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
silja@cs.uu.nl

Abstract

Sensitivity analysis in a Hidden Markov model (HMM) usually amounts to applying a change to its parameters and re-computing its output of interest. Recently it was shown that, as in Bayesian networks, a simple mathematical function describes the relation between a model parameter and a probability of interest in an HMM. Up till now, however, no special purpose algorithms existed for determining this function. In this paper we present a new and efficient algorithm for doing so, which exploits the recursive properties of an HMM.

1 Introduction

Hidden Markov models (HMMs) are frequently applied statistical models for capturing processes that evolve over time. An HMM can be represented by the simplest type of Dynamic Bayesian network (see for details Smyth, Heckerman & Jordan, 1997; Murphy (2002)), which entails that all sorts of algorithms available for (Dynamic) Bayesian networks can be straightforwardly applied to HMMs.

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. An analysis in which a single parameter is varied, is called a *one-way* sensitivity analysis; in an n*-way* analysis $n > 1$ parameters are varied simultaneously. 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 (see Coupé et al. (2000) for an overview and comparison of existing algorithms). Recently, it was shown that similar functions describe the relation between model parameters and output probabilities in HMMs (Charitos & Van der Gaag, 2004). For computing the constants of these functions, it was suggested to represent the HMM as a Bayesian network, unrolled for a fixed number of time slices, and to use the above mentioned 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 this paper we present a new and efficient algorithm for computing the constants of the sensitivity function in HMMs, which exploits the recursive properties of an HMM. After presenting some preliminaries concerning HMMs and sensitivity functions in Section 2, we review the known recursive expressions for different probabilities of interest in Sections 3 and 4; more specifically, we focus on so-called filter and prediction probabilities in Section 3 and on smoothing in Section 4. In these sections, we subsequently translate the recursive expressions into functions of model parameters and present algorithms for computing the constants of the associated sensitivity functions. We discuss relevant related work in Section 5 and conclude the paper with directions for future research in Section 6.

2 Preliminaries

For each time t, an HMM consists of a single hidden variable whose state can be observed by some test or sensor. The uncertainty in the test or sensor

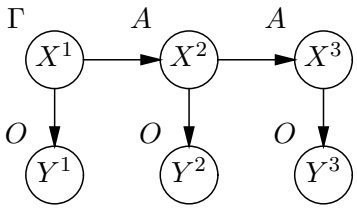

Figure 1: A Bayesian network representation of an HMM unrolled for three time slices.

output is captured by a set of observation probabilities; the transitions among the states in subsequent time steps, or *time slices*, are captured by a set of transition probabilities. In this paper we concentrate on HMMs with discrete observable variables. We further assume that the model is time-invariant, i.e. the probability parameters do not depend on time. More formally, an HMM now is a statistical model $H = (X, Y, A, O, \Gamma)$, where for each time $t \geq 1$:

- X^t is the hidden variable; its states are denoted by x_i^t, $i = 1, \ldots, n$, $n \geq 2$;
- Y^t is the observable variable, with states denoted by y_j^t, $j = 1, \ldots, m$, $m \geq 2$; the notation y_e^t is used to indicate actual evidence;
- A is the transition matrix with entries $a_{i,j} = p(x_j^{t+1} \mid x_i^t)$, $i, j = 1, \ldots, n$;
- O is the observation matrix with entries $o_{i,j} = p(y_j^t \mid x_i^t)$, $i = 1, \ldots, n$, $j = 1, \ldots, m$;
- Γ is the initial vector for X^1 with entries $\gamma_i = p(x_i^1)$, $i = 1, \ldots, n$.

Figure 1 shows a Bayesian network representation of an HMM unrolled for three time slices.

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 $p(X^t \mid y_e^{1:T})$, where $y_e^{1:T}$ is short for the sequence of observations $y_e^1, \ldots, y_e^T$. 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 Russel & Norvig (2003, chapter 15)). This algorithm computes for all hidden states i at time t, the following two probabilities:

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

resulting in

$$p(x_i^t \mid y_e^{1:T}) = \frac{p(x_i^t,\, y_e^{1:T})}{p(y_e^{1:T})} = \frac{F(i,t) \cdot B(i,t)}{\sum_{i=1}^{n} F(i,t) \cdot B(i,t)}$$

Alternatively, the HMM can be represented as a Bayesian network unrolled for $\max\{t, T\}$ time slices, upon which standard Bayesian network inference algorithms can be used.

