Learning Temporal Point Processes for Efficient Retrieval of Continuous Time Event Sequences

Vinayak Gupta[1], Srikanta Bedathur[1], and Abir De[2]

IIT Delhi[1], IIT Bombay[2]

Contribution: Sequence Retrieval via Temporal Point Process

A novel sequence retrieval model, called NeuroSeqRet that learns to retrieve a relevant set of sequences for a given query, from a large corpus of sequences. It applies a trainable unwarping function on the query sequence, to make it comparable with corpus sequences and two MTPP guided neural relevance models which offer a tradeoff between accuracy and efficiency. Moreover, we also propose an optimization framework to learn binary sequence embeddings from the relevance scores, suitable for the locality-sensitive hashing.

Preliminaries

Temporal Point Processes (TPPs) are state-of-the-art models for learning timestamped event-streams.

- Event sequence: $\mathcal{S}_k = \{e_i = (x_i, t_i) | t_i < t_{i+1}\}$.
- Time: $t_i \in \mathbb{R}^+$ and discrete mark: $x_i \in \mathcal{C}$.
- Historical events influence future: $e_{k+1} \sim \mathcal{S}_k$.

Common in social networks (posts and comments), healthcare (hospital visits), shopping (purchases and reviews), finance (stocks and market trends) etc.

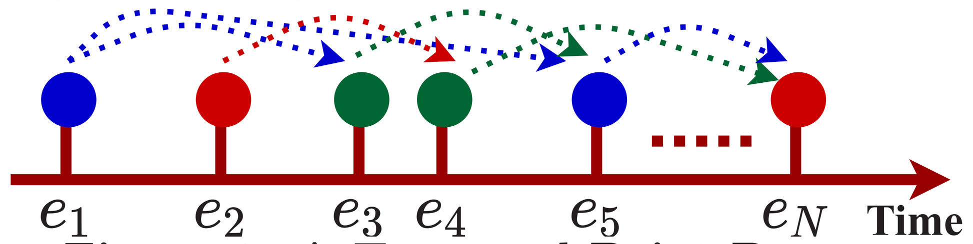

Figure 1. A Temporal Point Process

Motivation: Designing retrieval models specifically for continuous-time event sequences (CTES) has largely been unaddressed in the past.

Our Model: NeuroSeqRet

A family of supervised retrieval models for CTES and a trainable locality sensitive hashing (LSH) based method for very large datasets.

Key Contributions:

- Query unwarping function for better modeling of query-corpus sequence similarity.
- A family of self-attention and cross-attention based models for modeling temporal dynamics.
- Learnable hashing to compress sequence embeddings into binary hash vectors while limiting the loss due to compression.

First of its kind application of MTPP.

Detailed Overview

NeuroSeqRet has the following components:

I. Query Unwarping ($U_\phi(\cdot)$)**:** Use an unconstrained monotonic neural network (UMNN)[1].

$$U_\phi(t) = \int_0^T u_\phi(\tau)\, d\tau + \eta,$$

where $\eta \sim \mathcal{N}(0, \sigma)$ and $u_\phi : \mathbb{R} \rightarrow \mathbb{R}_+$.

- Unbiasedness *i.e.* a small value of $||U(t) - t||$.
- Monotonicity: sequence order remains same.

II. Similarity Metrics ($s_{p,U}(\mathcal{H}_q, \mathcal{H}_c)$)**:** Model-based and model-independent similarity.

Model-Independent

- Wasserstein distance between times.
- Mark overlap between $\mathcal{H}_q$ and $\mathcal{H}_c$.

Model-Based Fisher's Kernel over sequence embeddings.

$$\kappa_{p_\theta}(\mathcal{H}_q, \mathcal{H}_c) = \boldsymbol{v}_{p_\theta}(U(\mathcal{H}_q))^\top \boldsymbol{v}_{p_\theta}(\mathcal{H}_c),$$

where $\boldsymbol{v}_{p_\theta}$ denotes a sequence embedding. These embeddings are derived from self-and cross-attention variants of NeuroSeqRet.

- SelfAttn-NSQ: Transformer Hawkes[2] to compute independently compute the likelihood of sequences. Supports LSH for efficient retrieval in large datasets.
- CrossAttn-NSQ: Jointly learns the likelihood of future $\mathcal{H}_c$ events given complete $\mathcal{H}_q$.

