Efficient Inference Algorithms for Network Activities

Trần Quốc Long

Khoa Công nghệ thông tin
Đại học Công nghệ
Đại học Quốc gia Hà Nội

27 tháng Hai, 2015
Event attribution is an important missing data problem in event modeling based on temporal point processes. In a real-world scenario, a number of distinct conflict cascades can simultaneously diffuse, and we assume that the occurrence of one conflict tends to increase the probability of another conflict, i.e., the sequence shows chain reactions where an event tends to involve several actors, e.g., a timestamped event recording Taliban attacking civilians.

Efficient Inference Algorithms

Long

Introduction

Community

Interval-censored

Click-to-Conversion

Conclusion

Armed Conflict Location and Event Data

36.3% dyadic events in the Afghan dataset are without the actor information — e.g., an event with civilian casualty is observed but we did not observe who carried out the act.
Information manipulation in social media: paid to, e.g., spread grey or illegal content: defamatory rumors and false advertising ...

Entertainment sector: upticks in activity closely correspond to promotion campaigns for recent or up-coming releases of movies or TV programs
Management and maintenance of aging infrastructures

- 700K underground water pipes of a large city: preventative rehabilitation and replacement are the key activities for pipe asset management
- Understanding of the failure mechanism in *repairable* pipes and modeling the stochastic behavior of the recurrences of pipe failure
New point process models

- Community structure and network influence.
- Interval-censored data.
- Click-to-Conversion in advertisement.
- Tailored variational inference under MM framework.
Temporal Event Modeling

- Temporal events: occurrences of events over time

\[ t_1 < t_2 < \cdots < t_n \]

usually recorded over an observation window \([0, T]\)

- A variety of applications
  - infection and spread of contagious diseases
  - earthquakes (magnitudes and locations): spatial temporal event
  - sequence of retweets in Tweeter
  - sequence of user queries submitted to a search engine
  - posts and comments within a user group in a social media site
Temporal Event Modeling

• **Goals** of temporal event modeling
  — studying the mechanisms that give rise to the recurrence of events
  — predicting how the events will behave in the future
  — designing intervention measures to modify the dynamic processes of event recurrences

• **Network** temporal events
  — Events are associated with the identity of users/nodes that generate the events.
  — The effect of *network association* (i.e. influence between pairs, influence of community) on the recurrence of events.
Hawkes Processes

Hawkes process given events at \( \{t_1, t_2, \ldots \} \)

- Intensity function (rate of new events) depends on history.

\[
\lambda(t) = \mu + \alpha \sum_{t_i < t} \exp(-\omega(t - t_i))
\]

- Likelihood for \( \{t_1, t_2, \ldots, t_n\} \) over observation window \([0, T]\)

\[
L(\{t_1, t_2, \ldots, t_n\}) = \left( \prod_{i=1}^{n} \lambda(t_i) \right) \exp \left( - \int_{0}^{T} \lambda(t) \, ds \right)
\]
Multi-dimensional Hawkes Process

Definition

- One Hawkes process for all the events generated by one user
- The influence between users are explicitly modeled by coupling the individual Hawkes processes:

  \[ \lambda_i(t) = \mu_i + \sum_{i: t_{\ell} < t} \alpha_{ii\ell} \kappa(t - t_{\ell}) \]

- \( \alpha_{ij} \geq 0 \) captures the degree of influence of \( j \) on \( i \)
  - \( F = [\alpha_{ij}] \) encoding diffusion network structure (infectivity).
- \( \mu_i \) background rate of occurrence.
- \( \kappa(t) \) a decay function, e.g, \( \kappa(t) = \omega \exp(-\omega t) \)
Multi-dimensional Hawkes Process
Parameter Estimation

- Given \( m \) samples/cascades \( \{c_1, \ldots, c_m\} \), from the user population

\[
c_i = \{(t_{1i}^c, i_{1i}^c), \ldots, (t_{ni}^c, i_{ni}^c)\}, \quad t_{\ell}^c \in [0, T_c]
\]

where \( (t_{\ell}^c, i_{\ell}^c) \): event occurring at the \( i_{\ell}^c \)-th dimension at time \( t_{\ell}^c \).

