Making Gestural Input from Arm-Worn Inertial Sensors More Practical

Louis Kratz[1,2], T. Scott Saponas[1], Dan Morris[1]
lak24@drexel.edu ssaponas@microsoft.com dan@microsoft.com

[1]Microsoft Research, Redmond, WA
[2]Department of Computer Science, Drexel University, Philadelphia, PA

CONSTANT TIME COMPLEXITY

Eq. 10 in our main submission has complexity of $O(N^2 D)$ due to the Forwards-Backwards procedure used to compute $\mathrm{p}(O_{t_r+1:t+1})$ and $\mathrm{p}(O_{t_m:t+1})$ at each time instance. We may efficiently compute $\mathrm{p}(O_{t_r+1:t+1})$ and $\mathrm{p}(O_{t_m:t+1})$ using values at the previous time instance. Specifically, we use the information at time t to quickly compute the values at time $t+1$. Both values in the update step are formulated as the same problem: given a sequence $O_{t_w:t}$ of a fixed duration $(t - t_w)$ and the sequence's likelihood $\mathrm{p}(O_{t_w:t}|\lambda)$, compute the likelihood of the next subsequence $O_{t_w+1:t+1}$ without re-evaluating the entire sequence.

Review of the Forwards-Backwards Algorithm

The forwards backwards algorithm computes the likelihood of being in state j and observing the sequence $O_{t_w:t}$ by

$$\alpha_t(j) = \mathrm{p}(z_t = j, O_{t_w:t}|\lambda) \ , \tag{0.1}$$

where z_t is the value of the hidden state at time t, λ is the HMM, and α is the forwards message from the forwards backwards algorithm.

The likelihood of the observation sequence given the model is computed by marginalization:

$$\mathrm{p}(O_{t_w:t}|\lambda) = \sum_{j=1}^{N} \mathrm{p}(z_t = j, O_{t_w:t}|\lambda) = \sum_{j=1}^{N} \alpha_t(j) \ , \tag{0.2}$$

where N is the number of states in the HMM.

The value of $\alpha_t(j)$ is computed sequentially by

$$\alpha_t(j) = \left[\sum_{i=1}^{N} a_{ij} \alpha_{t-1}(i) \right] b_j(o_t) \ , \tag{0.3}$$

where a_{ij} is the HMM's transition probability, and $b_j(o_t)$ its emission density. The initialization is

$$\alpha_{t_w}(j) = \boldsymbol{\pi}_j b_j(o_{t_w}) \ , \tag{0.4}$$

where $\boldsymbol{\pi}_j$ is the initial state likelihood.

Matrix Representation

First, we formulate the forwards step as a matrix multiplication. Let $\boldsymbol{\alpha}_t$ be an $N \times 1$ vector. Also, let

$$\mathbf{B}_t = \begin{bmatrix} b_1(o_t) & 0 & \dots & 0 \\ 0 & b_2(o_t) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_N(o_t) \end{bmatrix} \tag{0.5}$$

We can re-write Eq. 0.3 as

$$\boldsymbol{\alpha}_t = \mathbf{B}_t \mathbf{A}^T \boldsymbol{\alpha}_{t-1} \tag{0.6}$$

and the initialization in Eq. 0.4 as

$$\boldsymbol{\alpha}_1 = \mathbf{B}_1 \boldsymbol{\pi} \ . \tag{0.7}$$

Exploiting Redundant Computations

To store redundant computations in the forwards procedure, we define a matrix

$$\mathbf{G}_{t_w:t} = \mathbf{A}^T \mathbf{B}_{t-1} \dots \mathbf{A}^T \mathbf{B}_{t_w} \ . \tag{0.8}$$

Note that $\boldsymbol{\alpha}_t$ (and subsequently $\mathrm{p}(O_{t_w:t}|\lambda)$) may be computed from $\mathbf{G}_{t_w:t}$ by

$$\boldsymbol{\alpha}_t = \mathbf{B}_t \mathbf{G}_{t_w:t} \boldsymbol{\pi} \ . \tag{0.9}$$

At time $t+1$, we compute the next matrix $\mathbf{G}_{t_w+1:t+1}$ from $\mathbf{G}_{t_w:t}$ by

$$\mathbf{G}_{t_w+1:t+1} = \mathbf{A}^T \mathbf{B}_t \mathbf{G}_{t_w:t} \mathbf{B}_{t_w}^{-1} \left[\mathbf{A}^T \right]^{-1} . \tag{0.10}$$

Note that $\left[\mathbf{A}^T \right]^{-1}$ is fixed and may be pre-computed, and computing $\mathbf{B}_{t_w}^{-1}$ is trivial since $\mathbf{B}_{t_w}$ is diagonal.

To summarize, we use Eq. 0.10 to update a matrix that we use to compute the likelihood via Eq. 0.9. The update step in Eq. 0.10 is only as computationally complex as matrix multiplication (naive algorithm $O(N^3)$). We use this algorithm to compute $\mathrm{p}(O_{t_r+1:t+1})$ and $\mathrm{p}(O_{t_m:t+1})$ in Eq. 10 of our main submission. As a result, at each time instance t the complexity of the update step in Eq. 10 of our main submission is $O(N^3)$, constant with respect to the maximum duration D of the gesture.

Numeric Stability

In practice, Eq. 0.10 may be numerically unstable if $\mathbf{B}_{t_w}$ or $\mathbf{A}^T$ are not invertible or poorly conditioned. Such conditions, however, are easy to identify. Our implementation identifies such numeric issues and uses the linear-time algorithm on such occasions. Numerical issues may also be reduced by restricting all emission likelihoods to be nonzero (i.e., forcing $\mathbf{B}_{t_w}$ to be invertible) or scaling $\mathbf{B}_{t_w}$ with a procedure similar to the scaled Forwards-Backwards algorithm [1].

REFERENCES

1. Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE* (1989), 257–286.