A Model for Temporal Dependencies in Event Streams

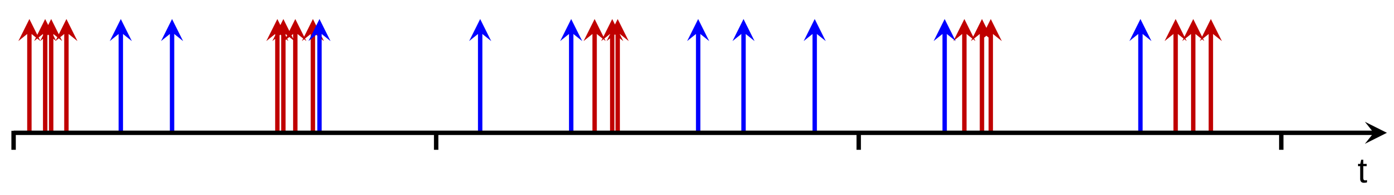
Asela Gunawardana, Christopher Meek, and Puyang Xu

Microsoft Research

Data

Event (t, l) has a **time** t and **label** l
Event sequence $x = (t_1, l_1), \ldots, (t_n, l_n)$
E.g. search queries:
(Dec-12-2011-09:42, "NIPS schedule"),
(Dec-12-2011-09:44, "NIPS program"),
(Dec-12-2011-13:02, "Granada restaurants")

Simplified example: $l \in \{■, ■\}$



Conditional Intensity Models (CIMs)

Model events forward in time:

$$p(x) = \prod_{i=1}^{n} p(t_i, l_i | h_i)$$

where the **history** $h_i = h_i(x) = (t_1, l_1), \ldots, (t_{i-1}, l_{i-1})$

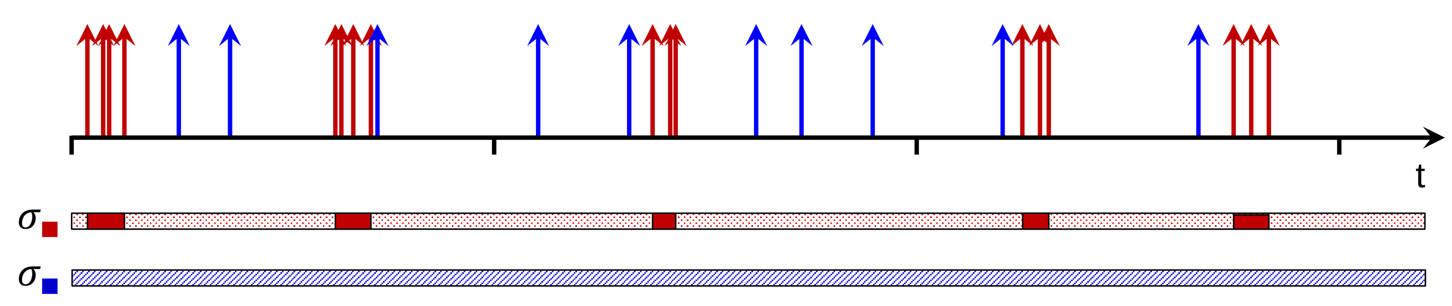
Factorizing as $p(t_i|h_i)p(l_i|h_i, t_i)$ or $p(l_i|h_i)p(t_i|h_i, l_i)$ complicated even for simplified example

Conditional Intensity Model:

$$p(t_i, l_i | h_i) = \lambda_{l_i}(t_i|h_i) e^{-\Lambda_{l_i}(t_i|h)} \prod_{l \neq l_i} e^{-\Lambda_l(t_i|h)} = \prod_l \lambda_l(t_i|h_i)^{\mathbb{I}(l=l_i)} e^{-\Lambda_l(t_i|h)}$$

Conditional intensity $\lambda_l(t|h) \geq 0$, 0 for t before end of h, $\quad \Lambda_l(t|h) = \int_{-\infty}^{t} \lambda_l(\tau|h) d\tau$

Piecewise-Constant CIMs (PCIMs)



PCIMs restrict $\lambda_l(t|h)$ to be piecewise constant.

