Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.

Statistics in Cyber-Security

Nick Heard

Department of Mathematics, Imperial College London
&
Heilbronn Institute for Mathematical Research

25 July, 2018

Imperial College London

Heilbronn Institute for Mathematical Research

Collaborators

- Matthew Price-Williams, Xinyu Zhang, Francesco Sanna Passino, Karl Hallgren (Imperial College)
- Silvia Metelli (Alan Turing Institute)
- Patrick Rubin-Delanchy (University of Bristol)
- Melissa Turcotte (Los Alamos National Laboratory)

Statistical Cyber Research

Data science techniques have an important role to play in the next generation of cyber-security defences.

Inside a typical enterprise computer network, a number of high-volume data sources are available which could enable the discovery and prevention of cyber-attacks and other nefarious network activity.

At Imperial, our interests are in developing statistical, probability model-based techniques for identifying the most subtle intrusion attempts using these data sources.

The advantage of such approaches is the ability to learn, from historical data, complex patterns of normal computer and network behaviour, so that anomalies can be detected which would not stand out otherwise; one example is unusual network traversal using valid (but possibly stolen) credentials.

Data sources and platforms

- Network flow data — IP→IP, next level protocol, ports, TCP flags, number of packets/bytes, start time, duration
- Authentication events — usernames, computers, success, type, time
- Host-sensor data — network events, processes, memory usage, time, duration, lock/unlock
- Physical — building access control, sensors, IoT

High volume, high frequency data which require thinning (screening, triage) and parallel processing:

- Hadoop — MapReduce and Spark
- Algorithms which scale well or can run in the stream

Methodological Approaches

Analyses can be performed at different levels of resolution:

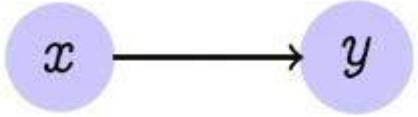
- Entire network analysis - graph theory, spectral decompositions, community detection, clustering. Also high level traffic summaries for network oversight.
- Node-based models - building statistical models of the processes run by a host, its network connectivity, pattern of life.
- Edge-based models - detecting beacons to specific IP addresses, temporal dependence on neighbouring edges, typical packet sizes.

All of these viewpoints, and others, can yield cyber-security analytics.
There will not be one statistical test that answers all questions.
Power must be obtained by combining several possibly weak indicators into a strong overall signal.

This talk will provide examples of each class of analysis, summarising published work and current research.

Edges

An edge between two hosts or IP addresses ("nodes") is the basic fundamental unit for behavioural modelling (perhaps counter-intuitively, rather than nodes).



Edges should typically be directed, for example a client x connecting to a server y. The presence of the edge will be taken to mean that x has *ever been observed* to initiate a connection to y.

In NetFlow, some edges will carry only machine-driven, automated connections; some will be entirely human-driven; others will have a mixture of both.

Distinguishing Automated from Human Traffic

Automated edges typically carry "super-human" traffic volumes. But not always, and so we need more sophisticated filters.

Automated events are often highly periodic, corresponding to scheduled beacons pushing refreshes and updates, or "keep-alives".

(Heard, Rubin-Delanchy, and Lawson, 2014) We can scan for periodicities in event times by inspecting the periodogram after time T,

$$\widehat{S}(f) = \frac{1}{T}\left|\sum_{t=1}^{T}\{\mathrm{d}N(t) - N(T)/T\}e^{-2\pi\imath ft}\right|^2.$$

which can be efficiently calculated at each of the Fourier frequencies $f_k = k/T$, $k = 1, \ldots, \lfloor T/2 \rfloor$, via the Fast Fourier Transform.

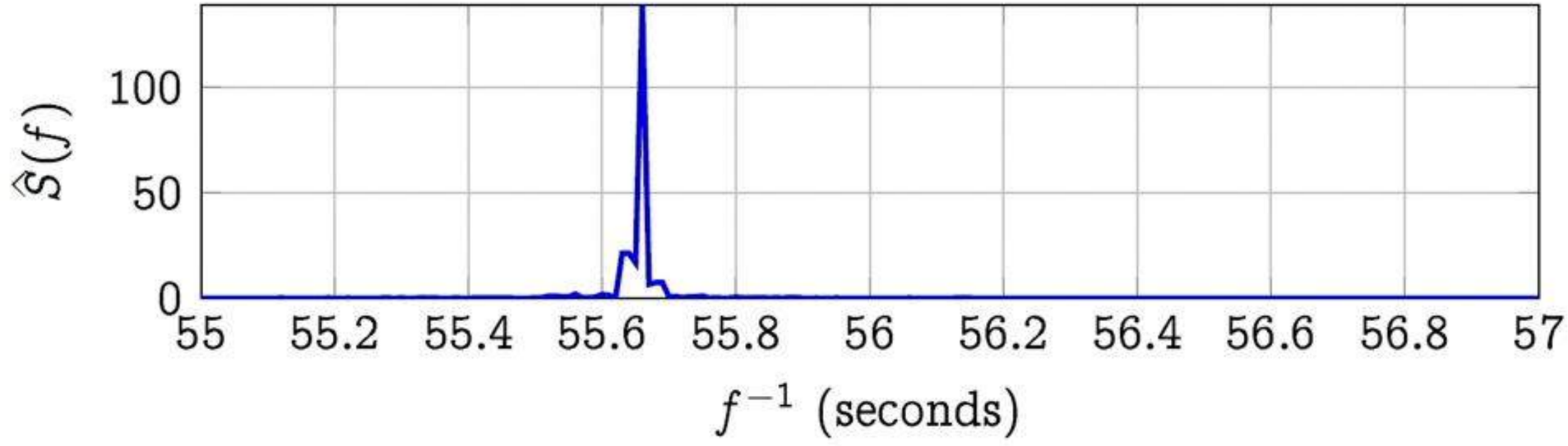
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:



Distinguishing Automated from Human Traffic

Automated edges typically carry "super-human" traffic volumes. But not always, and so we need more sophisticated filters.

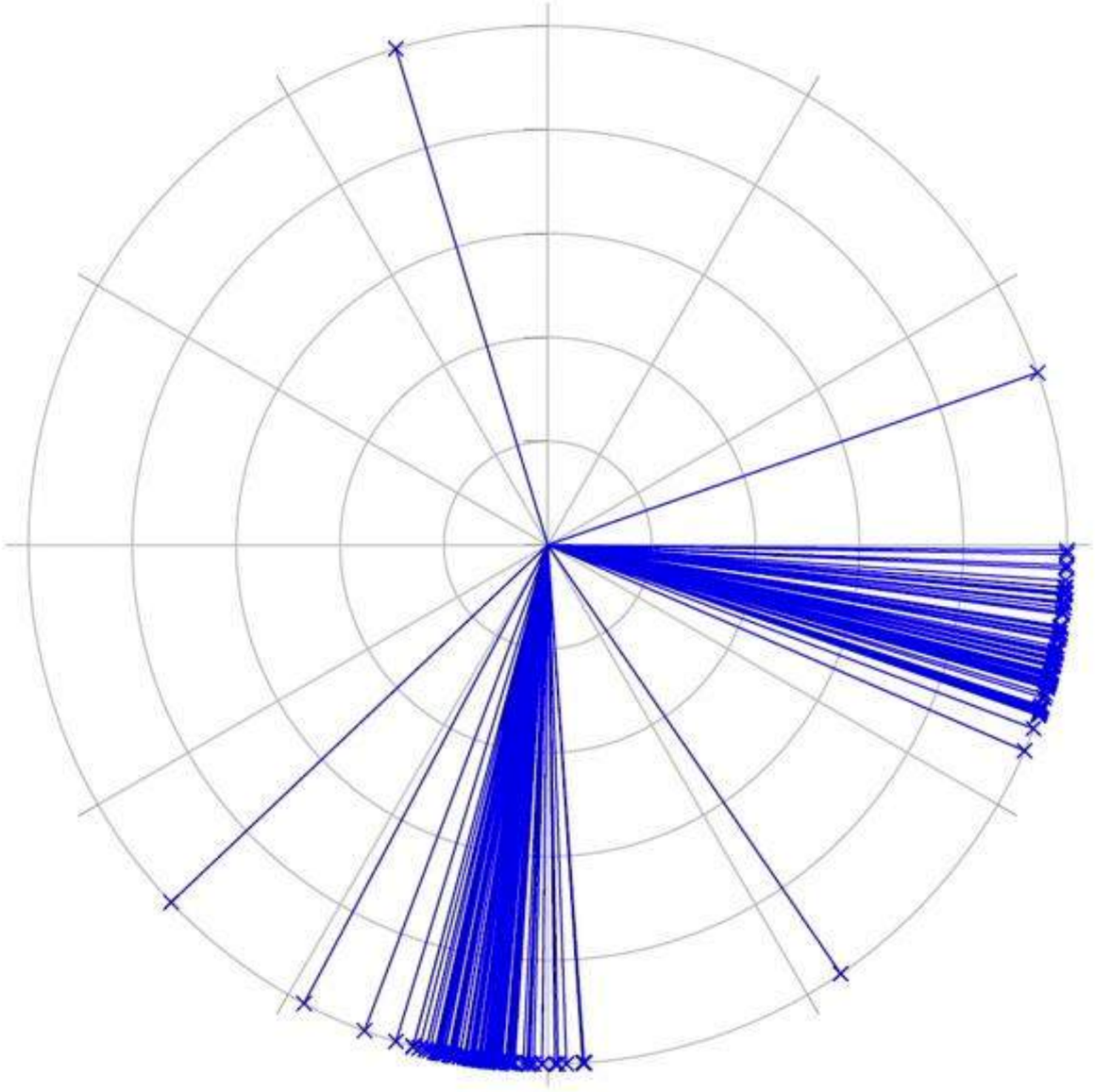
Automated events are often highly periodic, corresponding to scheduled beacons pushing refreshes and updates, or "keep-alives".

(Heard, Rubin-Delanchy, and Lawson, 2014) We can scan for periodicities in event times by inspecting the periodogram after time T,

$$\widehat{S}(f) = \frac{1}{T}\left|\sum_{t=1}^{T}\{\mathrm{d}N(t) - N(T)/T\}e^{-2\pi\imath ft}\right|^2.$$

which can be efficiently calculated at each of the Fourier frequencies $f_k = k/T$, $k = 1, \ldots, \lfloor T/2 \rfloor$, via the Fast Fourier Transform.

Dropbox events projected onto a 55.65-second clock as unit vectors



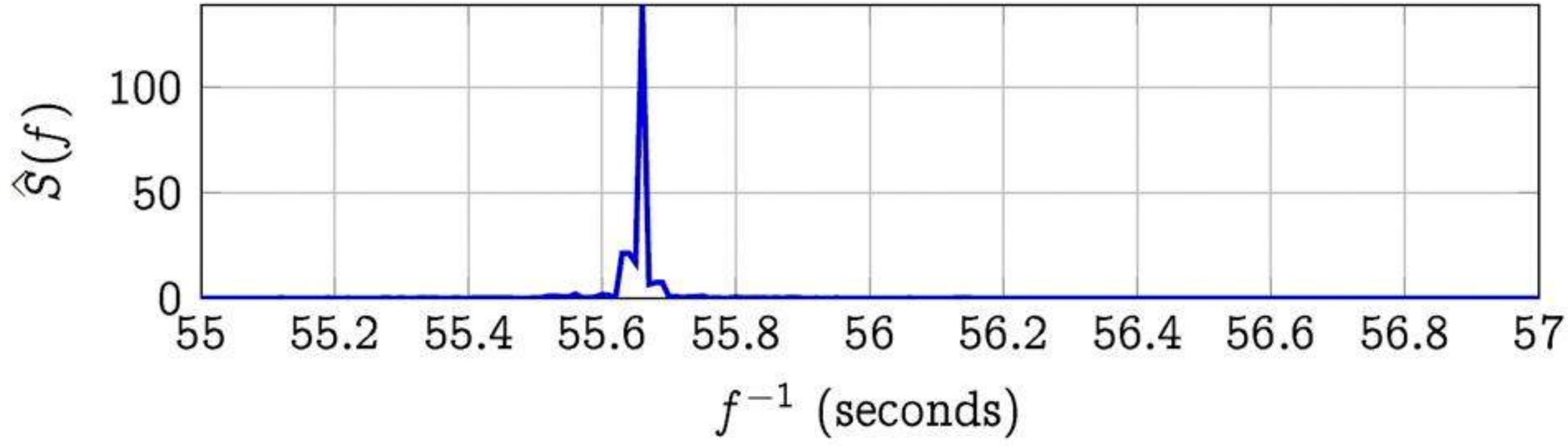
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:



Distinguishing Automated from Human Traffic

Automated edges typically carry "super-human" traffic volumes. But not always, and so we need more sophisticated filters.

Automated events are often highly periodic, corresponding to scheduled beacons pushing refreshes and updates, or "keep-alives".

(Heard, Rubin-Delanchy, and Lawson, 2014) We can scan for periodicities in event times by inspecting the periodogram after time T,

$$\widehat{S}(f) = \frac{1}{T}\left|\sum_{t=1}^{T}\{\mathrm{d}N(t) - N(T)/T\}e^{-2\pi\imath ft}\right|^2.$$

which can be efficiently calculated at each of the Fourier frequencies $f_k = k/T$, $k = 1, \ldots, \lfloor T/2 \rfloor$, via the Fast Fourier Transform.

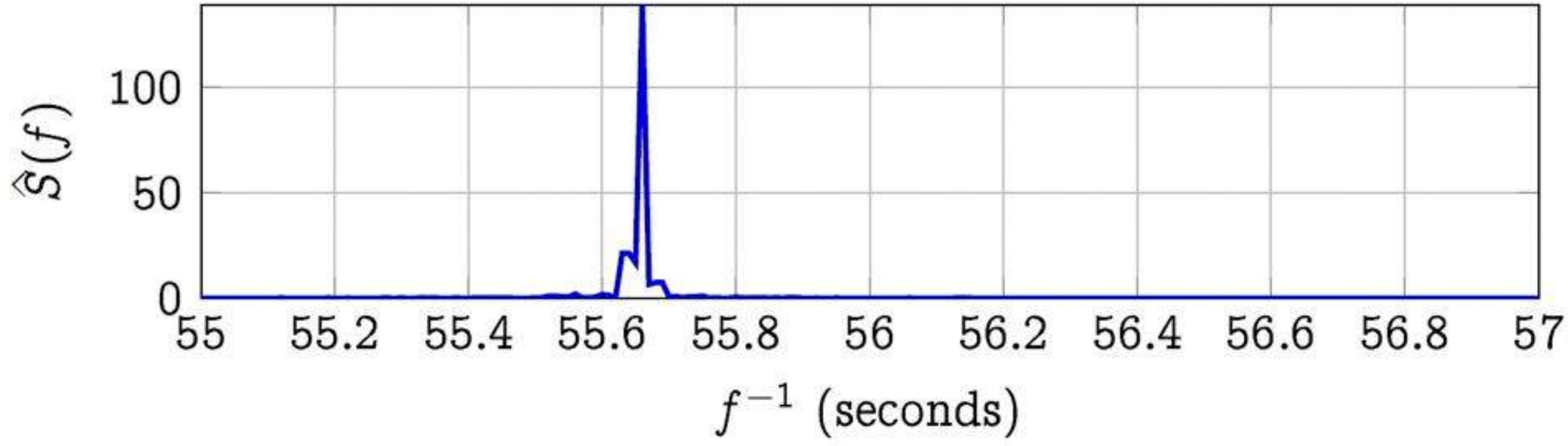
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

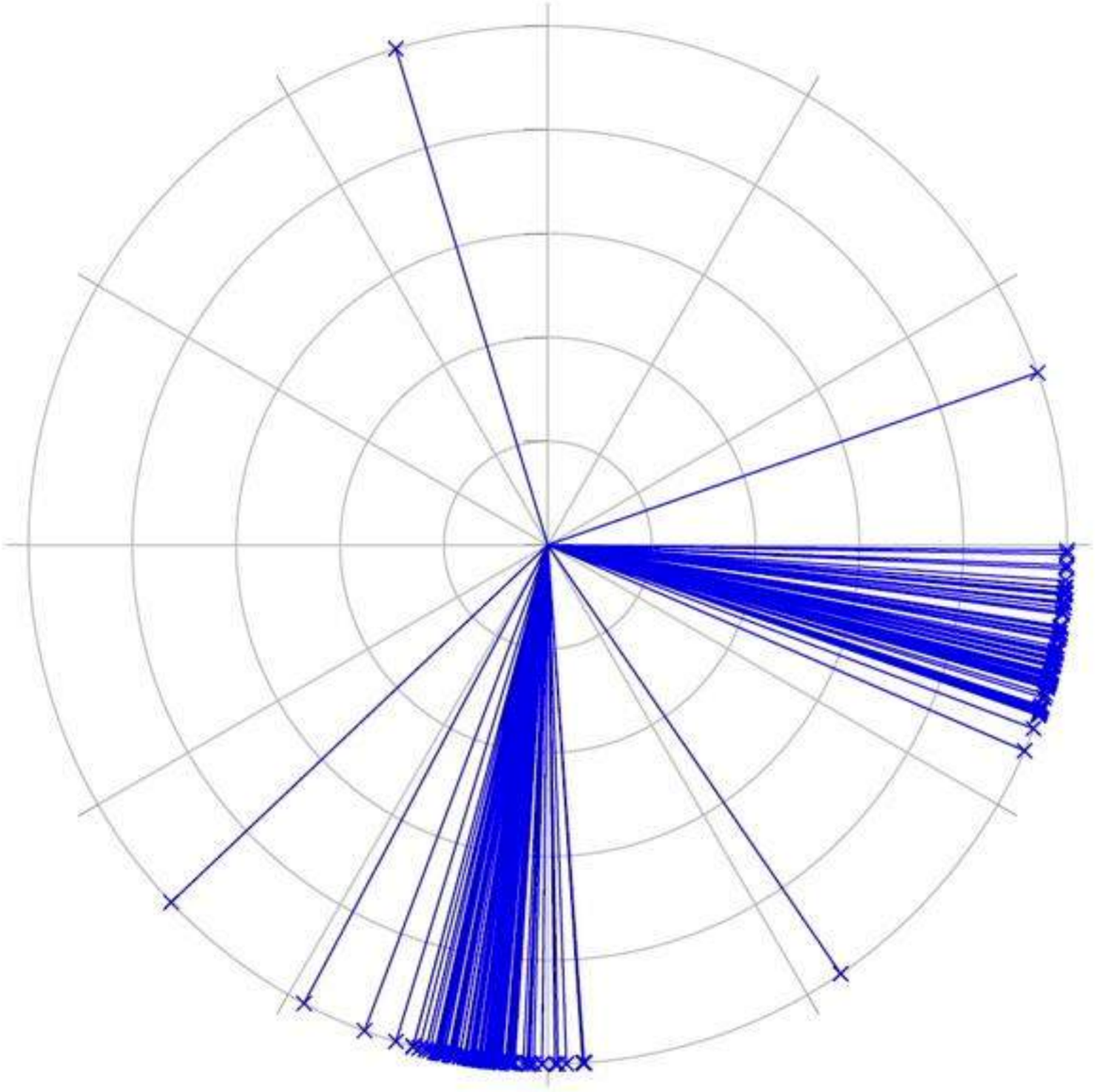
Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

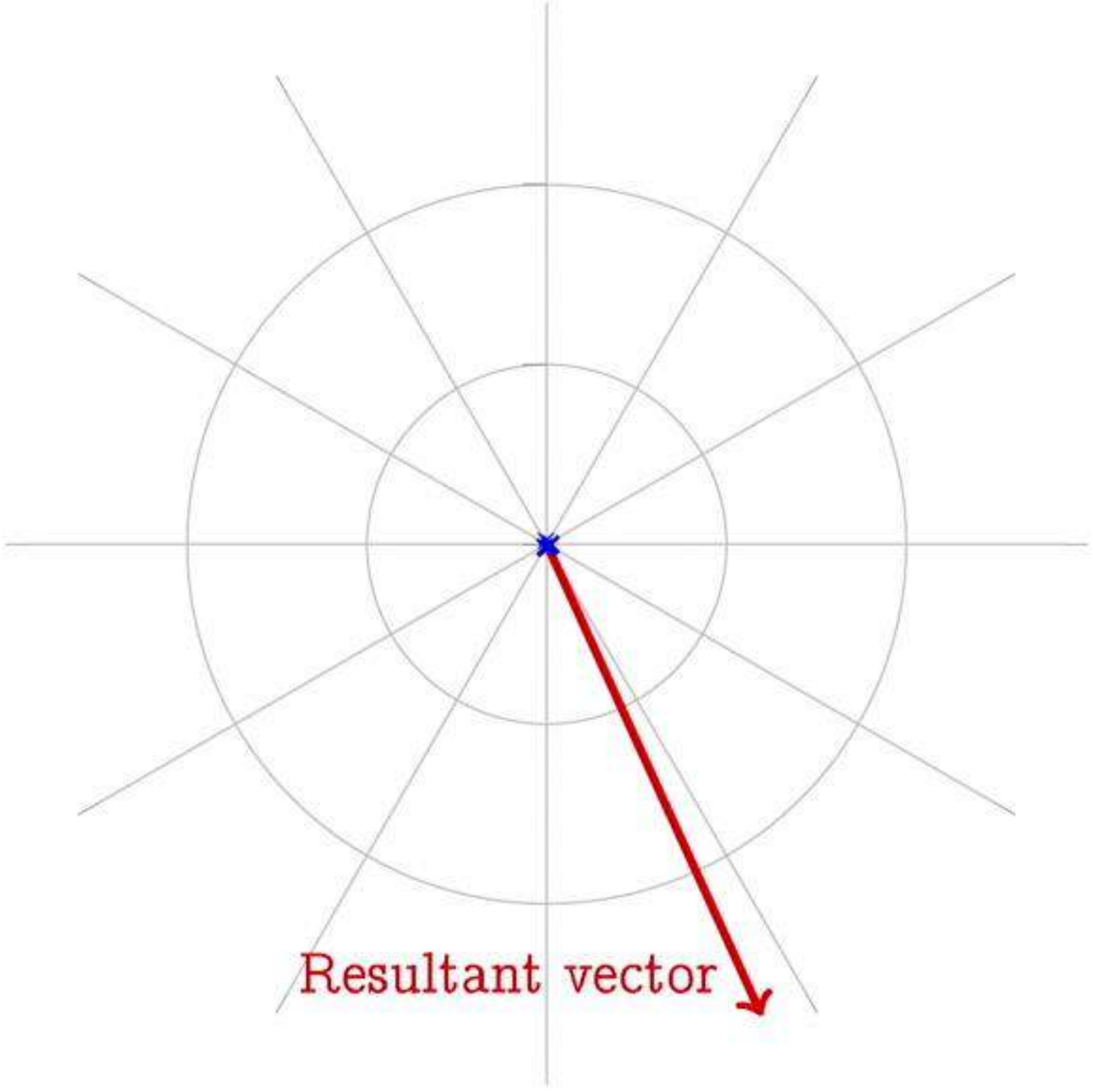
Example: My computer to Dropbox:



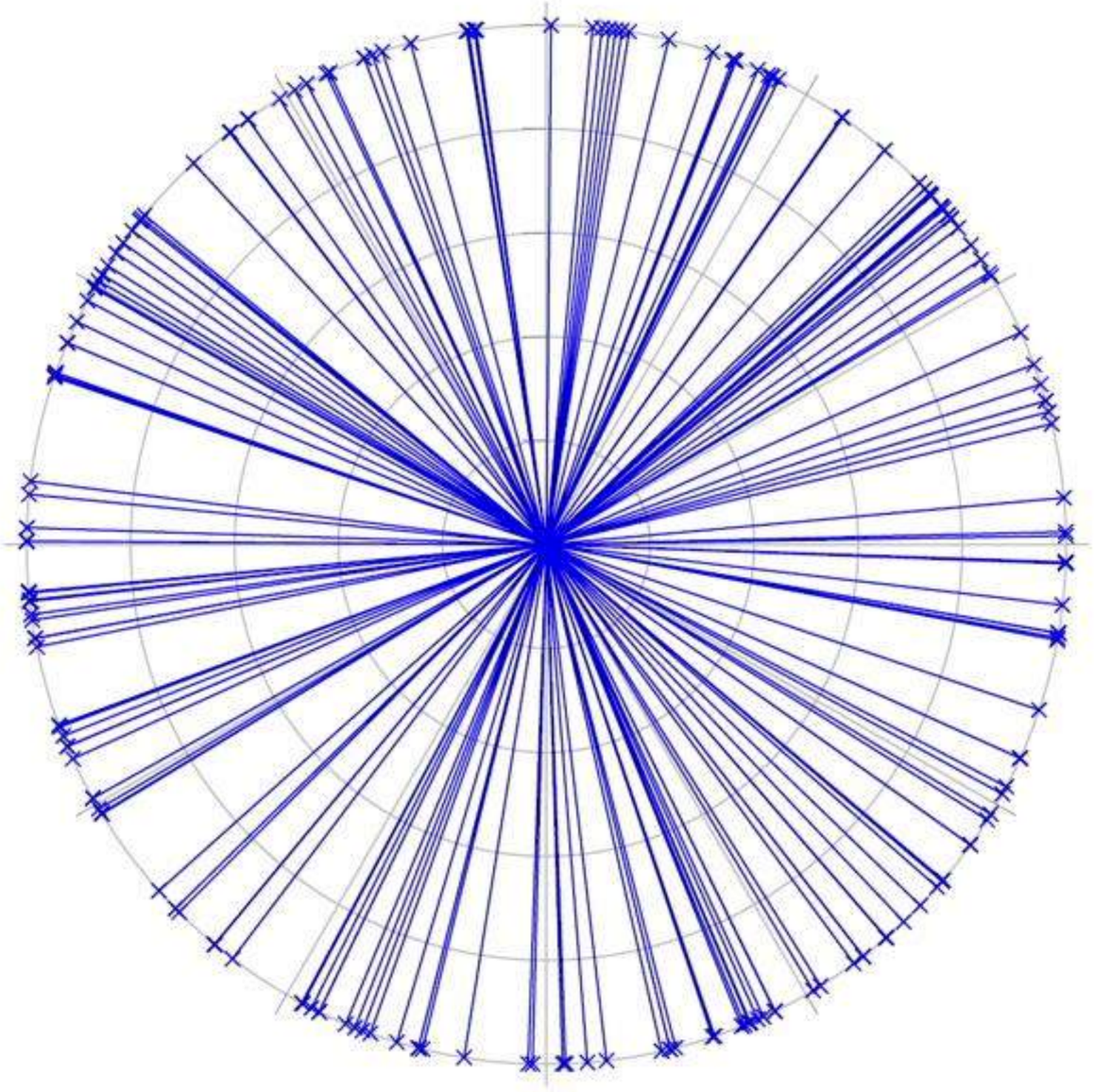
Dropbox events projected onto a 55.65-second clock as unit vectors



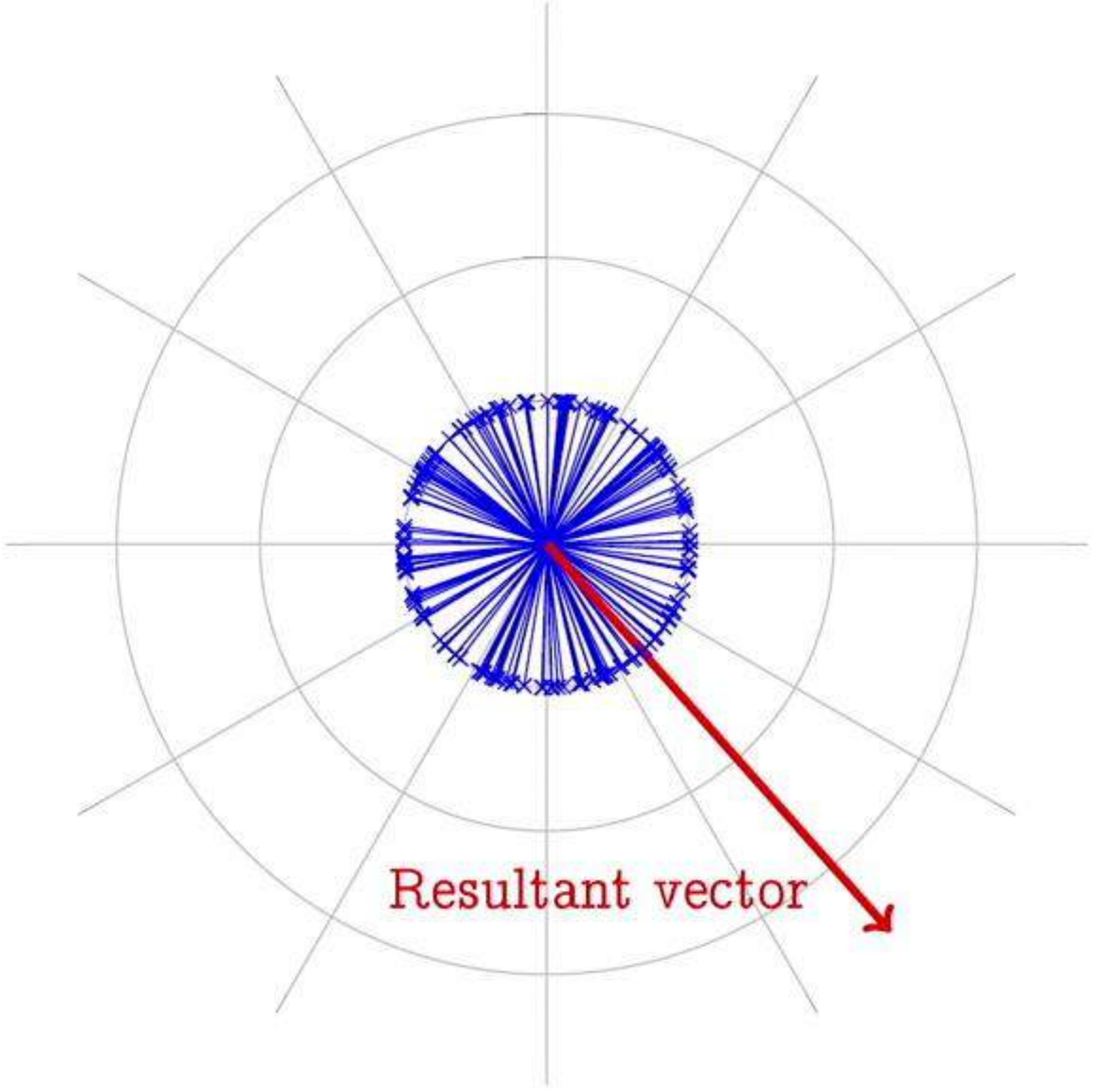
Dropbox events projected onto a 55.65-second clock as unit vectors



Dropbox events projected onto a 43-second clock as unit vectors



Dropbox events projected onto a 43-second clock as unit vectors



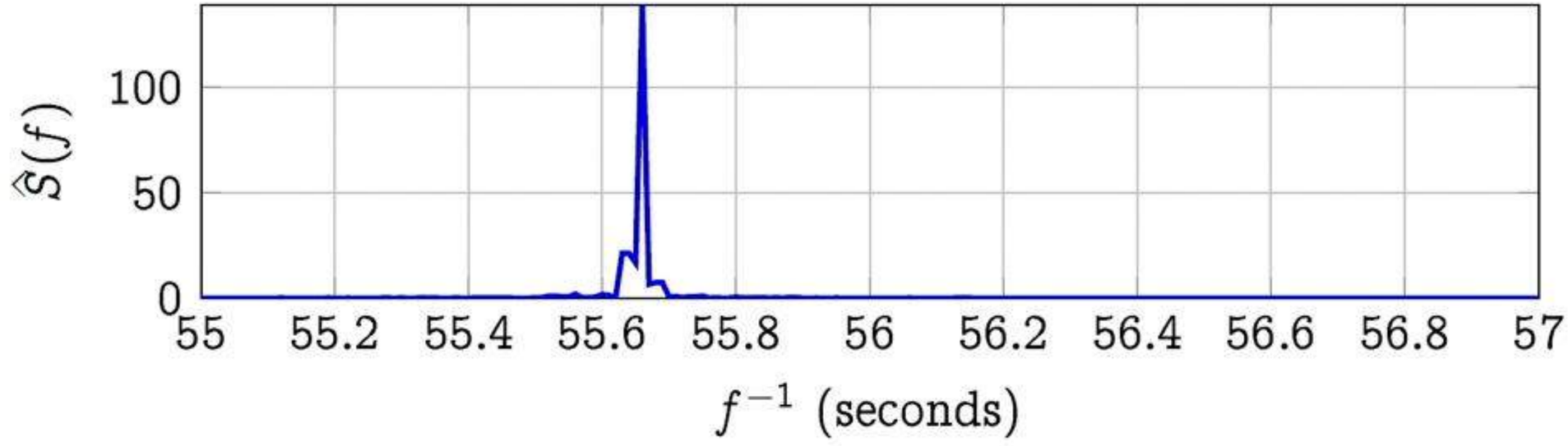
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

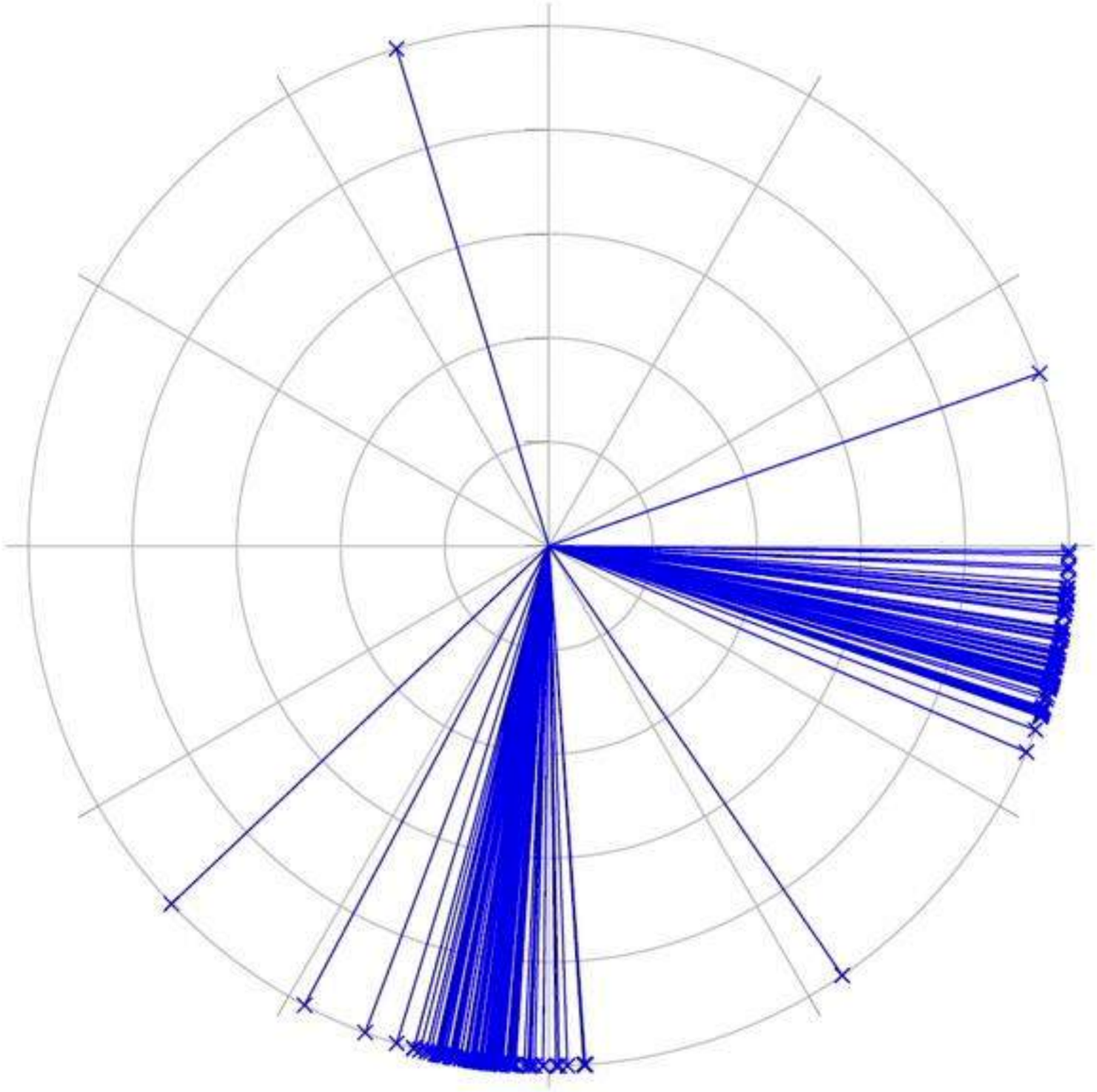
Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:

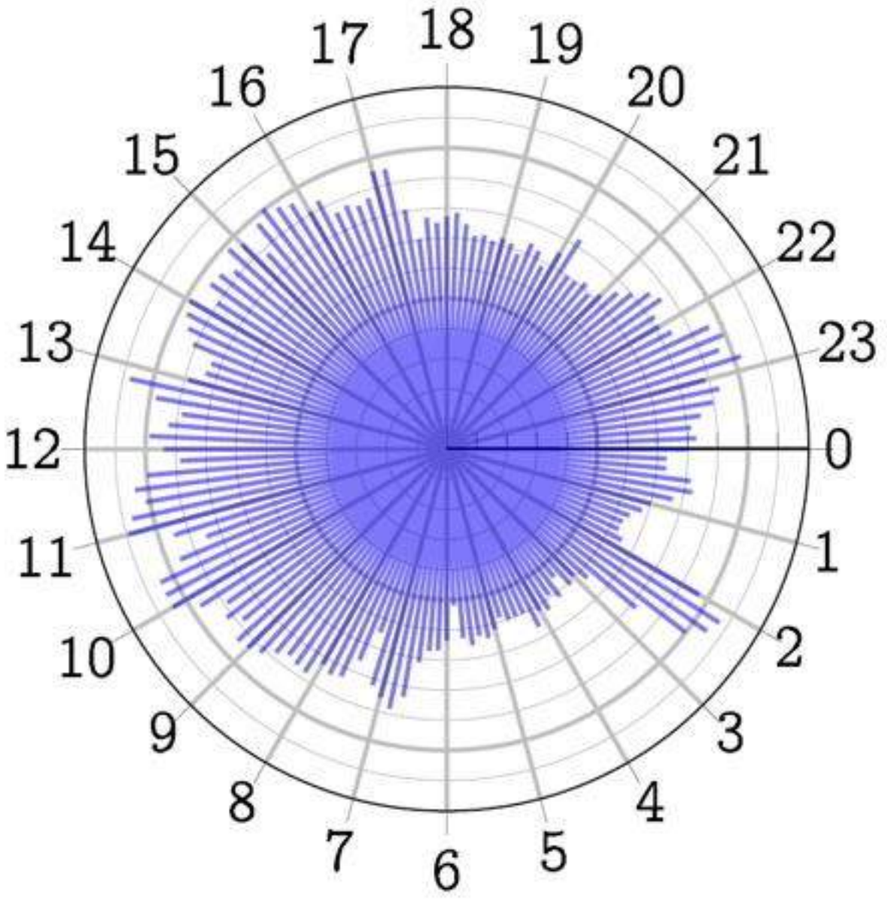


Dropbox events projected onto a 55.65-second clock as unit vectors

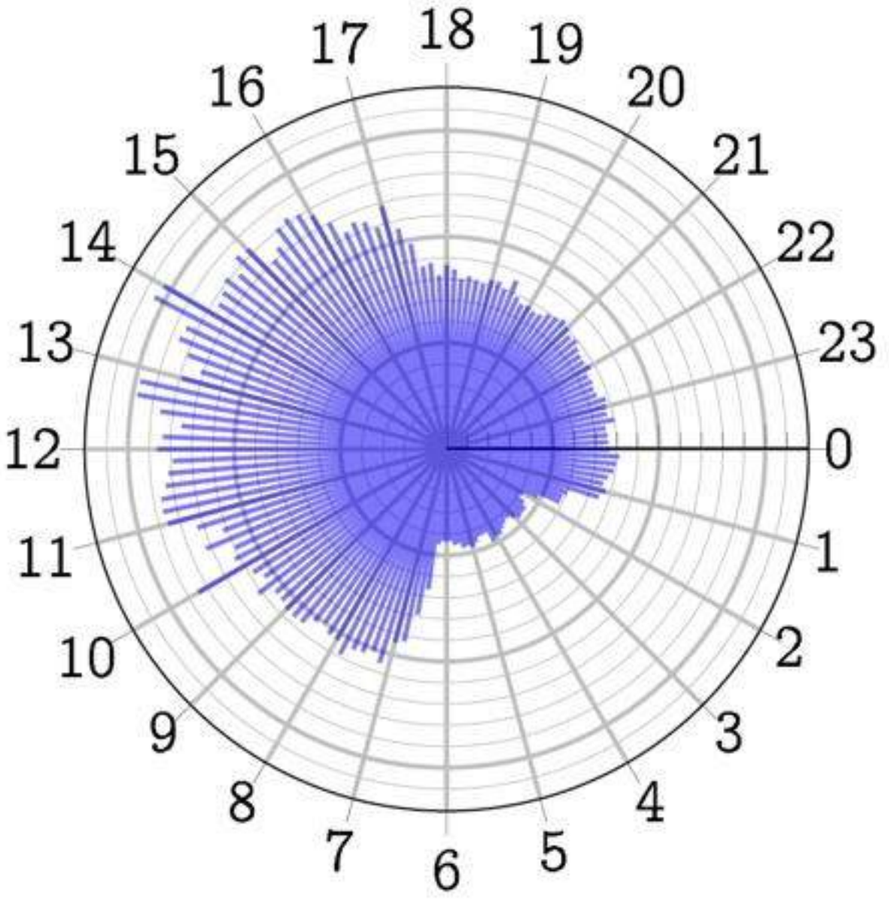


NetFlow events on my computer, before and after filtering

Before filtering



After filtering



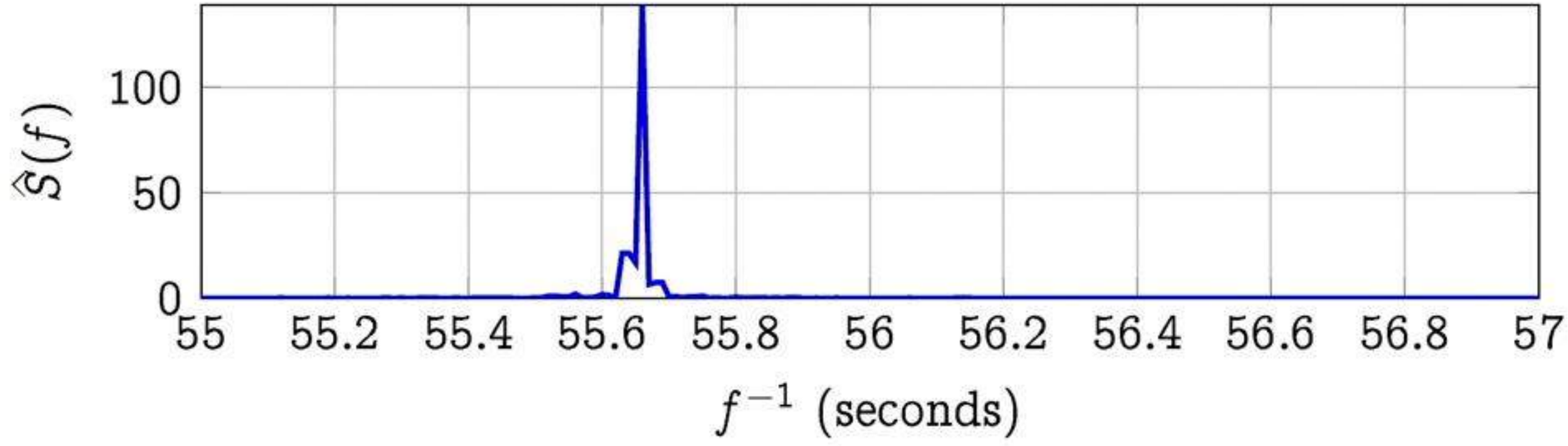
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

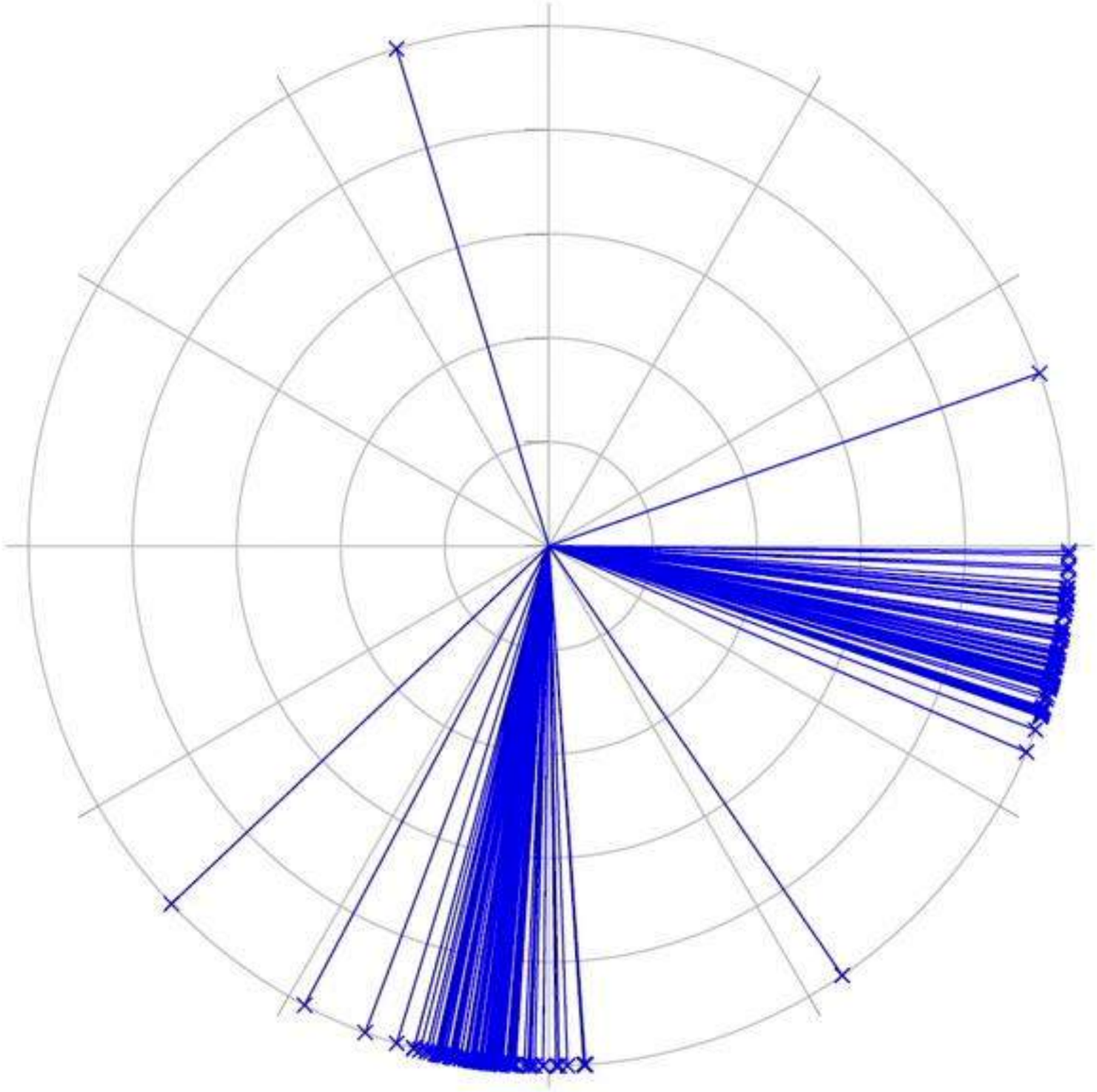
Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:

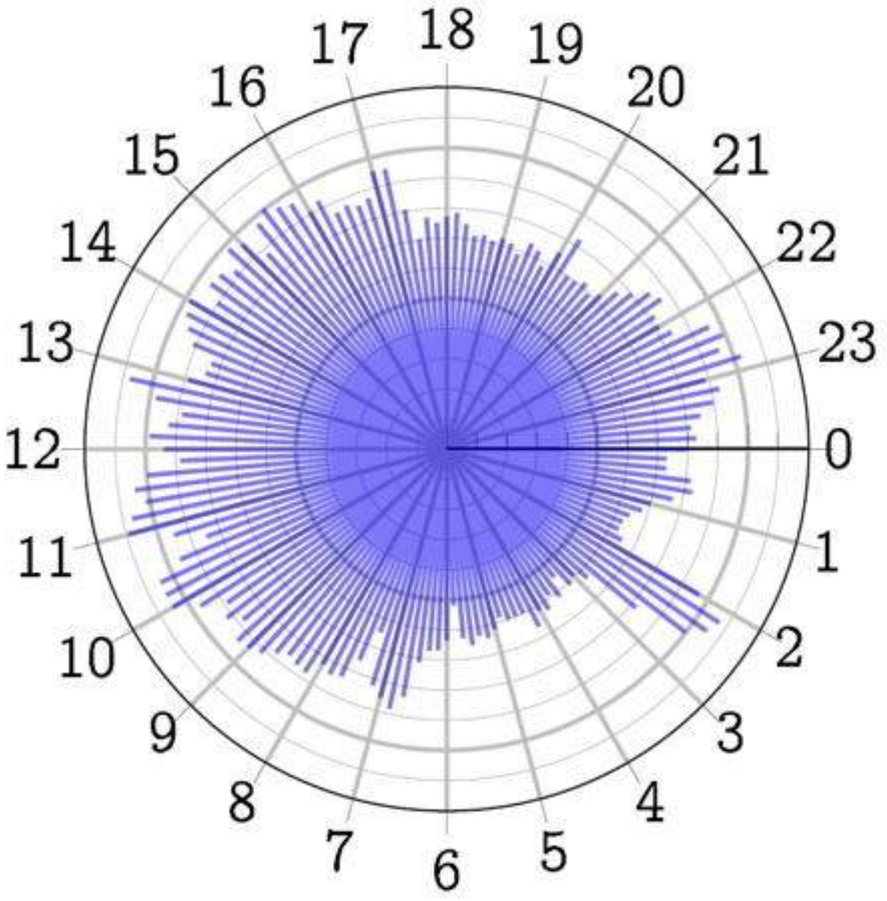


Dropbox events projected onto a 55.65-second clock as unit vectors

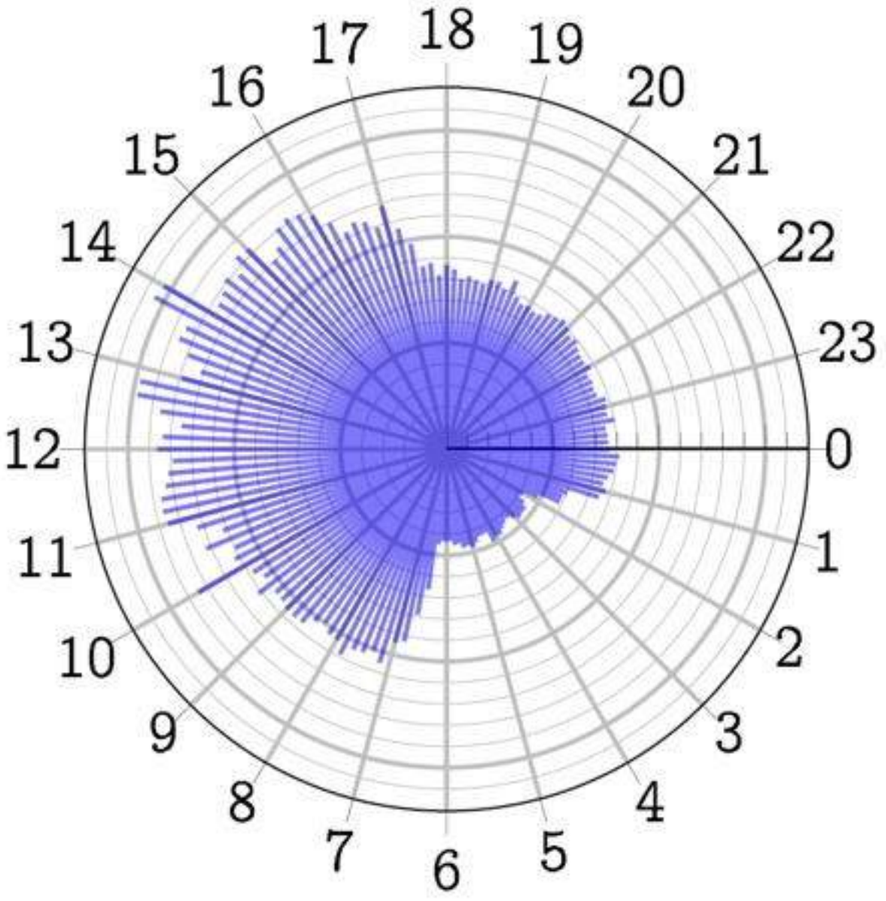


NetFlow events on my computer, before and after filtering

Before filtering



After filtering

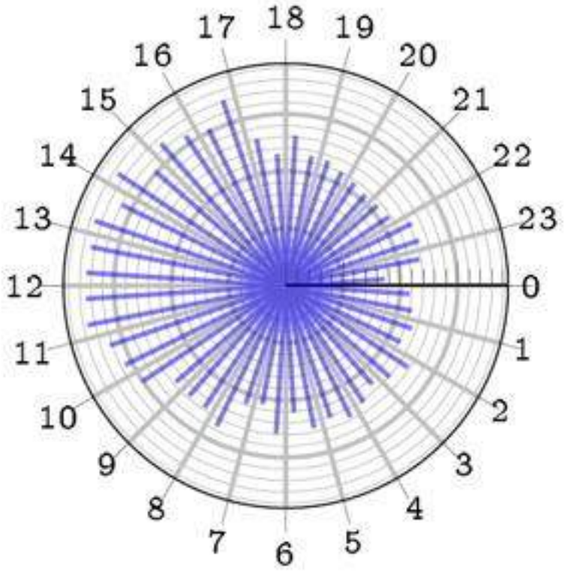


Classifying events on edges (with F. Sanna Passino)

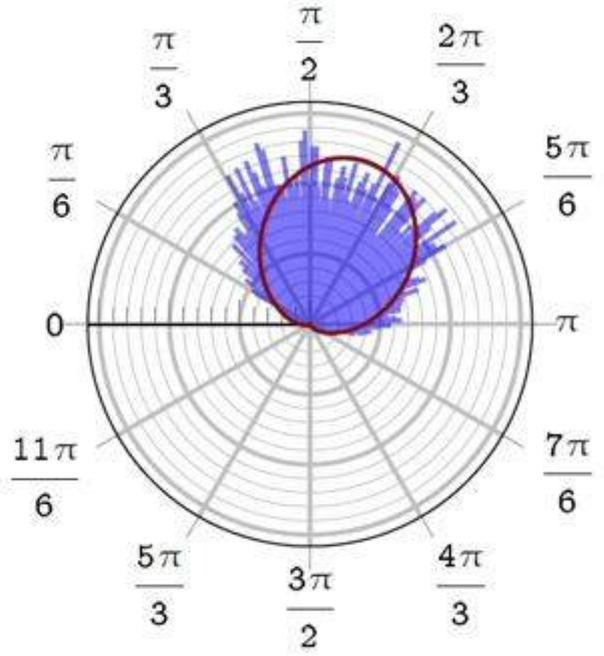
- The Fourier analysis approach quickly classifies edges containing automated traffic
  - If there is a mixture, the automated traffic will normally be higher volume and dominate
- We might further want to classify every single NetFlow event as being human or automated. We can examine:
  - How well synchronised an event is with the periodic phase for that edge
  - Whether the time of day is consistent with human behaviour on that edge/node

- We fit a Bayesian mixture model to learn the human and automated components:
  - f_A: Wrapped normal density for learning phase (circular mean) and variance of beacons
  - f_H: Piecewise constant density to consistently estimate human event distribution
- Events are probabilistically attributed to either f_A or f_H
- Example: 13.107.42.11 (`outlook.com`), polling at $\approx 8s$ intervals

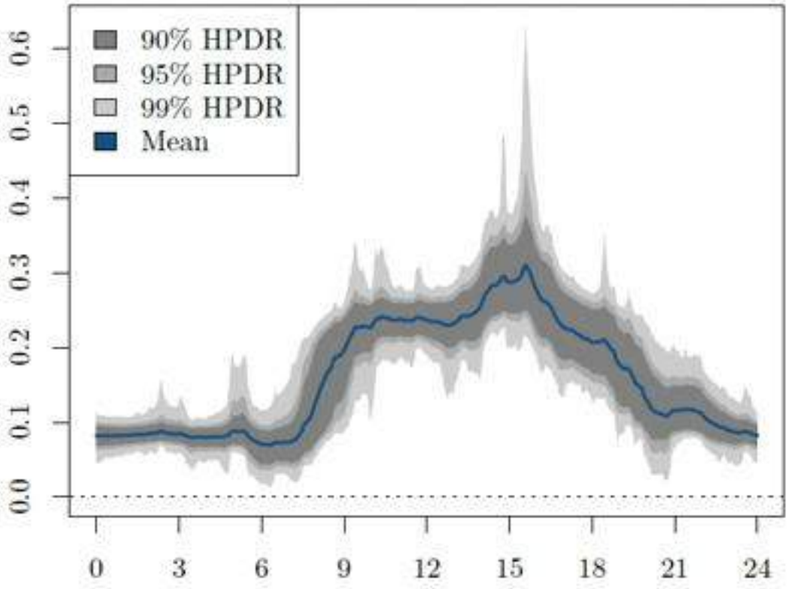
Event time distribution



Wrapped normal



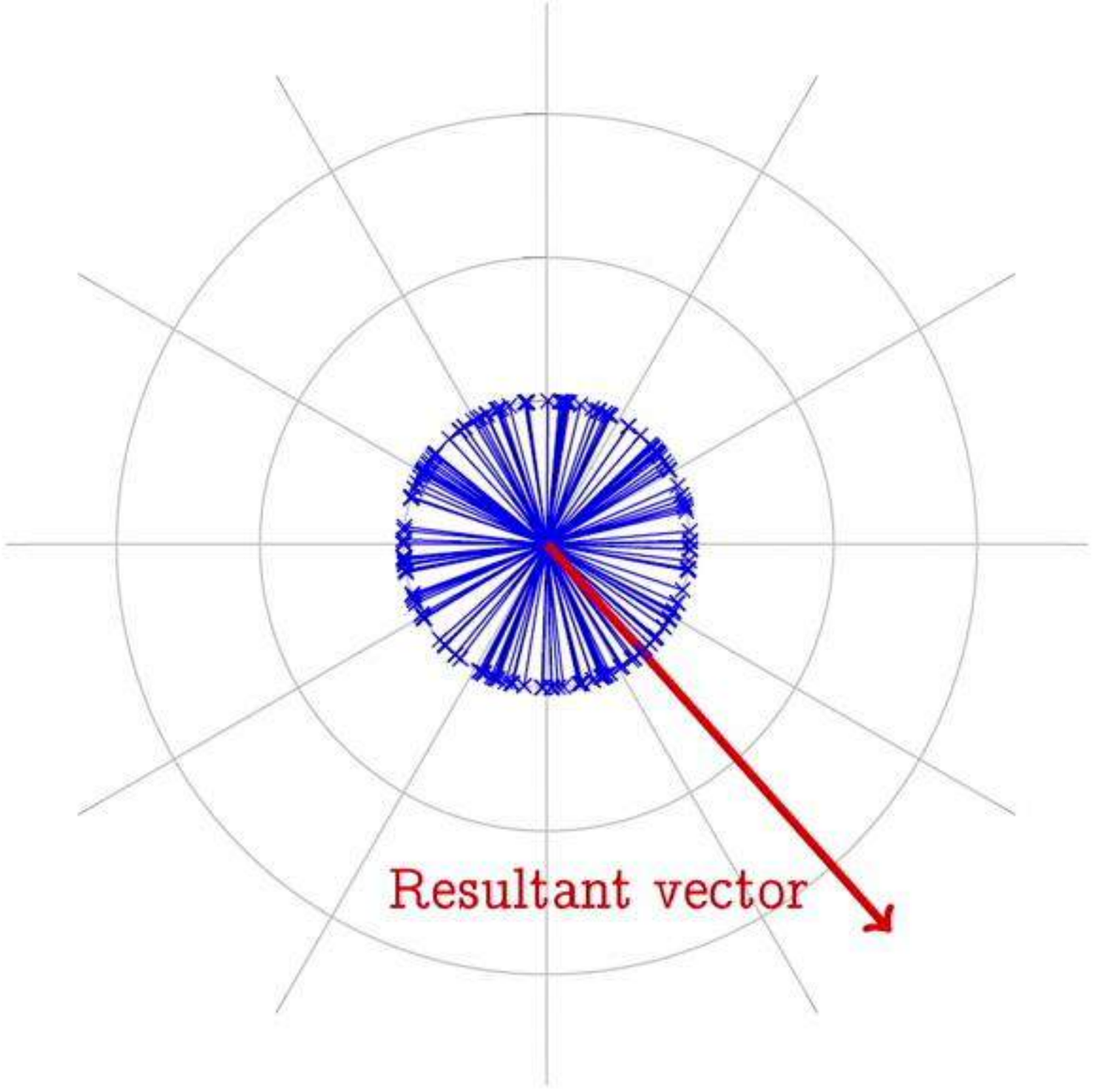
(Averaged) Step function density



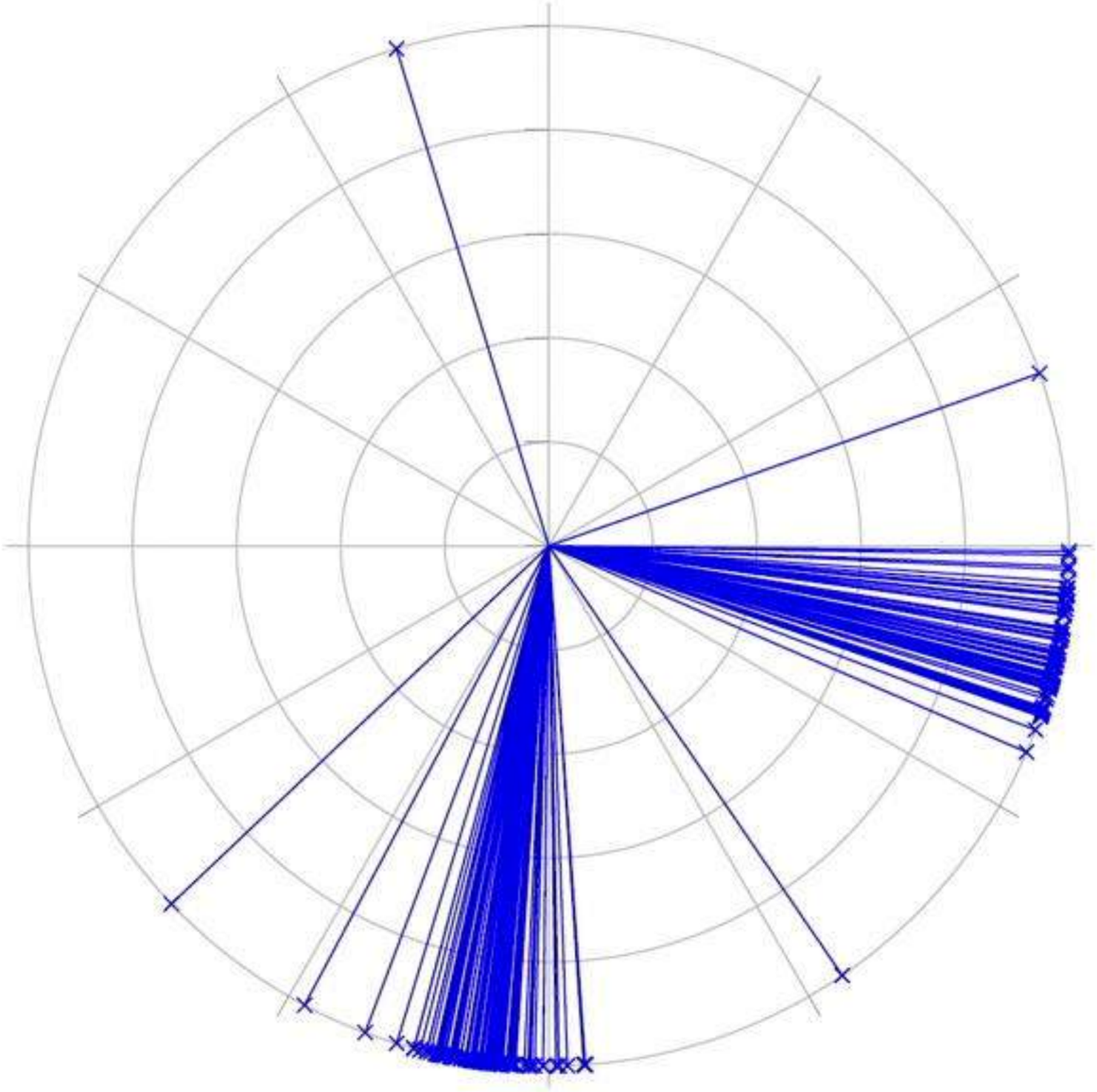
Classifying events on edges (with F. Sanna Passino)

- The Fourier analysis approach quickly classifies edges containing automated traffic
  - If there is a mixture, the automated traffic will normally be higher volume and dominate
- We might further want to classify every single NetFlow event as being human or automated. We can examine:
  - How well synchronised an event is with the periodic phase for that edge
  - Whether the time of day is consistent with human behaviour on that edge/node

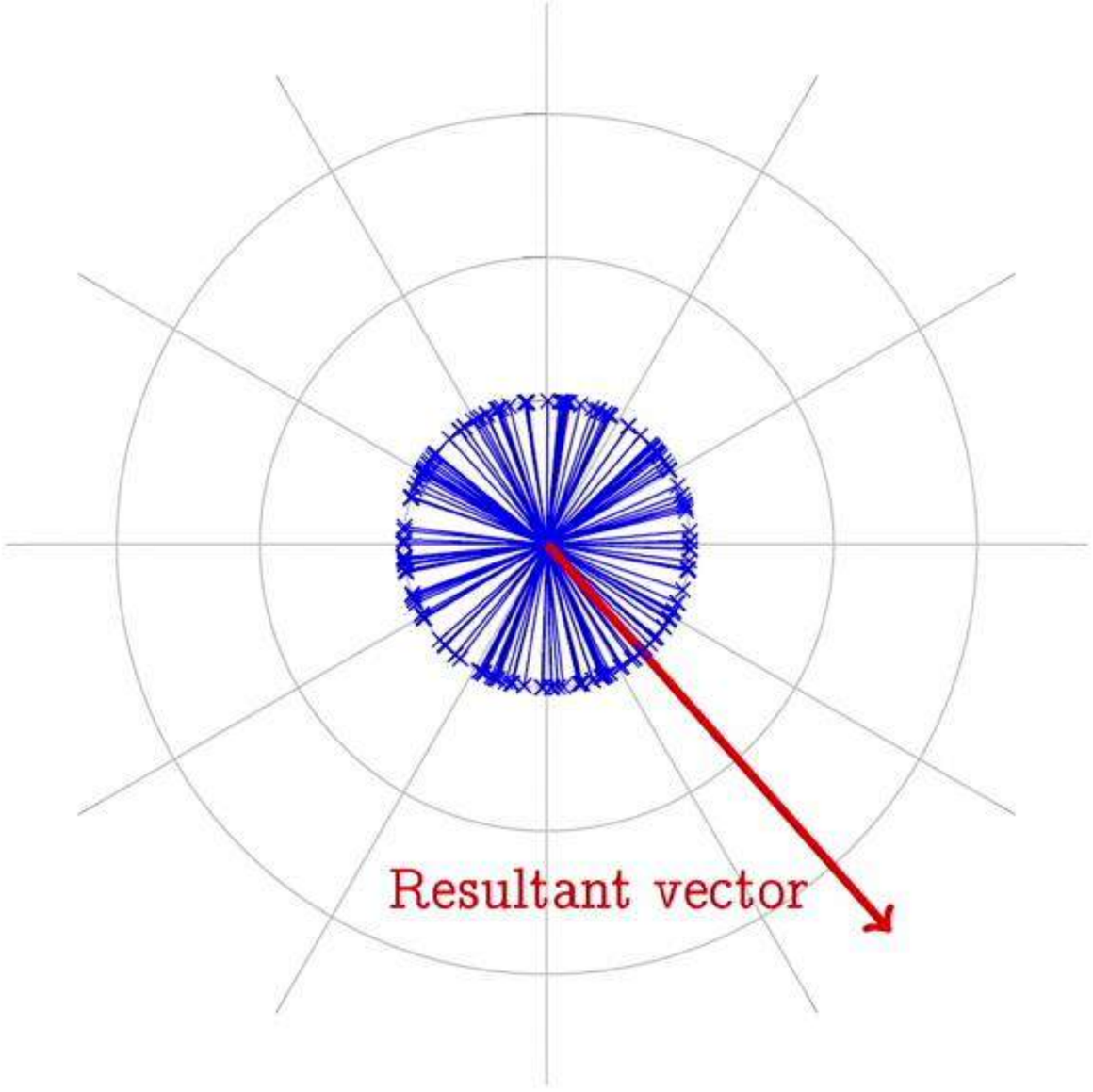
Dropbox events projected onto a 43-second clock as unit vectors



Dropbox events projected onto a 55.65-second clock as unit vectors



Dropbox events projected onto a 43-second clock as unit vectors

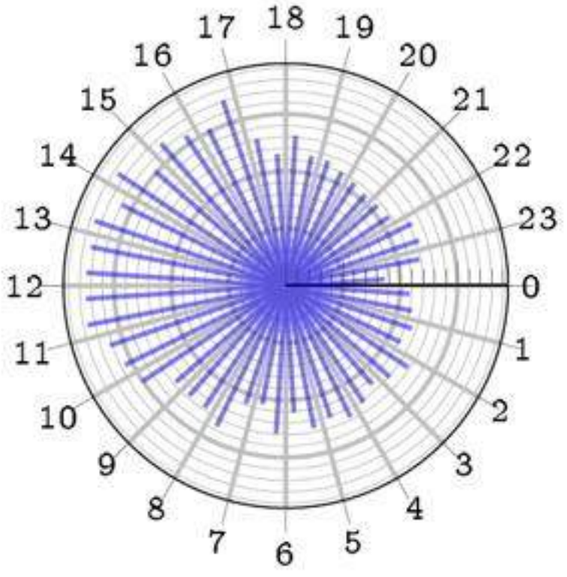


Classifying events on edges (with F. Sanna Passino)

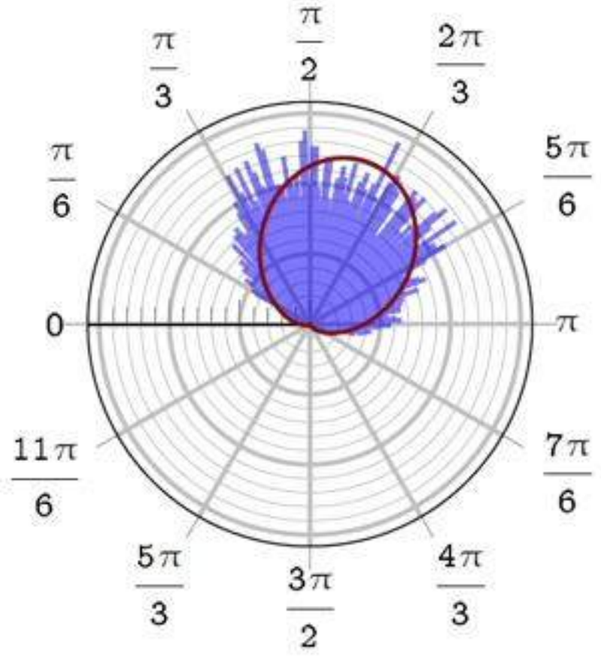
- The Fourier analysis approach quickly classifies edges containing automated traffic
  - If there is a mixture, the automated traffic will normally be higher volume and dominate
- We might further want to classify every single NetFlow event as being human or automated. We can examine:
  - How well synchronised an event is with the periodic phase for that edge
  - Whether the time of day is consistent with human behaviour on that edge/node