The outcome $p(x_i^t \mid y_e^{1:T})$ depends on the probability parameters specified for the model. To study the effects of possible inaccuracies in these parameters on the computed output, a *sensitivity analysis* can be done. To this end, we establish the *sensitivity function* $p(x_i^t \mid y_e^{1:T})(\theta)$ that describes our output of interest in terms of parameter θ, where θ can be any model parameter, i.e. an initial probability, an observation probability or a transition probability.[1]

In the context of Bayesian networks, sensitivity analysis has been studied extensively by various researchers (see Van der Gaag, Renooij & Coupé (2007) for an overview and references). 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 (Mitrophanov, Lomsadze & Borodovsky, 2005). The main difference between sensitivity analysis in Bayesian networks and in Hidden Markov models in essence is that a single parameter in an HMM may occur multiple times. 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 (Charitos & Van der Gaag, 2004). The difference with the general n-way function for Bayesian networks is, however, that the n parameters are constrained to

[1] If a parameter $\theta = p(v_j \mid \pi)$ for a variable V is varied, we must ensure that still $\sum_i p(v_i \mid \pi) = 1$. To this end, all probabilities $p(v_i \mid \pi)$, $i \neq j$, are co-varied proportionally: $p(v_i \mid \pi)(\theta) = p(v_i \mid \pi) \cdot \frac{1-\theta}{1-p(v_j \mid \pi)}$. For binary-valued V this simplifies to $p(v_i \mid \pi)(\theta) = 1 - \theta$.

all be equal, which reduces the number of required constants. We now summarise the known results for sensitivity functions in HMMs (Charitos & Van der Gaag, 2004; Charitos, 2007). For the probability of evidence as a function of a model parameter θ, we have the following polynomial function:

$$p(y_e^{1:T})(\theta) = d_N^T \cdot \theta^N + \ldots + d_1^T \cdot \theta + d_0^T$$

where $N = T$ if $\theta = o_{r,s}$, $N = T-1$ if $\theta = a_{r,s}$, $N = 1$ for $\theta = \gamma_r$, and coefficients $d_N^T, \ldots, d_0^T$ are constants with respect to the various parameters. For the joint probability of a hidden state and evidence, as a function of a model parameter θ, we have the following polynomial function:

$$p(x_v^t, y_e^{1:T})(\theta) = c_{v,N}^t \cdot \theta^N + \ldots + c_{v,1}^t \cdot \theta + c_{v,0}^t$$

where

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

and coefficients $c_{v,N}^t, \ldots, c_{v,0}^t$ are constants with respect to the various parameters. The same general forms apply to prior marginals over X, by taking $T = 0$. Note that prior probabilities are not affected by variation in observation parameters $o_{r,s}$.

Up till now, no special purpose algorithms for establishing the coefficients of the sensitivity functions in an HMM were available, which means that Bayesian network algorithms need to be used to this end. In the next sections, we present a new and efficient algorithm for computing the coefficients for sensitivity functions in HMMs. To this end, we exploit the repetitive character of the model parameters and knowledge about the polynomial form of the sensitivity functions presented above. We will discuss the inference tasks filtering/prediction and smoothing separately.

3 Filter and Prediction Coefficients

In this section we establish the coefficients of the sensitivity function $p(x_v^t \mid y_e^{1:T})(\theta)$, $t \geq T$, for various model parameters θ. Note that the sensitivity function for a probability $p(x_v^t \mid y_e^{1:T})$ is a quotient of the sensitivity functions for $p(x_v^t, y_e^{1:T})$ and for $p(y_e^{1:T})$, and that $p(y_e^{1:T}) = \sum_{z=1}^n p(x_z^t, y_e^{1:T})$. Therefore, given the polynomial form of the functions, the coefficients for the sensitivity functions for $p(x_v^t, y_e^{1:T})$ provide enough information to establish all required coefficients. The remainder of this section is therefore devoted to obtaining the coefficients for $p(x_v^t, y_e^{1:T})$ as a function of θ.

