Efficient Sensitivity Analysis in Hidden Markov Models

Silja Renooij

Technical Report UU-CS-2011-025
August 2011

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Efficient Sensitivity Analysis in Hidden Markov Models

Silja Renooij

Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
S.Renooij@uu.nl

Abstract

Sensitivity analysis in hidden Markov models (HMMs) is usually performed by means of a perturbation analysis where a small change is applied to the model parameters, upon which the output of interest is re-computed. Recently it was shown that a simple mathematical function describes the relation between HMM parameters and an output probability of interest; this result was established by representing the HMM as a (dynamic) Bayesian network. Up till now, however, no special purpose algorithms existed for determining this function. In this paper we present a new and efficient algorithm for computing sensitivity functions in HMMs; it is the first algorithm to this end which exploits the recursive properties of an HMM, while not relying on a Bayesian network representation.

1 Introduction

Hidden Markov models (HMMs) are frequently applied statistical models for describing processes that evolve over time. Applications of hidden Markov models are found in areas such as speech recognition, machine translation and bioinformatics (see [1] for an overview). An HMM can be represented by the simplest type of dynamic Bayesian network [2, 3], which entails that in addition to the algorithms associated with HMMS, all sorts of algorithms available for (dynamic) Bayesian networks can be straightforwardly applied to HMMs as well.

HMMs specify a number of parameter probabilities, which are bound to be inaccurate to at least some degree. Sensitivity analysis is a standard technique for studying the effects of parameter inaccuracies on the output of a model. In the context of HMMs, sensitivity analysis is usually performed by means of a perturbation analysis where a small change is applied to the parameters, upon which the output of interest is re-computed [4, 5]. For Bayesian networks, a simple mathematical function exists that describes the relation between one or more network parameters and an output probability of interest. Various algorithms are available for computing the constants of this so-called sensitivity function. Recently, it was shown that similar functions describe the relation between model parameters and output probabilities in HMMs [6]. For computing the constants of these functions, it was suggested to represent the HMM as a dynamic Bayesian network, unrolled for a fixed number of time slices, and to use the aforementioned algorithms for computing the constants of the sensitivity function. The drawback of this approach is that the repetitive character of the HMM, with the same parameters occurring for each time step, is not exploited in the computation of the constants. As such, using standard Bayesian network algorithms may not be the most efficient approach to determining sensitivity functions for HMMs. In a

previous workshop paper [7], we introduced the ideas behind a new algorithm for computing the constants of the sensitivity function in HMMs. In this paper we present the details of this efficient algorithm, which exploits the recursive properties of an HMM. To the best of our knowledge, it is the first algorithm for computing HMM sensitivity functions that does not rely on a Bayesian network representation.

This paper is organised as follows. In Section 2, we present some preliminaries concerning HMMs, Bayesian networks and sensitivity functions. In Section 3, we discuss how to compute sensitivity functions that describe the effects of variation in the initial parameters of an HMM. For variation in transition and observation parameters we need a more complex procedure; the general idea behind this procedure is described in Section 4, whereas details are provided in Section 5. We discuss relevant related work in Section 6 and conclude the paper with directions for future research in Section 7.

2 Preliminaries

In this section we present some preliminaries concerning Bayesian networks, hidden Markov models, and sensitivity analysis. Throughout this paper, variables will be denoted by capital letters, and their values by lower case.

2.1 Bayesian networks

A *Bayesian network* is a discrete, static statistical model for representing and reasoning about a domain of application. In essence, a Bayesian network is a concise representation of the joint probability distribution on the set of statistical variables relevant to the application domain [8, 9]. A Bayesian network $\mathcal{B}$ combines an acyclic directed graph $G = (V_G, A_G)$, representing the statistical variables and their dependencies by means of nodes V_G and arcs A_G, with a set of conditional probability distributions $\Theta = \{p(V \mid \pi_V) \mid V \in V_G\}$ that describe the strengths of the various dependences between a node V and its immediate predecessors π_V in the graph. More formally, the Bayesian network defines the unique distribution

$$p(V_G) = \prod_{V \in V_G} p(V \mid \pi_V)$$

on V_G, that respects the probabilistic independences read from the digraph G by means of the d-separation criterion [9]. As such, the network provides for computing any prior or posterior probability over its variables. Computing probabilities from Bayesian networks, also known as inference, is in general NP-hard [10]. However, inference in a Bayesian network whose directed graph takes the form of a tree, where every node has at most one parent, requires a number of computations which is linear in the number of nodes [9].

A *dynamic Bayesian network* can cope with discrete-time evolving processes by repeating and connecting a Bayesian network for a number of time steps, or *time slices* [2]. The relations among the variables within a time slice are taken to be instantaneous, whereas the relationships across time slices are considered temporal.

2.2 Hidden Markov models

In this section we review the necessary background on hidden Markov models (HMMs), their relation to dynamic Bayesian networks and the recursive properties that underlie inference in HMMs.

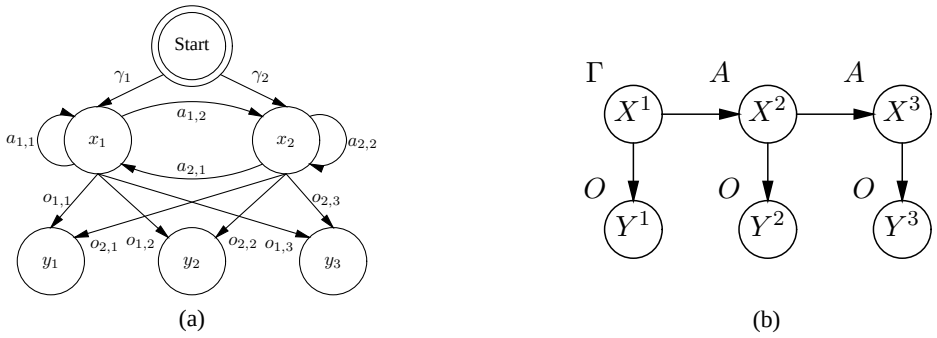


Figure 1: A hidden Markov model representation (a) and its dynamic Bayesian network representation, unrolled for three time slices (b).

2.2.1 Definition

A hidden Markov model [11, 12] consists of a discrete time Markov chain, repeating a single hidden variable X with a finite number of states. The chain is stationary, i.e. the probability of transitioning from one state to another is time-invariant. The state of the hidden variable in each time step can be indirectly observed by some memoryless test or sensor Y. The uncertainty in the discrete test or sensor output is captured by a set of observation probabilities, which are also time-invariant. Generalisations of HMMs with continuous variables exist, but are not considered here. More formally, an HMM is a statistical model $H = (X, Y, A, O, \Gamma)$, where

- variable X has $n \geq 2$ states, denoted by x_i, $1 \leq i \leq n$;
- variable Y has $m \geq 2$ states, denoted by y_j, $1 \leq j \leq m$;
- transition matrix A has entries $a_{i,j} = p(x_j \mid x_i)$, $1 \leq i, j \leq n$;
- observation matrix O has entries $o_{i,j} = p(y_j \mid x_i)$, $1 \leq i \leq n$, $1 \leq j \leq m$;
- initial vector Γ has entries $\gamma_i = p(x_i)$, $1 \leq i \leq n$.

Figure 1(a) shows an example HMM where X has two states, and Y three.

An HMM can be seen as a special case of a dynamic Bayesian network unrolled for a number of time slices (see for details [2, 3]). The time slice under consideration is explicitly indicated by a superscript for the variables and their values. More specifically, an HMM is a (dynamic) Bayesian network $\mathcal{H} = (G, \Theta)$, where

- $V_G = \{X^k, Y^k \mid 1 \leq k \leq t\}$ captures the two HMM variables X and Y repeated over t time steps;
- $A_G = \{X^k \rightarrow Y^k \mid 1 \leq k \leq t\} \cup \{X^{k-1} \rightarrow X^k \mid 2 \leq k \leq t\}$ captures the Markov property of the chain and the independence of the observations;
- Θ, the set of conditional probability distributions, is a union of

$$\{p(x_j^k \mid x_i^{k-1}) = a_{i,j} \mid 2 \leq k \leq t, 1 \leq i, j \leq n\},$$

$\{p(y_j^k \mid x_i^k) = o_{i,j} \mid 1 \leq k \leq t, 1 \leq i \leq n, 1 \leq j \leq m\}$, and
$\{p(x_i^1) = \gamma_i \mid 1 \leq i \leq n\}$

Figure 1(b) shows a dynamic Bayesian network representation of the HMM from Figure 1(a), unrolled for three time slices.

In the remainder of this paper we use the notation $\mathbf{y}_\mathbf{e}^t$ to indicate actual evidence for variable Y in time slice t, and $\mathbf{y}_\mathbf{e}^{t_i:t_j}$ to denote a sequence of observations $\mathbf{y}_\mathbf{e}^{t_i}, \ldots, \mathbf{y}_\mathbf{e}^{t_j}$.

