Supplementary Material for : Bayesian On-line Learning of the Hazard Rate in Change-Point Problems

Robert C. Wilson

Department of Psychology, Green Hall,

Princeton University, Princeton, NJ 08540, USA

Matthew R. Nassar and **Joshua I. Gold**

Department of Neuroscience, 116 Johnson Pavilion,

University of Pennsylvania, Philadelphia, PA 19104, USA

May 24, 2010

1 Exponential families

The methods used in this paper are particularly useful when the generating distribution comes from the exponential family. These distributions are completely specified in terms of a finite number of sufficient statistics, η, and can be written in the form:

$$p(\mathbf{x}|\eta) = H(\mathbf{x}) \exp\left(\eta^T \mathbf{U}(\mathbf{x}) - A(\eta)\right) \tag{1}$$

where $A(\eta)$ is given by

$$A(\eta) = \log\left\{\int H(\mathbf{x}) \exp\left(\eta^T \mathbf{U}(\mathbf{x})\right) d\mathbf{x}\right\} \tag{2}$$

Exponential family distributions are particularly convenient because the conjugate prior is also a member of this family, taking the form

$$p(\eta|\chi_0, v) = \tilde{H}(\eta) \exp\left(\eta^T \chi_0 - vA(\eta) - \tilde{A}(\chi_0, v)\right) \tag{3}$$

where χ_0 and v are the prior hyperparameters. Thus, we can write the posterior distribution for run length r_t as

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(r_t)}) &= \int p(\mathbf{x}_{t+1}|\eta) p(\eta|\mathbf{x}_t^r) d\eta \\ &= \frac{\int \prod_{i=t-r_t}^{t+1} p(\mathbf{x}_i|\eta) p(\eta|\chi_0, v) d\eta}{\int \prod_{i=t-r_t}^{t} p(\mathbf{x}_i|\eta) p(\eta|\chi_0, v) d\eta} \\ &= H(\mathbf{x}_{t+1}) \frac{\int \tilde{H}(\eta) \exp\left(\eta^T \left(\chi_0 + \sum_{i=t-r_t}^{t+1} \mathbf{U}(\mathbf{x}_i)\right) - (r_t + v + 1) A(\eta)\right) d\eta}{\int \tilde{H}(\eta) \exp\left(\eta^T \left(\chi_0 + \sum_{i=t-r_t}^{t} \mathbf{U}(\mathbf{x}_i)\right) - (r_t + v + 1) A(\eta)\right) d\eta} \end{aligned} \tag{4}$$

and therefore only have to keep track of a finite number of sufficient statistics; i.e.,

$$\chi_t = \chi_0 + \sum_{i=t-r_t}^{t+1} \mathbf{U}(\mathbf{x}_i) \tag{5}$$

and

$$v_t = v_0 + r_t \tag{6}$$

for each run length to fully specify the distribution.

2 Update algorithm for general change-point hierarchy

To derive the message-passing algorithm for the most general case, we first must introduce a suitable notation. We define $a_0^{(n)}$ and $b_0^{(n)}$ as the prior parameters of the beta distributions over the hazard rate in the nth layer of the hierarchy, and $(a_t^{(n)} - a_0^{(n)})$ and $(b_t^{(n)} - b_0^{(n)})$ to describe the number of change-points and non-change-points counted in each layer. We then group these together as vectors

$$\mathbf{a}_t = \left[a_t^{(1)}, a_t^{(2)}, ..., a_t^{(N-1)}\right] \quad \text{and} \quad \mathbf{b}_t = \left[b_t^{(1)}, b_t^{(2)}, ..., b_t^{(N-1)}\right] \tag{7}$$

to further simplify the notation. Similarly, we define $\mathbf{h}_t$ as

$$\mathbf{h}_t = \left[h_t^{(1)}, h_t^{(2)}, ..., h_t^{(N-1)}\right] \tag{8}$$

But note that it is more convenient not to include $h^{(0)}$ in this vector. In this notation, then, we have

$$p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}) = \sum_{r_t}\sum_{\mathbf{a}_t}\sum_{\mathbf{b}_t} p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}) p(r_t, \mathbf{a}_t, \mathbf{b}_t, \mathbf{x}_{1:t}) \tag{9}$$

In a similar manner to before we can compute $p(r_t, \mathbf{a}_t, \mathbf{b}_t, \mathbf{x}_{1:t})$ recursively; i.e.,

$$\begin{aligned} p(r_t, \mathbf{a}_t, \mathbf{b}_t, \mathbf{x}_{1:t}) &= \sum_{r_{t-1}}\sum_{\mathbf{a}_{t-1}}\sum_{\mathbf{b}_{t-1}} p(r_t, r_{t-1}, \mathbf{a}_t, \mathbf{a}_{t-1}, \mathbf{b}_t, \mathbf{b}_{t-1}, \mathbf{x}_{1:t}) \\ &= \sum_{r_{t-1}}\sum_{\mathbf{a}_{t-1}}\sum_{\mathbf{b}_{t-1}} p(r_t, \mathbf{a}_t, \mathbf{b}_t, \mathbf{x}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}, \mathbf{x}_{1:t-1}) p(r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}, \mathbf{x}_{1:t-1}) \\ &= \sum_{r_{t-1}}\sum_{\mathbf{a}_{t-1}}\sum_{\mathbf{b}_{t-1}} p(r_t, \mathbf{a}_t, \mathbf{b}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(r_t)}) p(r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}, \mathbf{x}_{1:t-1}) \end{aligned} \tag{10}$$