- We fit a Bayesian mixture model to learn the human and automated components:
  - f_A: Wrapped normal density for learning phase (circular mean) and variance of beacons
  - f_H: Piecewise constant density to consistently estimate human event distribution
- Events are probabilistically attributed to either f_A or f_H
- Example: 13.107.42.11 (`outlook.com`), polling at $\approx 8s$ intervals

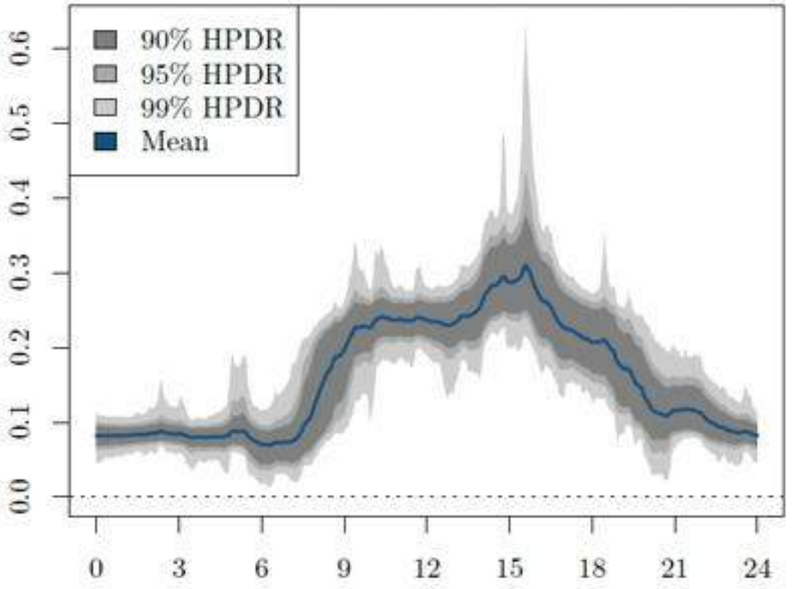
Event time distribution



Wrapped normal



(Averaged) Step function density

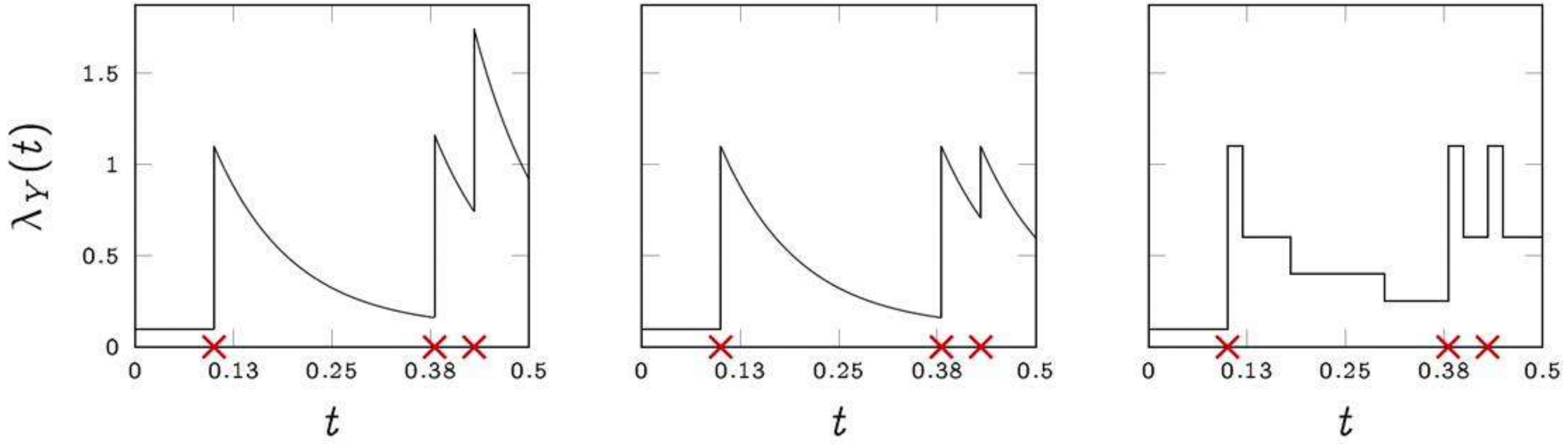


Modelling bursts of events (with M. Price-Williams)

Even human-generated network connections do not arrive as an (inhomogeneous) Poisson process. They occur in bursts, on the same edge and between edges.

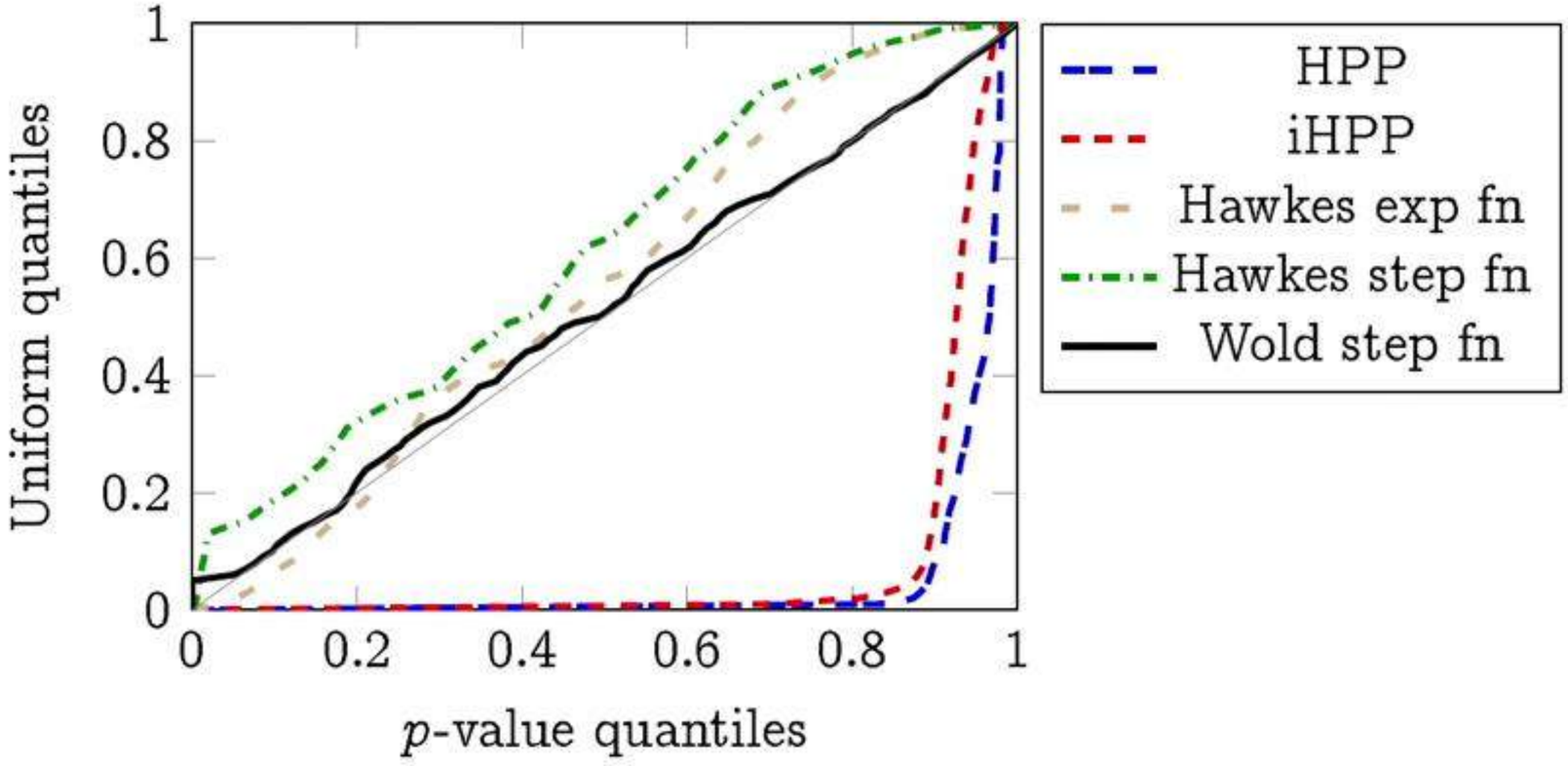
We model arrivals of event times $y_1, y_2, \ldots$ as a Wold process with self-exciting conditional intensity

$$\lambda_Y(t) = \lambda + \sum_{j=1}^{\ell} \lambda_j \mathbb{I}_{[\tau_{j-1},\tau_j)}(t - y_{Y(t)})$$



Advantages:

- Flexible changepoint model for excitation function provides consistent estimator
- Capturing *burstiness* negates the need to model seasonality, which has complex variations day-on-day



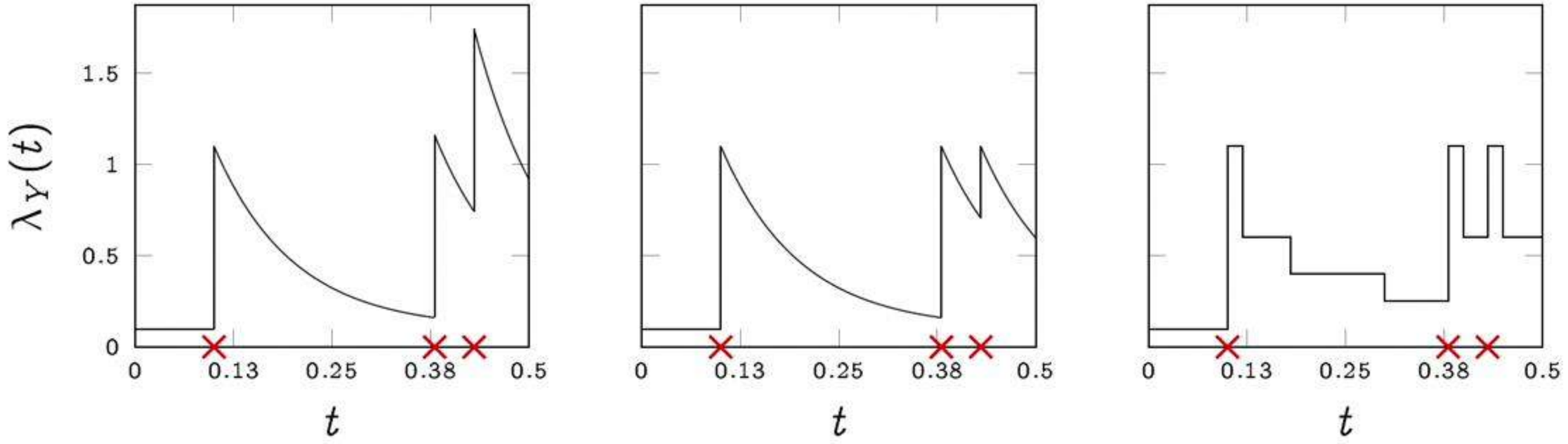
The idea has been extended to node-based modelling of all outgoing edges, such that events on one edge from a node can trigger events on its other edges

Modelling bursts of events (with M. Price-Williams)

Even human-generated network connections do not arrive as an (inhomogeneous) Poisson process. They occur in bursts, on the same edge and between edges.

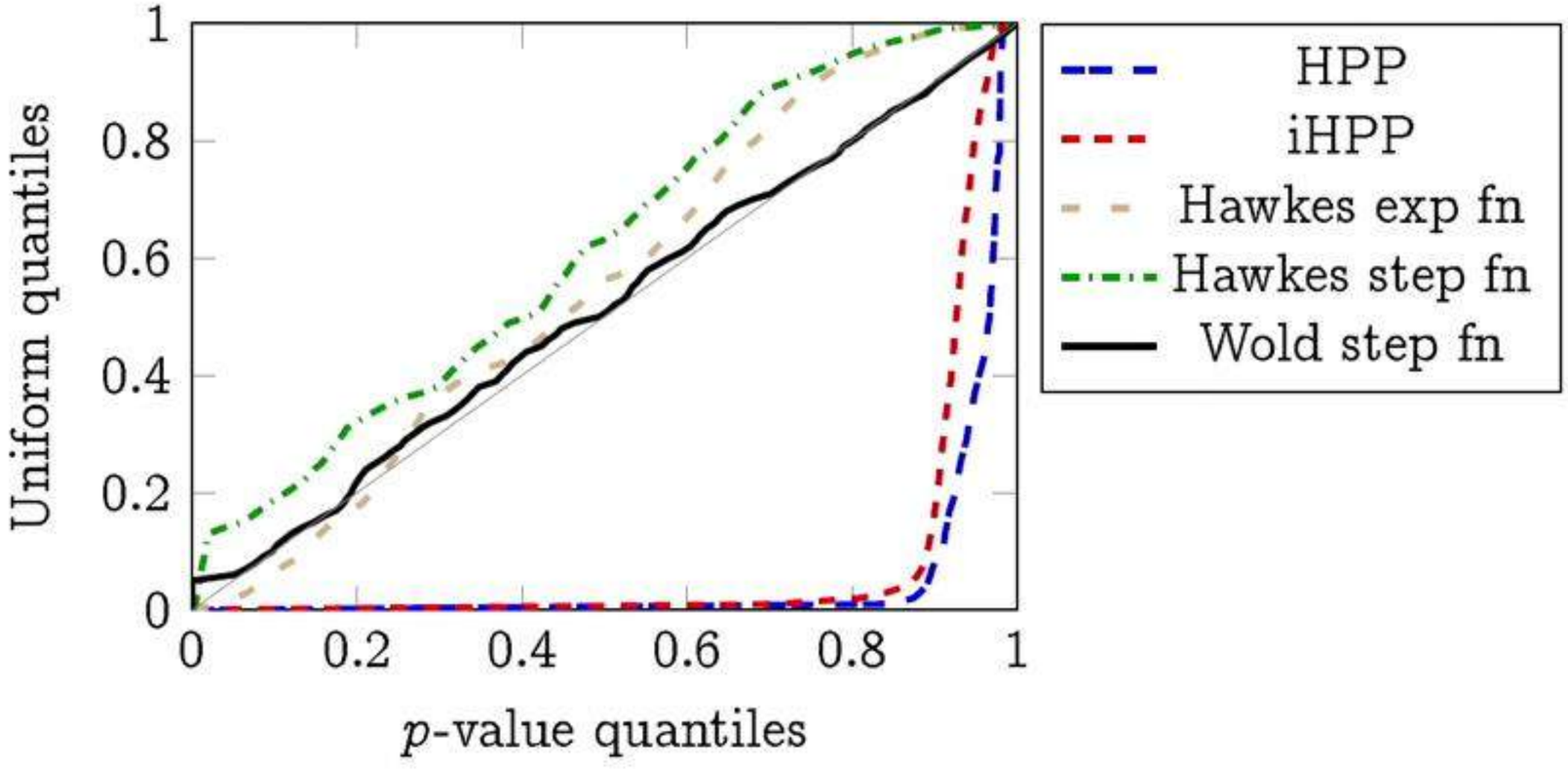
We model arrivals of event times $y_1, y_2, \ldots$ as a Wold process with self-exciting conditional intensity

$$\lambda_Y(t) = \lambda + \sum_{j=1}^{\ell} \lambda_j \mathbb{I}_{[\tau_{j-1},\tau_j)}(t - y_{Y(t)})$$



Advantages:

- Flexible changepoint model for excitation function provides consistent estimator
- Capturing *burstiness* negates the need to model seasonality, which has complex variations day-on-day



The idea has been extended to node-based modelling of all outgoing edges, such that events on one edge from a node can trigger events on its other edges

Nodes

Monitoring node connectivity

We can monitor the sequence of destinations a node connects to and look for bursts of unusual connectivity.

Each connection event is scored, and surprise is aggregated using control charts or p-value combination techniques.

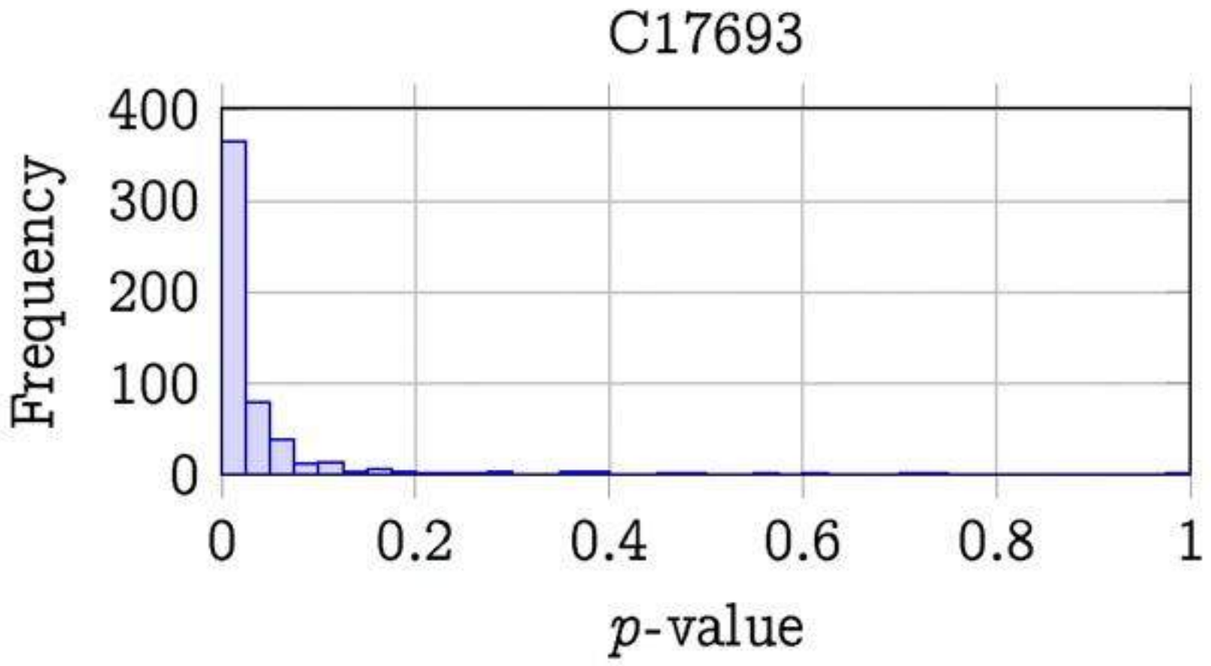
(Heard and Rubin-Delanchy, 2016) Server modelling: The sequence of clients $x_1, x_2, \ldots$ connecting to a server y were modelled as a multinomial, with an unbounded number of categories and a Dirichlet process prior (with some base measure αF_0) on the category probabilities.

The p-value score for observation x_{n+1}:

$$p_{n+1} = \sum_{x \in V : \alpha_x^* \leq \alpha_{x_{n+1}}^*} \frac{\alpha_x^*}{\alpha^*}$$

- $\alpha_x^* = \alpha F_0(x) + \sum_{i=1}^{n} \mathbb{I}_x(x_i)$ and $\alpha^* = \alpha + n$.

Edge scoring: For each unique client x connecting to server y, the minimum p-value was obtained from the sequence and beta-transformed to a p-value score for the edge (x, y)

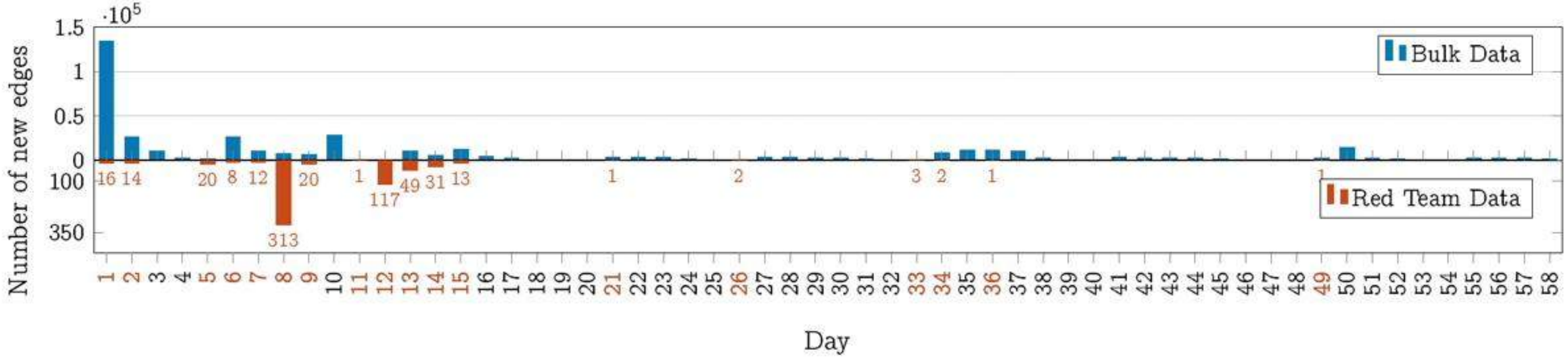


Node scoring: For each client x, the p-values across all of its edges were combined using Fisher's method.

- Red team attacks in Los Alamos enterprise network were found in this way

New edges (with S. Metelli)

The red team attack within public Los Alamos enterprise network data contained a spate of new edges being formed between hosts which had not previously connected.



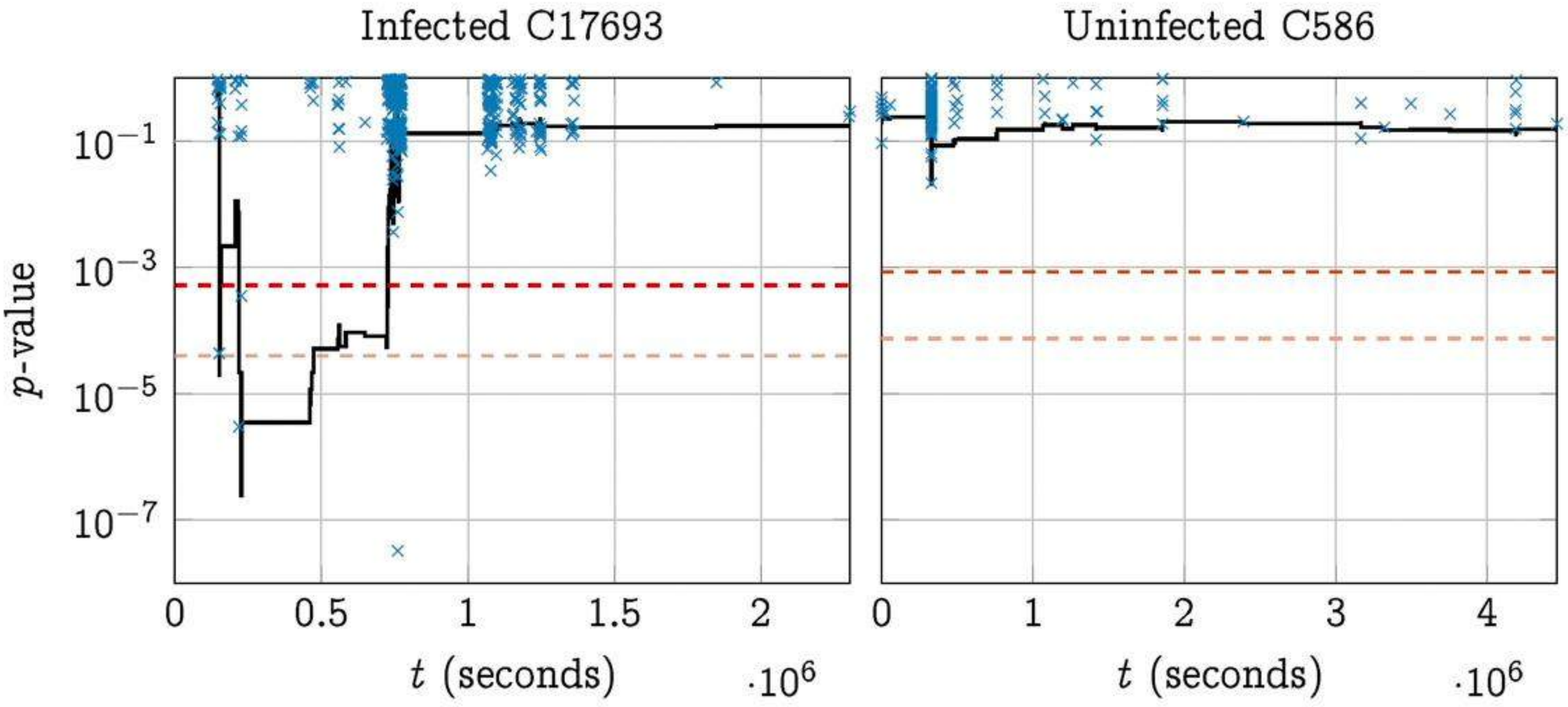
Different hosts form new edges at very different rates, and an unsurprising destination for one host may be highly unusual for another.

