Adaptive Sequential Bayesian Change Point Detection

Ryan Turner

Whistler, BC
December 12, 2009

Joint work with Yunus Saatci and Carl Edward Rasmussen
Motivation

- Handle nonstationarity in time series
- Avoid making point estimates of (changing) parameters
- Modular framework
- Tractability
- Online
- Probabilistic predictions
- Minimal hand tuning
Ingredients

- The time since the last change point, namely the run length $r_t$
- The underlying predictive model (UPM) $p(x_t|x_{(t-\tau):t-1} =: x_t^{(\tau)}, \theta_m)$ for any $\tau \in [1, \ldots, (t-1)]$, at time $t$
- The hazard function $H(r|\theta_h)$
- The hyper-parameters $\theta := \{\theta_h, \theta_m\}$

Figure: Sample drawn from BOCPD.
Previous Work

- Test based approaches
- Retrospective Bayesian approaches
- Bayesian Online Change Point Detection (BOCPD) (e.g., Adams & MacKay 2007)
- BOCPD sensitive to hyper-parameters
The BOCPD Algorithm

The goal in BOCPD is to calculate the posterior run length at time $t$, i.e., $p(r_t|x_{1:t})$, sequentially.

$$p(x_{t+1}|x_{1:t}) = \sum_{r_t} p(x_{t+1}|x_{1:t}, r_t) p(r_t|x_{1:t}) = \sum_{r_t} p(x_{t+1}|x_t^{(r)}) p(r_t|x_{1:t}),$$

(1)

$$\gamma_t := p(r_t, x_{1:t}) = \sum_{r_{t-1}} p(r_t, r_{t-1}, x_{1:t})$$

$$= \sum_{r_{t-1}} p(r_t|r_{t-1}) p(x_t|r_{t-1}, x_t^{(r)}) p(r_{t-1}, x_{1:t-1}) \cdot \gamma_{t-1}.$$  

(2)

This defines a forward message passing scheme $p(r_t|x_{1:t}) \propto \gamma_t$. 
Learn by maximizing (log) marginal likelihood, the evidence
Done by decomposing into the one-step-ahead predictive likelihoods

\[
\log p(x_{1:T} \mid \theta) = \sum_{t=1}^{T} \log p(x_t \mid x_{1:t-1}, \theta) \tag{3}
\]

Compute derivatives using forward propagation
The derivatives of the UPM \( \frac{\partial}{\partial \theta_m} p(x_t \mid r_{t-1}, x_t^{(r)}, \theta_m) \)
The derivatives of the hazard function \( \frac{\partial}{\partial \theta_h} p(r_t \mid r_{t-1}, \theta_h) \)
Improvements

- **Pruning**
  - Naive implementation is $O(T^2)$
  - Eliminate low probability messages for $O(T)$

- **Modularity**
  - Any hazard function $H(t) \in [0, 1]$
  - Any model that provides a posterior predictive
  - Gaussian process regression, Bayesian linear regression, and Kernel Density Estimation

- **Caching**
  - Repetitive predictions under given run length
  - Use intelligent caching $p(x_t|r_{t-1}, x_{t}^{(r)})$
Well Log Data

We used the logistic hazard, $H(t) = h \sigma(at + b)$, and used an IID Gaussian UPM, with the aim of detecting changes in mean and variance. After learning the parameters our method has a better predictive likelihood than Adams & MacKay 2007.

Figure: The BOCPD run length distribution on the well log data.
Tried the “30 industry portfolios” data set (from Ken French repository). Change points found coincide with significant events: the climax of the Internet bubble, the burst of the Internet bubble, and the 2004 presidential election.

Figure: The BOCPD run length distribution between 1998 and 2008.
Table: A summary of comparing the negative log predictive likelihoods (NLL) (nats/observation) on test data. We also include the 95% error bars on the NLL and the p-value that the joint model/learned hypers has a higher NLL using a one sided t-test.

<table>
<thead>
<tr>
<th>Method</th>
<th>NLL</th>
<th>error bars</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well Log TIM</td>
<td>1.53</td>
<td>0.0449</td>
<td>&lt;1e-10</td>
</tr>
<tr>
<td>fixed hypers</td>
<td>0.313</td>
<td>0.0267</td>
<td>6e-04</td>
</tr>
<tr>
<td>learned hypers</td>
<td><strong>0.247</strong></td>
<td><strong>0.0293</strong></td>
<td>NA</td>
</tr>
<tr>
<td>Industry Portfolios TIM</td>
<td>42.6</td>
<td>0.246</td>
<td>&lt;1e-10</td>
</tr>
<tr>
<td>indep.</td>
<td>39.64</td>
<td>0.217</td>
<td>0.271</td>
</tr>
<tr>
<td>joint</td>
<td><strong>39.54</strong></td>
<td><strong>0.213</strong></td>
<td>NA</td>
</tr>
</tbody>
</table>
Summary

- Extended work of Adams and MacKay 2007
- Made more general through hyperparameter learning
- Increases predictive performance on real-world datasets
- Extended modularity to non-trivial UPMs
- Improved efficiency using pruning and caching