To get a handle on the change-point prior, we can write it as the marginal over $\mathbf{h}_t$, i.e.

$$p(r_t, \mathbf{a}_t, \mathbf{b}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) = \int p(r_t, \mathbf{a}_t, \mathbf{b}_t | \mathbf{h}_t, r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) p(\mathbf{h}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) d\mathbf{h}_t \tag{11}$$

where the integral is over the interval $[0, 1]$ in each dimension and

$$p(\mathbf{h}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) = \prod_{n=1}^{N-1} \frac{\Gamma\left(a_{t-1}^{(n)} + 1\right)\Gamma\left(b_{t-1}^{(n)} + 1\right)}{\Gamma\left(a_{t-1}^{(n)} + b_{t-1}^{(n)} + 1\right)} \left(h_t^{(n)}\right)^{a_{t-1}^{(n)}} \left(1 - h_t^{(n)}\right)^{b_{t-1}^{(n)}} \tag{12}$$

To understand $p(r_t, \mathbf{a}_t, \mathbf{b}_t | \mathbf{h}_t, r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1})$, we note that there are only 2^{N-1} non-zero entries, corresponding to the number of possibilities arising from allowing each of the N levels of the hierarchy to have a change-point or not. Then, for a particular possibility, i, we define $\mathcal{C}_i$ as the set of all levels experiencing a change-point and $\bar{\mathcal{C}}_i$ as the set of levels not experiencing a change-point. Thus we can write

$$
\begin{aligned}
& p(r_t, \mathbf{a}_t, \mathbf{b}_t | \mathbf{h}_t, r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) \\
& \qquad = \sum_{i=1}^{2^N} \delta(r_t - R(r_{t-1}, \mathcal{C}_i)) \delta(\mathbf{a}_t - \mathbf{A}(\mathbf{a}_{t-1}, \mathcal{C}_i)) \delta(\mathbf{b}_t - \mathbf{B}(\mathbf{b}_{t-1}, \mathcal{C}_i)) \prod_{m \in \mathcal{C}_i} h_t^{(m)} \prod_{n \in \bar{\mathcal{C}}_i} (1 - h_t^{(n)})
\end{aligned} \tag{13}
$$

where

$$
R(r_{t-1}, \mathcal{C}_i) = \begin{cases} 0 & \text{if level } N \in \mathcal{C}_i \\ r_{t-1} + 1 & \text{if level } N \notin \mathcal{C}_i \end{cases} \tag{14}
$$

$$
A_n(\mathbf{a}_{t-1}, \mathcal{C}_i) = \begin{cases} 0 & \text{if level } n \in \mathcal{C}_i \\ a_{t-1}^{(n)} & \text{if level } n \notin \mathcal{C}_i \text{ and level } n+1 \notin \mathcal{C}_i \\ a_{t-1}^{(n)} + 1 & \text{if level } n \notin \mathcal{C}_i \text{ and level } n+1 \in \mathcal{C}_i \end{cases} \tag{15}
$$

$$
B_n(\mathbf{b}_{t-1}, \mathcal{C}_i) = \begin{cases} 0 & \text{if level } n \in \mathcal{C}_i \\ b_{t-1}^{(n)} + 1 & \text{if level } n \notin \mathcal{C}_i \text{ and level } n+1 \notin \mathcal{C}_i \\ b_{t-1}^{(n)} & \text{if level } n \notin \mathcal{C}_i \text{ and level } n+1 \in \mathcal{C}_i \end{cases} \tag{16}
$$

which leads to the following expression for the change-point prior

$$
\begin{aligned}
& p(r_t, \mathbf{a}_t, \mathbf{b}_t | r_{t-1}, \mathbf{a}_{t-1}, \mathbf{b}_{t-1}) \\
& \qquad = \sum_{i=1}^{2^N} \delta(r_t - R(r_{t-1}, \mathcal{C}_i)) \delta(\mathbf{a}_t - \mathbf{A}(\mathbf{a}_{t-1}, \mathcal{C}_i)) \delta(\mathbf{b}_t - \mathbf{B}(\mathbf{b}_{t-1}, \mathcal{C}_i)) \prod_{m \in \mathcal{C}_i} \tilde{h}_t^{(m)} \prod_{n \in \bar{\mathcal{C}}_i} (1 - \tilde{h}_t^{(n)})
\end{aligned} \tag{17}
$$

where

$$
\tilde{h}_t^{(n)} = \frac{a_{t-1}^{(n)} + 1}{a_{t-1}^{(n)} + b_{t-1}^{(n)} + 2} \tag{18}
$$

Thus we have a (fairly) simple message-passing algorithm for inference and prediction in a change-point heirarchy.