We require models for both the rate at which a host makes new edges, and the identities of those connections.

We model the conditional intensity of a new directed connection being formed for every possible (source,destination) pair $x \in X$, $y \in Y$:

$$\lambda_{xy}(t) = \mathbb{I}_{(X \times Y) \setminus G_t}\{(x, y)\}\ r(t)\ \exp\{\alpha \cdot (N_x^+(t), N_y^-(t), I_{x,1}(t), I_{x,2}(t)) + \beta_{xy} \cdot Z_{xy}(t)\}$$

- G_t is the network graph of existing edges at time t
- $r(t)$ is a hypothetical model for time of day/day of week variability, treated as a nuisance parameter
- $N_x^+(t), N_y^-(t)$ are the out/in degrees of nodes x and y at time t
- $I_{x,1}(t), I_{x,2}(t)$ indicate if the most recent one or two connections were new edges
- $Z_{xy}(t)$ represents the *attraction* between x and y; from either
  - hard-thresholding, clustering clients and servers
  - soft-thresholding, latent feature models with Indian buffet process prior



Control chart thresholds at the 1% (- - -) and 0.1% (- - -) significance levels.

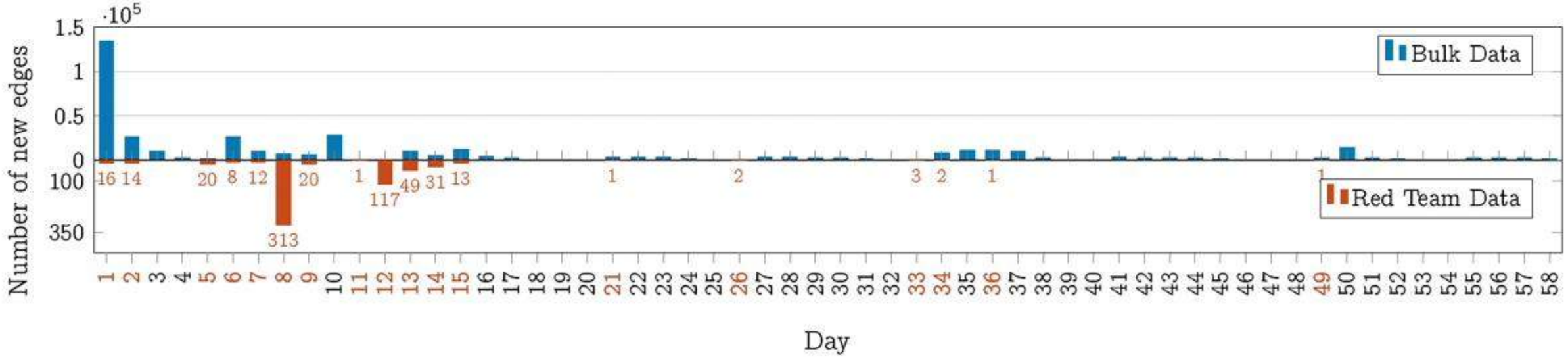
We model the conditional intensity of a new directed connection being formed for every possible (source,destination) pair $x \in X$, $y \in Y$:

$$\lambda_{xy}(t) = \mathbb{I}_{(X \times Y) \setminus G_t}\{(x, y)\}\ r(t)\ \exp\{\alpha \cdot (N_x^+(t), N_y^-(t), I_{x,1}(t), I_{x,2}(t)) + \beta_{xy} \cdot Z_{xy}(t)\}$$

- G_t is the network graph of existing edges at time t
- $r(t)$ is a hypothetical model for time of day/day of week variability, treated as a nuisance parameter
- $N_x^+(t), N_y^-(t)$ are the out/in degrees of nodes x and y at time t
- $I_{x,1}(t), I_{x,2}(t)$ indicate if the most recent one or two connections were new edges
- $Z_{xy}(t)$ represents the *attraction* between x and y; from either
  - hard-thresholding, clustering clients and servers
  - soft-thresholding, latent feature models with Indian buffet process prior

New edges (with S. Metelli)

The red team attack within public Los Alamos enterprise network data contained a spate of new edges being formed between hosts which had not previously connected.



Different hosts form new edges at very different rates, and an unsurprising destination for one host may be highly unusual for another.

We require models for both the rate at which a host makes new edges, and the identities of those connections.

We model the conditional intensity of a new directed connection being formed for every possible (source,destination) pair $x \in X$, $y \in Y$:

$$\lambda_{xy}(t) = \mathbb{I}_{(X \times Y) \setminus G_t}\{(x, y)\}\ r(t)\ \exp\{\alpha \cdot (N_x^+(t), N_y^-(t), I_{x,1}(t), I_{x,2}(t)) + \beta_{xy} \cdot Z_{xy}(t)\}$$

- G_t is the network graph of existing edges at time t
- $r(t)$ is a hypothetical model for time of day/day of week variability, treated as a nuisance parameter
- $N_x^+(t), N_y^-(t)$ are the out/in degrees of nodes x and y at time t
- $I_{x,1}(t), I_{x,2}(t)$ indicate if the most recent one or two connections were new edges
- $Z_{xy}(t)$ represents the *attraction* between x and y; from either
  - hard-thresholding, clustering clients and servers
  - soft-thresholding, latent feature models with Indian buffet process prior

Related applications

Similar ideas can be used for host-based data, such as the WLS logs from Los Alamos.

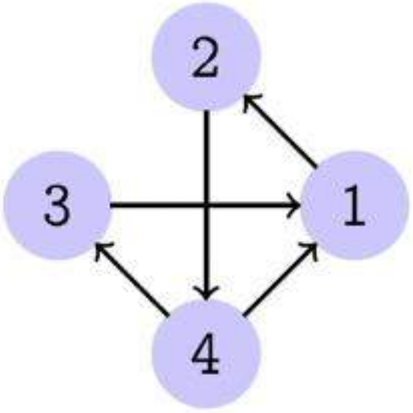
- Modelling the sequences of processes executed by a computer, and measuring accumulated surprise in unlikely processes or poorly predicted new processes
  - Data fusion, combining the host-level process and network-level connection sequences is current work at Turing, under the Data Centric Engineering programme
- Port-scoring: Modelling of server ports in NetFlow on an edge, looking for unusual services (previous work with J. Neil, now @Microsoft)

Whole Graph

Adjacency matrix approaches (with Rubin-Delanchy, Sanna Passino)

We can consider the binary, directed adjacency matrix A_t for the entire network, defined such that $(A_t)_{ij} = 1$ iff $i \to j$ by time t.

$$G_t = \quad \Longrightarrow A_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



A low rank approximation of A_t, or some Laplacian-style transformation, can provide a statistical estimate of the underlying structure.

- SVD approximation: $A \approx \hat{A}_k = U_k \cdot V_k^{\mathsf{T}}$ where U_k, V_k represent the first k columns of the full SVD of A. Small k prevents overfitting
- i/jth row of U_k / V_k provides a latent notional position of client i/server j in $\mathbb{R}^k$

$\widehat{A}_k$ predicts new edges, and identifies anomalous edges

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{pmatrix}$$

Poisson factorisation (with M. Turcotte, F. Sanna Passino)

Alternatively, A_t can be formulated as a matrix of counts. For example, $(A_t)_{ij} = N_{ij}(t)$, the number of connections from user i to computer/process j by time t.
These counts can be regarded as preferential scores, analogous to problems in recommender systems.

In Turcotte et al., 2016 we considered a Poisson factorisation model

$$A_{ij} \sim \text{Poisson}(u_i \cdot v_j^\intercal)$$

- $u_{i\ell} \sim \Gamma(a, \xi_i)$, $v_{j\ell} \sim \Gamma(a, \eta_j)$ $(\ell = 1, \ldots, k)$
- $\xi_i, \eta_j \sim \Gamma(.5, .01)$

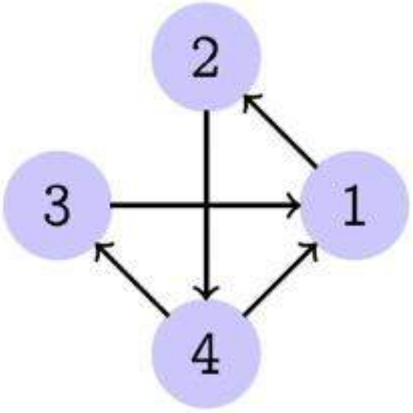
A p-value for anomaly detection was given by the estimated upper tail probability of the observed count; combined for each client using Fisher's method to detect red team.

Current work incorporating known groupings of computers/users with unknown latent factors.

Adjacency matrix approaches (with Rubin-Delanchy, Sanna Passino)

We can consider the binary, directed adjacency matrix A_t for the entire network, defined such that $(A_t)_{ij} = 1$ iff $i \to j$ by time t.

$$G_t = \quad \Longrightarrow A_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



A low rank approximation of A_t, or some Laplacian-style transformation, can provide a statistical estimate of the underlying structure.

- SVD approximation: $A \approx \hat{A}_k = U_k \cdot V_k^{\mathsf{T}}$ where U_k, V_k represent the first k columns of the full SVD of A. Small k prevents overfitting
- i/jth row of U_k / V_k provides a latent notional position of client i/server j in $\mathbb{R}^k$

Poisson factorisation (with M. Turcotte, F. Sanna Passino)

Alternatively, A_t can be formulated as a matrix of counts. For example, $(A_t)_{ij} = N_{ij}(t)$, the number of connections from user i to computer/process j by time t.
These counts can be regarded as preferential scores, analogous to problems in recommender systems.

In Turcotte et al., 2016 we considered a Poisson factorisation model

$$A_{ij} \sim \text{Poisson}(u_i \cdot v_j^\intercal)$$

- $u_{i\ell} \sim \Gamma(a, \xi_i)$, $v_{j\ell} \sim \Gamma(a, \eta_j)$ $(\ell = 1, \ldots, k)$
- $\xi_i, \eta_j \sim \Gamma(.5, .01)$

A p-value for anomaly detection was given by the estimated upper tail probability of the observed count; combined for each client using Fisher's method to detect red team.

Current work incorporating known groupings of computers/users with unknown latent factors.

Topic modelling (with X. Zhang)

Network traffic flowing from an IP address often comes from a mixture of multiple individuals, each exhibiting their own mixtures of behavioural norms. The presence/absence of these components naturally varies over time.

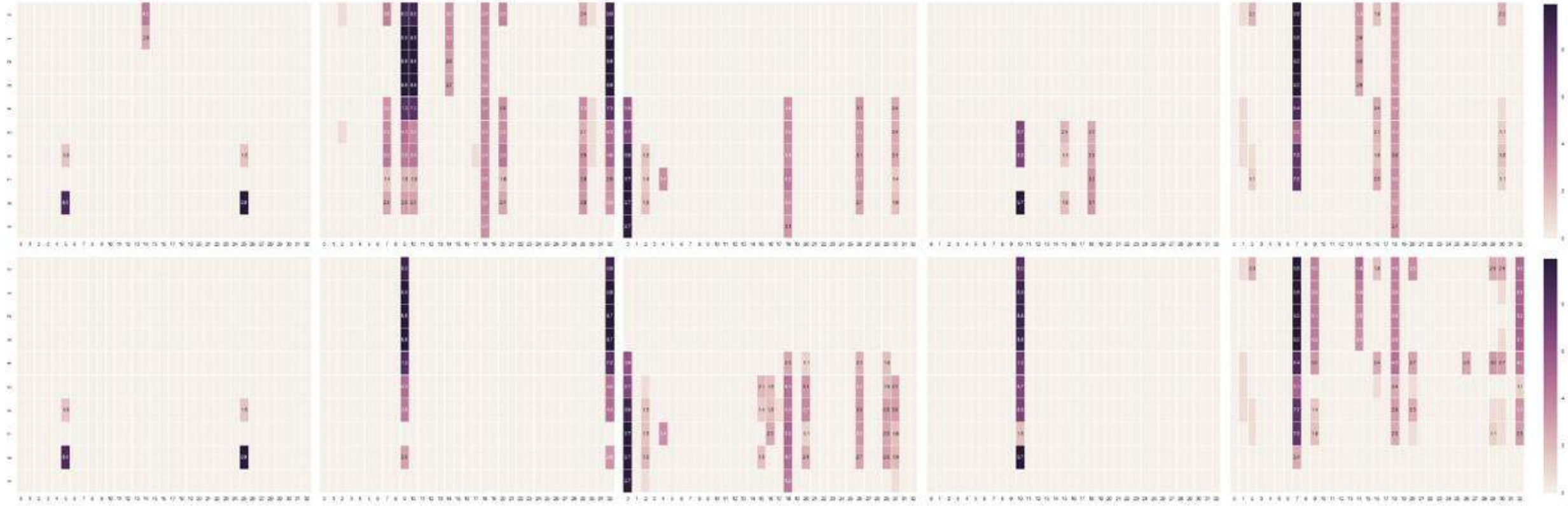
Building accurate network models could be better achieved by decomposing these mixtures.

Latent feature models such as *topic modelling*, typically used in text analysis for automatically classifying the (inferred) topics present in documents, can be deployed.

In cyber, words can be destination IP addresses visited by a client, and topics correspond to users or their behavioural modes (Heard, Palla, and Skoularidou, 2016).

Bayesian nonparametrics allow potentially infinitely many, temporally-occurring topics.

LANL servers visited by 5 users over 10 days



Each matrix has days as rows and computers as columns. Top: True separation into the 5 users. Bottom: Inferred "topics".

Topic modelling (with X. Zhang)

Network traffic flowing from an IP address often comes from a mixture of multiple individuals, each exhibiting their own mixtures of behavioural norms. The presence/absence of these components naturally varies over time.

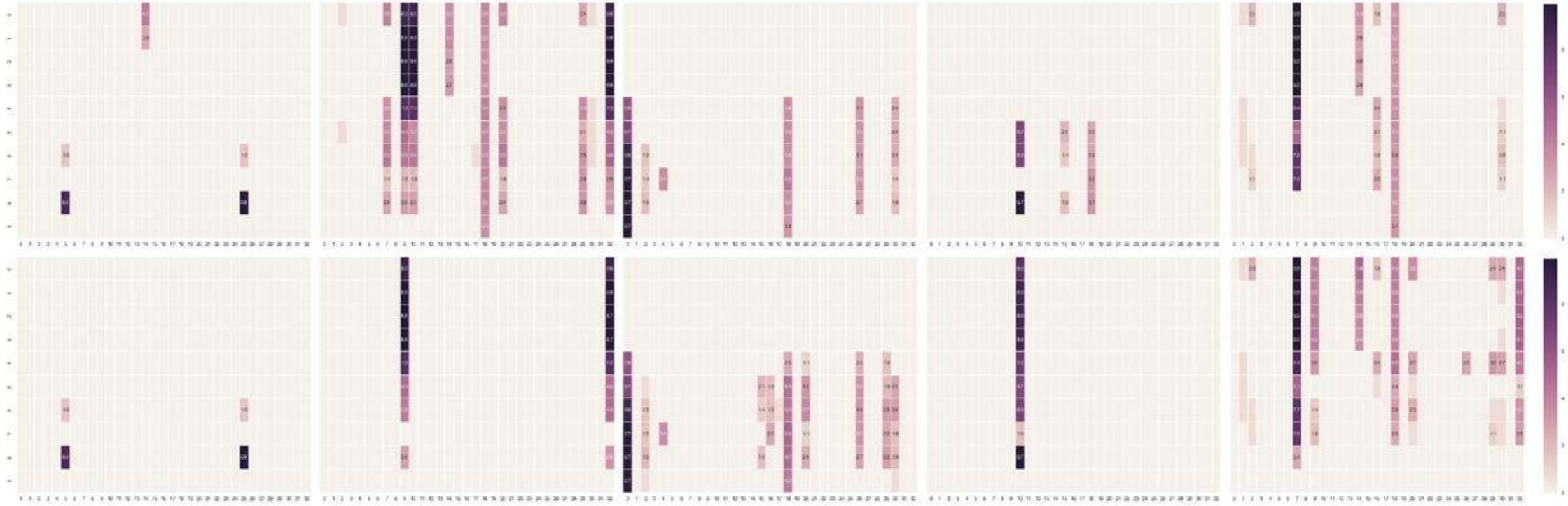
Building accurate network models could be better achieved by decomposing these mixtures.

Latent feature models such as *topic modelling*, typically used in text analysis for automatically classifying the (inferred) topics present in documents, can be deployed.

In cyber, words can be destination IP addresses visited by a client, and topics correspond to users or their behavioural modes (Heard, Palla, and Skoularidou, 2016).

Bayesian nonparametrics allow potentially infinitely many, temporally-occurring topics.

LANL servers visited by 5 users over 10 days

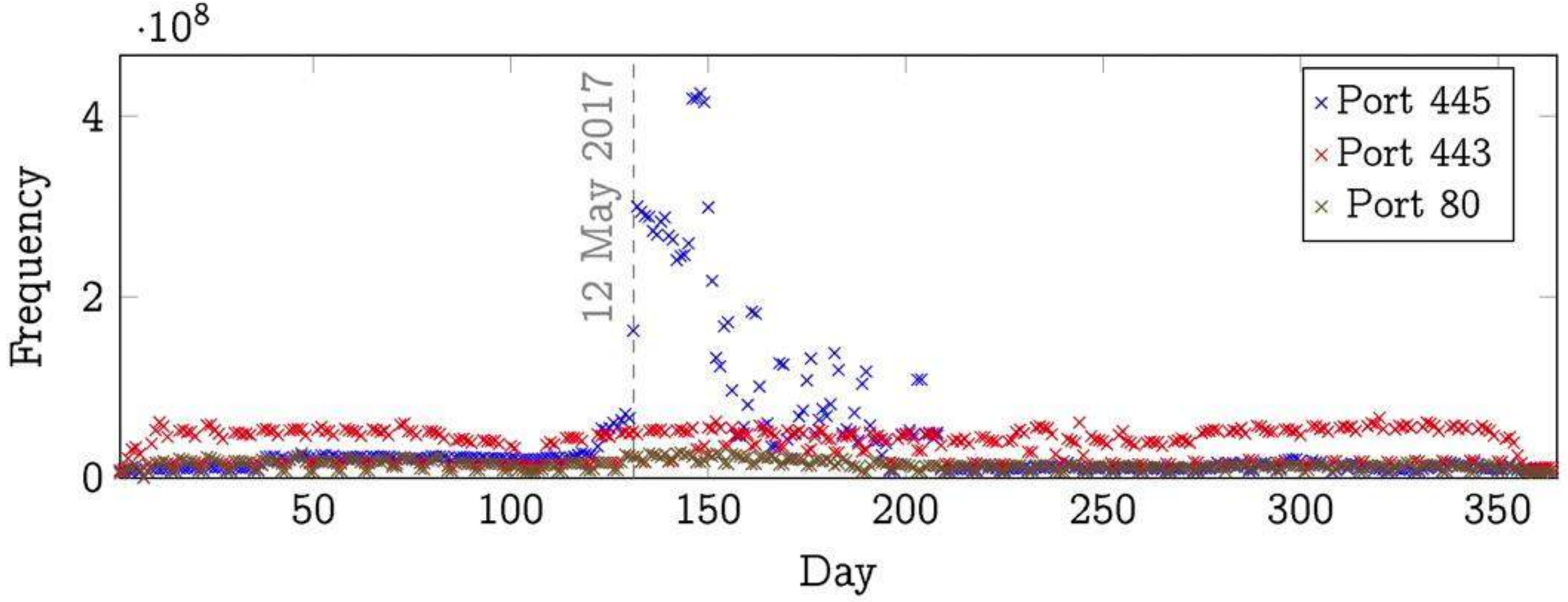


Each matrix has days as rows and computers as columns. Top: True separation into the 5 users. Bottom: Inferred "topics".

Network-wide change detection (with K. Hallgren)

Finally, it can be informative to model the volumes of different types of traffic passing through the network over time, to detect any significant changes.

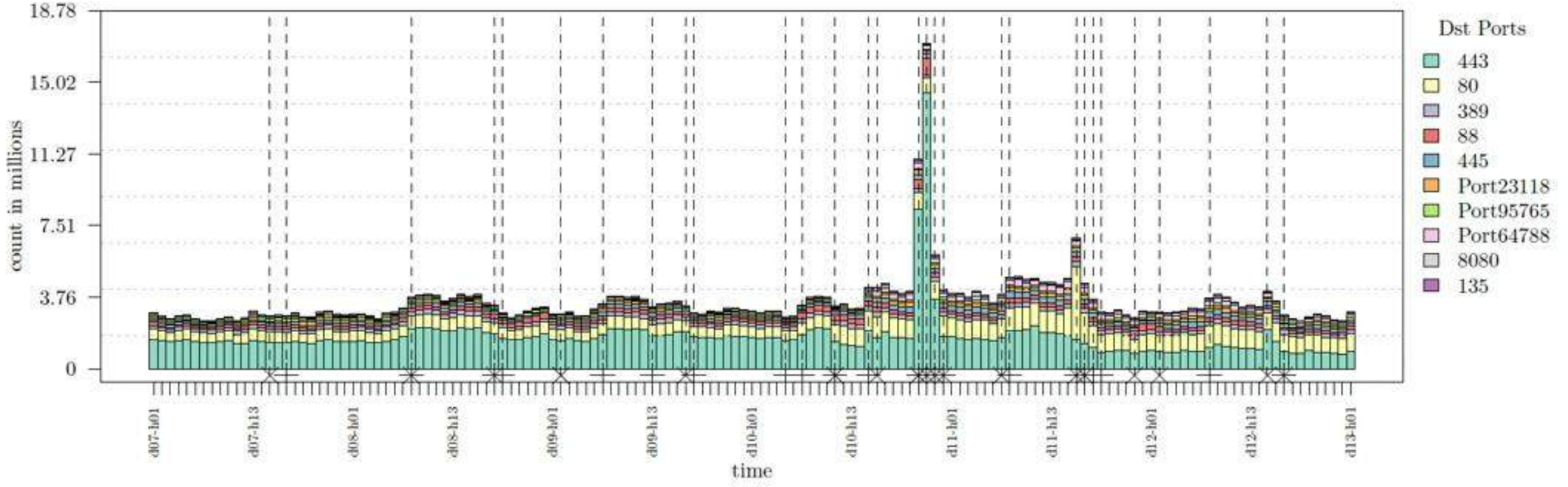
For example, the 2017 Wannacry ransomware attack (the largest cyber attack to hit the UK so far) produced a strong peak in traffic on TCP service port 445, amongst others.