III. Learnable Hashing Assigns binary (+1 or -1) hash vectors to sequences. Optimizes the following:

- Even distribution of ± 1 in vectors.
- Approximates sign$(\cdot)$ using tanh$(\cdot)$.
- Hash code entries avoid redundancy.

Trainable hashing is used in IR applications, however, such an approach is novel for CTES retrieval.

References

[1] Wehenkel, A. and Louppe, G. Unconstrained monotonic neural networks. In *NeurIPS*, 2019.

[2] Zuo S.*et al.* Transformer Hawkes Process. In *ICML*, 2020.

Sequence Retrieval Performance

	Mean Average Precision (MAP)				
	Audio	Cel.	Ele.	Health	Sports
MASS	51.1	58.2	19.3	26.4	54.7
UDTW	50.7	58.7	20.3	28.1	54.5
Sharp	52.4	59.8	22.8	28.6	56.8
RMTPP	48.9	57.6	18.7	24.8	50.3
Rank-RMTPP	52.6	60.3	23.4	29.3	55.8
SAHP	49.4	57.2	19.0	26.0	53.9
Rank-SAHP	52.9	61.8	26.5	31.6	55.1
THP	51.8	60.3	21.3	27.9	54.2
Rank-THP	54.3	63.1	29.4	33.6	56.3
SelfAttn-NSQ	55.8	64.4	30.7	35.9	57.6
CrossAttn-NSQ	56.2	65.1	32.4	37.4	58.7

Table 1. Retrieval Performance: MAP (%)

	NDCG@10				
	Audio	Cel.	Ele.	Health	Sports
MASS	20.7	38.7	9.1	13.6	22.3
UDTW	21.3	39.6	9.7	14.7	22.9
Sharp	21.9	40.6	11.7	16.8	23.7
RMTPP	20.1	39.4	8.3	12.3	19.1
Rank-RMTPP	22.4	41.2	11.4	15.5	23.9
SAHP	20.4	39.0	8.7	13.2	22.6
Rank-SAHP	23.3	42.1	13.3	17.5	25.4
THP	22.1	40.3	10.4	14.4	22.9
Rank-THP	25.4	44.2	15.3	19.7	26.5
SelfAttn-NSQ	25.9	45.8	16.5	20.4	27.8
CrossAttn-NSQ	28.3	46.9	18.1	22.0	27.9

Table 2. Retrieval Performance: NDCG@10 (%)

Experiment Setting: Five real-world datasets – Audio, Celebrity (Cel.), Electricity (Ele.), Health and Sports. Evaluation metrics – Mean Average Precision (MAP) and NDCG@10.

Baselines: We use: (i) time-series retrieval models - MASS, UDTW, Sharp and (ii) MTPP models - RMTPP, SAHP, THP. We also use ranking-loss based MTPP models with prefix Rank-.

