Deep Tree Models for 'Big' Biological Data

Ioannis Kontoyiannis
U of Cambridge

joint work with
Lambros Mertzanis, Athina Panotopoulou, Maria Skoularidou

19th IEEE Intern Workshop on Signal Proc Advances in Wireless Comm
Kalamata, Greece, June 2018

Co-financed by Greece and the European Union

Motivation

$\rightsquigarrow$ **Inference** for **discrete biological time series** is often **hard**

Motivation

$\rightsquigarrow$ **Inference for discrete biological time series is often hard**

$\rightsquigarrow$ **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**

Motivation

$\rightsquigarrow$ **Inference for discrete biological time series is often hard**

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

Earlier Work

$\rightsquigarrow$ **Rissanen**'s 1983-1986 fundamental work on
the **Minimum Description Length** (**MDL**) principle
and the introduction of tree/FSMX sources

$\rightsquigarrow$ The basic results of **Willems** et al 1995-2000 on data compression
via **Context Tree Weighting** (**CTW**) and related algorithms

$\rightsquigarrow$ Classical inferential procedures and asymptotics
of **Bühlmann** et al's 1999-2004
Variable-Memory Markov chains (**VLMC**)

Fixed- and Variable-Memory Markov Chain Models

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

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

Fixed- and 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 $(X_{n-1}, \ldots, X_{n-d})$

Fixed- and 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

Fixed- and 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

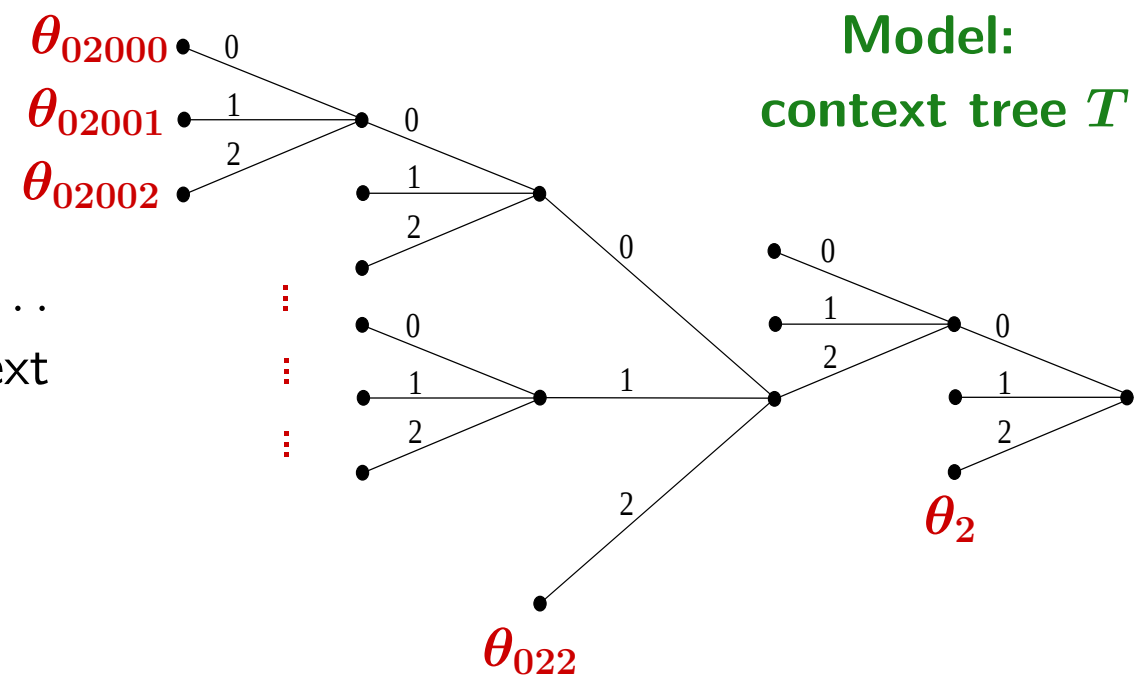
Idea Use *variable length contexts* described by a **context tree** $\boldsymbol{T}$

