Bayesian Suffix Trees
& Context Tree Weighting

Ioannis Kontoyiannis
Cambridge U

International Zurich Seminar on Information and Communication
February 2018

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

Co-financed by Greece and the European Union

Motivation

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

Inference | Signal processing
Machine learning | Communications

Motivation

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

Inference | Signal processing
Machine learning | Communications

$\leadsto$ **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

$\leadsto$ **Need A LOT of data**

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 A LOT of data**

$\rightsquigarrow$ **Difficulty**: **Big Data**

Most existing methods do not realistically scale with large data

Even "Big Data" are not enough for classical estimation

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

Earlier Work

- The starting point of our work is based in part on:
  - **Rissanen**'s 1983 – 1986 fundamental work on the Minimum Description Length (MDL) principle and the introduction of tree/FSMX sources
  - The basic results of **Willems** et al on data compression via **Context Tree Weighting** (CTW) and related algorithms
- Some of our first results can be seen as generalizations or extensions of results and algorithms in these earlier works
- Here we ignore the information-theoretic connection entirely and present everything from the point of view of Bayesian statistics (and applications)
- Our framework can also be viewed as a Bayesian version of Bühlmann et al's **VLMC**

Outline

Background: **Variable-memory Markov chains**

Bayesian Modelling and Inference

Prior structure, marginal likelihood, the posterior

Efficient Algorithms

MMLA, MAPT, k-MAPT, MCMC

Outline

Background: Variable-memory Markov chains

Bayesian Modelling and Inference

Prior structure, marginal likelihood, the posterior

Efficient Algorithms

MMLA, MAPT, k-MAPT, MCMC

Theory $\rightsquigarrow$ **The algorithms work**

$\rightsquigarrow$ **Classical justifications & asymptotics**

Experimental Results $\rightsquigarrow$ **How the algorithms work**

Outline

Background: Variable-memory Markov chains

Bayesian Modelling and Inference

Prior structure, marginal likelihood, the posterior

Efficient Algorithms

MMLA, MAPT, k-MAPT, MCMC

Theory $\rightsquigarrow$ **The algorithms work**

$\rightsquigarrow$ **Classical justifications & asymptotics**

Experimental Results $\rightsquigarrow$ **How the algorithms work**

Applications

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

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 $P(X_n|X_{n-1}, \ldots, X_{n-d})$ one for each context $(X_{n-1}, \ldots, X_{n-d})$

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 $P(X_n | X_{n-1}, \ldots, X_{n-d})$ one for each context $(X_{n-1}, \ldots, X_{n-d})$

Problem $\boldsymbol{m^d}$ grows very fast, e.g., with $m = 8$ symbols and memory length $d = 10$, we need $\approx 10^9$ distributions

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 $P(X_n | X_{n-1}, \ldots, X_{n-d})$
one for each context $(X_{n-1}, \ldots, X_{n-d})$

Problem $\boldsymbol{m^d}$ grows very fast, e.g., with $m = 8$ symbols
and memory length $d = 10$, we need $\approx 10^9$ distributions

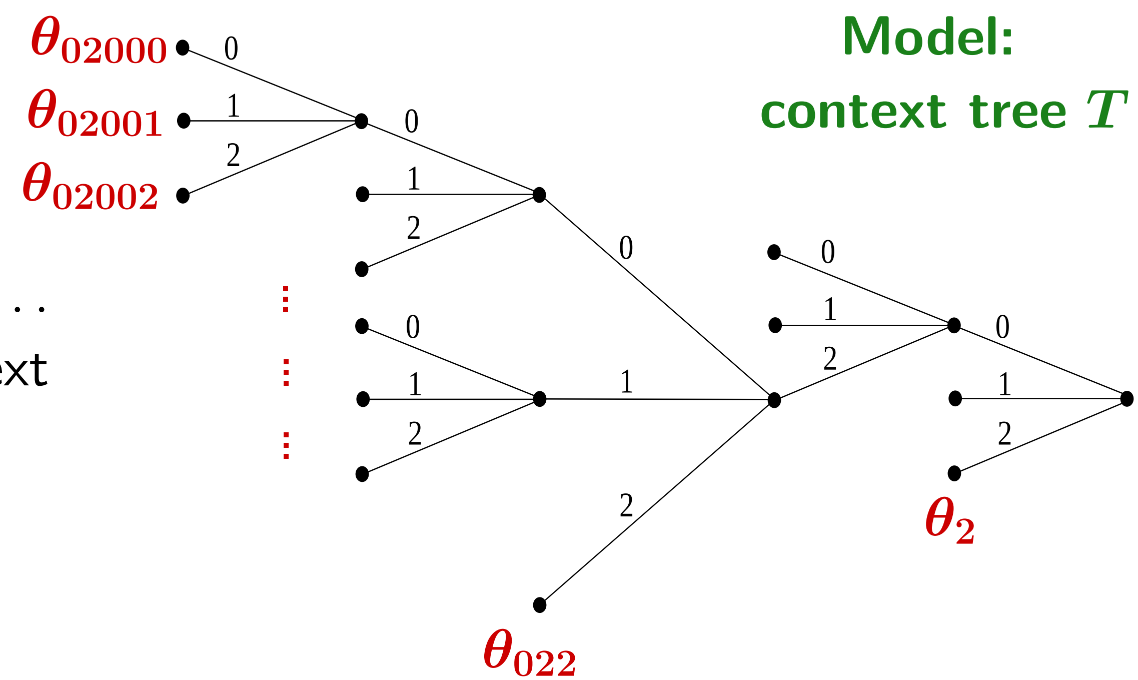
Idea Use *variable length contexts* described by a **context tree** $\boldsymbol{T}$

