Context Tree Weighting for Signal Processing, Bayesian Inference and Model Selection: Theory and Algorithms

<u>Name:</u> Skoularidou Maria, I.D.: 61321

<u>Submission Date:</u> 31 July 2015, Friday

<u>Advisor:</u> Professor Dellaportas P.

Contents

1 Introduction **5**

1.1 Overview 5

1.2 Stating the problem 5

1.3 Prior Distributions 8

1.3.1 Context Trees' Prior Distribution 8

1.3.2 θ's Prior Distribution 10

2 Mean Marginal Likelihood Algorithm (a.k.a. CTW) **11**

2.1 Preliminaries 11

2.2 The Algorithm 13

2.3 The Corresponding Theorem 14

3 Maximum A Posteriori Probability Tree Algorithm (a.k.a. CTM) **17**

3.1 The Algorithm 17

3.2 The Corresponding Theorem 18

4 Two illustrative examples **23**

4.1 Second Order Binary Markov Chain 23

4.2 Third Order Ternary Markov Chain 27

5 Simulations **33**

5.1 A Toy Example 33

5.2 Binary Renewal Process 34

5.3 A Ternary, Fifth Order Markov Chain 35

5.4 An Octal, Second Order Markov Chain 44

5.5 A Binary, Tenth Order Markov Chain 47

5.6 A Binary Hidden Markov Model 50

5.7 A More Complex Hidden Markov Model 51
5.8 A Ternary, Fifth Order Markov Chain, with noise 52
5.9 Noisy Markovian Samples 54
5.10 Remarks 59

6 Appendix 61
6.1 Model N^o1 61
6.2 Model N^o2 62
6.3 Model N^o3 64
6.4 Model N^o4 66
6.5 Model N^o5 69
6.6 Model N^o6 71
6.7 Model N^o7 73
6.8 Model N^o8 76
6.9 Model N^o9 78

Περίληψη

Σκοπος της παρούσας εργασίας είναι να ερευνήσει, να επεκτείνει και να χρησιμοποιήσει μια οικογένεια αλγορίθμων που εδραιώθηκαν τα τελευταία είκοσι χρόνια στη Θεωρία Πληροφορίας, υπό την σκέπη της μεθόδου 'Context Tree Weighting'. Παρόλο που αυτοί οι αλγόριθμοι αρχικά εμπνεύστηκαν και εφαρμόστηκαν σε προβλήματα κωδικοποίησης και συμπίεσης δεδομένων, υποστηρίζουμε οτι το πεδίο εφαρμογών τους εκτείνεται σε ευρύ φάσμα προβλημάτων στατιστικής συμπερασματολογίας και επεξεργασίας σήματος. Θα εξετάσουμε τον αλγόριθμο εύρεσης της εκ των υστέρων μεγαλύτερης πιθανότητας δέντρου (Maximum A Posteriori Probability Tree Algorithm – MAPT) ως μια αποδοτική μέθοδο συμπερασματολογίας κατά Bayes, στο πλαίσιο δεδομένων διακριτών χρονολογικών σειρών. Ο εν λόγω αλγόριθμος υπολογίζει το εκ των υστέρων πιθανότερο (δενδρικό) μοντέλο καθώς και την πιθανότητα που του αντιστοιχεί. Παράλληλαα, παρουσιάζουμε σχετικά πειραματικάῶά αποτελέσματα τόσο σε ανεξάρτητα δεδομένα όσο και σε πιο σύνθετα που έχουν παραχθεί από αλυσίδες Markov μεταβλητού μήκους, στα οποία φανερώνεται η απόδοση του αλγορίθμου.

Abstract

The goal of the present thesis is to explore, extend and utilize a family of algorithms that arose over the past twenty years in the Information Theory literature, under the umbrella of “Context Tree Weighting”. Although these methods were originally motivated by and applied to problems in source coding and data compression, we argue that there range of applicability extends to a large variety of problems in statistical inference and signal processing. We will examine the Maximum A Posteriori Probability Tree Algorithm (MAPT) as an efficient method for Bayesian inference, in the context of discrete series data. The MAPT algorithm computes the maximum a posteriori probability tree model, as well as the corresponding model posterior probability. Experimental results will be given, illustrating its performance, both on independent data and on more complex signals generated by variable memory Markov chains.

Acknowledgments

First, I would like to acknowledge my sincere gratitude to my advisor Professor Petros Dellaportas. Being his student has influenced me deeply. I am indebted for mentoring me in such a kind way and for teaching me how to look at the essence of things.

I also want to thank Professor Ioannis Kontogiannis for all the support and generosity he provided through this procedure. This thesis, part of which belongs to our on-going research, was totally inspired by him.

Finally, I would like to thank Athina Panotopoulou and Aristeidis Panos for our both intellectual and enjoyable collaboration. Part of the code that has been used in this thesis has been developed by Athina.

Chapter 1

Introduction

1.1 Overview

In this thesis we shall examine a class of Markov chains, the so-called Variable Length Markov Chains (V.L.M.C.) which is a significant class of tree structured models for stationary discrete time series. Examples of such time series include DNA sequence data or binary sequences, for example from information or computing technology. Specifically we wish to utilize *"Context Tree Weighting"* (CTW) a widely used method for finite memory tree sources in which one can make use of a context tree which contains for each string (context) the number of each symbol that have followed this context in a source sequence. In this framework, given the past source symbols, one can use this context tree to estimate the actual "state" of the finite memory tree source. Subsequently, this state is used to estimate the distribution that generates the next source symbol.

1.2 Stating the problem

In a series of papers, namely [5],[6],[7],[8],[9],[10],[11],[12],[13],[14] F.J. Willems *et. al* described in detail a new approach of the Context Tree Weighting both for binary ([5], [6], [7], [9], [10], [11], [12], [13], [14]) and multi-alphabet ([8]) sources from the universal coding point of view. A few years later, Paul Wolf examined and further extended the compression ability of the CTW [15] mostly in the text compression area. Here, we will focus on the Bayesian inference extension of this specific method. To do so, we need to state the framework in which we will introduce our perspective.

Throughout this thesis, we deal with dth-order homogeneous Markov chains with values in the finite alphabet $\mathcal{A} = \{0, 1, \cdots, m-1\}$ (alphabet size $m \geq 2$ and maximum depth d, $d \in \mathbb{N} \setminus 0$ are fixed). Specifically, for the process $\{X_n\}$ we define the conditional distribution of each X_i, $i \in \mathbb{N} \setminus 0$ given the previous d symbols $(X_{i-d}, X_{i-d+1} \cdots, X_{i-1})$ where we denote by ${X_i}^j$ any vector of random variables $(X_i, X_{i+1}, \cdots, X_{j-1}, X_j)$, $i \leq j$ and similarly

$x_i^j \in \mathcal{A}^{j-i+1}$ for a string $(x_i, x_{i+1}, \cdots, x_{j-1}, x_j)$ representing a realization of the random variables X_j^i. The key element in specifying these distributions is the *context function* $C : A^d \to T$, which maps each length-d context x_{i-d}^{i-1} to a (typically strictly) shorter suffix $C(x_{i-d}^{i-1}) = x_{i-j}^{i-1}$ of itself, for some $0 \leq j \leq d$. Then the Markov property for $\{X_n\}$ takes the form:

$$P(x_1^n | x_{d-1}^0) = \prod_{i=1}^{n} P(x_i | x_{i-1}^{i-d}) = \prod_{i=1}^{n} P(x_i | C(x_{i-d}^{i-1})) \tag{1.1}$$

The range T of C is a subset of $\cup_{i=0}^{d} \mathcal{A}^i$ where we adopt the convention that the set $\mathcal{A}^0$ contains only the empty string λ. We assume that the set T is *proper*. Moreover, strings will be considered as concatenations of m-ary symbols, $m \geq 2$, where m equals the size of the alphabet $\mathcal{A}$, hence, $s = x_{i-d+1}x_{i-d+2}\cdots x_0$ with $x_i \in \mathcal{A} = \{0, 1, \cdots, m-1\}$ and $i = 0, 1, \cdots, d-1$. Note that we index the symbols in the string from right to left starting with zero and going negative. If we have two strings $s = x_{i-d}x_{i-d+1}\cdots x_0$ and $s' = x'_{i-d'+1}x'_{i-d'+2}\cdots x'_0$ then $s's = x'_{i-d'+1}x'_{i-d'+2}\cdots x'_0 x_{i-d+1}x_{i-d+2}\cdots x_0$ is the concatenation of both. If $\mathcal{S}$ is a set of strings, then $\mathcal{S} \times x \triangleq \{sx : s \in \mathcal{S}\}$ for $x \in \mathcal{A} = \{0, 1, \cdots, m-1\}$.

We say that a string $s = x_{i-d+1}x_{i-d+2}\cdots x_0,\ x_i \in \mathcal{A} = \{0, 1, \cdots, m-1\},\ i = 0, 1, \cdots, d-1$ is a *suffix* of a string $s' = x'_{i-d'+1}x'_{i-d'+2}\cdots x'_0$ if $d \leq d'$ and $x_{-i} = x_{-i'}$ for $i = 0, \cdots, d-1$. Observe that, under these assumptions, the context function C is completely determined by its range T, since, for any string x_{i-d}^{i-1} there is exactly one element of T which is a suffix x_{i-j}^{i-1} of x_{i-d}^{i-1}.

To complete the specification of the (conditional) distribution of the process $\{X_n\}$, in addition to the context set T, with every element $s \in T$ we associate a probability vector $\theta_s = (\theta_s(0), \theta_s(1), \cdots, \theta_s(m-1))$, where the $\theta_s(j)$ are nonnegative and sum to one, $\sum_{i=0}^{m-1} \theta_s(i) = 1$. Then, the probability $P(x_1^n | x_{-d+1}^0)$ is,

$$P(x_1^n|x_{-d+1}^0) = \prod_{i=1}^{n} P(x_i|x_{i-1}^{i-d}) = \prod_{i=1}^{n} P(x_i|C(x_{i-d}^{i-1})) = \prod_{i=1}^{n} \theta_{C(x_{i-d}^{i-1})}(x_i) \quad (1.2)$$

Note that, instead of taking the product sequentially in time, we can take a product over all possible contexts $s \in T$ and express this probability as,

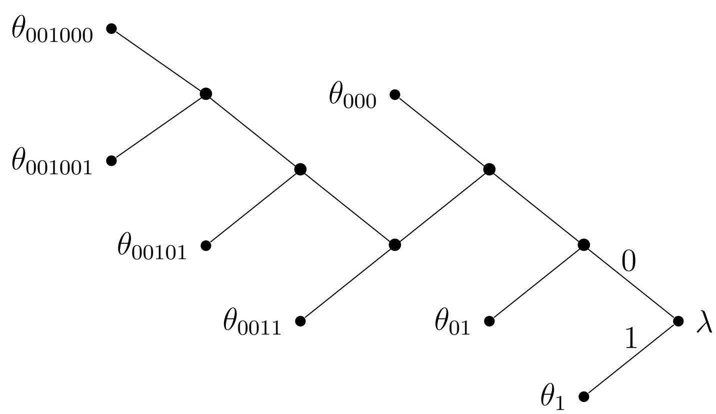
$$P(x_1^n|x_{-d+1}^0) = \prod_{s\in T}\prod_{j\in\mathcal{A}} \theta_s(j)(\alpha_s(j)) \quad (1.3)$$

where each element $\alpha_s(j)$ of the vector $\alpha_s = (\alpha_s(0), \alpha_s(1), \cdots, \alpha_s(m-1))$ represents the amount of times symbol $j \in \mathcal{A}$ follows context s in x_1^n.

To summarize, the (conditional) distribution of the Markov chain $\{X_n\}$ is described by a *proper* context set T, and by a collection $\theta = \{\theta_s : s \in T\}$ of probability distributions $\theta_s = (\theta_s(0), \theta_s(1), \cdots, \theta_s(m-1))$ for each element of the context set T.

The distribution of $\{X_n\}$ is determined as in (1.2), once we have specified a (proper) context set T – the *model* – and a collection $\theta = \{\theta_s : s \in T\}$ of probability vectors θ_s, $\forall s \in T$ – the *parameters*. Note that the context set T can be described as a tree. Therefore, we will refer to models T as context trees, context sets, or simply as models, interchangeably. In the tree representation, the context corresponding to the empty string λ is the root of the tree.

Now, we will illustrate an example to clarify the procedure. Consider a 6th order binary Markov chain, defined by the context tree T shown below, and by a collection of (known) parameters $\theta = \{\theta_s : s \in T\}$, where θ_s is a probability vector corresponding to leaf s in T



Then we are able to compute the likelihood of any arbitrary string via (1) or (2). For example, with d=6 and n = 15, the string,

$$\underbrace{0,0,1,0,1,0,}_{x_{-5}^{0}}\underbrace{0,0,1,1,0,0,0,0,1,0,1,0,0,1,0}_{x_1^{15}}$$

has probability given by (1),

$$\theta_{01}(0), \theta_{00101}(0), \theta_{000}(1), \theta_1(1), \theta_1(0), \theta_{01}(0), \theta_{0011}(0), \theta_{000}(0),$$

$$\theta_{000}(1), \theta_1(0), \theta_{01}(1), \theta_1(0), \theta_{01}(0), \theta_{00101}(1), \theta_1(0)$$

The corresponding count vectors are,

$$\alpha_1 = (4,1),\ \alpha_{01} = (3,1),\ \alpha_{000} = (1,2),\ \alpha_{0011} = (1,0),\ \alpha_{00101} = (1,1)$$

with all other s having all-zero count vectors α_s. The probability of $x_1^1 5$ given x_{-5}^0 as expressed in (2) is:

$$\theta_{01}(0)^3 \cdot \theta_{00101}(0) \cdot \theta_{000}(1)^2 \cdot \theta_1(1) \cdot \theta_1(0)^4 \cdot \theta_{0011}(0) \cdot \theta_{000}(0) \cdot \theta_{01}(1) \cdot \theta_{00101}(0)$$

1.3 Prior Distributions

First, we need to introduce some appropriate prior distributions based at which we will proceed with our main focus.

1.3.1 Context Trees' Prior Distribution

Given a fixed depth D and an arbitrary $\beta \in [0.5, 1)$, we define a prior distribution on models T of maximal depth $d, d \leq D$, as,

$$\pi(T) = \pi_D(T) = \alpha^{|T|-1} \beta^{|T|-L_D(T)} \tag{1.4}$$

where $|T|$ denotes the amount of T's leaves, $L_D(T)$ denotes the amount of T's leaves at maximum depth (i.e. at depth D) and

$$\alpha^{m-1} + \beta = 1 \ \Leftrightarrow \alpha = (1-\beta)^{\frac{1}{m-1}}$$

The following lemma states that $\pi(T)$ defines indeed a probability distribution

Lemma 1.3.1 *For all d, $d \leq D$ and $\beta \in [0.5, 1)$*

$$\sum_{T \in T(D)} \pi_D(T) = 1 \tag{1.5}$$

where the sum is over the all possible proper context trees T of depth no greater than D in the collection $T(D)$.