Variable-Memory Markov Chains: An Example

Alphabet $m = 3$ symbols

Memory length $d = 5$

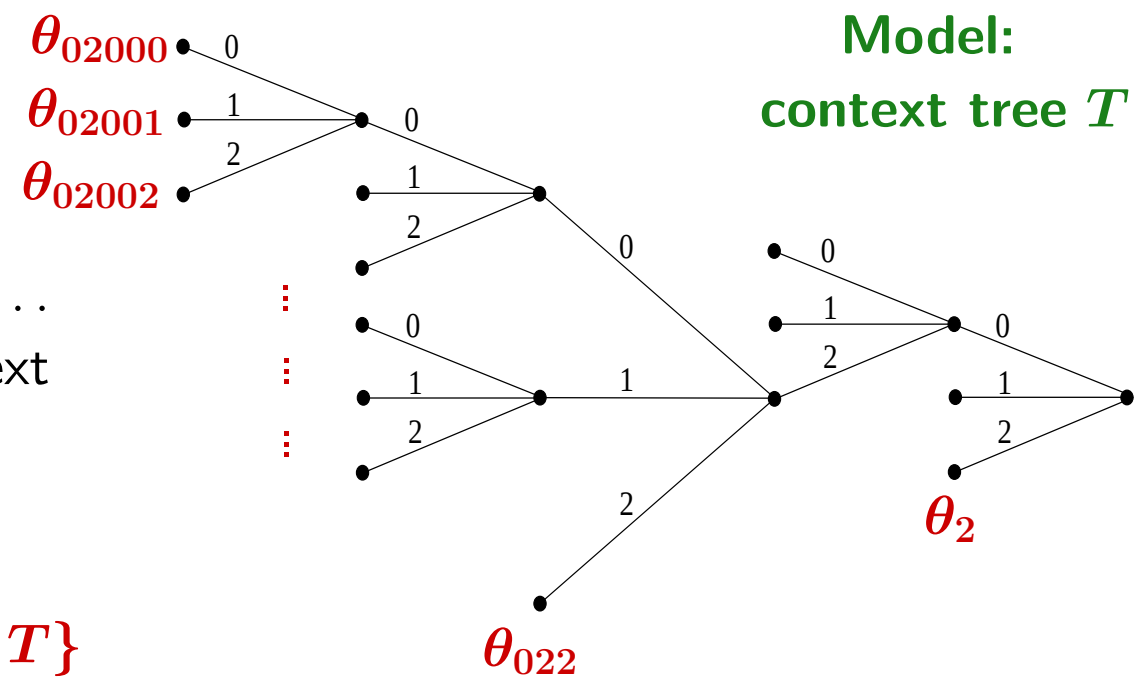
Each past string $X_{n-1}, X_{n-2}, \ldots$ corresponds to a unique context on a leaf of the tree



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

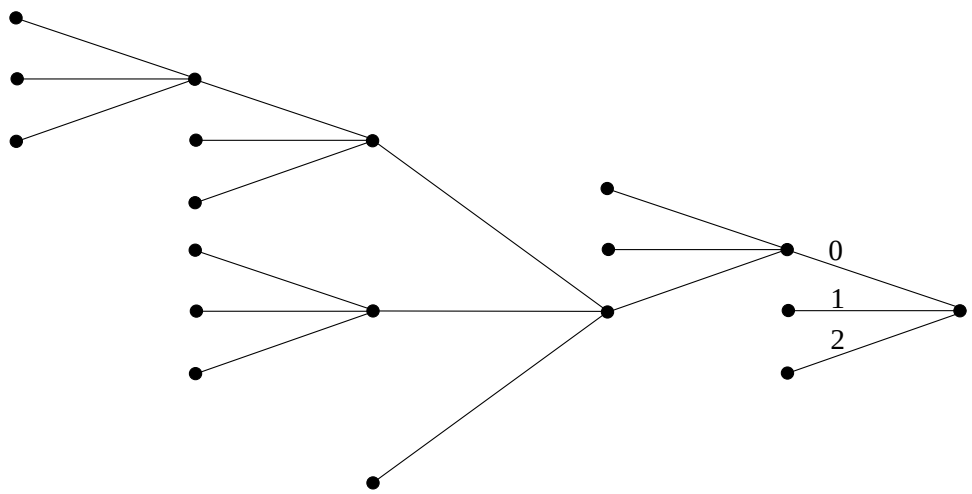
Variable-Memory Representation: Advantages