Aim to monitor for changes in a range of different features:

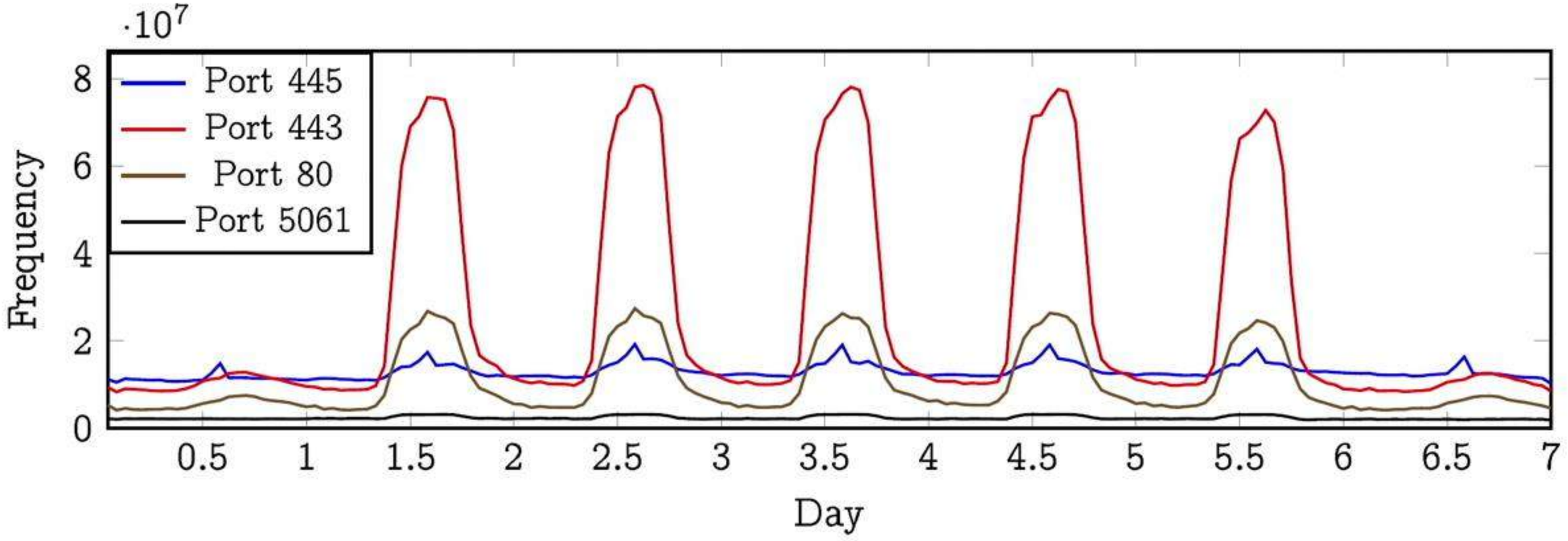
- services
- traffic volumes
- data volumes
- geo-locations

We are interested in changepoint analysis with marked changepoints indicating which aspects of the data collection have changed.



ICL TCP traffic by time of week (2017, pre Wannacry)

Much seasonal variation across the week $\Longrightarrow$ too many changepoints



Difficulties applying changepoint detection

Changepoint models rely on partitioning the passage of time into segments, and fitting relatively simple models within each segment to provide a global model with higher complexity.

Often the model is an oversimplification, and therefore more changepoints are fitted than would be preferable.

Modelling seasonality explicitly, for example, is one alternative; but rigid models of seasonality often fail, since no two weeks are the same. And generally, building such bespoke models is labour-intensive.

Instead, current work admits the model for data within changepoint segments may be misspecified, and requires clear discontinuity for a changepoint rather than gradual drift.

Bayesian changepoint analysis: an alternative view

Let $\tau = (\tau_1, \tau_2, \ldots)$ be a sequence of (unknown) changepoints in a time-indexed piecewise deterministic process θ_t, such that $\theta_t = \theta^{(j)}$ for $\tau_j \leq t < \tau_{j+1}$.

Assuming discrete-time (not essential), suppose we observe $X = (X_1, X_2, \ldots, X_n)$ with each X_t independently drawn from

$$\mathbb{P}(X_t | \theta_t)$$

Bayesian inference on τ is computationally tractable when assuming independent, conjugate priors for the segment parameters $\{\theta^{(j)}\}$. (Or priors well-approximated by conjugate priors, admitting the possibility of importance sampling.)

- The parameters $\{\theta^{(j)}\}$ can then be integrated out, and commonly the resulting marginal likelihood for X conditional on a proposed vector of changepoints factorises as product of joint distributions of the observables for each segment,

$$\mathbb{P}(X|\tau) = \prod_k \mathbb{P}(X_{\tau_k}, \ldots, X_{\tau_{k+1}-1}).$$

Example: Poisson counts

Suppose within a segment,

$$\theta_1 \sim \Gamma(\alpha, \beta), \qquad X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{Poisson}(\theta_1)$$

Importantly, and slightly paradoxically, this sampling scheme is *equivalent* to the following dynamic-θ strategy:

$$\begin{aligned}
\theta_1 &\sim \Gamma(\alpha, \beta), & X_1 &\sim \text{Poisson}(\theta_1) \\
\theta_2 &\sim \Gamma(\alpha + X_1, \beta + 1), & X_2 &\sim \text{Poisson}(\theta_2) \\
&\vdots \\
\theta_n &\sim \Gamma\left(\alpha + \sum_{j=1}^{n-1} X_j, \beta + n - 1\right), & X_n &\sim \text{Poisson}(\theta_n)
\end{aligned}$$

The segment marginal likelihood is of course

$$\mathbb{P}(X_1, \ldots, X_n) = \frac{\beta^{\alpha}}{(\beta + n)^{\alpha + \sum_{j=1}^{n} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{n} X_j)}{\Gamma(\alpha)}$$

and this naturally factorises as the product of predictive probabilities

$$\prod_{i=1}^{n} \mathbb{P}(X_i | X_{i-1}, \ldots, X_1) = \prod_{i=1}^{n} \frac{(\beta + i - 1)^{\alpha + \sum_{j=1}^{i-1} X_j}}{(\beta + j)^{\alpha + \sum_{j=1}^{i} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{i} X_j)}{\Gamma(\alpha + \sum_{j=1}^{i-1} X_j)}$$

By De Finetti's representation theorem, it is precisely this structure that guarantees exchangeability of observations within a segment.
So to relax the assumption of exchangeability, and allow temporal trends within segments but still maintain analytic tractability, we can simply break this full conditioning.

Robust Bayesian changepoint analysis

For example, consider Markov order-k conditioning:

$$\mathbb{P}_k(X_1, \ldots, X_n) := \prod_{i=1}^{n} \mathbb{P}_k(X_i | X_{i-1}, \ldots, X_{i-k}) =$$

$$\frac{\beta^\alpha}{(\beta + k + 1)^{\alpha + \sum_{j=1}^{k+1} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{k+1} X_j)}{\Gamma(\alpha)} \prod_{i=k+2}^{n} \frac{(\beta + i - 1)^{\alpha + \sum_{j=i-k}^{i-1} X_j}}{(\beta + j)^{\alpha + \sum_{j=i-k}^{i} X_j}} \frac{\Gamma(\alpha + \sum_{j=i-k}^{i} X_j)}{\Gamma(\alpha + \sum_{j=i-k}^{i-1} X_j)}$$

Averaging $\mathbb{P}_k(X_1, \ldots, X_n)$ over a prior Q on k (e.g. geometric) provides a more robust model for the data in a changepoint segment: $\mathbb{P}_Q(X_1, \ldots, X_n) := \int \mathbb{P}_k(X_1, \ldots, X_n) \mathrm{d}Q(k)$.

The segment marginal likelihood is of course

$$\mathbb{P}(X_1, \ldots, X_n) = \frac{\beta^{\alpha}}{(\beta + n)^{\alpha + \sum_{j=1}^{n} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{n} X_j)}{\Gamma(\alpha)}$$

and this naturally factorises as the product of predictive probabilities

$$\prod_{i=1}^{n} \mathbb{P}(X_i | X_{i-1}, \ldots, X_1) = \prod_{i=1}^{n} \frac{(\beta + i - 1)^{\alpha + \sum_{j=1}^{i-1} X_j}}{(\beta + j)^{\alpha + \sum_{j=1}^{i} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{i} X_j)}{\Gamma(\alpha + \sum_{j=1}^{i-1} X_j)}$$

By De Finetti's representation theorem, it is precisely this structure that guarantees exchangeability of observations within a segment.
So to relax the assumption of exchangeability, and allow temporal trends within segments but still maintain analytic tractability, we can simply break this full conditioning.

Robust Bayesian changepoint analysis

For example, consider Markov order-k conditioning:

$$\mathbb{P}_k(X_1, \ldots, X_n) := \prod_{i=1}^{n} \mathbb{P}_k(X_i | X_{i-1}, \ldots, X_{i-k}) =$$

$$\frac{\beta^\alpha}{(\beta + k + 1)^{\alpha + \sum_{j=1}^{k+1} X_j}} \frac{\Gamma(\alpha + \sum_{j=1}^{k+1} X_j)}{\Gamma(\alpha)} \prod_{i=k+2}^{n} \frac{(\beta + i - 1)^{\alpha + \sum_{j=i-k}^{i-1} X_j}}{(\beta + j)^{\alpha + \sum_{j=i-k}^{i} X_j}} \frac{\Gamma(\alpha + \sum_{j=i-k}^{i} X_j)}{\Gamma(\alpha + \sum_{j=i-k}^{i-1} X_j)}$$

Averaging $\mathbb{P}_k(X_1, \ldots, X_n)$ over a prior Q on k (e.g. geometric) provides a more robust model for the data in a changepoint segment: $\mathbb{P}_Q(X_1, \ldots, X_n) := \int \mathbb{P}_k(X_1, \ldots, X_n) \mathrm{d}Q(k)$.

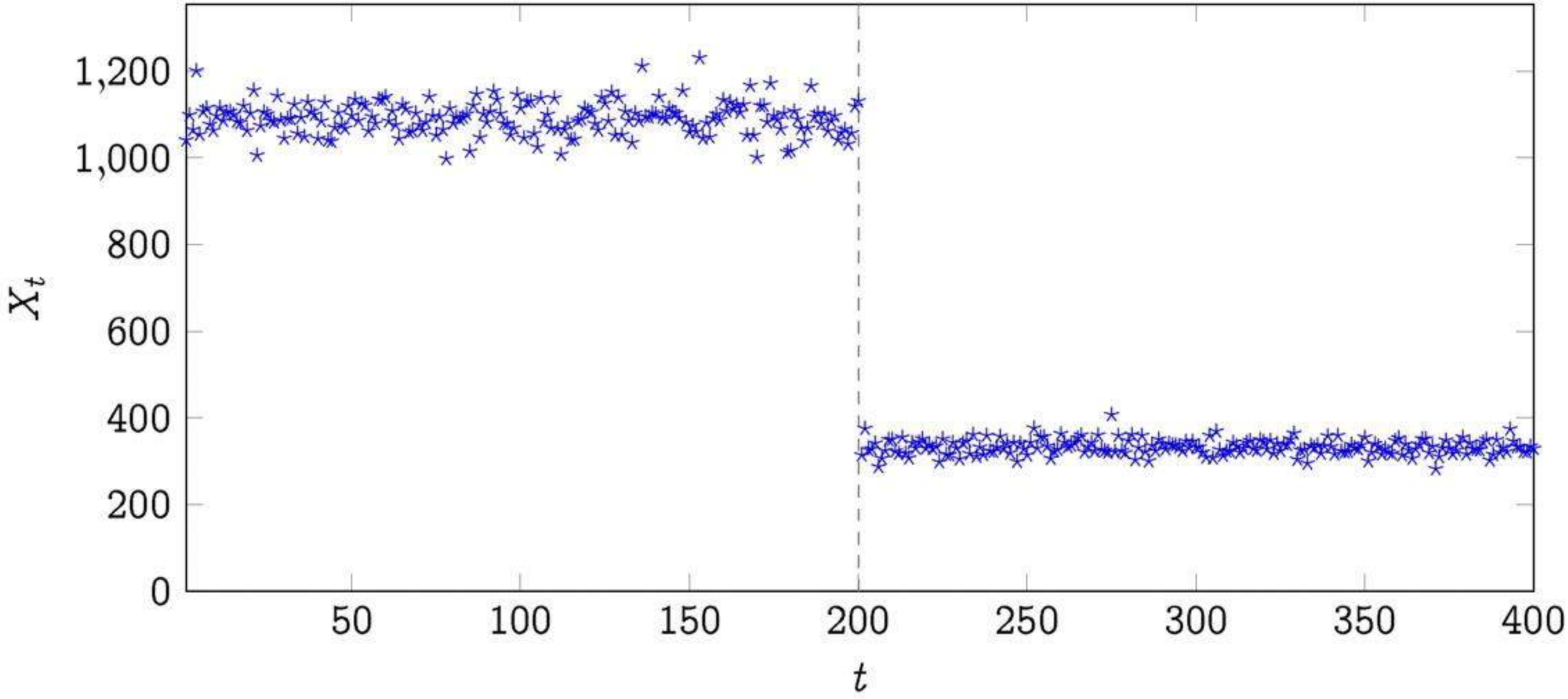
For the dynamic-θ formulation, this is analogous to drawing

$$\theta_n \sim \Gamma\left(\alpha + \sum_{j=n-k}^{n-1} X_j, \beta + k - 1\right), \qquad X_n \sim \text{Poisson}(\theta_n)$$

Interpretation of this truncated conditioning: *There may have been a changepoint at any Q-distributed time in the past, truncated by the inferred changepoint τ_j.*

$\implies \tau_1, \tau_2, \ldots$ therefore take on the interpretation of *definite* changepoints.

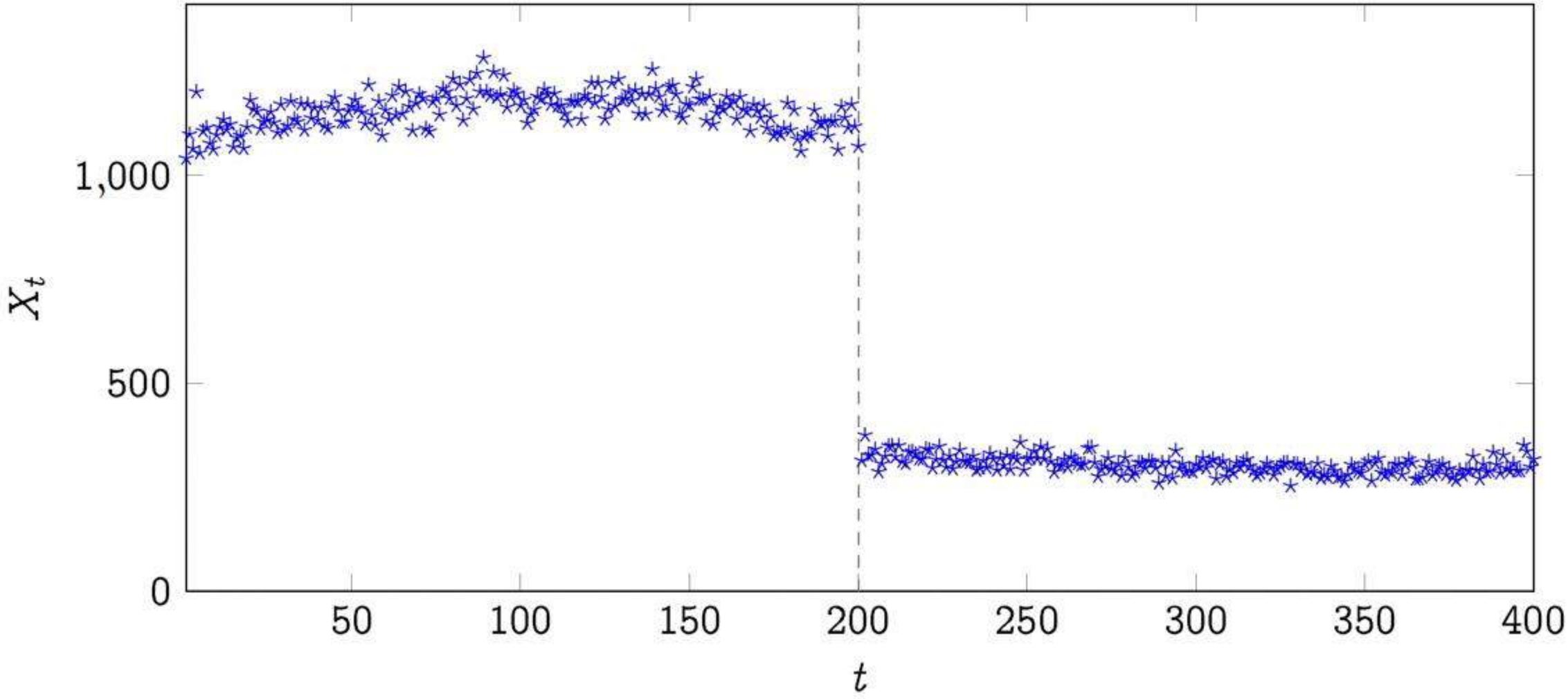
Example simulation



$\alpha = 0.002, \beta = 1.$

$k = \infty$

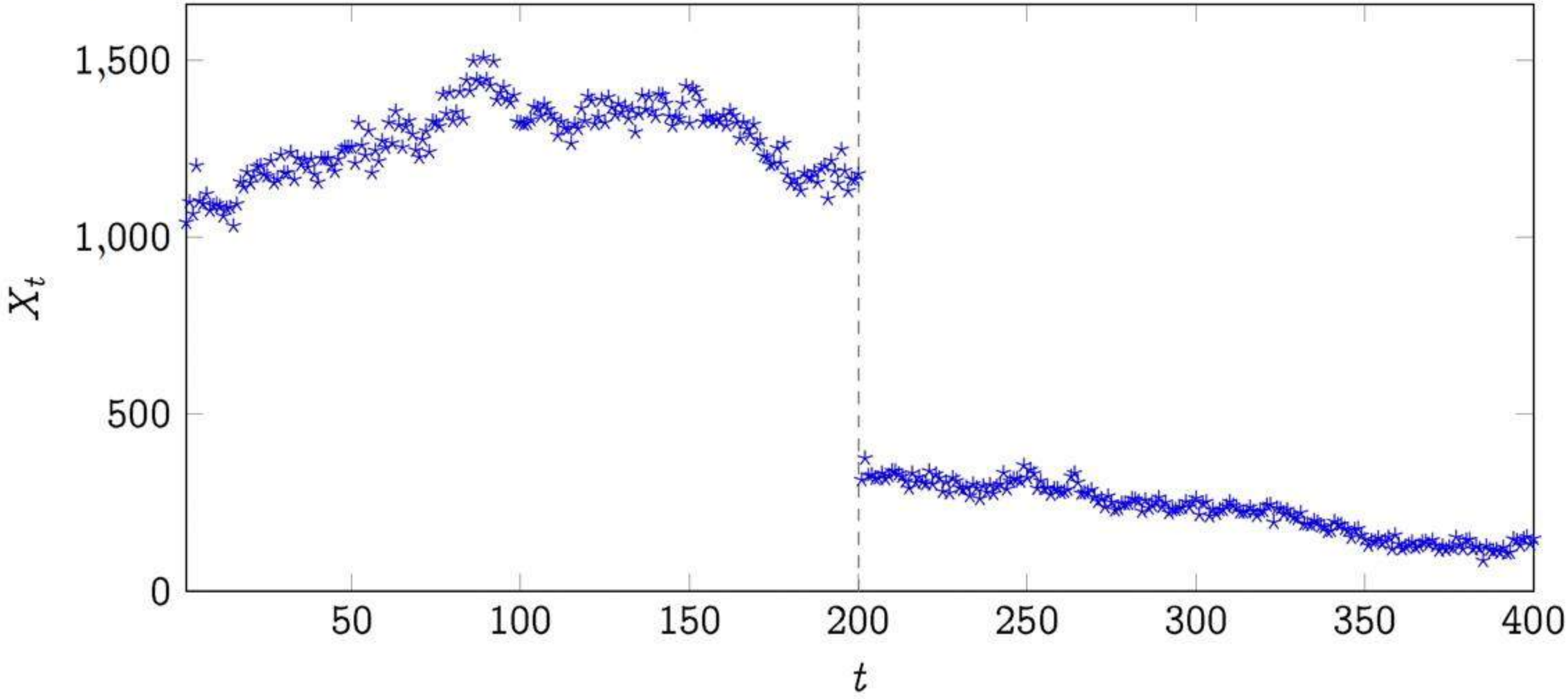
Example simulation



$\alpha = 0.002, \beta = 1.$

$k = 10$

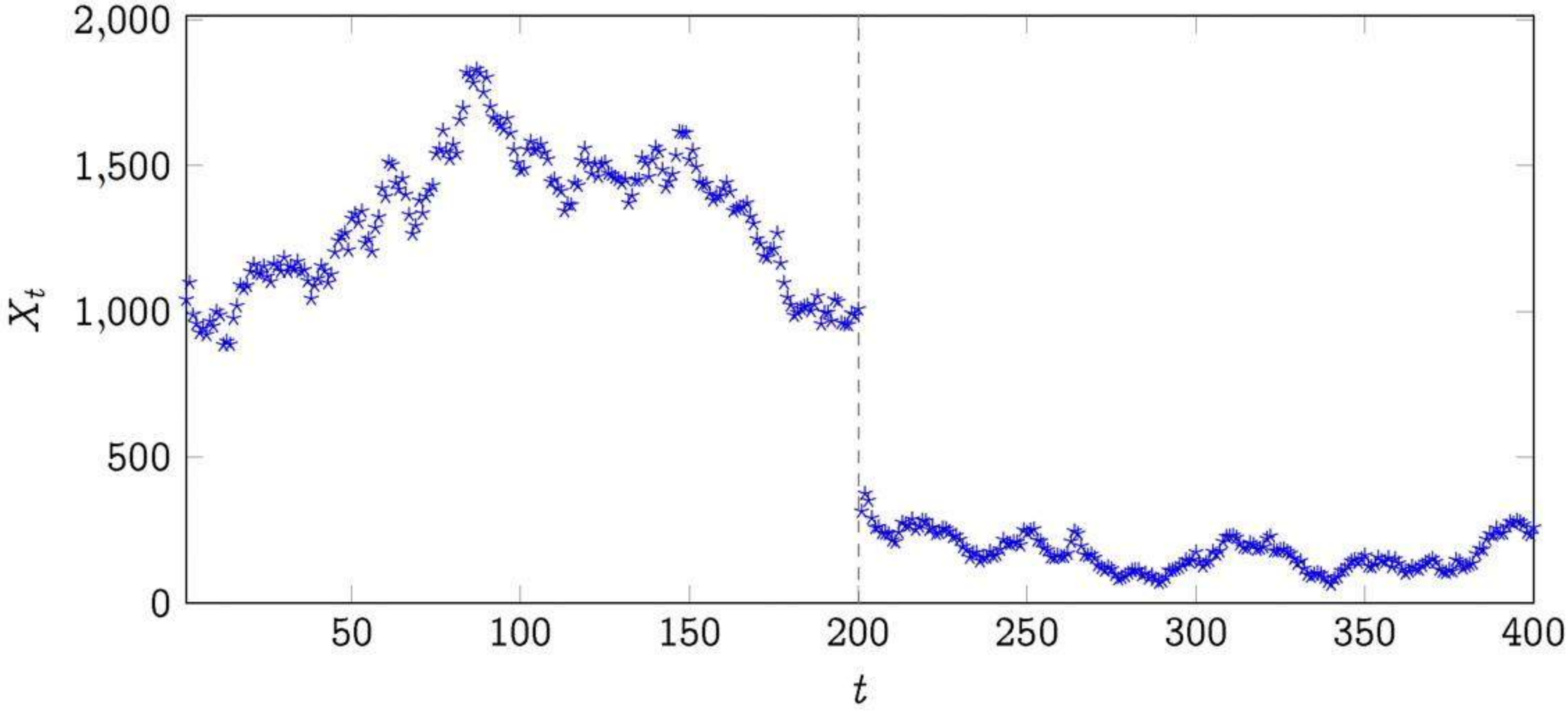
Example simulation



$\alpha = 0.002, \beta = 1.$

$k = 5$

Example simulation



$\alpha = 0.002, \beta = 1.$

$k = 1$

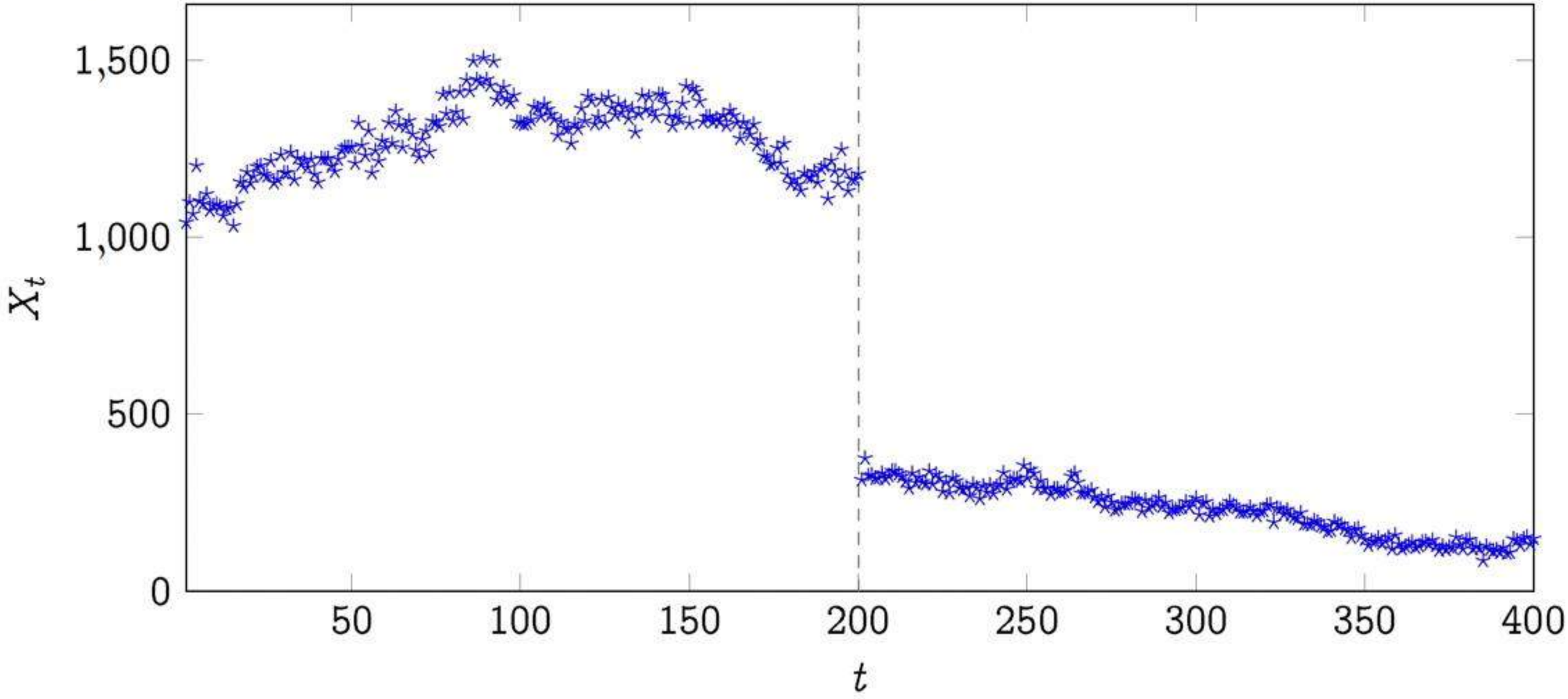
For the dynamic-θ formulation, this is analogous to drawing

$$\theta_n \sim \Gamma\left(\alpha + \sum_{j=n-k}^{n-1} X_j, \beta + k - 1\right), \qquad X_n \sim \text{Poisson}(\theta_n)$$

Interpretation of this truncated conditioning: *There may have been a changepoint at any Q-distributed time in the past, truncated by the inferred changepoint τ_j.*

$\implies \tau_1, \tau_2, \ldots$ therefore take on the interpretation of *definite* changepoints.