3.1 Filter and Prediction Recursions

We will now review the recursive expression for filter probabilities (see for details Russel & Norvig (2003, chapter 15)) and make explicit the relation between filter and prediction probabilities. Starting with the latter, we find by conditioning on X^T and exploiting the independence $X^t \perp Y^{1:T} \mid X^T$ for $T < t$, that

$$p(x_v^t, y_e^{1:T}) = \sum_{z=1}^n p(x_v^t \mid x_z^T) \cdot p(x_z^T, y_e^{1:T}) \quad (1)$$

The second factor in this summation is a filter probability; the first can be computed from similar probabilities in time slice $t-1$ for $t > T+1$:

$$p(x_v^t \mid x_z^T) \;=\; \sum_{w=1}^n a_{w,v} \cdot p(x_w^{t-1} \mid x_z^T) \quad (2)$$

and equals for $t = T+1$:

$$p(x_v^t \mid x_z^T) = a_{z,v} \quad (3)$$

Now consider the filter probability, i.e. the case where $T = t$. Exploiting the conditional independences $Y^t \perp Y^{1:t-1} \mid X^t$ and $X^t \perp Y^{1:t-1} \mid X^{t-1}$, and conditioning on X^{t-1}, we have the following relation between probabilities $p(x_v^t, y_e^{1:t})$ and $p(x_v^{t-1}, y_e^{1:t-1})$ for two subsequent time slices $t-1$ and t, $t > 1$:

$$p(x_v^t, y_e^{1:t}) \;=\; o_{v,e_t} \cdot \sum_{z=1}^n a_{z,v} \cdot p(x_z^{t-1}, y_e^{1:t-1}) \quad (4)$$

where e_t corresponds to the value of Y observed at time t. For time slice $t = 1$, we have that

$$p(x_v^1, y_e^1) = p(y_e^1 \mid x_v^1) \cdot p(x_v^1) = o_{v,e_1} \cdot \gamma_v \quad (5)$$

Note that for a prior marginal $p(x_i^t)$ we find the same expressions with o_{v,e_t} omitted.

Finally, consider the case where $T < t$. Equations 1 and 2 show that we now basically need to prolong the recursion in Equation 4 from time T to time t, except that for the time slices $T+1$ up to and including t no evidence is available. The absence of evidence can be implemented by multiplying with 1 rather than o_{v,e_t}. In the remainder of this section, we will therefore assume, without lack of generality, that $T = t$.

We will now translate the above relations into functions of the three types of model parameter. We already know that those functions are polynomial in the parameter under consideration, and we know the degree of the functions. However, we have yet to establish what the coefficients are and how to compute them. For ease of exposition concerning the covariation of parameters, we assume in the remainder of this section that all variables are binary-valued, i.e. $n = m = 2$.

3.2 Initial Parameters

We consider the sensitivity function $p(x_v^t, y_e^{1:t})(\theta_\gamma)$ for model parameter $\theta_\gamma = \gamma_r$. Note that γ_v, the parameter associated with the state of interest for X^t, corresponds either to θ_γ ($v = r$) or its complement $1-\theta_\gamma$ ($v \neq r$). From Equation 4 it now follows that for $t = 1$:

$$p(x_v^1, y_e^1)(\theta_\gamma) = \begin{cases} o_{v,e_1} \cdot \theta_\gamma + 0 & \text{if} \quad v = r \\ -o_{v,e_1} \cdot \theta_\gamma + o_{v,e_1} & \text{if} \quad v \neq r \end{cases}$$

and from Equation 5 we have for $t > 1$:

$$p(x_v^t, y_e^{1:t})(\theta_\gamma) = \\ = \sum_{z=1}^{2} o_{v,e_t} \cdot a_{z,v} \cdot p(x_z^{t-1}, y^{1:t-1})(\theta_\gamma)$$