Proof The proof is by induction. Note first that for D = 0, 1 it is trivial to see that the result holds as:

1. D=0:

$$\pi_D(\lambda) = \alpha^{1-1}\ \beta^{1-1} = 1$$

2. D=1:

   - $$\pi_D(\lambda) = \alpha^{1-1}\ \beta^{1-0} = \ \beta$$

   - $$\pi_D(T) = \alpha^{m-1}\ \beta^{m-m} = \ \alpha^{m-1}$$

   but $\alpha^{m-1} + \beta = 1$

Also observe that we can write any tree T which does not contain only the root node λ, as the union $T = \cup_j T_j$ of a collection of m subtrees $T_0, T_1, \cdots, T_{m-1}$. Clearly we will then have,

$$|T| = \sum_{j=0}^{m-1} |T_j|, \text{ and } \ L_D(T) = \sum_{j=0}^{m-1} L_{D-1} T_j \tag{1.6}$$

For the inductive step, suppose that the result holds for all depths less than or equal to some $d \leq D - 1$, i.e. $\sum_{T \in T(d)} \pi_d(T) = 1$ holds for all $d \leq D - 1$. We will show that it holds for d+1 as well. Let Λ denote the tree that consists only of the root node λ. Using (1.6), we have,

$$\begin{aligned}
\sum_{T\in T(d+1)} \pi_{d+1}(T) &= \pi_{d+1}(\Lambda) + \sum_{T\in T(d+1), T\neq\Lambda} \alpha^{|T|-1}\beta^{|T|-L_{d+1}(T)} \\
&= \beta + \sum_{T_0,T_1,\ldots,T_{m-1}\in T(d)} \alpha^{\sum_{j=0}^{m-1}|T_j|-1}\ \beta^{\sum_{j=0}^{m-1}|T_j|-\sum_{j=0}^{m-1}L_d(T_j)} \\
&= \beta + \alpha^{m-1} \sum_{T_0,T_1,\ldots,T_{m-1}\in T(d)} \prod_{j=0}^{m-1} \alpha^{|T_j|-1}\ \beta^{|T_j|-L_d(T_j)} \\
&= \beta + \alpha^{m-1} \prod_{j=0}^{m-1} \sum_{T_j\in T(d)} \pi_d(T_j) \\
&= \beta + \alpha^{m-1} = 1 \qquad \blacksquare
\end{aligned}$$

1.3.2 θ's Prior Distribution

Given a model T we define a prior distribution on the probability vectors $\theta = \{\theta_s : s \in T\}$ on the leaves s of the context tree T. By convention, when we write $\prod_{s\in T}$ or $\sum_{s\in T}$ we take the corresponding sum or product over all the *leaves* s of the tree, not all its nodes. We place an independent Dirichlet($\frac{1}{2}, \frac{1}{2}, \cdots, \frac{1}{2}$) distribution on each θ_s so that $\pi(\theta|T) = \prod_{s\in T}\pi(\theta_s)$,where

$$\pi(\theta_s) = \pi(\theta_s(0), \theta_s(1), \cdots, \theta_s(m-1)) = \frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}} \prod_{j=0}^{m-1} \theta_s(j) \propto \prod_{j=0}^{m-1} \theta_s(j) \quad (1.7)$$

Finally, given the model T and the associated parameters $\theta = \{\theta_s : s \in T\}$, the likelihood of the observations is given as in (1) and (2)

$$P(x_1^n|x_{-d+1}^0) = P(x_1^n|x_{-d+1}^0, \theta, T) = \prod_{s\in T} \prod_{j=0}^{m-1} \theta_s(j)^{\alpha_s(j)} \quad (1.8)$$

where, again, $\alpha_s(x)$ denotes the amount of times x follows the context s in x_1^n.

Chapter 2

Mean Marginal Likelihood Algorithm (a.k.a. CTW)

2.1 Preliminaries

In this chapter we are going to analyze the Mean Marginal Likelihood Algorithm. To do so, first, we have to introduce two more quantities.

In Section 1.3.2. we placed a prior distribution on each parameter θ_s. An important property of this prior specification is that the parameters θ can easily be integrated out, so that the marginal likelihoods $P(x|T)$ can be expressed in closed form, a state that is declared and proved in the Lemma below (and it is based on a standard computation [4])

Lemma 2.1.1 *The marginal likelihood $P(x|T)$ of the observations x given a model T is,*

$$P(x|T) = \int_\theta P(x, \theta|T) d(\theta) = \int_\theta P(x|T, \theta)\pi(\theta|T) d(\theta) = \prod_{s \in T} P_e(\alpha_s) \qquad (2.1)$$

where the count vectors α_s are defined in (3) as before, and where the quantity $P_e(\alpha)$ is given by,

$$P_e(\alpha) = \frac{\prod_{j=0}^{m-1} \frac{1}{2}(\frac{3}{2}) \cdots (\alpha(j) - \frac{1}{2})}{\frac{m}{2}(1 + \frac{m}{2}) \cdots (n - 1 + \frac{m}{2})} \qquad (2.2)$$

for a count vector $\alpha = (\alpha(0), \alpha(1), \cdots, \alpha(m-1))$, where $n = \alpha(0) + \alpha(1) + \cdots + \alpha(m-1)$, and with the convention that any empty product is taken to be equal to 1.

Proof From (6) and the definition of the prior on θ, we have,

$$\begin{aligned}
P(x|T) &= \int_\theta P(x|T,\theta)\pi(\theta|T)d(\theta) \\
&= \int_\theta \left[\prod_{s\in T}\prod_{j=0}^{m-1}\theta_s(j)^{\alpha_s(j)}\right]\left[\prod_{s\in T}\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}\prod_{j=0}^{m-1}\theta_s(j)^{-\frac{1}{2}}\right]\prod_{s\in T}d\theta_s \\
&= \prod_{s\in T}\left\{\int_{\theta_s}\left[\prod_{j=0}^{m-1}\theta_s(j)^{\alpha_s(j)}\right]\left[\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}\prod_{j=0}^{m-1}\theta_s(j)^{-\frac{1}{2}}\right]d\theta_s\right\} \\
&= \prod_{s\in T}\left\{\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}\int_{\theta_s}\left[\prod_{j=0}^{m-1}\theta_s(j)^{\alpha_s(j)-\frac{1}{2}}\right]d\theta_s\right\} \\
&= \prod_{s\in T}\left\{\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}B_S\int_{\theta_s}\frac{1}{B_s}\left[\prod_{j=0}^{m-1}\theta_s(j)^{(\alpha_s(j)+\frac{1}{2})-1}\right]d\theta_s\right\} \\
&= \prod_{s\in T}\left\{\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}B_S\right\}
\end{aligned}$$

where

$$B_s = \frac{\prod_{j=0}^{m-1}\Gamma(\alpha_s(j)+\frac{1}{2})}{\Gamma(n_s+\frac{m}{2})}$$

is the normalizing constant of the Dirichlet distribution with parameters $\alpha = (\alpha_s(0)+\frac{1}{2}, \alpha_s(1)+\frac{1}{2}, \cdots, \alpha_s(m-1)+\frac{1}{2})$, where $n_s = \alpha_s(0)+\alpha_s(1)+\cdots+\alpha_s(m-1)$. Therefore, using standard properties of the Gamma function,

$$\begin{aligned}
P(x|T) &= \prod_{s\in T}\left\{\frac{\Gamma(\frac{m}{2})}{\pi^{\frac{m}{2}}}\frac{\prod_{j=0}^{m-1}\Gamma(\alpha_s(j)+\frac{1}{2})}{\Gamma(\frac{2n_s+m}{2})}\right\} \\
&= \prod_{s\in T}\left\{\frac{(m-2)!!}{2^{\frac{m-1}{2}}}\frac{\prod_{j=0}^{m-1}\frac{(2\alpha_s(j)-1)!!}{2^{\alpha_s(j)}}}{\frac{(2n_s+m-2)!!}{2^{\frac{2n_s+m-1}{2}}}}\right\} \\
&= \prod_{s\in T}\left\{\frac{2^{n_s}(m-2)!!}{(2n_s+m-2)!!}\prod_{j=0}^{m-1}\left[\frac{1}{2}\left(\frac{3}{2}\right)\cdots\left(\alpha_s(j)-\frac{1}{2}\right)\right]\right\} \\
&= \prod_{s\in T}P_e(\alpha_s)
\end{aligned}$$

as claimed, where the double-factorial function is defined as usual by $n!! = \prod_{i:0\le 2i\le n}(n-2i)$. ■

2.2 The Algorithm

Now, we can proceed to the description of the *Mean Marginal Likelihood Algorithm*, an effective algorithm that computes the mean marginal likelihood $P(x)$ of the observed samples. This method takes as input:

- The alphabet's $\mathcal{A} = \{0, 1, \cdots, m-1\}$ size m
- The maximum context depth D
- The observations x^n_{-D+1}, where $x^n_{-D+1} \in \mathcal{A}^{n+D}$
- The value of the prior parameter β

And it executes the following steps:

1. Build an m-are tree T_{MMLA} whose leaves are all the contexts x^{i-1}_{i-D}, $i = 1, 2, \cdots, n$ that appear in the observations string x^n_{-D+1}. If some node $s \in T_{MMLA}$ is at depth $d < D$ and some but not all of its children are also in T_{MMLA}, then add all its remaining children as well, so that T_{MMLA} is a proper tree.

2. Compute the count vector α_s, at each node s of the tree T_{MMLA} (not only at the leaves), and note that the α_s will be the all-zero vector for the additional leaves included in the last step of (1).

3. Compute the probability $P_{e,s} = P_e(\alpha_s)$ given by (8), at each node s of the tree T_{MMLA} recalling the convention that $P_e(\alpha_s) = 1$ when α_s is the all-zero count vector.

4. Write sj for the concatenation of context s and symbol j, corresponding to the jth child of node s. Starting at the leaves and proceeding recursively towards the root, compute the mixture probabilities,

$$P_{w,s} = \begin{cases} P_(e,s), & \text{if s is a leaf} \\ \beta\ P_{e,s} + (1-\beta)\prod_{j=0}^{m-1} P_{w,sj}, & \text{otherwise} \end{cases} \tag{2.3}$$

   at each node s of the tree T_{MMLA}

5. Output the mixture probability $P_{x,\lambda}$ at the root λ

2.3 The Corresponding Theorem

The following theorem states that the MMLA indeed computes the mean marginal likelihood of the samples x.

Theorem 2.3.1 *The mixture probability $P_{w,\lambda}$ at the root λ computed by the MMLA is exactly the mean marginal likelihood of the observations,*

$$P_{w,\lambda} = \sum_{T \in \mathcal{T}(D)} \pi_D(T) \int_\theta P(x_1^n | x_{-D+1}^0, T, \theta) \pi(\theta|T) d(\theta) \tag{2.4}$$

where the sum is over the context trees T in the collection $\mathcal{T}(D)$ of all proper context trees of depth no greater than D and $P(x_1^n|x_{-D+1}^0, T, \theta)$ is an alternative way of interpretation of the quantity $P(x|T,\theta)$.

Proof First we note that, without loss of generality, we may assume that the tree T_{MMLA} is the complete tree of depth D; if some node s of the complete tree is not in T_{MMLA} , we simply assume that it has an all-zero count vector α_s. The proof is again by induction. We adopt the notation of the proof of Lemma 1.3.1 and observe that, in view of Lemma 1.3.2, it suffices to show that,

$$P_{w,\lambda} = \sum_{T \in \mathcal{T}(D)} \pi_D(T) \prod_{s \in T} P_e(\alpha_s) \tag{2.5}$$

We claim that the following more general statement holds (*inductive hypothesis*): For any node s at depth d with $0 \le d \le D$, we have,

$$P_{w,s} = \sum_{U \in \mathcal{T}(D-d)} \pi_{D-d}(U) \prod_{u \in U} P_e(\alpha_{su}) \tag{2.6}$$

where su denotes the concatenation of contexts s and u. Clearly (2.6) implies (2.5) upon taking $s = \lambda$, and (2.6) is trivially true for nodes s at level D, since it reduces to the fact that $P_{w,s} = P_{e,s}$ for leaves s, by definition.

Suppose (2.6) holds for all nodes s at depth d for some fixed $0 < d \le D$.

Let s be a node at depth d-1, then, by the inductive hypothesis,

$$
\begin{aligned}
P_{w,s} &= \beta\ P_e(\alpha_s) + (1-\beta) \prod_{j=0}^{m-1} P_{w,sj} \\
&= \beta\ P_e(\alpha_s) + (1-\beta) \prod_{j=0}^{m-1} \left[\sum_{T_j \in \mathcal{T}(D-d)} \pi_{D-d}(T_j) \prod_{t \in T_j} P_e(\alpha_{sjt}) \right] \\
&= \beta\ P_e(\alpha_s) + (1-\beta) \sum_{T_0,T_1,\ldots,T_{m-1} \in \mathcal{T}(D-d)} \prod_{j=0}^{m-1} \left[\pi_{D-d}(T_j) \prod_{t \in T_j} P_e(\alpha_{sjt}) \right] \\
&= \beta\ P_e(\alpha_s) + \frac{1-\beta}{\alpha^{m-1}} \sum_{T_0,T_1,\ldots,T_{m-1} \in \mathcal{T}(D-d)} \pi_{D-d+1}(\cup_j T_j) \left[\prod_{j=0}^{m-1} \prod_{t \in T_j} P_e(\alpha_{sjt}) \right]
\end{aligned}
$$

where sjt denotes the concatenation of context s, then symbol j, then context t, in that order, and where for the last step we have used that $\pi_D(T) = \alpha^{m-1} \prod_{j=0}^{m-1} \pi_{D-1}(T_j)$. Concatenating every symbol j with every leaf of the corresponding tree T_j, we end up with all the leaves of the larger tree $\cup_j T_j$. Therefore,

$$
P_{w,s} = \beta\ P_e(\alpha_s) + \frac{1-\beta}{\alpha^{m-1}} \sum_{T_0,T_1,\ldots,T_{m-1} \in \mathcal{T}(D-d)} \pi_{D-d+1}(\cup_j T_j) \prod_{t \in \cup_j T_j} P_e(\alpha_{st})
$$

and since $1-\beta = \alpha^{m-1}$ and $\pi_d(\Lambda) = \beta$ for all $d \geq 1$,

$$
\begin{aligned}
P_{w,s} &= \pi_{D-d+1}(\Lambda) P_e(\alpha_s) + \sum_{T_0,T_1,\ldots,T_{m-1} \in \mathcal{T}(D-d)} \pi_{D-d+1}(\cup_j T_j) \prod_{t \in \cup_j T_j} P_e(\alpha_{st}) \\
&= \pi_{D-d+1}(\Lambda) P_e(\alpha_s) + \sum_{T \in \mathcal{T}(D-d+1),\ T \neq \Lambda} \pi_{D-d+1}(T) \prod_{t \in T} P_e(\alpha_{st}) \\
&= \sum_{T \in \mathcal{T}(D-d+1)} \pi_{D-d+1}(T) \prod_{s \in T} P_e(\alpha_s)
\end{aligned}
$$

This establishes (2.6) for all nodes s at depth $d-1$, completing the inductive step and the proof of the theorem. ■

Chapter 3

Maximum A Posteriori Probability Tree Algorithm (a.k.a. CTM)

Finally, we propose an efficient algorithm that identifies the a posteriori most likely tree model.

3.1 The Algorithm

As with the MMLA, the MAPT algorithm takes as input:

- The alphabet's $\mathcal{A} = \{0, 1, \cdots, m-1\}$ size m
- The maximum context depth D
- The observations x^n_{-D+1}, where $x^n_{-D+1} \in \mathcal{A}^{n+D}$
- The value of the prior parameter β

Using these, it executes the following steps:

1. Build an m-ary tree T_{MMLA} from the contexts produced by x^n_{-D+1}, and compute the count vectors α_s and the corresponding probabilities $P_{e,s} = P_e(\alpha_s)$ at all nodes s of the tree T_{MMLA} as in steps (1)-(3) of the MMLA

2. Write sj for the concatenation of context s and symbol j, corresponding to the jth child of node s. Starting at the leaves and proceeding recursively towards the root, compute the maximal probabilities,

$$P_{m,s} = \begin{cases} \beta, & \text{if s is a leaf at depth } d < D \\ P_(e,s), & \text{if s is a leaf at depth D} \\ \max\{\beta \ P_{e,s}, (1-\beta)\prod_{j=0}^{m-1} P_{m,sj}\}, & \text{otherwise} \end{cases} \tag{3.1}$$

   at each node s of the tree T_{MMLA}

3. Starting at the root node and proceeding recursively with its descendants, for each node s: If the maximum in (3.1) is achieved by the first term, then prune all its descendants from the tree T_{MMLA}; otherwise, repeat the same process at each of the m children of node s.