- The likelihood with \( K(t) \equiv \int_0^t \kappa(s)ds \),

\[
\mathcal{L}(\mathbf{F}, \mathbf{\mu}) = \sum_c \left( \sum_{n=1}^{n_c} \log \left( \mu_{i_n^c} + \sum_{t_{\ell}^c < t} \alpha_{i_n^c i_{\ell}^c} \kappa(t_n^c - t_{\ell}^c) \right) \right.

\left. - T_c \sum_{i=1}^l \mu_i - \sum_{i=1}^l \sum_{n=1}^{n_c} a_{i i_n^c} K(T_c - t_n^c) \right),
\]

- Convex optimization problem:

\[
\min_{\mathbf{F} \geq 0, \mathbf{\mu} \geq 0} -\mathcal{L}(\mathbf{F}, \mathbf{\mu})
\]
Research Problems

Inference problems under various scenarios

- The network forms clusters/communities.
- Input data is interval-censored.
- Click-to-Conversion modeling in Online Advertisement.
Research Problems

Inference problems under various scenarios

- The network forms clusters/communities.
- Input data is interval-censored.
- Click-to-Conversion modeling in Online Advertisement.

There are efficient inference algorithms for these scenarios.
Community Detection

- Current interests in Community detection
  - Network infectivity/influence inference
  - Clustering and community detection from static links.
  - Regularization approach

- We directly model the interaction between group participation and network infectivity.

- Using variational inference framework, we propose an efficient community detection algorithm with closed-form and parallel updates.

Problem statement: Given the timestamped events from $I$ users in the form of a set of cascades $t = \{(t^c_n, i^c_n)\}$, $n = 1 \ldots N_c$, $c = 1 \ldots C$, find a coherent clustering of the users.
Naively, one needs $O(l^2)$ parameters to model interactions or infectivity rate among $l$ users/nodes.

In reality, the infectivity matrix is sparse as most pairs of users/nodes has no interaction.

Furthermore, groups of users/nodes having similar responses to other groups leads to a low rank structure on the infectivity matrix.
Sparse and Low-rank Network Infectivity
Restricting solution space

- **Zhou et. al (2013):**
  - Enforcing low-rank via nuclear norm minimization.
  - Enforcing sparsity via $\ell_1$ norm minimization.
  - Disadvantage: a dense matrix is used in optimization.

$$
\min_{F \geq 0, \mu \geq 0} -\mathcal{L}(F, \mu) + \lambda_1 \|F\|_* + \lambda_2 \|F\|_1.
$$

- We proposed direct modeling of interaction between group participation and network infectivity
  - Introducing latent variable $Z_i$ for group participation of $i$-th user/node.
  - Explicit low-rank assumption.
  - Reducing number of parameters.
We propose the decomposition of infectivity into group participation $Z$ and celebrity index $\beta$.

$$\alpha_{ij} = \beta_j \langle Z_i, Z_j \rangle = \beta_j \sum_{g=1}^{G} z_{ig} z_{jg}, i \neq j,$$

$$F - \text{diag}(F) = ZZ^T \text{diag}(\beta) - \text{diag}(ZZ^T \text{diag}(\beta)).$$

- Self-exciting parameter $\alpha_i$ is not decomposed.
- Decomposition of intensity function

$$\lambda_i(t) = \mu_i + \sum_{i \neq i} \sum_{t \leq t} \sum_{g=1}^{G} \beta_{ig} z_{ig} z_{i_g} \kappa(t - t_i) + \alpha_i \sum_{t \leq t} \kappa(t - t_i).$$
Modeling

Likelihood function

- Conjugate prior on $Z$: Gamma distribution
  \[ z_{ig} \sim \text{Gamma}(a_g^0, b_g^0) \]