The polynomial $p(x_v^t, y_e^{1:t})(\theta_\gamma)$ requires two coefficients: $c_{v,1}^t$ and $c_{v,0}^t$. Since each initial parameter is used only in time step 1, as the above expressions demonstrate, the coefficients for $t > 1$ can be established through a simple recursion for each $N = 0, 1$:

$$c_{v,N}^t = \sum_{z=1}^{2} o_{v,e_t} \cdot a_{z,v} \cdot c_{z,N}^{t-1}$$

with $c_{v,0}^1 = 0$ if $v = r$, and o_{v,e_1} otherwise; in addition $c_{v,1}^1 = o_{v,e_1}$ if $v = r$, and $-o_{v,e_1}$ otherwise.

3.3 Transition Parameters

We consider the sensitivity function $p(x_v^t, y_e^{1:t})(\theta_a)$ for model parameter $\theta_a = a_{r,s}$. From Equations 4 and 5 it follows that for $t = 1$ we find a constant, $p(x_v^1, y_e^1)(\theta_a) = o_{v,e_1} \cdot \gamma_v$, and for $t > 1$,

$$p(x_v^t, y_e^{1:t})(\theta_a) = \qquad (6)$$

$$= o_{v,e_t} \cdot \sum_{z=1}^{n} a_{z,v}(\theta_a) \cdot p(x_z^{t-1}, y_e^{1:t-1})(\theta_a)$$

$$= o_{v,e_t} \cdot a_{r,v}(\theta_a) \cdot p(x_r^{t-1}, y_e^{1:t-1})(\theta_a) + \\ + o_{v,e_t} \cdot a_{\overline{r},v}(\theta_a) \cdot p(x_{\overline{r}}^{t-1}, y_e^{1:t-1})(\theta_a)$$

where $\overline{r}$ denotes the state of X other than r. In the above formula, $a_{r,v}(\theta_a)$ equals θ_a for $v = s$ and $1 - \theta_a$ for $v \neq s$; $a_{\overline{r},v}$ is independent of θ_a.

The polynomial $p(x_v^t, y_e^{1:t})(\theta_a)$ requires t coefficients: $c_{v,N}^t$, $N = 0, \ldots, t-1$. To compute these coefficients, building upon Equation 6 above, we designed a procedure which constructs a set of matrices containing the coefficients of the polynomial sensitivity functions for each hidden state and each time slice. We call this procedure the *Coefficient-Matrix-Fill* procedure.

3.3.1 The Coefficient-Matrix-Fill Procedure

The Coefficient-Matrix-Fill procedure constructs a matrix for each time slice under consideration, and fills this matrix with the coefficients of the polynomial functions relevant for that time slice. In this section, we will detail the procedure for computing the coefficients of $p(x_v^t, y_e^{1:t})(\theta_a)$. In the sections following, we demonstrate how a similar procedure can be used to compute the coefficients given an observation parameter, and in case $T \neq t$.

The basic idea For each time slice $k = 1, \ldots, t$ we construct an $n \times k$ matrix F^k. A row i in F^k contains exactly the coefficients for the function $p(x_i^k, y_e^{1:k})(\theta_a)$, so the procedure in fact computes the coefficients for the sensitivity functions for *all* n hidden states and *all* time slices up to and including t. 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 $p(x_i^k, y_e^{1:k})(\theta_a)$. The entries of matrix F^1 are set to their correct values in

the initialisation phase of the procedure. 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.