4. After all nodes have been exhausted in (3), output the resulting tree T_1^* and the maximal probability at the root $p_{m,\lambda}$

3.2 The Corresponding Theorem

The following theorem states that the MAPT algorithm indeed identifies the a posteriori most likely model.

Theorem 3.2.1 *The tree T_1^* produced by the MAPT algorithm $\forall \beta \in [0.5, 1)$ is indeed the maximum a posteriori probability tree,*

$$\pi(T_1^*|x) = \max_{T \in \mathcal{T}(D)} \pi(T|x) \tag{3.2}$$

and the maximal probability at the root satisfies,

$$P_{m,\lambda} = \pi(T_1^*, x) \tag{3.3}$$

Proof As with the proof of Theorem 2.1 , we note that, without loss of generality, we may assume that the tree TMMLA is the complete tree of depth D. It is easy to see that, for $\beta \in [0.5, 1)$ this assumption is equivalent to that in the description of the algorithm, giving the same initial values to all leaves of T_{MMLA}.

The proof is once again by induction, and we adopt the same notation as in the proofs of Lemma 1.1 and Theorem 2.1. First we will prove that,

$$P_{m,\lambda} = \max_{T \in \mathcal{T}(D)} \pi(T, x) \tag{3.4}$$

which is equivalent to,

$$P_{m,\lambda} = \max_{T \in \mathcal{T}(D)} \pi_D(T) \prod_{s \in T} P_e(\alpha_s) \tag{3.5}$$

As in the proof of Theorem 2.1, we claim that the following more general statement holds (*inductive hypothesis*): For any node s at depth d with $0 \leq d \leq D$, we have,

$$P_{m,s} = \max_{U \in \mathcal{T}(D-d)} \pi_{D-d}(U) \prod_{u \in U} P_e(\alpha_{su}) \tag{3.6}$$

where su denotes the concatenation of contexts s and u. Taking $s = \lambda$ in (3.6) gives (3.5), and (3.6) is trivially true for nodes s at level D, since it reduces to the fact that $P_{m,s} = P_{e,s}$ for leaves s, by definition.

For the inductive step, we assume that (3.6) holds for all nodes s at depth d for some fixed $0 < d \leq D$ and consider a node s at depth d-1. By the inductive hypothesis we have,

$$\begin{aligned}
P_{m,s} &= \max\left\{\beta \; P_e(\alpha_s), (1-\beta) \prod_{j=0}^{m-1} P_{m,sj}\right\} \\
&= \max\left\{\beta \; P_e(\alpha_s), (1-\beta) \prod_{j=0}^{m-1} \left[\max_{T_j \in \mathcal{T}(D-d)} \pi_{D-d}(T_j) \prod_{t \in T_j} P_e(\alpha_{sjt})\right]\right\} \\
&= \max\left\{\beta \; P_e(\alpha_s), (1-\beta) \max_{T_0, T_1, \ldots, T_{m-1} \in \mathcal{T}(D-d)} \prod_{j=0}^{m-1} \left[\pi_{D-d}(T_j) \prod_{t \in T_j} P_e(\alpha_{sjt})\right]\right\} \\
&= \max\left\{\beta \; P_e(\alpha_s), \frac{1-\beta}{\alpha^{m-1}} \max_{T_0, T_1, \ldots, T_{m-1} \in \mathcal{T}(D-d)} \pi_{D-d+1}(\cup_j T_j) \left[\prod_{j=0}^{m-1} \prod_{t \in T_j} P_e(\alpha_{sjt})\right]\right\}
\end{aligned}$$

Arguing as in the proof of Theorem 2.1,

$$\begin{aligned} P_{m,s} &= \max\left\{\pi_{D-d+1}(\Lambda)P_e(\alpha_s), \max_{T_0,T_1,\ldots,T_{m-1}\in\mathcal{T}(D-d)} \pi_{D-d+1}(\cup_j T_j) \prod_{t\in\cup_j T_j} P_e(\alpha_{st})\right\} \\ &= \max\left\{\pi_{D-d+1}(\Lambda)P_e(\alpha_s), \max_{T\in\mathcal{T}(D-d+1),\ T\neq\Lambda} \pi_{D-d+1}(T)\prod_{t\in T} P_e(\alpha_{st})\right\} \\ &= \max_{T\in\mathcal{T}(D-d+1)} \pi_{D-d+1}(T)\prod_{s\in T} P_e(\alpha_s) \end{aligned}$$

This establishes (3.6) for all nodes s at depth $d-1$, completing the inductive step and hence also proving (3.4) and (3.5).

To complete the proof of the theorem, it now suffices to show that,

$$P_{m,\lambda} = \pi(T_1^*, x) \tag{3.7}$$

that implies,

$$\max_{T\in\mathcal{T}(D)} \pi(T,x) = \pi(T_1^*, x) \tag{3.8}$$

By Lemma 1.2, (3.7) is equivalent to

$$P_{m,\lambda} = \pi_D(T_1^*)\prod_{s\in T_1^*} P_e(\alpha_s) \tag{3.9}$$

and, once again, we will establish the following more general statement: For any node s at depth d with $0 \le d \le D$, we have,

$$P_{m,s} = \max\left\{\beta\ P_e(\alpha_s), (1-\beta)\prod_{j=0}^{m-1} P_{m,sj}\right\} = \pi_{D-d}(T(s))\prod_{t\in T(s)} P_e(\alpha_{st}) \tag{3.10}$$

where $T(s)$ is the tree that the MAPT algorithm would produce if it started its step (3) at node s. Taking $s=\lambda$ in (3.9) gives (3.7), and (3.9) is again trivially true for leaves s at level D, by the definition of the maximal probabilities $P_{m,s}$

Finally, for the inductive step assume (3.10) holds for all nodes at depth $0 < d \le D$, and let s be a node at depth d-1. We consider two separate cases:

1. If the maximum in (3.10) is achieved by the first term, then $P_{m,s} = \beta P_e(\alpha_s)$ and T(s) consists of s only, so that (3.10) holds trivially

2. If the maximum in (3.10) is achieved by the second term, then $T(s) = \cup_j T(sj)$, and using the inductive hypothesis, we obtain:

$$\begin{aligned}
P_{m,s} &= (1-\beta)\prod_{j=0}^{m-1} P_{m,sj} \\
&= (1-\beta)\prod_{j=0}^{m-1}\left[\pi_{D-d}(T(sj))\prod_{t\in T(sj)} P_e(\alpha_{sjt})\right] \\
&= \pi_{D-d+1}(\cup_j T(sj))\prod_{j=0}^{m-1}\prod_{t\in T(sj)} P_e(\alpha_{sjt}) \\
&= \pi_{D-d+1}(\cup_j T(sj))\prod_{t\in \cup_j T(sj)} P_e(\alpha_{st}) \\
&= \pi_{D-d+1}(T(s))\prod_{s\in T(s)} P_e(\alpha_{st})
\end{aligned}$$

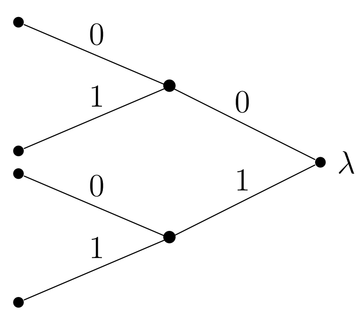
This establishes (3.10) and completes the proof of the theorem. ■

Chapter 4

Two illustrative examples

4.1 Second Order Binary Markov Chain

Here, we shall compute step-by-step the marginal likelihood and the maximum posterior probability tree of a second order binary Markov Chain. Suppose we are given the string “0,0,1,1,1,0,0,0,0,0” with the corresponding nodes’ set $S = \{\lambda, 0, 1, 00, 01, 10, 11\}$ that builds the following context tree:



First, we need to calculate the count vectors α for each node $s \in S$ at each depth D, $D \leq 2$, produced by the specific string in a manner explained in detail in chapter 2:

- D=0: $\alpha_\lambda = (5, 3)$
- D=1: $\alpha_0 = (4, 1)$, $\alpha_1 = (1, 2)$
- D=2: $\alpha_{00} = (3, 1)$, $\alpha_{01} = (1, 0)$, $\alpha_{10} = (0, 1)$, $\alpha_{11} = (1, 1)$

Based on those count vectors, we shall calculate the estimated probabilities P_{es} for each node $s \in S$ at each depth D, $D \leq 2$,:

- D=0:

$$P_{es}(\lambda) = \frac{\frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{9}{2} \times \frac{1}{2} \times \frac{3}{2} \times \frac{5}{2}}{8!} = \frac{45}{2^{15}} \approx 1.3733 \times 10^{-3}$$

- D=1:

$$P_{es}(0) = \frac{\frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{1}{2}}{5!} \approx \frac{7}{2^8} = 2.734 \times 10^{-2}$$

$$P_{es}(1) = \frac{\frac{1}{2} \times \frac{1}{2} \times \frac{3}{2}}{3!} = \frac{1}{2^4} = 6.25 \times 10^{-2}$$

- D=2:

$$P_{es}(00) = \frac{\frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{1}{2}}{4!} = \frac{5}{2^7} \approx 3.906 \times 10^{-2}$$

$$P_{es}(01) = \frac{\frac{1}{2}}{1!} = 0.5$$

$$P_{es}(10) = \frac{\frac{1}{2}}{1!} = 0.5$$

$$P_{es}(11) = \frac{\frac{1}{2} \times \frac{1}{2}}{2!} = \frac{1}{2^3} = 0.125$$

Next, we may proceed in computing the mixture-weighted probabilities P_w and the maximal probabilities P_m for each node $s \in S$ at each depth D, $D \leq 2$, using those P_{es} in a bottom-up traversal manner.

- D=2:
  - $$P_w(00) = P_{es}00 \approx 3.906 \times 10^{-2}$$
  - $$P_w(01) = P_{es}(01) = 0.5$$
  - $$P_w(10) = P_{es}(10) = 0.5$$
  - $$P_w(11) = P_{es}(11) = 0.125$$
  - $$P_m(00) = P_{es}00 \approx 3.906 \times 10^{-2}$$
  - $$P_m(01) = P_{es}(01) = 0.5$$
  - $$P_m(10) = P_{es}(10) = 0.5$$
  - $$P_m(11) = P_{es}(11) = 0.125$$
- D=1:
  - $$\begin{aligned} P_w(0) &= \beta P_{es}(0) + (1-\beta)\prod_{i=0}^{1}(P_w(0i)) \\ &\approx \beta(2.734 \times 10^{-2} - 0.5 \times 3.906 \times 10^{-2}) + 0.5 \times 3.906 \times 10^{-2} \\ &= \beta \times 7.81 \times 10^{-3} + 1.953 \times 10^{-2} \end{aligned}$$

–

$$
\begin{aligned}
P_w(1) &= \beta P_{es}(1) + (1-\beta)\prod_{i=0}^{1}(P_w(1i)) = \beta(6.25\times 10^{-2} - 0.5\times 0.125) + 0.5\times 0.125 \\
&= 6.25\times 10^{-2}
\end{aligned}
$$

–

$$
\begin{aligned}
P_m(0) &= \max\{\beta P_{es}(0), (1-\beta)\prod_{i=0}^{1}(P_m(0i))\} \\
&\approx \max\{\beta\times 2.734\times 10^{-2}, (1-\beta)0.5\times 3.906\times 10^{-2}\} \\
&= \max\{\beta\times 2.734\times 10^{-2}, (1-\beta)1.953\times 10^{-2}\} \\
&= \beta\times 2.734\times 10^{-2}
\end{aligned}
$$

–

$$
\begin{aligned}
P_m(1) &= \max\{\beta P_{es}(1), (1-\beta)\prod_{i=0}^{1}(P_m(1i))\} \\
&= \max\{\beta\times 6.25\times 10^{-2}, (1-\beta)0.5\times 0.125\} \\
&= \beta\times 6.25\times 10^{-2}
\end{aligned}
$$

- D=0:

–

$$
\begin{aligned}
P_w(\lambda) &= \beta P_{es}(\lambda) + (1-\beta)\prod_{i=0}^{1}(P_w(i)) \\
&\approx \beta\times 1.3733\times 10^{-3} + (1-\beta)6.25\times 10^{-2}(\beta\times 7.81\times 10^{-3} + 1.953\times 10^{-2}) \\
&= 1.2207\times 10^{-3} + \beta\times 6.4\times 10^{-4} - \beta\times 4.883\times 10^{-4}
\end{aligned}
$$

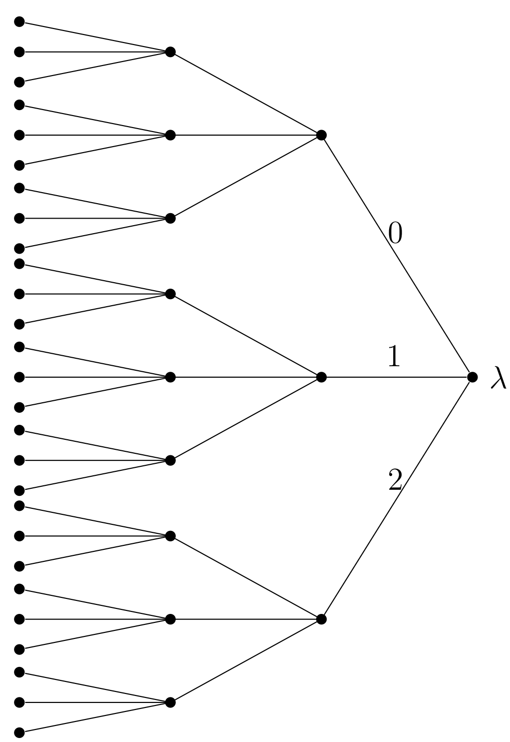
–

$$
\begin{aligned}
P_m(\lambda) &= \max\{\beta P_{es}(\lambda), (1-\beta)\prod_{i=0}^{1}(P_m(i))\} \\
&\approx \max\{\beta\times 1.3733\times 10^{-3}, (1-\beta)\times\beta^2\times 6.25\times 10^{-2}\times 2.734\times 10^{-2}\} \\
&= \max\{\beta\times 1.3733\times 10^{-3}, (1-\beta)\times\beta^2\times 1.709\times 10^{-3}\} \\
&= \beta\times 1.3733\times 10^{-3}
\end{aligned}
$$

For several values of β , $\beta \in [0.5, 1)$ we obtain different mean marginal likelihoods and maximal probabilities of the observations. For example,. for $\beta = 0.5$ and $\beta = 0.9$ the mean marginal likelihoods and maximal probabilities at the root would be $P_w(\lambda) = 1.4186 \times 10^{-3}$ and $P_w(\lambda) = 1.4012 \times 10^{-3}$ and $P_m(\lambda) = 6.867 \times 10^{-4}$ and $P_w(\lambda) = 1.2360 \times 10^{-3}$, respectively.
On the other hand, given the fact that $\beta \in [0.5, 1)$ in this specific case we shall always obtain the root as the Maximum A Posteriori Probability Tree.

