Change-point methods in disturbance identification and probabilistic labeling of events

Emmanuel Yashchin (joint work with Nianjun Zhou and Anuradha Bhamidipaty)

IBM Research, Yorktown Heights, New York

Joint Statistical Meetings, August 2021

Outline

- Motivation, Machine Learning Setting
- Baseline
- Change-point approach to disturbance identification
- Application to Service Ticket Labeling
- Concluding Remarks

1. Motivation

Data: File DB = Electric utility service records (tickets).

Fields of a service ticket: Incident ID, Outage start/end times, Substation, *Storm ID,* Cause description, Number of customers affected, etc.

Question of interest: Is the ticket storm-related?

Data quality issues: The field *Storm ID* is often missing or unreliable.

Needed: To bring *DB* to the state where all storms are identified, and tickets labeled as storm-related or not.

With labeled data, we can answer questions of type:

- How many storm-related tickets are expected in each period of time, by substation?
- What are contribution of infrastructure factors (number of poles, transformers, miles of lines) to the cost of outages?
- What are contribution of Geographic features?
- Effect of weather-related variables (precipitation, wind speeds, wind gusts)?

General Machine Learning Setting

Data: Unlabeled or Labeled Unreliably.

Labeling: Infeasible or prohibitively expensive => No training set.

Of interest: *Probabilistic Labels (PL)*.
E.g. *Prob{Ticket is storm-related} = 0.8*

Used in many areas, incl.:
Survival Analysis (Flehinger et al. 2002)
Probabilistic Networks (Peleg 1980)
Binary Classification (Peng et al. 2014)
Metric Learning (Huai et al. 2018)
Active Learning (Xue 2020)

PL generation: Costly, typically requires a labeled training set.

Change-point methods: Can be used for efficient and automated generation of PL under the conditions when disturbances do not dominate data set. In such cases, labeled training set is not needed.

Basic idea: *Baseline* process characteristics using *robust* estimation methods and imputation could be obtained for the complete time range; Disturbance periods are then identified, and their characteristics contrasted against baseline. PL can be obtained using a form of contrasts.

Basic Approach

(i) Estimate *baseline* process characteristics using *robust* estimation methods

(ii) Use *imputation* to ensure that *baseline* covers the complete time range

(iii) Obtain and *parametrize* a *measure of deviation* between the process characteristics and the *baseline.*

(iv) Establish acceptable / unacceptable levels for the *parameters*.

(v) Define and set performance characteristics (false alarm rate, sensitivity) for control scheme responsible for *detecting* disturbances.

