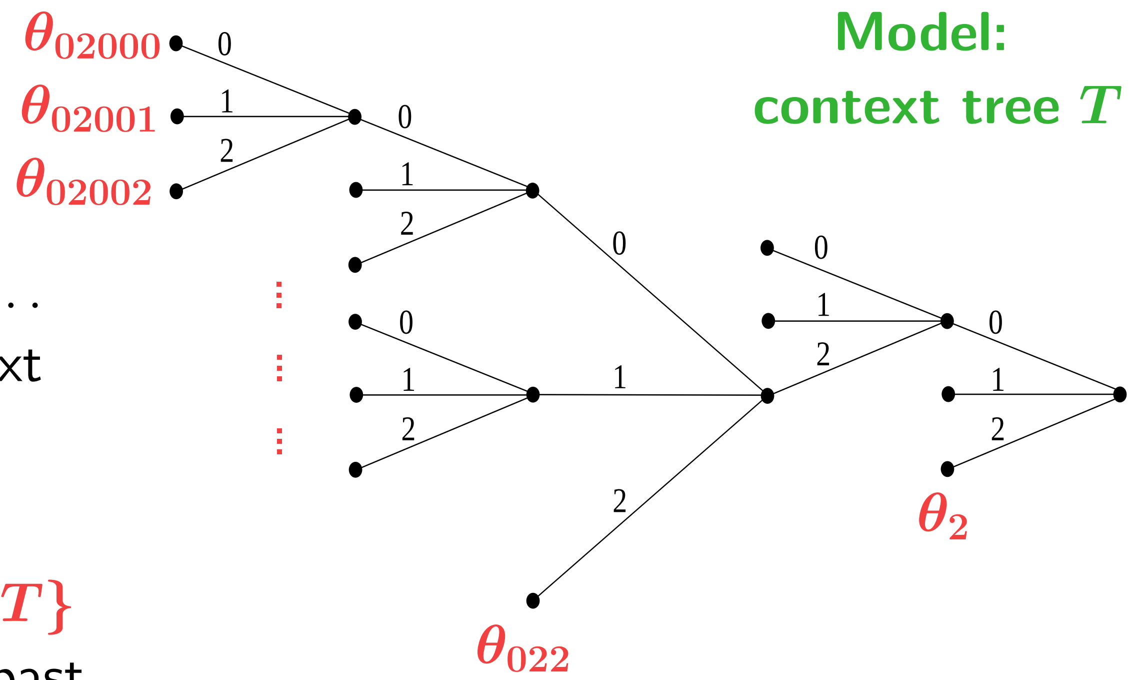
Each past string $X_{n-1}, X_{n-2}, \ldots$ corresponds to a unique context on a leaf of the tree

Model: context tree T



Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols
Memory length $d = 5$



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

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

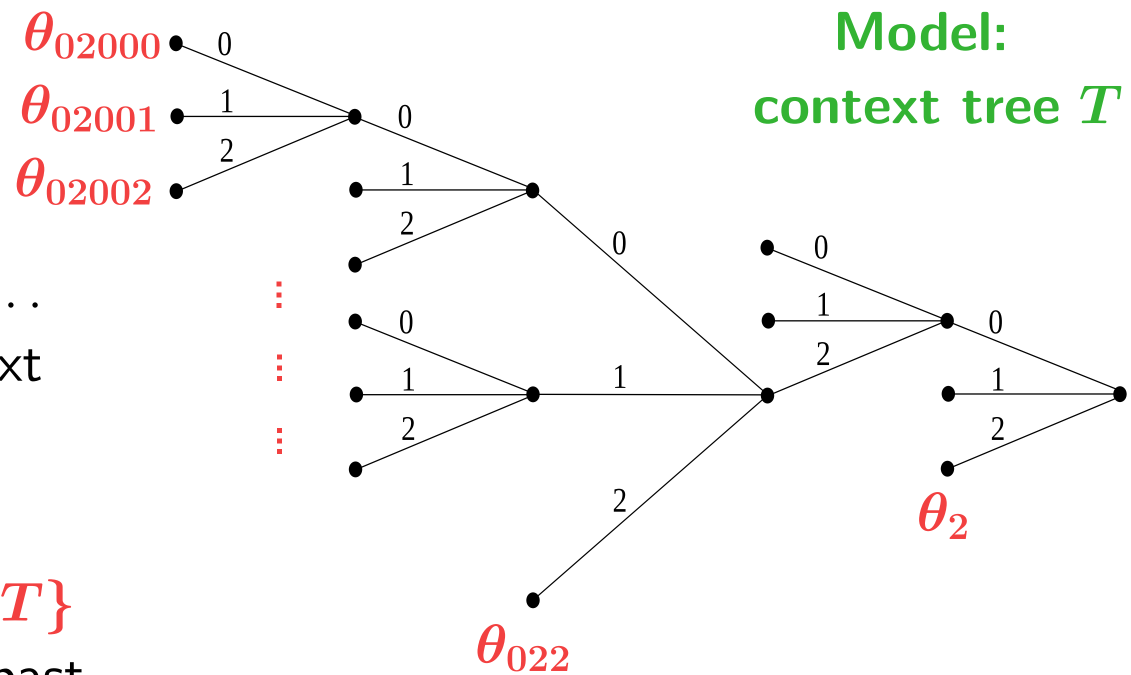
The distr of X_n given the past
is given by the distr on that leaf

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

Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols
Memory length $d = 5$

Model: context tree T



Each past string $X_{n-1}, X_{n-2}, \ldots$
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. $P(X_n = 1 | X_{n-1} = 0, X_{n-2} = 2, X_{n-2} = 2, X_{n-3} = 1, \ldots) = \theta_{022}(1)$

$\rightsquigarrow$ **Parsimony**

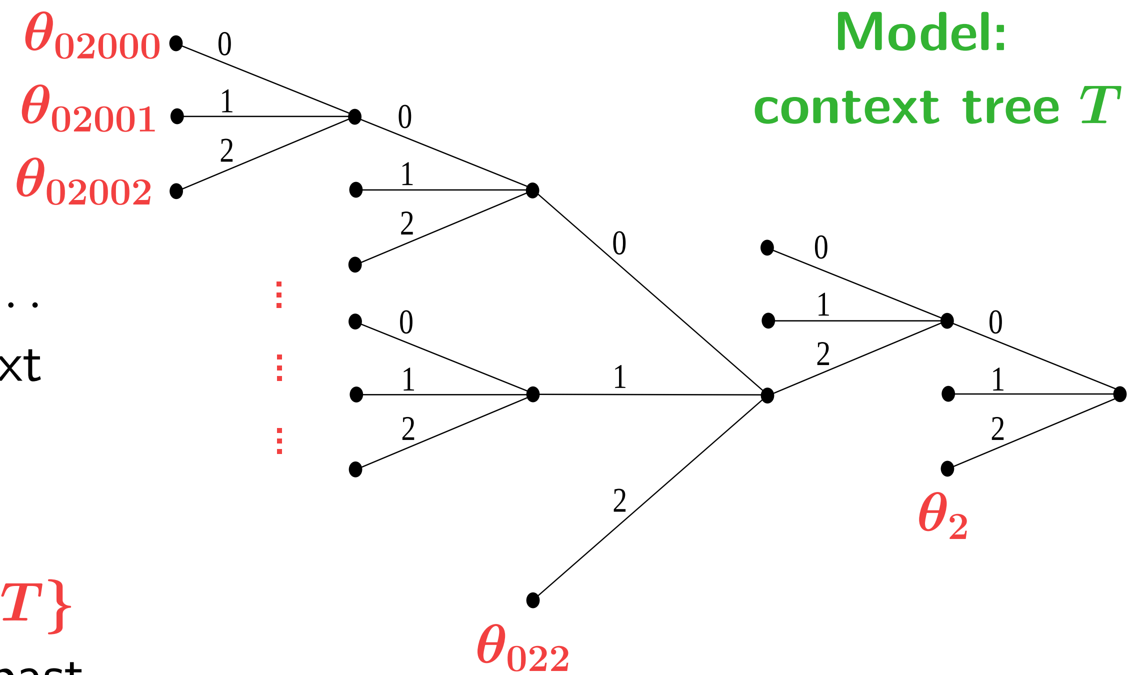
Instead of $3^5 = 243$ conditional distributions only need 13

$\Rightarrow$ potentially huge savings in general

Variable-memory Markov chains: An example

Alphabet $m = 3$ symbols
Memory length $d = 5$

Model: context tree T



Each past string $X_{n-1}, X_{n-2}, \ldots$
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. $P(X_n = 1 | X_{n-1} = 0, X_{n-2} = 2, X_{n-2} = 2, X_{n-3} = 1, \ldots) = \theta_{022}(1)$

$\rightsquigarrow$ **Parsimony**

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

$\Rightarrow$ potentially huge savings in general

$\rightsquigarrow$ 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 $\pi(\theta_s|T) \sim$ 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 $\beta \in (0,1)$:

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

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 $\boldsymbol{\pi(\theta_s|T)} \sim$ Dirichlet$(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$

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

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

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 $\pi(\theta_s|T) \sim$ Dirichlet$(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$

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

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

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

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

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

Bayesian inference for VMMCs

Given. Data $X = X_{-D+1}, \ldots X_0, X_1, \ldots, X_n$
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**:

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

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

Main obstacle

Determination of the **prior predictive likelihood**:

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

$\rightsquigarrow$ the number of models in the sum grows *doubly exponentially* in D

Computation of the marginal likelihood

Given the structure of the model, it is not surprising
that 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 $\boldsymbol{P_e(a_s)} = \frac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$

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

Computation of the marginal likelihood

Given the structure of the model, it is not surprising
that 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 $\boldsymbol{P_e(a_s)} = \frac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j) - 1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$

with the count vectors $a_s = (a_s(0), \ldots, a_s(m-1))$ 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** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$
Alphabet size m **Maximum depth** D
Prior parameter β

△ **1.** [*Tree.*] Construct a tree with nodes corresponding to all contexts of length $1, 2, \ldots, D$ contained in X

△ **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

△ **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

Bayesian Context Tree algorithm (BCT)

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

Given. **Data** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$
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 $X_{-D+1},\ldots,X_0,X_1,X_2,\ldots,X_n$

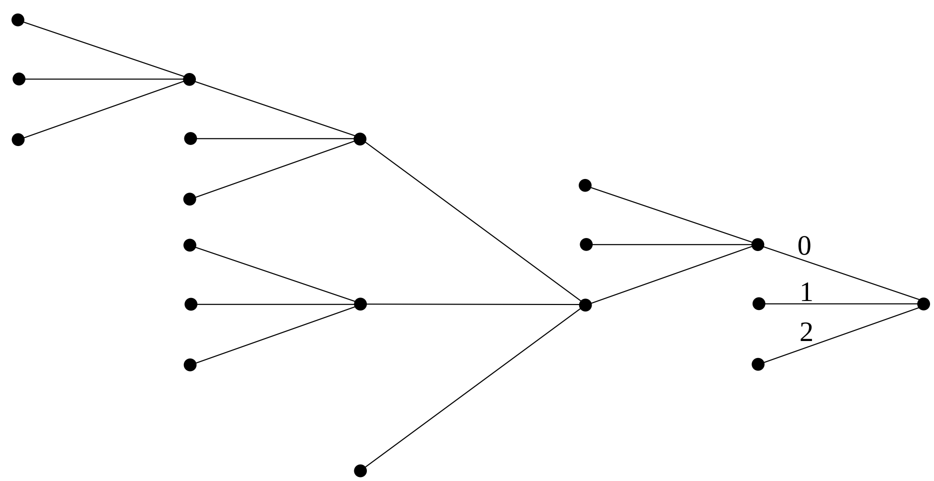
Alphabet size $m=3$

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 ith best subtree starting at s