4.2 Third Order Ternary Markov Chain

In accordance with Section 1, here, we shall compute the marginal likelihood and the maximum posterior probability tree of a third order ternary Markov Chain.
Given the input string "2,2,1,1,1,1,1,1,1,1,1,1,1,2,0,1,1,1,2,1" and the corresponding nodes' set $S = \{\lambda, 0, 00, 01, 02, 000, 001, 002, 010, 011, 012, 020, 021, 022, 1, 10, 11, 12, 100, 101, 102, 110, 111, 112, 120, 121, 122, 2, 20, 21, 22, 200, 201, 202, 210, 211, 212, 220, 221, 222\}$ we obtain the following context tree:



As in Section 1, first, we shall calculate the count vectors α for each node $s \in S$ at each depth D, $D \leq 3$, produced by the specific string:

- D=0: $\alpha_\lambda = (1, 14, 2)$
- D=1: $\alpha_0 = (0, 1, 0), \quad \alpha_1 = (0, 12, 2), \quad \alpha_2 = (1, 1, 0)$
- D=2:
  - $\alpha_{00} = (0, 0, 0), \quad \alpha_{01} = (0, 0, 0), \quad \alpha_{02} = (0, 1, 0)$
  - $\alpha_{10} = (0, 1, 0), \quad \alpha_{11} = (0, 10, 2), \quad \alpha_{12} = (0, 1, 0)$
  - $\alpha_{20} = (0, 0, 0), \quad \alpha_{21} = (1, 1, 0), \quad \alpha_{22} = (0, 0, 0)$
- D=3:
  - * $\alpha_{000} = (0, 0, 0), \quad \alpha_{001} = (0, 0, 0), \quad \alpha_{002} = (0, 0, 0)$
    * $\alpha_{010} = (0, 0, 0), \quad \alpha_{011} = (0, 0, 0), \quad \alpha_{012} = (0, 0, 0)$
    * $\alpha_{020} = (0, 0, 0), \quad \alpha_{021} = (0, 1, 0), \quad \alpha_{022} = (0, 0, 0)$
  - * $\alpha_{100} = (0, 0, 0), \quad \alpha_{101} = (0, 0, 0), \quad \alpha_{102} = (0, 1, 0)$
    * $\alpha_{110} = (0, 1, 0), \quad \alpha_{111} = (0, 8, 2), \quad \alpha_{112} = (0, 1, 0)$
    * $\alpha_{120} = (0, 0, 0), \quad \alpha_{121} = (0, 0, 0), \quad \alpha_{122} = (0, 1, 0)$
  - * $\alpha_{200} = (0, 0, 0), \quad \alpha_{201} = (0, 0, 0), \quad \alpha_{202} = (0, 0, 0)$
    * $\alpha_{210} = (0, 0, 0), \quad \alpha_{211} = (1, 1, 0), \quad \alpha_{212} = (0, 0, 0)$
    * $\alpha_{220} = (0, 0, 0), \quad \alpha_{221} = (0, 0, 0), \quad \alpha_{222} = (0, 0, 0)$

Then, we calculate the estimated probabilities P_{es} for each node $s \in S$ at each depth D, $D \leq 3$,:

- D=0:

$$\begin{aligned}
P_{es}(\lambda) &= \frac{\frac{1}{2} \times \frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{9}{2} \times \frac{11}{2} \times \frac{13}{2} \times \frac{15}{2} \times \frac{17}{2} \times \frac{19}{2} \times \frac{21}{2} \times \frac{23}{2} \times \frac{25}{2} \times \frac{27}{2} \times \frac{1}{2} \times \frac{3}{2}}{\frac{3}{2} \times \frac{5}{2} \ldots \frac{33}{2} \times \frac{35}{2}} \\
&= \frac{3}{29 \times 31 \times 33 \times 35} \approx 2.8892 \times 10^{-6}
\end{aligned}$$

- D=1:
  - $$P_{es}(0) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$\begin{aligned} P_{es}(1) &= \frac{\frac{1}{2} \times \frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{9}{2} \times \frac{11}{2} \times \frac{13}{2} \times \frac{15}{2} \times \frac{17}{2} \times \frac{19}{2} \times \frac{21}{2} \times \frac{23}{2}}{\frac{3}{2} \times \frac{5}{2} \ldots \frac{27}{2} \times \frac{29}{2}} \\ &= \frac{3}{25 \times 27 \times 29} \approx 1.5326 \times 10^{-4} \end{aligned}$$
  - $$P_{es}(2) = \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{3}{2} \times \frac{5}{2}} = \frac{1}{15} \approx 6.667 \times 10^{-2}$$
- D=2:
  - $$P_{es}(02) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$P_{es}(10) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$\begin{aligned} P_{es}(11) &= \frac{\frac{1}{2} \times \frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{9}{2} \times \frac{11}{2} \times \frac{13}{2} \times \frac{15}{2} \times \frac{17}{2} \times \frac{19}{2} \times \frac{1}{2} \times \frac{3}{2}}{\frac{3}{2} \times \frac{5}{2} \ldots \frac{23}{2} \times \frac{25}{2}} \\ &= \frac{3}{21 \times 23 \times 25} \approx 2.4845 \times 10^{-4} \end{aligned}$$
  - $$P_{es}(12) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$

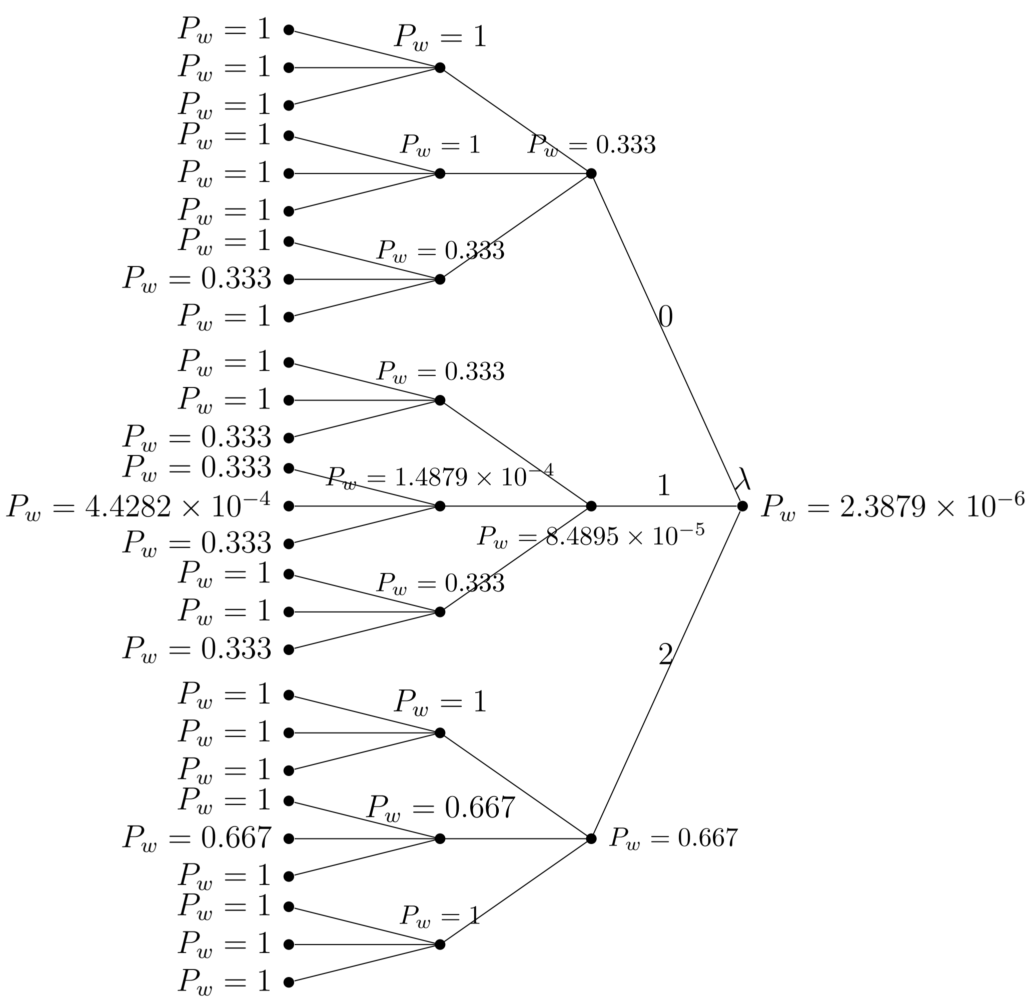
- $$P_{es}(21) = \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{3}{2} \times \frac{5}{2}} = \frac{1}{15} \approx 6.667 \times 10^{-2}$$

- D=3:
  - $$P_{es}(021) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$P_{es}(102) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$P_{es}(110) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$\begin{aligned} P_{es}(111) &= \frac{\frac{1}{2} \times \frac{1}{2} \times \frac{3}{2} \times \frac{5}{2} \times \frac{7}{2} \times \frac{9}{2} \times \frac{11}{2} \times \frac{13}{2} \times \frac{15}{2} \times \frac{1}{2} \times \frac{3}{2}}{\frac{3}{2} \times \frac{5}{2} \ldots \frac{19}{2} \times \frac{21}{2}} \\ &= \frac{3}{17 \times 19 \times 21} \approx 4.4228 \times 10^{-4} \end{aligned}$$
  - $$P_{es}(112) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$P_{es}(122) = \frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$$
  - $$P_{es}(211) = \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{3}{2} \times \frac{5}{2}} = \frac{1}{15} \approx 6.667 \times 10^{-2}$$

The corresponding P_{es} for each count vector of the form $\alpha_s = (0,0,0),\ s \in S$ equals one.

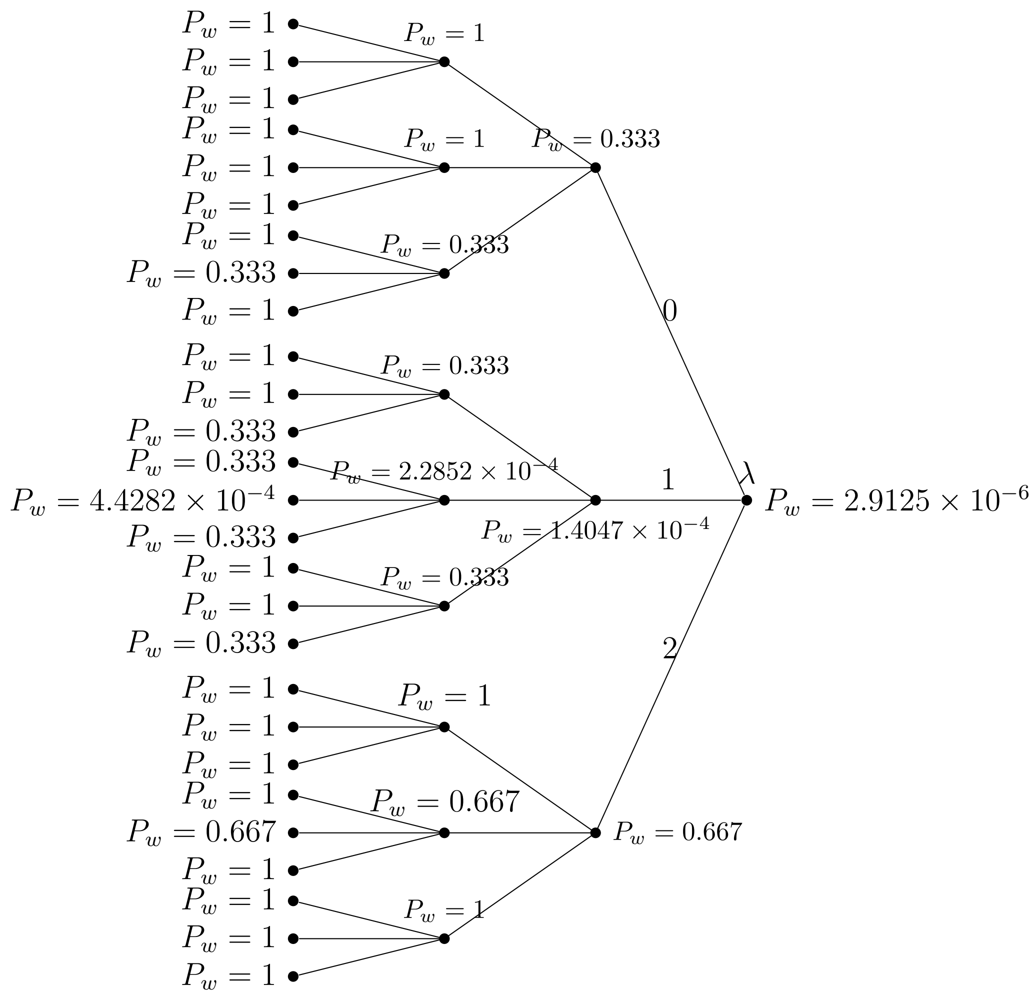
Then, we proceed with the calculation of the mixture-weighted probabilities P_w and the maximal probabilities P_m for each node $s \in S$ at each depth D, $D \leq 3$, in accordance with Section 1.
For brevity reasons, we shall illustrate them in a tree-structure , for several values of β.

- $\beta = 0.5$:
  Mixture-weighted probabilities:



The maximal probability P_m at the tree's root is $P_m(\lambda) = 1.4446 \times 10^{-6}$ and the Maximum A Posteriori Probability Tree is the root itself.

- $\beta = 0.9$:



Here, the maximal probability P_m at the root is $P_m(\lambda) = 2.6003 \times 10^{-6}$ (i.e. almost twice the probability of $\beta = 0.5$) and the Maximum A Posteriori Probability Tree is the root.

Chapter 5

Simulations

Note: Throughout this chapter we are going to use the following notation:

- alphabet's size: $|A|$
- input's size: n
- maximum depth (i.e. suffix's length): D

Moreover, we shall interpret the probabilities in a (base two) log-scale to overcome underflow issues .