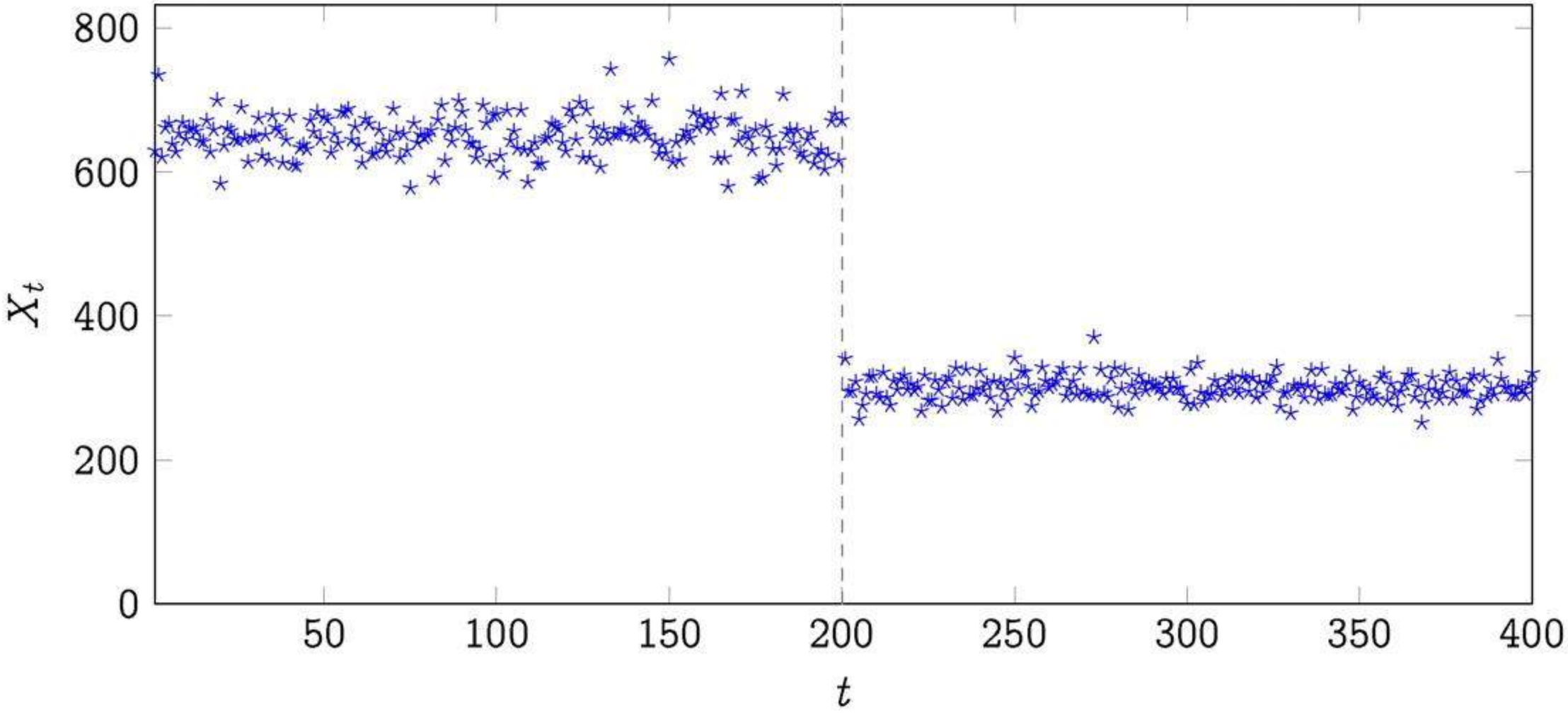
Example simulation



$\alpha = 0.002, \beta = 1.$

$k = 5$

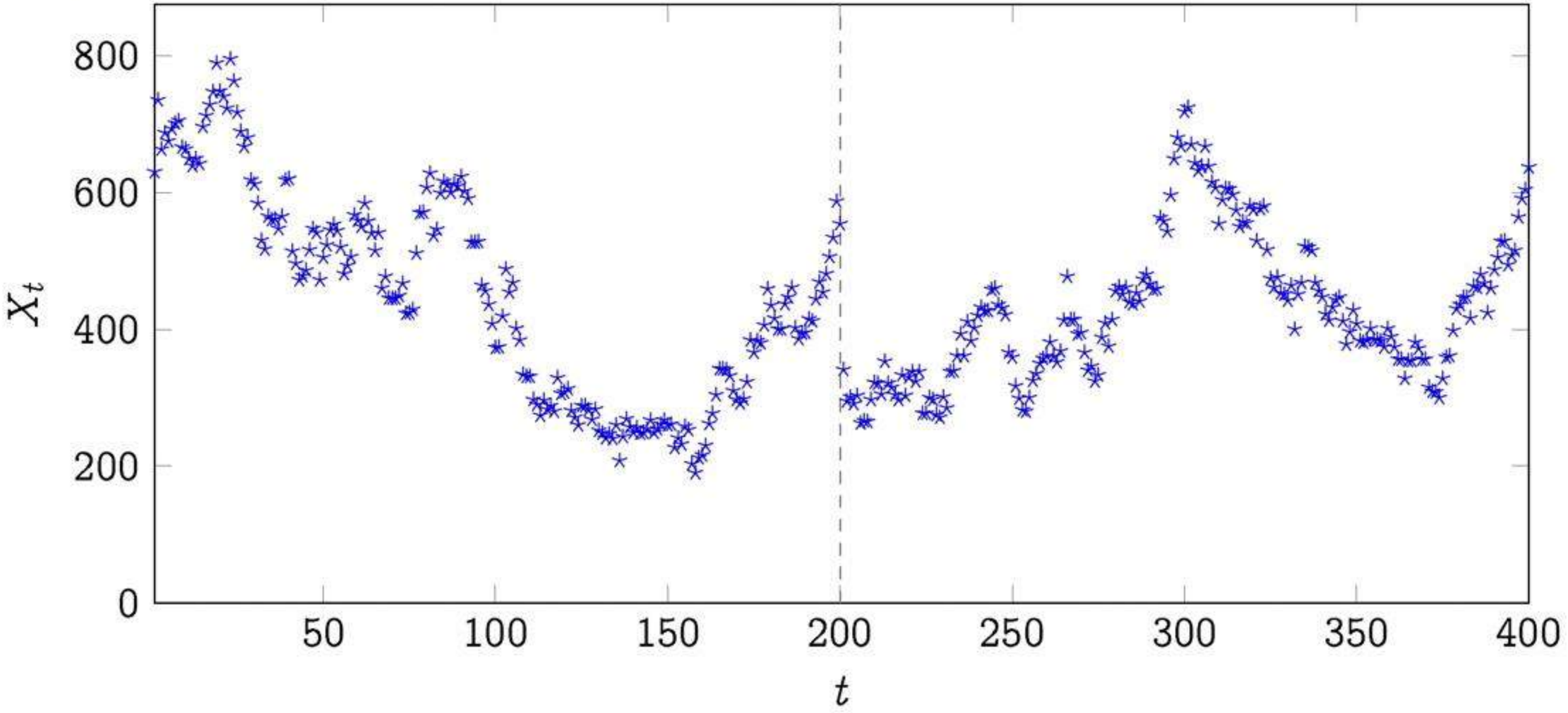
Example simulation



$\alpha = 0.02, \beta = 10.$

$k = \infty$

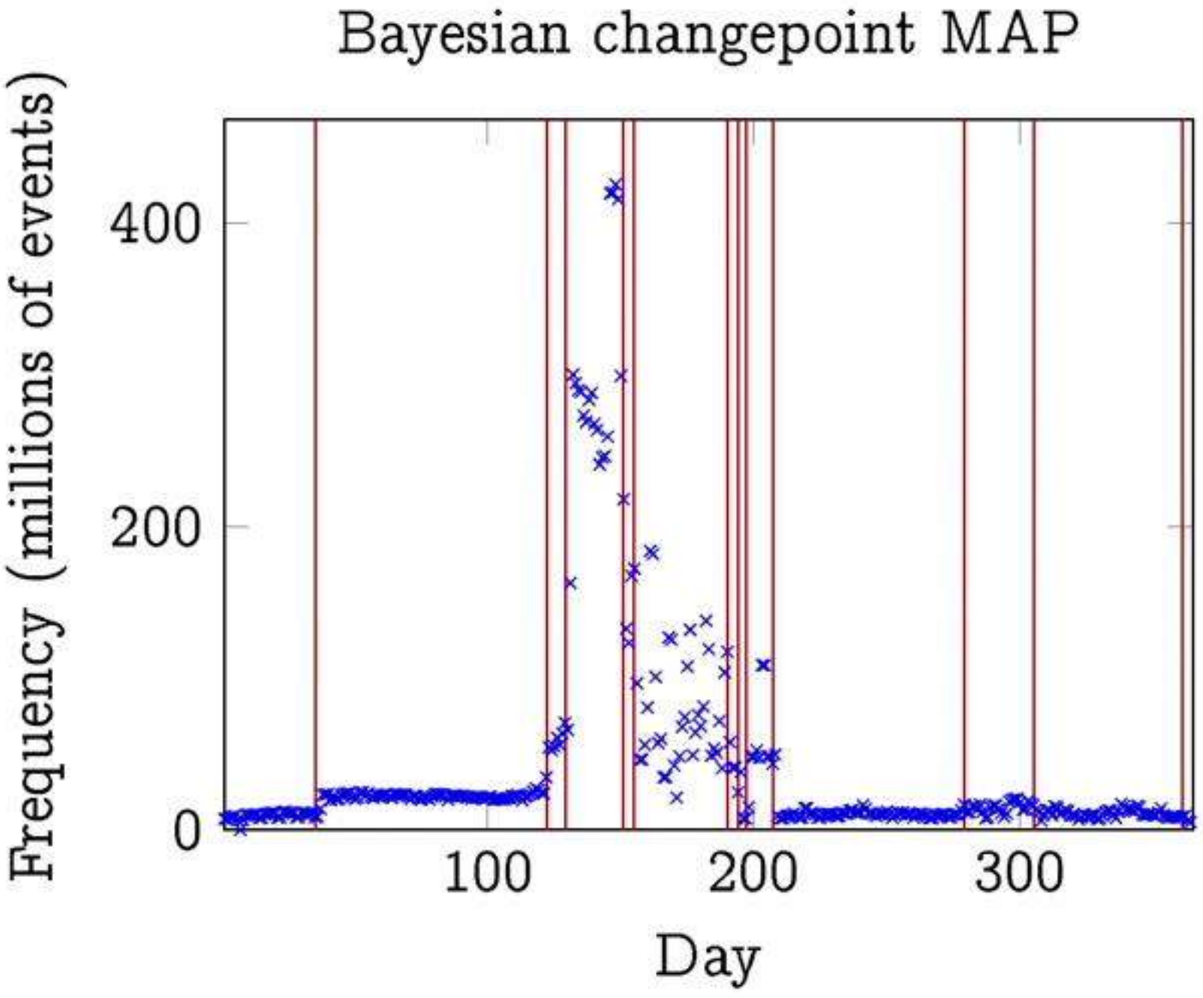
Example simulation

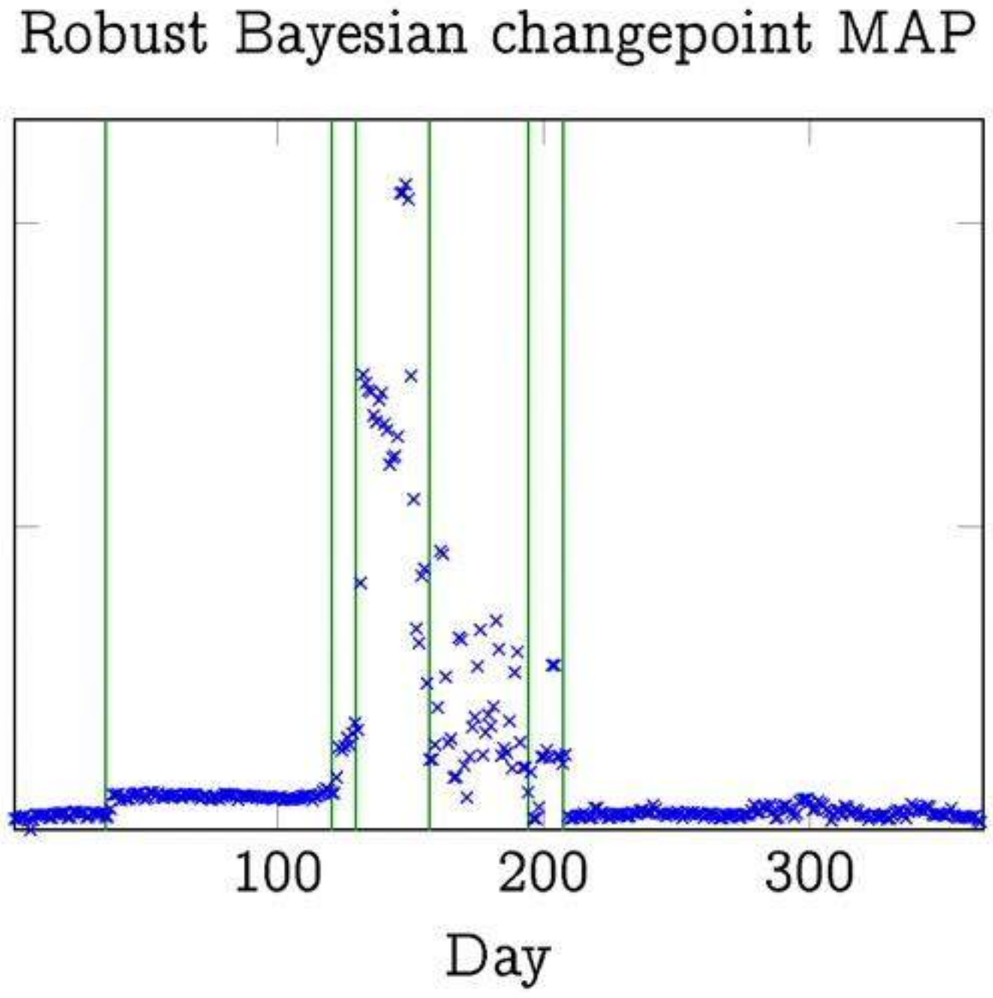


$\alpha = 0.02, \beta = 10.$

$k = 1$

2017 Port 445 data





Conclusions

Statistical methods provide a principled framework for automating the detection of significantly unusual cyber behaviour.

Analyses can be performed at different scales, ranging from edge-level to full graph analyses.

At each level of resolution, the models will typically be under-specified due to the complex natures of both human and automated network traffic. But the calculus of probability still provides a coherent scale for priotising the most interesting discoveries.

Much future work should be concerned with identifying robust anomaly detection methods, through a combination of:

- Robust models
- Combining evidence/performing data fusion to synthesise multiple weak signals into a strong signal

References

Nick Heard, Konstantina Palla, and Maria Skoularidou. "Topic modelling of authentication events in an enterprise computer network". In: *IEEE Intelligence and Security Informatics Conference (ISI2016), Cybersecurity and Big Data*. IEEE. 2016.

Nick Heard and Patrick Rubin-Delanchy. "Network-wide anomaly detection via the Dirichlet process". In: *IEEE Big Data Analytics for Cybersecurity Computing (BDAC2016)*. IEEE. 2016.

Nick Heard, Patrick Rubin-Delanchy, and Daniel Lawson. "Filtering Automated Polling Traffic in Computer Network Flow Data". In: *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*. IEEE. 2014, pp. 268–271.

Melissa Turcotte et al. "Poisson Factorization for Peer-Based Anomaly Detection". In: *IEEE Intelligence and Security Informatics Conference (ISI2016), Cybersecurity and Big Data*. IEEE. 2016.

Conclusions

Statistical methods provide a principled framework for automating the detection of significantly unusual cyber behaviour.

Analyses can be performed at different scales, ranging from edge-level to full graph analyses.

At each level of resolution, the models will typically be under-specified due to the complex natures of both human and automated network traffic. But the calculus of probability still provides a coherent scale for priotising the most interesting discoveries.

Much future work should be concerned with identifying robust anomaly detection methods, through a combination of:

- Robust models
- Combining evidence/performing data fusion to synthesise multiple weak signals into a strong signal

Related applications

Similar ideas can be used for host-based data, such as the WLS logs from Los Alamos.

- Modelling the sequences of processes executed by a computer, and measuring accumulated surprise in unlikely processes or poorly predicted new processes
  - Data fusion, combining the host-level process and network-level connection sequences is current work at Turing, under the Data Centric Engineering programme
- Port-scoring: Modelling of server ports in NetFlow on an edge, looking for unusual services (previous work with J. Neil, now @Microsoft)

We model the conditional intensity of a new directed connection being formed for every possible (source,destination) pair $x \in X$, $y \in Y$:

$$\lambda_{xy}(t) = \mathbb{I}_{(X \times Y) \setminus G_t}\{(x, y)\}\ r(t)\ \exp\{\alpha \cdot (N_x^+(t), N_y^-(t), I_{x,1}(t), I_{x,2}(t)) + \beta_{xy} \cdot Z_{xy}(t)\}$$

- G_t is the network graph of existing edges at time t
- $r(t)$ is a hypothetical model for time of day/day of week variability, treated as a nuisance parameter
- $N_x^+(t), N_y^-(t)$ are the out/in degrees of nodes x and y at time t
- $I_{x,1}(t), I_{x,2}(t)$ indicate if the most recent one or two connections were new edges
- $Z_{xy}(t)$ represents the *attraction* between x and y; from either
  - hard-thresholding, clustering clients and servers
  - soft-thresholding, latent feature models with Indian buffet process prior

Monitoring node connectivity

We can monitor the sequence of destinations a node connects to and look for bursts of unusual connectivity.

Each connection event is scored, and surprise is aggregated using control charts or p-value combination techniques.

(Heard and Rubin-Delanchy, 2016) Server modelling: The sequence of clients $x_1, x_2, \ldots$ connecting to a server y were modelled as a multinomial, with an unbounded number of categories and a Dirichlet process prior (with some base measure αF_0) on the category probabilities.

The p-value score for observation x_{n+1}:

$$p_{n+1} = \sum_{x \in V : \alpha_x^* \leq \alpha_{x_{n+1}}^*} \frac{\alpha_x^*}{\alpha^*}$$

- $\alpha_x^* = \alpha F_0(x) + \sum_{i=1}^{n} \mathbb{I}_x(x_i)$ and $\alpha^* = \alpha + n$.

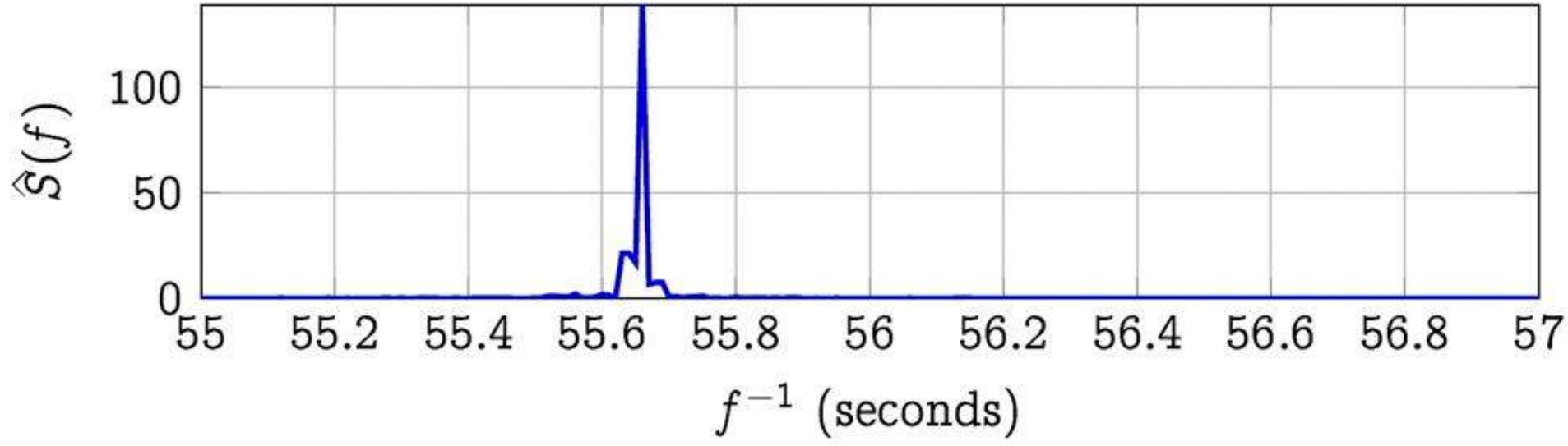
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:



Distinguishing Automated from Human Traffic

Automated edges typically carry "super-human" traffic volumes. But not always, and so we need more sophisticated filters.

Automated events are often highly periodic, corresponding to scheduled beacons pushing refreshes and updates, or "keep-alives".

