Adaptively Modeling Cyber-Physical Systems with Applications in Change Detection

Joshua Plasse
Collaborators: Jordan Noble, Kary Myers

Imperial College London
Los Alamos National Labs

September 2017
A cyber-physical system (CPS) combines cyber components with physical devices to form complex systems. Examples include: smart cities, self-driving automobiles and office buildings.

Issues with a CPS include faults and intrusions and can have severe consequences if they go undetected.

This creates the need for a flexible approach for monitoring a CPS for faults and intrusions in real-time.

Talk Discusses:
- an adaptive modeling framework,
- a change detection method.
The data analyzed comes from an HVAC system at Los Alamos National Laboratories (LANL).

Eight story building equipped with 292 sensors reporting communications between building control units and variable air volumes.
Consider monitoring \textbf{bivariate} data streams of the form

$$\langle d_1, d_2, \ldots, d_t, \ldots \rangle \quad d_t = (x_t, s_t),$$

\textbf{physical reading} \quad \textbf{interarrival time}

where $d_t$ is modeled using a \textbf{time evolving normal gamma distribution} with density

$$f(d_t|\theta_t) = \frac{\beta_t^{\alpha_t} \gamma_t^{1/2}}{\Gamma(\alpha_t) \sqrt{2\pi}} s_t^{\alpha_t - 1/2} \exp \left[ -\beta_t s_t - \gamma_t s_t (x_t - \mu_t)^2 \right],$$

and $\theta_t = (\mu_t, \gamma_t, \alpha_t, \beta_t)$.

\textbf{Omitted:} a \textbf{streaming validation procedure} which justifies this modeling assumption on the HVAC data.
The normal-gamma distribution has a lot of desirable properties that make it attractive for monitoring a CPS:
- $X_t | S_t \sim \mathcal{N}(\mu_t, f(\gamma_t, s_t)) \implies$ model dependence,
- $S_t \sim \Gamma(\alpha_t, \beta_t) \implies$ flexibility in monitoring interarrival times.

**Figure 1:** Histograms of the interarrival time distributions for three airflow sensors. **Blue lines** correspond to gamma densities fitted via maximum likelihood estimation to show flexibility.
Adaptive Parameter Estimation

Need a way to **efficiently** and **adaptively** estimate $\theta_t$.

Figure 2: Illustration showing how airflow distributions **drift**.
Idea: exponentially **down-weight** historic data as new data is observed using **forgetting factors**.

Consider a **weighted log-likelihood function** of the form:

\[
\mathcal{L}_\lambda(\theta \mid \mathbf{d}_{1:t}) = \sum_{k=1}^{t} w_k \left[ \alpha \log(\beta) + \frac{\log(\gamma)}{2} - \log (\Gamma(\alpha)) + \left( \alpha - \frac{1}{2} \right) \log(s_k) - \beta s_k - \frac{\gamma s_k (x_k - \mu)^2}{2} \right],
\]

- \( w_k = \prod_{p=k}^{t-1} \lambda_p \) are the **weights** associated with the data,
- \( \lambda_p \in [0, 1] \) is a **forgetting factor**.
The forgetting factors are tuned online via a \textit{stochastic gradient descent} step.
Update Equations

Recursive updates:
\[
\tilde{\mu}_t = \frac{m_{xs,t}}{m_{s,t}} \\
\tilde{\gamma}_t = \frac{n_t}{\xi_t} \\
\tilde{\alpha}_t = \frac{(3 - \zeta_t) + \sqrt{(\zeta_t - 3)^2 + 24\zeta_t}}{12\zeta_t} \\
\tilde{\beta}_t = \frac{\tilde{\alpha}_t n_t}{m_{s,t}}
\]

Auxiliary updates:
\[
n_t = \lambda_{t-1}n_{t-1} + 1 \\
m_{f(x,s),t} = \lambda_{t-1}m_{f(x,s),t-1} + f(x_t, s_t) \\
\xi_t = m_{x^2s,t} - \tilde{\mu}_t^2 m_{xs,t} \\
\zeta_t = \log \left( \frac{m_{s,t}}{n_t} \right) - \frac{m_{\log(s),t}}{n_t} \\
m_{f(x,s),t} = \sum_{k=1}^{t} w_k f(x_k, s_k)
\]

These updates do not require any observations prior to time \(t\) to maintain up-to-date estimates. This makes the adaptive modeling framework efficient for the data streaming from a CPS.
The Change Detector

Consider two normal-gamma distributions: one with adaptive parameter estimates $\tilde{\theta}_t$ and the other having static (usual MLEs) estimates $\hat{\theta}_t$. The main idea:

- during stationary periods $\tilde{\theta}_t$ and $\hat{\theta}_t$ should be roughly the same,
- when the stream experiences a change $\tilde{\theta}_t$ and $\hat{\theta}_t$ should diverge.

If this dissimilarity can be quantified, a change could be flagged whenever $\tilde{\theta}_t$ and $\hat{\theta}_t$ get “too far” apart.
Use the **KL-Divergence** and flag a change whenever

\[ \text{KL}_t \left( f(\tilde{\theta}_t) \| f(\hat{\theta}_t) \right) > \varepsilon_t. \]

**Simulation study:**

- compare to R’s implementation of **PELT** (Killick, Eckley 2014),
- consider changes in all subsets of \( \theta_t = (\mu_t, \gamma_t, \alpha_t, \beta_t) \),
- use average detection delay and proportion of change points correctly detected and detections that are not false as performance metrics.
Given a performance measure, all points above the line $y = x$ indicates our method outperforming PELT.

Our method, overall, outperforms PELT with respect to the speed (left panel) and accuracy of true detections (center), but is marginally worse for false detections (right panel).
Conclusions

This talk has presented a complete framework for monitoring a general CPS that allows for:

- temporally adaptive and efficient parameter estimation,
- and change detection.

Accepted for publication:

Contact: jp215@ic.ac.uk
Appendix:

Model validation:

- Suppose we “think” a stream \( v_{1:N} \) has CDF \( F_V \) parameterized by some vector \( \theta \).
- Method idea: apply the PIT sequentially to \( v_{1:N} \) and compute a KS score.

The method

- For each \( t = 1, \ldots, N \)
  - compute: \( \tilde{\theta}_t \),
  - apply PIT: \( \tilde{u}_t = 1 - F_V(v_t | \tilde{\theta}_t) \).
- Compute empirical CDF \( \tilde{F}_u \) using \( \{\tilde{u}_t\}_{t=1}^N \).
- Return the score \( \max_{u \in \{\tilde{u}_t\}_{t=1}^N} \left| \tilde{F}_U(u) - u \right| \).