5.1 A Toy Example

The input string of this model, has been constructed via independent and identically distributed (a.k.a. I.I.D.) random variables
$X_i \sim Bern(0.05), i = 1 : 500$. The values of the parameters are: $|A| = 2$, n=500, D=15 and $\beta = 0.5$.
The prior probability of the real model (i.e. the root) is:

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^0 \times 0.5 = 0.5$$

The Maximum A Posteriori Tree algorithm detects the true model (i.e. the MAP tree is only one node, the root) with corresponding posterior probability:

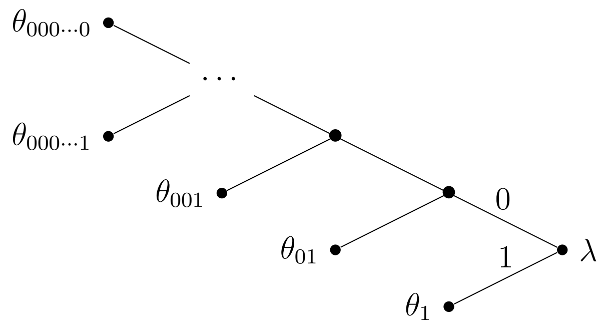
$$\pi(T_{MAP}|x) = \frac{\pi(x, T_{MAP})}{\pi(T_{MAP})} = \frac{P_m(\lambda)}{P_w(\lambda)} = 2^{-144.73130+144.6330} \approx 0.9341$$

We increased the sample size (n=50,000) and maximal depth (D = 50) and obtained the exact same model with posterior probability :

$$\pi(T_{MAP}|x) = \frac{\pi(x, T_{MAP})}{\pi(T_{MAP})} = \frac{P_m(\lambda)}{P_w(\lambda)} = 2^{-14145.75290+14145.73774} \approx 0.9895$$

5.2 Binary Renewal Process

The second model would look like:



where, the associated parameter vector at depth 1 is $\theta_1 = (0, 1)$, at the all-zero leaf s at depth 20 is $\theta_{00\cdots0} = (1, 0)$ and at each leaf corresponding to a context s of the form $00\cdots01$ at depth $2 \leq 20$ is:

$$\theta_s(1) = 1 - \theta_s(0) = \left[\sum_{i=|s|-1}^{20} \frac{i}{|s|-1}\right]^{-1}$$

The rest parameters' values are $|A| = 2$, n=100.000, D=45 and $\beta = 0.5$. The prior probability of the real model (i.e. the root) is:

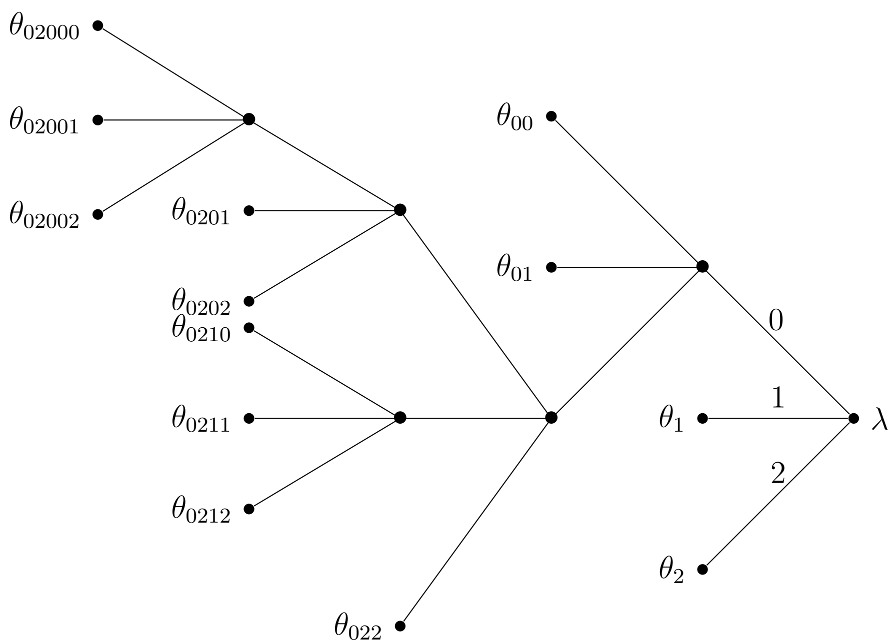
$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^{20} \times 0.5^{21} \approx 4.5475 \times 10^{-13}$$

As in the former example, the Maximum A Posteriori Tree is the true model with posterior probability:

$$\pi(T_{MAP}|x) = \frac{\pi(x, T_{MAP})}{\pi(T_{MAP})} = \frac{P_m(\lambda)}{P_w(\lambda)} = 2^{-27231.47185+27216.70234} \approx 3.5802 \times 10^{-3}$$

5.3 A Ternary, Fifth Order Markov Chain

The third model's parameters are $|A| = 3$ and $\beta = 0.5$. The data were drawn by the simulation of the following model:



For each leaf of this Context Tree, we associate a parameter vector $\theta_{leaf} = (\theta_0, \theta_1, \ldots, \theta_{m-1})$ that assumes a distribution over [0,1] (also, for each leaf, $\sum_{i=0}^{m-1} \theta_i = 1$).

$$\theta_2 = (0.2, 0.4, 0.4)$$

$$\theta_1 = (0.4, 0.4, 0.2)$$

$$\theta_{00} = (0.4, 0.2, 0.4)$$

$$\theta_{01} = (0.3, 0.6, 0.1)$$

$$\theta_{022} = (0.5, 0.3, 0.2)$$

$$\theta_{0212} = (0.1, 0.3, 0.6)$$

$$\theta_{0211} = (0.05, 0.25, 0.7)$$

$$\theta_{0210} = (0.35, 0.55, 0.1)$$

$$\theta_{0202} = (0.1, 0.2, 0.7)$$

$$\theta_{0201} = (0.8, 0.05, 0.15)$$

$$\theta_{02002} = (0.7, 0.2, 0.1)$$

$$\theta_{02001} = (0.1, 0.1, 0.8)$$

$$\theta_{02000} = (0.3, 0.45, 0.25)$$

We have repeated the same experiment for several suffixes' and input lengths and obtained the following MAPT and posterior probabilities:

1. Here we illustrate the prior and posterior probabilities among with the MAP tree for $\beta = 0.5$ and suffix length $D = 1, 2, 3, 4, 5, \dots, 100$.

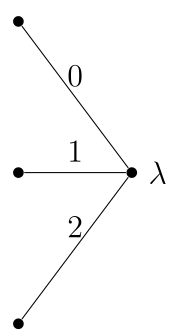
   - D=1:
     Prior probability :

     $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5 \times 0.5^0 = 0.5$$

     Posterior probability :

     $$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-15339.293121127295+15339.293121127284} \approx 0.999999999992$$

     MAP tree:

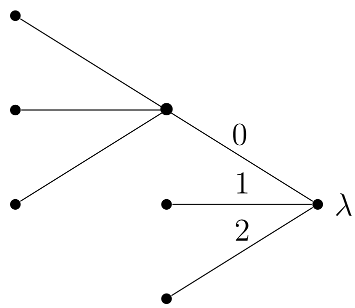
   - D=2:
     Prior probability :

     $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^2 \times 0.5^2 = 6.25 \times 10^{-2}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-14877.50540+14877.50538} \approx 0.9999802$$

MAP tree:



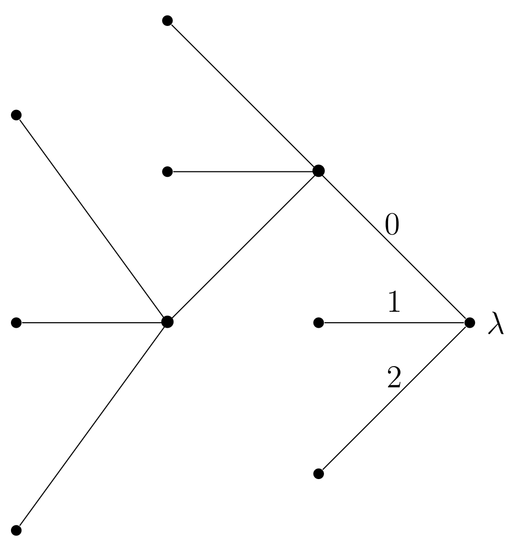
- D=3:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^3 \times 0.5^4 = 7.8125 \times 10^{-3}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-14849.19104+14849.19071} \approx 0.99977376$$

MAP tree:



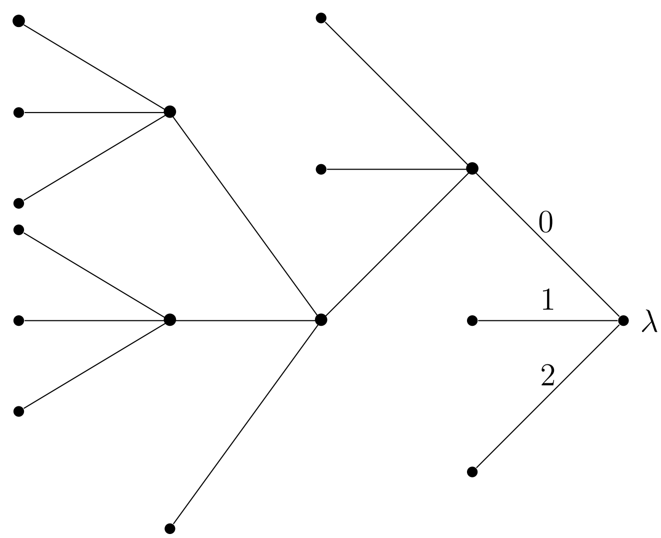
- D=4:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^5 \times 0.5^5 \approx 9.766 \times 10^{-4}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-14777.64852-14777.64089} \approx 0.99472256599$$

MAP tree:



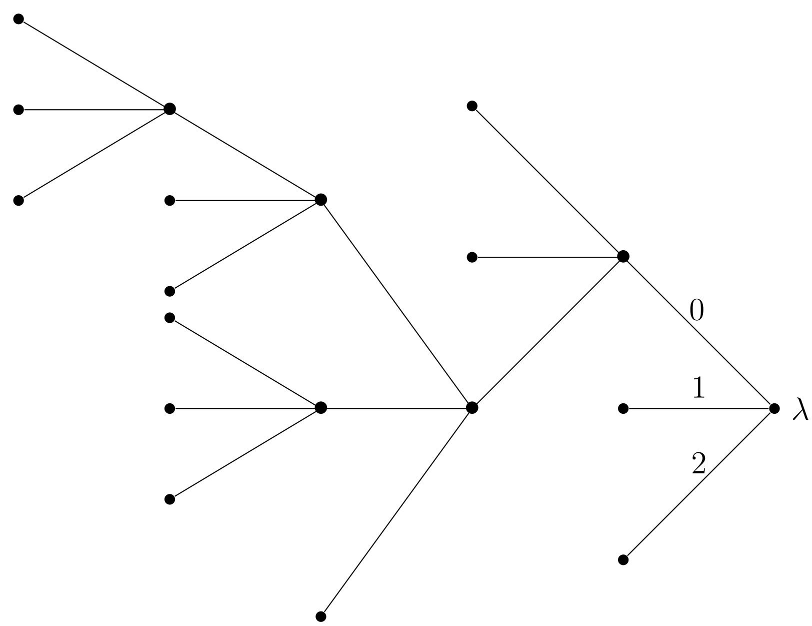
- D=5:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{10} \approx 1.5259 \times 10^{-5}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-14767.53029+14767.11465} \approx 0.749686335$$

MAP tree (true model):



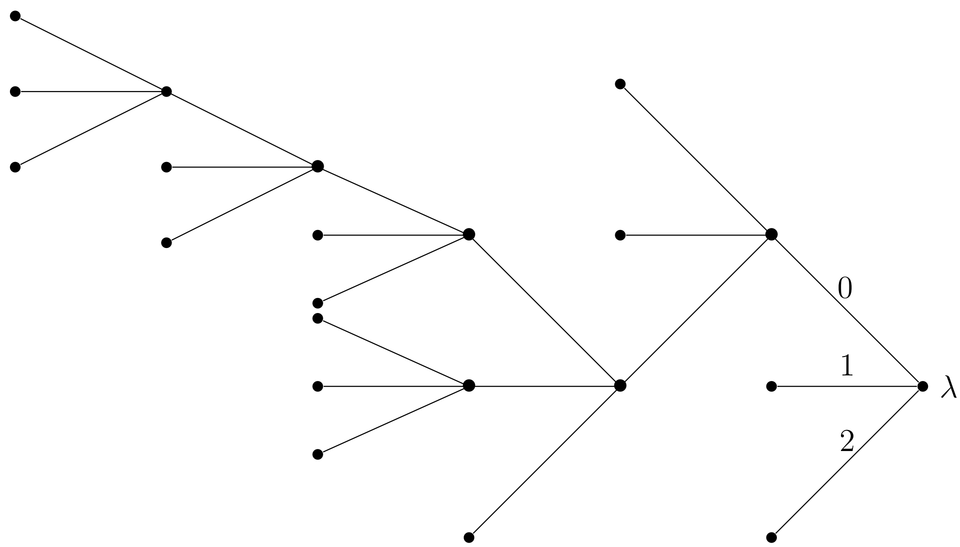
- D=6:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^7 \times 0.5^{12} \approx 1.907 \times 10^{-6}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-14770.35336 + -14768.16645} \approx 0.21962125$$

MAP tree:



- D=7:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-14770.53029+14766.68944} \approx 0.0697894445$$

  MAP tree: true model

- D=8:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-14770.53029+14766.87929} = 0.080444448$$

  MAP tree: true model

- D=9:
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-14770.53029+14766.55952} \approx 0.0637794$$

  MAP tree: true model

- D=10:
  Prior probability :

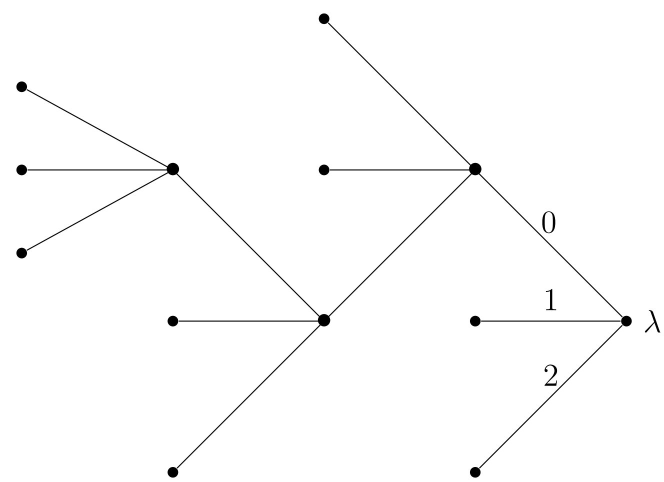
$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-14770.53029+14766.57514} \approx 0.0644735$$

  MAP tree: true model

2. Next, we shall perform the same experiment for input lengths $n = \{1,000, 2,000, 10,000, 20,000\}$, keeping the other parameters fixed ($|A| = 3,\ \beta = 0.5, D = 10$)

- n=1,000:
  Prior probability :
  $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^4 \times 0.5^9 \approx 1.221 \times 10^{-4}$$
  Posterior probability :
  $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-1515.91020+1511.52971} \approx 0.0480112$$
  MAP tree:
  

- n=2,000
  Prior probability :
  $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$
  Posterior probability :
  $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-3019.24671+3011.03859} \approx 3.381 \times 10^{-3}$$
  MAP tree: true model

- n=10,000
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-14770.53029+14766.57514} \approx 0.0644735$$

  MAP tree: true model

- n=20,000
  Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

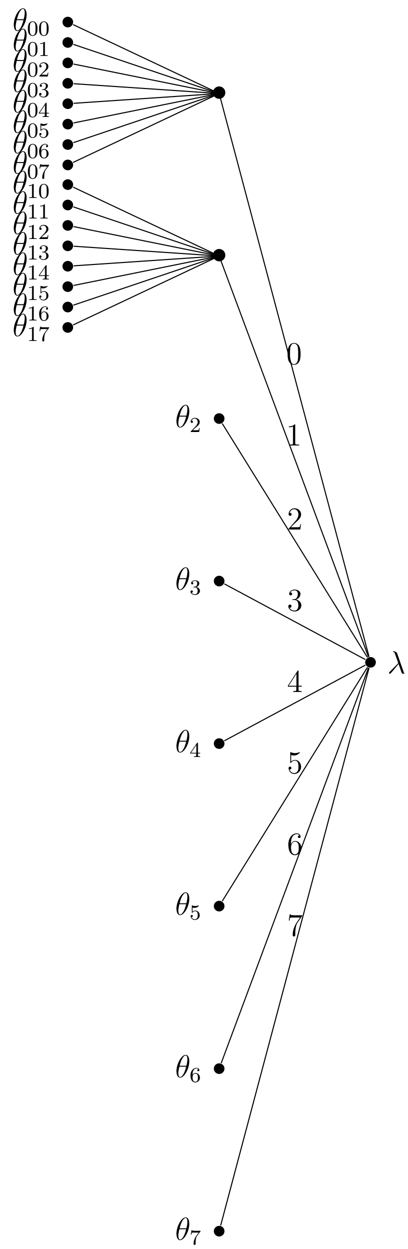
  Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-29479.73131+29479.22982} \approx 0.7063783$$