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

Model: context tree T

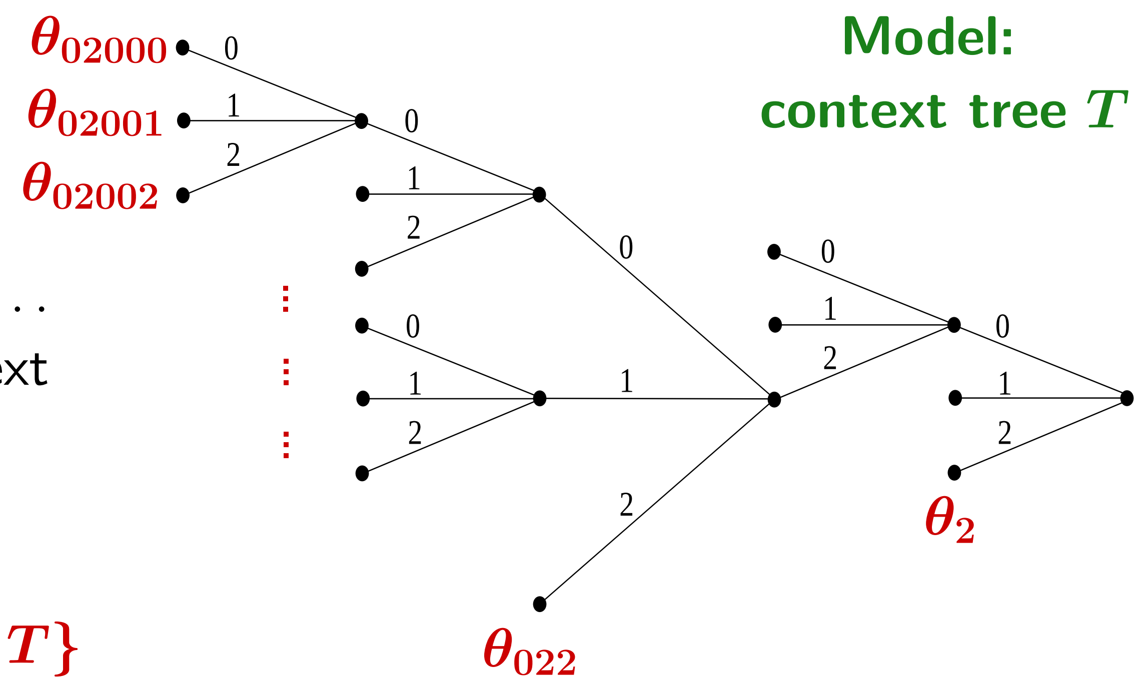


Variable-Memory Markov Chains: An Example

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

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 Representation: Advantages

- **Parsimony** E.g. above with memory length 5 instead of $3^5 = 243$ conditional distributions, only need to specify 13

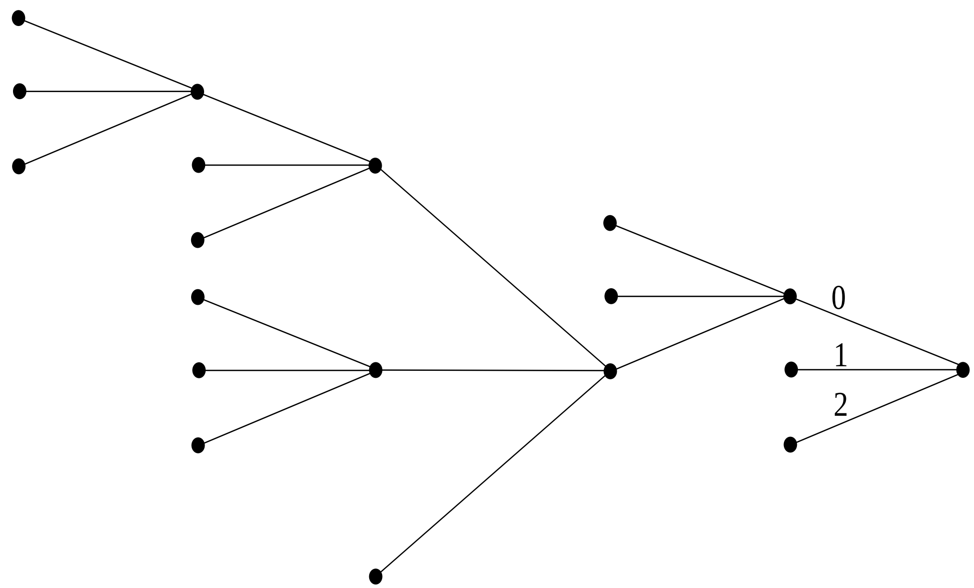
- For an alphabet of size m and memory depth d there are m^d contexts $\Rightarrow$ potentially huge savings

Variable-Memory Representation: Advantages

- **Parsimony** E.g. above with memory length 5 instead of $3^5 = 243$ conditional distributions, only need to specify 13

- For an alphabet of size m and memory depth d there are m^d contexts $\Rightarrow$ potentially huge savings

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



VMMCs: Computation of the Likelihood

Notation.

1. Models $\equiv$ Trees
2. X_i^j denotes the block $(X_i, X_{i+1}, \ldots, X_j)$
3. $\theta = \{\theta_s; s \in T\}$ for all the parameters (given T)
4. $X = X_{-d+1}, \ldots X_0, X_1, \ldots, X_n$ all the observed data
5. Suppress dependence of the likelihood on the past $X^0{}_{-d+1}$

VMMCs: Computation of the Likelihood

Notation.

1. Models $\equiv$ Trees
2. X_i^j denotes the block $(X_i, X_{i+1}, \ldots, X_j)$
3. $\theta = \{\theta_s; s \in T\}$ for all the parameters (given T)
4. $X = X_{-d+1}, \ldots X_0, X_1, \ldots, X_n$ all the observed data
5. Suppress dependence of the likelihood on the past X_{-d+1}^0

The **likelihood** of $X = X_1^n$ is:

$$f(X) \;=\; f(X_1^n | X_{-d+1}^0, \theta, T) \;=\; \prod_{i=1}^n P(X_i | X_{i-d}^{i-1}) \;=\; \prod_{s \in T} \prod_{j \in A} \theta_s(j)^{a_s(j)}$$

where the **count vectors a_s** are defined by:

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

Bayesian Modelling for VMMCs

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

[**Lemma**: This is OK]

Bayesian Modelling for VMMCs

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

[**Lemma**: This is OK]

Prior on parameters Given a context tree T, the parameters $\theta = \{\theta_s; s \in T\}$ are taken to be independent

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

Bayesian Modelling for VMMCs

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

[**Lemma**: This is OK]

Prior on parameters Given a context tree T, the parameters $\theta = \{\theta_s; s \in T\}$

are taken to be independent

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

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

the likelihood of $X = X_1^n$ is as above:

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

Bayesian Inference for VMMCs

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

Bayesian Inference for VMMCs

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

Max model depth D

The "one and only" goal of Bayesian inference

Determination of the **posterior distributions**:

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

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

Bayesian Inference for VMMCs

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

The "one and only" 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 **mean marginal 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 $\boldsymbol{P_e(a_s)} = \dfrac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$

with the count vectors a_s 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 $\boldsymbol{P_e(a_s)} = \dfrac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$

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

What perhaps should be surprising is that the entire
mean marginal likelihood $f(X) = \sum_T \pi(T) f(X|T)$
can also be computed effectively

The Mean Marginal Likelihood Algorithm (MMLA)

Given. **Data** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$ [*The algorithm*
Alphabet size m **Maximum depth** D *formerly known*
Prior parameter β *as CTW*]

The Mean Marginal Likelihood Algorithm (MMLA)

Given. **Data** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$ [*The algorithm formerly known as CTW*]

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

The Mean Marginal Likelihood Algorithm (MMLA)

Given. **Data** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$ [*The algorithm formerly known as CTW*]
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

The Mean Marginal Likelihood Algorithm (MMLA)

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

[*The algorithm formerly known as CTW*]

△ **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 MMLA Computes the Mean Marginal Likelihood