Structure $S = \{\sigma_l(t, h)\}$, $\sigma_l(t, h)$ is a **state function** (coloring of the t-axis) for each label l
Parameters $\theta = \{\lambda_{ls}\}$

λ_{ls} is an intensity parameter for each label l and state (color) s
Likelihood:

$$p(x|S, \theta) = \prod_{l,s} \lambda_{ls}^{c_{ls}(x)} e^{-\lambda_{ls} d_{ls}(x)}$$

$c_{ls}(x)$ is the count of label l when $\sigma_l(t, h) = s$ in event sequence x
$d_{ls}(x)$ is the total duration when $\sigma_l(t, h) = s$ in event sequence x

Parameter Learning

Product of Gammas is conjugate prior, even though the likelihood isn't a product of exponentials!!!

$$p(\lambda_{ls}|\alpha_{ls}, \beta_{ls}) = \frac{\beta_{ls}}{\Gamma(\alpha_{ls})} \lambda_{ls}^{\alpha_{ls}-1} e^{-\beta_{ls}\lambda_{ls}}$$

Closed form for posterior:

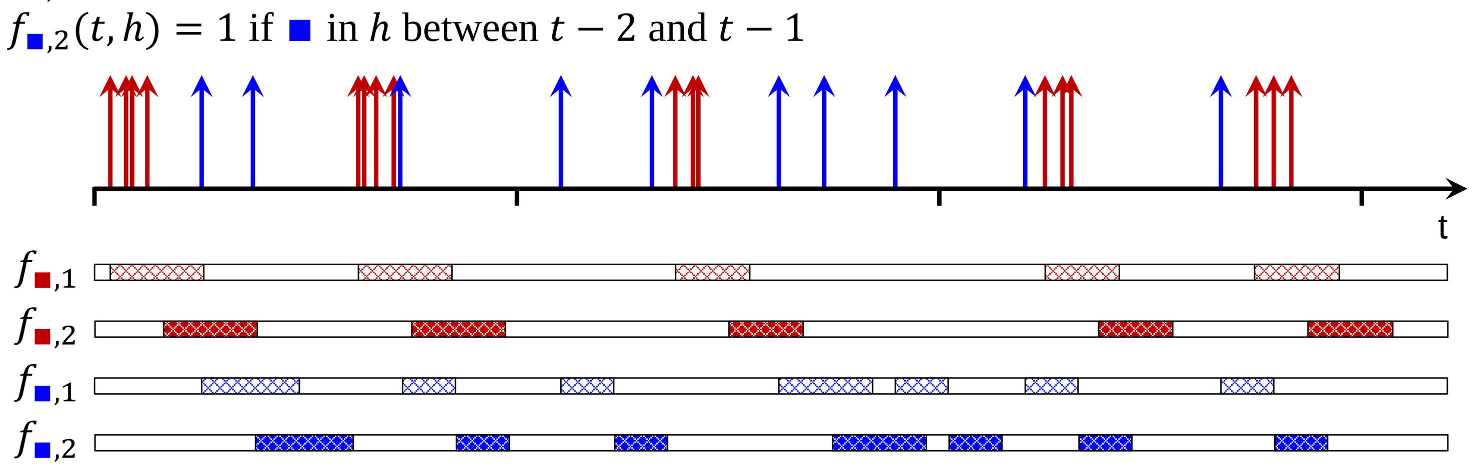
$$p(\lambda_{ls}|\alpha_{ls}, \beta_{ls}, x, S) = p(\lambda_{ls}|\alpha_{ls} + c_{ls}(x), \beta_{ls} + d_{ls}(x))$$

Closed form marginal likelihood:

$$p(x|S) = \prod_{ls} \gamma_{ls}(x) \qquad \gamma_{ls}(x) = \frac{\beta_{ls}^{\alpha_{ls}}}{\Gamma(\alpha_{ls})} \frac{\Gamma(\alpha_{ls} + c_{ls}(x))}{(\beta_{ls} + d_{ls}(x))^{\alpha_{ls} + c_{ls}(x)}}$$

Structure Learning

Start with a set $\mathcal{B}$ of **basis state functions** $f(t, h)$
E.g.:
$f_{■,1}(t, h) = 1$ if ■ in h between $t-1$ and t
$f_{■,2}(t, h) = 1$ if ■ in h between $t-2$ and $t-1$

