Neural Hawkes Particle Smoothing:

Imputing Missing Events in Continuous-Time Event Streams

Hongyuan Mei¹, Guanghui Qin², Jason Eisner¹
¹Center for Language and Speech Processing, Department of Computer Science, Johns Hopkins University, USA
²Department of Physics, Peking University, China

Overview

Neural Hawkes process (NHP: Mei & Eisner, NeurIPS 2017)

\[ p_{\text{NHP}}(z|x) = p(z|x) \times p_{\text{miss}}(z) \]

- Missingness mechanism that determines missing events \( z \)
- \( p(z|x) \): What / When / How-Many missing events?

Why? Impute past to predict future; train with Monte Carlo EM

Sequential Monte Carlo

Draw \( z_1, \ldots, z_M \) from a proposal distribution \( q(z|x) \) and weight them \( w \propto p(z|x)/q(z|x) \)

Example: stochastically impute a taxi’s pick-up events \( \star \) given its observed drop-off events \( \clubsuit \).

Below shows one sequential step, which determines the next event after \( \square \) at time \( t_1 \) --- either an unobserved event at time \( t \in (t_1, t_2) \) or the next observed event at \( t_2 \).

- Particle filtering proposes next event \( \star \) conditioned only on history summarized as \( \square \) by LSTM

Minimum Bayes Risk Decoding

Define optimal transport distance \( L(z, z^*) \)

- Aligning two events in \( z \) and \( z^* \) has cost \( |t - t^*| \)
- An unaligned event in \( z \) or \( z^* \) has cost \( C \)
- Find optimal alignment \( a \) by dynamic programming

Seek \( z \) with small expected loss

\[ \sum_{m=1}^{M} w_m L(z, z_m) \]

Seeking: for each \( C \), actual improvement \( \rightarrow \) is always in the positive direction of the steepest improvement \( \rightarrow \)

Training the Proposal Distribution (only for particle smoothing)

Minimize \( \beta \text{KL}(p||q) + (1 - \beta) \text{KL}(q||p) \) between \( q(z|x) \) and \( p(z|x) \)

- \( p \) includes missingness mechanism: don’t propose what you know won’t be missing!
- Inclusive KL: learn to propose every \( z \) that is probable under \( p(z|x) \)
- Exclusive KL: learn to avoid proposing any \( z \) that is not probable under \( p(z|x) \)

Does particle smoothing help (vs. filtering)?

Each point is a single gold seq, showing \( \log q \) of proposing it under the two methods

Datasets:
- 10 synthetic (left)
- Elevator (mid)
- NYC taxi (right)