Theorem

The weighted probability $P_{w,\lambda}$ given by the MMLA at the root λ is exactly equal to the mean marginal 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 MMLA Computes the Mean Marginal Likelihood

Theorem

The weighted probability $P_{w,\lambda}$ given by the MMLA at the root λ is exactly equal to the mean marginal 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 MMLA computes a "doubly exponentially hard" quantity
in $O(n \cdot D^2)$ time

The MMLA can be updated *sequentially*

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

Maximum A Posteriori Probability Tree Algorithm (MAPT)

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

[*The algorithm formerly known as CTM*]

△ **1.** [*Tree.*] and △ **2.** [*Estimated probabilities.*]
Construct the tree and compute a_s and $P_{e,s}$ as before

Maximum A Posteriori Probability Tree Algorithm (MAPT)

Given. **Data** $X = X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$ [*The algorithm formerly known as CTM*]

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

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

Maximum A Posteriori Probability Tree Algorithm (MAPT)

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

[*The algorithm formerly known as CTM*]

△ **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

Theorem: The MAPT Computes the MAP Tree

Theorem

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

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

Theorem: The MAPT Computes the MAP Tree

Theorem

The (pruned) tree T_1^* resulting from the MAPT 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 MMLA

The MAPT also computes a doubly exponentially hard quantity in $O(n \cdot D^2)$ 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

Finding the k A Posteriori Most Likely Trees (k-MAPT)

△ **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 MAPT

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

k-MAPT Finds the k A Posteriori Most Likely Trees

Theorem

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

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

k-MAPT Finds the k A Posteriori Most Likely Trees

Theorem

The k trees $T_1^*, T_2^*, \ldots, T_k^*$ described recursively at the root
after the k-MAPT 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-MAPT is $O(n \cdot D^2 \cdot k^m)$ in both time and space

This is one of the very few examples of nontrivial Bayesian models
for which the area near the mode of the posterior is explicitly identifiable
certainly the most complex/interesting one

Experimental results: MAP model for a 5th Order Chain

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

MAPT Find MAP models with max depth $D = 1, 2, 3, \ldots,\ \beta = 1/2$

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

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

Experimental results: MAP model for a 5th Order Chain

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

Alphabet size $m = 3$

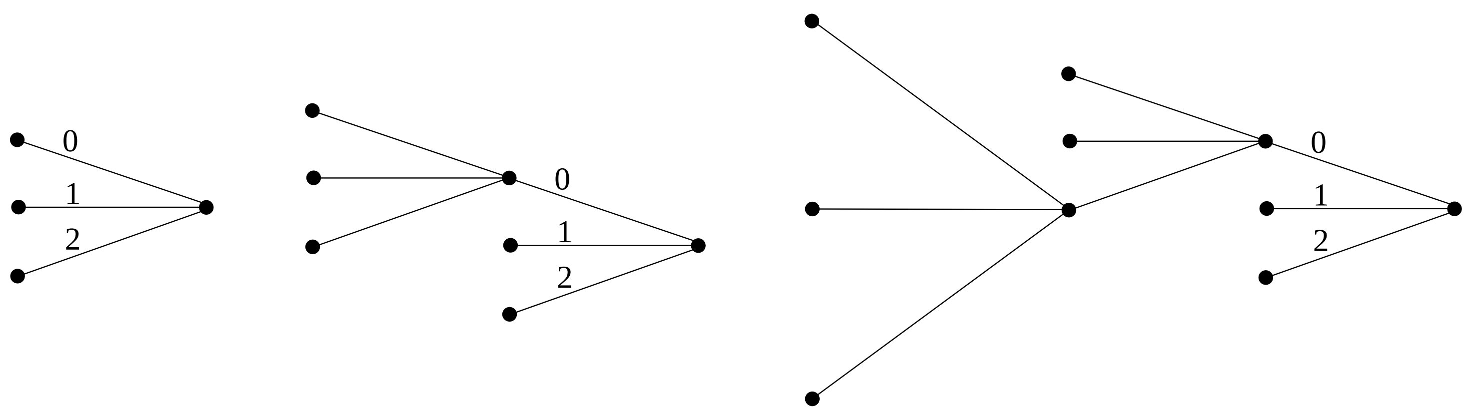
VMMC with $d = 5$ as in the example

Data length $n = 10000$ samples

MAPT Find MAP models with max depth $D = 1, 2, 3, \ldots,\ \beta = 1/2$

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

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



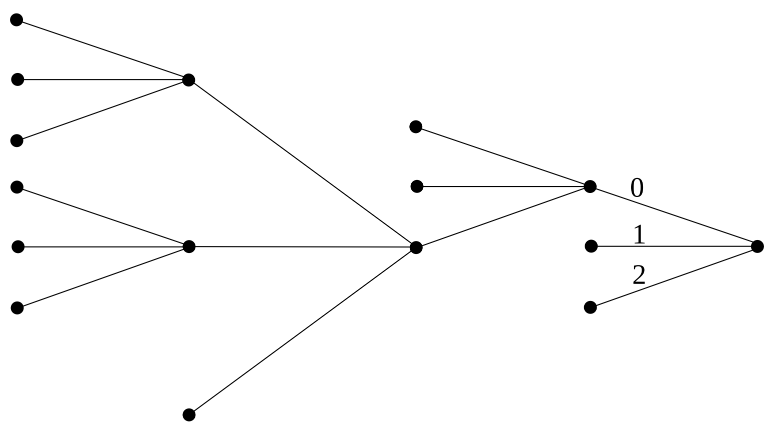
$D = 1$ $D = 2$ $D = 3$

MAP model for a 5th Order Chain (cont'd)

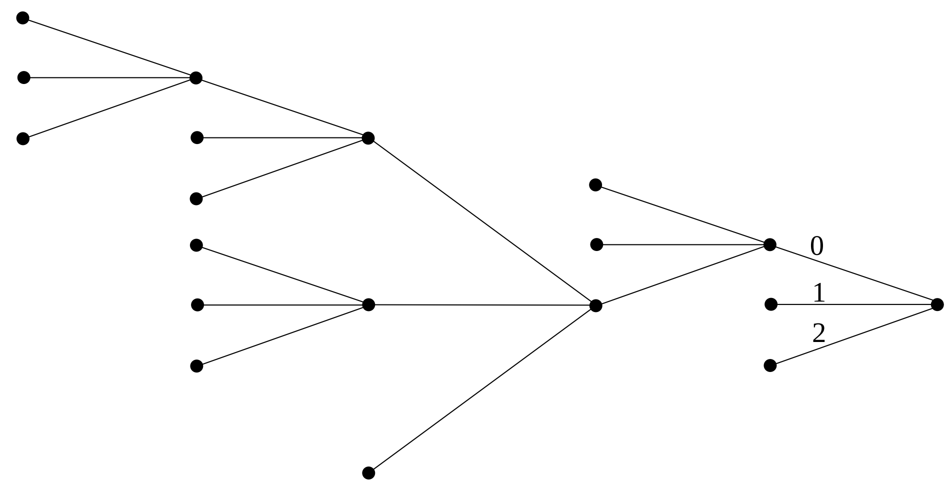
5th order VMMC data $X_{-D+1}, \ldots, X_0, X_1, X_2, \ldots, X_n$

$$m = 3,\ d = 5,\ n = 10000$$

MAPT results with $\beta = 1/2$



$D = 4$



$5 \leq D \leq 10$: TRUE model

Additional Results

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

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

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 $P_{w,\lambda}$ = mean marginal likelihood and $P_e(a_s) = P_{e,s}$ are the estimated probabilities in MMLA