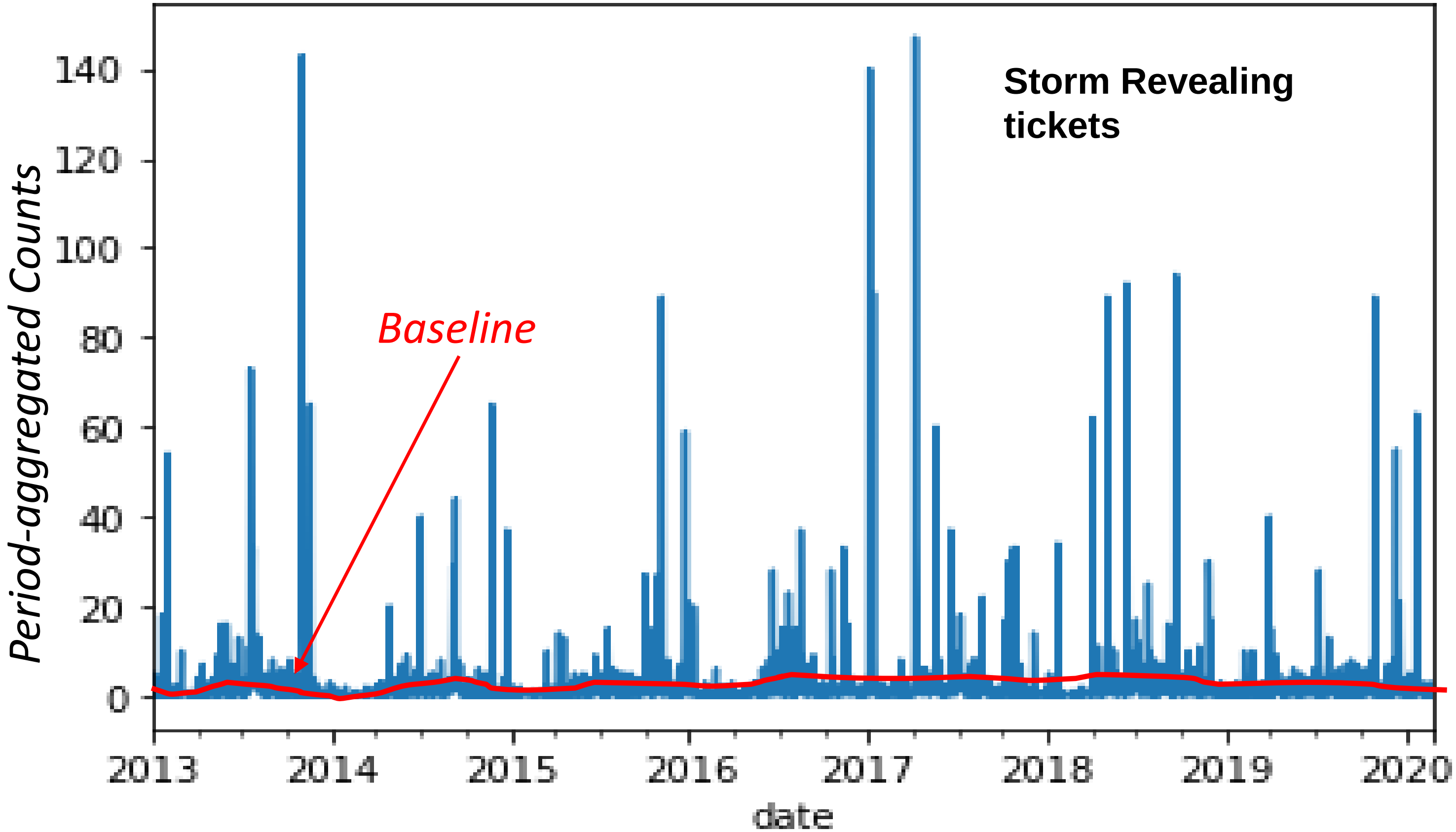
(vi) Apply control scheme and *identify* disturbances, *endpoints*.

(vii) Obtain *Probabilistic Labels* (PL)

(viii) Validate methodology against any partial labeling, if available; validate relevance against other objectives (e.g., prediction, classification).

2. Baseline

Consider the problem of Utility Service Ticket management. We use a special category of tickets, named Disturbance-Revealing-Tickets (DRT) to implement the task of disturbance (*storm*) identification. Data is summarized *daily.* Counts for or a given substation *XYZ:*



Main task: Identify *baseline* rate. After that, we will be able to identify *storm* periods.

Robust (trimmed) method for daily baseline

For a given month, we have the daily counts of *DRT-type* tickets as a random variable vector $\boldsymbol{X} = \{X_i\}$. Here the index i is the *date.*

One way to obtain robust (trimmed) estimation of the mean *daily* rate for X under non-storm conditions is to compute the *monthly* rate and then assign this rate for every day of the month. Suppose the month contains *D* days. Let *r* = number of points trimmed from each side.

Defaults: D = 30, r = 10

Procedure:

- Remove (trim) the top *r* *daily* rates & bottom *r* rates from *monthly* data.
- Calculate the trimmed mean $\bar{X}_{\{r\}}$ from the remaining (*D* – 2**r*) data points.
- Apply bias adjustment *b*, set $\hat{\lambda} = \bar{X}_{\{r\}} + b$
- Prevent $\hat{\lambda}$ from being too small => Apply threshold β_l: $\hat{\lambda} = \max(\hat{\lambda}, \beta_l)$

Defaults: b = 0.15, b_l = 0.2

Baseline: Sequence $\{\hat{\lambda}_i\}$, *i = 1,2,...* *Post-processing:* Optional (e.g. via local smoothing)

Other possibilities: E.g., apply above procedure to sliding window of total length = *D* days, with *i = mid-point* of window.