Fill operations The recursive steps in the various formulas are implemented by transitioning from matrix F^k to F^{k+1} for $k \geq 1$. 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 this polynomial be represented in row i of matrix F^k, i.e. $f^k_{i,\cdot} = (c_0, \ldots, c_{k-1})$. In transitioning from matrix F^k to F^{k+1}, three types of operation (or combinations thereof) can be applied to $p(\theta)$:

- summation with another polynomial $(-)p^*(\theta)$ of the same degree: this just requires summing the coefficients of the same order, i.e. summing entries with the same column number;
- multiplication with a constant d: the resulting polynomial is represented in row i of matrix F^{k+1} by $f^{k+1}_{i,\cdot} = (d \cdot c_0, \ldots, d \cdot c_{k-1}, 0)$. Note that F^{k+1} has an additional column $k+1$, which is unaffected by this operation.
- multiplication with θ: the resulting kth-degree polynomial is represented in row i of matrix F^{k+1} by $f^{k+1}_{i,\cdot} = (0, c_0, \ldots, c_{k-1})$. This operation basically amounts to shifting entries from F^k one column to the right.

Fill contents: initialisation Matrix F^1 is initialised by setting $f^1_{i,1} = o_{i,e_1} \cdot \gamma_i$ for $i = 1, 2$. Matrices $F^2, \ldots, F^t$ are initialised by filling them with zeroes.

Fill contents: $k = 2, \ldots, t$ We will now provide the details for filling matrix F^k, $k > 1$. Following Equation 6, position j in row i of matrix F^k, $f^k_{i,j}$, $k > 1$, is filled with:

if $i = s$ then for $1 < j < k$:

$$o_{i,e_k} \cdot (f^{k-1}_{r,j-1} + a_{\overline{r},i} \cdot f^{k-1}_{\overline{r},j})$$

if $i \neq s$ then for $1 < j < k$:

$$o_{i,e_k} \cdot (-f^{k-1}_{r,j-1} + f^{k-1}_{r,j} + a_{\overline{r},i} \cdot f^{k-1}_{\overline{r},j})$$

For $j = 1$, the general cases above are simplified by setting $f^{k-1}_{r,j-1} = 0$. This boundary condition captures 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 $f^{k-1}_{\cdot,j} = 0$ in that case.

Complexity In each of the k steps, an $n \times k$ matrix is filled. This matrix contains the coefficients for the functions $p(x^k_i, y^{1:k}_e)(\theta_a)$ 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 save space by storing only two matrices at all times. The runtime complexity for a straightforward implementation of the algorithm is $O(n^2 \cdot t^2)$, which is 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 one.

Example 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:

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

Suppose we are interested in the sensitivity functions for the two states of X^3 as a function of parameter $\theta_a = a_{2,1} = p(x^t_1 \mid x^{t-1}_2) = 0.15$, for all $t > 1$. Suppose the following sequence of observations is obtained: y^1_2, y^2_1 and y^3_1. To compute the coefficients for the sensitivity functions, the following matrices are constructed by the Coefficient-Matrix-Fill procedure:

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

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

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

$$= \frac{\begin{bmatrix} 0.03563 & 0.06 \\ 0.07425 & -0.072 \end{bmatrix}}{0.10988 \quad -0.012} +$$

and finally, $F^3 =$

$$= \left[\begin{array}{c} o_{1,1} \cdot a_{1,1} \cdot f^2_{1,1} \\ o_{2,1} \cdot (f^2_{2,1} + a_{1,2} \cdot f^2_{1,1}) \end{array} \right.$$

$$\begin{array}{r} o_{1,1} \cdot (f^2_{2,1} + a_{1,1} \cdot f^2_{1,2}) \\ o_{2,1} \cdot (-f^2_{2,1} + f^2_{2,2} + a_{1,2} \cdot f^2_{1,2}) \end{array}$$

$$\left. \begin{array}{r} o_{1,1} \cdot f^2_{2,2} \\ -o_{2,1} \cdot f^2_{2,2} \end{array} \right]$$

$$= \frac{\begin{bmatrix} 0.02538 & 0.09844 & -0.054 \\ 0.06843 & -0.12893 & 0.0648 \end{bmatrix}}{0.09381 \quad -0.03049 \quad 0.0108} +$$

From which we can conclude, for example:

$$p(x^3_1 \mid y^{1:3}_e)(\theta_a) = \frac{-0.054 \cdot \theta_a^2 + 0.098 \cdot \theta_a + 0.025}{0.011 \cdot \theta_a^2 - 0.030 \cdot \theta_a + 0.094}$$

and also,

$$p(x^2_2 \mid y^{1:2}_e)(\theta_a) = \frac{-0.072 \cdot \theta_a + 0.074}{-0.012 \cdot \theta_a + 0.110}$$