Either entire line consists of $*$ meaning "prune at s"

Or jth 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 ith 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 $T_1^*, T_2^*, \ldots, T_k^*$ 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* $$\frac{\pi(T|X)}{\pi(T'|X)} = \frac{\pi(T)}{\pi(T')}\frac{\prod_{s\in T, s\notin T'} P_e(a_s)}{\prod_{s\in T', s\notin T} P_e(a_s)}.$$

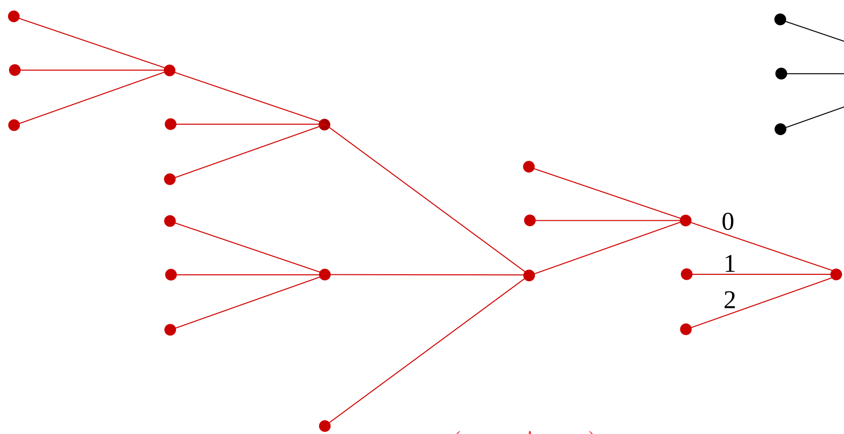
for ANY pair of models T, T'

(iii) *Full conditional density of θ*

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

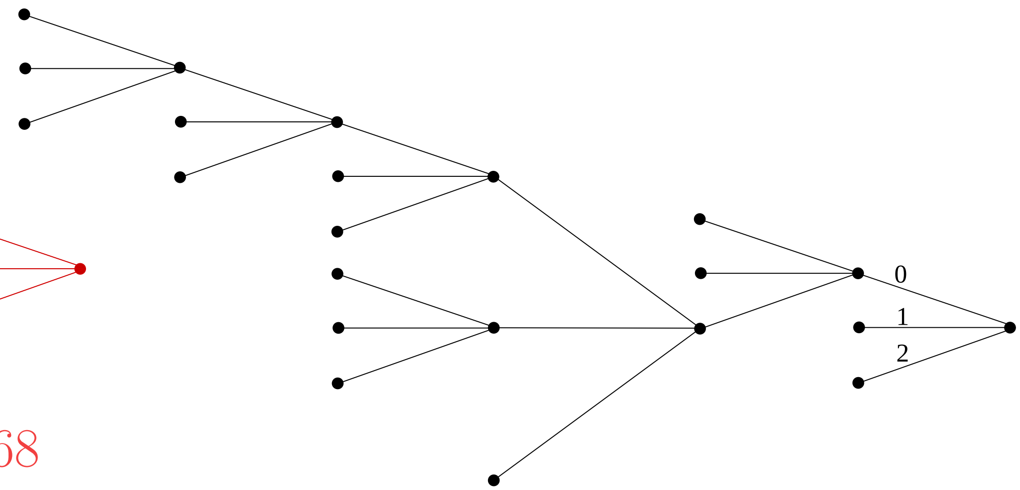
k-BCT models for the same 5th order chain

$D = 10$ ⇝ more than 10^{5900} models $\quad$ $n = 10000$, $k = 3$, $\beta = 3/4$

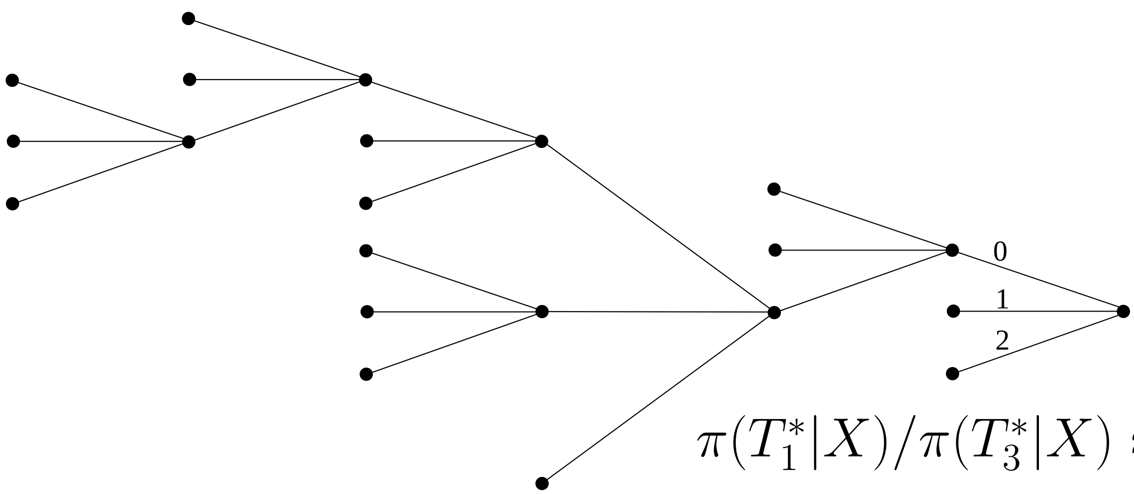


$\pi(T_1^*|X) \approx 0.368$

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



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



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

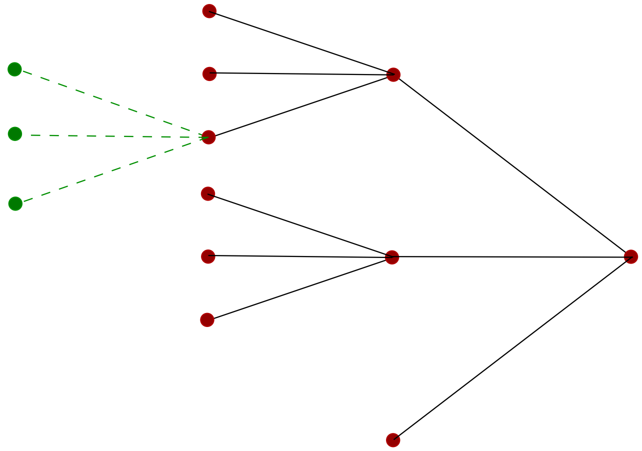
MCMC exploration of the posterior

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

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

MCMC exploration of the posterior

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



Run BCT algorithm

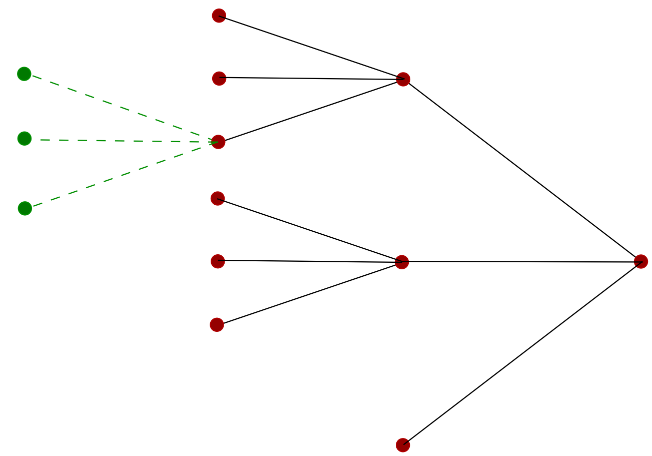
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif

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



Run BCT algorithm
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif
Iterate: At each t:

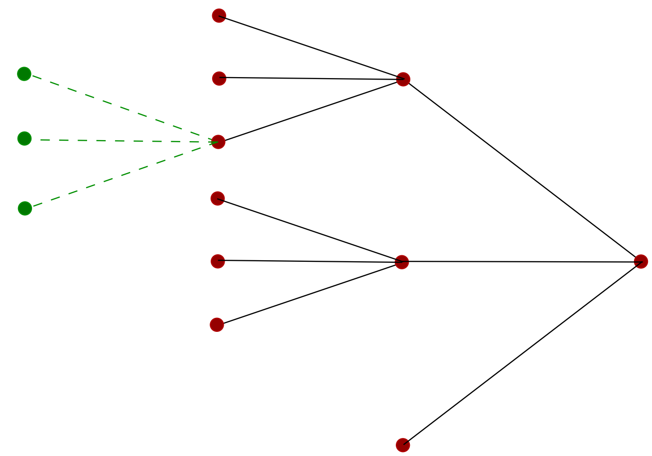
△ [*Metropolis proposal*] Given $T(t)$ propose T' by randomly adding or removing m sibling leaves

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

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

MCMC exploration of the posterior

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



Run BCT algorithm
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ 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'

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

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

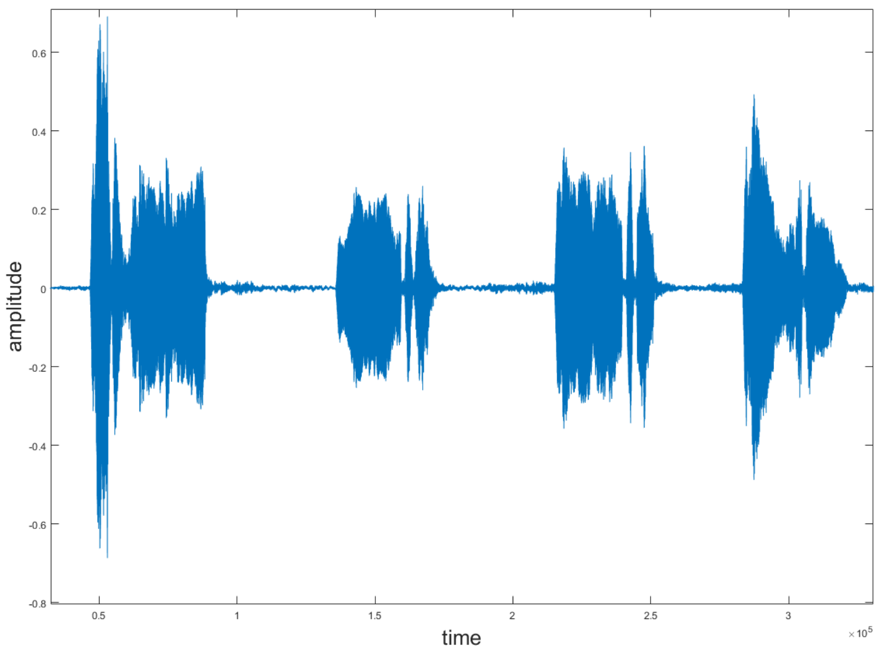
$$\prod_{s \in T(t+1)} \text{Dirichlet}(a_s(0) + 1/2, a_s(1) + 1/2, \ldots, a_s(m-1) + 1/2)$$