2.2.2 Inference

Inference in temporal models typically amounts to computing the marginal distribution over X at time t, given the evidence up to and including time T, that is $p(X^t \mid \mathbf{y}_\mathbf{e}^{1:T})$. If $T = t$, this inference task is known as *filtering*, $T < t$ concerns *prediction* of a future state, and *smoothing* is the task of inferring the past, that is $T > t$. For exact inference in an HMM, the efficient *Forward-Backward algorithm* is available (see for details [13, chapter 15]). This algorithm computes for all hidden states i at time $t \leq T$, the following two probabilities:

- forward probability $F(i,t) = p(x_i^t, \mathbf{y}_\mathbf{e}^{1:t})$, and
- backward probability $B(i,t) = p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_i^t)$

resulting in

$$p(x_i^t \mid \mathbf{y}_\mathbf{e}^{1:T}) = \frac{p(x_i^t, \mathbf{y}_\mathbf{e}^{1:T})}{p(\mathbf{y}_\mathbf{e}^{1:T})} = \frac{p(x_i^t, \mathbf{y}_\mathbf{e}^{1:T})}{\sum_{j=1}^{n} p(x_j^t, \mathbf{y}_\mathbf{e}^{1:T})} = \frac{F(i,t) \cdot B(i,t)}{\sum_{j=1}^{n} F(j,t) \cdot B(j,t)} \tag{1}$$

For $T < t$, the algorithm can be applied by taking $B(i,t) = 1$ and adopting the convention that the configuration of an empty set of observations is TRUE, i.e $\mathbf{y}_\mathbf{e}^{T+1:t} \equiv$ TRUE, resulting in

$$F(i,t) = p(x_i^t, \mathbf{y}_\mathbf{e}^{1:t}) = p(x_i^t, \mathbf{y}_\mathbf{e}^{1:T}, \text{TRUE}) = p(x_i^t, \mathbf{y}_\mathbf{e}^{1:T})$$

The three standard inference tasks of filtering, prediction and smoothing in hidden Markov models are all concerned with inferring the probability of a hidden state from a sequence of observations. Two other interesting tasks are the prediction of future observations, i.e. $p(\mathbf{y}_\mathbf{e}^t \mid \mathbf{y}_\mathbf{e}^{1:T})$ for $T < t$, and finding the most probable explanation, that is, $\arg\max_{x_i^{1:t}} p(x_i^{1:t} \mid \mathbf{y}_\mathbf{e}^{1:t})$. We will disregard the latter and briefly discuss the former. We note that the probability $p(\mathbf{y}_\mathbf{e}^t \mid \mathbf{y}_\mathbf{e}^{1:T})$, $T < t$, can be computed as the fraction of the two probabilities $p(\mathbf{y}_\mathbf{e}^t \mathbf{y}_\mathbf{e}^{1:T})$, $T < t$, and $p(\mathbf{y}_\mathbf{e}^{1:T})$; these, in turn, can be straightforwardly computed from forward probabilities:

$$p(\mathbf{y}_\mathbf{e}^{1:t}) = \sum_{i=1}^{n} p(x_i^t \mathbf{y}_\mathbf{e}^{1:t}) = \sum_{i=1}^{n} F(i,t)$$

Note that if $t > T + 1$ then $p(\mathbf{y}_\mathbf{e}^t \mathbf{y}_\mathbf{e}^{1:T})$ can be computed by setting all inbetween observations $\mathbf{y}_\mathbf{e}^k$, $T < k < t$, to TRUE as above.

The Forward-Backward algorithm has a $O(n^2 \cdot \max\{t, T\})$ computational complexity, where n is the number of hidden states of X. Alternatively, the HMM can be represented as a dynamic Bayesian network unrolled for $\max\{t, T\}$ time slices, upon which standard Bayesian network inference algorithms

can be used. In fact, Smyth, Heckerman and Jordan [3] have shown that the Forward-Backward algorithm can be seen as a special case of Pearl's Belief propagation algorithm for inference in Bayesian networks [9].

2.2.3 Recursive probability expressions

In this paper we present an algorithm for computing the coefficients of sensitivity functions for HMMs. This algorithm resembles the Forward-Backward algorithm for inference in HMMs and similarly exploits the repetitive character of the model parameters of an HMM. In this section we review the recursive expressions upon which the Forward-Backward algorithm is based (see e.g. [13, chapter 15]) and which are important for understanding the remainder of the paper.

Filtering We first consider a probability $p(x_v^t, \mathbf{y_e}^{1:t})$ for a specific state v of X, which we will call a *filter probability*. Note that this probability is the same as the forward probability $F(v,t)$ in the Forward-Backward algorithm.

For time slice $t = 1$ we simply have that

$$p(x_v^1, \mathbf{y_e}^1) = p(\mathbf{y_e}^1 \mid x_v^1) \cdot p(x_v^1) = o_{v,\boldsymbol{e^1}} \cdot \gamma_v$$

where $\boldsymbol{e^1}$ corresponds to the state of Y that is actually observed at time 1. For time slices $t > 1$ we exploit the fact that, given X^t, Y^t is independent of $Y^1, \ldots, Y^{t-1}$, written $Y^t \perp Y^{1:t-1} \mid X^t$, and find

$$p(x_v^t, \mathbf{y_e}^{1:t}) \;=\; p(x_v^t, \mathbf{y_e}^{1:t-1}, \mathbf{y_e}^t) \;=\; p(\mathbf{y_e}^t \mid x_v^t) \cdot p(x_v^t, \mathbf{y_e}^{1:t-1})$$

The first factor in the above product corresponds to an observation parameter; conditioning the second factor on the n states of X^{t-1} and exploiting the independence $X^t \perp Y^{1:t-1} \mid X^{t-1}$, we find

$$p(x_v^t, \mathbf{y_e}^{1:t-1}) \;=\; \sum_{z=1}^{n} p(x_v^t \mid x_z^{t-1}) \cdot p(x_z^{t-1}, \mathbf{y_e}^{1:t-1}) = \sum_{z=1}^{n} a_{z,v} \cdot p(x_z^{t-1}, \mathbf{y_e}^{1:t-1})$$

Taken together, we find for $F(v,t) = p(x_v^t, \mathbf{y_e}^{1:t})$ the recursive expression

$$F(v,t) = \begin{cases} o_{v,\boldsymbol{e^1}} \cdot \gamma_v & \text{if} \quad t = 1 \\ o_{v,\boldsymbol{e^t}} \cdot \sum_{z=1}^{n} a_{z,v} \cdot F(z, t-1) & \text{if} \quad t > 1 \end{cases} \tag{2}$$

Prediction We now consider a probability $p(x_v^t, \mathbf{y_e}^{1:T})$ with $t > T$. In Section 2.2.2 we noted that the Forward-Backward algorithm can be applied to compute such *prediction probabilities*, basically by prolonged filtering, i.e. computing $F(v,t)$ with an empty set of evidence for $Y^{T+1:t}$. This absence of evidence can be implemented by replacing, for $t > T$ in Equation 2, the term $o_{v,\boldsymbol{e^t}}$ by 1.

A special case of the prediction task, with $T = 0$, is the computation of a prior marginal $p(x_v^t)$. This probability can be computed as a filter probability with absence of evidence for all $Y^{1:t}$. Since the prediction task can thus be seen as a special case of the filtering task, we will refrain from explicitly considering prediction as a separate task in the remainder of this paper.

Smoothing Finally, we consider a probability $p(x_v^t,\, \mathbf{y}_\mathbf{e}^{1:T})$ with $t < T$, which we will call a *smoothing probability*. By exploiting the independence $Y^{t+1:T} \perp Y^{1:t} \mid X^t$, we have that

$$p(x_v^t,\, \mathbf{y}_\mathbf{e}^{1:T}) = p(x_v^t,\, \mathbf{y}_\mathbf{e}^{1:t},\, \mathbf{y}_\mathbf{e}^{t+1:T}) = p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t) \cdot p(x_v^t,\, \mathbf{y}_\mathbf{e}^{1:t}) \tag{3}$$

The second term in this product is again a filter probability. We now further focus on the first term, which is the same as the backward probability $B(v,t)$ in the Forward-Backward algorithm. By conditioning this term on X^{t+1} and exploiting the independences $X^t \perp Y^{t+1:T} \mid X^{t+1}$ and $Y^{t+1} \perp Y^{t+2:T} \mid X^{t+1}$ for $T > t+1$, we find that

$$\begin{aligned} p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t) &= \sum_{z=1}^{n} p(\mathbf{y}_\mathbf{e}^{t+1} \mid x_z^{t+1}) \cdot p(\mathbf{y}_\mathbf{e}^{t+2:T} \mid x_z^{t+1}) \cdot p(x_z^{t+1} \mid x_v^t) \\ &= \sum_{z=1}^{n} o_{z,\boldsymbol{e^{t+1}}} \cdot a_{v,z} \cdot p(\mathbf{y}_\mathbf{e}^{t+2:T} \mid x_z^{t+1}) \end{aligned}$$

For $t+1 = T$, this results in

$$p(\mathbf{y}_\mathbf{e}^{T:T} \mid x_v^{T-1}) = \sum_{z=1}^{n} p(\mathbf{y}_\mathbf{e}^{T} \mid x_z^{T}) \cdot p(x_z^{T} \mid x_v^{T-1}) = \sum_{z=1}^{n} o_{z,\boldsymbol{e^T}} \cdot a_{v,z}$$

Taken together, we find for $B(v,t) = p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t)$ the recursive expression

$$B(v,t) = \begin{cases} \displaystyle\sum_{z=1}^{n} o_{z,\boldsymbol{e^T}} \cdot a_{v,z} & \text{if} \quad t = T-1 \\ \displaystyle\sum_{z=1}^{n} o_{z,\boldsymbol{e^{t+1}}} \cdot a_{v,z} \cdot B(z,t+1) & \text{if} \quad t < T-1 \end{cases} \tag{4}$$

2.3 Sensitivity analysis

The value of any probability of interest in a statistical model depends on the probability parameters specified for the model. To study the robustness of the computed output to possible inaccuracies in these parameters, a *sensitivity analysis* can be performed.