(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'

Additional Results

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

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

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

for ANY pair of models T, T'

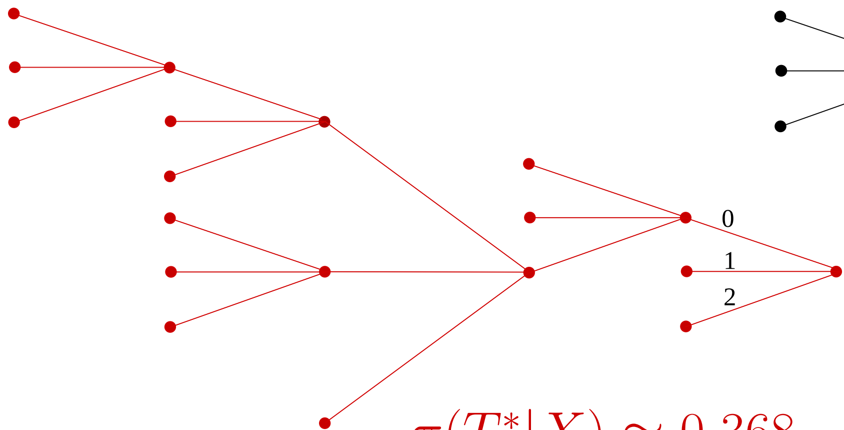
(iii) *Full conditional density of θ*

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

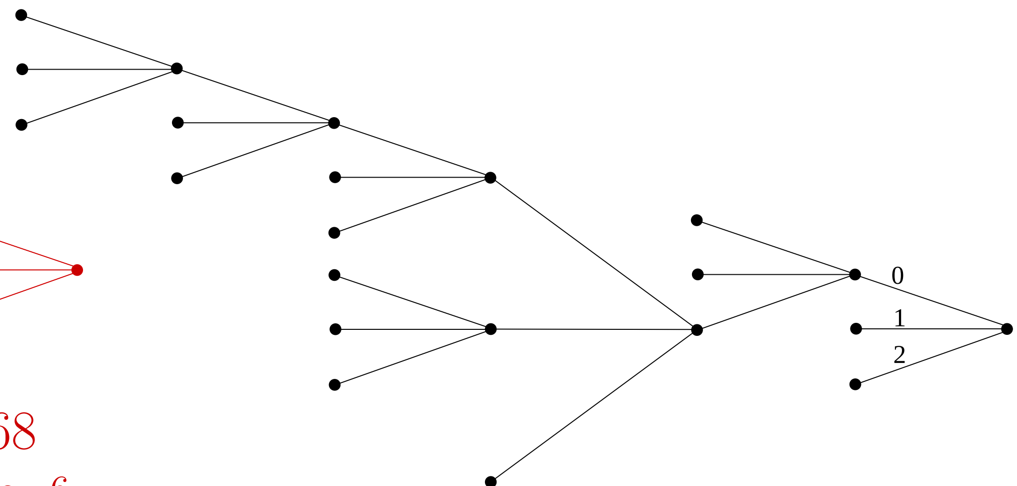
k-MAPT models for the same 5th Order Chain

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

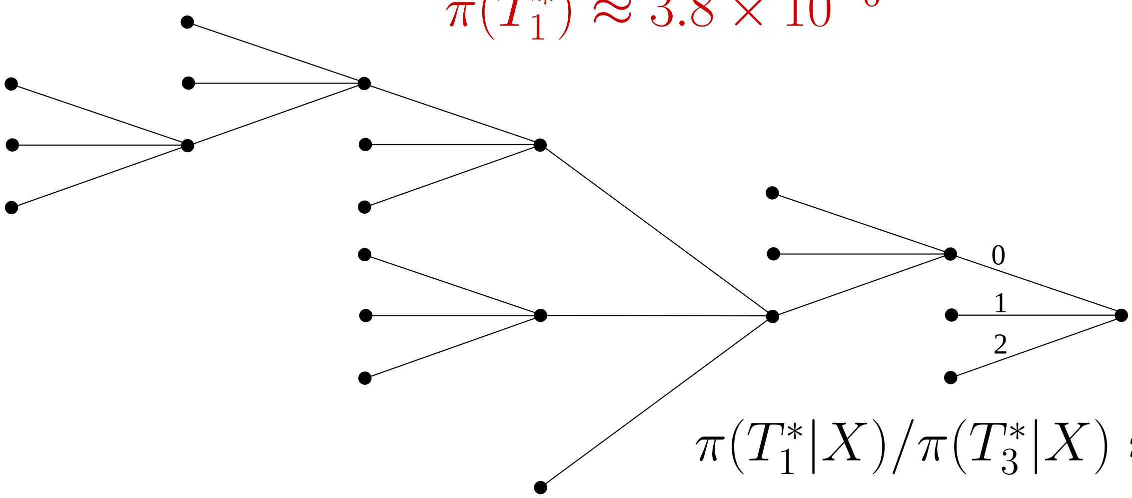
$\boldsymbol{n = 10000}$, $k = 3$, $\beta = 3/4$



