

# 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  $\tau \in [1, \dots, (t-1)]$ , at time  $t$
- The **hazard function**  $H(r|\theta_h)$
- The hyper-parameters  $\theta := \{\theta_h, \theta_m\}$

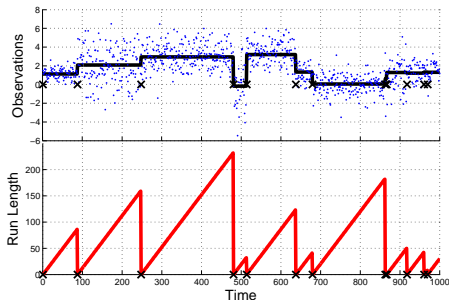


Figure: Sample drawn from BOCPD.

- 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.

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

$$\begin{aligned} \gamma_t := p(r_t, x_{1:t}) &= \sum_{r_{t-1}} p(r_t, r_{t-1}, x_{1:t}) \\ &= \sum_{r_{t-1}} \underbrace{p(r_t|r_{t-1})}_{\text{hazard}} \underbrace{p(x_t|r_{t-1}, x_t^{(r)})}_{\text{likelihood (UPM)}} \underbrace{p(r_{t-1}, x_{1:t-1})}_{\gamma_{t-1}}. \end{aligned} \quad (2)$$

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

$$\log p(x_{1:T}|\theta) = \sum_{t=1}^T \log p(x_t|x_{1:t-1}, \theta) \quad (3)$$

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

- Pruning

- Naive implementation is  $\mathcal{O}(T^2)$
- Eliminate low probability messages for  $\mathcal{O}(T)$

- 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.

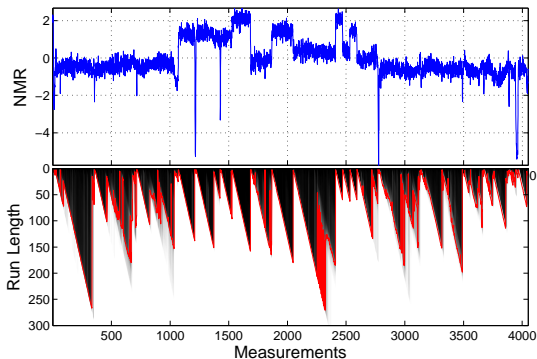


Figure: The BOCPD run length distribution on the well log data.

# Industry Portfolios

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.

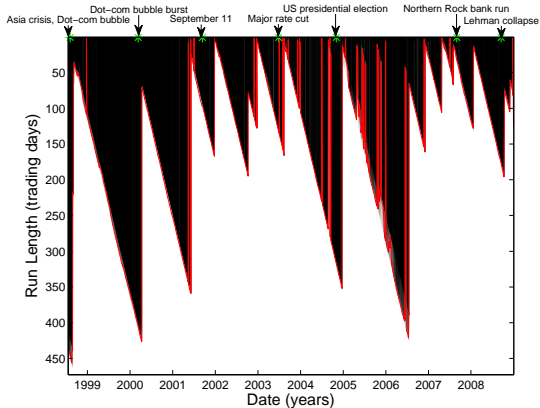


Figure: The BOCPD run length distribution between 1998 and 2008.

# Results

**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.

Well Log			
Method	NLL	error bars	p-value
TIM	1.53	0.0449	<1e-10
fixed hypers	0.313	0.0267	6e-04
learned hypers	<b>0.247</b>	0.0293	NA

Industry Portfolios			
Method	NLL	error bars	p-value
TIM	42.6	0.246	<1e-10
indep.	39.64	0.217	0.271
joint	<b>39.54</b>	0.213	NA

- 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