↝ **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

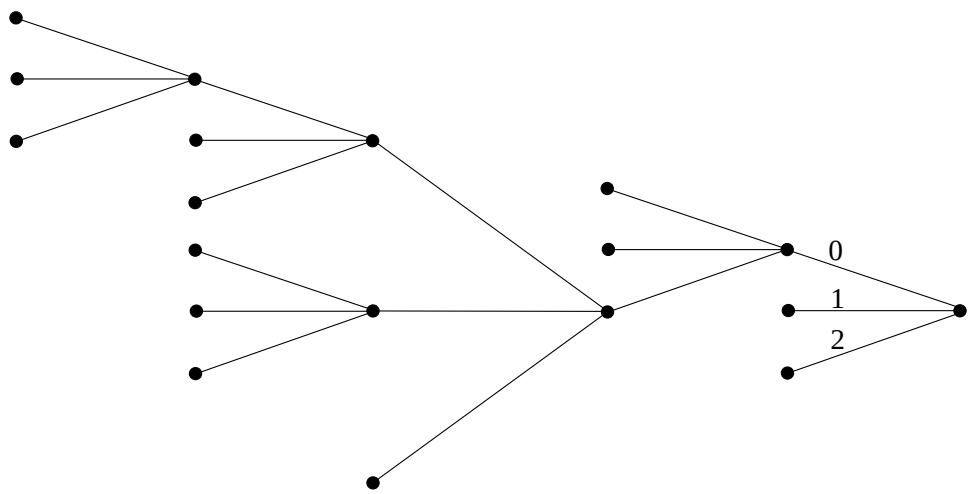


Variable-Memory Representation: Advantages

⇝ **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



Applications

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

Bayesian Modelling of VMMCs

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

Bayesian Modelling of VMMCs

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

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

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

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

[**Lemma**: This is OK]

Bayesian Modelling of VMMCs

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

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

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

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 Inference of VMMCs

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

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

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 Inference of VMMCs

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

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

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

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

Model selection goal: The **posterior distribution**

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

Bayesian Inference of VMMCs

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

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

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

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

Model selection goal: The **posterior distribution**

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

Main obstacle: 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

Maximum A Posteriori Probability Tree Algorithm (MAPT)

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

Maximum A Posteriori Probability Tree Algorithm (MAPT)

△ **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 count vectors a_s and the probabilities

$$\boldsymbol{P_{e,s}} = \frac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$$

where $M_s = a_s(0) + \cdots + a_s(m-1)$

Maximum A Posteriori Probability Tree Algorithm (MAPT)

△ **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 count vectors a_s and the probabilities

$$\boldsymbol{P_{e,s}} = \frac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$$

where $M_s = a_s(0) + \cdots + a_s(m-1)$

△ **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)

△ **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 count vectors a_s and the probabilities

$$\boldsymbol{P_{e,s}} = \frac{\prod_{j=0}^{m-1}[(1/2)(3/2)\cdots(a_s(j)-1/2)]}{(m/2)(m/2+1)\cdots(m/2+M_s-1)}$$

where $M_s = a_s(0) + \cdots + a_s(m-1)$

△ **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

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

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

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

(ii) *Mean marginal likelihood* $f(X)$ computed like the MAP but with averages instead of maxima

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

(ii) *Mean marginal likelihood* $f(X)$ computed like the MAP but with averages instead of maxima

(iii) *Model posterior probabilities* $\pi(T|X) = \dfrac{\pi(T)\prod_{s\in T} P_{e,s}}{f(X)}$

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

(ii) *Mean marginal likelihood* $f(X)$ computed like the MAP but with averages instead of maxima

(iii) *Model posterior probabilities*
$$\pi(T|X) = \frac{\pi(T)\prod_{s\in T} P_{e,s}}{f(X)}$$

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

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

(ii) *Mean marginal likelihood* $f(X)$ computed like the MAP but with averages instead of maxima

(iii) *Model posterior probabilities*
$$\pi(T|X) = \frac{\pi(T)\prod_{s\in T} P_{e,s}}{f(X)}$$

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

(v) *Full conditional density of θ*
$$\pi(\theta|T,X) \sim \prod_{s\in T} \text{Dirichlet}(a_s(0)+1/2, a_s(1)+1/2, \ldots, a_s(m-1)+1/2)$$

Additional Results

(i) *Top k MAP models* $T_1^*, T_2^*, \ldots, T_k^*$

(ii) *Mean marginal likelihood* $f(X)$ computed like the MAP but with averages instead of maxima

(iii) *Model posterior probabilities* $$\pi(T|X) = \frac{\pi(T)\prod_{s\in T} P_{e,s}}{f(X)}$$

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

(v) *Full conditional density of θ*

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

(vi) *MCMC exploration of the posterior*

Metropolis-within-Gibbs sampling from $\pi(\theta, T|X)$ using (iv) and (v)