- Log-likelihood of a set of cascades $t = \{(t^c_n, i^c_n)\}$, $n = 1 \ldots N_c$, $c = 1 \ldots C$
  \[
  \mathcal{L}(Z, t) = \text{const} + \sum_{i=1}^{l} \sum_{g=1}^{G} (a_g^0 - 1) \ln z_{ig} - b_g^0 z_{ig} \\
  + \sum_{c=1}^{C} \sum_{n=1}^{N_c} \ln \left( \mu_{i_n} + \sum_{\ell < n} \alpha_{i_n \ell} \kappa(t^c_n - t^c_\ell) + \alpha_{i_n} \sum_{\ell < n} \kappa(t^c_n - t^c_\ell) \right) \\
  - \sum_{c=1}^{C} \sum_{i=1}^{l} \int_{0}^{T_c} \left[ \mu_i + \sum_{t_n < t} \alpha_{i_n} \kappa(t - t^c_n) + \alpha_i \sum_{t_n < t} \kappa(t - t^c_n) \right] dt,
  \]
Variational Inference
Mean-field approximation

- No easy factorization of posterior $P(Z|t)$.
- Log-likelihood decomposition, for all distribution $q$

$$
\ln P(t) = \ln \sum_{Z} L(Z, t) = E_q [L(Z, t)] + \mathcal{E}[q] + KL(q \| P(Z|t)).
$$

- Remarkably, one only needs the mean-field assumption on the approximation distribution.

$$
q(Z) = \prod_{i=1}^{l} q_i(Z_i).
$$
 Variational Inference  
Evidence lower bound (ELBO)

**Theorem**

The expectation $E_Q [\mathcal{L}(\mathbf{Z}, \mathbf{t})]$ is lower-bounded by

$$
\sum_{i=1}^{I} \sum_{g=1}^{G} (a_g^0 - 1) E_Q [\ln z_{ig}] - b_g^0 E_Q [z_{ig}] + \sum_{c=1}^{C} \sum_{n=1}^{N_c} \eta_n^c \ln \mu_n^c + \gamma_n^c \ln \left( \alpha_n^c \sum_{\ell < n, \ell_c \neq i_n} \kappa(t_n^c - t_{\ell}^c) \right) 
$$

$$
+ \sum_{\ell < n} \sum_{g=1}^{G} \eta_{\ell n}^{gc} E_Q \left[ \ln(\beta_{\ell}^{gc} z_{\ell n}^{gc} z_{\ell c}^{gc} \kappa(t_n^c - t_{\ell}^c)) \right] - \eta_n^c \ln \eta_n^c - \sum_{\ell < n} \sum_{g=1}^{G} \gamma_{\ell n}^{gc} \ln \gamma_{\ell n}^{gc} - \gamma_n^c \ln \gamma_n^c 
$$

$$
- \sum_{c=1}^{C} \sum_{i=1}^{I} \mu_i T_c - \sum_{c=1}^{C} \sum_{i=1}^{I} \sum_{n=1}^{N_c} \sum_{g=1}^{G} \beta_{\ell}^{gc} E_Q \left[ z_{ig} z_{n}^{gc} \kappa(T_c - t_n^c) \right] K(T_c - t_n^c) - \sum_{c=1}^{C} \sum_{i=1}^{I} \sum_{n=1}^{N_c} \alpha_i K(T_c - t_n^c),
$$

for all positive $\eta_n^c, \eta_{\ell n}^{gc}, \gamma_n^c$ such that $\eta_n^c + \sum_{\ell < n, \ell_c \neq i_n} \eta_{\ell n}^{gc} + \gamma_n^c = 1$. 
Variational Inference
Proposed algorithm

- From standard variational inference framework
  \[ \ln q_i^*(\mathbf{Z}_i) = \mathbb{E}_{q-Z_i}[\mathcal{L}(\mathbf{Z}, \mathbf{t})] + \text{const}. \]

- **Gamma distribution** for latent variables \( z_{ig} \sim \text{Gamma}(a_{ig}, b_{ig}). \)
- **Closed-form update** for auxiliary variables \( \eta_n^c, \eta_{\iota n}^g, \gamma_n^c. \)
- **Closed-form update** for individual parameters \( \mu_i, \alpha_i, \beta_i. \)

