Nonparametric Embeddings of Sparse High-Order Interaction Events

Zheng Wang [*] [1] **Yiming Xu** [*] [2] **Conor Tillinghast** [2] **Shibo Li** [1] **Akil Narayan** [2] [3] **Shandian Zhe** [1]

Abstract

High-order interaction events are common in real-world applications. Learning embeddings that encode the complex relationships of the participants from these events is of great importance in knowledge mining and predictive tasks. Despite the success of existing approaches, *e.g.*, Poisson tensor factorization, they ignore the sparse structure underlying the data, namely the occurred interactions are far less than the possible interactions among all the participants. In this paper, we propose Nonparametric Embeddings of Sparse High-order interaction events (NESH). We hybridize a sparse hypergraph (tensor) process and a matrix Gaussian process to capture both the asymptotic structural sparsity within the interactions and nonlinear temporal relationships between the participants. We prove strong asymptotic bounds (including both a lower and an upper bound) of the sparsity ratio, which reveals the asymptotic properties of the sampled structure. We use batch-normalization, stick-breaking construction and sparse variational GP approximations to develop an efficient, scalable model inference algorithm. We demonstrate the advantage of our approach in several real-world applications.

1 Introduction

Many real-world applications are filled with interaction events between multiple entities or objects, *e.g.*, the purchases happened among *customers*, *products* and *shopping pages* at `Amazon.com`, and tweeting between *twitter users* and *messages*. Embedding these events, namely, learning a representation of the participant objects to encode their complex relationships, is of great importance and interest, in discovering hidden patterns from data, *e.g.*, clusters and outliers, and performing downstream tasks, such as recommendation and online advertising.

While Poisson tensor factorization is a popular framework for the representation learning of those events, current methods, *e.g.*, (Chi and Kolda, 2012; Hansen et al., 2015; Hu et al., 2015b; Schein et al., 2015; 2016; 2019), are mostly based on a multilinear factorization form, *e.g.*, (Tucker, 1966; Harshman, 1970), and therefore might be inadequate to estimate complex, nonlinear temporal relationships in data. More important, existing methods overlook the structural sparsity underlying these events. That is, the observed interactions are far less than all the possible interactions among the participants (*e.g.*, 0.01%). Many factorization models rely on tensor algebras and demand all the tensor entries (*i.e.*, interactions) should be observed (Kolda and Bader, 2009; Kang et al., 2012; Choi and Vishwanathan, 2014). Even for those entry-wise factorization models (Rai et al., 2014; Zhao et al., 2015; Du et al., 2018), from the Bayesian viewpoint, they are equivalent to first generating the entire tensor and then marginalizing out the unobserved entries. In practice, however, the observed interactions are often very sparse, and their proportion can get even smaller with the increase of objects. For example, in online shopping, with the growth of users and items, the number of actual purchases (while growing) takes a smaller percentage of all possible purchases, i.e., all (user, item, shopping-page) combinations, because the latter grows much faster.

In this paper, we propose NESH, a novel nonparametric Poisson factorization approach for high-order interaction events embedding. Not only does NESH flexibly estimate various nonlinear temporal relationships of the participants, it also can capture structural sparsity within the present interactions, absorbing both the structural traits and hidden relationships into the embeddings. Our major contributions are the following:

- Model. We hybridize the recent sparse tensor (hypergraph) processes (STP) (Tillinghast and Zhe, 2021) and matrix Gaussian processes (MGP) to develop a sparse event model, where the embeddings are in charge of both generating the interactions and modulating the event rates, hence can jointly encode the temporal relationships and sparse structure knowledge.

[*]Equal contribution [1]School of Computing, University of Utah [2]Department of Mathematics, University of Utah [3]Scientific Computing and Imaging (SCI) Institute, University of Utah. Correspondence to: Shandian Zhe <zhe@cs.utah.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

- Theory. We use Poisson tail estimate, Bernstein's inequality and L'Hôpital's rule to prove strong asymptotic bounds of the sparsity ratio, including both a lower and upper bound. The prior work Tillinghast and Zhe (2021) only shows the sparsity ratio of the sampled tensors asymptotically converges to zero, yet never gives an estimate of the convergence rate. Our new result reveals more theoretical insight of STP in producing sparse structures, and can also characterize the classical sparse graph generation models (Caron and Fox, 2014; Williamson, 2016).
- Algorithm. We use the stick-breaking construction of the normalized hypergraph process to compute the embedding prior, and then use batch-normalization and variational sparse GP framework to develop an efficient and scalable model estimation algorithm.

For evaluation, we conducted simulations to demonstrate that our theoretical bounds can indeed match the actual sparsity ratio and capture the asymptotic trend. Hence they can provide a reasonable convergence rate estimate and characterize the behavior of the prior. We then tested our approach NESH on three real-world datasets. NESH achieves much better predictive performance than the existing methods that use Poisson tensor factorization, additional time steps, local time dependency windows and triggering kernels. NESH also outperforms the same model with the sparse hypergraph prior removed, which demonstrates the importance of accounting for the structure sparsity. We then looked into the embeddings estimated by NESH, and found interesting patterns, including the clusters of users, sellers, and item categories in online shopping, and groups of states where car crash accidents happened.

2 Background

We assume that we observed K-way interactions among K types of objects or entities (*e.g.*, *customers*, *products* and *sellers*). We denote by D_k the number of objects of type k, and index each object by i_k ($1 \le i_k \le D_k$). We then index a particular interaction by a tuple $\mathbf{i} = (i_1, \ldots, i_K)$. We may observe multiple occurrences of a particular interaction. We denote the sequence of these events by $\mathbf{s_i} = [s_{\mathbf{i}1}, \ldots, s_{\mathbf{i}m_\mathbf{i}}]$ where $s_{\mathbf{i}j}$ is the time-stamp when j-th event occurred ($1 \le j \le m_\mathbf{i}$) and $m_\mathbf{i}$ is the total number of the occurrences of $\mathbf{i}$. Suppose we have observed events of a collection of interactions, $\mathcal{S} = \{\mathbf{s}_{\mathbf{i}_1}, \ldots, \mathbf{s}_{\mathbf{i}_N}\}$, we aim to learn an embedding for each participant object. Note that one object may participate in multiple, distinct interactions. We denote by $\mathbf{u}_j^k$ the embeddings for object j of type k, which is an R dimensional vector. We stack the embeddings of the objects of type k into a embedding matrix $\mathbf{U}^k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_{D_k}^k]^\top$, and denote by $\mathcal{U} = \{\mathbf{U}^1, \ldots, \mathbf{U}^K\}$ all the embedding matrices.

To estimate the embeddings from $\mathcal{S}$, a popular approach is tensor factorization. We can introduce a K-mode tensor $\mathcal{Y} \in \mathbb{R}^{D_1 \times \ldots \times D_K}$ accordingly, where each mode k includes D_k objects, and each entry $\mathbf{i}$ corresponds to an event sequence $\mathbf{s_i}$. For event modeling, we can use the popular (homogeneous) Poisson processes, and the probability of $\mathbf{s_i}$ is given by

$$p(s_\mathbf{i}|\lambda_\mathbf{i}) = e^{-\int_0^T \lambda_\mathbf{i} \mathrm{d}t} \prod\nolimits_{j=1}^{m_\mathbf{i}} \lambda_\mathbf{i} = e^{-T\lambda_\mathbf{i}} \lambda_\mathbf{i}^{m_\mathbf{i}}, \quad (1)$$