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**

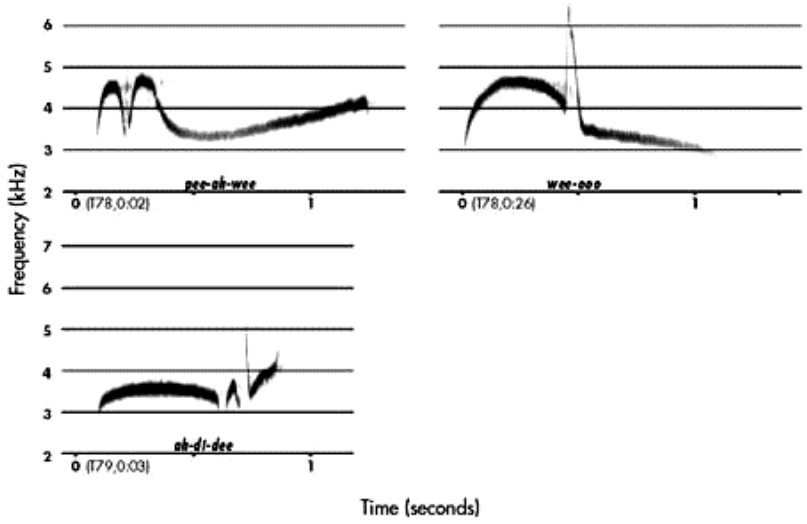
$\rightsquigarrow$ **1st/2nd order Markov renewal structure**

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"]

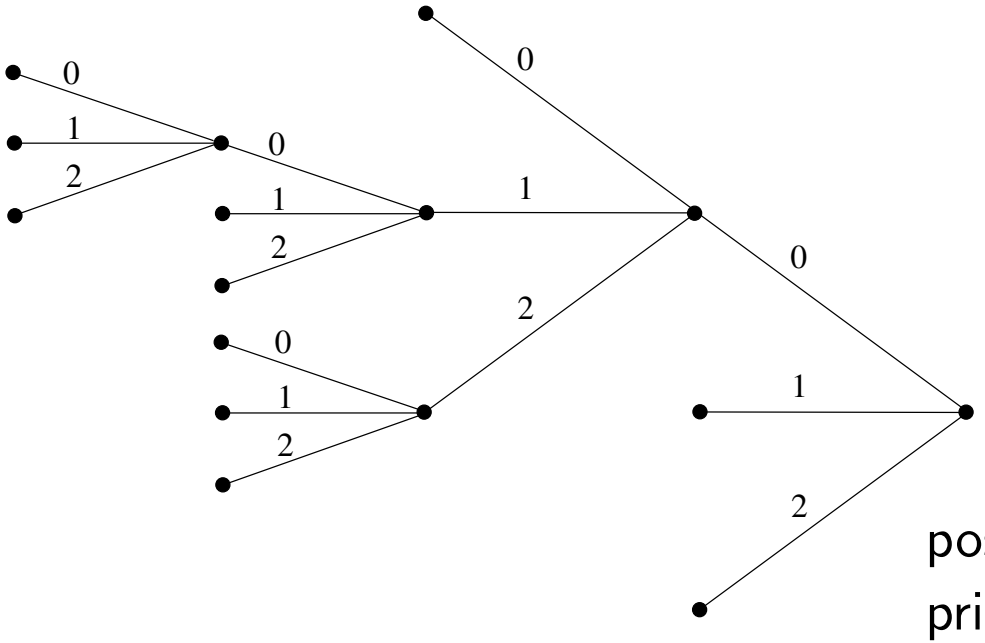


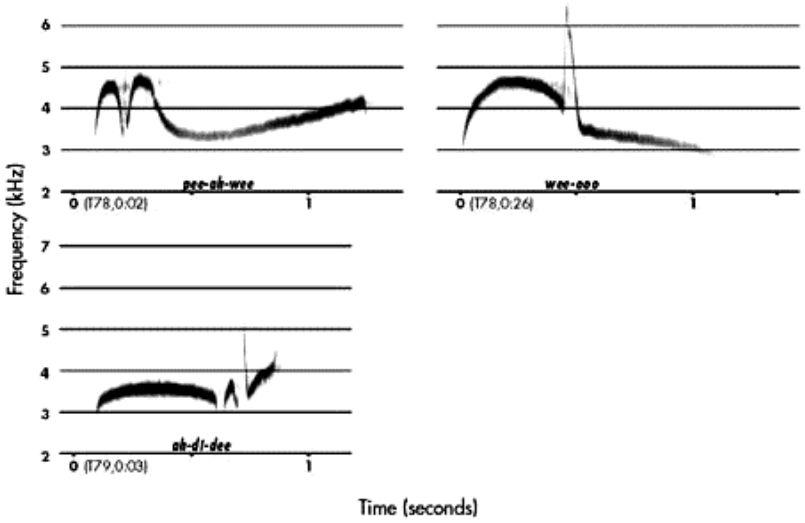
A Fun Data Set: Wood Peewee Bird Song

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"]

