Changepoint detection for time series prediction

Allen B. Downey

Olin College of Engineering

My background:

- Predoc at San Diego Supercomputer Center.
- Dissertation on workload modeling, queue time prediction and malleable job allocation for parallel machines.
- Recent: Network measurement and modeling.
- Current: History-based prediction.

Connection?

- Resource allocation based on prediction.
- Prediction based on history.
- Historical data characterized by changepoints (nonstationarity).

Three ways to characterize variability:

- Noise around a stationary level.
- Noise around an underlying trend.
- Abrupt changes in level: changepoints.

Important difference:

- Data prior to a changepoint is irrelevant to performance after.

Example: wide area networks

- Some trends (accumulating queue).
- Many abrupt changepoints.
  - Beginning and end of transfers.
  - Routing changes.
  - Hardware failure, replacement.

Example: parallel batch queues

- Some trends (daily cycles).
- Some abrupt changepoints.
  - Start/completion of wide jobs.
  - Queue policy changes.
  - Hardware failure, replacement.

My claim:

- Many systems are characterized by changepoints where data before a changepoint is irrelevant to performance after.
- In these systems, good predictions depend on changepoint detection, because old data is wrong.

Discussion?

Two kinds of prediction:

- Single value prediction.
- Predictive distribution.
  - Summary stats.
  - Intervals.
  - $P(error > thresh)$
  - $E[cost(error)]$

If you assume stationarity, life is good:

- Accumulate data indefinitely.
- Predictive distribution = observed distribution.

But this is often not a good assumption.

If the system is nonstationary:

- Fixed window? Exponential decay?
- Too far: obsolete data.
- Not far enough: loss of useful info.

If you know where the changepoints are:

- Use data back to the latest changepoint.
- Less information immediately after.

If you don't know, you have to guess.

$$P(i) = \text{prob of a changepoint at time } i$$

Example:

- 150 data points.
- $P(50) = 0.7$
- $P(100) = 0.5$

How do you generate a predictive distribution?

Two steps:

- Derive $P(i+)$: prob that i is the latest changepoint.
- Compute weighted mix going back to each i.

Example:

$$\begin{array}{ccc} & P(50) = 0.7 & P(100) = 0.5 \\ P(\emptyset) = 0.15 & P(50+) = 0.35 & P(100+) = 0.5 \end{array}$$

Predictive distribution =

$$\begin{aligned} 0.50 \quad &\cdot \quad edf(100, 150) \oplus \\ 0.35 \quad &\cdot \quad edf(50, 150) \oplus \\ 0.15 \quad &\cdot \quad edf(0, 150) \end{aligned}$$

So how do you generate the probabilities $P(i+)$?

Three steps:

- Bayes' theorem.
- Simple case: you know there is 1 changepoint.
- General case: unknown # of changepoints.

Bayes' theorem (diachronic interpretation)

$$P(H|E) = \frac{P(E|H)}{P(E)} P(H)$$

- H is a hypothesis, E is a body of evidence.
- $P(H|E)$: posterior
- $P(H)$: prior
- $P(E|H)$ is usually easy to compute.
- $P(E)$ is often not.

Unless you have a suite of exclusive hypotheses.

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{P(E)}$$

$$P(E) = \sum_{H_j \in S} P(E|H_j)P(H_j)$$

In that case life is good.

- If you know there there is exactly one changepoint in an interval...
- ...then the $P(i)$ are exclusive hypotheses,
- and all you need is $P(E|i)$.

Which is pretty much a solved problem.

What if the # of changepoints is unknown?

- $P(i)$ are no longer exclusive.
- But the $P(i+)$ are.
- And you can write a system of equations for $P(i+)$.

$$P(i^+) = P(i^+|\oslash)\ P(\oslash) + \sum_{j<i} P(i^+|j^{++})\ P(j^{++})$$

- $P(j^{++})$ is the prob that the second-to last changepoint is at i.
- $P(i^+|j^{++})$ reduces to the simple problem.
- $P(\oslash)$ is the prob that we have not seen two changepoints.
- $P(i^+|\oslash)$ reduces to the simple problem (plus).