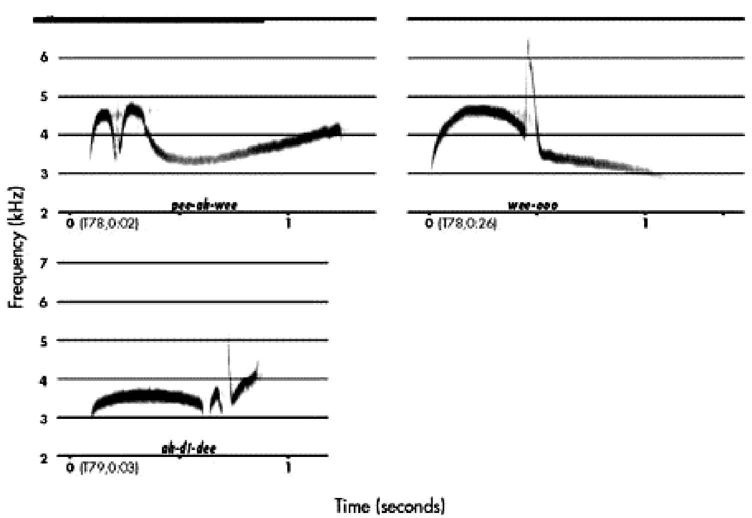
A fun data set: Wood Peewee bird song

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: MAP model

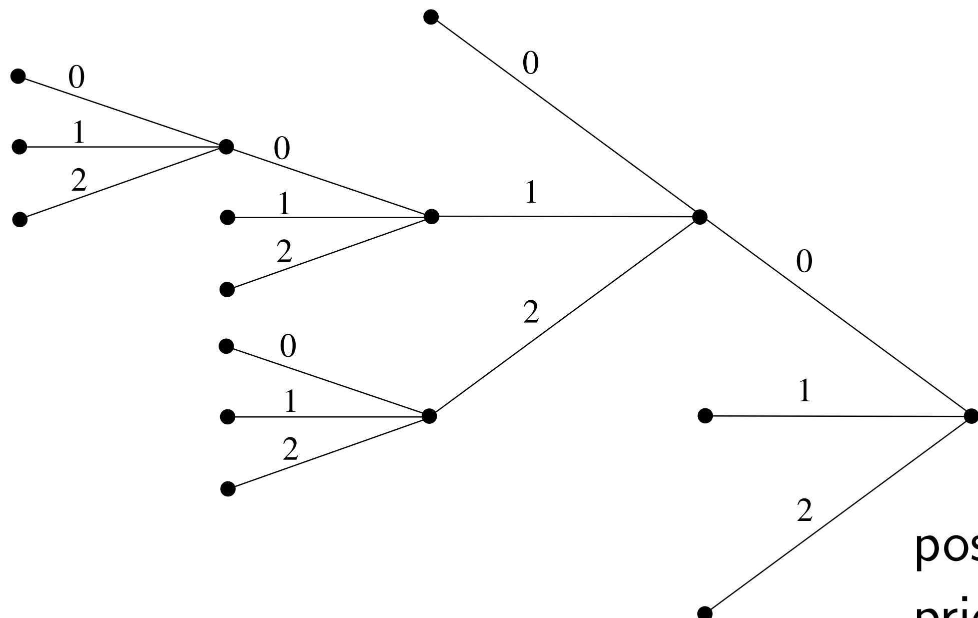
Data Recorded bird song data, transcribed as a sequence of (mono-)phthongs
Goal: Understand structure, complexity, variation and function

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

[Berchtold-Raftery (2002) "The MTD model"]

***k*-BCT** With $n = 1327$ samples

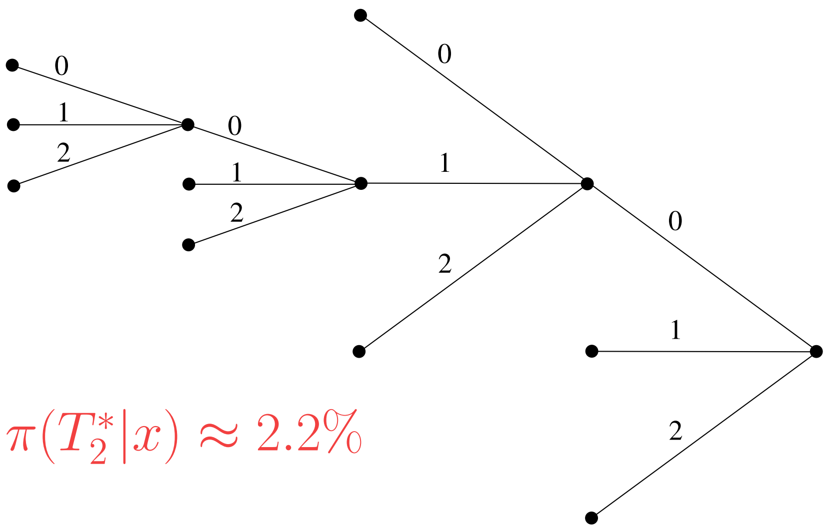
$m = 3$, $\beta = 3/4$, $k = 5$ and depth $D = 20$



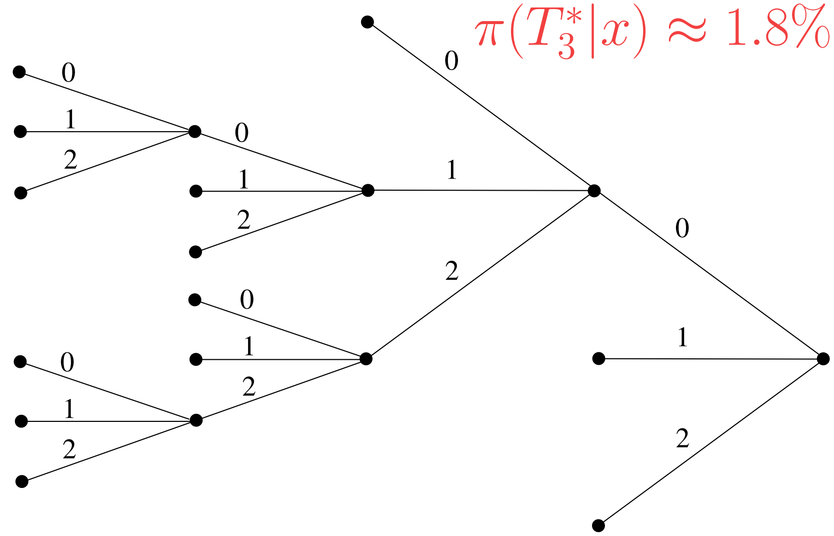
posterior: $\pi(T_1^*|x) \approx 12.6\%$