where T is the total time span of all the event sequences, and $\lambda_\mathbf{i} > 0$ is the rate (or intensity) of the interaction $\mathbf{i}$. Since the probability is only determined by the event count, we can place the count value $m_\mathbf{i}$ in the entry $\mathbf{i}$ of $\mathcal{Y}$, and perform count tensor factorization. Classical tensor factorization approaches include Tucker decomposition (Tucker, 1966), CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970), *etc*. Tucker decomposition assumes $\mathcal{Y} = \mathcal{W} \times_1 \mathbf{U}^1 \times_2 \ldots \times_K \mathbf{U}^K$, where $\mathcal{W} \in \mathbb{R}^{r_1 \times \ldots \times r_K}$ is a parametric core tensor, $\{\mathbf{U}^k | 1 \le k \le K\}$ are embedding matrices, and $\times_k$ is the tensor-matrix product at mode k (Kolda, 2006), which is very similar to the matrix-matrix product. If we set all $r_k = R$, and constrain $\mathcal{W}$ to be diagonal, Tucker decomposition is reduced to CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970). While numerous tensor factorization algorithms have been developed, *e.g.*, (Chu and Ghahramani, 2009; Kang et al., 2012; Choi and Vishwanathan, 2014), most of them inherit the CP or Tucker form. To perform count tensor factorization, we can use Poisson process likelihood (1) for each entry, and apply the Tucker/CP decomposition to the rates $\{\lambda_\mathbf{i}\}$ or log rates $\{\log(\lambda_\mathbf{i})\}$ (Chi and Kolda, 2012; Hu et al., 2015b). A more refined strategy is to further partition the events into a series of time steps, *e.g.*, by weeks or months, augment the count tensor with a time-step mode (Xiong et al., 2010; Schein et al., 2015; 2016; 2019), and jointly estimate the embeddings of these steps, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots\}$. We can also model the dependencies between the time steps with some dynamics, *e.g.*, (Xiong et al., 2010).

3 Model

Despite the success of existing Poisson tensor factorization approaches, they might be restricted in that (1) the commonly used CP/Tucker factorization over the rates are multilinear to the embeddings and therefore cannot capture more complex, nonlinear relationships between the interaction participants; (2) the homogeneous assumption, *i.e.*, constant event rate, might be oversimplified, overlook temporal variations of the rates, and hence miss critical temporal patterns. More important, (3) in many real-world applications, the present interactions are very sparse, when contrasted to all possible interactions. For example, despite the massive online transactions, the ratio between the number of actual transactions and *all* possible transactions (*i.e.*, all combinations of (*customer, product, seller*) is tiny[1], slightly above zero (*e.g.*, 0.01%). This proportion can get even smaller with the growth of customers, products and sellers, because their combinations can grow much faster. Similar observations can be found in clicks in online advertising, message tweeting, *etc*. Existing methods, however, are not aware of this data sparsity, and lack an effective modeling framework to embed the underlying sparse structures. To overcome these limitations, we propose NESH, a novel nonparametric embedding model for sparse high-order interaction events, presented as follows.

3.1 Nonparametric Sparse Event Modeling for High-Order Interactions

First, to highlight the sparse structure within the observed events, we view the participants as nodes, and their interactions as K-way hyperedges in a hypergraph. Each edge connects K participants (nodes), corresponding to a particular interaction $\mathbf{i}$. Attached to $\mathbf{i}$ is a sequence of events $\mathbf{s}_\mathbf{i}$ — the occurrence history of $\mathbf{i}$. Our goal is to learn an embedding for each node, which is able to not only estimate the complex temporal relationships between the nodes, but also capture the traits of the sparse hypergraph structure. To this end, we follow (Tillinghast and Zhe, 2021; Caron and Fox, 2014) to construct a stochastic process to sample the hypergraph, with a guarantee of sparsity in the asymptotic sense. Specifically, for each node type $k(1 \le k \le K)$, we sample a set of Gamma processes (Hougaard, 1986; Brix, 1999) to represent an infinite number of nodes and their weights,

$$W_{k,r}^\alpha \sim \Gamma\text{P}(\beta_\alpha) \;\; (1 \le k \le K, 1 \le r \le R) \qquad (2)$$

where β_α is a Lebesgue base measure confined to $[0, \alpha](\alpha > 0)$. Next, we use these ΓPs to construct a product-measure sum, with which as the mean measure to sample a Poisson point process (PPP) (Kingman, 1992), which represents the sampled edges of the hypergraph,

$$\begin{aligned} &M = \sum\nolimits_{r=1}^R W_{1,r}^\alpha \times \ldots \times W_{K,r}^\alpha, \\ &T|\{W_{k,r}^\alpha\}_{1\le k\le K, 1\le r\le R} \sim \text{PPP}(M). \end{aligned} \qquad (3)$$

Accordingly, T has the following form,

$$T = \sum_{\mathbf{i}\in\mathcal{E}} c_\mathbf{i} \cdot \delta_{\Theta_\mathbf{i}^\alpha},$$

where each point represents an hyperedge (interaction), $\mathcal{E}$ is the set of all the sampled points, $c_\mathbf{i} > 0$ is the count of the point $\mathbf{i}$, and $\Theta_\mathbf{i}^\alpha = \{(\theta_{1i_1}^\alpha, \ldots, \theta_{Ki_K}^\alpha)\}$ represents the location of that point and comes from the ΓPs, and $\delta_{[\cdot]}$ is the Dirac measure. In essence, the ΓPs sample infinite nodes for each type $k(1 \le k \le K)$, and then the PPP picks the nodes from each type to sample the hyperedges, *i.e.*, multiway interactions.

To examine the sparsity, we look into the nodes in the sampled hyperedges $\mathcal{E}$, which are referred to as "active" nodes. They are participants of the interactions. For example, if an edge $2 - 3 - 1$ is sampled, then node 2, 3, 1 (of type 1, 2, 3 respectively) are active nodes. Denote by D_k^α the number of distinct active nodes of type k. If we connect all the active nodes of the K types, we will have $\prod_{k=1}^K D_k^\alpha$ hyperedges (interactions) in total, *i.e.*, the volume. Sparsity means the proportion of the sampled edges in all possible edges is very small, and the former grows slower than the latter, with the increase of active nodes. Denote by N^α the number of sampled edges. The sparsity is guaranteed by

Lemma 3.1 (Corollary 3.1.1 (Tillinghast and Zhe, 2021)**).** $N^\alpha = o(\prod_{k=1}^K D_k^\alpha)$ *almost surely as* $\alpha \to \infty$*, i.e.,* $\lim\limits_{\alpha\to\infty} \frac{N^\alpha}{\prod_{k=1}^K D_k^\alpha} = 0 \quad a.s.$

Note that this is an asymptotic notion of structural sparsity — with the hyper-graph volume growing (*i.e.*, increasing α), the proportion of the sampled edges is tending to zero. It is different from other notions (Choi and Vishwanathan, 2014; Hu et al., 2015a) where the sparsity means data is dominated by zero values.

Given the sparse hypergraph prior, we then sample the observed edges (interactions) and associated events, $\mathcal{D} = \{(\mathbf{i}_1, \mathbf{s}_{\mathbf{i}_1}), \ldots, (\mathbf{i}_N, \mathbf{s}_{\mathbf{i}_N})\}$. Since they are always finite, we can use the standard PPP construction (Kingman, 1992) to sample these observations, which is computationally much more convenient and efficient. Specifically, we normalize the mean measure $M = \sum_{r=1}^R W_{1,r}^\alpha \times \ldots \times W_{K,r}^\alpha$ in (3) to obtain a probability measure, and use it to sample the N points (*i.e.*, edges/interactions) independently. To normalize M, we need to first normalize each ΓP $W_{k,r}^\alpha(1 \le k \le K, 1 \le r \le R)$, which gives a Dirichlet process (DP) (Ferguson, 1973), with the strength as $\beta_\alpha([0, \alpha]) = \alpha$, and base measure as the normalized base measure of $W_{k,r}^\alpha$ that is a uniform distribution in $[0, \alpha]$,

$$G_r^k \sim \text{DP}\big(\alpha, \text{Uniform}([0, \alpha])\big), \qquad (4)$$

where $1 \le k \le K$ and $1 \le r \le R$. The normalized M is $\frac{1}{R}\sum_{r=1}^R G_r^1 \times \ldots \times G_r^K$. To capture rich structural information, we follow "Model-II" in (Tillinghast and Zhe, 2021) to sample multiple DP weights for each node. Specifically, we drop the locations, and only sample the weights, which follow the GEM distribution (Griffiths, 1980; Engen, 1975;

[1] see Amazon data samples (http://jmcauley.ucsd.edu/data/amazon/) and dataset information in our experiments in Sec. 6.