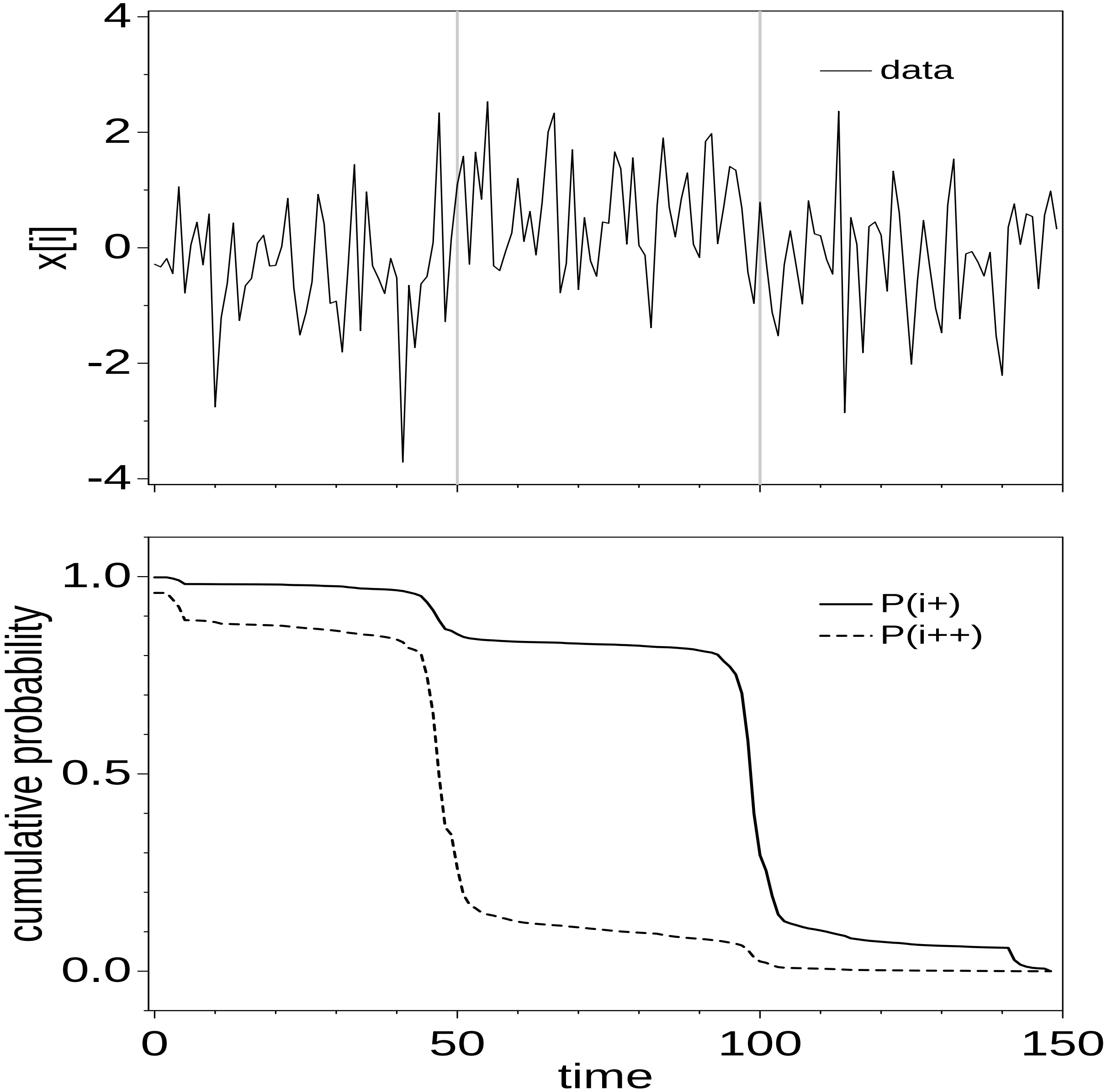
Great, so what's $P(j^{++})$?