---

**Algorithm 1** Network Inference and Community Detection (*NetCodec*)

**INPUT:** Set of cascades \( \{(t_n^c, i_n^c)_{n=1...N_c}\}_{c=1...C} \);
**OUTPUT:** \( \mu, \alpha, \beta, \mathbf{Z} = \text{Gamma}(\mathbf{A}, \mathbf{B}); \)
**Initialization:** \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{l \times G}, \mu, \alpha, \beta \in \mathbb{R}^l \)

**while** not converged **do**
  **for** all users/nodes \( i \) **do**
    Update \( i \)-th row of \( \mathbf{A} \) and \( \mathbf{B} \);
    Update auxiliary variables;
  **end for**
  Update \( \mu, \alpha, \) and \( \beta \);
**end while**
Variational Inference

Update formulas

\[
\begin{align*}
a_{ig} &= a_g^0 + \sum_{c=1}^C \sum_{n=1}^{N_c} \sum_{\ell < n} \eta_{\ell n}^{gc} \delta_{\ell n}^c, \\
b_{ig} &= b_g^0 + \sum_{c=1}^C \left[ \sum_{n=1}^{N_c} \beta_{\ell n}^c \mathbb{E}_{q} \left[ z_{n g}^c \right] K(T_c - t_n^c) + \sum_{j \neq i} \sum_{n=1}^{N_c} \beta_i \mathbb{E}_{q} \left[ z_{j g} \right] K(T_c - t_n^c) \right], \\
\eta_{n}^c &\propto \mu_{\bar{r}_n^c}, \quad \eta_{\ell n}^{gc} \propto \beta_{\ell n}^c \kappa(t_n^c - t_{\ell}^c) \mathbb{E}_{q} \left[ \ln z_{\bar{r}_n^c g}^c \right] + \mathbb{E}_{q} \left[ \ln z_{\bar{r}_\ell n}^c g \right], \\
\gamma_{n}^c &\propto \alpha_{\bar{r}_n^c} \sum_{\ell < n, \bar{r}_\ell n = \bar{r}_n^c} \kappa(t_n^c - t_{\ell}^c), \\
\mu_i &= \frac{\sum_{c=1}^C \sum_{n=1}^{N_c} \sum_{\bar{r}_n^c = i} \eta_{n}^c}{\sum_{c=1}^C T_c}, \quad \alpha_i = \frac{\sum_{c=1}^C \sum_{n=1}^{N_c} \sum_{\bar{r}_n^c = i} \gamma_{n}^c}{\sum_{c=1}^C \sum_{n=1}^{N_c} \sum_{\bar{r}_n^c = i} K(T_c - t_n^c)}, \\
\beta_i &= \frac{\sum_{c=1}^C \sum_{n=1}^{N_c} \sum_{\bar{r}_n^c \neq i} \sum_{\ell < n, \bar{r}_\ell n = i} \sum_{g=1}^G \eta_{\ell n}^{gc}}{\sum_{c=1}^C \sum_{j \neq i} \sum_{n=1}^{N_c} \sum_{\bar{r}_n^c = i} \sum_{g=1}^G \mathbb{E}_{q} \left[ z_{j g} z_{i g} \right] K(T_c - t_n^c)}. 
\end{align*}
\]
Variational Inference
Implementation

- **Stopping criteria**: we used relative change in the ELBO
  \[
  \frac{|\text{ELBO}^{new} - \text{ELBO}^{old}|}{|\text{ELBO}^{old}|} < 10^{-4}.
  \]

  In our experience, the algorithm stops before 40 iterations.

- **Data sweeps**
  — Naive implementation need one data sweep per user.
  — The ELBO holds for any set of auxiliary variables.
  — Careful book keeping reduces to one sweep for all users.

- **Relevant history**
  — Exponent kernel leads to a small number of relevant events.
  — Complexity $O((N + I)G)$ per iteration.

- **Parallelization**
  — All update formulas could be accumulated in parallel with respect to the events.
Experimental results

Datasets

- **Synthetic data:**
  - 500 nodes, both cross-group and core-group scenarios.
  - Time window $T = 10^6$, #events $N = 3 \times 10^5$.

- **MemeTracker data** (www.memetracker.org)
  - Site activities with links, select most active 500 sites.