2.3.1 Sensitivity analysis in Bayesian networks

In the context of Bayesian networks, sensitivity analysis has been studied extensively [14, 15, 16, 17, 18, 19, 20, 21]. In a Bayesian network, a simple functional relationship exists between any parameter and any output probability of interest. This functional relationship is called the *sensitivity function*. More specifically, an N-way *sensitivity function*, describing the effect of simultaneously varying N parameters, is either an N-variate polynomial or an N-variate rational function, where each variable has degree at most one. For example, the 3-way sensitivity function relating a joint or marginal output probability $p(v)$ for a (set of) variable(s) V to three network parameters θ_i, $i = 1, 2, 3$, has the following form:

$$\begin{aligned} p(v)(\theta_1, \theta_2, \theta_3) &= c^{111} \cdot \theta_3 \cdot \theta_2 \cdot \theta_1 + c^{110} \cdot \theta_3 \cdot \theta_2 + c^{101} \cdot \theta_3 \cdot \theta_1 + c^{011} \cdot \theta_2 \cdot \theta_1 + \\ &\quad + c^{100} \cdot \theta_3 + c^{010} \cdot \theta_2 + c^{001} \cdot \theta_1 + c^{000} \end{aligned}$$

where c^{ijk} are constants with respect to the parameters. This form holds under the standard assumption of proportional co-variation of the other parameters from the same (conditional) distribution. That is, if a parameter $\theta = p(v_j \mid \pi)$ for a variable V is varied, then for each $i \neq j$, $p(v_i \mid \pi)(\theta) = p(v_i \mid \pi) \cdot \frac{1-\theta}{1-p(v_j|\pi)}$. For binary-valued V, co-variation simplifies to $p(v_i \mid \pi)(\theta) = 1 - \theta$.

A sensitivity function for a posterior probability of interest is a quotient of two polynomials, since $p(v \mid e) = p(v\, e)/p(e)$, and hence a rational function.

To determine an N-variate sensitivity function, an exponential number of 2^N coefficients need to be computed. This can be either done by computing the output probability of interest for 2^N different combinations of values for the N parameters, and solving the resulting system of 2^N linear equations [18]. Note that this approach requires us to perform inference an exponential number of times, and returns nothing more than the sensitivity function for the given probability of interest. A more efficient approach is to use specially tailored versions of the junction tree inference algorithm [17, 20]. Approaches that assume all N parameters are taken from the same conditional probability distribution are even more efficient [16], but irrelevant for this paper.

2.3.2 Sensitivity analysis in Hidden Markov models

In the context of HMMs, sensitivity analysis is usually performed by means of a perturbation analysis where a small change is applied to the parameters, upon which the output of interest is re-computed [4, 5]. The main difference between sensitivity analysis in Bayesian networks and that in hidden Markov models in essence is, that a single parameter in an HMM may occur multiple times when multiple time slices are considered. A one-way sensitivity analysis in an HMM, therefore, amounts to an N-way analysis in its Bayesian network representation, where N equals the number of time slices under consideration. It is therefore no surprise that for HMMs sensitivity functions are similar to those for Bayesian networks [6]. The difference with the general N-way function for Bayesian networks is that the N parameters are constrained to all be equal, which reduces the number of required constants from exponential to polynomial in N. For example, if the above mentioned parameters θ_i, $i = 1, 2, 3$, represent a single transition parameter $\theta \equiv \theta_1 = \theta_2 = \theta_3$ in time slices 1, 2, and 3, then the sensitivity function for output probability $p(v)$ reduces to

$$p(v)(\theta) = c_3 \cdot \theta^3 + c_2 \cdot \theta^2 + c_1 \cdot \theta + c_0$$

for constants c_i, $i = 0, \ldots, 3$.

We now summarise the known results for sensitivity functions in HMMs [6, 22]. For the joint probability of a hidden state and evidence as a function of a model parameter θ, we have the following univariate polynomial sensitivity function:

$$p(x_v^t,\, \mathbf{y}_\mathbf{e}^{1:T})(\theta) = \sum_{i=0}^{N} c_i \cdot \theta^i \tag{5}$$

where

$$N = \begin{cases} t-1 & \text{if} \quad \theta = a_{r,s} \text{ and } t \geq T \\ T & \text{if} \quad \theta = o_{r,s} \text{ and } v = r \\ T-1 & \text{if} \quad \theta = o_{r,s} \text{ and } v \neq r, \text{ or } \theta = a_{r,s}, t < T \text{ and } v = r \\ T-2 & \text{if} \quad \theta = a_{r,s},\ t < T \text{ and } v \neq r \\ 1 & \text{if} \quad \theta = \gamma_r \end{cases}$$

and the coefficients c_i are constant with respect to the various parameters. Coefficients c_i do depend on the hidden state v and time slice t under consideration; therefore we will often write $c^t_{v,i}$ rather than c_i in the remainder of this paper.

For prior marginals $p(x^t_v)$ over X as a function of a model parameter θ, we have the above form with $N = 0$ for $\theta = o_{r,s}$, $N = t - 1$ for $\theta = a_{r,s}$, and $N = 1$ for $\theta = \gamma_r$. For the probability of evidence $p(\mathbf{y}_\mathbf{e}^{1:T})$, we have that $N = T$ for observation parameters, $N = T - 1$ for transition parameters, and again $N = 1$ for initial parameters.

3 Sensitivity of HMM output to initial parameter variation

We saw in the previous section that one-way sensitivity functions for HMMs are polynomial in the parameter under consideration; in addition, we know the degree of the functions. However, we have yet to establish what the coefficients of these polynomials are and how to compute them. We will demonstrate in this section that since initial parameters are only used in the first time slice, it is quite straightforward to compute the coefficients of a sensitivity function for model parameter $\theta_\gamma = \gamma_r$. We will consider the sensitivity functions for the inference tasks mentioned in Section 2.2.2.

For ease of exposition concerning the co-variation of parameters, we assume in the remainder of this section that all variables are binary-valued, i.e. $n = m = 2$. Note that γ_v, the initial parameter associated with the state of interest v for X^t, now corresponds to either θ_γ (if $v = r$) or its complement $1 - \theta_\gamma$ (if $v \neq r$).

Filtering From the recursive expression for the filter probability in Equation 2 it follows that for $T = t = 1$,

$$p(x^1_v, \mathbf{y}^1_\mathbf{e})(\theta_\gamma) \;=\; \begin{cases} o_{v,\boldsymbol{e^1}} \cdot \theta_\gamma + 0 & \text{if} \quad v = r \\ -o_{v,\boldsymbol{e^1}} \cdot \theta_\gamma + o_{v,\boldsymbol{e^1}} & \text{if} \quad v \neq r \end{cases}$$

and for $T = t > 1$,

$$p(x^t_v, \mathbf{y}^{1:t}_\mathbf{e})(\theta_\gamma) = \sum_{z=1}^{2} o_{v,\boldsymbol{e^t}} \cdot a_{z,v} \cdot p(x^{t-1}_z, \mathbf{y}^{1:t-1}_\mathbf{e})(\theta_\gamma)$$

From Equation 5 we have that the polynomial $p(x^t_v, \mathbf{y}^{1:t}_\mathbf{e})(\theta_\gamma)$ requires two coefficients: $c^t_{v,1}$ and $c^t_{v,0}$. Since each initial parameter is used only in time slice 1, as the above expressions demonstrate, the coefficients for $T = t > 1$ can be established through a simple recursion for each $N = 0, 1$:

$$c^t_{v,N} = \sum_{z=1}^{2} o_{v,\boldsymbol{e^t}} \cdot a_{z,v} \cdot c^{t-1}_{z,N}$$

with $c^1_{v,0} = 0$ if $v = r$, and $c^1_{v,0} = o_{v,\boldsymbol{e^1}}$ otherwise; in addition $c^1_{v,1} = o_{v,\boldsymbol{e^1}}$ if $v = r$, and $c^1_{v,1} = -o_{v,\boldsymbol{e^1}}$ otherwise.