  MAP tree: true model

5.4 An Octal, Second Order Markov Chain

The fourth model has $|A| = 8$, n=50,000 and D=5 . Here, we will consider two cases, one for $\beta = 0.5$ and another for $\beta = 1 - 0.5^7 = 0.9921875$
Model:



where:

$$\theta_7 = (0.15, 0.25, 0.1, 0.175, 0.025, 0.0125, 0.0375, 0.25)$$
$$\theta_6 = (0.01, 0.04, 0.5, 0.08, 0.02, 0.05, 0.2, 0.1)$$
$$\theta_5 = (0.02, 0.03, 0.09, 0.01, 0.15, 0.3, 0.25, 0.15)$$
$$\theta_4 = (0.7, 0.1, 0.03, 0.02, 0.04, 0.01, 0.01, 0.09)$$
$$\theta_3 = (0.1, 0.15, 0.1, 0.05, 0.1, 0.15, 0.1, 0.25)$$
$$\theta_2 = (0.2275, 0.0725, 0.01, 0.04, 0.08, 0.02, 0.5, 0.05)$$
$$\theta_{17} = (0.2, 0.2, 0.05, 0.3, 0.025, 0.05, 0.1, 0.075)$$
$$\theta_{16} = (0.08, 0.1225, 0.6, 0.04, 0.0725, 0.02, 0.5, 0.01)$$
$$\theta_{15} = (0.005, 0.02, 0.1, 0.175, 0.235, 0.165, 0.1, 0.2)$$
$$\theta_{14} = (0.1, 0.12, 0.23, 0.025, 0.125, 0.25, 0.06, 0.09)$$
$$\theta_{13} = (0.09, 0.02, 0.01, 0.6, 0.12, 0.04, 0.09, 0.03)$$
$$\theta_{12} = (0.45, 0.0125, 0.075, 0.0375, 0.1, 0.025, 0.25, 0.05)$$
$$\theta_{11} = (0.15, 0.02, 0.15, 0.13, 0.3, 0.19, 0.05, 0.01)$$
$$\theta_{10} = (0.1, 0.25, 0.15, 0.1, 0.1, 0.05, 0.1, 0.15)$$
$$\theta_{07} = (0.0075, 0.05, 0.23, 0.5, 0.08, 0.07, 0.0125, 0.05)$$
$$\theta_{06} = (0.075, 0.3, 0.05, 0.1, 0.2, 0.025, 0.2, 0.05)$$
$$\theta_{05} = (0.1, 0.01, 0.04, 0.05, 0.5, 0.02, 0.08, 0.2)$$
$$\theta_{04} = (0.05, 0.025, 0.05, 0.075, 0.3, 0.2, 0.2, 0.1)$$
$$\theta_{03} = (0.8, 0.1, 0.004, 0.015, 0.015, 0.006, 0.02, 0.04)$$
$$\theta_{02} = (0.01, 0.05, 0.05, 0.13, 0.15, 0.3, 0.12, 0.19)$$
$$\theta_{01} = (0.09, 0.06, 0.25, 0.125, 0.025, 0.23, 0.12, 0.1)$$
$$\theta_{00} = (0.25, 0.1, 0.15, 0.1, 0.05, 0.1, 0.15, 0.1)$$

1. $\beta = 0.5$:
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)} = 0.5^3 \times 0.5^{22} \approx 2.9802 * 10^{-8}$$

   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-117464.35101+117464.35090} \approx 0.9999283$$

   MAP tree: true model

2. $\beta = 0.9921875$:
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)}0.0078125^3 \times 0.9921875^{22} \approx 4.0127 * 10^{-7}$$

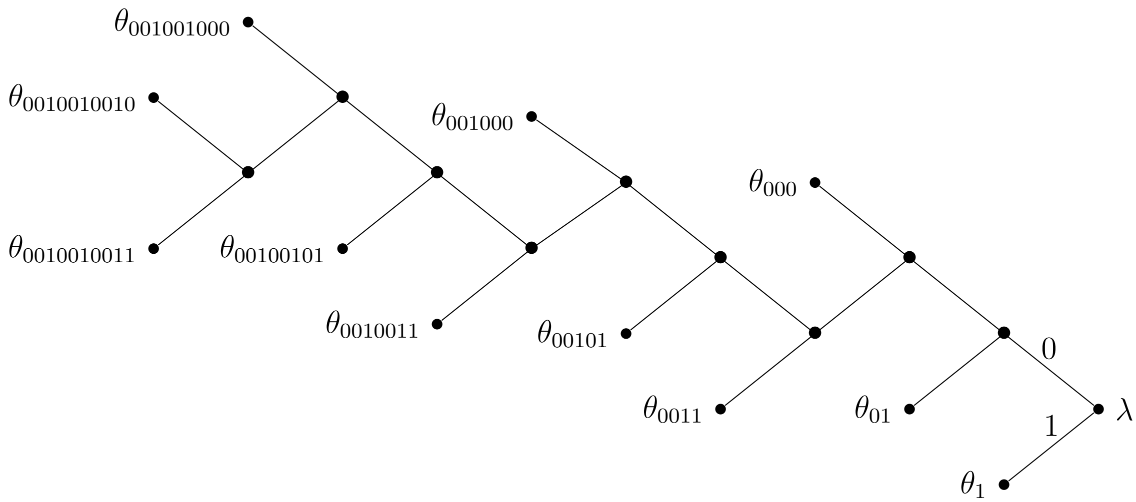
   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-117460.59994220+117460.59994186} \approx 0.9999998$$

   MAP tree: true model

5.5 A Binary, Tenth Order Markov Chain

The fifth model concerns a binary Markov Chains for several suffixes' and input lengths. The true model is the following:



where:

$$\theta_1 \sim Bern(1/150)$$

$$\theta_{01} \sim Bern(1/100)$$

$$\theta_{000} \sim Bern(1/20)$$

$$\theta_{0011} \sim Bern(1/50)$$

$$\theta_{00101} \sim Bern(1/150)$$

$$\theta_{001000} \sim Bern(1/30)$$

$$\theta_{0010011} \sim Bern(1/80)$$

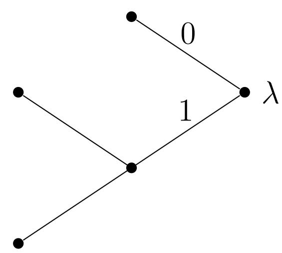
$$\theta_{00100101} \sim Bern(1/40)$$

$$\theta_{001001000} \sim Bern(1/60)$$

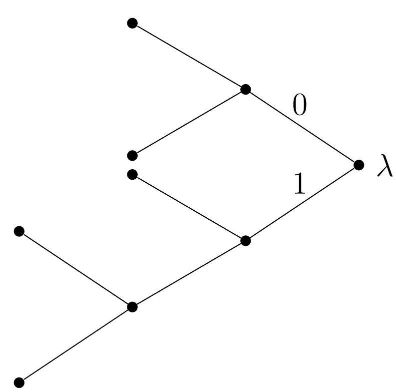
$$\theta_{0010010010} \sim Bern(1/90)$$

$$\theta_{0010010011} \sim Bern(1/45)$$

1. Parameters: D=5, β=0.5, n=1,000:
   Prior probability :
   $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^2 \times 0.5^3 = 0.03125$$
   Posterior probability :
   $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-294.22663+292.41814} \approx 0.2854907$$
   MAP tree:



2. Parameters: D=10, β=0.5, n=10,000:
   Prior probability :
   $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^4 \times 0.5^5 = 0.001953125$$
   Posterior probability :
   $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-2828.05537+2825.68343} \approx 0.1931851$$
   MAP tree:



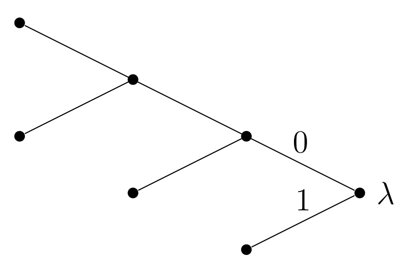
3. Parameters: D=50, β=0.5, n=100,000
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^3 \times 0.5^4 = 0.0078125$$

   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-27722.25641+27721.46262} \approx 0.5768250$$

   MAP tree:



4. Parameters: D=50, β=0.5, n=1,000,000
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^3 \times 0.5^4 = 0.0078125$$

   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-274211.79266+274210.95177} \approx 0.5582987$$

   MAP tree: as in the former simulation

5.6 A Binary Hidden Markov Model

The present Hidden Markov Model (H.M.M.) was analysed in [3] and motivated by neurosciense framework [2,4]. From the entropy to the statistical structure of spike trains). Here, $\{Y_n\}$ is a Markov Chain with state space $S_Y = \{1, 2, 3\}$, initial probability vector $Q = (1, 0, 0)$ and transition matrix:

$$P = \begin{pmatrix} 0.999 & 0.0005 & 0.0005 \\ 0.0005 & 0.999 & 0.0005 \\ 0.0005 & 0.0005 & 0.999 \end{pmatrix}$$

We are interested in $\{X_n\}$ (the model's R.V.), where:

$$X_n|Y_n = 1 \sim Bern(0.005)$$

$$X_n|Y_n = 2 \sim Bern(0.02)$$

$$X_n|Y_n = 3 \sim Bern(0.05)$$

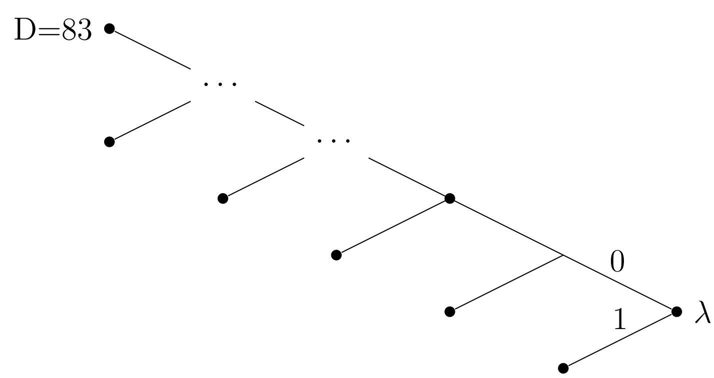
(parameters: $|A| = 2$, n=1,000,000, D=100 and $\beta = 0.5$)
Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^{83} \times 0.5^{84} \approx 5.346 \times 10^{-51}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-158020.22726+158006.78471} \approx 8.9823 \times 10^{-5}$$

MAP tree:



5.7 A More Complex Hidden Markov Model

In the following experiment $\{Y_n\}$ is a Markov Chain as in the fourth model and we are interested in $\{X_n\}$ (the model's R.V.), where:

$$X_n = 0 \; if \; Y_n = 0$$

$$X_n \sim U\{0,1,2,3,4\} \; if \; Y_n = 1 \; or \; 2$$

$$X_n \sim Bin(4, 0.9) \; if \; Y_n = 3 \; or \; 4$$

$$X_n \sim Bin(4, 0.1) \; if \; Y_n = 5, 6 \; or \; 7$$

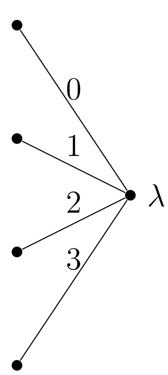
(parameters: $|A| = 5$, n=100,000 , D=7 and $\beta = 1 - 2^{-|A|+1} = 0.9375$)
Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^3 \times 0.5^4 = 0.0078125$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-198847.97211+198847.40378} = 0.67434$$

MAP tree:



5.8 A Ternary, Fifth Order Markov Chain, with noise

Here, we will consider the following setting:

- $\{Y_n\}$ is a Markov Chain as in the third model

- $X_n = Y_n + Bern(0.01)(mod3)$

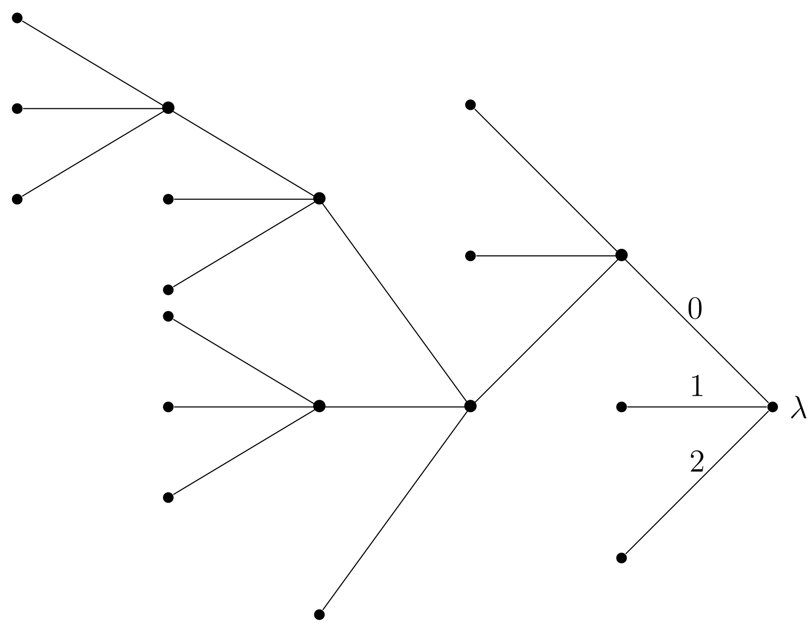
(model's parameters: $|A| = 3$, n=100,000, D=12 and $\beta = 0.5$)
Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{13} \approx 1.907 \times 10^{-6}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-148516.27069+148516.261176} \approx 0.9934$$

MAP tree (true model):



5.9 Noisy Markovian Samples

In this HMM, we study the *MAP* case for several β's and sample sizes. Here $\{Y_n\}$ is a Markov Chain with state space $S_Y = \{0, 1, 2, 3\}$, initial probability vector $Q = (1, 0, 0, 0)$ and transition matrix:

$$P = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

We wish to explore $\{X\}$ where $X_n = Y_n + Bern(0.05)(mod 4)$ (parameters: $|A| = 4$, $n = \{10,000, 50,000, 200,000, 500,000, 1,000,000\}$ and D=5.)