- **Earthquake data** (earthquake.usgs.gov)
  - 16000 earthquakes of minimum magnitude 4 all over the world for 1 year.
Experimental results
Performance evaluation

- Average of 10 random runs

- Normalized Mutual Information

\[
\text{NMI}(\Omega, \Gamma) = \frac{\sum_k \sum_j \Pr\{\Omega_k \cap \Gamma_j\} \log \frac{\Pr\{\Omega_k \cap \Gamma_j\}}{\Pr\{\Omega_k\}\Pr\{\Gamma_j\}}}{(E[\Omega] + E[\Gamma])/2},
\]

- Kendal Rank Correlation

\[
\text{RankCorr}(x, y) = \frac{\text{concordant pairs} - \text{discordant pairs}}{l \times (l - 1)/2},
\]

- Relative Error

\[
\text{RelErr}(F_1, F_2) = \frac{1}{p^2} \sum_{i,j=1}^{l} \left| \frac{\alpha_{ij}^2 - \alpha_{ij}^1}{\alpha_{ij}^1} \right|
\]

- Predictive Likelihood
Experimental results
Synthetic data

(a) RankCorr  (b) RelErr  (c) PredLik  (d) NMI

**Figure:** Cross-group scenario

(a) RankCorr  (b) RelErr  (c) PredLik

**Figure:** Core group scenario.
Experimental results
Memetracker

(a) NMI (10 clusters)
(b) NMI (20 clusters)
(c) 20 clusters (NetCodec)

Figure: Clustering results on MemeTracker dataset.
Figure: Clustering results on Earthquake dataset.
Due to limitation in resources, technologies and user privacy, time-stamped data are not available in many situations.

- New links between 2 crawls of websites.
- Aggregated statistics in epidemic disease.

For a network of $U$ nodes, the interval-censored data is a set $K$ of intervals and number of events in each interval.

- $C = \{(a_i, b_i, c_i)\}$, $i = 1, \ldots, K$.
- $a_i, b_i$ is the start time and end time of the $i$-th interval.
- $c_i = [c_{iu}] \in \mathbb{N}^U$, $c_{iu}$ is the number of events at $u$-th node in $[a_i, b_i]$.
- $c_i$ is usually very sparse.

Could we still infer relationship/influence among the nodes?
Prior works focused on Poisson processes for interval-censored data

- Parametric Poisson process (Streit. 2000, Fan. 2009).
- Non-homogeneous Poisson process with pseudo MLE (Sun et. al. 1995).
- Full MLE (Wellner et. al 2000).

Memoryless property of Poisson processes: number of events in new intervals are independent of previous intervals.
Given interval censored data \( C = \{(a_i, b_i, c_i)\}, i = 1, \ldots K \), maximize the likelihood

\[
\Theta^{ic} = \arg \max_{\Theta} \mathcal{L}^{ic}(\Theta) \triangleq \log P(C; \Theta).
\]

\[
\mathcal{L}^{ic}(\Theta) = \log \sum_u \int_t P(t, u, C; \Theta) dt,
\]

where \( P(t, u, C; \Theta) \) is the likelihood of the Hawkes process on full data restricted to satisfy the aggregated counts in \( C \).
Monte-Carlo EM

Considering the time-stamps and the identity of events hidden

- **E-step**: find the posterior distribution

\[ P^{(k)} = P(t, u | C; \Theta^{(k)}) \]

- **M-step**:

\[
\max_{\Theta \geq 0} Q(\Theta; \Theta^{(k)}) \triangleq E_{P^{(k)}} \log P(t, u, C; \Theta).
\]

- The expectation maxization is hard even if one knows a closed form posterior distribution.
- Substitute expectation with average from simulated samples

\[
\max_{\Theta \geq 0} Q^{MC}(\Theta; \Theta^{(k)}) \triangleq \frac{1}{S} \sum_{s=1}^{S} \log P(t^s, u^s, C; \Theta).
\]