Smoothing In case $T > t$, we have from Equation 3 for the smoothing probability that we need to multiply the functions $p(x^t_v, \mathbf{y}^{1:t}_\mathbf{e})(\theta_\gamma)$ and $p(\mathbf{y}^{t+1:T}_\mathbf{e} \mid x^t_v)(\theta_\gamma)$. Since $Y^{t+1:T} \perp X^1 \mid X^t$ for $1 \leq t < T$, the probability $p(\mathbf{y}^{t+1:T}_\mathbf{e} \mid x^t_v)$ is not affected by changes in the initial parameters. Hence the function $p(\mathbf{y}^{t+1:T}_\mathbf{e} \mid x^t_v)(\theta_\gamma)$ is simply a constant probability, which can be computed using standard inference.

Predicting future observations In Section 2.2.2 we mentioned the prediction of future observations as another interesting inference task. We showed that the probability of a certain observation at time t can be computed straightforwardly by using forward (filter) probabilities for time slices 1 through t. The coefficients for the sensitivity function $p(\mathbf{y}_\mathbf{e}^t \mid \mathbf{y}_\mathbf{e}^{1:T})(\theta_\gamma)$, $T < t$, can therefore be established by computing the coefficients of the functions $p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta_\gamma)$ for all n hidden states v, and summing the coefficients corresponding to terms of the same degree.

For determining sensitivity of output probabilities to variations in transition parameters or observation parameters, we need a more complex procedure, which is introduced in the next section.

4 The Coefficient-Matrix-Fill procedure

To compute the coefficients of the polynomial sensitivity function in Equation 5 for transition and observation parameters, we designed a procedure which basically constructs a set of matrices containing these coefficients for each hidden state and each time slice. We call this procedure the *Coefficient-Matrix-Fill* procedure. In this section we describe the basic idea of the procedure, the operations it uses and discuss its complexity.

4.1 The basic idea

For sensitivity functions $p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta)$ related to a filter probability, we have from Equation 5 that we need to establish coefficients $c_{v,j}^t$, $j = 0, \ldots, N$, where $N = t-1$ for a transition parameter θ_a and $N = T$ for an observation parameter θ_o. To compute these coefficients, we construct a series of "Forward" matrices F^k, $k = 1, \ldots, N+1$, with the following properties:

- each matrix F^k has size $n \times k$ for $\theta = \theta_a$, or size $n \times (k+1)$ for $\theta = \theta_o$;
- a row i in F^k contains exactly the coefficients for the function $p(x_i^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta)$;
- a column j in F^k contains all coefficients of the $(j-1)$th-order terms of the n polynomials.

More specifically, entry $f_{i,j}^k$ equals the coefficient $c_{i,j-1}^k$ of the sensitivity function $p(x_i^k, \mathbf{y}_\mathbf{e}^{1:k})(\theta)$. The Coefficient-Matrix-Fill procedure therefore in fact computes the coefficients for the sensitivity functions for *all* n hidden states and *all* time slices up to and including t.

From Equation 3 we have that for sensitivity functions related to a smoothing probability, we require the computation of a series of "Backward" matrices B^k, in addition to the forward matrices for the filter component. More specifically, matrices B^k will serve to compute the coefficients of the function $p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t)(\theta)$. This function is again a univariate polynomial for each model parameter.[1]

Proposition 4.1. *Let $H = (X, Y, A, O, \Gamma)$ be an HMM as before. Consider a probability of interest $p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t)$ with $T > t$, and let θ be a parameter from A, O, or Γ in H. Then, the one-way sensitivity function $p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t)(\theta)$ equals*

$$p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x_v^t)(\theta) = d_{v,N}^t \cdot \theta^N + \ldots + d_{v,1}^t \cdot \theta + d_{v,0}^t$$

[1] Note that this may seem counter-intuitive as it concerns the function for a *conditional* probability and should therefore be a quotient of polynomials; since X^t is an ancestor of $Y^{t+1} \ldots Y^T$, however, the factorisation of $p(\mathbf{y}_\mathbf{e}^{t+1:T}, x_v^t)$ includes $p(x_v^t)$, which cancels out the denominator.

where coefficients $d^t_{v,N}, \ldots, d^t_{v,0}$ are constants with respect to θ, and

$$N = \begin{cases} T - t & \textit{if} \quad \theta = o_{r,s} \textit{ or } \theta = a_{r,s} \\ 0 & \textit{if} \quad \theta = \gamma_r \end{cases}$$

Proof. The fact that the function under consideration is a univariate polynomial in θ, follows directly from the recursive expression for the backward probability in Equation 4. Moreover, from Equation 3 we have that the degree of $p(x^t_v, \mathbf{y}_\mathbf{e}^{1:T})(\theta)$ equals the sum of the degrees of $p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x^t_v)(\theta)$ and $p(x^t_v, \mathbf{y}_\mathbf{e}^{1:t})(\theta)$. The degrees of both $p(x^t_v, \mathbf{y}_\mathbf{e}^{1:T})(\theta)$ and $p(x^t_v, \mathbf{y}_\mathbf{e}^{1:t})(\theta)$ are given in Equation 5; the degree of $p(\mathbf{y}_\mathbf{e}^{t+1:T} \mid x^t_v)(\theta)$ can be directly established as their difference. □

For matrices B^k, the Coefficient-Matrix-Fill procedure should again establish the coefficients of a univariate polynomial function in θ; we assume that θ is either a transition parameter or an observation parameter, since initial parameters were already discussed in Section 3. To compute these coefficients $d^t_{v,j}$, $j = 0, \ldots, N$, we construct $N + 1 = T - t + 1$ matrices B^k, $k = t, \ldots, T$. Each matrix B^k has size $n \times (T - k + 1)$, where entry $b^k_{i,j}$ equals the coefficient $d^k_{i,j-1}$ of the function $p(\mathbf{y}_\mathbf{e}^{k+1:T} \mid x^k_v)(\theta)$.

4.2 Initialisation and fill operations

The Coefficient-Matrix-Fill procedure starts by filling the entries of matrix F^1 in accordance with the $t = 1$ case in the recursive expression for filter probabilities (Equation 2); matrix B^T is filled with all 1's. All other matrices F^k, $k > 1$, and B^k, $t \leq k < T$, are initialised with zeroes and subsequently filled with their correct contents by the procedure.

In Section 5 it will become clear that the matrices F^k for $k > 1$ are built solely from the entries in F^{k-1}, the transition matrix A and the observation matrix O; a similar observation applies to matrices B^k for $k < T$. We will now discuss the basic operations required to fill the matrices. We focus on the "Forward" matrices F^k, with similar observations applying to the "Backward" matrices B^k. The Coefficient-Matrix-Fill procedure basically implements the recursive steps in the various formulas from Section 2.2.3 by transitioning from matrix F^k to F^{k+1}. To illustrate this transition, consider an arbitrary $(k-1)$th-degree polynomial in θ,

$$p(\theta) = c_{k-1} \cdot \theta^{k-1} + \ldots + c_1 \cdot \theta + c_0$$

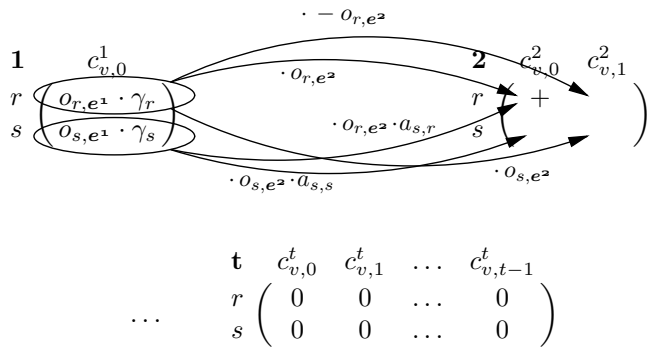
and let the coefficients of this polynomial be represented in row i of matrix F^k, i.e. $f^k_{i,\cdot} = \langle c_0, \ldots, c_{k-1} \rangle$. In transitioning from matrix F^k to F^{k+1}, three types of operation (or combinations thereof) can be applied to $p(\theta)$:

(I) summation with another polynomial $p'(\theta)$ of the same degree;

(II) multiplication with a constant d;

(III) multiplication with θ.