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



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

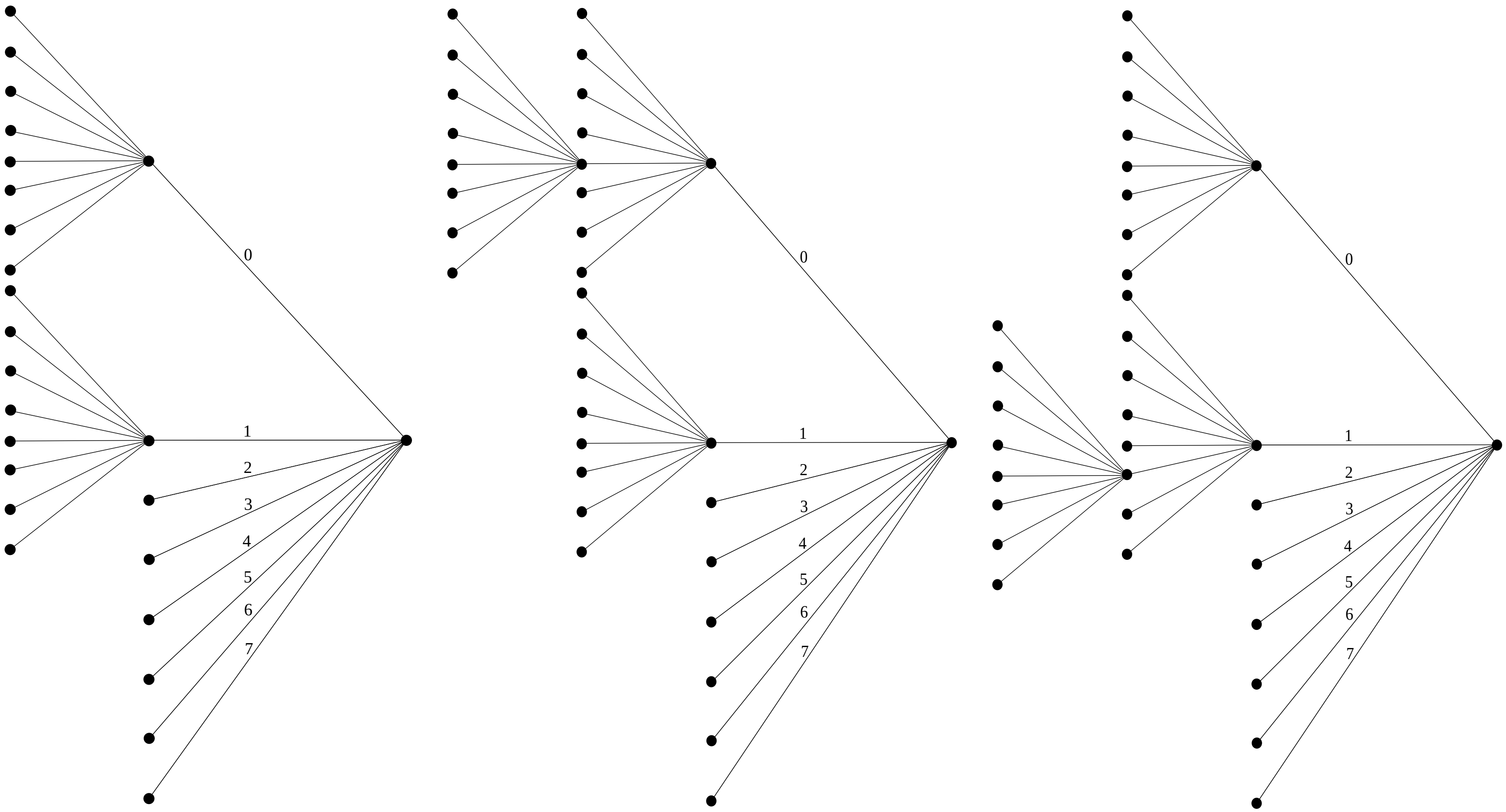


$$\pi(T_1^*|X)/\pi(T_3^*|X) \approx 8.82$$

k-MAPT for a 2nd Order, 8-Symbol Chain

2nd order VMMC: alphabet $m = 8$, memory $d = 2$, $n = 50000$ samples

k-MAPT: $k = 3$ top models, with $D = 5$, $\beta = 1 - 2^{-7}$, total $\approx 10^{1233}$ models



T_1^*: true model, $\pi(T_1^*|X) \approx 1$, $\pi(T_1^*) \approx 10^{-7}$

Metropolis-within-Gibbs Exploration of the Posterior

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

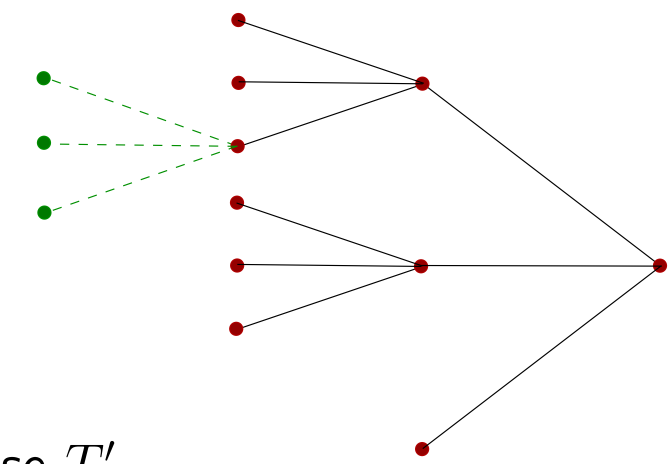
Run MAPT algorithm

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

Iterate: At each t:

Metropolis-within-Gibbs Exploration of the Posterior

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



Run MAPT algorithm

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

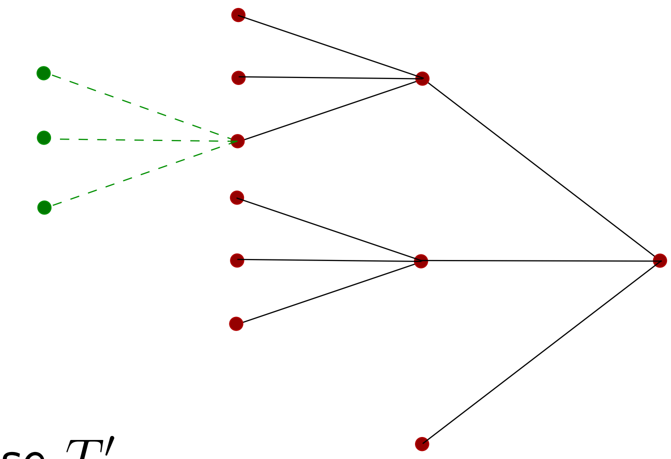
Iterate: At each t:

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

Metropolis-within-Gibbs Exploration of the Posterior

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

Parameters m, D, β



Run MAPT algorithm

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

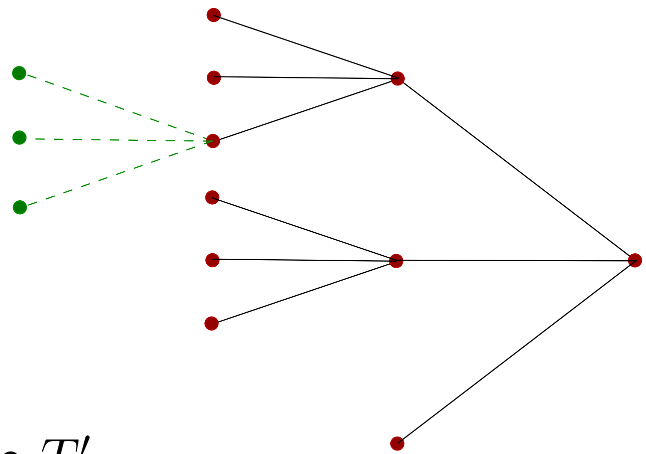
Iterate: At each t:

△ [*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'

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

Metropolis-within-Gibbs Exploration of the Posterior



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

Run MAPT 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'