prior: $\pi(T_1^*) \approx 4 \times 10^{-5}$

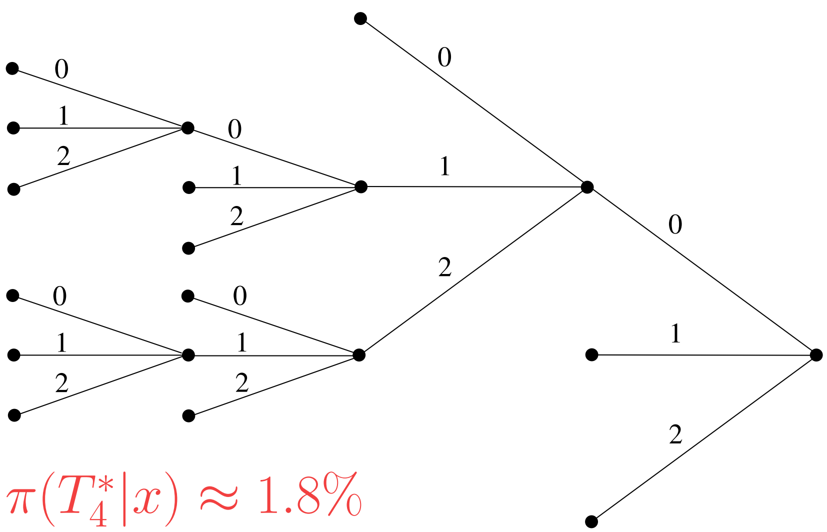
Wood Peewee bird song: Next 4 models



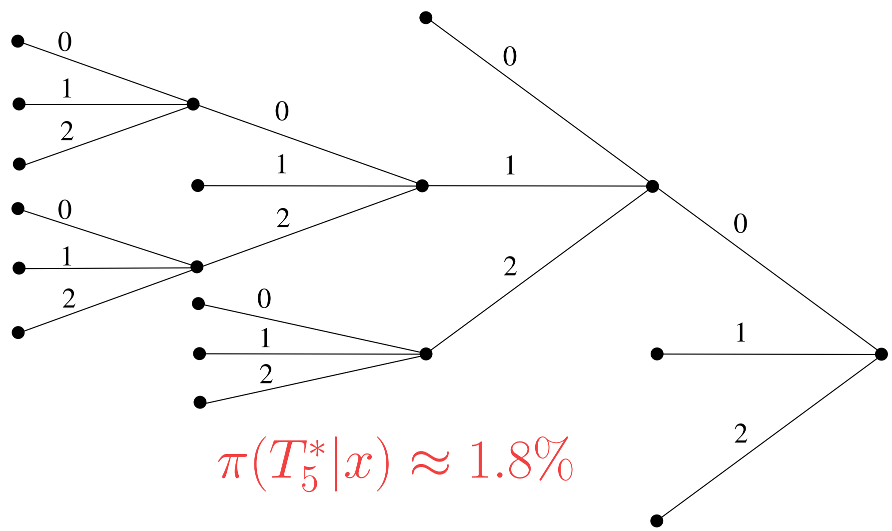
$\pi(T_2^*|x) \approx 2.2\%$



$\pi(T_3^*|x) \approx 1.8\%$



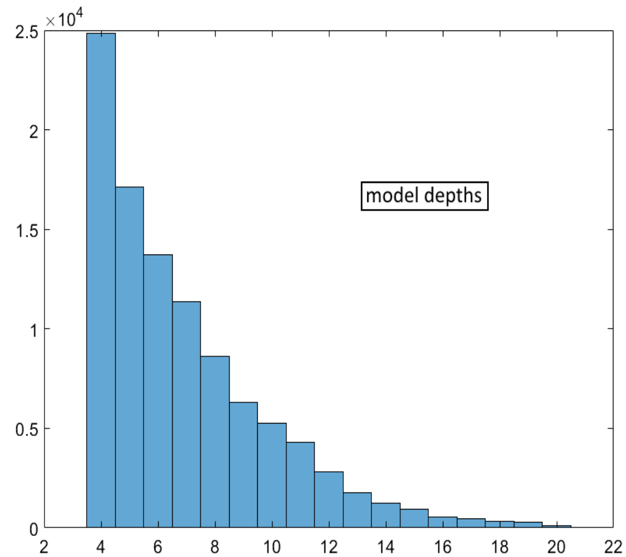
$\pi(T_4^*|x) \approx 1.8\%$

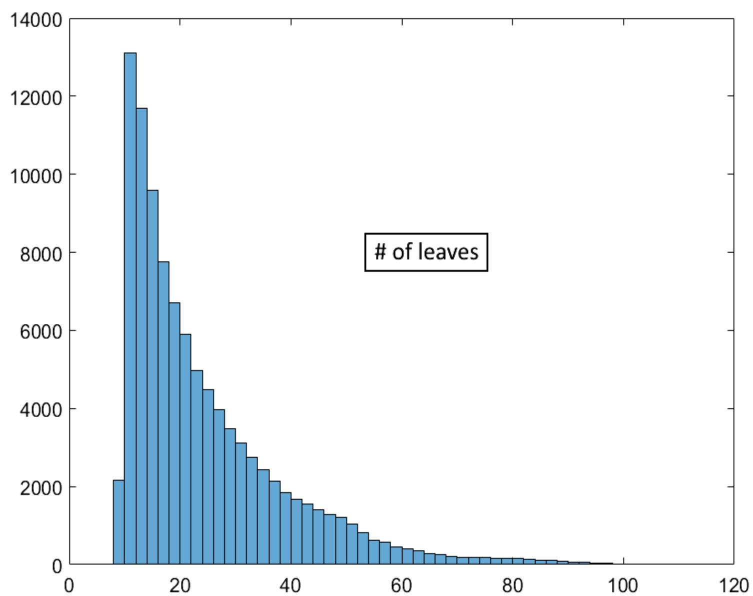


$\pi(T_5^*|x) \approx 1.8\%$

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

MCMC results on Peewee data



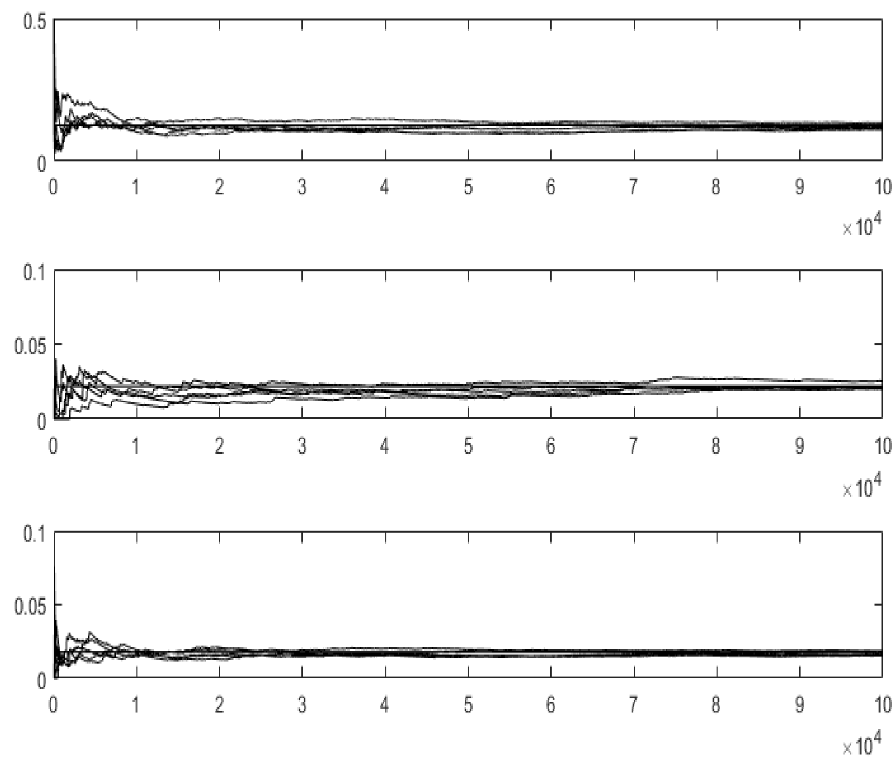


After 10^6 iterations:

Acceptance rate ≈ 59 – 61%

≈ 306,000 models visited

posterior of models visited ≈ 61.3%



[⇝ Markov order estimation]

Truly Bayesian entropy estimation

Given **Data** $X = X_{-D+1}, \ldots, X_0, X_1, \ldots, X_n$

Parameters m, D, β

Run BCT algorithm

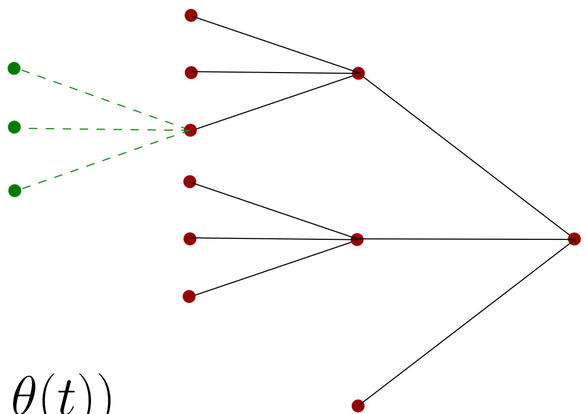
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif

Truly Bayesian entropy estimation

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

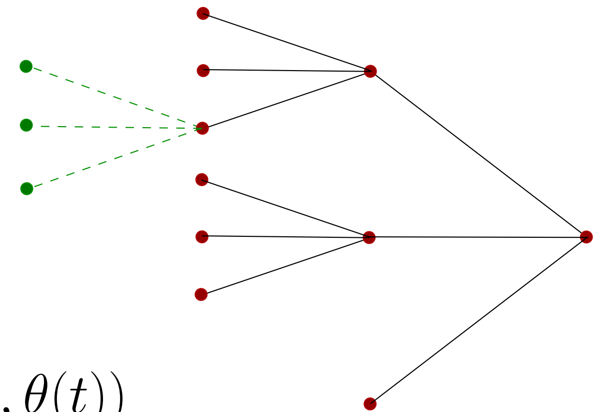
Run BCT algorithm
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif
Iterate: At each t:

△ [*Metropolis-Gibbs step*]
Obtain model and parameters sample $(T(t), \theta(t))$



Truly Bayesian entropy estimation

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



Run BCT algorithm
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif
Iterate: At each t:

△ [*Metropolis-Gibbs step*]
Obtain model and parameters sample $(T(t), \theta(t))$

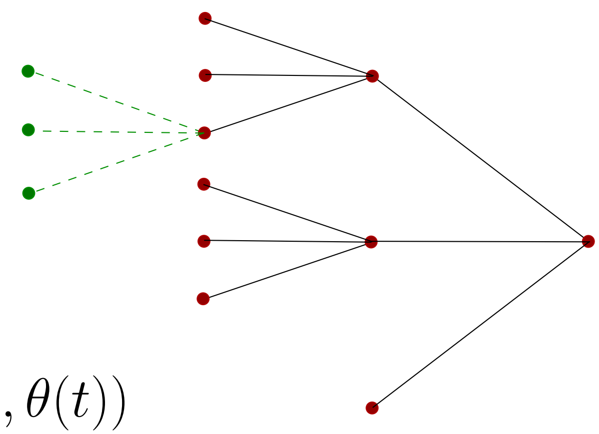
△ [*Entropy step*] Obtain entropy sample $H(t)$:

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

Truly Bayesian entropy estimation

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

Run BCT algorithm
Initialize: $T(0) = T_1^*$ and $\theta(0) \sim \prod_{s \in T(0)}$ Unif
Iterate: At each t:



△ [*Metropolis-Gibbs step*]
Obtain model and parameters sample $(T(t), \theta(t))$

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

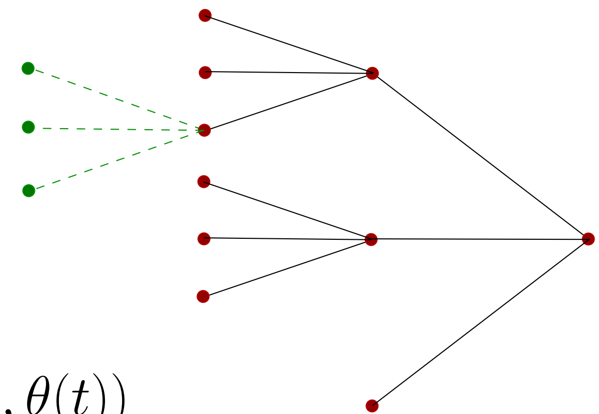
⇝ either compute $H(t) = H(T(t), \theta(t))$ directly

⇝ 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** $X = X_{-D+1}, \ldots, X_0, X_1, \ldots, X_n$
Parameters m, D, β

Run BCT algorithm



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

Iterate: At each t:

△ [*Metropolis-Gibbs step*]
Obtain model and parameters sample $(T(t), \theta(t))$

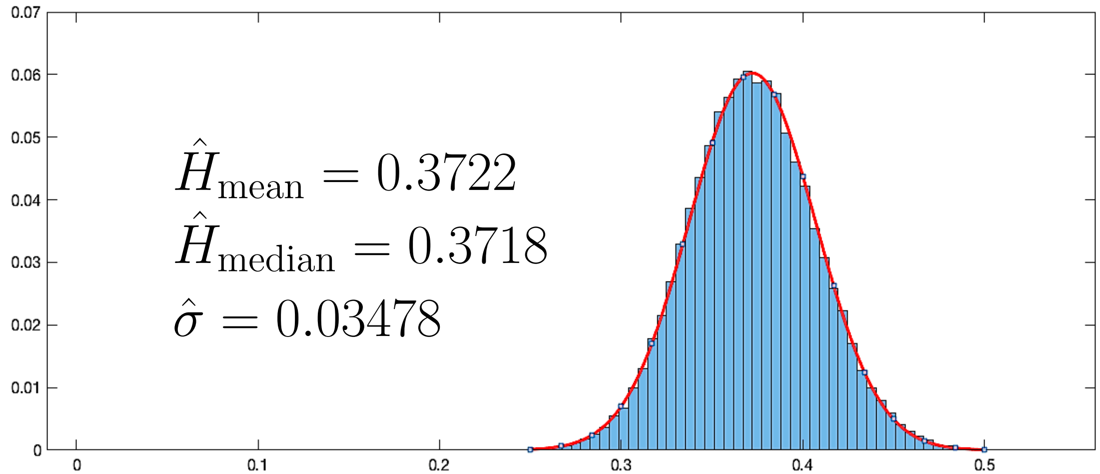
△ [*Entropy step*] Obtain entropy sample $H(t)$:

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

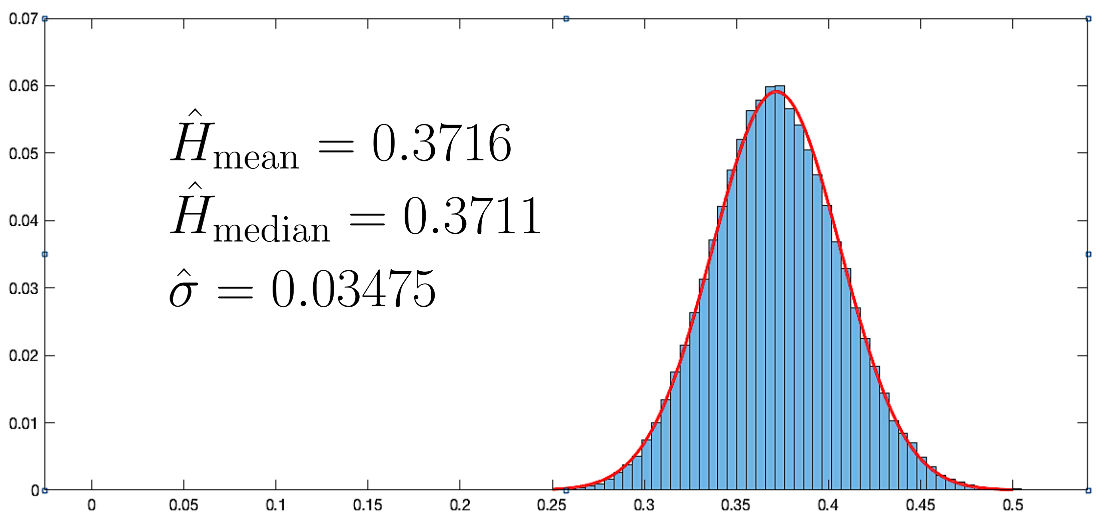
⇝ or generate $Y_1^L \sim (T(t), \theta(t))$
and estimate $H(t) = -\frac{1}{L} \log P(Y_1^L | T(t), \theta(t))$