$$P(j^{++}) = \sum_{k>j} P(j^{++}|k^{+})\ P(k^{+})$$

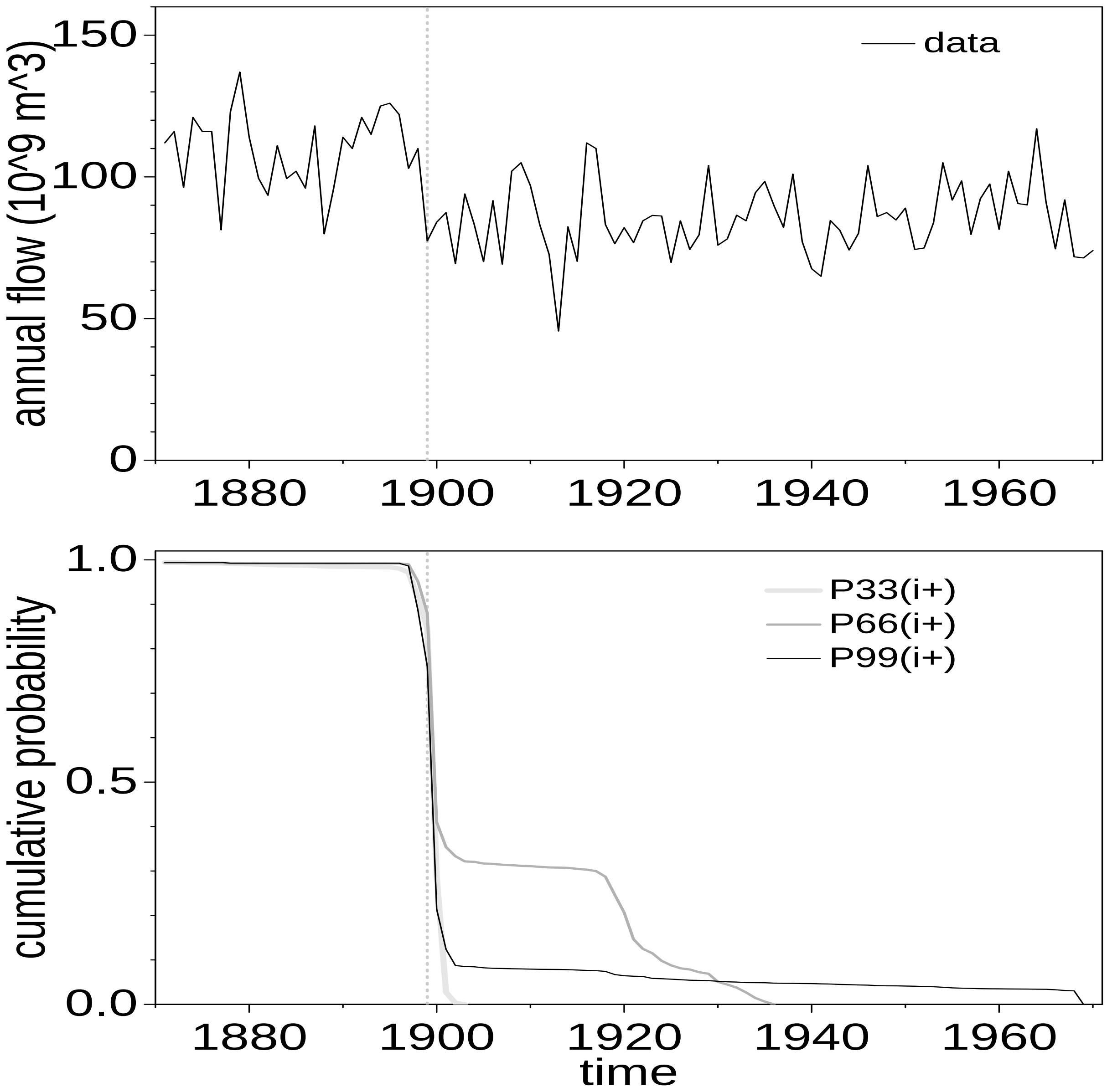
- $P(j^{++}|k^{+})$ is just $P(j^{+})$ computed at time k.
- So you can solve for $P(^{+})$ in terms of $P(^{++})$.
- And $P(^{++})$ in terms of $P(^{+})$.
- And at every iteration you have a pretty good estimate.
- Paging Dr. Jacobi!

Implementation:

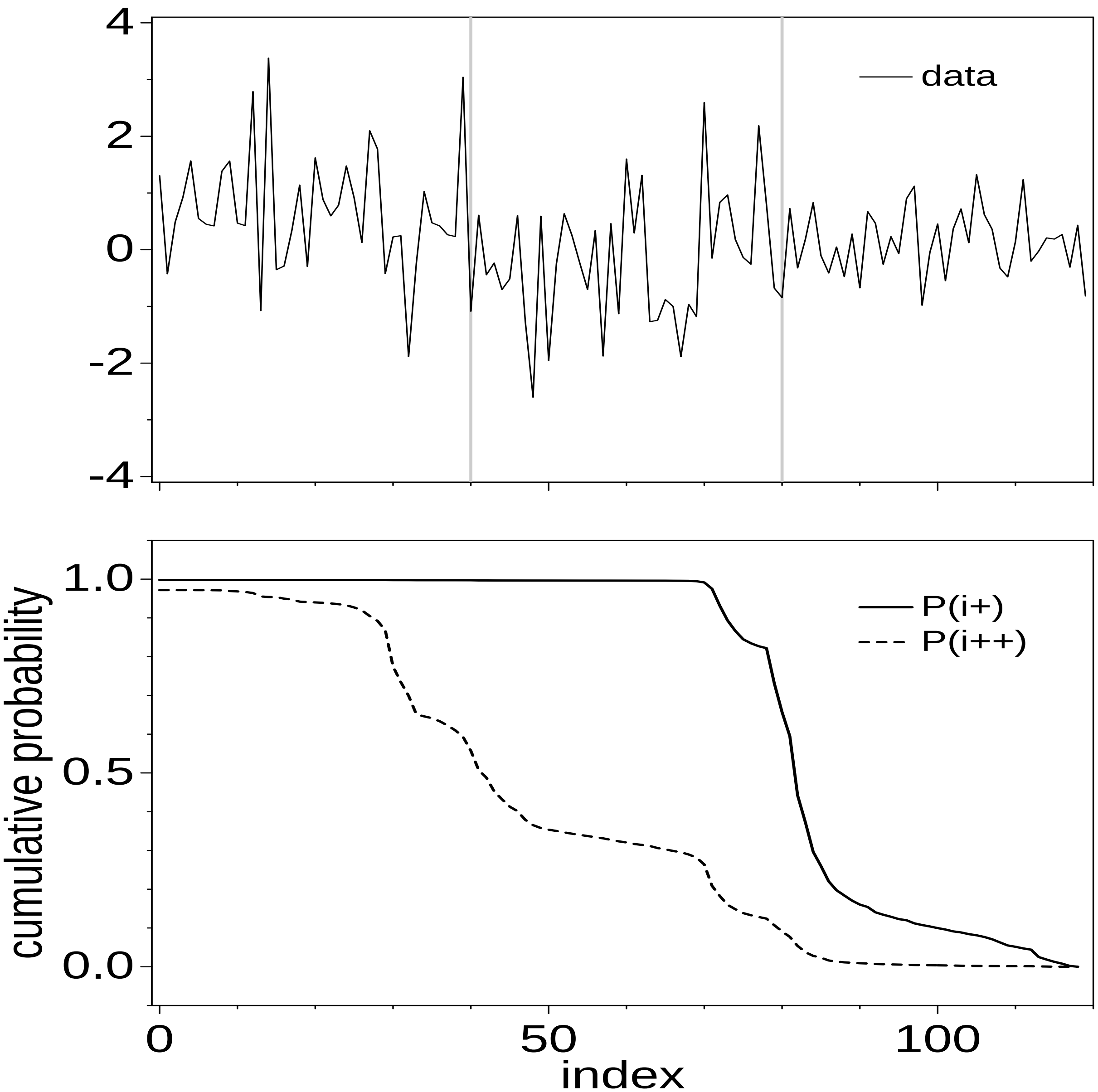
- Need to keep $n^2/2$ previous values.
- And $n^2/2$ summary statistics.
- And it takes n^2 work to do an update.
- But, you only have to go back two changepoints,
- ...so you can keep n small.



- Synthetic series with two changepoints.
- $\mu = -0.5, 0.5, 0.0$
- $\sigma = 1.0$
- $P(\oslash) = 0.04$



- The ubiquitous Nile dataset.
- Change in 1898.
- Estimated probs can be mercurial.



- Can also detect change in variance.
- $\mu = 1, 0, 0$
- $\sigma = 1, 1, 0.5$
- Estimated $P(i^{+})$ is good.
- Estimated $P(i^{++})$ less certain.

- Qualitative behavior seems good.
- Quantitative tests:
  - Compare to GLR for online alarm problem.
  - Test predictive distribution with synthetic data.
  - Test predictive distribution with real data.

Changepoint problems:

- Detection: online alarm problem.
- Location: offline partitioning.
- Tracking: online prediction.

Proposed method does all three. Starting simple...

Online alarm problem:

- Observe process in real time.
- μ_0 and σ known.
- τ and μ_1 unknown.
- Raise alarm ASAP after changepoint.
- Minimize delay.
- Minimize false alarm rate.