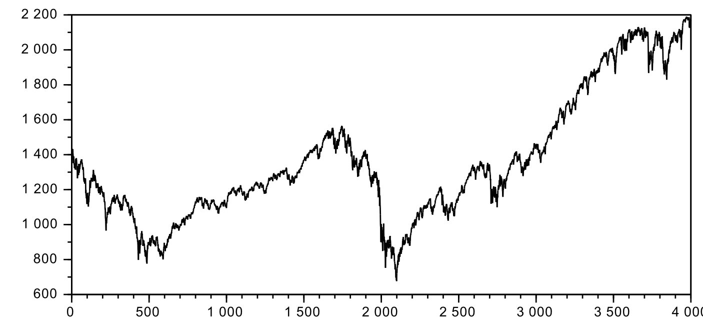
$$\frac{\pi(T'|X)}{\pi(T(t)|X)} = \frac{\pi(T')}{\pi(T(t))} \frac{\prod_{s \in T', s \notin T(t)} P_e(a_s)}{\prod_{s \in T(t), s \notin T'} P_e(a_s)}$$

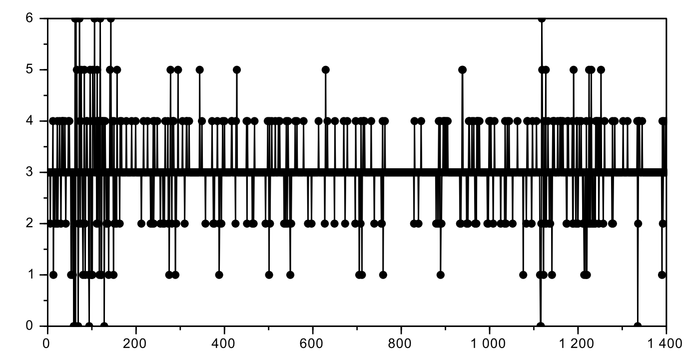
△ [*Gibbs step*] Take $\theta(t+1) \sim$ sample from the full cond'al density

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

Experimental Results: Quantized S&P 500 Data

Data Price changes X on $n = 22900$ trading days quantized to seven values

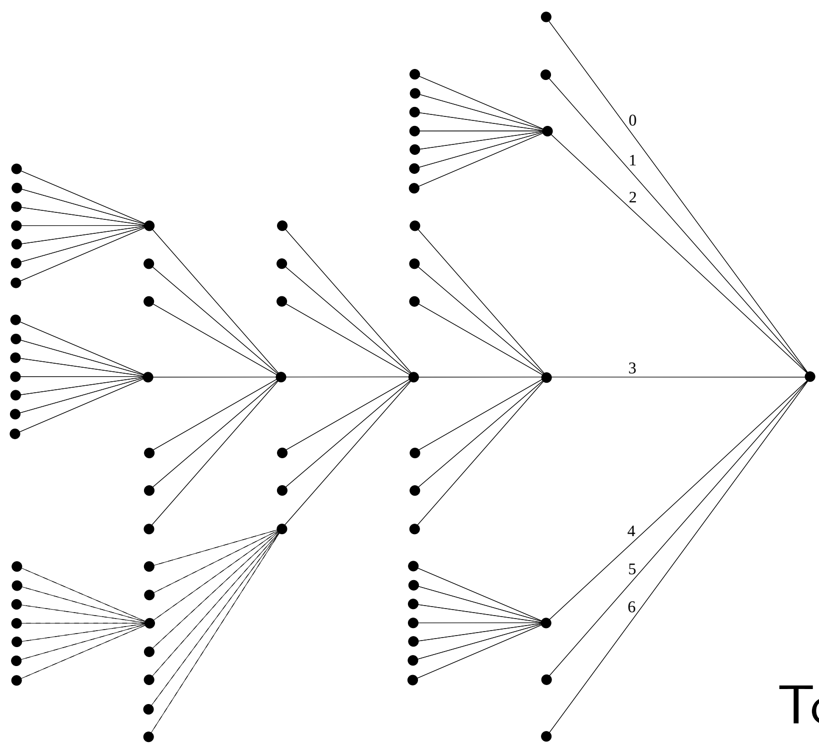


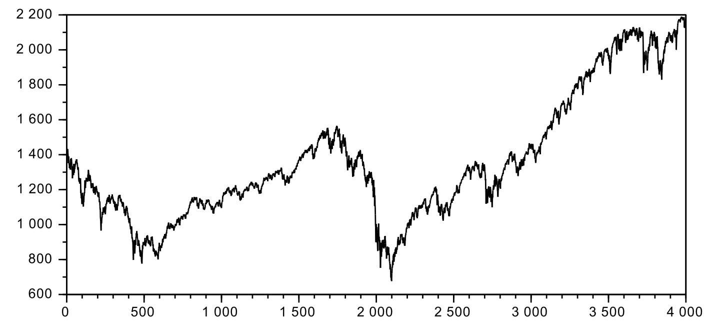


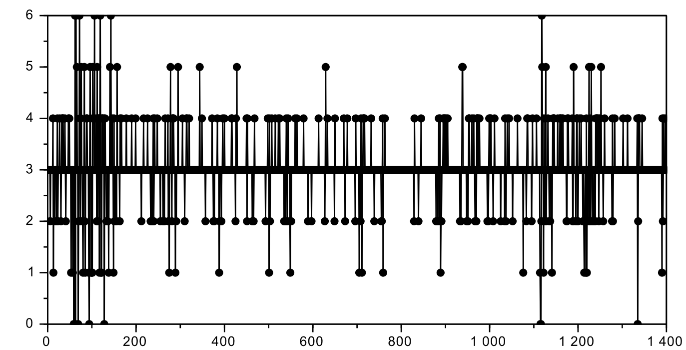
Experimental Results: Quantized S&P 500 Data