Case (I) just requires summing the coefficients of the same order, i.e. adding entries with the same column number. In case (II), the resulting polynomial is represented in row i of matrix F^{k+1} by $f^{k+1}_{i,\cdot} = \langle d \cdot c_0, \ldots, d \cdot c_{k-1}, 0 \rangle$; note that F^{k+1} has an additional column $k + 1$, which is unaffected by this operation. In case (III) the resulting kth-degree polynomial is represented in row i of matrix F^{k+1} by $f^{k+1}_{i,\cdot} = \langle 0, c_0, \ldots, c_{k-1} \rangle$; this operation basically amounts to shifting entries from F^k one column to the right. The global idea behind the Coefficient-Matrix-Fill procedure is illustrated in Figure 2.



$$\cdots \quad \begin{array}{c} \\ r \\ s \end{array} \begin{array}{c} \mathbf{t} \\ \left(\begin{array}{c} 0 \\ 0 \end{array} \right. \end{array} \begin{array}{cccc} c^t_{v,0} & c^t_{v,1} & \cdots & c^t_{v,t-1} \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{array} \left. \begin{array}{c} \\ \\ \end{array} \right)$$

Figure 2: An example of transitioning from matrix F^1 to F^2 in the Coefficient-Matrix-Fill procedure; here constants of the sensitivity function relating a filter probability to a transition parameter $\theta_a = a_{r,s}$ are computed.

4.3 Posterior probabilities

Recall that the inference tasks of filtering, prediction and smoothing actually concern the computation of posterior probabilities. Since a posterior probability $p(x^t_i \mid \mathbf{y}_\mathbf{e}^{1:T})$ can be immediately established from $p(x^t_i, \mathbf{y}_\mathbf{e}^{1:T})$ and $p(\mathbf{y}_\mathbf{e}^{1:T}) = \sum_{j=1}^{n} p(x^t_j, \mathbf{y}_\mathbf{e}^{1:T})$ (see Equation 1), the matrices constructed by the Coefficient-Matrix-Fill procedure contain all information for establishing the sensitivity function for the probability of evidence, and hence for the sensitivity function for a posterior probability.

4.4 Complexity

In case $T \leq t$, the Coefficient-Matrix-Fill procedure fills at most $t+1$ matrices of increasing sizes $n \times k$, $k = 1, \ldots, t+1$. Each matrix contains the coefficients for the functions $p(x^k_i, \mathbf{y}_\mathbf{e}^{1:k})(\theta)$ for all i, so the procedure computes the coefficients for the sensitivity functions for *all* hidden states and *all* time slices up to and including t. If we are interested in only one specific time slice t, then we can exploit the fact that each matrix F^k only requires information stored in matrix F^{k-1}, and therefore save space by storing only two matrices at all times. In case $T > t$, the Coefficient-Matrix-Fill procedure in addition fills $T - t + 1$ matrices of increasing sizes $n \times k$, $k = 1, \ldots, T - t + 1$.

The runtime complexity for a straightforward implementation of the algorithm is $O(n^2 \cdot \max\{t, T\}^2)$, which is $\max\{t, T\}$ times that of the Forward-Backward algorithm. This is due to the fact that per hidden state we need to compute k numbers per time step rather than a single one.

5 Sensitivity to transition and observation parameters

In this section we describe in detail how the Coefficient-Matrix-Fill procedure computes the coefficients of one-way sensitivity functions for transition parameters and for observation parameters. For ease of exposition concerning the co-variation of parameters, we again assume in the remainder of this section that all variables are binary-valued, i.e. $n = m = 2$.

5.1 Sensitivity of filtering to transition parameter variation

In this section we consider sensitivity functions $p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta_a)$ for a *filter* probability and *transition* parameter $\theta_a = a_{r,s}$. From the recursive expression for filter probabilities (Equation 2), it follows that for $t = 1$ we have a constant:

$$p(x_v^1, \mathbf{y}_\mathbf{e}^1)(\theta_a) = o_{v,\boldsymbol{e^1}} \cdot \gamma_v \tag{6}$$

and for $t > 1$,

$$p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta_a) = o_{v,\boldsymbol{e^t}} \cdot \sum_{z=1}^{2} a_{z,v}(\theta_a) \cdot p(x_z^{t-1}, \mathbf{y}_\mathbf{e}^{1:t-1})(\theta_a)$$

Recall that $\theta_a = a_{r,s}$; therefore, in the above formula, $a_{r,v}(\theta_a)$ equals θ_a for $v = s$ and $1 - \theta_a$ for $v \neq s$; $a_{z,v}$ for $z \neq r$ is independent of θ_a. As a result we conclude that for $t > 1$,

$$p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta_a) = \tag{7}$$

$$\begin{cases} o_{v,\boldsymbol{e^t}} \cdot \theta_a \cdot p(x_r^{t-1}, \mathbf{y}_\mathbf{e}^{1:t-1})(\theta_a) + \; o_{v,\boldsymbol{e^t}} \cdot a_{\overline{r},v} \cdot p(x_{\overline{r}}^{t-1}, \mathbf{y}_\mathbf{e}^{1:t-1})(\theta_a) & \text{if } \; v = s \\ o_{v,\boldsymbol{e^t}} \cdot (1 - \theta_a) \cdot p(x_r^{t-1}, \mathbf{y}_\mathbf{e}^{1:t-1})(\theta_a) + \; o_{v,\boldsymbol{e^t}} \cdot a_{\overline{r},v} \cdot p(x_{\overline{r}}^{t-1}, \mathbf{y}_\mathbf{e}^{1:t-1})(\theta_a) & \text{if } \; v \neq s \end{cases}$$

where $\overline{r}$ denotes the state of X other than r.

From Equation 5, we have that the polynomial $p(x_v^t, \mathbf{y}_\mathbf{e}^{1:t})(\theta_a)$ requires t coefficients: $c_{v,N}^t$, $N = 0, \ldots, t-1$. To compute these coefficients, the Coefficient-Matrix-Fill procedure builds upon Equations 6 and 7 above to fill its matrices. We will now describe the details of the fill contents of the matrices.

Fill contents: initialisation The $n \times 1$ matrix F^1 is initialised by setting, for $i = 1, 2$, $f_{i,1}^1 = o_{i,\boldsymbol{e^1}} \cdot \gamma_i$. The remaining matrices F^k of size $n \times k$, $2 \leq k \leq t$, are initialised by filling them with zeroes.

Fill contents: $F^k, k = 2, \ldots, t$ Column j of matrix F^k should be filled using elements from the jth column of F^{k-1} that are summed or multiplied with a constant, and elements from the $(j-1)$th column of F^{k-1} that are multiplied with θ_a. More specifically, following Equation 7, position j in row i of matrix F^k, $f_{i,j}^k$, is filled with

$$\begin{array}{ll} o_{i,\boldsymbol{e^k}} \cdot (f_{r,j-1}^{k-1} + a_{\overline{r},i} \cdot f_{\overline{r},j}^{k-1}) & \text{if } i = s \text{ and } 1 < j < k \\ o_{i,\boldsymbol{e^k}} \cdot (-f_{r,j-1}^{k-1} + f_{r,j}^{k-1} + a_{\overline{r},i} \cdot f_{\overline{r},j}^{k-1}) & \text{if } i \neq s \text{ and } 1 < j < k \end{array}$$

For $j = 1$, these general cases are simplified by setting $f_{r,j-1}^{k-1} = 0$. This boundary condition captures the property that entries in the first column correspond to coefficients of the zero-order terms of the polynomials and can therefore never result from a multiplication with θ_a. Similarly, since the coefficients for column $j = k$, $k > 1$, can *only* result from multiplication by θ_a, we set $f_{\cdot,j}^{k-1} = 0$ in that case.

Example 5.1. *Consider an HMM with binary-valued hidden state X and binary-valued evidence variable Y. Let $\Gamma = [0.20, 0.80]$ be the initial vector for X^1, and let transition matrix A and observation matrix O be as follows:*

$$A = \begin{bmatrix} 0.95 & 0.05 \\ 0.15 & 0.85 \end{bmatrix} \textit{ and } O = \begin{bmatrix} 0.75 & 0.25 \\ 0.90 & 0.10 \end{bmatrix}$$

Suppose we are interested in the sensitivity functions for the two states of X^3 as a function of transition parameter $\theta_a = a_{2,1} = p(x_1^t \mid x_2^{t-1}) = 0.15$, for all $t > 1$. Suppose the following sequence of observations is obtained: $y_{\mathbf{2}}^1$, $y_{\mathbf{1}}^2$ and $y_{\mathbf{1}}^3$. To compute the coefficients for the sensitivity functions, the following matrices are constructed by the Coefficient-Matrix-Fill procedure:

$$F^1 = \begin{bmatrix} o_{1,2}\cdot\gamma_1 \\ o_{2,2}\cdot\gamma_2 \end{bmatrix} = \begin{bmatrix} 0.25\cdot 0.20 \\ 0.10\cdot 0.80 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

$$F^2 = \begin{bmatrix} o_{1,1}\cdot a_{1,1}\cdot f_{1,1}^1 & o_{1,1}\cdot f_{2,1}^1 \\ o_{2,1}\cdot(f_{2,1}^1 + a_{1,2}\cdot f_{1,1}^1) & -o_{2,1}\cdot f_{2,1}^1 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.75\cdot 0.95\cdot 0.05 & 0.75\cdot 0.08 \\ 0.90\cdot(0.08 + 0.05\cdot 0.05) & -0.90\cdot 0.08 \end{bmatrix} = \begin{bmatrix} 0.03563 & 0.060 \\ 0.07425 & -0.072 \end{bmatrix}$$

and finally,

$$F^3 = \begin{bmatrix} o_{1,1}\cdot a_{1,1}\cdot f_{1,1}^2 & o_{1,1}\cdot(f_{2,1}^2 + a_{1,1}\cdot f_{1,2}^2) & o_{1,1}\cdot f_{2,2}^2 \\ o_{2,1}\cdot(f_{2,1}^2 + a_{1,2}\cdot f_{1,1}^2) & o_{2,1}\cdot(-f_{2,1}^2 + f_{2,2}^2 + a_{1,2}\cdot f_{1,2}^2) & -o_{2,1}\cdot f_{2,2}^2 \end{bmatrix} =$$

$$= \begin{bmatrix} 0.02538 & 0.09844 & -0.0540 \\ 0.06843 & -0.12893 & 0.0648 \end{bmatrix}$$