Standardization

Various control charting procedures could be applied to the sequence of *daily* counts $\{X_i\}$, *i = 1,2,...* to detect unacceptably high deviations from the estimated baseline rates $\{\hat{\lambda}_i\}$.
One simple way: convert $\{X_i\}$ to scores $\{Y_i\}$ via:

$$Y_i = \left\{ \frac{X_i - \hat{\lambda}_i}{\hat{\sigma}_i} \right\}$$

where $\hat{\sigma}_i$ is the *scaling* process. E.g., $\hat{\sigma}_i$ = sqrt($\hat{\lambda}_i$), if we are willing to work under Poisson assumption – however, there are several complicating factors:

- *over-dispersion* in $\{X_i\}$
- *serial correlation* in $\{X_i\}$, $\{\hat{\sigma}_i\}$, $\{\hat{\lambda}_i\}$

Nevertheless, applying an *adjusted* Cusum procedure to $\{Y_i\}$ enables one to detect disturbances and identify regimes and endpoints.

3. Change-point approach

Given: the *scores* $\{Y_i\}$, i = 1, 2, …

Define: the set of scheme values $\{S_i$, i = 1, 2, …$\}$ as follows:

$$S_0 = s_0, \quad S_i = \max[\,0,\ S_{i-1} + (Y_i - k)] \quad (evidence\ curve),$$

where k = *reference value*, s_0 = *headstart.*

$$k = (\mu_{Y,accept} + \mu_{Y,unaccept})/2$$

Defaults: $\mu_{Y,accept} = \mu_0 = 0,\ \mu_{Y,unaccept} = \mu_1 = 2$ ➔ $k = 1$

Declare: Disturbance episode at time T if $S_i > h$, where h is chosen via:

Average Run Length $\{\mu = \mu_{Y,\ accept}\}$ = ARL_0 (False alarm rate)

Notes: (a) h *can* be obtained using approximation: $\{Y_i\} = N(\mu_Y, 1)$, however calibration / adjustments are typically needed.

Recommended: Additive correction for h. E.g. (i) we want ARL_0 = 15000, (ii) the iid Normal assumption suggests: $ARL_0(h = 4, k = 1)$ = 15000. (iii) however, given the nature of $\{Y_i\}$, add 2 to h to achieve goal => $h = 6$

(b) Alternative design *by quantile:* select h by solving:

Prob{Run Length > $L0$ | $\mu = \mu_{Y,\ accept}\}$ = 0.99 (False alarm rate)

Disturbance boundary determination

Process $\{X_i\}$ undergoes regime-switching: Baseline -> Disturbance -> Baseline …

The score process $\{Y_i\}$ switches accordingly.

Goal: Estimate disturbance boundaries.

Settings: On-line vs Off-line

Possible Cusum deployments: *re-starting* vs *non-restarting.*

Re-starting mode: Typically, auto-restart to s_0 is not recommended in *quality monitoring* applications, as restart should only be done after validating that process is acceptable (this might require special interventions not reflected in data).