McCloskey, 1965), and obtain

$$\widehat{G}_r^k = \sum_{j=1}^{\infty} \omega_{rj}^k \cdot \delta_j. \tag{5}$$

Accordingly, we construct a probability measure over all possible edges (interactions),

$$\widehat{M} = \sum_{\mathbf{i}=(1,\ldots,1)}^{(\infty,\ldots,\infty)} w_{\mathbf{i}} \cdot \delta_{\mathbf{i}}, \tag{6}$$

where $w_{\mathbf{i}} = \frac{1}{R}\sum_{r=1}^{R}\prod_{k=1}^{K}\omega_{ri_k}^k$. We then sample each observed interaction $\mathbf{i}_n \sim \widehat{M}$, and the probability is

$$p(\mathcal{E}) = \prod\nolimits_{n=1}^{N} p(\mathbf{i}_n) = \prod\nolimits_{n=1}^{N} w_{\mathbf{i}_n}. \tag{7}$$

Now, it can be seen that from (4) and (5), for each node j of type k, we have sampled a set of R weights $\{\omega_{1j}^k, \ldots, \omega_{Rj}^k\}$ from R DPs. From (6), we can see these weights reflect the activity of the node interacting with other nodes (of different types). Each weight naturally represents the sociability in one community/group, and these communities are overlapping. We use these sociabilities to construct the embeddings of the nodes. Therefore, they encode the sparse structural information underlying the observed interactions [2]

Given the sampled interactions $\mathcal{E}$, we then sample their occurred events $\mathcal{S} = [\mathbf{s}_{\mathbf{i}_1}, \ldots, \mathbf{s}_{\mathbf{i}_N}]$. To flexibly capture the temporal patterns, we use non-homogeneous Poisson processes. For each observed interaction $\mathbf{i}_n$, we consider a raw rate function $\rho_{\mathbf{i}_n}(t)$, and then link it to a positive rate function by taking the square, $\lambda_{\mathbf{i}_n}(t) = \left(\rho_{\mathbf{i}_n}(t)\right)^2$. Note that we can also use $\exp(\cdot)$, which, however, performs worse in our experiments. In order to capture the complex relationships of the rate functions and their temporal variations, we jointly sample the collection of rate functions, $\boldsymbol{\rho} = \{\rho_{\mathbf{i}_n}(t) | 1 \le n \le N\}$, from a matrix Gaussian process (MPG) (Rasmussen and Williams, 2006),

$$\boldsymbol{\rho} \sim \mathcal{MN}(\mathbf{0}; \kappa_1(\mathbf{x}_{\mathbf{i}}, \mathbf{x}_{\mathbf{i}'}), \kappa_2(t, t')), \tag{8}$$

where $\kappa_1(\cdot,\cdot)$ is the (row) covariance function across different interactions (hyperedges), the inputs are the embeddings of participant nodes, $\mathbf{x}_{\mathbf{i}} = [\mathbf{u}_{i_1}^1; \ldots; \mathbf{u}_{i_K}^K]$, and $\kappa_2(\cdot,\cdot)$ is the (column) covariance function about the time. We can choose nonlinear kernels for κ_1 and κ_2 to capture the complex relationships and temporal dependencies within $\boldsymbol{\rho}$. Given the rate functions, we then sample the observed event sequences from

$$p(\mathcal{S}|\boldsymbol{\lambda}) = \prod_{n=1}^{N} \exp\Big(-\int_0^T \left(\rho_{\mathbf{i}_n}(t)\right)^2 \mathrm{d}t\Big) \prod_{j=1}^{m_{\mathbf{i}_n}} \left(\rho_{\mathbf{i}}(s_{\mathbf{i}_n j})\right)^2, \tag{9}$$

where T is the total time span across all the event sequences.

From (7) and (9), we can see that, via coupling the DPs and matrix GP, both the structural properties in the sparsely observed interactions and hidden temporal relationships of the participant nodes in the events can be grasped and absorbed into the embeddings.

3.2 Theoretical Analysis of Sparsity

Although Tillinghast and Zhe (2021) has proved the asymptotic sparsity guarantee of the hypergraph process in (3) (referred to as sparse tensor process in their paper), *i.e.*, Lemma 3.1, the conclusion is rough in that we have no idea how the sparsity of the sampled hyper-graph varies along with more and more active nodes. We only know that at the limit, the sparsity ratio becomes zero. While Caron and Fox (2014) gave some convergence rate estimate in their binary graph generating models under a similar modeling framework, the estimate is only available when using generalized ΓPs (GΓPs) (Hougaard, 1986) with a particular parameter range (see Theorem 10 in their paper). The estimate is not available for the popular ordinary ΓPs as in our model. GΓPs cannot be normalized as DPs and are much harder/inconvenient for computation and inference. To extract more theoretical insight, we prove asymptotic bounds of the sparsity ratio for our hyper-graph process, which not only deepen our understanding of the properties of the sampled structures, but also fill the gap of prior works.

Lemma 3.2. *For a sparse hyper-graph process defined as in* (3)*, for all sufficiently large* α*, there exists an absolute*

[2] Model-II in (Tillinghast and Zhe, 2021) actually has made an additional adjustment on top of (3). According to the superposition theorem, $\mathrm{PPP}(\sum_{r=1}^{R} W_{1,r}^{\alpha} \times \ldots \times W_{K,r}^{\alpha}) \overset{D}{=} \sum_{r=1}^{R} \mathrm{PPP}(W_{1,r}^{\alpha} \times \ldots \times W_{K,r}^{\alpha})$. It means (3) essentially samples R hypergraphs independently and places them together. Model-II further performs a probabilistic merge of the R hypergraphs. To see this, from (5) the nodes of each type in each hypergraph are indexed by the same set of integers $(1, 2, 3, \ldots)$, and so all the possible edges in each hypergraph are indexed by the same set of index tuples. From (6) and (7), the probability of sampling a particular edge indexed by $\mathbf{i}$ is $\omega_{\mathbf{i}} = \frac{1}{R}\sum_{r=1}^{R}\prod_{k=1}^{K}\omega_{ri_k}^k$. This can be explained as the following merging procedure. We randomly select one hypergraph (with probability $\frac{1}{R}$) and check if edge $\mathbf{i}$ has been sampled in that hypergraph. If it has, we add edge $\mathbf{i}$ in the new graph; otherwise, we do not add the edge. Since in each hypergraph r, the probability of edge $\mathbf{i}$ being sampled is $\prod_{k=1}^{K}\omega_{ri_k}^k$, the overall probability of sampling the edge $\mathbf{i}$ in the new graph is the average, $\omega_{\mathbf{i}} = \frac{1}{R}\sum_{r=1}^{R}\prod_{k=1}^{K}\omega_{ri_k}^k$. It is trivial to see that the merged hypergraph is still asymptotically sparse: since these hypergraphs can be viewed as independently sampled based on the same set of nodes, each of which is asymptotically sparse, their summation is also asymptotically sparse. The benefit is that since we align these hypergraphs via the integer indices of the nodes, we can assign multiple sociabilities for each node to be better able to capture the abundant structural information.

constant $C > 0$ such that, with probability at least $1 - (C\alpha)^{-K}$,

$$\frac{e^{-1.03(2K)^{1/K}K(\log\alpha)^{1/K}}}{2K\log\alpha}\cdot\left[\frac{1.82}{(K-1)\log(1.01\alpha)}\right]^K \leq \frac{N^\alpha}{\prod_{k=1}^K D_k^\alpha} \leq \left[\frac{2.11}{(K-1)\log(0.99\alpha)}\right]^K.$$

Proof sketch. We first use concentration inequalities, including Poisson tail estimate (Vershynin, 2018) and Bernstein's inequality, and L'Hopital's rule to bound the measure of each ΓP on $[0, \alpha]$, and then take a union bound over $k = 1 \ldots K$ to obtain an upper bound of N^α. The lower bound is more technical, and requires a careful estimate of the support of the intensity measure that appears in the sampled entries with high probability, for which we apply a novel probabilistic argument. We mainly combine Poisson tail estimates, union bounds, the Bernoulli distribution and L'Hôpital's rule to bound each D_k^α and then derive the lower bound of N^α. We leave the details in Appendix.