Note that the coefficients for the probability of evidence function follow from summing the entries in each column of F^t (see e.g. F^2 and F^3 above). □

3.4 Observation Parameters

We consider the sensitivity function $p(x^t_v,\, y^{1:t}_e)(\theta_a)$ for model parameter $\theta_o = o_{r,s}$. From Equation 4 it follows that for $t = 1$:

$$p(x^1_v,\, y^1_e)(\theta_o) = \tag{7}$$

$$= \begin{cases} o_{v,e_1} \cdot \gamma_v & \text{if} \quad v \neq r; \\ \theta_o \cdot \gamma_r & \text{if} \quad v = r \text{ and } e_1 = s; \\ (1 - \theta_o) \cdot \gamma_r & \text{if} \quad v = r \text{ and } e_1 \neq s; \end{cases}$$

and from Equation 5 we have for $t > 1$:

$$p(x^t_v,\, y^{1:t}_e)(\theta_o) =$$

$$= o_{v,e_t}(\theta_o) \cdot \sum_{z=1}^{2} a_{z,v} \cdot p(x^{t-1}_z,\, y^{1:t-1})(\theta_o) \tag{8}$$

where $o_{v,e_t}(\theta_o)$ equals o_{v,e_t} for $v \neq r$, θ_o for $v = r$ and $e_t = s$, and $1 - \theta_o$ for $v = r$ and $e_t \neq s$.

The polynomial function $p(x^t_v,\, y^{1:t}_e)(\theta_o)$ requires $t + 1$ coefficients: $c^t_{v,N}$, $N = 0, \ldots, t$. We compute these coefficients, building upon the Equations 7 and 8, again using our Coefficient-Matrix-Fill procedure. The contents and size of the matrices differ from the case with transition parameters and are specified below.

Fill contents: initialisation F^1 is an $n \times 2$ matrix, initialised in accordance with Equation 7. All F^k, $k = 2, \ldots, t$, are $n \times (k + 1)$ matrices and are initialised by filling them with zeroes.

Fill contents: $k = 2, \ldots, t$ Following Equation 8, position j in row i of matrix F^k, $f^k_{i,j}$, $k > 1$, is filled with the following for $j = 2, \ldots, k$:

$$\begin{cases} \sum_{z=1}^{2} a_{z,i} \cdot f^{k-1}_{z,j-1} & \text{if} \quad i = r,\ e_t = s; \\ \sum_{z=1}^{2} a_{z,i} \cdot (f^{k-1}_{z,j} - f^{k-1}_{z,j-1}) & \text{if} \quad i = r,\ e_t \neq s; \\ o_{\overline{r},e_k} \cdot \sum_{z=1}^{2} a_{z,\overline{r}} \cdot f^{k-1}_{z,j} & \text{if} \quad i \neq r; \end{cases}$$

For $j = 1$ and $j = k + 1$ we again simplify the above formulas where necessary, to take into account boundary conditions.

4 Smoothing Coefficients

In this section we consider establishing the coefficients of the sensitivity function $p(x^t_v \mid y^{1:T}_e)(\theta)$ for various model parameters θ, in the situation where $t < T$. We again focus only on $p(x^t_v,\, y^{1:T}_e)(\theta)$.

4.1 Recursion for Smoothing

Recall from Section 2 that $p(x^t_v,\, y^{1:T}_e) = B(i,t) \cdot F(i,t)$, or

$$p(x^t_v,\, y^{1:T}_e) = p(y^{t+1:T}_e \mid x^t_v) \cdot p(x^t_v,\, y^{1:t}_e) \tag{9}$$