Data Price changes X on $n = 22900$ trading days quantized to seven values

k-MAPT top $k = 5$ trees with $m = 7, D = 260, \beta = 0.95$



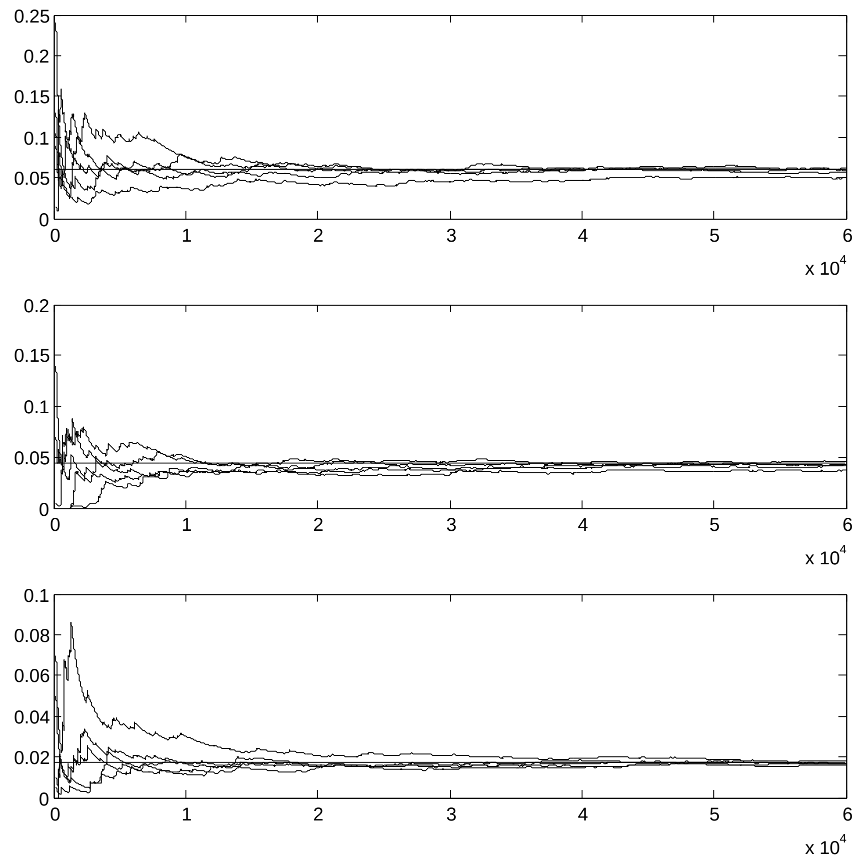




Total posterior of top 5 models $\approx 6\%$

MCMC Results on S&P 500 Data

After 10^6 iterations: Acceptance rate $\approx 45\%$, ≈ 340000 models visited



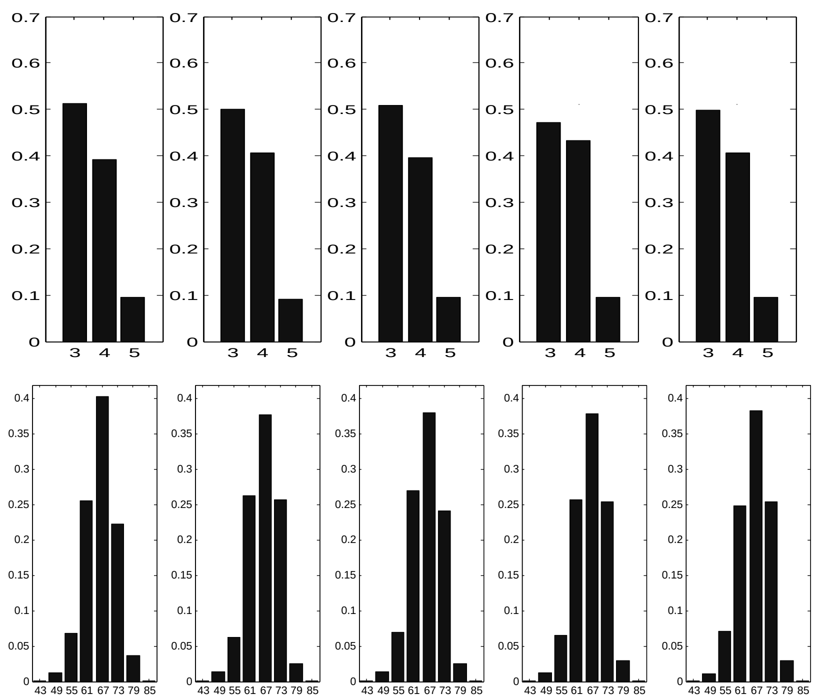
MCMC Results on S&P 500 Data

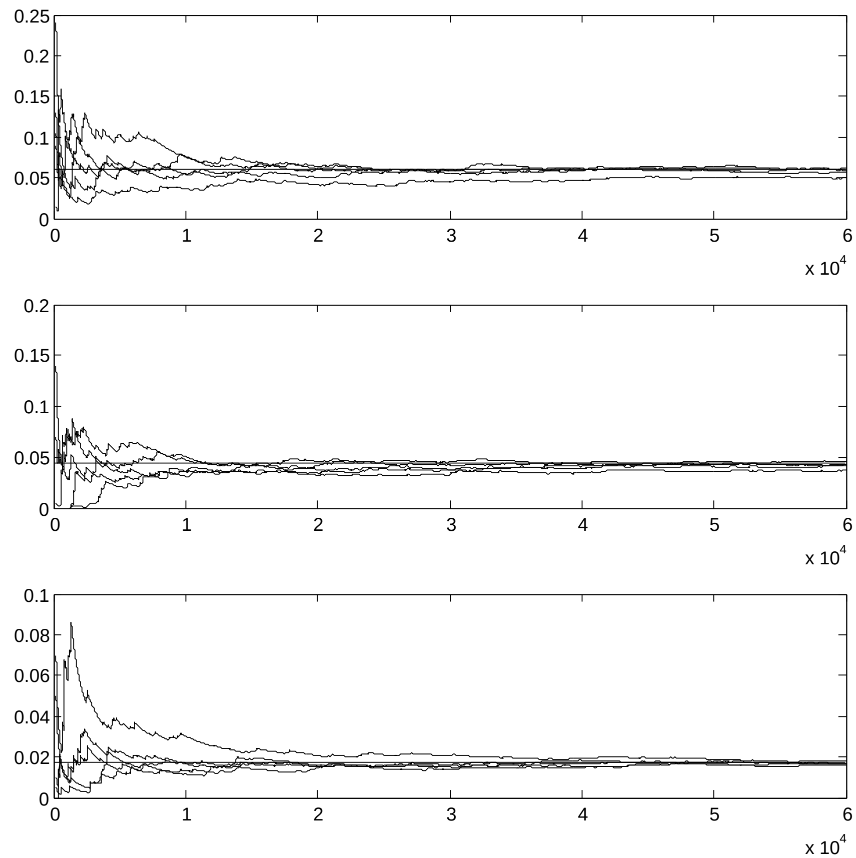
After 10^6 iterations: Acceptance rate $\approx 45\%$, ≈ 340000 models visited

Top 1000 models:

Depth 3–5, # leaves 43–85

Total posterior $\approx 25\%$





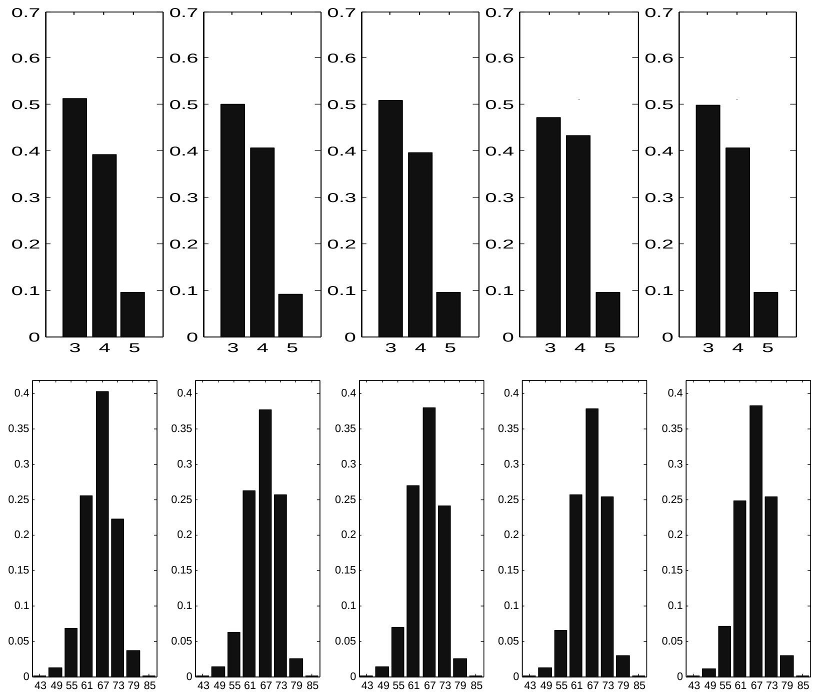
MCMC Results on S&P 500 Data

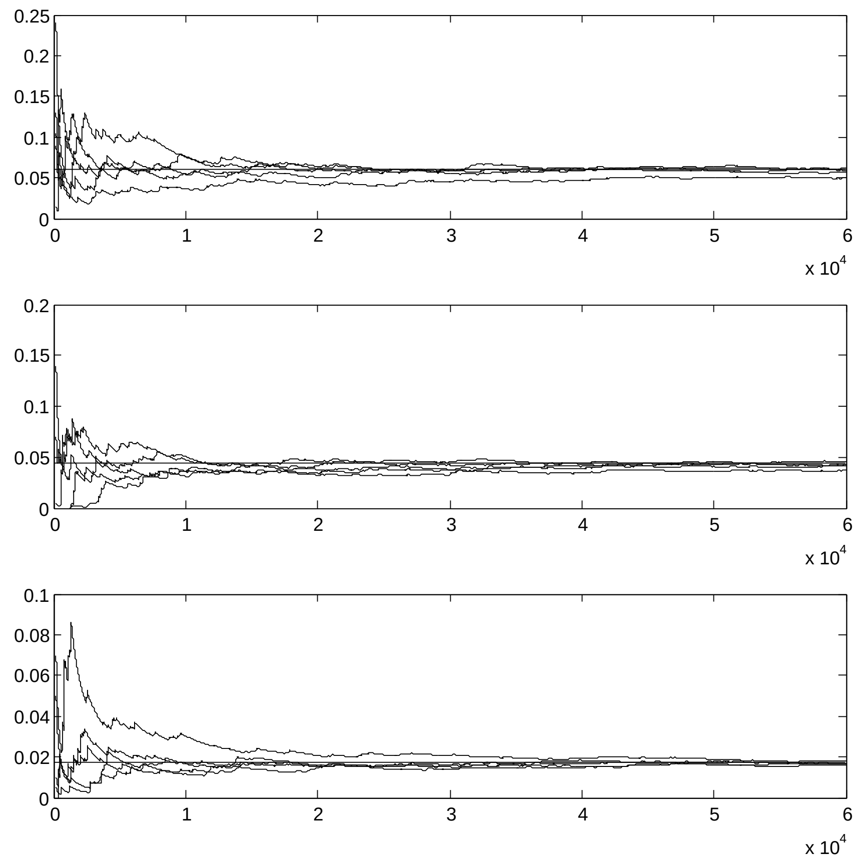
After 10^6 iterations: Acceptance rate $\approx 45\%$, ≈ 340000 models visited

Top 1000 models:

Depth 3–5, # leaves 43–85

Total posterior $\approx 25\%$





[⇝ Markov order estimation]

Outline of Theoretical Results

"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 mean marginal 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

Outline of Theoretical Results

"Theorem 1" [**BIC/MDL connection**]

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

$$\begin{aligned} f(X_{n+1}|X^n_{-D+1}) &= \sum_T \int_\theta \underbrace{f(X_{n+1}|X^n_{-D+1}, \theta, T)}_{likelihood} \; \underbrace{\pi(\theta, T|X^n_{-D+1})}_{posterior} \, d\theta \\ &= \frac{f(X_1^{n+1}|X^0_{-D+1})}{f(X_1^n|X^0_{-D+1})} \end{aligned}$$

(i) can be computed online

(ii) converges to the true conditional at the fastest possible rate

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

Outline of Theoretical Results

Theorem 3 [**Asymptotic consistency**]

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

$$\pi(\cdot,\cdot|X) \xrightarrow{\mathcal{D}} \delta_{(T^*,\theta^*)} \quad \text{a.s.}$$

Outline of Theoretical Results

Theorem 3 [**Asymptotic consistency**]

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

$$\pi(\cdot,\cdot|X) \xrightarrow{\mathcal{D}} \delta_{(T^*,\theta^*)} \quad \text{a.s.}$$

Theorem 4 [**Asymptotic normality**]

For any ergodic VMMC $\{X_n\}$ of depth no more than D and stationary distribution π, suppose $\theta^{(n)} \sim \pi(\cdot|X^n_{-D+1}, T^*)$, and let $\bar{\theta}^{(n)}$ denote its mean. Then $\bar{\theta}^{(n)} \to \theta^*$ a.s. and

$$\sqrt{n}\Big[\theta^{(n)} - \bar{\theta}^{(n)}\Big] \xrightarrow{\mathcal{D}} N(0, J) \quad \text{a.s.}$$

[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]

A Large Data Set: Spike Trains

Data Single neuron spike train in frontal eye fields (FEF) area
located in the frontal cortex (Brodmann area 8)
of the primate (monkey) brain

Study FEF-V4 coupling during attention
FEF is responsible for saccadic and voluntary eye movement
Important role in the control of visual attention

MAPT With $n \approx 10^8$ data points (ms resolution)
$m = 2$, $\beta = 1/2$ and depth $D = 130$

[MIT-NIH data: Gregoriou-Gotts-Zhou-Desimone $Science$ (2012)]

A Large Data Set: Spike Trains

Data Single neuron spike train in frontal eye fields (FEF) area

Study FEF-V4 coupling during attention

MAPT With $n \approx 10^8$ data points (ms resolution)
$m = 2$, $\beta = 1/2$ and depth $D = 130$

Resulting MAPT model

Number of leaves: $|\boldsymbol{T}| = \mathbf{1054}$
Max depth: $\boldsymbol{D} = \mathbf{130}$
Max number of 1s/context: **3** (and two contexts with 4)
Max number consecutive 1s: **2** (chemistry)
Departure from simple renewal at **30ms**

$\rightsquigarrow$ **1st/2nd order Markov renewal structure**

Extensions, Applications

$\rightsquigarrow$ **Results on real (and some "big") data**

- ▷ Satellite image data
- ▷ Genetics (DNA/RNA)
- ▷ Neuroscience
- ▷ Financial data
- ▷ Wind and rainfall measurements
- ▷ Whale/dolphin/bird song data

Applications

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