4 Algorithm

The matrix GP in our model, coupled with DPs, is computational costly. When the number of present interactions and/or the number of their events are large, we will have to compute a huge row and/or column covariance matrix, their inverse and determinants, which is very expensive or even infeasible. To address these issues, we use the stick-breaking construction (Sethuraman, 1994), sparse variational GP approximation (Titsias, 2009; Hensman et al., 2013) and batch normalization (Ioffe and Szegedy, 2015) to develop an efficient, scalable variational inference algorithm.

Specifically, we use the stick-breaking construction to sample the DP weights (or GEM distribution),

$$v_{rj}^k \sim \text{Beta}(1, \alpha), \;\; \omega_{rj}^k = v_{rj}^k \prod\nolimits_{l=1}^{j-1}(1 - v_{rl}^k), \tag{10}$$

where $1 \leq j \leq \infty$. Therefore, we only need to estimate the stick-breaking variables $\{v_{rj}^k\}$, from which we can outright calculate the weights (or sociabilities). Since these weights can be very small and close to zero, we use their logarithm to construct the embedding of each node j of type k, $\mathbf{u}_j^k = [\log(\omega_{1j}^k); \ldots; \log(\omega_{Rj}^k)]$.

Next, to conveniently handle the matrix GP prior in (8), we unify all the raw rate functions as one function of the embeddings and time, $\rho_{\mathbf{i}}(t) = f(\mathbf{x}_{\mathbf{i}}, t)$, over which we assign a GP prior with a product covariance (kernel) function, $\kappa([\mathbf{x}_{\mathbf{i}}, t], [\mathbf{x}_{\mathbf{i}'}, t']) = \kappa_1(\mathbf{x}_{\mathbf{i}}, \mathbf{x}_{\mathbf{i}'})\kappa_2(t, t')$. This is computationally equivalent to (8) because they share the same covariance function. But we only need to deal with one function. Accordingly, the function values at the event time-stamps (across all the interactions), $\mathbf{f} = \{f(\mathbf{x}_{\mathbf{i}_n}, s_{\mathbf{i}_n j})\}_{n,j}$ follow a multivariate Gaussian distribution,

$$p(\mathbf{f}|\mathcal{U}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \kappa(\mathbf{X}, \mathbf{X})), \tag{11}$$

where each row of $\mathbf{X}$ consist of the embedding $\mathbf{x}_{\mathbf{i}_n}$ and a time-stamp $s_{\mathbf{i}_n j}$. Combing with (7) (9) (10), the joint probability of our model is

$$\begin{aligned} &p(\{\mathbf{v}_j^k\}_{1\leq j\leq D_k, 1\leq k\leq K}, \mathcal{E}, \mathcal{S}, \mathbf{f}) \\ &= \prod_{k=1}^K\prod_{j=1}^{D_k}\prod_{r=1}^R \text{Beta}(v_{rj}^k|1,\alpha)\cdot\mathcal{N}\left(\mathbf{f}|\mathbf{0},\kappa(\mathbf{X},\mathbf{X})\right) \\ &\cdot\prod_{n=1}^N \omega_{\mathbf{i}_n}\exp(-\int_0^T\left(f(\mathbf{x}_{\mathbf{i}_n},t)\right)^2\mathrm{d}t)\prod_{j=1}^{m_{\mathbf{i}_n}}\left(f(\mathbf{x}_{\mathbf{i}_n}, s_{\mathbf{i}_n j})\right)^2, \end{aligned} \tag{12}$$

where $\mathbf{v}_j^k = \{v_{jr}^k\}_{1\leq r\leq R}$. The stick-breaking variables associated with inactive nodes, *i.e.*, the nodes that do not participate any interactions, have been marginalized out.

Next, to dispense with the huge covariance matrix in (12) for $\mathbf{f}$, we use the sparse variational GP approximation (Hensman et al., 2013) to develop a variational inference algorithm. Specifically, we introduce a small set of pseudo inputs $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_h]^\top$ for $f(\cdot)$, where h is far less than the dimension of $\mathbf{f}$. We then define the pseudo outputs $\mathbf{b} = [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_h)]^\top$. We augment our model by jointly sampling $\{\mathbf{f}, \mathbf{b}\}$. Due to the GP prior over $f(\cdot)$, $\{\mathbf{f}, \mathbf{b}\}$ follow a multivariate Gaussian distribution that can be decomposed as $p(\mathbf{f}, \mathbf{b}) = p(\mathbf{b})p(\mathbf{f}|\mathbf{b})$, where $p(\mathbf{b}) = \mathcal{N}\big(\mathbf{b}|\mathbf{0}, \kappa(\mathbf{Z}, \mathbf{Z})\big)$, $p(\mathbf{f}|\mathbf{b}) = \mathcal{N}(\mathbf{f}|\mathbf{m}_{f|b}, \mathbf{\Sigma}_{f|b})$ is a conditional Gaussian distribution, $\mathbf{m}_{f|b} = \kappa(\mathbf{X}, \mathbf{Z})\kappa(\mathbf{Z}, \mathbf{Z})^{-1}\mathbf{b}$ and $\mathbf{\Sigma}_{f|b} = \kappa(\mathbf{X}, \mathbf{X}) - \kappa(\mathbf{X}, \mathbf{Z})\kappa(\mathbf{Z}, \mathbf{Z})^{-1}\kappa(\mathbf{Z}, \mathbf{X})$. The probability of the augmented model has the following form,

$$p(\text{Joint}) = \text{OtherTerms}\cdot p(\mathbf{b})p(\mathbf{f}|\mathbf{b})p(\mathcal{S}|\mathbf{f}), \tag{13}$$

where

$$p(\mathcal{S}|\mathbf{f}) = \prod_{n=1}^N \exp(-\int_0^T (f(\mathbf{x}_{\mathbf{i}_n}, t))^2\mathrm{d}t)\prod_{j=1}^{m_{\mathbf{i}_n}}(f(\mathbf{x}_{\mathbf{i}_n}, s_{\mathbf{i}_n j}))^2$$

is the likelihood of the events. Compared to (12), we just replace the Gaussian prior over $\mathbf{f}$ by the joint Gaussian prior over $\{\mathbf{f}, \mathbf{b}\}$. If we marginalize out $\mathbf{b}$, we will recover the original distribution (12). Now, we construct a variational evidence lower bound (ELBO) to avoid computing the covariance matrix $\kappa(\mathbf{X}, \mathbf{X})$. To this end, we introduce a variational posterior for $\{\mathbf{f}, \mathbf{b}\}$, $q(\mathbf{f}, \mathbf{b}) = q(\mathbf{b})p(\mathbf{f}|\mathbf{b})$, where $q(\mathbf{b}) = \mathcal{N}(\mathbf{b}|\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^\top)$, and $\mathbf{L}$ is a lower triangular matrix. Note that $\mathbf{L}\mathbf{L}^\top$ is essentially a Cholesky decomposition, and we use it to ensure the positive definiteness of the posterior covariance matrix. We then derive the EBLO