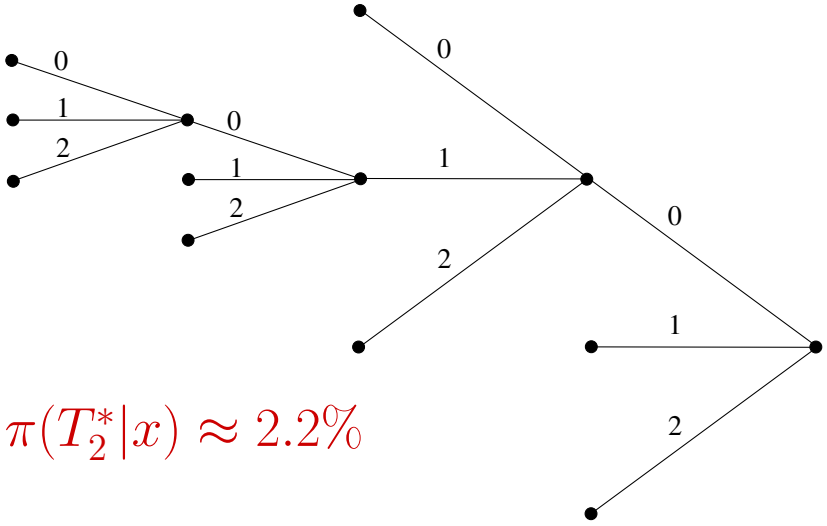
MAPT With $n = 1327$ samples
$m = 3$, $\beta = 3/4$ and depth $D = 10$



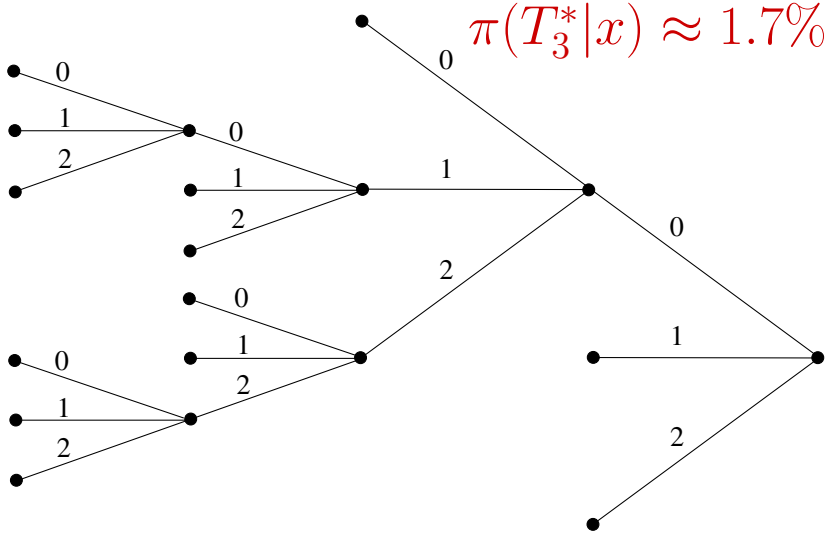


posterior: $\pi(T_1^*|x) \approx 12.4\%$
prior: $\pi(T_1^*) \approx 3 \times 10^{-4}$