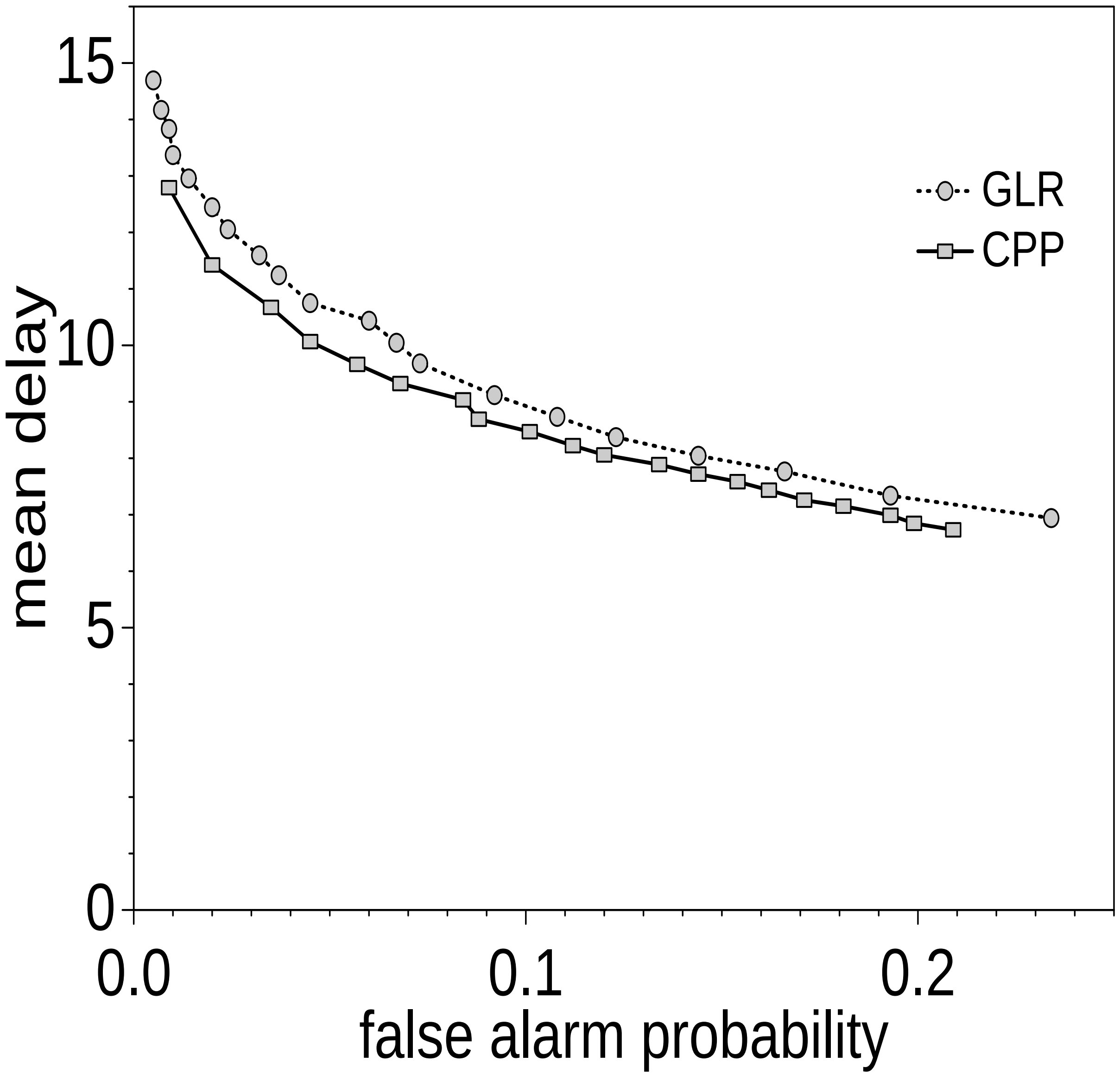
GLR = generalized likelihood ratio.

- Compute decision function g_k.
- $E[g_k] = 0$ before the changepoint,
- ... increases after.
- Alarm when $g_k > h$.
- GLR is optimal when μ_1 is known.