(Heard, Rubin-Delanchy, and Lawson, 2014) We can scan for periodicities in event times by inspecting the periodogram after time T,

$$\widehat{S}(f) = \frac{1}{T}\left|\sum_{t=1}^{T}\{\mathrm{d}N(t) - N(T)/T\}e^{-2\pi\imath ft}\right|^2.$$

which can be efficiently calculated at each of the Fourier frequencies $f_k = k/T$, $k = 1, \ldots, \lfloor T/2 \rfloor$, via the Fast Fourier Transform.

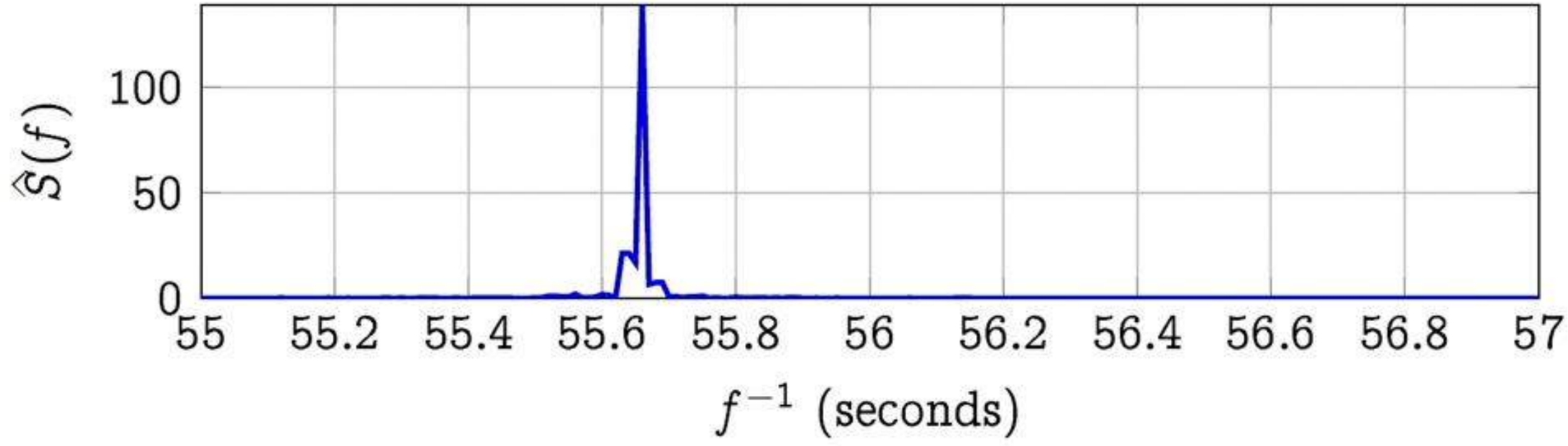
$\widehat{S}(f)$ is proportional to the squared magnitude of the resultant vector when the events are represented as unit vectors on the f^{-1}-second clock.

By the CLT, if events arrive as a Poisson process then asymptotically $\forall f, \widehat{S}(f) \sim \chi^2_2$.

Fisher's g-test provides approximate p-values for the relative magnitude of the peak of $\widehat{S}$,

$$g(\widehat{S}) = \frac{\max_k \widehat{S}(f_k)}{\sum_{1 \le k \le \lfloor T/2 \rfloor} \widehat{S}(f_k)}.$$

Example: My computer to Dropbox:

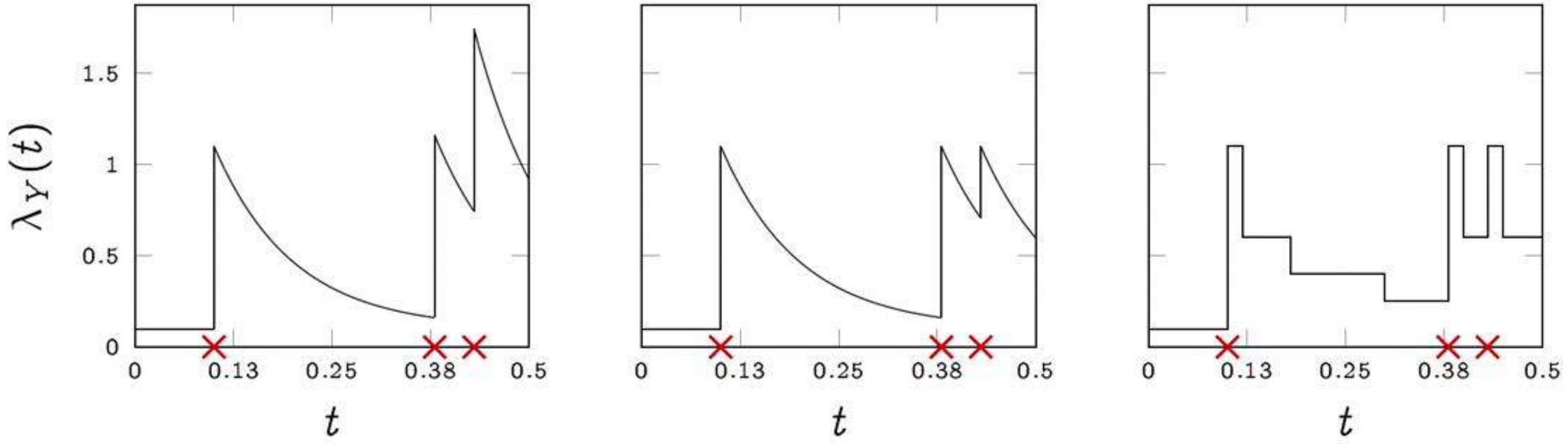


Modelling bursts of events (with M. Price-Williams)

Even human-generated network connections do not arrive as an (inhomogeneous) Poisson process. They occur in bursts, on the same edge and between edges.

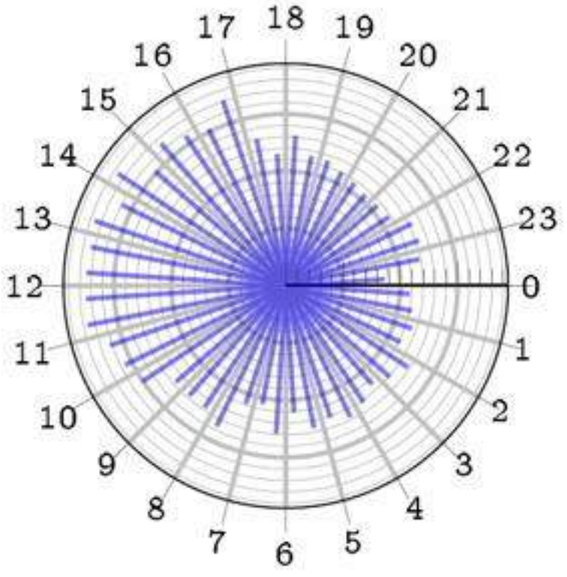
We model arrivals of event times $y_1, y_2, \ldots$ as a Wold process with self-exciting conditional intensity

$$\lambda_Y(t) = \lambda + \sum_{j=1}^{\ell} \lambda_j \mathbb{I}_{[\tau_{j-1},\tau_j)}(t - y_{Y(t)})$$

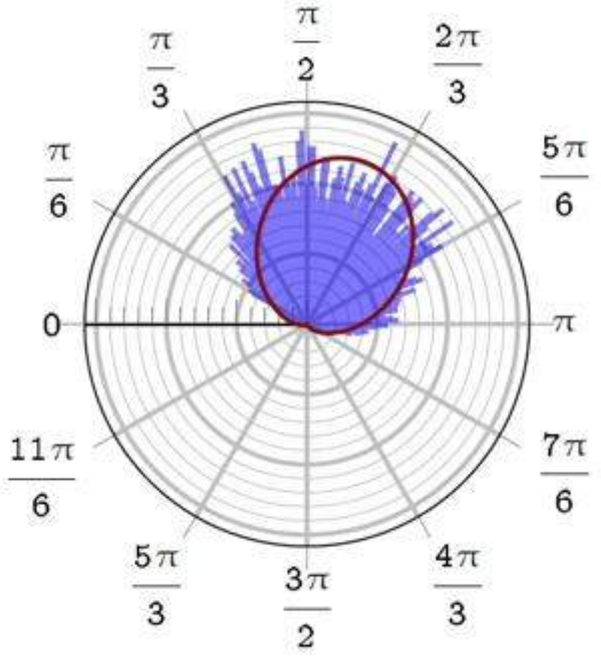


- We fit a Bayesian mixture model to learn the human and automated components:
  - f_A: Wrapped normal density for learning phase (circular mean) and variance of beacons
  - f_H: Piecewise constant density to consistently estimate human event distribution
- Events are probabilistically attributed to either f_A or f_H
- Example: 13.107.42.11 (`outlook.com`), polling at $\approx 8s$ intervals

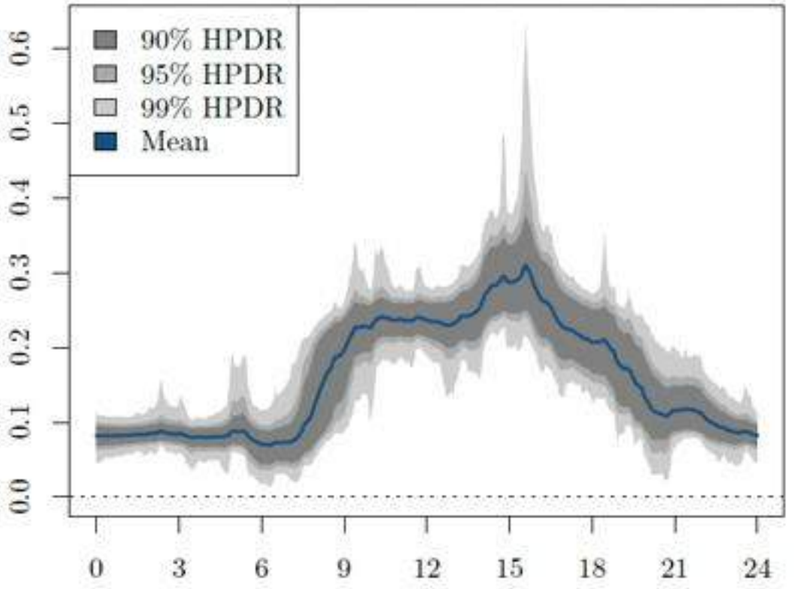
Event time distribution



Wrapped normal



(Averaged) Step function density

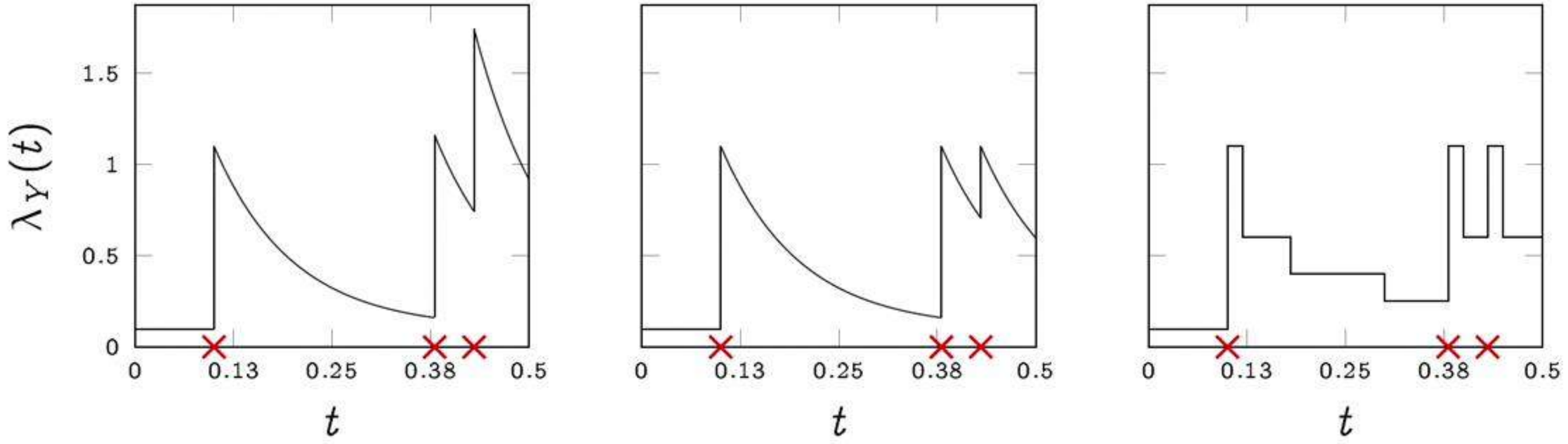


Modelling bursts of events (with M. Price-Williams)

Even human-generated network connections do not arrive as an (inhomogeneous) Poisson process. They occur in bursts, on the same edge and between edges.

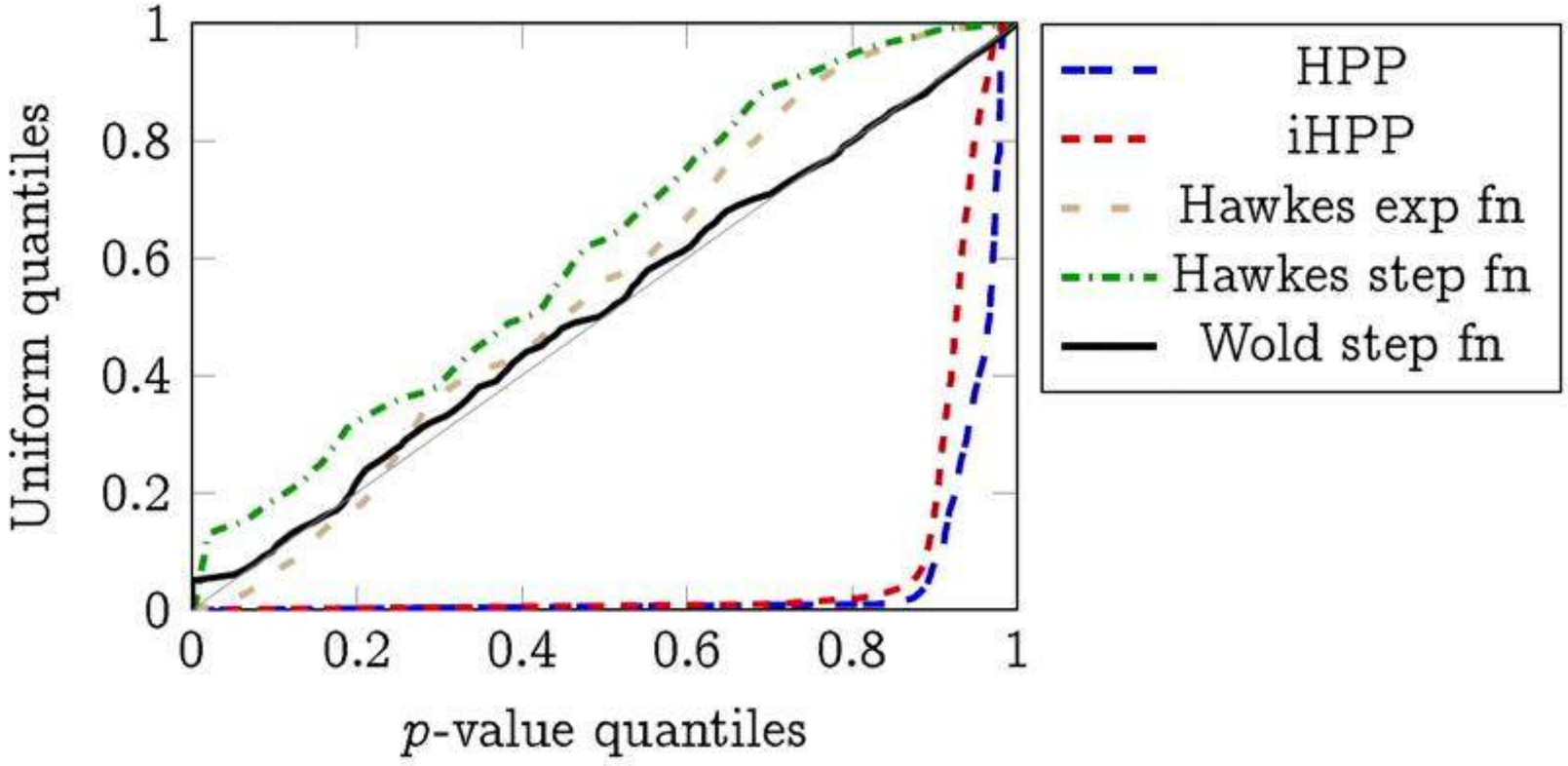
We model arrivals of event times $y_1, y_2, \ldots$ as a Wold process with self-exciting conditional intensity

$$\lambda_Y(t) = \lambda + \sum_{j=1}^{\ell} \lambda_j \mathbb{I}_{[\tau_{j-1},\tau_j)}(t - y_{Y(t)})$$



Advantages:

- Flexible changepoint model for excitation function provides consistent estimator
- Capturing *burstiness* negates the need to model seasonality, which has complex variations day-on-day



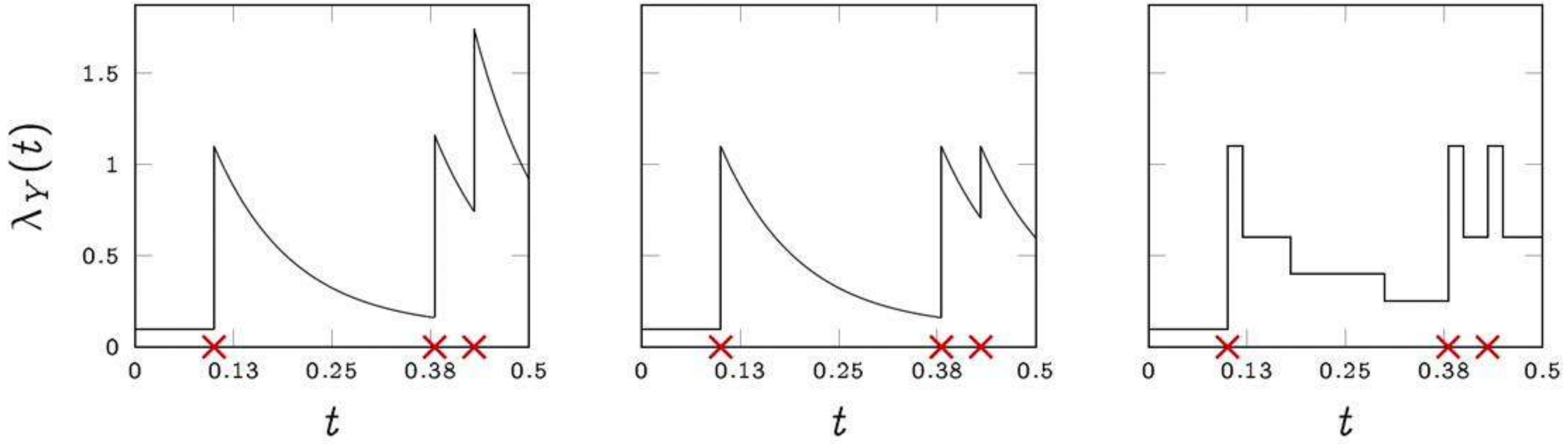
The idea has been extended to node-based modelling of all outgoing edges, such that events on one edge from a node can trigger events on its other edges

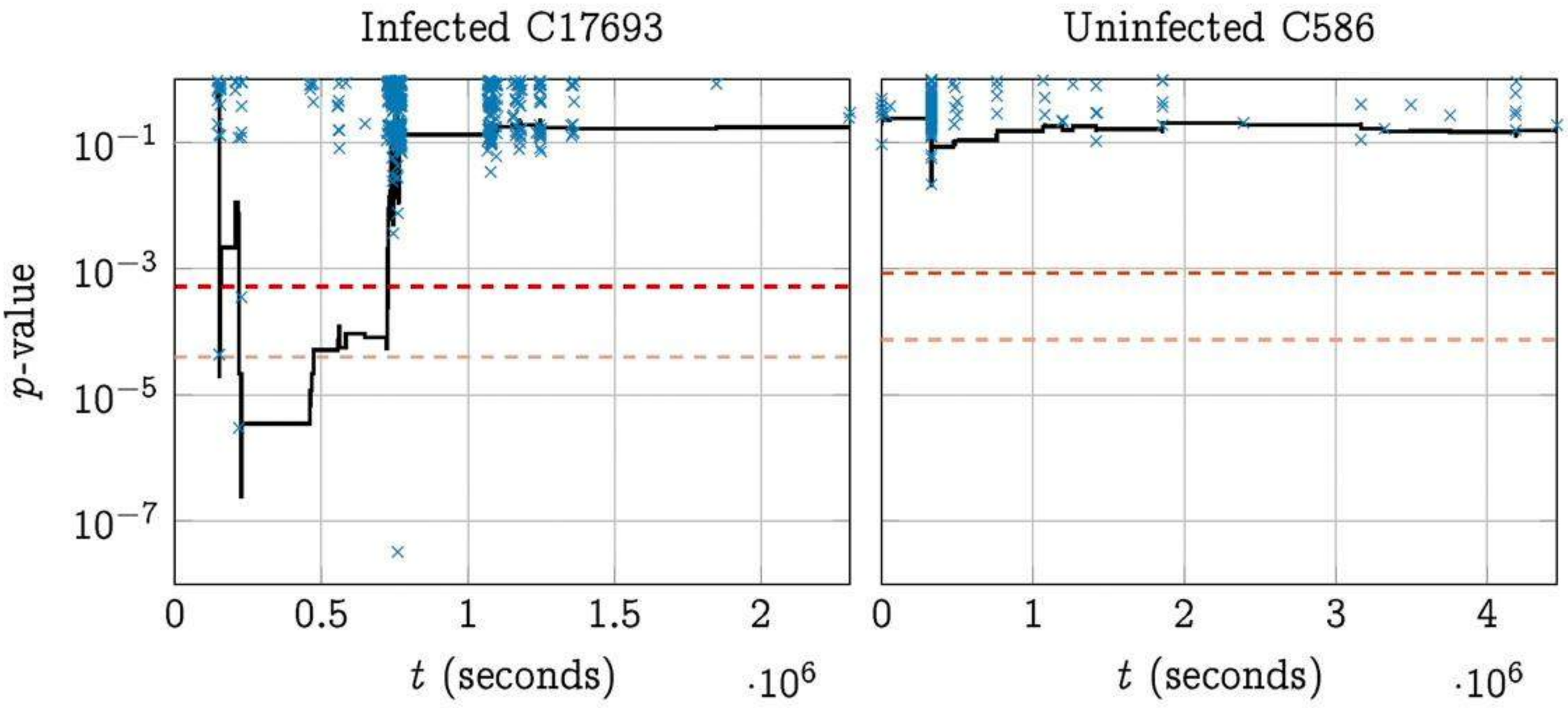
Modelling bursts of events (with M. Price-Williams)

Even human-generated network connections do not arrive as an (inhomogeneous) Poisson process. They occur in bursts, on the same edge and between edges.

We model arrivals of event times $y_1, y_2, \ldots$ as a Wold process with self-exciting conditional intensity

$$\lambda_Y(t) = \lambda + \sum_{j=1}^{\ell} \lambda_j \mathbb{I}_{[\tau_{j-1},\tau_j)}(t - y_{Y(t)})$$





Control chart thresholds at the 1% (- - -) and 0.1% (- - -) significance levels.

$\widehat{A}_k$ predicts new edges, and identifies anomalous edges

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Poisson factorisation (with M. Turcotte, F. Sanna Passino)

Alternatively, A_t can be formulated as a matrix of counts. For example, $(A_t)_{ij} = N_{ij}(t)$, the number of connections from user i to computer/process j by time t.
These counts can be regarded as preferential scores, analogous to problems in recommender systems.

In Turcotte et al., 2016 we considered a Poisson factorisation model

$$A_{ij} \sim \text{Poisson}(u_i \cdot v_j^\intercal)$$

- $u_{i\ell} \sim \Gamma(a, \xi_i)$, $v_{j\ell} \sim \Gamma(a, \eta_j)$ $(\ell = 1, \ldots, k)$
- $\xi_i, \eta_j \sim \Gamma(.5, .01)$

A p-value for anomaly detection was given by the estimated upper tail probability of the observed count; combined for each client using Fisher's method to detect red team.

Current work incorporating known groupings of computers/users with unknown latent factors.