Estimate $\pi(H|X)$
Compute posterior of H from $(H(1), H(2), \ldots, H(N))$

Entropy of Peewee song



After 10^5 MCMC iterations



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

Prediction

Observe

The **posterior predictive distribution**

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

Prediction

Observe

The **posterior predictive distribution**

$$
\begin{aligned}
f(x_{n+1}|X^n_{-D+1}) &= \sum_T \int_\theta \underbrace{f(x_{n+1}|X^n_{-D+1},\theta,T)}_{likelihood}\ \underbrace{\pi(\theta,T|X^n_{-D+1})}_{posterior}\ d\theta \\
&= \frac{f(x_{n+1},X^n_1|X^0_{-D+1})}{f(X^n_1|X^0_{-D+1})} \\
&= \frac{\text{prior predictive likelihood up to } n+1}{\text{prior predictive likelihood up to } n}
\end{aligned}
$$

Prediction

Observe

The **posterior predictive distribution**

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

$$= \frac{f(x_{n+1},X_1^n|X^0_{-D+1})}{f(X_1^n|X^0_{-D+1})}$$

$$= \frac{\text{prior predictive likelihood up to } n+1}{\text{prior predictive likelihood up to } n}$$

(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

Observe

The **posterior predictive distribution**

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

$$= \frac{f(x_{n+1}, X_1^n|X^0_{-D+1})}{f(X_1^n|X^0_{-D+1})}$$

$$= \frac{\text{prior predictive likelihood up to } n+1}{\text{prior predictive likelihood up to } n}$$

(i) can be computed *sequentially*, online

(ii) converges to the true conditional distribution

(iii) achieves the minimax optimal risk in terms of log-loss

Compare BCT with

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

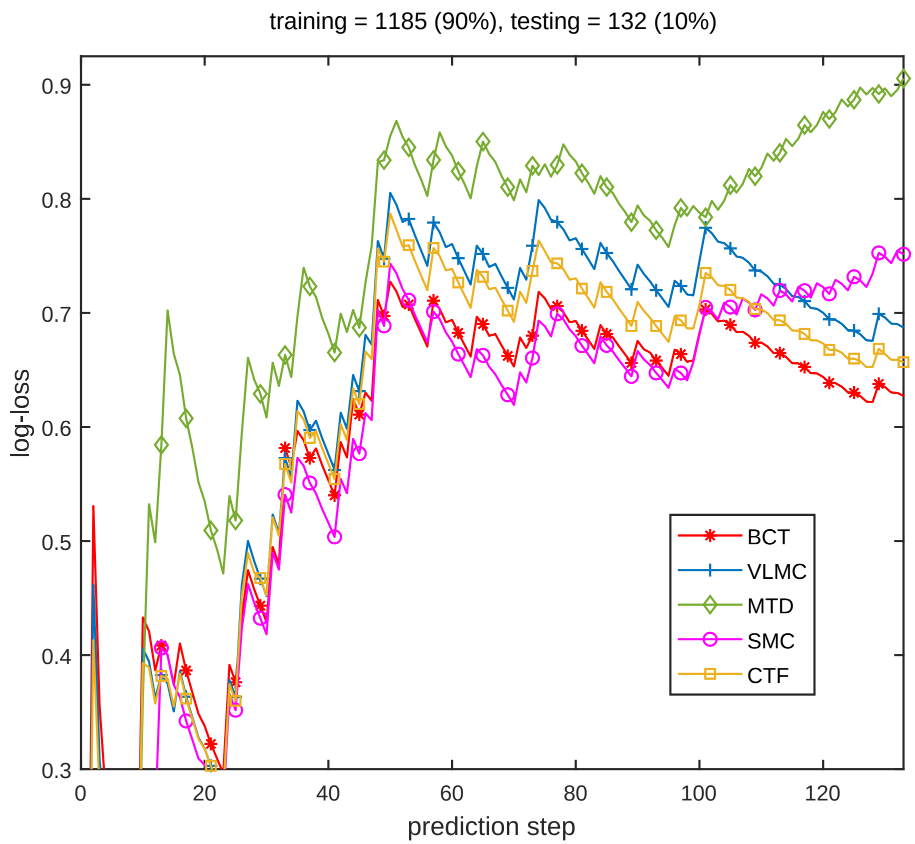
Dunson et al's **CTF** (Conditional Tensor Factorization)

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

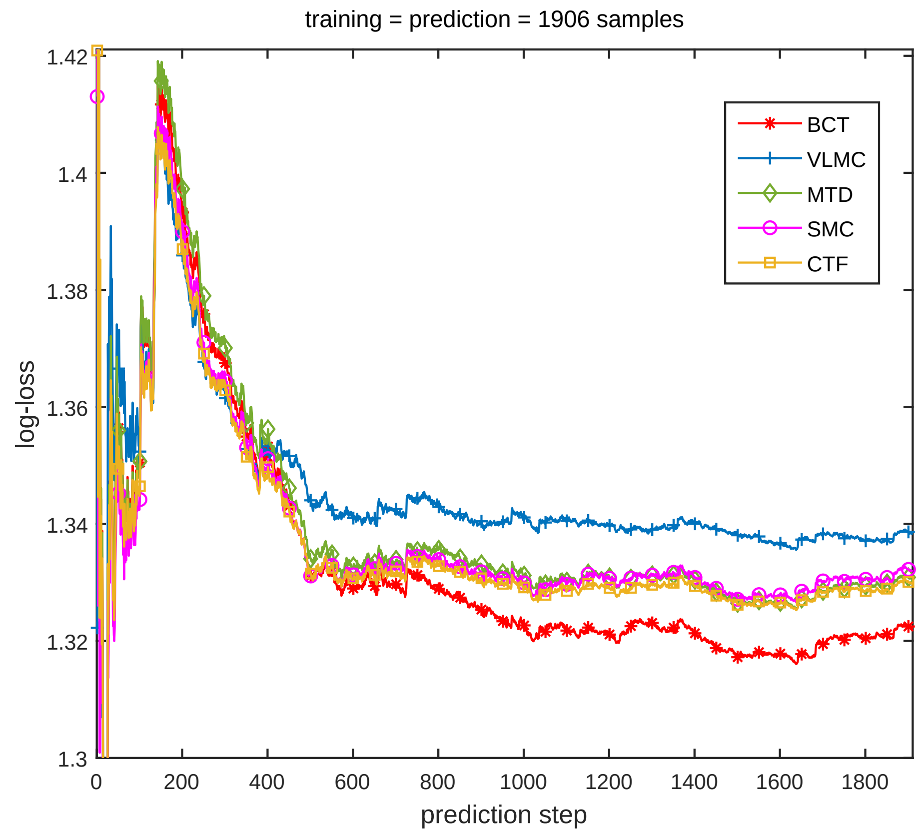
Xiong et al's **SMC** (Sparse Markov Chains)

Prediction results

Pewee song data



SARS-CoV-2 protein S gene



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 $j \in A$:

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

(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

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

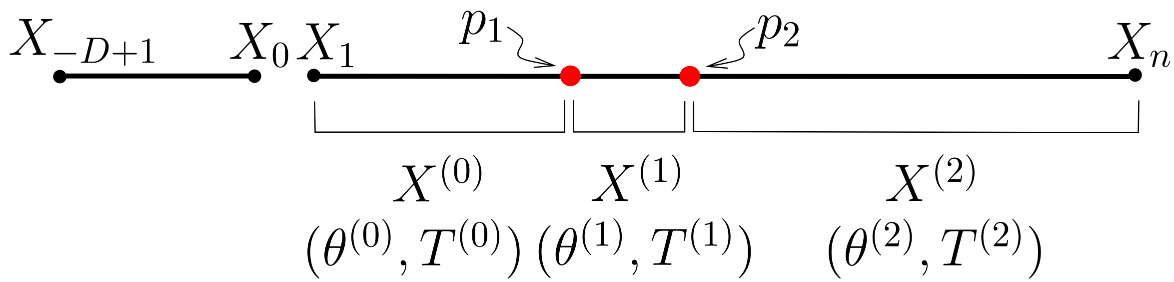
Theorem 4 [**Asymptotic normality**]

Let $\{X_n\}$ be an ergodic VMMC of depth $\leq D$, with stationary distr π
Suppose $\theta^{(n)} \sim \pi(\cdot|X^n_{-D+1}, T^*)$ and let $\bar{\theta}^{(n)}$ denote its mean. Then:

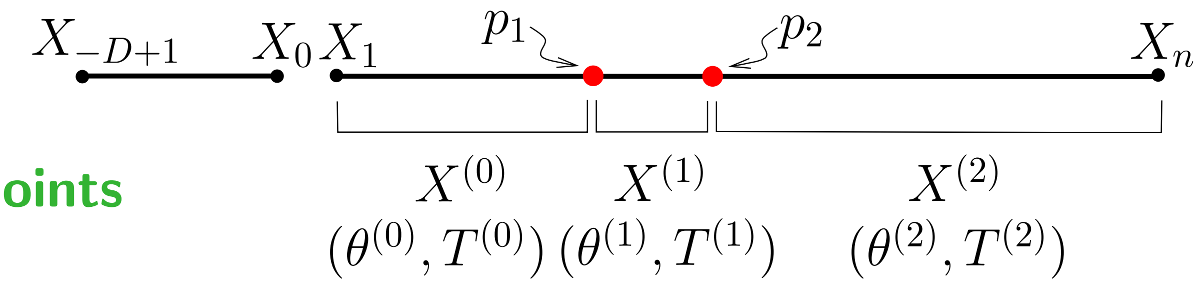
$$\sqrt{n}\left[\theta^{(n)} - \bar{\theta}^{(n)}\right] \xrightarrow{\mathcal{D}} N(0, J) \quad \text{with prob 1}$$

[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 $J_s = \frac{1}{\pi(s)}\left[\Theta_s^* - (\theta_s^*)^t(\theta_s^*)\right]$. 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

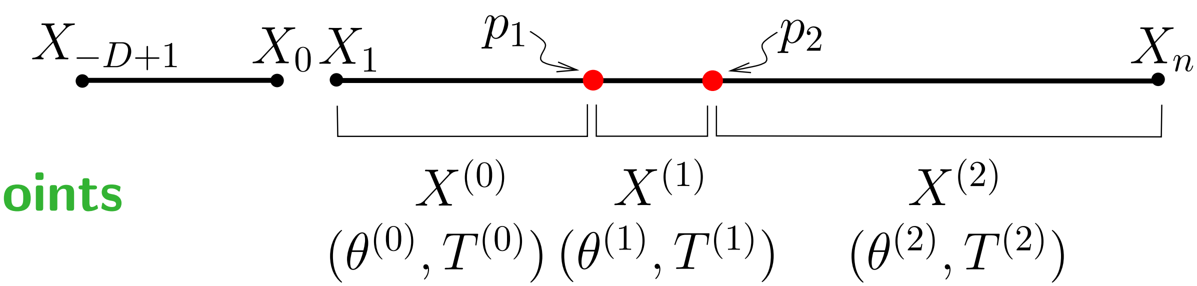


Number of changepoints

$$\mathbf{0 \leq \boldsymbol{\ell} \leq \boldsymbol{\ell}_{\max}}$$

Prior $\boldsymbol{\pi(\ell) \sim \mathrm{Po}(\lambda)} \Big| \ell \leq \ell_{\max}$