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{b},\mathbf{f})}\left[\log \frac{p(\text{Joint})}{q(\mathbf{b},\mathbf{f})}\right]$$
$$= \mathbb{E}_q\left[\log \frac{\text{OtherTerms} \cdot p(\mathbf{b})\cancel{p(\mathbf{f}|\mathbf{b})}p(\mathcal{S}|\mathbf{f})}{q(\mathbf{b})\cancel{p(\mathbf{f}|\mathbf{b})}}\right].$$

Now we can see that the full conditional Gaussian distributions $p(\mathbf{f}|\mathbf{b})$ is canceled. We only need to calculate the $h \times h$ covariance matrix for $p(\mathbf{b})$, which is very small. Hence, the cost is largely reduced. The detailed ELBO is given by

$$\mathcal{L} = -\text{KL}(q(\mathbf{b})\|p(\mathbf{b})) + \sum\nolimits_{n=1}^{N} \log w_{\mathbf{i}_n}$$
$$+ \sum\nolimits_{k=1}^{K} \sum\nolimits_{j=1}^{D_k} \sum\nolimits_{r=1}^{R} \log \text{Beta}(v_{rj}^k|1,\alpha)$$
$$- \sum_{n=1}^{N} \mathbb{E}_q \mathbb{E}_{p(t)}[T\left(f(\mathbf{x}_{\mathbf{i}_n}, t)\right)^2]$$
$$+ \sum_{n=1}^{N} \sum_{j=1}^{m_{\mathbf{i}_n}} \mathbb{E}_q[\log(f(\mathbf{x}_{\mathbf{i}_n}, s_{\mathbf{i}_n,j}))^2],$$

where $p(t) = \text{Uniform}(0,T)$, and $\text{KL}(\cdot,\cdot)$ is the Kullback-Leibler divergence. We maximize $\mathcal{L}$ to estimate the variational posterior $q(\mathbf{b})$ and the other parameters, including the stick-breaking variables $\{\mathbf{v}_j^k\}$, kernel parameters, *etc.* Due to the additive structure over both the interactions and their events, it is straightforward to combine with the reparameterization trick (Kingma and Welling, 2013) to perform efficient stochastic mini-batch optimization.

However, since our embeddings are constructed from the logarithm of the sociabilities (in $[0,1]$), $\mathbf{u}_j^k = [\log(\omega_{1j}^k);\ldots;\log(\omega_{Rj}^k)]$, and these sociabilities are often small, close to zero, their log scale can be quite big, *e.g.*, hundreds. As a result, when we feed the input $\mathbf{x}_\mathbf{i} = [\mathbf{u}_{i_1}^1,\ldots,\mathbf{u}_{i_K}^K]^\top$ to the GP kernel (*e.g.*, we used SE kernel in the experiments), it is easy to incur numerical issues or make the kernel matrix stuck to be diagonal. To address this issue, we use the batch normalization method (Ioffe and Szegedy, 2015). That is, we jointly estimate an (empirical) mean and standard deviation for each embedding element during our stochastic mini-batch optimization. Denote them by $\boldsymbol{\eta}$ and $\boldsymbol{\sigma}$. Each time, we first normalize each $\mathbf{x}_\mathbf{i}$ by

$$\mathbf{x}_\mathbf{i} \leftarrow \frac{\mathbf{x}_\mathbf{i} - \boldsymbol{\eta}}{\boldsymbol{\sigma}},$$

and then feed them to the kernel; $\boldsymbol{\eta}$ and $\boldsymbol{\sigma}$ are jointly updated with all the other parameters using stochastic gradients. We empirically found the numerical problem disappears, and the learning is effective (see Sec. 6).

Algorithm Complexity. The time complexity of our inference algorithm is $\mathcal{O}\big(mh^2 + KR\sum_{k=1}^K D_k\big)$ where $m = \sum_{n=1}^N m_{\mathbf{i}_n}$ is the total number of events. Since $h \ll m$, the computational cost is linear in m. The space complexity is $\mathcal{O}(h^2 + R\sum_{k=1}^K D_k)$, including the storage of the prior and posterior covariance matrices for pseudo outputs $\mathbf{b}$ and embeddings $\mathcal{U}$.

5 Related Work

It is natural to represent high-order interactions by multidimensional arrays or tensors. Tensor factorization is the fundamental framework for tensor analysis. Classical tensor factorization approaches include CP (Harshman, 1970) and Tucker (Tucker, 1966) decomposition, based on which numerous other methods have been proposed: (Chu and Ghahramani, 2009; Kang et al., 2012; Yang and Dunson, 2013; Choi and Vishwanathan, 2014; Du et al., 2018; Fang et al., 2021a), to name a few. Recently, nonparametric and/or neural network factorization models (Zhe et al., 2015; 2016b;a; Liu et al., 2018; Pan et al., 2020b; Tillinghast et al., 2020; Fang et al., 2021b; Tillinghast and Zhe, 2021) were developed to estimate nonlinear relationships in data, and have shown advantages over popular multilinear methods in prediction accuracy. When dealing with temporal information, existing methods mainly use homogeneous Poisson processes and decompose the event counts (Chi and Kolda, 2012; Hansen et al., 2015; Hu et al., 2015b). More advanced approaches further partition the time stamps into different steps, and perform count factorization across the time steps (Xiong et al., 2010; Schein et al., 2015; 2016; 2019). Recently, Zhe and Du (2018) used Hawkes processes to estimate the local triggering effects between the events, and modeled the triggering strength with a kernel of the embeddings of the interactions. Pan et al. (2020a) modeled the time decay as another kernel of the embeddings, and developed scalable inference for long event sequences. Wang et al. (2020) proposed a non-Hawkes, non-Poisson process to estimate the triggering and inhibition effects between the events. All these are temporal point processes that focus on rate modeling, and are different from the PPPs (with mean measure) in NESH to sample sparse interaction structures. Lloyd et al. (2015) proposed GP modulated Poisson processes and also used the square link to ensure a positive rate function. However, the work is purely about event modeling and does not learn any embedding. With the SE kernel, it derives an analytical form of ELBO. However, since our model includes the embedding (log sociabilities) in the GP prior, the ELBO is analytically intractable, and we use the reparameterization trick to conduct stochastic optimization. Recently, Pan et al. (2021) proposed a self-adaptable point process for event modeling, which can estimate both the triggering and inhibition effects within the events. More important, they construct a GP based component to enable a nonparametric estimate of the time decays of these effects. Their point process is not a Poisson process any more.

Our hyper-graph prior is inherited from the sparse tensor

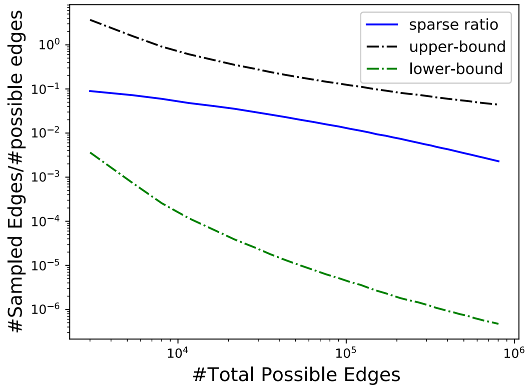


Figure 1: Sparsity ratio of the sampled hypergraphs and bounds.

process in (Tillinghast and Zhe, 2021), which can be viewed as a multi-dimensional extension of the pioneer work of Caron and Fox (2014; 2017), who first used completely random measures (CRMs) (Kingman, 1967; 1992; Lijoi et al., 2010), such as Gamma processes (ΓPs) (Hougaard, 1986), to generate sparse random graphs. However, these prior works only show the asymptotic sparse guarantee (*i.e.*, the sparsity ratio converges to zero at the limit), yet not giving any convergence rate estimate for popular ΓPs that are convenient for inference and computation. Our work fills this gap by giving strong asymptotic bounds about the sparsity, including both a lower and upper bound, which can reveal more refined insight about these sparse priors. Furthermore, we couple the sparse prior with matrix GPs to jointly sample the interactions (*i.e.*, hyperedges) and their event sequences. In this way, the embeddings can assimilate both the sparse structural information underlying the present interactions and the hidden temporal relationship of the participants.