- This is the original MLE problem for Hawkes processes.
Proposed algorithms

Algorithm 2 Parameter estimation via imputation

Input: $a_i, b_i, c_{iu}, i = 1 \ldots K, u = 1 \ldots U$;
Initialize $\Theta^{(1)} = (\mu^{(1)}, A^{(1)})$;
for $k = 1, 2, \ldots$ do
    Impute $t, u$ satisfying the counts $C$
    Re-estimate $\Theta^{(k+1)}$ with MLE solver;
end for
Sampling methods

Gibbs sampling

- Sample 1 event given other event fixed.
- Use Metropolis-Hastings for sampling each event.
- $P(t, u|C; \Theta)$ is not known but one can compute $P(t, u, C; \Theta)$

$$P(t, u, C; \Theta) = P(t, u|C; \Theta) \times \text{const.}$$
Sampling methods
Baseline heuristics

- **MID**: use mid-points.
- **EQUAL**: distribute events evenly.
- **RAND**: distribute events randomly.
- **INTSIM**: distribute events randomly according to intensity function.
Sampling quality

Using random time change theorem to convert duration to exponential random variables

Figure: Comparison of MLE solvers
Experiment results

Settings

- **Synthetic data:**
  - $A \in \mathbb{R}^{10 \times 10}$ random with spectral radius 0.8.
  - 100 cascades with time window $T = 100$.

- **Performance evaluation using relative error**

$$\text{RelErr}(F_1, F_2) = \frac{1}{p^2} \sum_{i,j=1}^{l} \left| \frac{\alpha_{ij}^2 - \alpha_{ij}^1}{\alpha_{ij}^1} \right|$$

- **Baseline method:** imputation with midpoints of intervals.
Experiment results

Interval size

Figure: Effect of interval size on the estimation quality.
Experiment results
Censored Intervals

Figure: Relative error with respect to the percentage of censored intervals.
Experiment results
Graph recovery

(a) Karate club’s graph

(b) ROC curves: True Positive (detected edge) vs. True Negative (detected non-edge) on the Karate graph.

Figure: Karate club dataset.
Advertisement market
Example: Sponsored search
Advertisement market
Example: Online ad
Advertisement market
Online advertisement parties

- Publishers (Google, WP)
- Advertisers (products/services)
- Users (customers)
- Impressions
  - Graphical, textual ads
  - Context, User information

Efficient Inference Algorithms
Introduction
Community
Interval-censored
Click-to-Conversion
Advertisement
Modeling
MLE
Experiments
Conclusion
Pricing models based on user activities

- **Cost-per-click (CPC):** everytime users click on the impression.
- **Cost-per-conversion (CPA):** only if users buy/adopt product.
- Market parties use *click through rate* and *conversion rate* to compute bids.
Pricing models based on user activities

- **Cost-per-click (CPC)**: everytime users click on the impression.
- **Cost-per-conversion (CPA)**: only if users buy/adopt product.
- Market parties use *click through rate* and *conversion rate* to compute bids.
Both clicks and conversions have periodic patterns.
Click-to-Conversion delay is approximately exponential.
Different campaigns have different conversion patterns.
Click-to-Conversion modeling
Thinned processes

- Conversion rate = Thinning probability = $p(x)$.
- Click-to-Conversion delay distribution = $\kappa(\delta, x)$.
- Known click-conversion association $\Rightarrow$ classification problem.
- What if click-conversion association is not known?
Click-to-Conversion modeling
Thinned processes

What if click-conversion association is not known?
One could only model the conversion mechanism in aggregate terms (e.g. how many conversions in the next hour/day/week).

We propose:
— Clicks as a Hawkes process.
— Conversions as a thinned process.
Click-to-Conversion modeling
Thinned processes

- Clicks: $\xi(t) = \xi_0 + \beta \sum_{t_n < t} \omega e^{-\omega(t-t_n)}$.
- Conversions $\lambda(\tau|t) = \mu + \alpha \sum_{t_n < \tau} p(x_n) \kappa(\tau - t_n, x_n)$.
- Parametric models

$$p(x) = \frac{1}{1 + e^{-\langle w_c, x \rangle}},$$

$$r(x) = e^{\langle w_d, x \rangle}, \kappa(\delta, x) = r(x) e^{-r(x)\delta}.$$
Efficient Inference Algorithms