We now find for example from F^3 that

$$p(x_1^3,\, \mathbf{y}_\mathbf{e}^{1:3})(\theta_a) = 0.02538 + 0.09844\cdot\theta_a - 0.054\cdot\theta_a^2$$

and from F^2 that

$$p(x_2^2,\, \mathbf{y}_\mathbf{e}^{1:2})(\theta_a) = 0.07425 - 0.072\cdot\theta_a$$

Likewise, by summing column entries, we can establish the coefficients for the probability of evidence functions:

$$p(\mathbf{y}_\mathbf{e}^{1:3})(\theta_a) = (f_{1,1}^3 + f_{2,1}^3) + (f_{1,2}^3 + f_{2,2}^3)\cdot\theta_a + (f_{1,3}^3 + f_{2,3}^3)\cdot\theta_a^2$$

and

$$p(\mathbf{y}_\mathbf{e}^{1:2})(\theta_a) = (f_{1,1}^2 + f_{2,1}^2) + (f_{1,2}^2 + f_{2,2}^2)\cdot\theta_a$$

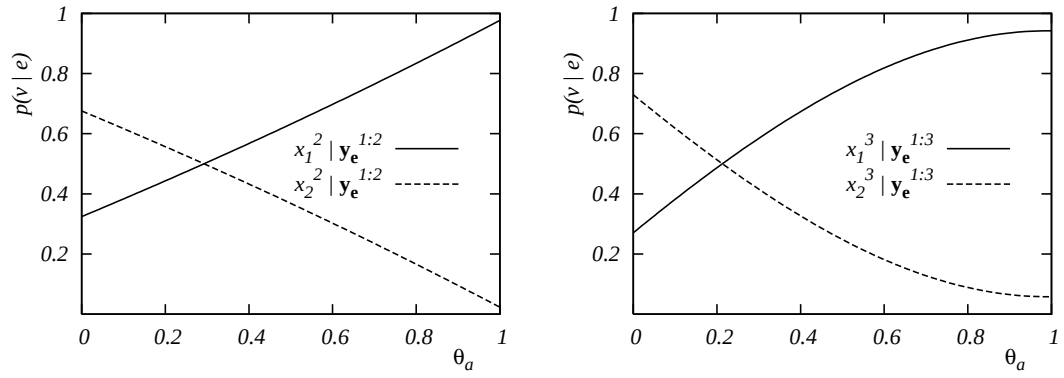
Together these give the following sensitivity functions for two filtering tasks:

$$p(x_1^3 \mid \mathbf{y}_\mathbf{e}^{1:3})(\theta_a) = \frac{-0.054\cdot\theta_a^2 + 0.09844\cdot\theta_a + 0.02538}{0.0108\cdot\theta_a^2 - 0.03049\cdot\theta_a + 0.09381}$$

and

$$p(x_2^2 \mid \mathbf{y}_\mathbf{e}^{1:2})(\theta_a) = \frac{-0.072\cdot\theta_a + 0.07425}{-0.012\cdot\theta_a + 0.10988}$$

which are displayed in Figure 3. □



(a) (b)

Figure 3: Sensitivity functions $p(X^2 \mid \mathbf{y_e}^{1:2})(\theta_a)$ for both states of X^2 (a), and $p(X^3 \mid \mathbf{y_e}^{1:3})(\theta_a)$ for both states of X^3 (b).

5.2 Sensitivity of filtering to observation parameter variation

In this section we consider sensitivity functions $p(x_v^t, \mathbf{y_e}^{1:t})(\theta_o)$ for a *filter* probability and *observation* parameter $\theta_o = o_{r,s}$. From the recursive expression for filter probabilities (Equation 2), it follows that for $t = 1$,

$$p(x_v^1, \mathbf{y_e}^1)(\theta_o) \quad = \quad \begin{cases} o_{v,\boldsymbol{e^1}} \cdot \gamma_v & \text{if} \quad v \neq r \\ \theta_o \cdot \gamma_r & \text{if} \quad v = r \text{ and } \boldsymbol{e^1} = s \\ (1-\theta_o) \cdot \gamma_r & \text{if} \quad v = r \text{ and } \boldsymbol{e^1} \neq s \end{cases} \tag{8}$$

and for $t > 1$,

$$p(x_v^t, \mathbf{y_e}^{1:t})(\theta_o) = o_{v,\boldsymbol{e^t}}(\theta_o) \cdot \sum_{z=1}^{2} a_{z,v} \cdot p(x_z^{t-1}, \mathbf{y_e}^{1:t-1})(\theta_o) \tag{9}$$

where, $o_{v,\boldsymbol{e^t}}(\theta_o)$ equals $o_{v,\boldsymbol{e^t}}$ for $v \neq r$, θ_o for $v = r$ and $\boldsymbol{e^t} = s$, and $1 - \theta_o$ for $v = r$ and $\boldsymbol{e^t} \neq s$, as above.

From Equation 5 we have that the polynomial function $p(x_v^t, \mathbf{y_e}^{1:t})(\theta_o)$ requires $t + 1$ coefficients: $c_{v,N}^t$, $N = 0, \ldots, t$. To compute these coefficients, the Coefficient-Matrix-Fill procedure builds upon Equations 8 and 9 above to fill its matrices. We will now describe the details of the fill contents of the matrices.

Fill contents: initialisation The $n \times 2$ matrix F^1 is initialised in accordance with Equation 8, i.e. row $f_{i,\cdot}^1$ is filled with

$$\begin{array}{ll} \langle o_{v,\boldsymbol{e^1}} \cdot \gamma_v, 0 \rangle & \text{if } i \neq r \\ \langle 0, \gamma_r \rangle & \text{if } i = r \text{ and } \boldsymbol{e^1} = s \\ \langle \gamma_r, -\gamma_r \rangle & \text{if } i = r \text{ and } \boldsymbol{e^1} \neq s \end{array}$$

The remaining matrices F^k, $2 \leq k \leq t$, are $n \times (k+1)$ matrices, which are initialised by filling them with zeroes.

Fill contents: $F^k, k = 2, \ldots, t$ Following Equation 9, position j in row i of matrix F^k, $f^k_{i,j}$, is filled with the following for $1 < j < k + 1$:

$$\begin{array}{ll} \sum_{z=1}^{2} a_{z,i} \cdot f^{k-1}_{z,j-1} & \text{if } i = r \text{ and } \boldsymbol{e^k} = s \\ \sum_{z=1}^{2} a_{z,i} \cdot (f^{k-1}_{z,j} - f^{k-1}_{z,j-1}) & \text{if } i = r \text{ and } \boldsymbol{e^k} \neq s \\ o_{\overline{r},\boldsymbol{e^k}} \cdot \sum_{z=1}^{2} a_{z,\overline{r}} \cdot f^{k-1}_{z,j} & \text{if } i \neq r \end{array}$$

For $j = 1$ and $j = k + 1$ we again simplify the above formulas where necessary, to take into account boundary conditions. More specifically, for $j = 1$ we set $f^{k-1}_{.,j-1} = 0$, and for $j = k + 1$ we set $f^{k-1}_{.,j} = 0$.

5.3 Sensitivity of smoothing to transition parameter variation

In this section we consider the sensitivity function $p(x^t_v, \mathbf{y}^{1:T}_{\mathbf{e}})(\theta_a)$, $T > t$, for a *smoothing* probability and *transition* parameter $a_{r,s}$. From Equation 3 we have that the coefficients of this polynomial can be established by standard polynomial multiplication of the polynomial functions $p(x^t_v, \mathbf{y}^{1:t}_{\mathbf{e}})(\theta_a)$ and $p(\mathbf{y}^{t+1:T}_{\mathbf{e}} \mid x^t_v)(\theta_a)$. Since the former is again a sensitivity function for a filter probability, we will further focus on the latter.