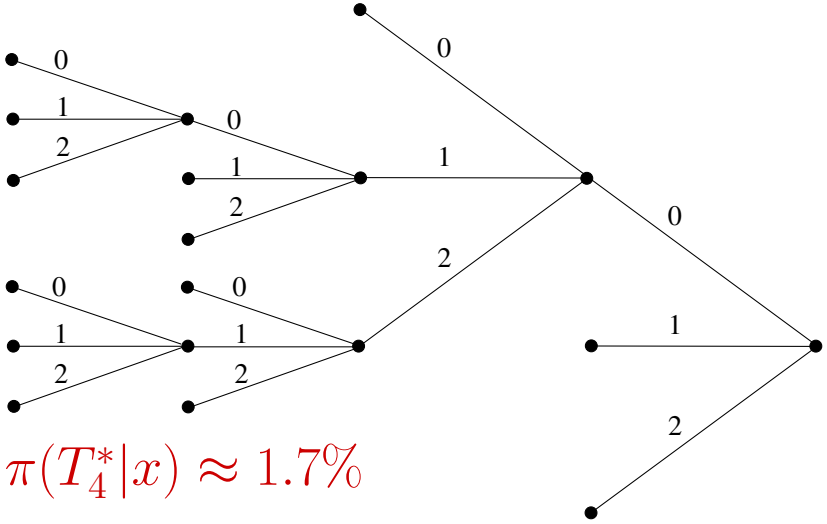
Wood Peewee Bird Song: Next 4 Models



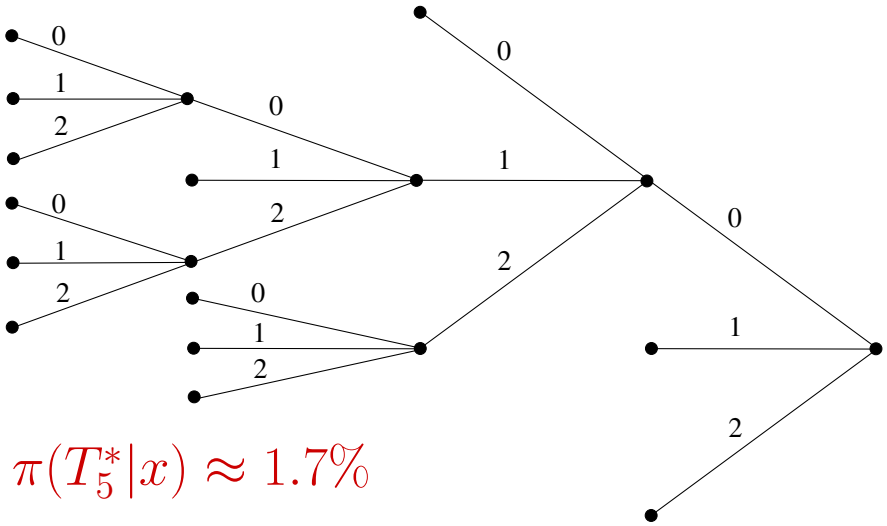
$\pi(T_2^*|x) \approx 2.2\%$



$\pi(T_3^*|x) \approx 1.7\%$



$\pi(T_4^*|x) \approx 1.7\%$



$\pi(T_5^*|x) \approx 1.7\%$

Bird Song Models: Comparison with Other Methods

	MAPT	VLMC	MTD	gMTD
result	T_1^*, $d = 5$	complex tree, $d = 18$	complete, $d = 10$	complete, $d = 2$
AIC	**687.4**	796.8	1102.1	966.8
BIC	**801.4**	1273.6	1143.5	1003.0

Bird Song Models: Comparison with Other Methods

	MAPT	VLMC	MTD	gMTD
result	T_1^*, $d = 5$	complex tree, $d = 18$	complete, $d = 10$	complete, $d = 2$
AIC	**687.4**	796.8	1102.1	966.8
BIC	**801.4**	1273.6	1143.5	1003.0

Our Bayesian framework gives

- △ interesting and *interpretable* results
- △ good models by any metric
- △ a quantitative measure of accuracy
- △ allows for more applications

Bird Song Models: Comparison with Other Methods

	MAPT	VLMC	MTD	gMTD
result	T_1^*, $d = 5$	complex tree, $d = 18$	complete, $d = 10$	complete, $d = 2$
AIC	**687.4**	796.8	1102.1	966.8
BIC	**801.4**	1273.6	1143.5	1003.0

Our Bayesian framework gives

- △ interesting and *interpretable* results
- △ good models by any metric
- △ a quantitative measure of accuracy
- △ allows for more applications
- △ rich model-selection information via k-MAPT and MCMC

  E.g., in 10^6 steps, with an acceptance rate of ≈ 0.575

  we visit 269562 different models

  The 100 most visited trees have 9-17 leaves and depths $4 \leq d \leq 6$

Extensions, Applications

$\rightsquigarrow$ **Results on empirical (including some "big") data**

- ▷ Genetics (DNA/RNA)
- ▷ Proteins and cross-omics data
- ▷ Neuroscience
- ▷ 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