Changepoint detection: A Bayesian setting



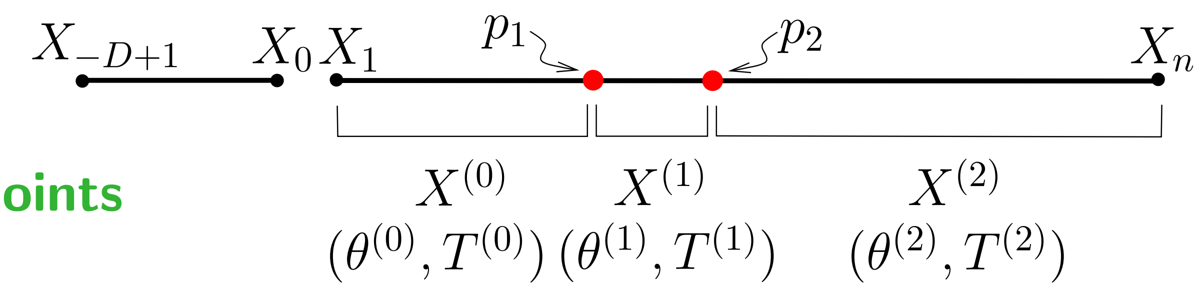
Number of changepoints

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

Prior $\pi(\ell) \sim \mathrm{Po}(\lambda)\Big|\ell \leq \ell_{max}$

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

Changepoint detection: A Bayesian setting



Number of changepoints

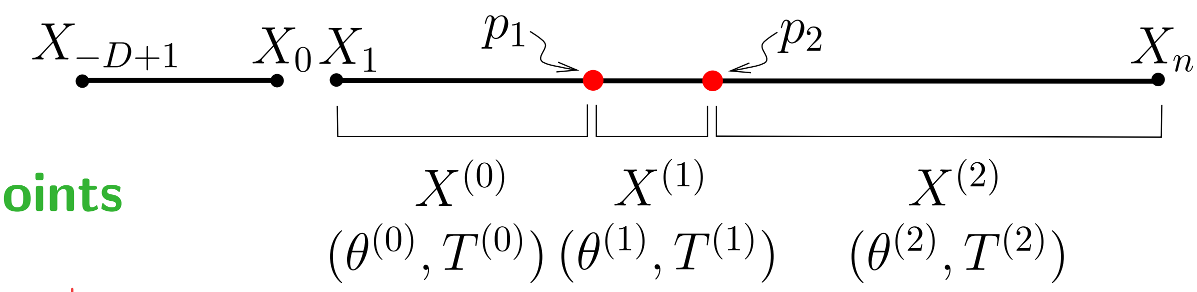
$\mathbf{0 \leq \boldsymbol{\ell} \leq \boldsymbol{\ell}_{\max}}$

Prior $\boldsymbol{\pi(\ell) \sim \mathrm{Po}(\lambda) \Big| \ell \leq \ell_{\max}}$

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

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

Changepoint detection: A Bayesian setting



Number of changepoints

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

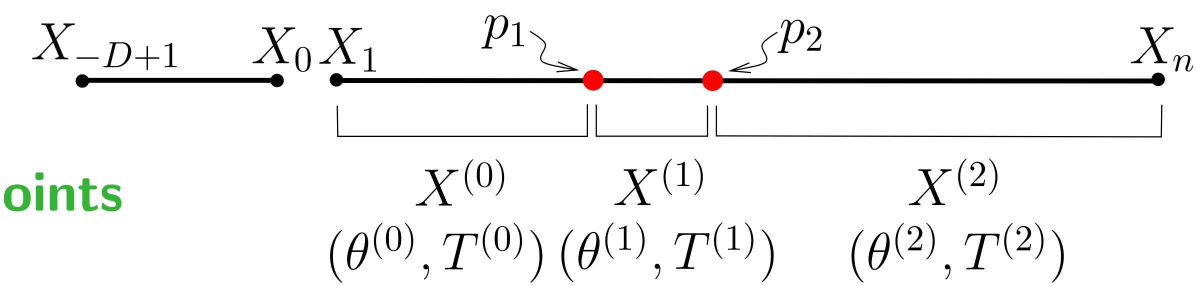
Prior $\pi(\ell) \sim \mathrm{Po}(\lambda)\Big|\ell \leq \ell_{max}$

Changepoint locations Given ℓ, the prior $\pi(p|\ell)$ of the locations $\boldsymbol{p} = (\boldsymbol{p_1, \ldots, p_\ell})$ is the distr of the even points in $2\ell + 1$ indep ordered draws from $\{1, 2, \ldots, n\}$ 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

$\mathbf{0} \le \boldsymbol{\ell} \le \boldsymbol{\ell}_{\max}$

Prior $\boldsymbol{\pi(\ell) \sim \mathrm{Po}(\lambda)\Big|\ell \le \ell_{\max}}$

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

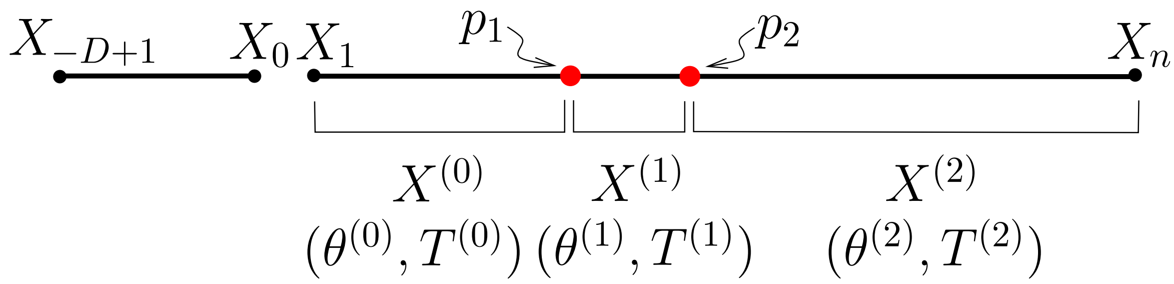
Prior on models Given ℓ, p, the $\ell + 1$ models $\{T^{(i)}\}$ are i.i.d. under $\boldsymbol{\pi(\{T^{(i)}\}|\ell, p)}$ each with distr $\pi_D(T^{(i)})$ as before

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

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

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

BCT changepoint detection



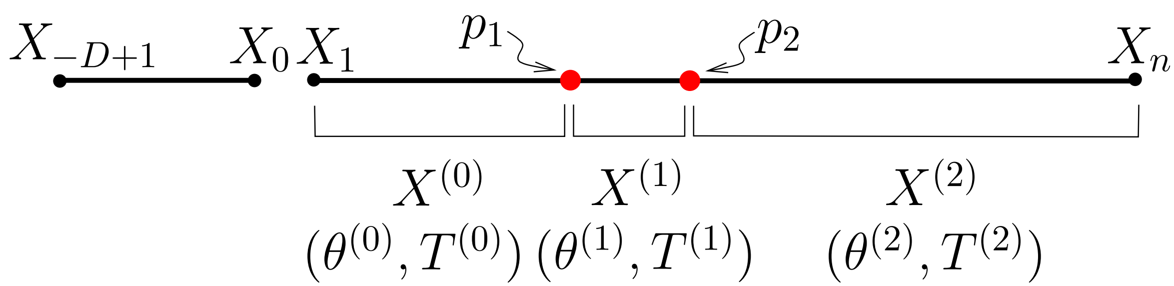
Goal From the likelihood

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

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

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** $\boldsymbol{\pi(\ell, p|X)}$

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

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

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

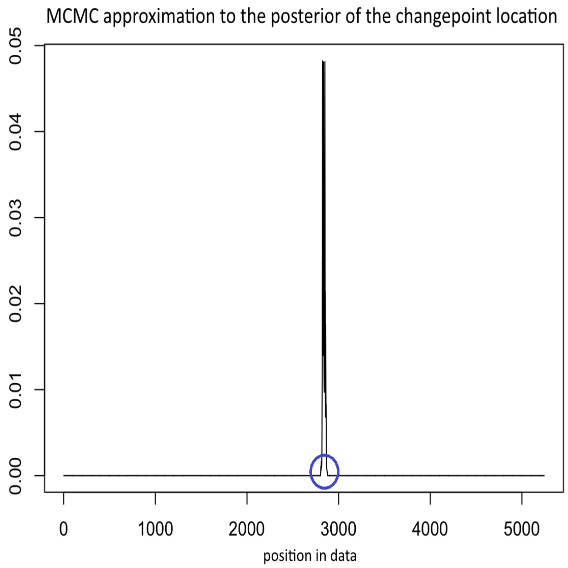
$\rightsquigarrow$ **Effective MCMC** We can sample from the desired posterior $\boldsymbol{\pi(\ell, p|X)}$ with a Metropolis-Hastings sampler that employs the CTW in each step

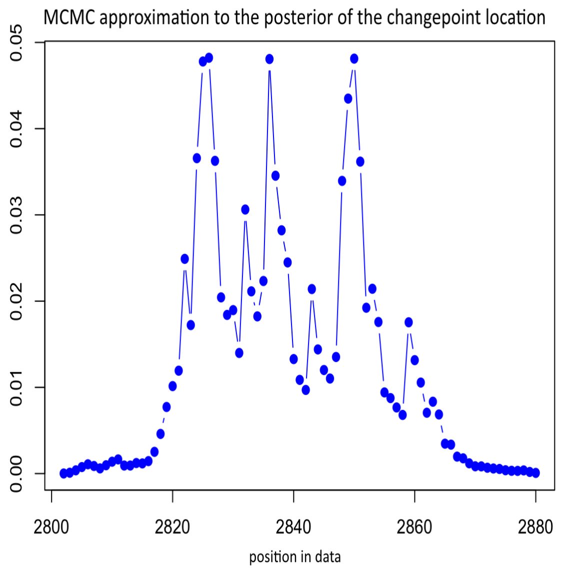
Changepoint detection results

Simian vacuolating virus 40:

DNA genome, $n = 5243$ bp

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

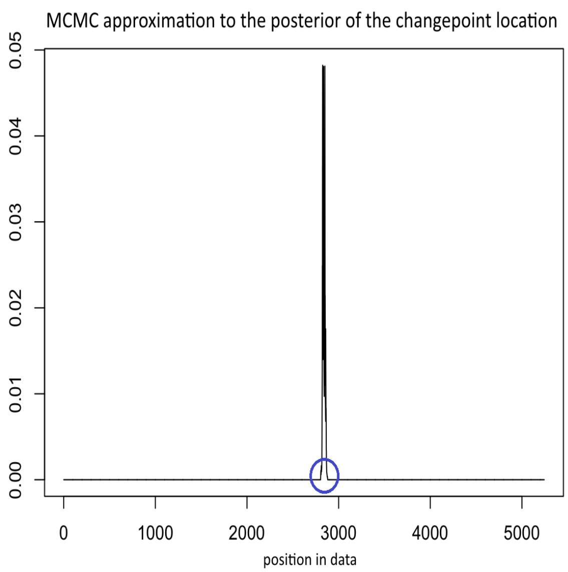


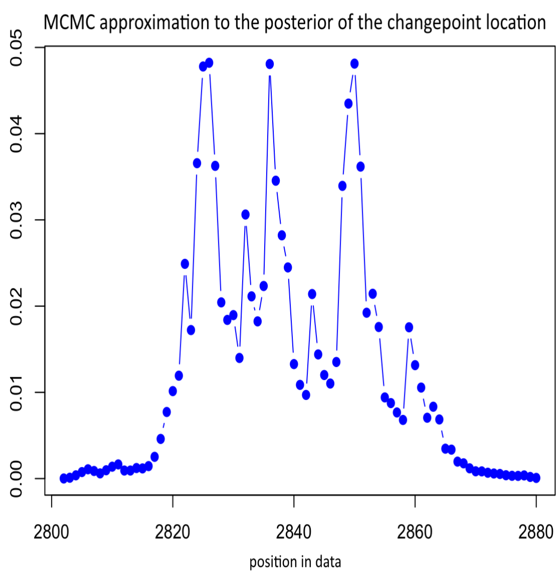


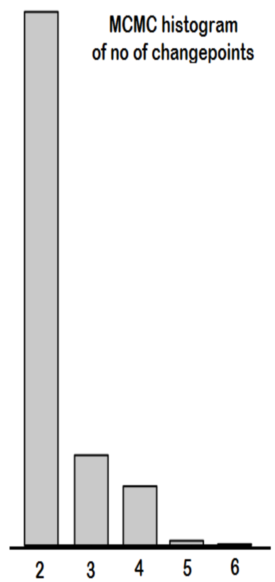
Changepoint detection results

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

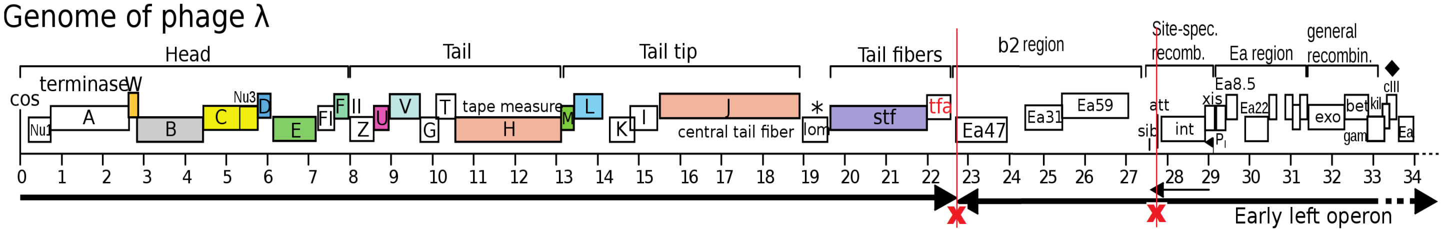
Enterophage λ bacterium: Examine $n = 35$ Kbp of its DNA genome, with $D = 10$, $\ell_{\max} = 10$





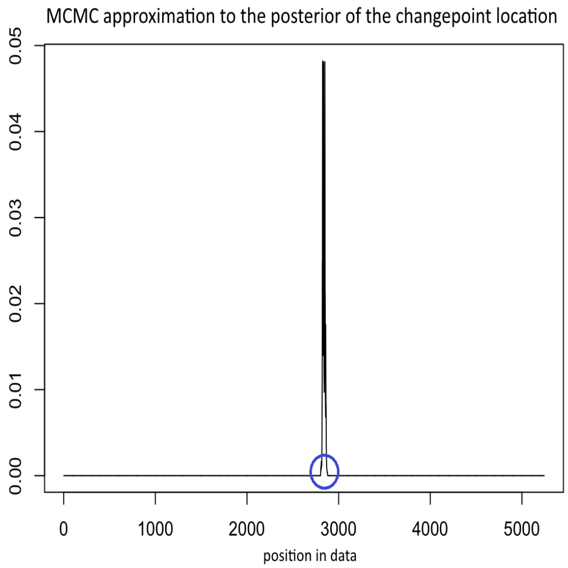


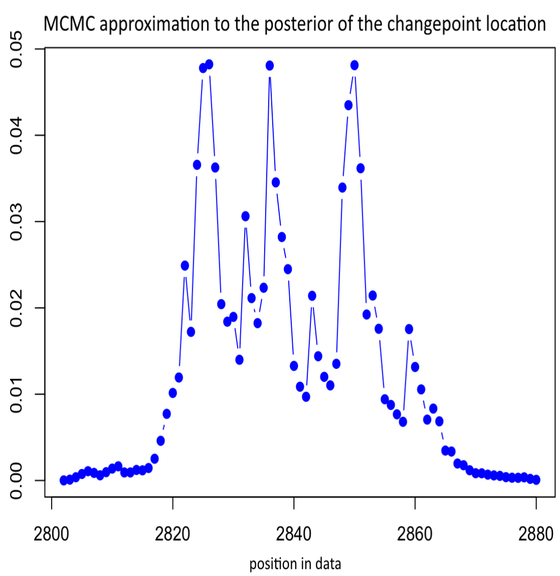
Genome of phage λ



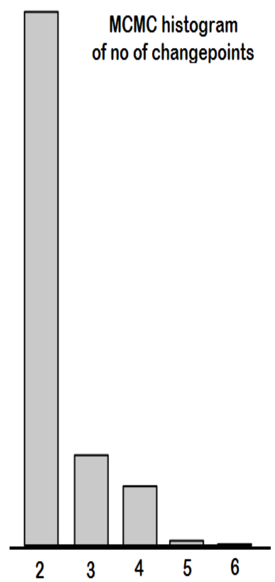
Changepoint detection results

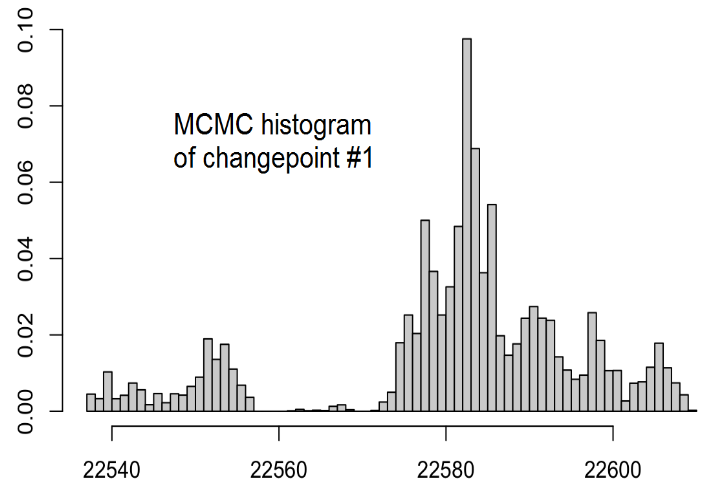
Simian vacuolating virus 40:
DNA genome, $n = 5243$ bp
MCMC: $D = 5$, $\ell_{\max} = 1$

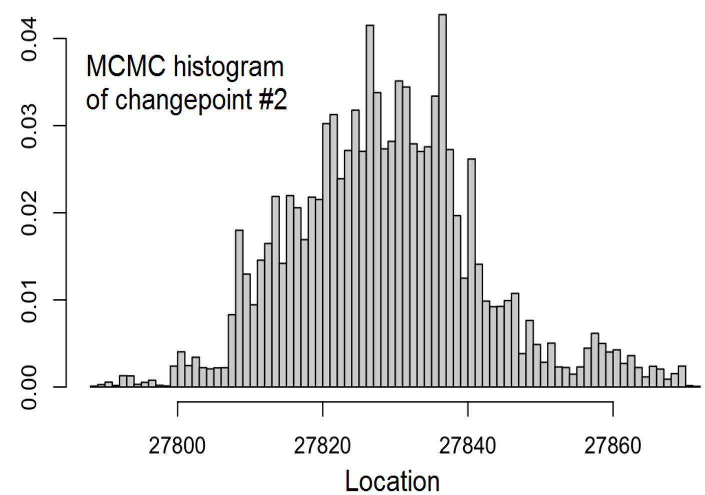




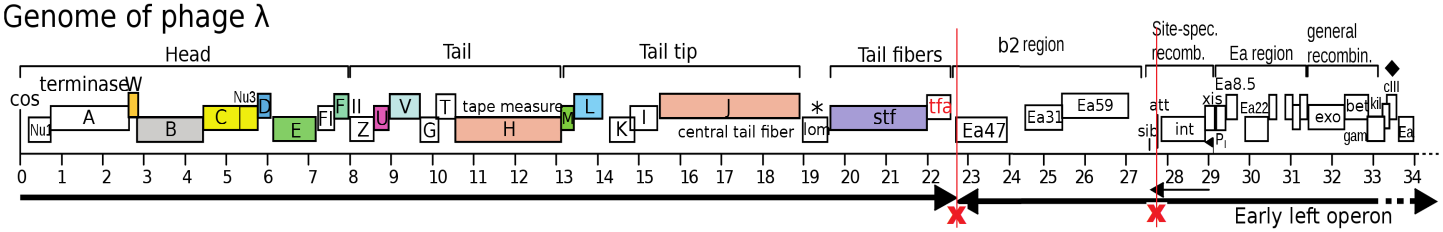
Enterophage λ bacterium: Examine $n = 35$ Kbp of its DNA genome, with $D = 10$, $\ell_{\max} = 10$







Genome of phage λ



Real-valued time series: The BCT-X framework

Goal Perform effective Bayesian inference with real-valued time series

Real-valued time series: The BCT-X framework

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 $A = \{0, 1, \ldots, m-1\}$ 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 $(\ldots, X_{n-2}, X_{n-1})$ as the distribution dictated by (M_s, ϕ_s), where $s \in T$ is the context corresponding to $Y_{n-D}, \ldots, Y_{n-1}$

Real-valued time series: The BCT-X framework

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 $A = \{0, 1, \ldots, m-1\}$ 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 $(\ldots, X_{n-2}, X_{n-1})$
as the distribution dictated by (M_s, ϕ_s), where $s \in T$ is the context
corresponding to $Y_{n-D}, \ldots, Y_{n-1}$

Main step: Inference

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

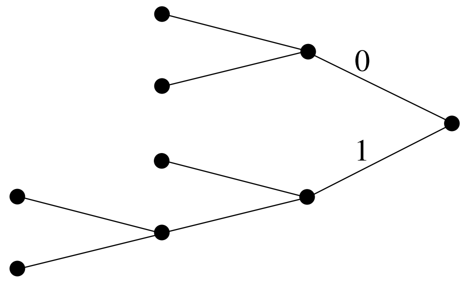
i. Perform inference jointly on (M, ϕ) and on (T, θ)

ii. Develop new methodological tools in applications

Example: The binary BCT-AR model class

Model class

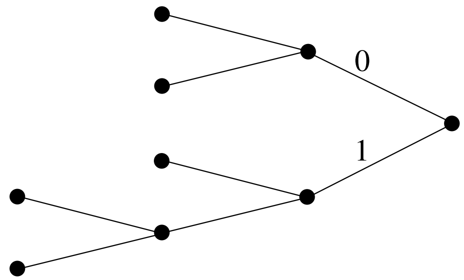
▷ Binary context trees T of depth $\leq D$