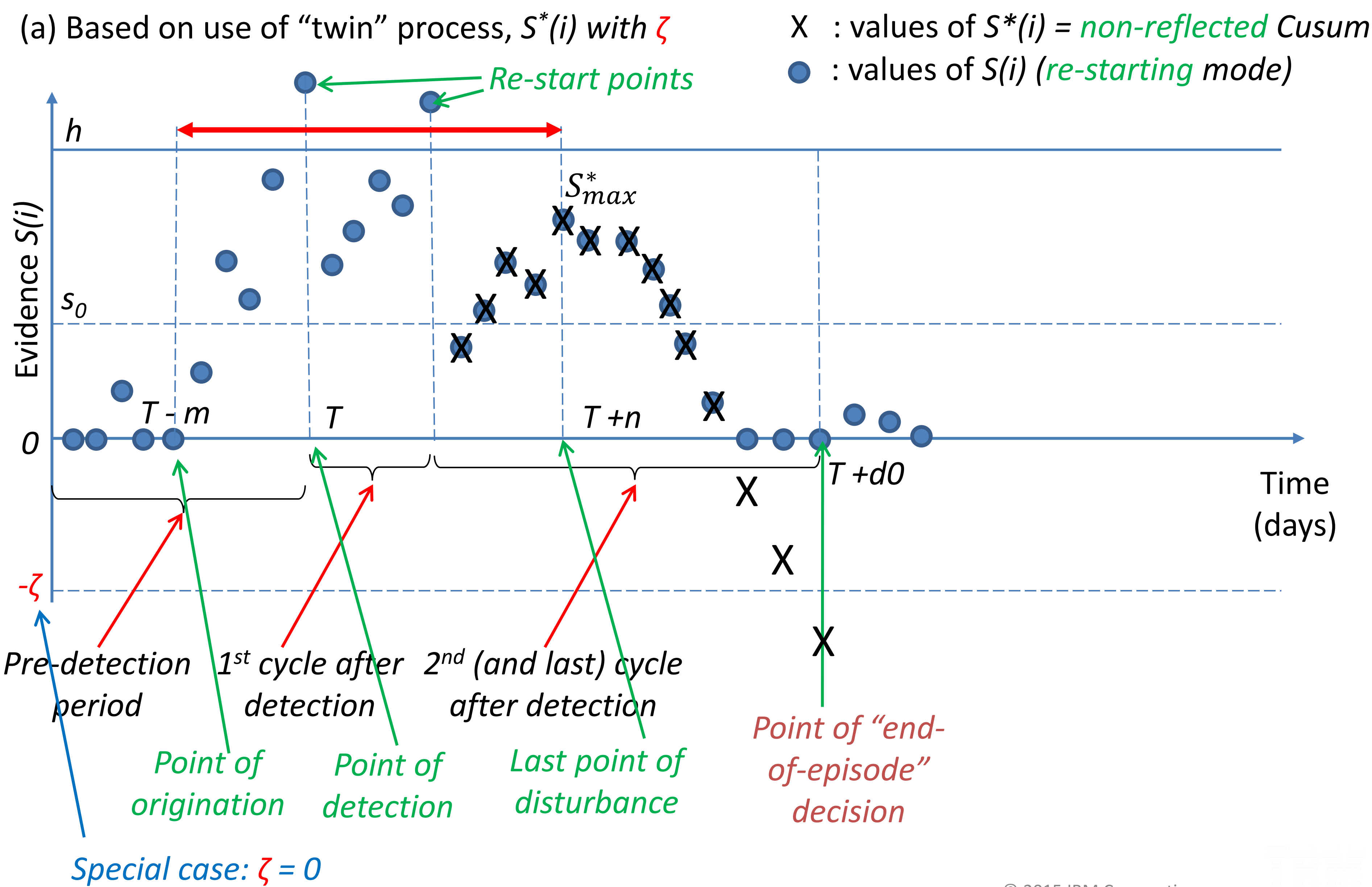
However, this mode is useful for *disturbance identification.*

Non-restarting mode: Sometimes used with *reflecting upper boundary* for Cusum (e.g., Gandy and Lau 2013); use without such boundary also possible (e.g. Yashchin 2012).

Asymmetric role of left / right bounds. Beginning part of disturbance often shows different stochastic behavior than ending part.

Performance criteria: Can be of standard type, e.g., MSE. However, boundary determination is often an *intermediate problem,* so the ultimate criterion should be tied to properties of probabilistic labeling and the related models.