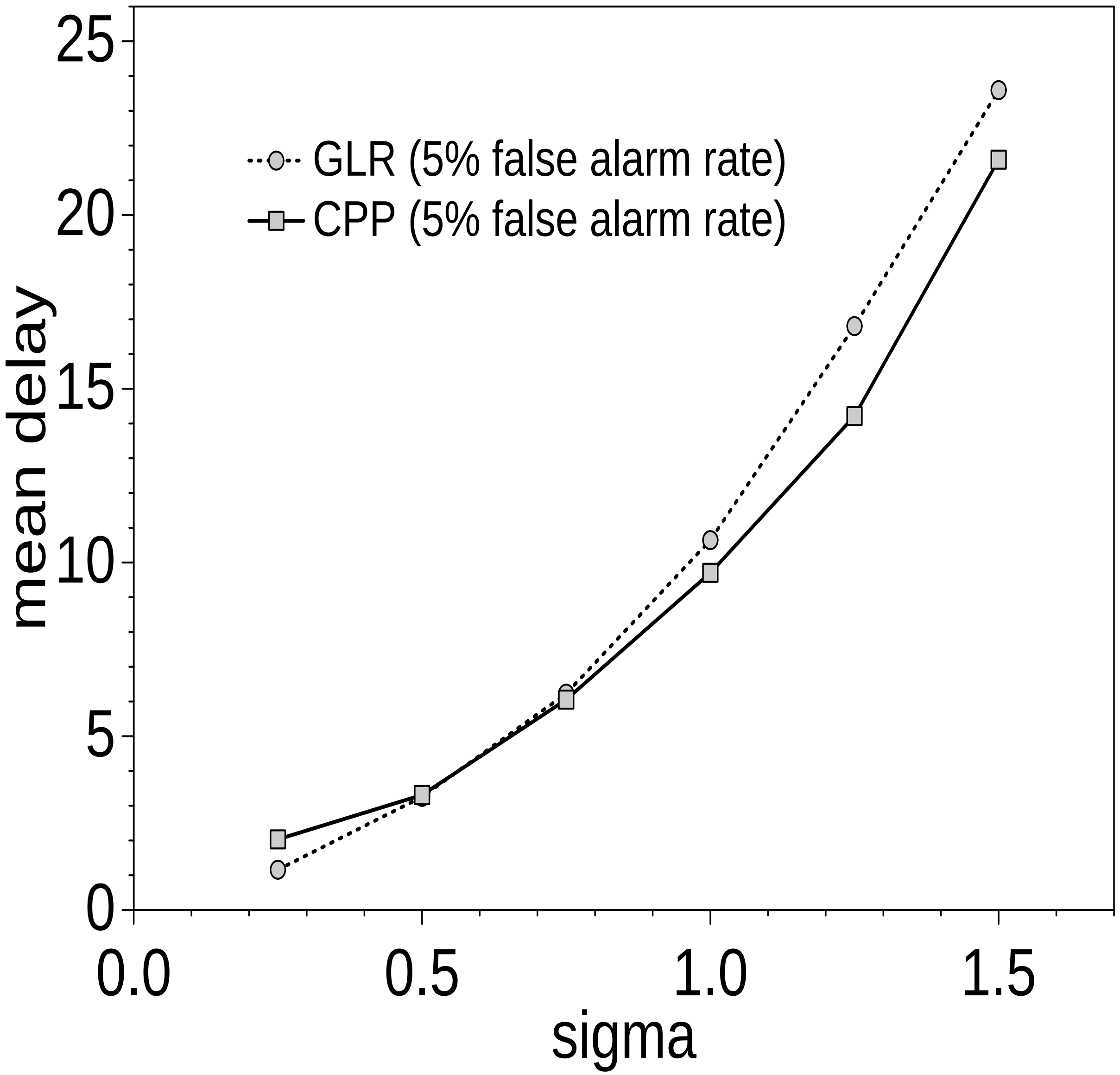
CPP = change point probability

$$P(changepoint) = \sum_{i=0}^{n} P(i^+)$$

- Alarm when $P(changepoint) > thresh$.



- $\mu = 0, 1$
- $\sigma = 1$
- $\tau \sim \text{Exp}(0.01)$
- Goodness = lower mean delay for same false alarm rate.



- Fix false alarm rate = 5%
- Vary σ.
- CPP does well with small S/N.

So it works on a simple problem.

Future work:

- Other changepoint problems (location, tracking).
- Other data distributions (lognormal).
- Testing robustness (real data, trends).

Related problem:

- How much categorical data to use?
- Example: predict queue time based on size, queue, etc.
- Possible answer: narrowest category that yields two changepoints.

Good news:

- Very general framework.
- Seems to work.
- Many possible applications.

Bad news:

- Need to apply and test in real application.
- n^2 space and time may limit scope.

- More at `allendowney.com/research/changepoint`
- Or email `downey@allendowney.com`