3 Pseudocode

In the following two boxes we present pseudocode for inferring a constant hazard rate (box 1) and for inference in a three-level change-point hierarchy (box 2).

1. Initialise node $\mathcal{N}(r_0, a_0, t=0)$: $w(r_0=0, a_0=0, t=1)=1$ and nodelist: $\mathcal{L}_{t=0}=\{\mathcal{N}(0,0,0)\}$

2. For all other nodes, set initial value of weight to zero, $w(r_t, a_t, t)=0$

3. **for** each time $t=1$ to T_{max}

    4. Set total weight to zero, $W_{total}=0$, and initialise nodelist to empty set, $\mathcal{L}_t=\emptyset$

    5. **for** all nodes in nodelist $\mathcal{L}_{t-1}$

        6. Observe data $\mathbf{x}_t$

        7. Compute predictive probability: $\pi(\mathbf{x}_t)=p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(r_{t-1})})$

        8. Compute estimate of hazard rate: $\tilde{h}_t=\frac{a_{t-1}+1}{a_{t-1}+b_{t-1}+2}$

        9. Send messages to children:

            To $\mathcal{N}(r_t=r_{t-1}+1, a_t=a_{t-1}, t)$: $w(r_t, a_t, t)=(1-\tilde{h}_{t-1})w(r_{t-1}, a_{t-1}, t-1)\pi(\mathbf{x}_t)$

            To $\mathcal{N}(r_t=0, a_t=a_{t-1}+1, t)$: $w(r_t, a_t, t)=w(r_t, a_t, t)+\ \tilde{h}_{t-1}w(r_{t-1}, a_{t-1}, t-1)\pi(\mathbf{x}_t)$

        10. Add new children to nodelist at time t

        11. Update $W_{total}=W_{total}+w(r_{t-1}, a_{t-1}, t-1)\pi(x_t)$

    12. **endfor**

    13. Normalize: **for** nodes in $\mathcal{L}_t$: $w(r_t, a_t, t)=w(r_t, a_t, t)/W_{total}$; **endfor**

    14. Predict: $p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t})=\sum_{r_t}\sum_{a_t}p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(r_t)})w(r_t, a_t, t)$

15. **endfor**

Box 1 – Pseudo-code for on-line learning of a constant hazard rate

1. Initialise node $\mathcal{N}(r_0, a_0, b_0, 0)$: $w(r_0 = 0, a_0 = 0, b_0 = 0, t = 0) = 1$ and list: $\mathcal{L}_{t=0} = \{\mathcal{N}(0,0,0,0)\}$

2. **for** each time $t = 1$ to T_{max}

   3. Initialise new nodelist to empty set, $\mathcal{L}_t = \emptyset$

   4. **for** all nodes $\in \mathcal{L}_{t-1}$

      5. Observe data $\mathbf{x}_t$ and compute predictive probability: $\pi(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(r_{t-1})})$

      6. Compute estimate of hazard rate: $\tilde{h}_t^{(1)} = \frac{a_{t-1}+1}{a_{t-1}+b_{t-1}+2}$

      7. Messages to children to update weights by

      to $\mathcal{N}(r_t = r_{t-1}+1, a_t = a_{t-1}, b_t = b_{t-1}+1, t)$: $(1-\tilde{h}_{t-1}^{(1)})(1-h^{(0)})w(r_{t-1}, a_{t-1}, b_{t-1}, t-1)\pi(\mathbf{x}_t)$

      to $\mathcal{N}(r_t = 0, a_t = a_{t-1}+1, b_t = b_{t-1}, t)$: $\tilde{h}_{t-1}^{(1)}(1-h^{(0)})w(r_{t-1}, a_{t-1}, b_{t-1}, t-1)\pi(\mathbf{x}_t)$

      to $\mathcal{N}(r_t = r_{t-1}+1, a_t = a_0, b_t = b_0, t)$: $(1-\tilde{h}_{t-1}^{(1)})h^{(0)}w(r_{t-1}, a_{t-1}, b_{t-1}, t-1)\pi(\mathbf{x}_t)$

      to $\mathcal{N}(r_t = 0, a_t = a_0, b_t = b_0+1, t)$: $\tilde{h}_{t-1}^{(1)}h^{(0)}w(r_{t-1}, a_{t-1}, b_{t-1}, t-1)\pi(\mathbf{x}_t)$

      8. Add new children to nodelist at time t

   9. **endfor**

   10. Prune nodes: $\mathcal{L}_t = \texttt{prune}(\mathcal{L}_t)$

   11. Normalize:

       $W_{total} = \sum_{\text{nodes}\in\mathcal{L}_t} w(r_t, a_t, b_t, t)$

       for all nodes in $\mathcal{L}_t$: $w(r_t, a_t, b_t, t) = w(r_t, a_t, b_t, t)/W_{total}$; **endfor**

   13. Predict: $p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}) = \sum_{r_t}\sum_{a_t}\sum_{b_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(r_t)})w(r_t, a_t, b_t, t)$

14. **endfor**

Box 2 – Pseudo-code for on-line learning of the hazard rate in a three-level change-point hierarchy.