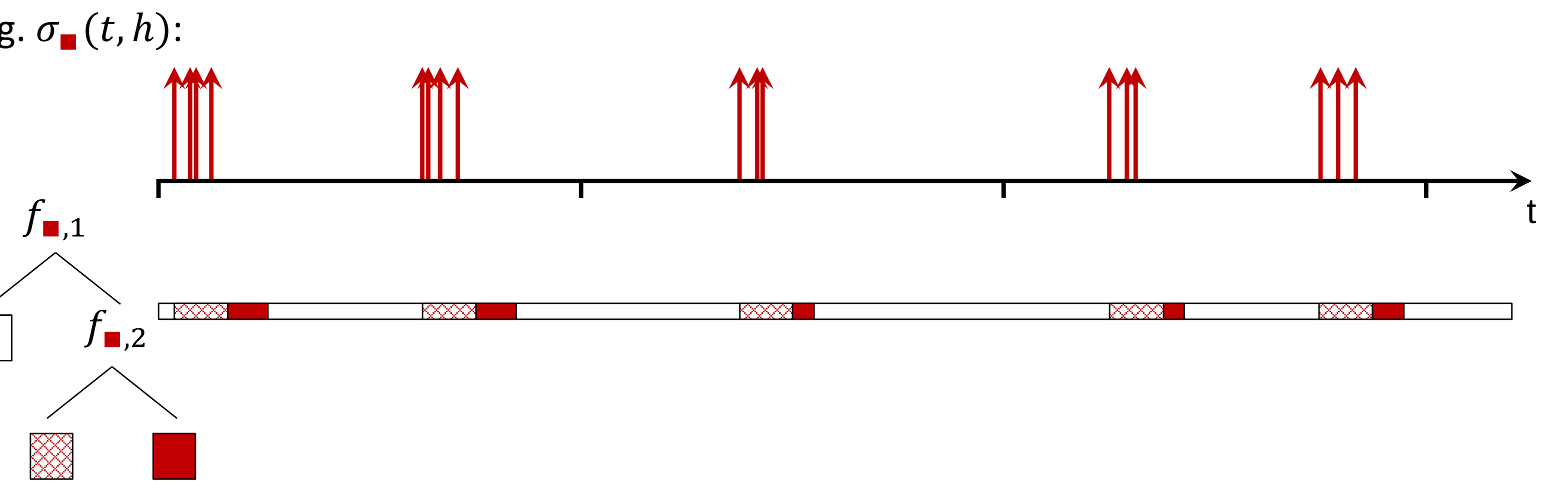


To learn $\sigma_l(t, h)$,
Start with $\sigma_l(t, h) = s_0$
Recursively select a state s and $f \in \mathcal{B}$, and **split** s using f:

$$\sigma_l'(t, h) = \begin{cases} s \wedge f(t, h) & \text{if } \sigma_l(t, h) = s \\ \sigma_l(t, h) & \text{o/w} \end{cases}$$

Yields a decision tree.

E.g. $\sigma_■(t, h)$:



Score splits using marginal likelihood.
Select state s and basis state function f to split on greedily.

Forecasting

Want estimate $\hat{p}$ of probability of a desired future outcome given the history h.
Forward sampling:
1. Sample $x^{(1)}, \ldots, x^{(M)} \sim p(x | h_{|h|-1}(x) = h)$
2. Set $\hat{p} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}(x^{(m)} \text{matches desired outcome})$

Problem: If desired outcome is rare, almost all samples will not match.

Importance Sampling:
1. Choose **proposal distribution** q
2. Sample $x^{(1)}, \ldots, x^{(M)} \sim q(x | h_{|h|-1}(x) = h)$
3. Set

$$\hat{p} = \frac{1}{\sum_{m=1}^{M} w^{(m)}} \sum_{m=1}^{M} w^{(m)} \mathbb{I}(x^{(m)} \text{matches desired outcome}), \quad w^{(m)} = \frac{p(x^{(m)} | h_{|h|-1}(x) = h)}{q(x^{(m)} | h_{|h|-1}(x) = h)}$$

To forecast probability of l^* happening between times t_a and t_b given history h, choose q to be a PCIM with cond. intensities $\lambda_l(t|h) + \lambda_l^*(t|h)$ with

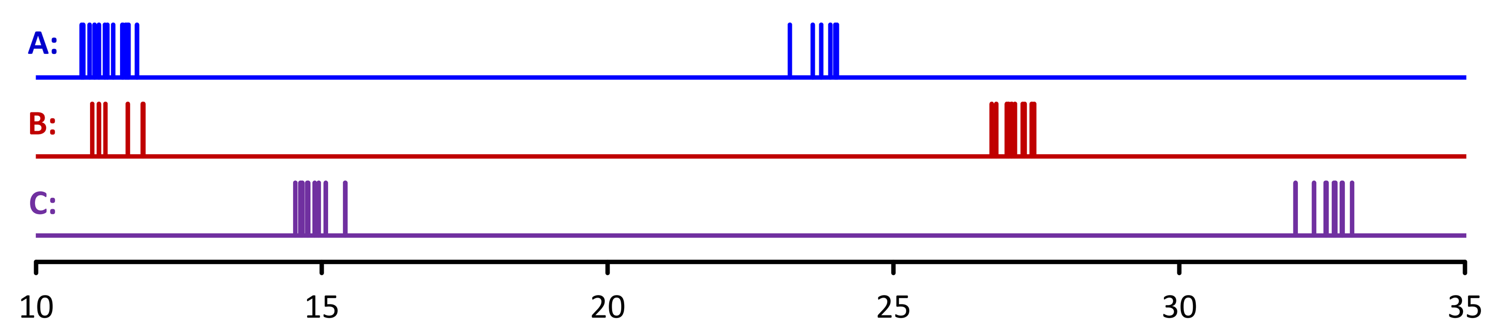
$$\lambda_l^*(t|h) = \begin{cases} \frac{1}{t_b - t_a} & \text{for } l = l^*, t \in [t_a, t_b], \text{and no } l^* \text{ in } h \text{ between } t_a \text{ and } t_b \\ 0 & \text{otherwise} \end{cases}$$