The second term in this product is again a filter probability, so we now further focus on the first term. By conditioning on X^{t+1} and exploiting independences (see for details (Russel & Norvig, 2003, chapter 15)), we have the following relation between probabilities $p(y^{t+1:T}_e \mid x^t_v)$ and $p(y^{t+2:T}_e \mid x^{t+1}_v)$ for $t + 1 < T$:

$$p(y^{t+1:T}_e \mid x^t_v) =$$

$$= \sum_{z=1}^{n} o_{z,e_{t+1}} \cdot a_{v,z} \cdot p(y^{t+2:T}_e \mid x^{t+1}_z) \tag{10}$$

For $t+1=T$, this reduces to

$$p(y_e^{T:T} \mid x_v^{T-1}) \quad = \quad \sum_{z=1}^{n} o_{z,e_T} \cdot a_{v,z} \cdot 1 \qquad (11)$$

Again we translate the above relations into functions of the various model parameters. From Equations 10 and 11 it follows that the function $p(y_e^{t+1:T} \mid x_v^t)(\theta)$ is polynomial in each model parameter.[2] Moreover, from Equation 9 we have that the degree of $p(x_v^t, y_e^{1:T})(\theta)$ equals the sum of the degrees of $p(y_e^{t+1:T} \mid x_v^t)(\theta)$ and $p(x_v^t, y_e^{1:t})(\theta)$. Since the degrees of both $p(x_v^t, y_e^{1:T})(\theta)$ and $p(x_v^t, y_e^{1:t})(\theta)$ are known (see Section 2), the degree of $p(y_e^{t+1:T} \mid x_v^t)(\theta)$ can be established as their difference. We thus have that

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

where

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

and coefficients $d_{v,N}^t, \ldots, d_{v,0}^t$ are constants with respect to the various parameters. The coefficients of the polynomial function $p(x_v^t, y_e^{1:T})(\theta)$, $T > t$, can thus be established by standard polynomial multiplication of $p(x_v^t, y_e^{1:t})(\theta)$ and $p(y_e^{t+1:T} \mid x_v^t)(\theta)$.

In the following we will establish exactly what the coefficients of $p(y_e^{t+1:T} \mid x_v^t)(\theta)$ are and how to compute them. For ease of exposition, we again take $n = m = 2$.

4.2 Initial Parameters

We consider the function $p(y_e^{t+1:T} \mid x_v^t)(\theta_\gamma)$, $t < T$, for model parameter $\theta_\gamma = \gamma_r$. The degree of this polynomial is 0. Indeed, from Equations 10 and 11 it follows that this function is constant with respect to an initial parameter. This constant is simply a probability which can be computed using standard inference.

4.3 Transition Parameters

The function $p(y_e^{t+1:T} \mid x_v^t)(\theta_a)$, $t < T$, with parameter $\theta_a = a_{r,s}$ requires $T-t+1$ coefficients. We again compute these coefficients using our Coefficient-Matrix-Fill procedure, where contents is now determined by Equations 10 and 11, and depends on the relation between $a_{v,z}$ and θ_a: if $v = r$ then $a_{v,z}$ equals θ_a for $z = s$, and $1 - \theta_a$ for $z = \bar{s}$; otherwise $a_{v,z}$ is constant.

To distinguish between computations that move forward in time, and the current ones which move backward in time, we will use matrices B^k, $t \leq k \leq T$, where $k = T$ is used purely as initialisation.

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

Fill contents: $k = T-1$ down to t Following Equations 10 and 11, position j in row i of matrix B^k, $b_{i,j}^k$, $k < T$, is filled with the following for $j = 2, \ldots, T-k$:

if $i = r$**:** $\left(\sum_{z=1}^{2} o_{z,e_{k+1}} \cdot b_{z,j-1}^{k+1}\right) + o_{\bar{s},e_{k+1}} \cdot b_{\bar{s},j}^{k+1}$

if $i \neq r$**:** $\sum_{z=1}^{2} o_{z,e_{k+1}} \cdot a_{i,z} \cdot b_{z,j}^{k+1}$

For $j = 1$ and $j = T-k+1$ we again have to take into account boundary conditions.