Additional Experiments and Hashing Analysis

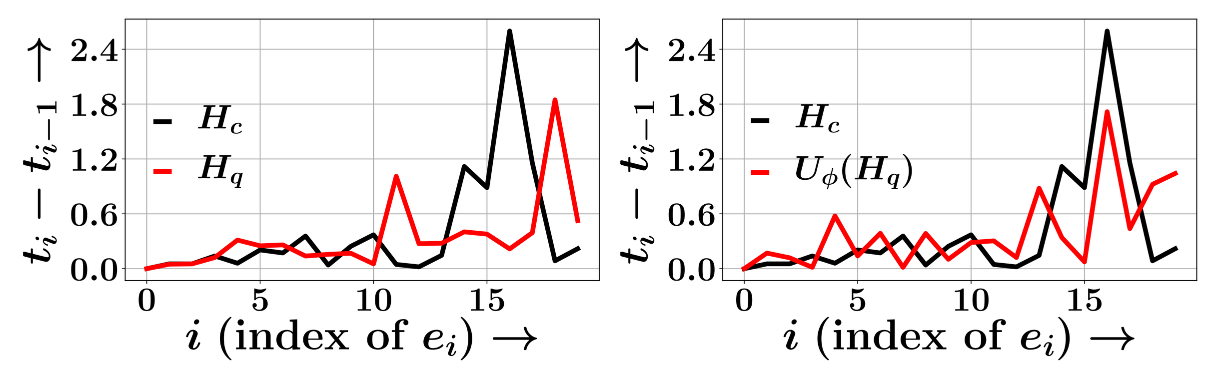

Figure 2. Effect of Unwarping

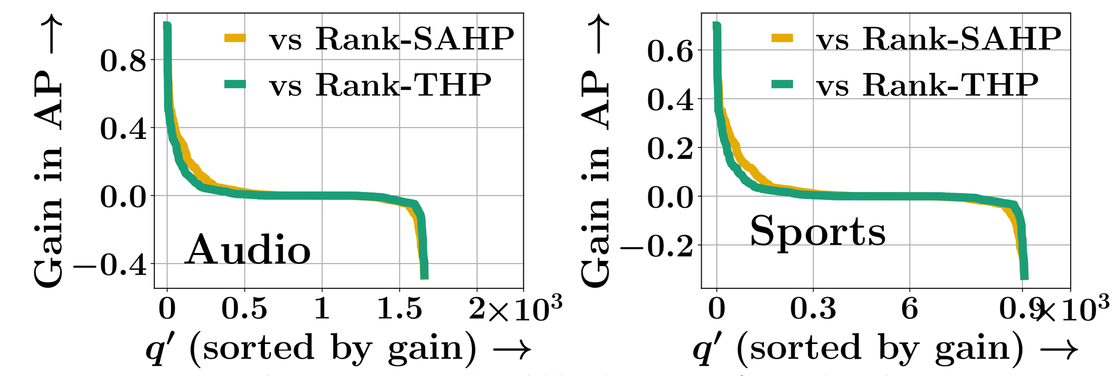

Figure 3. Drill-down Analysis

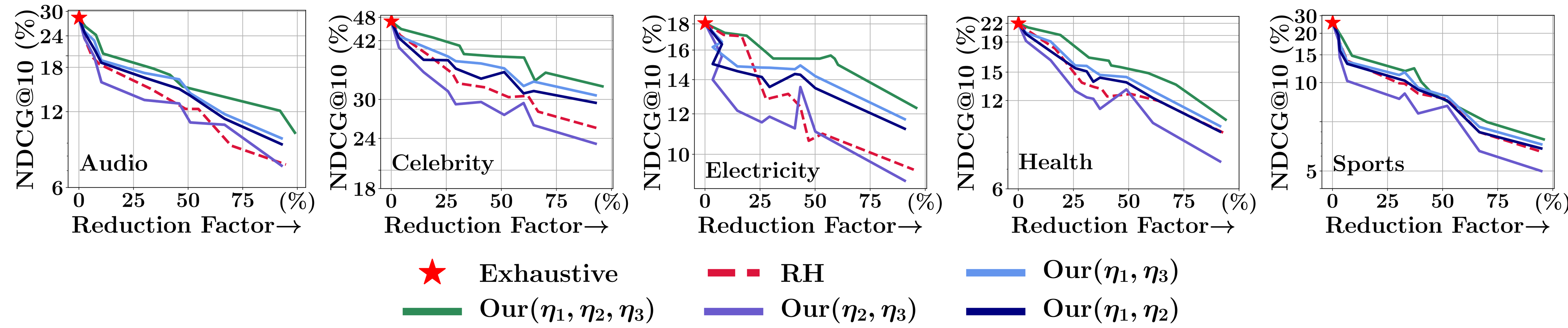

Figure 4. Tradeoff between NDCG@10 vs. Reduction factor

Unwarping: In Figure 2, we show the effect of trainable unwarping on a relevant query-corpus pair in Audio. $U_\phi(\cdot)$ learns to transform $\mathcal{H}_q$ in order to capture a high value of its latent similarity with $\mathcal{H}_c$. The results highlight that we that the performance deteriorates if we do not use an unwarping function.

Drill-down Analysis: In Figure 3, we show a comparative analysis between Rank-SAHP and Rank-THP to get the gain (or loss) for NeuroSeqRet for the average precision per sequence.

Hashing Analysis: In Figure 4, we show the efficiency of our hashing by plotting the reduction factor *i.e.* % reduction in the number of comparisons between query-corpus pairs w.r.t. the exhaustive comparisons for different hashing methods. The point marked as ⋆ indicates the case with exhaustive comparisons on the set of corpus sequences. Here, η_1, η_2, and η_3 are hyper-parameters for training loss, and Our(η_1, η_2, η_3) is the complete variant.