- Various samples' sizes ($\beta = 0.5$):

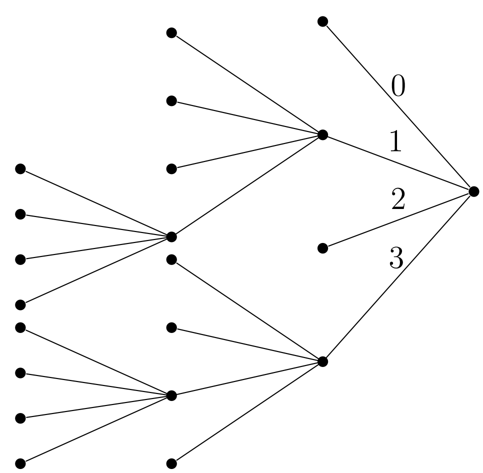
  1. n=10,000:
     Prior probability :

     $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^5 \times 0.5^{16} \approx 0.5^{21} = 4.7684 \times 10^{-7}$$

     Posterior probability :

     $$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-15088.86868+15086.21907} \approx 0.1593636$$

     MAP tree:



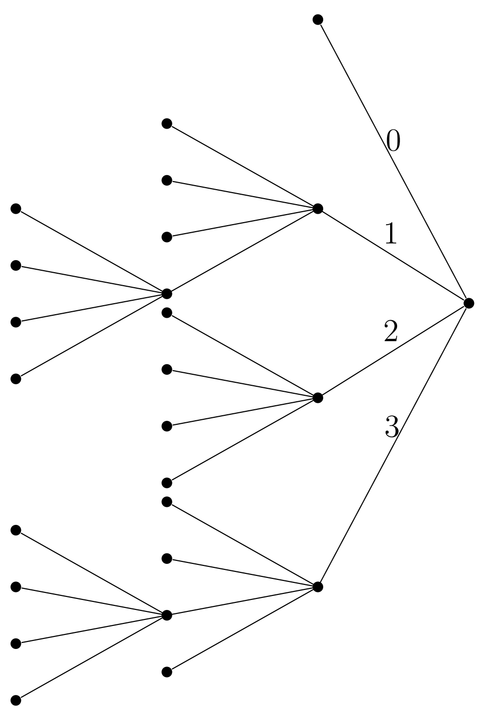
2. n=50,000:
   Prior probability :

   $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)} = 0.5^6 \times 0.5^{19} = 0.5^{25} \approx 2.98 \times 10^{-8}$$

   Posterior probability :

   $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-74974.14250+74972.72349} \approx 0.3739678$$

   MAP tree:

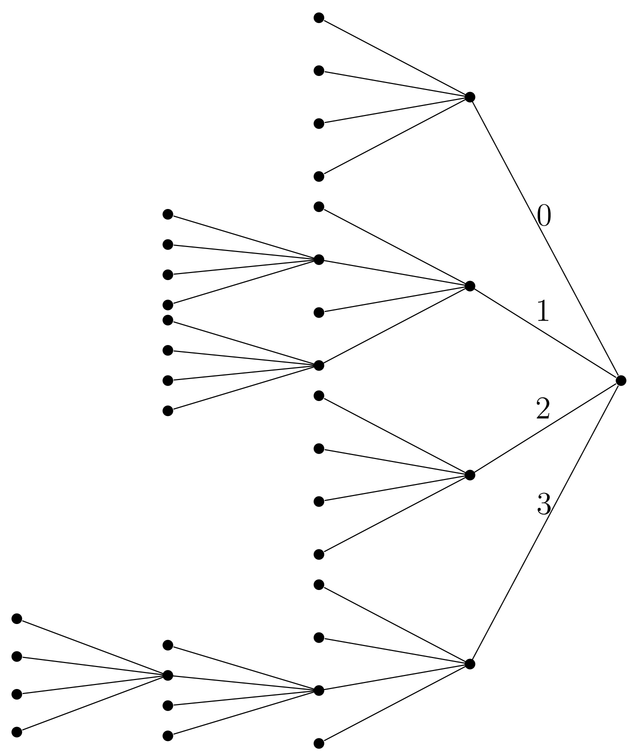
3. n=200,000:
   Prior probability :

   $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}}\beta^{|T^*|-L_D(T^*)} = 0.5^9 \times 0.5^{28} \approx 0.5^{37} = 7.276 \times 10^{-12}$$

   Posterior probability :

   $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-299185.274976+299184.6125} \approx 0.6317931$$

MAP tree:



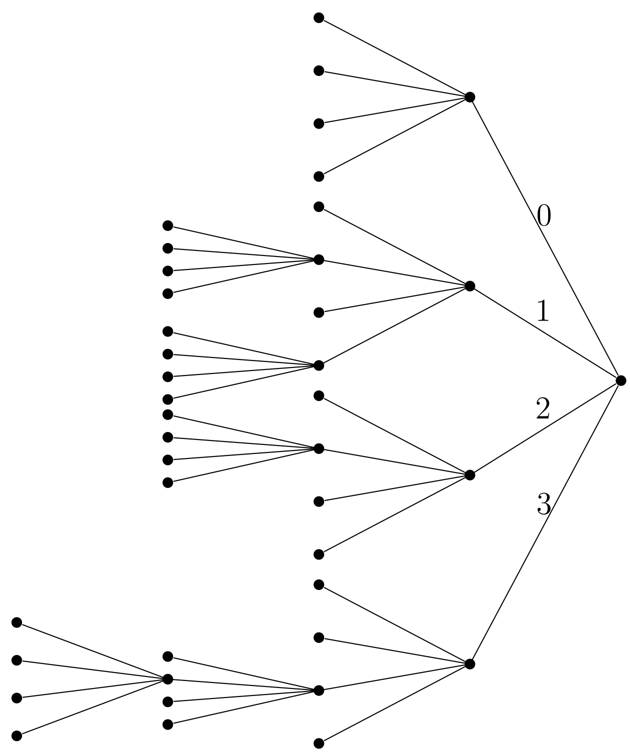
4. n=500,000:
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^{10} \times 0.5^{31} = 0.5^{41} \approx 4.547 \times 10^{-13}$$

   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-746028.031487+746027.373904} \approx 0.6339395$$

MAP tree:



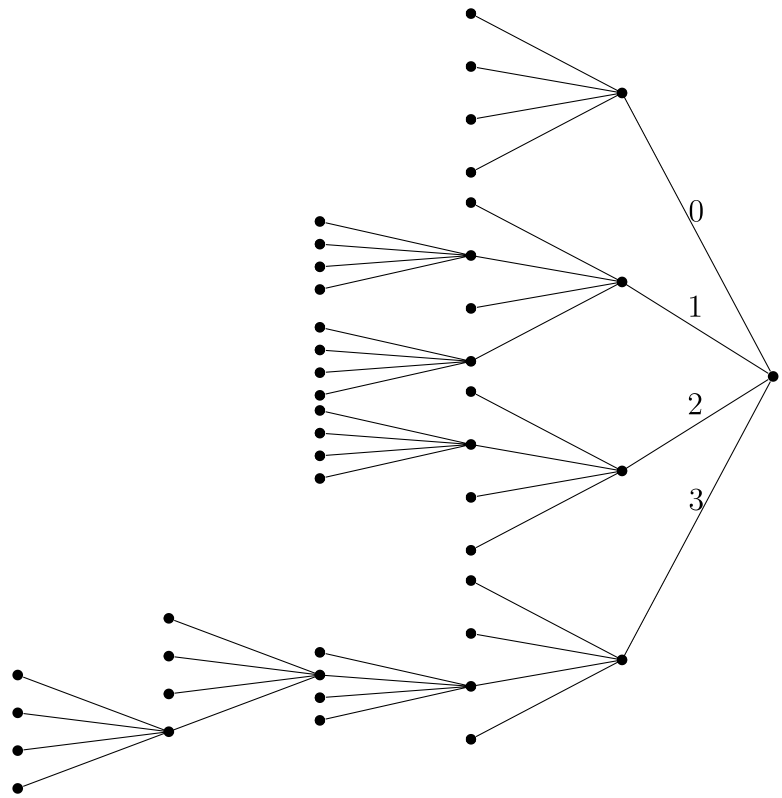
5. n=1,000,000:
   Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.5^{11} \times 0.5^{30} = 0.5^{41} \approx 4.547 \times 10^{-13}$$

   Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-1490440.68061+1490440.60432} \approx 0.9484936$$

MAP tree:



- Various β's (n=10,000):
  Here, though β varies, the MAP tree is always the same (as in figure)

  1. $\beta = 0.75$:
     Prior probability :

     $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.25^5 \times 0.75^{16} \approx 9.7877 \times 10^{-6}$$

     Posterior probability :

     $$\pi(T^*|x) = \frac{\pi(x,T^*)}{p(x)} = 2^{-15085.50928+15083.63437} \approx 0.273644$$

  2. $\beta = 0.875$:
     Prior probability :

     $$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.125^5 \times 0.875^{16} \approx 3.6031 \times 10^{-6}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-15086.950998+15085.61808} \approx 0.396965$$

3. β=0.9:
Prior probability :

$$\pi(T^*) = (1-\beta)^{\frac{|T^*|-1}{|A|-1}} \beta^{|T^*|-L_D(T^*)} = 0.1^5 \times 0.9^{16} \approx 1.8530 \times 10^{-6}$$

Posterior probability :

$$\pi(T^*|x) = \frac{\pi(x, T^*)}{p(x)} = 2^{-15087.910365+15086.615099} \approx 0.40746$$

5.10 Remarks

All nine experiments share some common characteristics:

1. Maximum A Posteriori Tree algorithm has tracked down the model in all cases – apart from the fifth model at which, increasing the sample size leads to the true model, eventually

2. Increase of the sample size leads to the right models with high posterior probability

3. Increase of the β parameter leads to smaller models-trees among with higher a posteriori probabilities – a fact that could be considered as an instance of Occam's Razor [1]

Chapter 6

Appendix

Below we present the Scilab code that produced the simulations:

6.1 Model $N^o 1$

```
function model1(length_)
    n=1 ;
    while n <=length_
        v=rand();
        a(n)=0;
        if    v < (1/20) then
                a(n) = 1 ;
        end
        n=n+1;
    end
    a=a';
    fprintfMat('tmp1.txt',a,"%1.0f")

endfunction
```

6.2 Model $N^o 2$

```
function model2test(length_)
    length_=length_-1;
    n=1;
    while n <length_
       a(n)=1;
       if (length_ - n)>=20 then
          Y=distribution();
       else Y=(length_ - n);
       end
       for j=1:Y
           a(n+j)=0;
       end
       n=n+Y+1;
    end
    a(n)=1;
    a=a';
    fprintfMat('tmpa.txt',a,  "%1.0f")
endfunction

function s=distribution()
    v=rand();
    s=20;
    if v<= [1/400] then
        s=1;
    elseif v<= [4/400] then
        s=2;
    elseif v<= [9/400] then
        s=3;
    elseif v<= [16/400] then
        s=4;
    elseif v<= [25/400] then
        s=5;
    elseif v<= [36/400] then
        s=6;
    elseif v<= [49/400] then
        s=7;
    elseif v<= [64/400] then
        s=8;
```

```
    elseif v<= [81/400] then
        s=9;
    elseif v<= [100/400] then
        s=10;
    elseif v<= [121/400] then
        s=11;
    elseif v<= [144/400] then
        s=12;
    elseif v<= [169/400] then
        s=13;
    elseif v<= [196/400] then
        s=14;
    elseif v<= [225/400] then
        s=15;
    elseif v<= [256/400] then
        s=16;
    elseif v<= [289/400] then
        s=17;
    elseif v<= [324/400] then
        s=18;
    elseif v<= [361/400] then
        s=19;
    end
endfunction
```

6.3 Model N^o3

```
function s=nextsymbol3(index)
       prob =   [0.2,0.4,0.4;    // \theta_2
                 0.4,0.4,0.2;    // \theta_1
                 0.4,0.2,0.4;    // \theta_0
                 0.3,0.6,0.1;    // \theta_01
                 0.5,0.3,0.2;    // \theta_022
                 0.1,0.3,0.6;    // \theta_0212
                 0.05,0.25,0.7;  // \theta_0211
                 0.35,0.55,0.1;  // \theta_0210
                 0.1,0.2,0.7;    // \theta_0202
                 0.8,0.05,0.15;  // \theta_0201
                 0.7,0.2,0.1;    // \theta_02002
                 0.1,0.1,0.8;    // \theta_02001
                0.3,0.45,0.25;] // \theta_02000
      v=rand();
      s=2;
      if  v<= prob(index,1) then
            s=0;
      elseif v<= (prob(index,1)+prob(index,2)) then
            s=1;
      end
endfunction

function model3(length_)
    for j=1:5
        a(j)=grand(1,1,'uin',0,2);
    end
    while j<length_
         if a(j)==2 then
             theta=1;// \theta_2
         elseif a(j)==1 then
             theta=2; // \theta_1
         elseif a(j-1)==0  then // 0
             theta=3; // \theta_00
         elseif a(j-1)==1 then
             theta=4; // \theta_01
         elseif a(j-2)==2 then // 02
             theta=5; // \theta_022
```

```
        elseif a(j-2)==1 then // 021
            if a(j-3)==2 then
                theta=6;  // \theta_0212
            elseif a(j-3)==1 then
                theta=7; // \theta_0211
            else theta=8; // \theta_0210
            end
        elseif a(j-3)==2 then // 020
            theta=9; // \theta_0202
        elseif a(j-3)==1 then
            theta=10; // \theta_0201
        elseif a(j-4)==2 then // 0200
            theta=11; // \theta_02002
        elseif a(j-4)==1 then
            theta=12; // \theta_02001
        else theta=13; // \theta_02000
        end
        a(j+1)=nextsymbol3(theta);
        j=j+1;
    end
    tmp="";
    for i=1:length_
       tmp = tmp + string(a(i));
    end
    print('tmp3.txt',tmp);
endfunction
```

6.4 Model N^o4

```
function s=nextsymbol4(index)
       prob = [0.15,0.25,0.1,0.175,0.025,0.0125,0.0375,0.25;   // \theta_7
               0.01,0.04,0.5,0.08,0.02,0.05,0.2,0.1;          // \theta_6
               0.02,0.03,0.09,0.01,0.15,0.3,0.25,0.15;        // \theta_5
               0.7,0.1,0.03,0.02,0.04,0.01,0.01,0.09;         // \theta_4
               0.1,0.15,0.1,0.05,0.1,0.15,0.1,0.25;           // \theta_3
               0.2275,0.0725,0.01,0.04,0.08,0.02,0.5,0.05;    // \theta_2
               0.2,0.2,0.05,0.3,0.025,0.05,0.1,0.075;         // \theta_17
               0.08,0.1225,0.6,0.04,0.0725,0.02,0.5,0.01;     // \theta_16
               0.005,0.02,0.1,0.175,0.235,0.165,0.1,0.2;      // \theta_15
               0.1,0.12,0.23,0.025,0.125,0.25,0.06,0.09;     // \theta_14
               0.09,0.02,0.01,0.6,0.12,0.04,0.09,0.03;       // \theta_13
               0.45,0.0125,0.075,0.0375,0.1,0.025,0.25,0.05; // \theta_12
               0.15,0.02,0.15,0.13,0.3,0.19,0.05,0.01;      // \theta_11
               0.1,0.25,0.15,0.1,0.1,0.05,0.1,0.15;         // \theta_10
               0.0075,0.05,0.23,0.5,0.08,0.07,0.0125,0.05;  // \theta_07
               0.075,0.3,0.05,0.1,0.2,0.025,0.2,0.05;       // \theta_06
               0.1,0.01,0.04,0.05,0.5,0.02,0.08,0.2;        // \theta_05
               0.05,0.025,0.05,0.075,0.3,0.2,0.2,0.1;       // \theta_04
               0.8,0.1,0.004,0.015,0.015,0.006,0.02,0.04;   // \theta_03
               0.01,0.05,0.05,0.13,0.15,0.3,0.12,0.19;      // \theta_02
               0.09,0.06,0.25,0.125,0.025,0.23,0.12,0.1;    // \theta_01
               0.25,0.1,0.15,0.1,0.05,0.1,0.15,0.1;]        // \theta_00

      v=rand();
      s=7;
      if  v<= prob(index,1) then
            s=0;
      elseif v<= sum(prob(index,1:2)) then
            s=1;
      elseif v<= sum(prob(index,1:3)) then
            s=2;
      elseif v<= sum(prob(index,1:4)) then
            s=3;
      elseif v<= sum(prob(index,1:5)) then
            s=4;
      elseif v<= sum(prob(index,1:6)) then
            s=5;
```

```
      elseif v<= sum(prob(index,1:7)) then
            s=6;
      end
endfunction

function model4(length_)
    for j=1:2
        a(j)=grand(1,1,'uin',0,7);
    end
    while j<length_
        if a(j)==7 then
            theta=1; // \theta_7
        elseif a(j)==6 then
            theta=2; // \theta_6
        elseif a(j)==5 then
            theta=3; // \theta_5
        elseif a(j)==4 then
            theta=4; // \theta_4
        elseif a(j)==3 then
            theta=5; // \theta_3
        elseif a(j)==2 then
            theta=6; // \theta_2
        elseif a(j)==1 then
            if a(j-1)==7 then
               theta=7; // \theta_17
            elseif a(j-1)==6 then
                theta=8; // \theta_16
            elseif a(j-1)==5 then
                theta=9; // \theta_15
            elseif a(j-1)==4 then
                theta=10; // \theta_14
            elseif a(j-1)==3 then
                theta=11; // \theta_13
            elseif a(j-1)==2 then
                theta=12; // \theta_12
            elseif a(j-1)==1 then
                theta=13; // \theta_11
            else theta=14; // \theta_10
            end
        elseif a(j-1)==7 then
```

```
            theta=15; // \theta_07
        elseif a(j-1)==6 then
            theta=16; // \theta_06
        elseif a(j-1)==5 then
            theta=17; // \theta_05
        elseif a(j-1)==4 then
            theta=18; // \theta_04
        elseif a(j-1)==3 then
            theta=19; // \theta_03
        elseif a(j-1)==2 then
            theta=20; // \theta_02
        elseif a(j-1)==1 then
            theta=21; // \theta_01
        else theta=22; // \theta_00
        end
        a(j+1)=nextsymbol4(theta);
        j=j+1;
    end
    tmp="";
    for i=1:length_
        tmp = tmp + string(a(i));
    end
    print('mp4.txt',tmp);
endfunction
```