4.4 Observation Parameters

The function $p(y_e^{t+1:T} \mid x_v^t)(\theta_o)$, $t < T$, with parameter $\theta_o = o_{r,s}$ requires $T-t+1$ coefficients. We again compute these coefficients using our Coefficient-Matrix-Fill procedure in a similar way as for the transition parameters above. The only difference is in the fill contents determined by Equations 10 and 11. This now depends on the relation between $o_{z,e_{t+1}}$ and θ_o: for $z = r$, $o_{z,e_{t+1}}$ equals θ_o if $e_{t+1} = s$, and $1 - \theta_a$ if $e_{t+1} \neq s$; for $z = \bar{r}$, $o_{z,e_{t+1}}$ is constant.

Fill contents: $k = T-1$ down to t Following Equations 10 and 11, position j in row i of matrix B^k, $b_{i,j}^k$, $k < T$, is filled with the following for $j = 2, \ldots, T-k$:

if $e_{k+1} = s$**:** $a_{i,r} \cdot b_{r,j-1}^{k+1} + o_{\bar{r},e_{k+1}} \cdot a_{i,\bar{r}} \cdot b_{\bar{r},j}^{k+1}$

if $e_{k+1} \neq s$**:** $-a_{i,r} \cdot b_{r,j-1}^{k+1} + a_{i,r} \cdot b_{r,j}^{k+1} + o_{\bar{r},e_{k+1}} \cdot a_{i,\bar{r}} \cdot b_{\bar{r},j}^{k+1}$

For $j = 1$ and $j = T-k+1$ we again take into account the boundary conditions.

[2] Note that this may seem counter-intuitive as it concerns the function for a *conditional* probability; since X^t is an ancestor of $Y^{t+1} \ldots Y^T$, however, the factorisation of $p(y_e^{t+1:T}, x_v^t)$ includes $p(x_v^t)$.

5 Related Work

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. Sensitivity analysis in HMMs is therefore a constrained form of n-way analysis in Bayesian networks, with all varied parameters having the same value at all times. As a result, a sensitivity function in an HMM requires a number of coefficients linear in the number of parameters varied, whereas in Bayesian networks in general an n-way sensitivity function requires an exponential number of coefficients, one for each possible subset of the n varied parameters. For Bayesian networks, n-way sensitivity analysis, with parameters from *different* CPTs, has been studied by only few (see Coupé et al. (2000) 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. The approach taken by Coupé et al. (2000) 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 the coefficients of the corresponding potential function is terms of the parameters under study. Our approach, on the contrary, does not depend on a specific computational architecture nor does it necessarily require a Bayesian network representation of the HMM. In addition, the operations we use are quite different, since we can exploit the fact that we have a polynomial function in a single parameter.

6 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 (Russel & Norvig, 2003, Section 15.3); further research is required to investigate if our procedure can be improved in similar or different ways.

The presented work can be extended quite straightforwardly to sensitivity functions which concern the prediction of future observations, i.e $p(y_e^t \mid y_e^{1:T})(\theta)$, $T < t$. More challenging will be to extend current research to sensitivity analysis in which different types of model parameter are varied simultaneously, and to extensions of HMMs.

References

Th. Charitos, L.C. van der Gaag (2004). Sensitivity properties of Markovian models. *Proceedings of Advances in Intelligent Systems - Theory and Applications Conference (AISTA)*. IEEE Computer Society.

Th. Charitos (2007). *Reasoning with Dynamic Networks in Practice*. PhD Thesis, Utrecht University, The Netherlands.

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

L.C. van der Gaag, S. Renooij, V.M.H. Coupé (2007). Sensitivity analysis of probabilistic networks. In: *Advances in Probabilistic Graphical Models*, Springer Series: Studies in Fuzziness and Soft Computing, Vol. 213, pp. 103-124.

A.Yu. Mitrophanov, A. Lomsadze, M. Borodovsky (2005). Sensitivity of hidden Markov models. *Journal of Applied Probability*, 42, pp. 632–642.

K.P. Murphy (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD Thesis, University of California, Berkeley.

S. Russel, P. Norvig (2003) *Artificial Intelligence: A Modern Approach*, Prentice Hall, Second Edition.

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