Efficient Sensitivity Analysis in HMMs

Silja Renooij

Department of Information and Computing Sciences, Utrecht University
The Netherlands S.Renooij@uu.nl

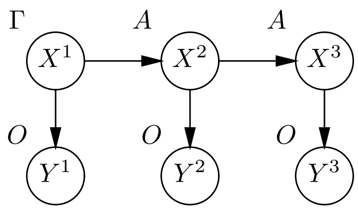
A full version of the paper was published in the Proceedings of the Fifth Workshop on Probabilistic Graphical Models (PGM-2010).

1 Introduction

A Hidden Markov model (HMM) is a frequently applied statistical model for capturing processes that evolve over time. The parameters specified in a HMM are often inaccurate, and sensitivity analyses can be employed to study the effects of these 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 [2]. Recently, however, it was shown that a simple mathematical function describes the relation between HMM parameters and an output probability of interest [1]. This result was established by representing the HMM as a (Dynamic) Bayesian network; for determining the so-called sensitivity function, it was suggested to use existing algorithms for sensitivity analysis in Bayesian networks. 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 computations. 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.

2 Hidden Markov Models

For each time t, an HMM consists of a hidden variable X^t, which can be indirectly observed through some test or sensor Y^t. The noise in the observation is captured in a matrix O containing observation probabilities. The transitions between the states of X in subsequent time steps are described by a matrix A with transition probabilities. Both O and A are time-invariant, and we assume X and Y to be discrete. The prior for the initial state of the system is given by vector Γ. An HMM thus has three types of probability parameters: initial (θ_γ), observation (θ_o), and transition (θ_a). The following shows a Bayesian network representation of an HMM unrolled for three time steps:



Inference in temporal models typically amounts to computing the distribution $p(X^t \mid y_e^{1:T})$, where $y_e^{1:T}$ is short for the sequence of observations $y_e^1, \ldots, y_e^T$ for $Y^1, \ldots, Y^T$. 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, which builds on the following recursive properties (see for details [3, Chapter 15]):

$$\text{for } t = 1: \quad p(x_v^1, y_e^1) = \quad p(y_e^1 \mid x_v^1) \cdot p(x_v^1) \quad = o_{v,e_1} \cdot \gamma_v \tag{1}$$

$$\text{for } t > 1: \quad p(x_v^t, y_e^{1:t}) = \quad = o_{v,e_t} \cdot \sum_{z=1}^{n} a_{z,v} \cdot p(x_z^{t-1}, y_e^{1:t-1}) \tag{2}$$

where v equals one of the n states of X, e_t denotes the state of Y observed at time t, $o_{v,e_t} = p(y_e^t \mid x_v^t)$ is an observation probability, and $a_{z,v} = p(x_v^t \mid x_z^{t-1})$ is a transition probability.

3 An Algorithm for Establishing Sensitivity Functions

Our algorithm builds upon the above mentioned recursive properties and the polynomial form of the sensitivity function $p(x_v^t,\, y_e^{1:T})(\theta) = c_{v,N}^t \cdot \theta^N + \ldots + c_{v,1}^t \cdot \theta + c_{v,0}^t$, to establish its coefficients. Here N depends on the type of θ (θ_γ, θ_o or θ_a), and $c_{v,N}^t, \ldots, c_{v,0}^t$ are constants with respect to the θ under consideration.

The basic idea We sketch our *Coefficient-Matrix-Fill procedure* for establishing the sensitivity function $p(x_v^t,\, y_e^{1:t})(\theta_a)$ for transition probability θ_a. For each time step $k = 1, \ldots, t$ we construct an $n \times k$ matrix F^k and fill this matrix with the coefficients of all polynomial functions relevant for that time step. A row i in F^k then contains exactly the coefficients for the function $p(x_i^k,\, y_e^{1:k})(\theta_a)$; 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 $p(x_i^k,\, y_e^{1:k})(\theta_a)$.

Example Consider an HMM with binary X and binary 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:

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

We are interested in the functions $p(X^3,\, y_e^{1:3})(\theta_a)$ for the two states of X^3 and parameter $\theta_a = a_{2,1} = p(x_1^t \mid x_2^{t-1}) = 0.15$. The following observations are obtained: y_2^1, y_1^2 and y_1^3. To compute the coefficients of the functions, three matrices are constructed by the Coefficient-Matrix-Fill procedure. First, the entries of matrix F^1 are set to their correct values in the initialisation phase of the procedure (see (1)):

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

The remaining 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; their fill contents is based on (2):

$$F^2 = \begin{bmatrix} o_{1,1} \cdot a_{1,1} \cdot f_{1,1}^1 & o_{1,1} \cdot f_{2,1}^1 \\ o_{2,1} \cdot (f_{2,1}^1 + a_{1,2} \cdot f_{1,1}^1) & -o_{2,1} \cdot f_{2,1}^1 \end{bmatrix} = \begin{bmatrix} 0.03563 & 0.06 \\ 0.07425 & -0.072 \end{bmatrix}$$

$$F^3 = \begin{bmatrix} o_{1,1} \cdot a_{1,1} \cdot f_{1,1}^2 & o_{1,1} \cdot (f_{2,1}^2 + a_{1,1} \cdot f_{1,2}^2) & o_{1,1} \cdot f_{2,2}^2 \\ o_{2,1} \cdot (f_{2,1}^2 + a_{1,2} \cdot f_{1,1}^2) & o_{2,1} \cdot (-f_{2,1}^2 + f_{2,2}^2 + a_{1,2} \cdot f_{1,2}^2) & -o_{2,1} \cdot f_{2,2}^2 \end{bmatrix} = \begin{bmatrix} 0.02538 & 0.09844 & -0.054 \\ 0.06843 & -0.12893 & 0.0648 \end{bmatrix}$$

From the above we can conclude

$$p(x_1^3,\, y_e^{1:3})(\theta_a) = -0.054 \cdot \theta_a^2 + 0.098 \cdot \theta_a + 0.025, \quad \text{but also} \quad p(x_2^2,\, y_e^{1:2})(\theta_a) = -0.072 \cdot \theta_a + 0.074$$

In addition, for each column j of F^k, the sum $\sum_{i=1}^n f_{i,j}^k$ of entries equals a coefficient of $p(y_e^{1:k})(\theta)$.

Complexity For each of the k time steps, an $n \times k$ matrix is filled with the coefficients for the functions $p(x_i^k,\, y_e^{1:k})(\theta_a)$ for all i. The procedure thus computes the coefficients for the sensitivity functions for *all hidden states* and *all time steps* up to and including t. The run-time 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.

References

[1] 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.

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

[3] S. Russel, P. Norvig (2003) *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2nd Edition.