6 Experiment

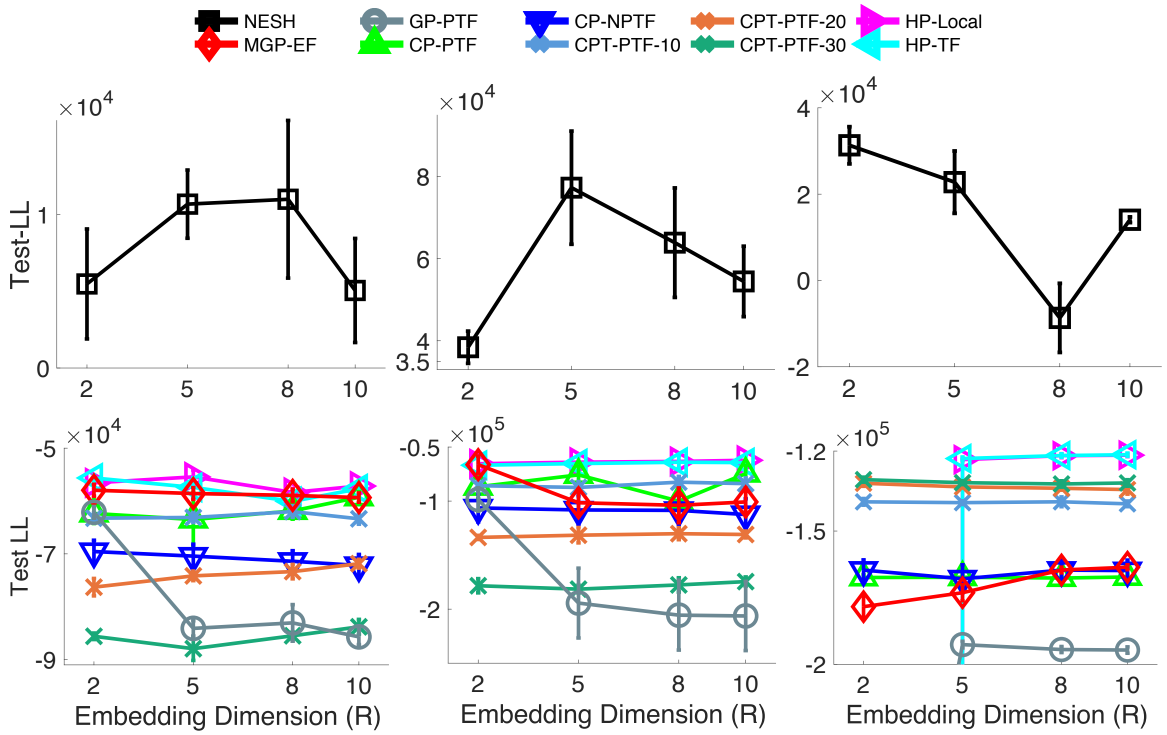
6.1 Sparsity Ratio Investigation

We first examined if our theoretical bounds in Lemma 3.2 match the actual sparsity ratio in the sampled hyper-graphs. To this end, we followed (Tillinghast and Zhe, 2021) to sample a series of hyper-graphs with three-way edges (interactions), namely $K = 3$. We set $R = 1$ and varied α in $[2, 20]$. For each particular α, we independently sampled 200 hypergraphs and computed average ratio of the sampled edges. We calculated the bounds accordingly. We show the results in a log-log plot as in Fig. 1. As we can see, the bounds clamp the actual sparsity ratio and match the trend well. The upper bound is tighter. Hence, these bounds can provide a reasonable estimate of the convergence rate and characterize the asymptotic behaviors of the structural sparsity.

6.2 Predictive Performance

Datasets. We then examined the predictive performance of NESH on the following real-world datasets. (1) *Taobao* (`https://tianchi.aliyun.com/dataset/dataDetail?dataId=53`), the shopping events in the largest online retail platform of China, from 07/01/2015 to 11/30/2015, which are interactions between 980 *users*, 274 *sellers*, 631 *items*, 58 *categories* and 2 *options*. There are in total $16,609$ distinct interactions and $69,833$ events. (2) *Crash* (`https://www.kaggle.com/usdot/nhtsa-traffic-fatalities`), fatal traffic crashes in US 2015, within 51 *states*, 288 *counties*, $2,098$ cities and 5 *landuse-types*. There are $8,691$ distinct interactions and $32,052$ events in total. (3) *Retail* (`https://tianchi.aliyun.com/dataset/dataDetail?dataId=37260`), online retail records from `tmall.com`. It includes interaction events among *stock items* and *customers*. We have $3,310$ and $1,000$ unique items and customers, among which are $31,122$ distinct interactions and $70,000$ events in total. We can see that all these datasets are very sparse. The existent interactions take 0.000085%, 0.0056% and 0.94% on *Taobao*, *Crash* and *Retail*, respectively.

Competing Methods. We compared with the following popular and/or state-of-the-art tensor decomposition methods that deal with interaction events. (1) CP-PTF, similar to (Chi and Kolda, 2012), the homogeneous Poisson process (PP) tensor decomposition, which uses CP to factorize the event rate for each particular interaction and the square link to ensure the positiveness (consistent with NESH). (2) CPT-PTF, similar to (Schein et al., 2015), which extends CP-PTF by introducing time steps in the tensor. The embeddings of the time steps are assinged a conditional Gaussian prior (Xiong et al., 2010) to model their dynamics. (3) GP-PTF, which uses GPs to estimate the square root of of the rate for each particular interaction as a nonlinear function of the associated embeddings. (4) CP-NPTF, non-homogeneous Poisson process tensor factorization where the event rate is modelled as a parametric form, $\lambda_{\mathbf{i}}(t) = t \cdot \left(\text{CP}(\mathbf{i})\right)^2$. Here $\text{CP}(\mathbf{i})$ is CP decomposition of the entry (interaction) $\mathbf{i}$. (5) HP-Local (Zhe and Du, 2018), Hawkes process based decomposition that uses a local time window to model the rate and to estimate the local excitation effects among the nearby events, (6) HP-TF (Pan et al., 2020a), another Hawkes process based on factorization method that models both the triggering strength and decay as kernel functions of the embeddings. Both HP-Local and HP-TF use a GP to model the base rate as a function of embeddings. In addition, we compared with (7) MGP-EF, matrix GP based events factorization. It is the same as our method in applying a matrix GP prior over the rates of distinct interactions. However, MGP-EF places a



(a) *Crash*

(b) *Taobao*

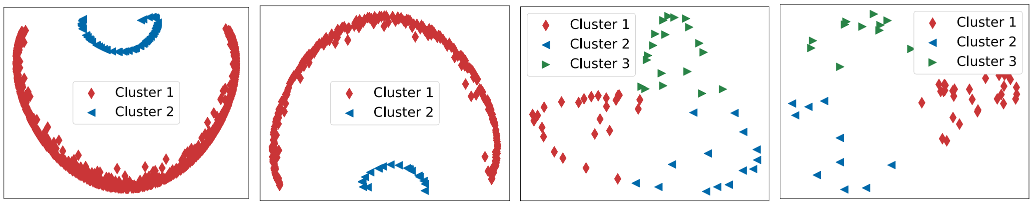
(c) *Retail*

Figure 2: Test log-likelihood (LL) on real-world datasets. CPT-PTF-{10, 20, 30} means running CPT-PTF with 10, 20 and 30 time steps. The results were averaged over five runs.

standard Gaussian prior over the embeddings, and so does not model the structure sparsity.