Example: The binary BCT-AR model class

Model class

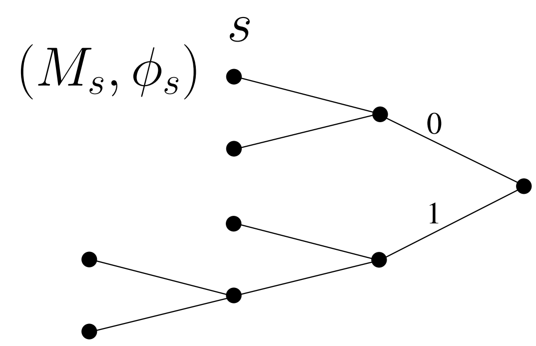
- Binary context trees T of depth $\leq D$
- Binary quantizer: $\mathbb{R} \to \{0, 1\}$



Example: The binary BCT-AR model class

Model class

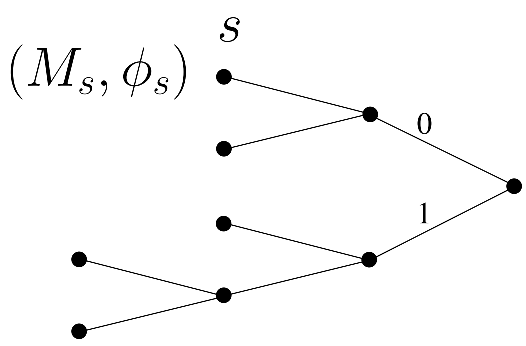
- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \to \{0,1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$
- ▷ Inv-Gamma/Gaussian AR(p) 'conjugate' priors



Example: The binary BCT-AR model class

Model class

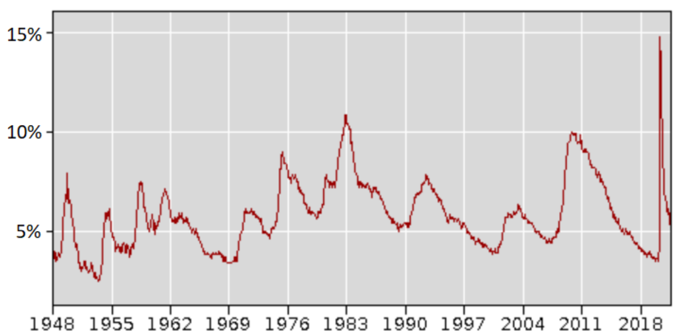
- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \to \{0, 1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$
- ▷ Inv-Gamma/Gaussian AR(p) 'conjugate' priors



Application US unemployment data

Sequence X of differences of quarterly rates

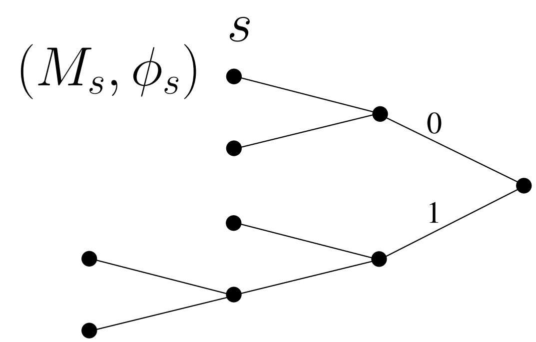
1948-2019: $n = 288$ observations



Example: The binary BCT-AR model class

Model class

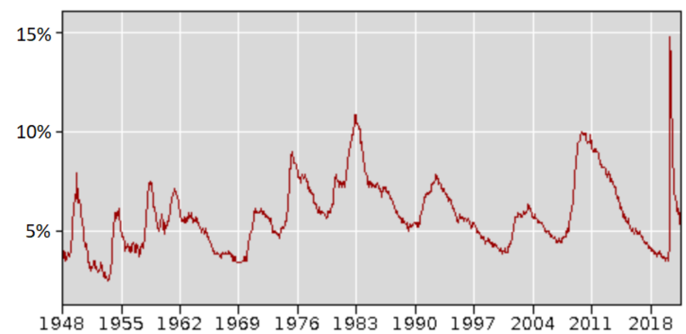
- ▷ Binary context trees T of depth $\leq D$
- ▷ Binary quantizer: $\mathbb{R} \to \{0, 1\}$
- ▷ AR model M_s and parameters ϕ_s at each $s \in T$
- ▷ Inv-Gamma/Gaussian AR(p) ‘conjugate’ priors



Application US unemployment data

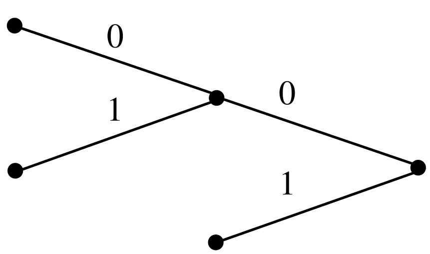
Sequence X of differences of quarterly rates

1948-2019: $n = 288$ observations



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

Estimated quantizer threshold is $\approx 0.15\%$



Mean squared error (MSE) of forecasts

	Prediction step				
Model	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**

Example: The binary BCT-ARCH model class

Modelling observations

- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$

Example: The binary BCT-ARCH model class

Modelling observations

- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$
- **Heteroskedasticity** (variance changes)
  $\rightsquigarrow$ model the variance

Example: The binary BCT-ARCH model class

Modelling observations

- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$
- **Heteroskedasticity** (variance changes)

  $\rightsquigarrow$ model the variance
- **Volatility clustering** $\rightsquigarrow$ allow dependence

  $\rightsquigarrow$ ARCH: $y_n \sim N(0, \sigma_n^2) \qquad \sigma_n^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j y_{n-j}^2$

Example: The binary BCT-ARCH model class

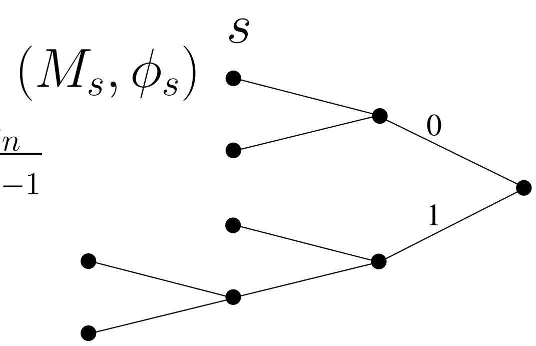
Modelling observations

- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$
- **Heteroskedasticity** (variance changes)
  $\rightsquigarrow$ model the variance
- **Volatility clustering** $\rightsquigarrow$ allow dependence

  $\rightsquigarrow$ ARCH: $y_n \sim N(0, \sigma_n^2) \qquad \sigma_n^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j y_{n-j}^2$
- **Leverage effect**: Asymmetric volatility response to +ve and -ve shocks

Example: The binary BCT-ARCH model class

Modelling observations



- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$
- **Heteroskedasticity** (variance changes)

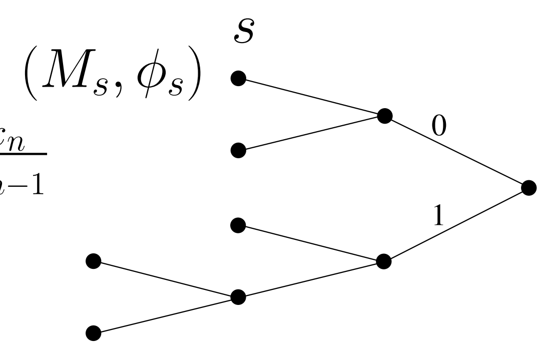
  $\rightsquigarrow$ model the variance
- **Volatility clustering** $\rightsquigarrow$ allow dependence

  $\rightsquigarrow$ ARCH: $$y_n \sim N(0, \sigma_n^2) \qquad \sigma_n^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j y_{n-j}^2$$
- **Leverage effect**: Asymmetric volatility response to +ve and -ve shocks

Model class

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

Example: The binary BCT-ARCH model class



Modelling observations

- **Nonstationarity** $\rightsquigarrow$ transform data: $y_n = \log \frac{x_n}{x_{n-1}}$
- **Heteroskedasticity** (variance changes)
  $\rightsquigarrow$ model the variance
- **Volatility clustering** $\rightsquigarrow$ allow dependence

  $\rightsquigarrow$ ARCH: $y_n \sim N(0, \sigma_n^2)$ $\quad \sigma_n^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j y_{n-j}^2$
- **Leverage effect**: Asymmetric volatility response to +ve and -ve shocks

Model class

- Binary context trees T of depth $\leq D$
- Binary quantizer with threshold $c = 0$
- ARCH model M_s and parameters ϕ_s at each $s \in T$
- Non-informative non-conjugate priors
  $\rightsquigarrow$ Laplace approximation for the marginal likelihoods

BCT-ARCH application: Market index data

Data FTSE, CAC and DAX $n = 7821$ daily values

Parameters ARCH $p_{\max} = 5$, $D = 5$

BCT-ARCH application: Market index data

Data FTSE, CAC and DAX $n = 7821$ daily values

Parameters ARCH $p_{\max} = 5$, $D = 5$

Prediction results

Table: Comparing the predictive ability of volatility models in terms of the cumulative log-loss

	BCT-ARCH	ARCH	GARCH	GJR	EGARCH	MSGARCH	SV
ftse	$-\mathbf{161.9}$	-157.7	-154.5	-159.7	-159.0	-159.7	-154.4
cac40	$-\mathbf{112.5}$	-108.6	-108.7	-111.0	-112.4	-109.2	-106.9
dax	$-\mathbf{111.7}$	-105.9	-105.4	-106.4	-107.5	-106.1	-103.2

BCT-ARCH application: Market index data

Data FTSE, CAC and DAX $n = 7821$ daily values

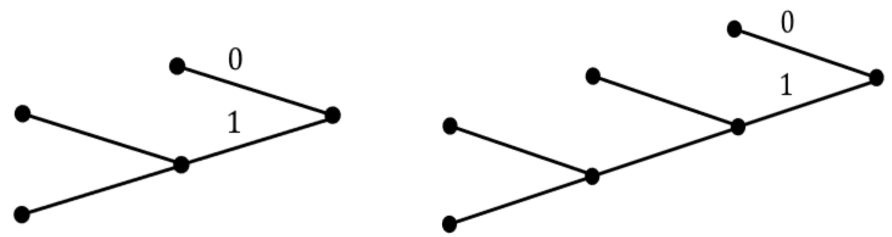
Parameters ARCH $p_{\max} = 5$, $D = 5$

Prediction results

Table: Comparing the predictive ability of volatility models in terms of the cumulative log-loss

	BCT-ARCH	ARCH	GARCH	GJR	EGARCH	MSGARCH	SV
ftse	**−161.9**	−157.7	−154.5	−159.7	−159.0	−159.7	−154.4
cac40	**−112.5**	−108.6	−108.7	−111.0	−112.4	−109.2	−106.9
dax	**−111.7**	−105.9	−105.4	−106.4	−107.5	−106.1	−103.2

Modelling results



MAP context-tree models for FTSE 100 (left), CAC 40 and DAX (right)

$\pi(T^*|x) \approx 95\%$ $\quad$ $\pi(T^*|x) \approx 64\%$ $\quad$ $\pi(T^*|x) \approx 49\%$

$\rightsquigarrow$ Enhanced leverage effect

Extensions and further results

⇝ Applications

Model selection	Estimation	Change-point detection
Segmentation	Anomaly detection	Markov order estimation
Filtering	Prediction	Entropy estimation
Causality testing	Compression	Content recognition

⇝ Results on real data

▷ Satellite imaging ▷ genetics ▷ neuroscience
▷ finance ▷ economics
▷ 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