See paper for extension to more general outcomes.

Experimental Results

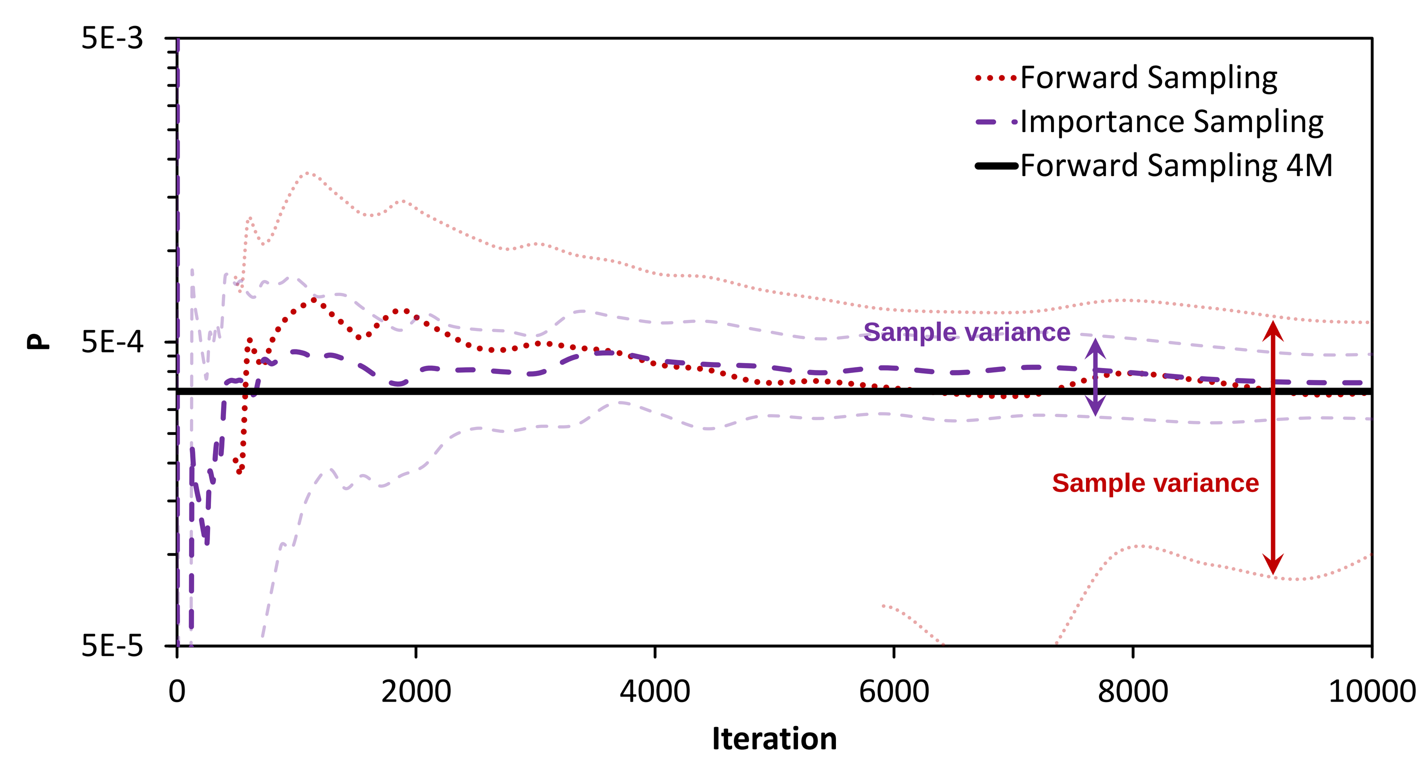
Synthetic Data

Generated data from known synthetic model:



Learned PCIM recovers structure perfectly.
Also recovers structure from more complex synthetic data (see paper).