Basic Procedures

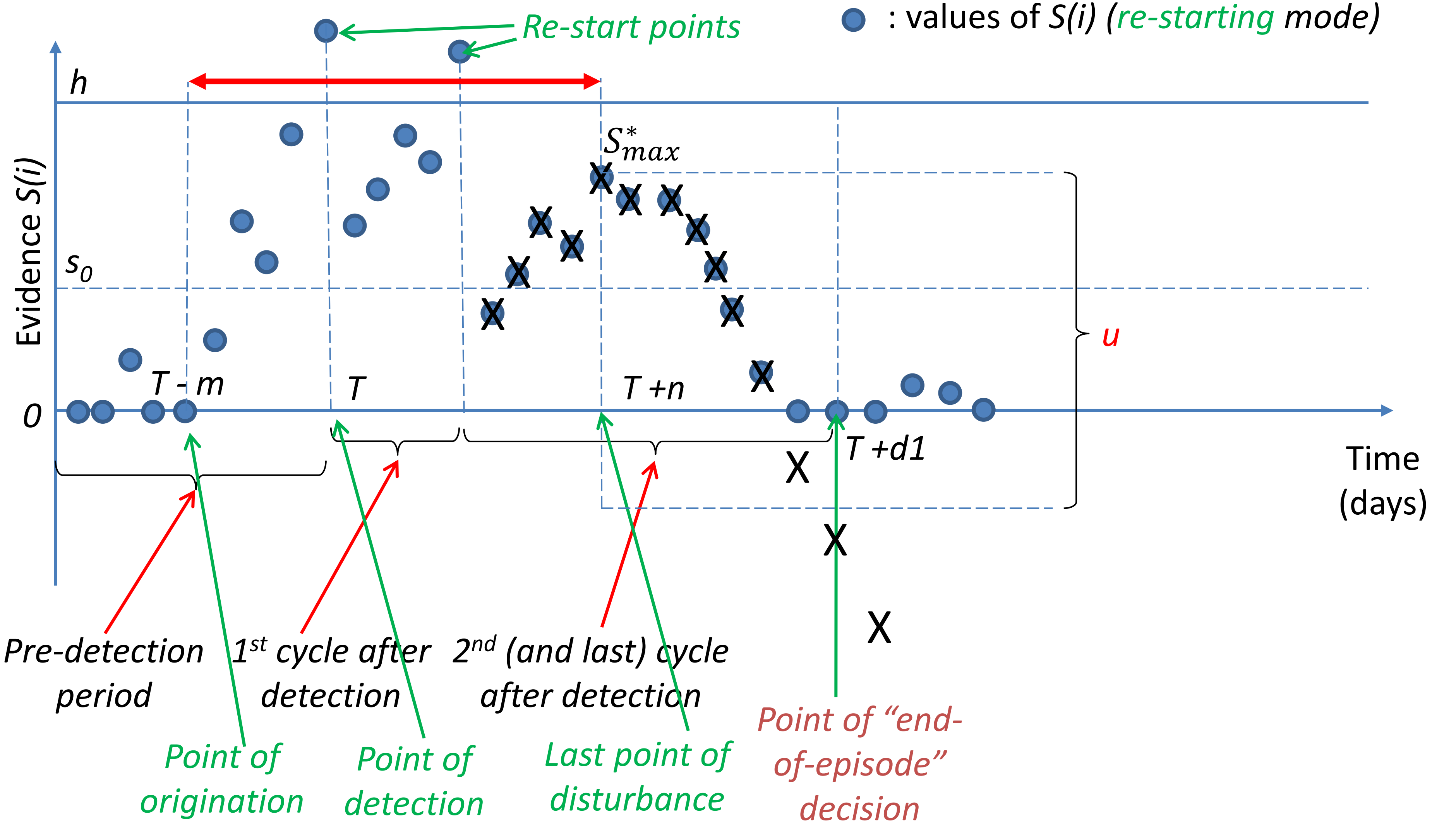


Procedures (cont)

(b) Based on “twin” process, $S^*(i)$ with *u*.