Consider the recursive expression for backward probabilities (Equation 4) and $a_{v,z}(\theta_a)$ with $\theta_a = a_{r,s}$. If $v = r$ then $a_{v,z}(\theta_a)$ equals θ_a for $z = s$, and $1 - \theta_a$ for $z \neq s$; otherwise $a_{v,z}(\theta_a)$ is constant. We now have that for $t = T - 1$,

$$p(\mathbf{y}^{T:T}_{\mathbf{e}} \mid x^T_v)(\theta_a) \;=\; \begin{cases} \sum_{z=1}^{2} o_{z,\boldsymbol{e^T}} \cdot a_{v,z} & \text{if} \quad v \neq r \\ o_{s,\boldsymbol{e^T}} \cdot \theta_a + o_{\overline{s},\boldsymbol{e^T}} \cdot (1 - \theta_a) & \text{if} \quad v = r \end{cases} \tag{10}$$

where $\overline{s}$ denotes the state of X other than s. For $t < T - 1$ we find

$$p(\mathbf{y}^{t+1:T}_{\mathbf{e}} \mid x^t_v)(\theta_a) \;= \tag{11}$$

$$\begin{cases} \sum_{z=1}^{2} o_{z,\boldsymbol{e^{t+1}}} \cdot a_{v,z} \cdot p(\mathbf{y}^{t+2:T}_{\mathbf{e}} \mid x^{t+1}_z)(\theta_a) & \text{if} \quad v \neq r \\ o_{s,\boldsymbol{e^{t+1}}} \cdot \theta_a \cdot p(\mathbf{y}^{t+2:T}_{\mathbf{e}} \mid x^{t+1}_s)(\theta_a) + o_{\overline{s},\boldsymbol{e^{t+1}}} \cdot (1 - \theta_a) \cdot p(\mathbf{y}^{t+2:T}_{\mathbf{e}} \mid x^{t+1}_{\overline{s}})(\theta_a) & \text{if} \quad v = r \end{cases}$$

From Proposition 4.1 we have that the polynomial $p(\mathbf{y}^{t+1:T}_{\mathbf{e}} \mid x^t_v)(\theta_a)$, $t < T$, requires $T - t + 1$ coefficients. To compute these coefficients, the Coefficient-Matrix-Fill procedure builds upon Equations 10 and 11 above to fill its matrices. Note the similarity between the equations: if $p(\mathbf{y}^{t+2:T}_{\mathbf{e}} \mid x^{t+1}_z)(\theta_a)$ in Equation 11 is replaced by 1, the expressions in Equation 10 result. We will now describe the details of the fill contents of the matrices, where the similarity between Equations 10 and 11 is exploited by using an additional matrix B^T.

Fill contents: initialisation The $n \times 1$ matrix B^T is initialised with 1's. The remaining matrices B^k, $t \leq k \leq T-1$, are $n \times (T-k+1)$ matrices which are initialised with zeroes.

Fill contents: $B^k, k = T-1$ down to t Following Equations 10 and 11, position j in row i of matrix B^k, $b^k_{i,j}$, is filled with the following for $1 < j < T-k+1$:

$$\begin{cases} o_{s,\boldsymbol{e^{k+1}}} \cdot b^{k+1}_{s,j-1} + o_{\bar{s},\boldsymbol{e^{k+1}}} \cdot \left(b^{k+1}_{\bar{s},j} - b^{k+1}_{\bar{s},j-1}\right) & \text{if } i = r \\ \sum_{z=1}^{2} o_{z,\boldsymbol{e^{k+1}}} \cdot a_{i,z} \cdot b^{k+1}_{z,j} & \text{if } i \neq r \end{cases}$$

We again have to take into account boundary conditions, that is, for $j = 1$ we set $b^{k+1}_{\cdot,j-1} = 0$ and for $j = T-k+1$ we set $b^{k+1}_{\cdot,j} = 0$.

5.4 Sensitivity of smoothing to observation parameter variation

In this section we consider the sensitivity function $p(x^t_v, \mathbf{y}_{\mathbf{e}}^{1:T})(\theta_o)$, $T > t$, for a *smoothing* probability and *observation* parameter $o_{r,s}$. For reasons explained above, we will to this end focus on the function $p(\mathbf{y}_{\mathbf{e}}^{t+1:T} \mid x^t_v)(\theta_o)$.

Consider the recursive expression for backward probabilities (Equation 4) and $o_{z,\boldsymbol{e^{t+1}}}(\theta_o)$ with $\theta_o = o_{r,s}$: if $z = r$ then $o_{z,\boldsymbol{e^{t+1}}}(\theta_o)$ equals θ_o for $\boldsymbol{e^{t+1}} = s$, and $1-\theta_a$ for $\boldsymbol{e^{t+1}} \neq s$; otherwise $o_{z,\boldsymbol{e^{t+1}}}(\theta_o)$ is constant. We now have that for $t = T-1$,

$$p(\mathbf{y}_{\mathbf{e}}^{T:T} \mid x^T_v)(\theta_o) \;=\; \begin{cases} \theta_o \cdot a_{v,r} + o_{\overline{r},\boldsymbol{e^T}} \cdot a_{v,\overline{r}} & \text{if} \quad \boldsymbol{e^T} = s \\ (1-\theta_o) \cdot a_{v,r} + o_{\overline{r},\boldsymbol{e^T}} \cdot a_{v,\overline{r}} & \text{if} \quad \boldsymbol{e^T} = \overline{s} \end{cases} \tag{12}$$

where $\overline{r}$ denotes the state of X other than r and $\overline{s}$ denotes the state of Y other than s. For $t < T-1$ we find,

$$p(\mathbf{y}_{\mathbf{e}}^{t+1:T} \mid x^t_v)(\theta_a) = \tag{13}$$

$$= \begin{cases} \theta_o \cdot a_{v,r} \cdot p(\mathbf{y}_{\mathbf{e}}^{t+2:T} \mid x^{t+1}_r)(\theta_o) + o_{\overline{r},\boldsymbol{e^{t+1}}} \cdot a_{v,\overline{r}} \cdot p(\mathbf{y}_{\mathbf{e}}^{t+2:T} \mid x^{t+1}_{\overline{r}})(\theta_o) & \text{if} \quad \boldsymbol{e^{t+1}} = s \\ (1-\theta_o) \cdot a_{v,r} \cdot p(\mathbf{y}_{\mathbf{e}}^{t+2:T} \mid x^{t+1}_r)(\theta_o) + o_{\overline{r},\boldsymbol{e^{t+1}}} \cdot a_{v,\overline{r}} \cdot p(\mathbf{y}_{\mathbf{e}}^{t+2:T} \mid x^{t+1}_{\overline{r}})(\theta_o) & \text{if} \quad \boldsymbol{e^{t+1}} = \overline{s} \end{cases}$$

From Proposition 4.1, we have that the polynomial $p(\mathbf{y}_{\mathbf{e}}^{t+1:T} \mid x^t_v)(\theta_o)$, $t < T$, requires $T-t+1$ coefficients. To compute these coefficients, the Coefficient-Matrix-Fill procedure builds upon Equations 12 and 13 above to fill its matrices. We will now describe the details of the fill contents of the matrices.

Fill contents: initialisation The $n \times 1$ matrix B^T is again initialised with 1's. All B^k, $t \leq k \leq T-1$, are $n \times (T-k+1)$ matrices which are initialised with zeroes.

Fill contents: $B^k, k = T - 1$ **down to** t Following Equations 12 and 13, position j in row i of matrix B^k, $b^k_{i,j}$, is filled with the following for $1 < j < T - k + 1$:

$$\begin{aligned} &a_{i,r} \cdot b^{k+1}_{r,j-1} + o_{\overline{r},\boldsymbol{e^{k+1}}} \cdot a_{i,\overline{r}} \cdot b^{k+1}_{\overline{r},j} && \text{if } \boldsymbol{e^{k+1}} = s \\ &-a_{i,r} \cdot b^{k+1}_{r,j-1} + a_{i,r} \cdot b^{k+1}_{r,j} + o_{\overline{r},\boldsymbol{e^{k+1}}} \cdot a_{i,\overline{r}} \cdot b^{k+1}_{\overline{r},j} && \text{if } \boldsymbol{e^{k+1}} \neq s \end{aligned}$$

We again take into account the boundary conditions by setting $b^{k+1}_{\cdot,j-1} = 0$ for $j = 1$ and $b^{k+1}_{\cdot,j} = 0$ for $j = T - k + 1$.

5.5 Sensitivity of predicted future observations

Sensitivity functions of the form $p(\mathbf{y}^t_{\mathbf{e}} \mid \mathbf{y}^{1:T}_{\mathbf{e}})(\theta)$, $T < t$, that describe the effects of parameter variation on the probability of a future observation can be straightforwardly established with the Coefficient-Matrix-Fill procedure for $\theta = \theta_a$ or $\theta = \theta_o$. This can be verified by observing that the coefficients of such functions follow directly from the coefficients of functions for filter probabilities (see Section 2.2.2), and that the Coefficient-Matrix-Fill procedure provides the necessary information for the filter probabilities for all times slices under consideration.