Settings. We implemented NESH, HP-Local, HP-TF and MGP-EF with Pytorch (Paszke et al., 2019), and the other methods with MATLAB. For all the approaches employing GPs, we used the same variational approximation as in NESH (our method), and set the number of pseudo inputs to 100. We used the square exponential (SE) kernel and initialized the kernel parameters with 1. For HP-Local, the local window size was set to 50. For our method, we chose α from $\{0.5, 1.0, 1.5, 2.5, 3\}$. We conducted stochastic mini-batch optimization for all the methods, where the batch size was set to 100. We used ADAM (Kingma and Ba, 2014) algorithm, and the learning rate was tuned from $\{5 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$. We ran each method for 400 epochs, which is enough to converge. We randomly split each dataset into 80% sequences for training, and the remaining 20% for test. We varied R, the dimension of the embeddings, from $\{2, 5, 8, 10\}$. For CPT-PTF, we tested the number of time steps from $\{10, 20, 30\}$. We ran the experiments for five times, and report the average test log-likelihood and its standard deviation in Fig. 2.

Results. As we can see from Fig. 2, NESH consistently outperforms all the competing methods by a large margin. Since the test log-likelihood of NESH is much larger than that of the other methods, we report the result of NESH separately (*i.e.*, in the top figures) so that we can compare the difference between the competing methods. It can be seen that MGP-EF is much better than GP-PTF, implying that introducing a time kernel to model non-homogeneous Poisson process rates is more advantageous. In addition, MGP-EF is much better than or comparable to CP-NPTF (see Fig. 2a). Since both methods uses non-homogeneous Poisson processes, the result demonstrates the advantage of nonparametric rate modeling (the former) over the parametric one (the latter). In most cases, HP-Local and HP-TF shows better or comparable prediction accuracy than MGP-EF. This is reasonable, because the two methods use Hawkes processes that can capture more refined temporal dependencies, *i.e.*, excitation effects between the events. However, both HP-Local and HP-TF cannot capture the sparse structure within the present interactions, and their performance is still much inferior to NESH. Finally, GP-PTF, HP-Local and HP-TF broke down at $R = 2$ on *Retail* dataset. We found their learning was unstable. In some splits, their predictive likelihood are very small, leading to much worse average likelihood than the other methods. Note that they did not use batch normalization as in NESH and MGP-EF, which might cause the learning instability under some settings.



(a) *Users* (b) *Sellers* (c) *Item Categories* (d) *States*

Figure 3: Structures of the estimated embeddings on *Taobao* (a, b, c), *Crash* (d). The points represent the participant nodes, and the colors indicate their cluster memberships.

6.3 Pattern Discovery

Next, we examined if NESH can discover hidden patterns from data. To this end, we set R to 5, and ran NESH on *Taobao* and *Crash*. Then we applied kernel PCA (Schölkopf et al., 1998) with the SE kernel to project the embeddings onto a plane. We then ran clustering algorithms to find potential structures. As we can see from Fig. 3 a and b, the embeddings of *users* and *items* from *Taobao* dataset exhibit interesting and clear cluster structures, which might correspond to separate interests/shopping habits. Note that we ran DBSCAN (Ester et al., 1996) rather than k-means to obtain the clusters. In addition, the embeddings of item *categories* on *Taobao* also shows a clear structure that was discovered by k-means (see Fig. 3c). Although the *Taobao* dataset have been completely anynoymized and we cannot investigate the meaning of these clusters, potentially they can be useful for tasks such as marketing (Zhang et al., 2017), recommendation (Liu et al., 2015; Tran et al., 2018) and click-through-rate prediction (Pan et al., 2019). In addition, the embeddings of the *states* from *Crash* dataset also exhibit clear structures (see Fig. 3d). We have checked the geolocation of these states and found the states grouped together are often neighborhoods. This is reasonable in that neighboring states might bear a resemblance to each other, in traffic regulations (*e.g.*, speed limit), road conditions, driving customs, weather changes. *etc*. All these might lead to similar or closely related patterns of traffic accident rates.

7 Conclusion

We have presented NESH, a novel nonparametric embedding method for sparse high-order interaction events. Not only can our method estimate the complex temporal relationships between the participants, our model is also able to capture the structural information underlying the observed sparse interactions. Our theoretical bounds enable convergence rate estimate and reveal insights about the asymptotic behaviors of the sparse prior over hypergraphs or tensors. In the future, we will extend our model to more expressive point processes, such as Hawkes processes, and discover more refined temporal patterns.

Acknowledgments

This work has been supported by NSF IIS-1910983, NSF DMS-1848508 and NSF CAREER Award IIS-2046295.

References

Brix, A. (1999). Generalized gamma measures and shot-noise cox processes. Advances in Applied Probability, pages 929–953.

Caron, F. and Fox, E. B. (2014). Sparse graphs using exchangeable random measures. arXiv preprint arXiv:1401.1137.

Caron, F. and Fox, E. B. (2017). Sparse graphs using exchangeable random measures. Journal of the Royal Statistical Society. Series B, Statistical Methodology, 79(5):1295.

Chi, E. C. and Kolda, T. G. (2012). On tensors, sparsity, and nonnegative factorizations. SIAM Journal on Matrix Analysis and Applications, 33(4):1272–1299.

Choi, J. H. and Vishwanathan, S. (2014). Dfacto: Distributed factorization of tensors. In Advances in Neural Information Processing Systems, pages 1296–1304.

Chu, W. and Ghahramani, Z. (2009). Probabilistic models for incomplete multi-dimensional arrays. AISTATS.

Du, Y., Zheng, Y., Lee, K.-c., and Zhe, S. (2018). Probabilistic streaming tensor decomposition. In 2018 IEEE International Conference on Data Mining (ICDM), pages 99–108. IEEE.

Engen, S. (1975). A note on the geometric series as a species frequency model. Biometrika, 62(3):697–699.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd, volume 96, pages 226–231.

Fang, S., Kirby, R. M., and Zhe, S. (2021a). Bayesian streaming sparse Tucker decomposition. In Uncertainty in Artificial Intelligence, pages 558–567. PMLR.

Fang, S., Wang, Z., Pan, Z., Liu, J., and Zhe, S. (2021b). Streaming Bayesian deep tensor factorization. In International Conference on Machine Learning, pages 3133–3142. PMLR.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. The annals of statistics, pages 209–230.

Griffiths, R. C. (1980). Lines of descent in the diffusion approximation of neutral wright-fisher models. Theoretical population biology, 17(1):37–50.

Hansen, S., Plantenga, T., and Kolda, T. G. (2015). Newton-based optimization for Kullback-Leibler nonnegative tensor factorizations. Optimization Methods and Software, 30(5):1002–1029.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Model and conditions for an"explanatory"multi-mode factor analysis. UCLA Working Papers in Phonetics, 16:1–84.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, pages 282–290. AUAI Press.

Hougaard, P. (1986). Survival models for heterogeneous populations derived from stable distributions. Biometrika, 73(2):387–396.

Hu, C., Rai, P., and Carin, L. (2015a). Zero-truncated poisson tensor factorization for massive binary tensors. In UAI.

Hu, C., Rai, P., Chen, C., Harding, M., and Carin, L. (2015b). Scalable bayesian non-negative tensor factorization for massive count data. In Proceedings, Part II, of the European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 9285, ECML PKDD 2015, pages 53–70, New York, NY, USA. Springer-Verlag New York, Inc.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pages 448–456. PMLR.

Kang, U., Papalexakis, E., Harpale, A., and Faloutsos, C. (2012). Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 316–324. ACM.

Ken-Iti, S. (1999). Lévy processes and infinitely divisible distributions. Cambridge university press.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kingman, J. (1967). Completely random measures. Pacific Journal of Mathematics, 21(1):59–78.

Kingman, J. (1992). Poisson Processes, volume 3. Clarendon Press.

Kolda, T. G. (2006). Multilinear operators for higher-order decompositions, volume 2. United States. Department of Energy.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. SIAM Review, 51(3):455–500.