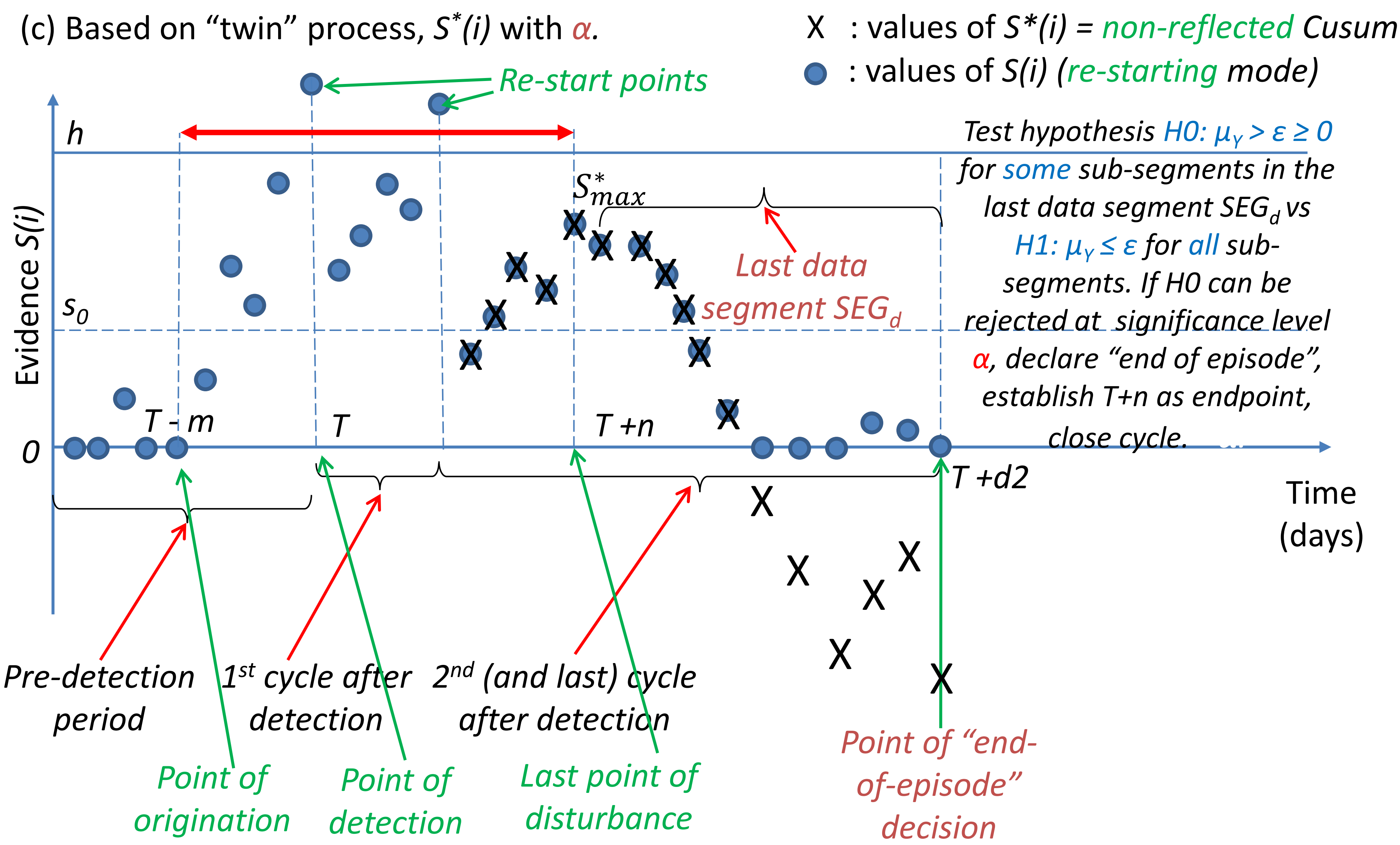
X : values of $S^*(i)$ = *non-reflected Cusum*

● : values of $S(i)$ (*re-starting mode*)



Procedures (cont)

(c) Based on "twin" process, $S^*(i)$ with α.



Few generalizations

1. Estimating the starting point T-m of disturbance as the first point of a signal-triggering trajectory introduces *positive bias*. This can be addressed by expanding the starting point *leftward* by including additional points (sequentially) as long as data the values Y(i) support the hypothesis of elevated rate, e.g.,

   (i) as long as $Y(i) > \mu_{0,}$ or

   (ii) as long as hypothesis of *disturbance* is supported vs *baseline* (can use process similar to that of establishing T+d, S_{max} and T+n, but going *leftward*).

2. Dynamic boundary adjustment: we may *not be obliged* to set the starting point at the detection time T. We can also be permitted to adjust disturbance boundaries and new info comes in.

3. Enhancement are possible based on area-specific disturbance patterns. E.g., for storms, it might be known that the effects appear within a *short time* but fade out *gradually.*

4. Covariates can be incorporated into the algorithm, e.g., via baseline adjustment.

Probabilistic Labeling

Let *p = Prob{Ticket is Disturbance-related}*

A *point estimate* of *p* for day *i* (delivered as PL):

$$\hat{p}_i = \max[0, \frac{X_i - \hat{\lambda}_i}{X_i}]$$

Confidence bounds: require additional assumptions.