6.5 Model N^o5

```
function s=nextsymbol5(index)
   prob = [(1/150), // \theta_1
           (1/100),// \theta_01
           (1/20), // \theta_000
           (1/50), // \theta_0011
           (1/150),// \theta_00101
           (1/30), // \theta_001000
           (1/80), // \theta_0010011
           (1/40), // \theta_00100101
           (1/60), // \theta_001001000
           (1/90), // \theta_0010010010
           (1/45)] // \theta_0010010011
      v=rand();
      s=0;
      if  v<= prob(index) then
            s=1;
      end
endfunction


function model5(length_)
    for j=1:10
        a(j)=grand(1,1,'uin',0,1);
    end
    while j<length_
         if a(j)==1 then
             theta=1;// \theta_1
         elseif a(j-1)==1 then
             theta=2; // \theta_01
         elseif a(j-2)==0  then
             theta=3; // \theta_000
         elseif a(j-3)==1 then
             theta=4; // \theta_0011
         elseif a(j-4)==1 then
             theta=5; // \theta_00101
         elseif a(j-5)==0 then
             theta=6; // \theta_001000
         elseif a(j-6)==1 then
```

```
            theta=7; // \theta_0010011
        elseif a(j-7)==1 then
            theta=8; // \theta_00100101
        elseif a(j-8)==0 then
            theta=9; // \theta_001001000
        elseif a(j-9)==0 then
            theta=10; // \theta_0010010010
        else theta=11; // \theta_0010010011
        end
        a(j+1)=nextsymbol5(theta);
        j=j+1;
    end
    tmp="";
    for i=1:length_
        tmp = tmp + string(a(i));
    end
    print('tmp5.txt',tmp);
endfunction
```

6.6 Model N^o6

```
function s=nextsymbol(index)
    P= [0.999, 0.0005, 0.0005;
        0.0005, 0.999, 0.0005;
        0.0005, 0.0005, 0.999;]
        v=rand();
        s=3;
        if  v<= P(index,1) then
            s=1;
        elseif v<= (P(index,1)+ P(index,2)) then
            s=2;
        end
endfunction

function mc(length_)
    for j=1
        y(j)=1;
    end
    while j<length_
        if y(j)==1 then
            k=1;
        elseif y(j)==2 then
            k=2;
        else k=3;
        end
        y(j+1)=nextsymbol(k);
        j=j+1;
    end
    y=y';
    t=1;
    while t<=length(y)
        if y(t)==1 then
            l=1;
        elseif y(t)==2 then
            l=2;
        else l=3;
        end
        a(t)=newsymbol(l);
        t=t+1;
```

```
    end
    a=a';
    fprintfMat('tmp6.txt',a,  "%1.0f")
endfunction

function w=newsymbol(ind)
    B= [0.005, 0.02, 0.05]
        u=rand();
        w=0;
        if  u<= B(ind) then
            w=1;
        end
endfunction
```

6.7 Model $N^o 7$

```
function s=nextsymbol4(index)
       prob = [0.15,0.25,0.1,0.175,0.025,0.0125,0.0375,0.25;   // \theta_7
               0.01,0.04,0.5,0.08,0.02,0.05,0.2,0.1;          // \theta_6
               0.02,0.03,0.09,0.01,0.15,0.3,0.25,0.15;        // \theta_5
               0.7,0.1,0.03,0.02,0.04,0.01,0.01,0.09;         // \theta_4
               0.1,0.15,0.1,0.05,0.1,0.15,0.1,0.25;           // \theta_3
               0.2275,0.0725,0.01,0.04,0.08,0.02,0.5,0.05;    // \theta_2
               0.2,0.2,0.05,0.3,0.025,0.05,0.1,0.075;         // \theta_17
               0.08,0.1225,0.6,0.04,0.0725,0.02,0.5,0.01;     // \theta_16
               0.005,0.02,0.1,0.175,0.235,0.165,0.1,0.2;      // \theta_15
               0.1,0.12,0.23,0.025,0.125,0.25,0.06,0.09;     // \theta_14
               0.09,0.02,0.01,0.6,0.12,0.04,0.09,0.03;       // \theta_13
               0.45,0.0125,0.075,0.0375,0.1,0.025,0.25,0.05; // \theta_12
               0.15,0.02,0.15,0.13,0.3,0.19,0.05,0.01;      // \theta_11
               0.1,0.25,0.15,0.1,0.1,0.05,0.1,0.15;         // \theta_10
               0.0075,0.05,0.23,0.5,0.08,0.07,0.0125,0.05;  // \theta_07
               0.075,0.3,0.05,0.1,0.2,0.025,0.2,0.05;       // \theta_06
               0.1,0.01,0.04,0.05,0.5,0.02,0.08,0.2;        // \theta_05
               0.05,0.025,0.05,0.075,0.3,0.2,0.2,0.1;       // \theta_04
               0.8,0.1,0.004,0.015,0.015,0.006,0.02,0.04;   // \theta_03
               0.01,0.05,0.05,0.13,0.15,0.3,0.12,0.19;      // \theta_02
               0.09,0.06,0.25,0.125,0.025,0.23,0.12,0.1;    // \theta_01
               0.25,0.1,0.15,0.1,0.05,0.1,0.15,0.1;]        // \theta_00

      v=rand();
      s=7;
      if  v<= prob(index,1) then
            s=0;
      elseif v<= sum(prob(index,1:2)) then
            s=1;
      elseif v<= sum(prob(index,1:3)) then
            s=2;
      elseif v<= sum(prob(index,1:4)) then
            s=3;
      elseif v<= sum(prob(index,1:5)) then
            s=4;
      elseif v<= sum(prob(index,1:6)) then
            s=5;
```

```
      elseif v<= sum(prob(index,1:7)) then
            s=6;
      end
endfunction

function hmm2(length_)
    for j=1:2
        a(j)=grand(1,1,'uin',0,7);
    end
    while j<length_
        if a(j)==7 then
            theta=1; // \theta_7
        elseif a(j)==6 then
            theta=2; // \theta_6
        elseif a(j)==5 then
            theta=3; // \theta_5
        elseif a(j)==4 then
            theta=4; // \theta_4
        elseif a(j)==3 then
            theta=5; // \theta_3
        elseif a(j)==2 then
            theta=6; // \theta_2
        elseif a(j)==1 then
            if a(j-1)==7 then
               theta=7; // \theta_17
            elseif a(j-1)==6 then
                theta=8; // \theta_16
            elseif a(j-1)==5 then
                theta=9; // \theta_15
            elseif a(j-1)==4 then
                theta=10; // \theta_14
            elseif a(j-1)==3 then
                theta=11; // \theta_13
            elseif a(j-1)==2 then
                theta=12; // \theta_12
            elseif a(j-1)==1 then
                theta=13; // \theta_11
            else theta=14; // \theta_10
            end
        elseif a(j-1)==7 then
```

```
            theta=15; // \theta_07
        elseif a(j-1)==6 then
            theta=16; // \theta_06
        elseif a(j-1)==5 then
            theta=17; // \theta_05
        elseif a(j-1)==4 then
            theta=18; // \theta_04
        elseif a(j-1)==3 then
            theta=19; // \theta_03
        elseif a(j-1)==2 then
            theta=20; // \theta_02
        elseif a(j-1)==1 then
            theta=21; // \theta_01
        else theta=22; // \theta_00
        end
        a(j+1)=nextsymbol4(theta);
        j=j+1;
    end
        t=1;
    while t<=length(a)
        if a(t)==0 then
            y(t)=0;
        elseif a(t)==1 | a(t)==2 then
            y(t)=grand(1,1,'uin',0,4);
        elseif a(t)==3 | a(t)==4 then
            y(t)=grand(1,1,'bin',4,0.9);
        else y(t)=grand(1,1,'bin',4,0.1);
        end
        t=t+1;
    end
    y=y';
    tmp="";
    for i=1:length_
      tmp = tmp + string(y(i));
    end
    print('hmm2.txt',tmp);
endfunction
```

6.8 Model $N^o 8$

```
function s=nextsymbol3(index)
       prob =   [0.2,0.4,0.4;    // \theta_2
                 0.4,0.4,0.2;    // \theta_1
                 0.4,0.2,0.4;    // \theta_0
                 0.3,0.6,0.1;    // \theta_01
                 0.5,0.3,0.2;    // \theta_022
                 0.1,0.3,0.6;    // \theta_0212
                 0.05,0.25,0.7;  // \theta_0211
                 0.35,0.55,0.1;  // \theta_0210
                 0.1,0.2,0.7;    // \theta_0202
                 0.8,0.05,0.15;  // \theta_0201
                 0.7,0.2,0.1;    // \theta_02002
                 0.1,0.1,0.8;    // \theta_02001
                 0.3,0.45,0.25;] // \theta_02000
      v=rand();
      s=2;
      if  v<= prob(index,1) then
            s=0;
      elseif v<= (prob(index,1)+prob(index,2)) then
            s=1;
      end
endfunction

function hmm3a(length_)
    for j=1:5
        a(j)=grand(1,1,'uin',0,2);
    end
    while j<length_
         if a(j)==2 then
             theta=1;// \theta_2
         elseif a(j)==1 then
             theta=2; // \theta_1
         elseif a(j-1)==0  then // 0
             theta=3; // \theta_00
         elseif a(j-1)==1 then
             theta=4; // \theta_01
         elseif a(j-2)==2 then // 02
             theta=5; // \theta_022
```

```
        elseif a(j-2)==1 then // 021
            if a(j-3)==2 then
                theta=6;  // \theta_0212
            elseif a(j-3)==1 then
                theta=7; // \theta_0211
            else theta=8; // \theta_0210
            end
        elseif a(j-3)==2 then // 020
            theta=9; // \theta_0202
        elseif a(j-3)==1 then
            theta=10; // \theta_0201
        elseif a(j-4)==2 then // 0200
            theta=11; // \theta_02002
        elseif a(j-4)==1 then
            theta=12; // \theta_02001
        else theta=13; // \theta_02000
        end
        a(j+1)=nextsymbol3(theta);
        j=j+1;
    end
        t=1;
    while t<=length(a)
        v=rand();
        s=0;
        if v<=0.01 then
            s=1;
        end
        y(t)=modulo(a(t)+s,3);
        t=t+1;
    end
    tmp="";
    for i=1:length_
       tmp = tmp + string(y(i));
    end
    print('hmm3a.txt',tmp);
endfunction
```

6.9 Model N^o9

```
function s=nextsymbol(index)
    P= [0.25, 0.25, 0.25, 0.25;
        (1/3), 0, (1/3),(1/3);
        0, 0.5, 0.5, 0;
        1,0, 0, 0;]
        v=rand();
        s=3;
        if  v<= P(index,1) then
            s=0;
        elseif v<= (P(index,1)+ P(index,2)) then
            s=1;
        elseif v<= (P(index,1)+ P(index,2)+P(index,3)) then
            s=2;
        end
endfunction

function hmm4(length_)
    for j=1
        y(j)=0;
    end
    while j<=length_
        if y(j)==0 then
            k=1;
        elseif y(j)==1 then
            k=2;
        elseif y(j)==2 then
            k=3;
        else k=4;
        end
        y(j+1)=nextsymbol(k);
        j=j+1;
    end
    y=y';
    t=1;
    while t<=length(y)
        u=rand();
        z=0;
        if u<=0.05 then
```

```
            z=1;
        end
        x(t)=modulo(y(t)+z,4);
        t=t+1;
    end
    x=x';
    tmp="";
    for i=1:length_
       tmp = tmp + string(x(i));
    end
    print('hmm4.txt',tmp);
endfunction
```

Bibliography

[1] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, *Occam's Razor,* Information Processing Letters, 24(6):377-380, April 1987.

[2] Y.Gao, *Statistical Models in Neural Information Processing,* PhD thesis, Division of Applied Mathematics, Brown University, Providence, RI, June 2004

[3] Y. Gao, I. Kontoyiannis, and E. Bienenstock. *From the entropy to the statistical structure of spike trains,* In IEEE Int. Symp. on Inform. Theory, Seattle, WA, July 2006

[4] Y. Gao, I. Kontoyiannis, and E. Bienenstock. *Estimating the entropy of binary time series: Methodology, some theory and a simulation study,* Entropy 10(2):71-99, July 2006

[5] F.M.J. Willems, *Coding for a binary independent piecewise-identically-distributed source,* Information Theory, IEEE Transactions on, 42(6):2210-2217, Nov 1996.

[6] F.M.J. Willems, *The context-tree weighting method: Extensions,* IEEE Trans. Inform. Theory, 44(2):792-798, 1998.

[7] F.M.J. Willems, A. Nowbahkt-Irani, and P.A.J. Volf, *Maximum a-posteriori tree models,* In 4^{th} International ITG Conference on Source and Channel Coding, Berlin, Germany, February 2002.

[8] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context tree weighting: Multi-alphabet sources,* In 14^{th} Symposium on Information Theory in the Benelux, Veldhoven, The Netherlands, May 2003.

[9] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context weighting: General finite context sources,* In 14^{th} Symposium on Information Theory in the Benelux, Veldhoven, The Netherlands, May 2003.

[10] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context tree weighting: Basic properties,* IEEE Trans. Inform. Theory, 41(3): 653-664, 1995.

[11] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context weighting for general finite context sources,* IEEE Trans. Inform. Theory, 42:1514-1520, 1996.

[12] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context tree maximizing,* In 2000 Conference on Information Sciences and Systems, Princeton, NJ, March 2000.

[13] F.M.J. Willems and P.A.J. Volf, *Context maximizing: Finding MDL decision trees,* In 15^{th} Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgium, May 1995.

[14] F.M.J. Willems and P.A.J. Volf, *A study of the context tree maximizing method,* In 16^{th} Symposium on Information Theory in the Benelux, Nieuwerkerk a/d IJssel, The Netherlands, May 1995.

[15] P.A.J. Volf, *Weighting Techniques in Data Compression: Theory and Algorithms,* PhD Thesis, Technical University of Eidhoven, Eidhoven, 2002.