6 Related Work

As mentioned before, in the area of hidden Markov models sensitivity analysis is typically implemented as a perturbation analysis where a small change is applied to one or more parameters and the output of interest re-computed. Since perturbation requires inference for each alteration of parameters, this is an inefficient way of performing a reliable sensitivity analysis. Using a (dynamic) Bayesian network representation of a hidden Markov model in essence allows us to exploit the available Bayesian network algorithms for establishing sensitivity functions; such functions give a complete description of the relation between parameters and output probabilities.

As argued in Section 2.3.2, varying a transition or observation parameter in an HMM corresponds to varying multiple parameters in its Bayesian network representation, one for each time slice under consideration. For Bayesian networks, N-way sensitivity analysis, with parameters from *different* conditional probability distributions, has been studied by only few (see [17] for an overview and comparison of research). For computing the coefficients of N-way sensitivity functions roughly three approaches, or combinations thereof, are known: symbolic propagation, solving systems of linear equations, and propagation of tables with coefficients. Symbolic propagation [23] yields an algebraic expression for a single probability of interest $p(x \mid \boldsymbol{e})$ in terms of all network parameters; by filling in the estimates for the parameters that are not varied, a sensitivity function in the varied parameters results. This method can be used to compute the exponential number of coefficients of the N-way sensitivity function in the dynamic Bayesian network representation of an HMM; it does not allow for directly exploiting the repetitive character of parameters in an HMM. A major disadvantage of symbolic propagation is, however, that it is very time-consuming, since it does not build on standard inference algorithms. Standard inference algorithms can be straightforwardly applied in methods that build upon solving a system of linear equations [14]: if N coefficients are required, then the parameters under consideration are perturbed N times, upon which the output of interest is re-computed; this results in a system of N linear equations. This approach, using inference N times, can be used to directly compute the linear number of coefficients for the sensitivity functions $p(X \mid \boldsymbol{e})$ for a single output variable of interest in an HMM.

Two other algorithms for N-way sensitivity analysis in Bayesian networks consist of tailored versions of the standard junction-tree algorithm [24] for inference. The algorithm by Kjærulff and Van der Gaag [20] requires $\frac{1}{N} \cdot 2^{N-1}$ propagations for an N-way sensitivity analysis, returning the coefficients for the sensitivity functions $p(X \mid \boldsymbol{e})$ for a single output variable of interest. The algorithm can be applied in the context of HMMs, but as with symbolic propagation, it does not allow for exploiting the repetitive character of HMM parameters. As a result, the number of coefficients computed is exponential in N, whereas ultimately only a linear number is required. The approach taken by Coupé et al. [17] resembles our Coefficient-Matrix-Fill procedure in the sense that a table or matrix of coefficients is constructed; their approach extends the junction-tree architecture to propagate vector tables rather than potential functions and defines operations on vectors to this end. Each vector table contains (partially computed) coefficients of the corresponding potential function in terms of the parameters under study. After accumulating the local coefficients, the coefficients of the N-way sensitivity function for a single output probability of interest $p(x \mid \boldsymbol{e})$ are returned.

We conclude that our approach differs from the approaches mentioned above in the sense that our approach

- does not depend on a specific computational architecture;
- does not require a Bayesian network representation of the HMM;
- exploits the fact that we have a polynomial function in a single parameter;
- serves to establish the coefficients for the sensitivity functions for $p(X^t \mid \boldsymbol{e})$ for *all* time slices 1 through t and all hidden states, rather than just for the single output variable X^t or a single output value.

7 Conclusions and Further Research

In this paper we introduced a new and efficient algorithm for computing the coefficients of sensitivity functions in hidden Markov Models, for all three types of model parameter. Earlier work on this topic suggested to use the Bayesian network representation of HMMs and associated algorithms for sensitivity analysis. In this paper we have shown that exploiting the repetitive character of HMMs results in a simple algorithm that computes the coefficients of the sensitivity functions for all hidden states and all time steps. Our procedure basically mimics the Forward-Backward inference algorithm, but computes coefficients rather than probabilities. Various improvements of the Forward-Backward algorithm for HMMs exist that exploit the matrix formulation [13, Section 15.3]; further research is required to investigate if our procedure can be improved in similar or different ways.

In Section 2.2.2 we mentioned the robustness of the most probable explanation (MPE) to parameter variation as another interesting sensitivity question. Iin HMMs the Viterbi algorithm rather than the Forward-Backward algorithm is used to compute MPEs; our guess is therefore that the Coefficient-Matrix-Fill procedure will not be directly suitable for establishing sensitivity functions that describe changes in MPE as a function of changes in HMM parameters. Related work on robustness of MPEs in Bayesian networks [25] could be used as a basis for further research. Another challenge will be to extend current research to sensitivity analysis in which different types of model parameter are varied simultaneously, and to extensions of HMMs.

Finally, future research efforts can be put in the study of general properties of sensitivity functions in HMMs. For example, perturbation bounds have been derived for hidden Markov models, which suggest that HMMs are considerably more sensitive to variations in observation parameters than to variations in transition or initial parameters [5]; it would be interesting to see if similar insights follow from the general form of an HMM sensitivity function.

References

[1] P. Dymarski (Ed.), Hidden Markov Models, Theory and Applications, InTech Open Access Publishers, 2011.

[2] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, PhD Thesis, University of California, Berkeley, 2002.

[3] P. Smyth, D. Heckerman, M.I. Jordan, "Probabilistic independence networks for hidden Markov probability models", Neural Computation 9 (1997) 227–269.

[4] P.-A. Coquelin, R. Deguest, R. Munos, "Sensitivity analysis in HMMs with application to likelihood maximization", in: Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems, 2009.

[5] A.Yu. Mitrophanov, A. Lomsadze, M. Borodovsky, "Sensitivity of hidden Markov models", Journal of Applied Probability 42 (2005) 632 – 642.

[6] Th. Charitos, L.C. van der Gaag, "Sensitivity properties of Markovian models", in: Proceedings of Advances in Intelligent Systems – Theory and Applications Conference (AISTA), Luxembourg, IEEE Computer Society, 2004.

[7] S. Renooij, "Efficient sensitivity analysis in hidden Markov models", in: Proceedings of the Fifth European Workshop on Probabilistic Graphical Models, HIIT Publications 2010-2, Helsinki, 2010, pp. 241 – 248.

[8] F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, second ed., Springer Verlag, 2007.

[9] J.Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[10] G.F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks", Artificial Intelligence 42 (1990) 393 – 405.

[11] L.E. Baum, T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains", The Annals of Mathematical Statistics 37 (1966) 1554 – 1563.

[12] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", in: Proceedings of the IEEE 77, 1989, pp. 257 – 286.

[13] S. Russel, P. Norvig, Artificial Intelligence: A Modern Approach, second ed., Prentice Hall, 2003.

[14] E. Castillo, J.M. Gutiérrez, A.S. Hadi, “Sensitivity analysis in discrete Bayesian networks”, IEEE Transactions on Systems, Man, and Cybernetics 27 (1997) 412 – 423.

[15] H. Chan, A. Darwiche, “When do numbers really matter?”, Journal of Artificial Intelligence Research 17 (2002) 265 – 287.

[16] H. Chan, A. Darwiche, “Sensitivity analysis in Bayesian networks: from single to multiple parameters”, in: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, VA, 2004, pp. 67 – 75.

[17] V.M.H. Coupé, F.V. Jensen, U. Kjærulff, L.C. van der Gaag, “A computational architecture for n-way sensitivity analysis of Bayesian networks”, Technical Report: Department of Computer Science, Aalborg University, 2000.

[18] V.M.H. Coupé, L.C. van der Gaag, “Properties of sensitivity analysis of Bayesian belief networks”, Annals of Mathematics and Artificial Intelligence 36 (2002) 323 – 356.

[19] L.C. van der Gaag, S. Renooij, “Analysing sensitivity data from probabilistic networks”, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, 2001, pp. 530 – 537.

[20] U. Kjærulff, L.C. van der Gaag, “Making sensitivity analysis computationally efficient”, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, 2000, pp. 317 – 325.

[21] K.B. Laskey, “Sensitivity analysis for probability assessments in Bayesian networks”, IEEE Transactions on Systems, Man, and Cybernetics 25 (1995) 901 – 909.

[22] Th. Charitos, Reasoning with Dynamic Networks in Practice, PhD Thesis: Utrecht University, The Netherlands, 2007.

[23] E. Castillo, J.M. Gutiérrez, A.S Hadi, “Parametric structure of probabilities in Bayesian networks”, in: Lectures Notes in Artificial Intelligence 946: Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer-Verlag, New York, 1995, pp. 89 – 98.

[24] S.L. Lauritzen, D.J. Spiegelhalter, “Local computations with probabilities on graphical structures and their application to expert systems”, Journal of the Royal Statistical Society, Series B, 50 (1988) 157 – 224.

[25] H. Chan, A. Darwiche, “On the robustness of most probable explanations”, in: Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, 2006, pp. 63 – 71.