4. Application to Service Ticket Labeling

Input: *File DB* = Electric utility service records (tickets). Number of Disturbance-Revealing-type tickets (DRTs) > 140,000, covering 55 substations, over the period of 7 years. With *default* processing setup, we return:

Output: *File DBM = DB + Info on detected storms + Probabilistic Labels*

High probability tickets not associated with knows storms

Original Fields

Added Fields

Incid_id	Outage_start_time	Substation	Storm_id	Cause_desc	Cust_affected	Found_storm_with_substation	Status	P_Label	Known_storm_ids
3584	1/9/2013 20:21	px1		EQUIPMENT\	100		N	0	
3643	1/9/2013 20:33	ox1		TREE\FALLEN	34		N	0	
3824	1/9/2013 20:48	ox2		TREE\FELL O	1		N	0	
3943	1/9/2013 20:58	fx1		TREE\FALLEN	65	fx1_2013-01-08_2013-01-09	E	0.95	[]
3943	1/9/2013 20:58	fx1		TREE\FALLEN	270	fx1_2013-01-08_2013-01-09	E	0.95	[]
4324	1/9/2013 21:40	wx1		TREE\FALLEN	32		N	0	
4363	1/9/2013 21:55	fx1		EQUIPMENT\	1	fx1_2013-01-08_2013-01-09	E	0.95	[]
4443	1/9/2013 22:16	hx1		TREE\FALLEN	0		N	0	
4463	1/9/2013 22:16	px2		TREE\FALLEN	28		N	0	
4503	1/9/2013 22:26	bx1		TREE\FELL O	1		N	0	
4684	1/9/2013 23:34	wx1		TREE\BRANC	30		N	0	
5383	1/10/2013 9:03	mx1		TREE\FALLEN	114	mx1_2013-01-10_2013-01-20	S	0.9	[127000]
5503	1/10/2013 9:19	bx2		TREE\FELL O	1		N	0	
5523	1/10/2013 9:20	mx1		TREE\FALLEN	6	mx1_2013-01-10_2013-01-20	S	0.9	[127000]
5783	1/10/2013 10:29	px3		TREE\FALLEN	1		N	0	
6543	1/10/2013 14:48	tx1		EQUIPMENT\	1		N	0	

No Storm_Id assigned in *original* data

Associating with Known Disturbances (Storms)

Validation

1. Task is challenging, esp. in the presence of data quality issues.
2. Tickets with *high probability* of being storm-related (e.g., $p > p_0 = 0.5$) are of special use.
3. Availability of *partially labeled* input is helpful. E.g., see measures based on:

D1 = # of pre-labeled tickets

D2 = # of high-probability tickets falling in the vicinity of *known storms*

D3 = # of discovered storm periods

D4 = # of high-probability tickets falling in the vicinity of *discovered storms* (*i.e. known + new*),

[D1 ∩ {not assigned a label p > 0.5}] / D1 = 3.1%

[D2 ∩ {not pre-labeled}] / D2 = 31% => high potential for discovering *additional* storm-related tickets associated with *known storms*).

[D3 ∩ {are not associated with known storms}] / D3 = 33%

[D4 ∩ {coming from the "*new*" part}] / D4 = 13% => indicates presence of missed storms

4. Re: *falsely identified storm periods*. In the absence of training data set, newly discovered storms were validated by customer – they indicated agreement with our results.

5. Other forms of validation: using *geographical neighborhoods*, variables recorded at *weather stations*.

Concluding Remarks

1. Change-point methods can play an important role in *machine learning* areas, providing opportunity to address *data quality* issues.
2. In areas where *probabilistic labels* are used, change-point methods can be helpful in of generating them, even in the absence of labeled training data.
3. Robust estimation techniques (e.g., trimming) useful for Baseline derivation.
4. Cusum methodology enables efficient determination of disturbance boundaries. It is adaptable to various requirements for decision-making time frames.
5. Validation feasible but challenging.