Compare convergence of forecasting algorithms:



Supercomputer Event Logs

Learn generative models of event logs from the BlueGene/L supercomputer at Lawrence Livermore Labs. 38 alert types.

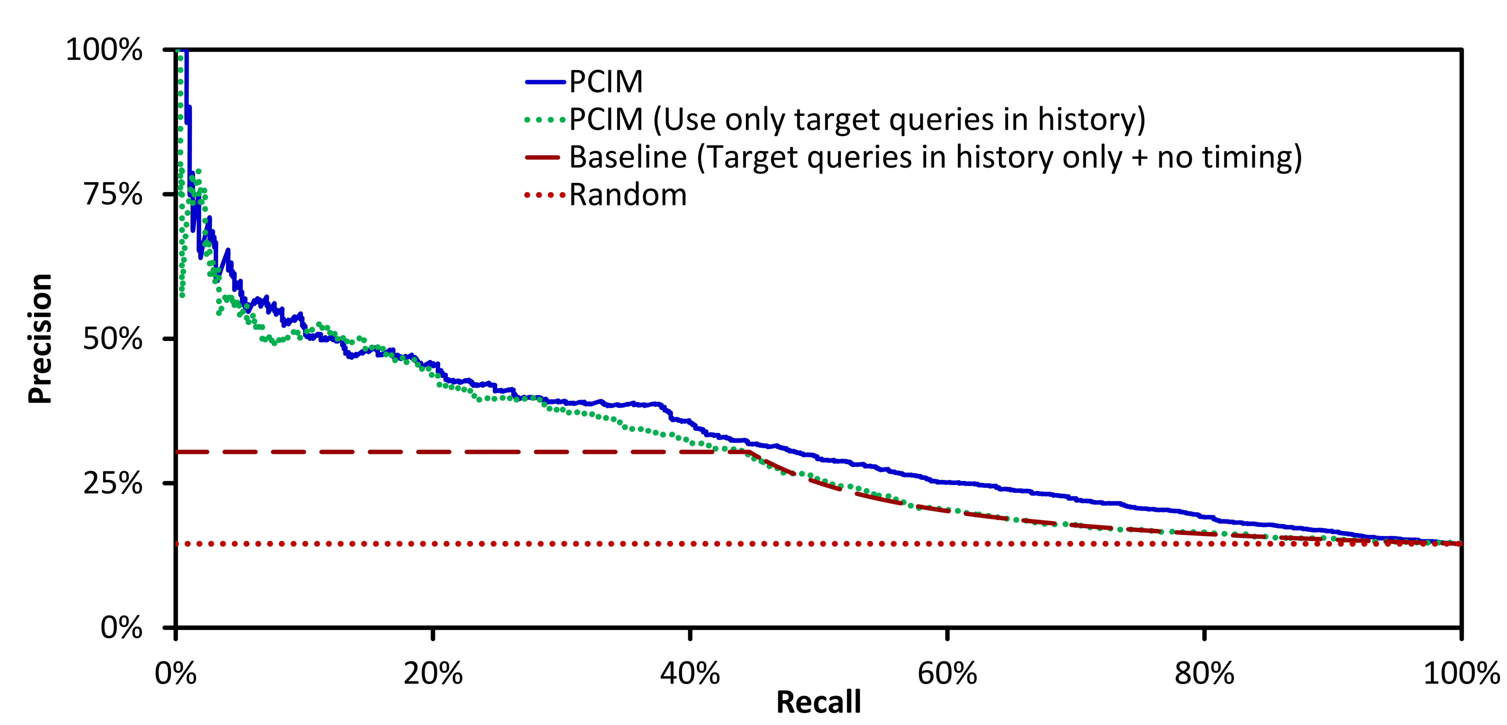
Training set: 311k alerts from 22k nodes | Test set: 69k alerts from 9k nodes

	Test Set Log Likelihood	Training Time
PCIM	-803k	11 min
Poisson Net (GLM)	-836k	3 hr 33 min

Web Search Query Logs

Training set: 385k web search queries from 23k users over 2 months.
Test set: 160k web search queries from 11k users over next month.
Queries from 36 categories.

Train PCIM, forecast whether test users will issue a query from Health & Wellness category in week 2 given their queries in week 1.



Other target queries and time ranges give similar results.

Related Work

CTBNs [Nodelman et al] model state trajectories of discrete random variables over time.

Poisson Nets [Rajaram et al, Truccolo et al] are CIMs with generalized linear models instead of decision trees. Rajaram et al use a Bayesian approach, Truccolo et al use L1 regularization.

CT-NOR & **Poisson Cascades** [Simma et al] are CIMs that assume parametric form for dependencies, don't address model selection, model only excitatory effects.