Introduction

Community

Interval-censored

Click-to-Conversion

Advertisement

Modeling

MLE

Experiments

Conclusion

MLE Algorithm

\[
L_{\text{click}} = \sum_{n=1}^{N_1} \ln \xi(t_n) - \int_0^T \xi(t) \, dt
\]

\[
L_{\text{conversion}} = \sum_{\ell=1}^{N_2} \ln \lambda(\tau_{\ell} | t) - \int_0^T \lambda(\tau | t) \, d\tau
\]

Algorithm 3 MLE algorithm

Require: \( \{ t_n, x_n \}_{n=1}^{N_1}, \{ \tau_{\ell} \}_{\ell=1}^{N_2} \)

Ensure: \( \xi_0, \beta, \omega \) maximize \( L_{\text{click}} \) and \( \mu, \alpha, w_c, w_d \) maximize \( L_{\text{conversion}} \).

Solve max \( L_{\text{click}} \) with standard Hawkes MLE for \( \xi_0, \beta, \omega \).

Initialize \( \mu, \alpha, w_c, w_d \).

while change in \( L_{\text{conversion}} \) is more than tolerance do

Compute \( \eta_{\ell}^{n} \)'s.

Compute \( \mu, \alpha \).

Optimize \( w_c, w_d \) using L-BFGS.

end while
MLE
Update formulas

\[ \eta_0^\ell = \frac{\mu}{\mu + \sum_{t_n < \tau_\ell} \alpha p(x_n) \kappa(\tau_\ell - t_n, x_n)}, \]
\[ \eta_n^\ell = \frac{\alpha p(x_n) \kappa(\tau_\ell - t_n, x_n)}{\mu + \sum_{t_n' < \tau_\ell} \alpha p(x_n') \kappa(\tau_\ell - t_n', x_n')}, \]
\[ \mu = \frac{\sum_{\ell=1}^{N_2} \eta_0^\ell}{T}, \quad \alpha = \frac{\sum_{\ell=1}^{N_2} \sum_{t_n < \tau_\ell} \eta_n^\ell}{\sum_{n=1}^{N_1} p(x_n) K(T - t_n, x_n)}, \]
\[ \nabla_{w_c} L = - \sum_{\ell=1}^{N_2} \sum_{t_n < \tau_\ell} \eta_n^\ell [1 - p(x_n)] x_n \]
\[ + \sum_{n=1}^{N_1} p(x_n) [1 - p(x_n)] K(T - t_n, x_n) x_n + \gamma w_c, \]
\[ \nabla_{w_d} L = - \sum_{\ell=1}^{N_2} \sum_{t_n < \tau_\ell} \eta_n^\ell [1 - r(x_n)(\tau_\ell - t_n)] x_n \]
\[ + \alpha \sum_{n=1}^{N_1} p(x_n) e^{-r(x_n)(T - t_n)} r(x_n) (T - t_n) x_n + \gamma w_d. \]
Experimental results
Dataset

Criteo data
- Ad campaign and user activity log in real time for 2 months
- About 15 million clicks and 1 million conversions.
- About 13 thousand ad campaigns.

Experiment setup
- Model is trained on a subset of 10 consecutive days.
- Model is tested on (unseen) subsequent days.
- \(x\)-day look ahead: Only use data at least \(x\)-day old to predict conversion volume.
Experimental results
1 hour look ahead

(a) Campaign with conversion rate is 0.28: 17 test days.

(b) Another campaign with conversion rate is 0.08.

(c) Another campaign with conversion rate is 0.03.
Experimental results
1 day look ahead

(a) Campaign with conversion rate is 0.28: 17 test days.

(b) Another campaign with conversion rate is 0.08.

(c) Another campaign with conversion rate is 0.03.
Experimental results
3 day look ahead

(a) Campaign with conversion rate is 0.28: 17 test days.

(b) Another campaign with conversion rate is 0.08.

(c) Another campaign with conversion rate is 0.03.
Conclusion

New models

- Community structure and network influence.
- Interval-censored data.
- Click-to-Conversion in advertisement.
- Tailored variational inference under MM framework.
Thank you for your attention