Lijoi, A., Prünster, I., et al. (2010). Models beyond the Dirichlet process. Bayesian nonparametrics, 28(80):342.

Liu, B., He, L., Li, Y., Zhe, S., and Xu, Z. (2018). Neuralcp: Bayesian multiway data analysis with neural tensor decomposition. Cognitive Computation, 10(6):1051–1061.

Liu, Y.-F., Hsu, C.-Y., and Wu, S.-H. (2015). Non-linear cross-domain collaborative filtering via hyper-structure transfer. In International Conference on Machine Learning, pages 1190–1198. PMLR.

Lloyd, C., Gunter, T., Osborne, M., and Roberts, S. (2015). Variational inference for gaussian process modulated poisson processes. In International Conference on Machine Learning, pages 1814–1822. PMLR.

McCloskey, J. W. (1965). A model for the distribution of individuals by species in an environment. Michigan State University. Department of Statistics.

Pan, F., Li, S., Ao, X., Tang, P., and He, Q. (2019). Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 695–704.

Pan, Z., Wang, Z., Phillips, J. M., and Zhe, S. (2021). Self-adaptable point processes with nonparametric time decays. Advances in Neural Information Processing Systems, 34.

Pan, Z., Wang, Z., and Zhe, S. (2020a). Scalable nonparametric factorization for high-order interaction events. In International Conference on Artificial Intelligence and Statistics, pages 4325–4335. PMLR.

Pan, Z., Wang, Z., and Zhe, S. (2020b). Streaming nonlinear Bayesian tensor decomposition. In Conference on Uncertainty in Artificial Intelligence, pages 490–499. PMLR.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In NeurIPS.

Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., and Carin, L. (2014). Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In Proceedings of the 31th International Conference on Machine Learning (ICML).

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press.

Schein, A., Linderman, S. W., Zhou, M., Blei, D. M., and Wallach, H. M. (2019). Poisson-randomized gamma dynamical systems. In NeurIPS.

Schein, A., Paisley, J., Blei, D. M., and Wallach, H. (2015). Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1045–1054. ACM.

Schein, A., Zhou, M., Blei, D. M., and Wallach, H. (2016). Bayesian poisson tucker decomposition for learning the structure of international relations. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pages 2810–2819. JMLR.org.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5):1299–1319.

Sethuraman, J. (1994). A constructive definition of dirichlet priors. Statistica sinica, pages 639–650.

Tillinghast, C., Fang, S., Zhang, K., and Zhe, S. (2020). Probabilistic neural-kernel tensor decomposition. In 2020 IEEE International Conference on Data Mining (ICDM), pages 531–540. IEEE.

Tillinghast, C. and Zhe, S. (2021). Nonparametric decomposition of sparse tensors. In International Conference on Machine Learning, pages 10301–10311. PMLR.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. In International Conference on Artificial Intelligence and Statistics, pages 567–574.

Tran, T., Lee, K., Liao, Y., and Lee, D. (2018). Regularizing matrix factorization with user and item embeddings for recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 687–696.

Tucker, L. (1966). Some mathematical notes on three-mode factor analysis. Psychometrika, 31:279–311.

Vershynin, R. (2018). High-dimensional probability: An introduction with applications in data science, volume 47. Cambridge university press.

Wang, Z., Chu, X., and Zhe, S. (2020). Self-modulating nonparametric event-tensor factorization. In International Conference on Machine Learning, pages 9857–9867. PMLR.

Williamson, S. A. (2016). Nonparametric network models for link prediction. The Journal of Machine Learning Research, 17(1):7102–7121.

Xiong, L., Chen, X., Huang, T.-K., Schneider, J., and Carbonell, J. G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In Proceedings of the 2010 SIAM International Conference on Data Mining, pages 211–222. SIAM.

Yang, Y. and Dunson, D. (2013). Bayesian conditional tensor factorizations for high-dimensional classification. Journal of the Royal Statistical Society B, revision submitted.

Zhang, C., Phang, C. W., Wu, Q., and Luo, X. (2017). Nonlinear effects of social connections and interactions on individual goal attainment and spending: Evidences from online gaming markets. Journal of Marketing, 81(6):132–155.

Zhao, Q., Zhang, L., and Cichocki, A. (2015). Bayesian cp factorization of incomplete tensors with automatic rank determination. IEEE transactions on pattern analysis and machine intelligence, 37(9):1751–1763.

Zhe, S. and Du, Y. (2018). Stochastic nonparametric event-tensor decomposition. In Advances in Neural Information Processing Systems, pages 6856–6866.

Zhe, S., Qi, Y., Park, Y., Xu, Z., Molloy, I., and Chari, S. (2016a). Dintucker: Scaling up Gaussian process models on large multidimensional arrays. In Thirtieth AAAI conference on artificial intelligence.

Zhe, S., Xu, Z., Chu, X., Qi, Y., and Park, Y. (2015). Scalable nonparametric multiway data analysis. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pages 1125–1134.

Zhe, S., Zhang, K., Wang, P., Lee, K.-c., Xu, Z., Qi, Y., and Ghahramani, Z. (2016b). Distributed flexible nonlinear tensor factorization. In Advances in Neural Information Processing Systems, pages 928–936.

Appendix

In this section, we provide detailed proof for Lemma 3.2. To make the ideas accessible to a broad audience, we also give a brief introduction to the Lévy-Khintchine formula as well as the intuition behind it. Our introduction includes the Gamma Processes (ΓPs), which are used to sample the intensity measures for a sequence of Poisson Point Processes (PPPs) that are used for sparse tensor/hypergraph construction, as a special case. A comprehensive treatment of the related topics can be found in, for instance, (Ken-Iti, 1999).

A The Lévy-Khintchine formula

The sparse tensor model introduced in (Tillinghast and Zhe, 2021) first generates a discrete measure using ΓPs, which are a special type of Lévy process. To better understand the process, we take a brief detour to Lévy processes.

A Lévy process $\{X_t\}_{t\geq 0}$ is an $\mathbb{R}^d$-valued random process such that

- $X_0 = 0$ a.s.;
- X_t has stationary and independent increments;
- For every t, X_t is right-continuous and has a well-defined left-limit.

X_t is uniquely determined by X_1, which is an infinitely divisible random variable. Indeed, if the characteristic function (CF) of X_1 is $\phi(\xi)$, then the CF of X_t is $\phi^t(\xi)$ for $t \geq 0$. A complete understanding of $\phi(\xi)$ is sufficient to characterize X_t, and this can be done via the Lévy-Khintchine formula:

Theorem 1 (Lévy-Khintchine)**.** *Let X be an $\mathbb{R}^d$-valued random variable. X is infinitely divisible if and only if the CF of X, $\phi_X(\xi) := \mathbb{E}[e^{i\xi\cdot X}]$, takes the form*

$$\phi_X(\xi) = \exp\left\{-\Psi(\xi)\right\}, \tag{14}$$

where

$$\Psi(\xi) = i(a\cdot\xi) + \frac{1}{2}\|\sigma\xi\|^2 + \int_{\mathbb{R}^d} (1 - e^{i\xi\cdot z} + i(\xi\cdot z)\mathbf{1}_{(0,1)})m(dz), \tag{15}$$

where m is a Borel measure on $\mathbb{R}^d$ satisfying $m(\{0\}) = 0$ and $\int_{\mathbb{R}^d}(1\wedge\|x\|^2)m(dx) < \infty$. Here Ψ is called the Lévy exponent of X.

As a consequence of Theorem 1, we conclude that the CF of every Lévy process X_t can be written as

$$\phi_t(\xi) = \mathbb{E}[e^{i\xi\cdot X_t}] = \exp\left\{-t\Psi(\xi)\right\},$$

where Ψ is the Lévy exponent of X_1.

To comprehend the path structure of X_t using (15), we appeal to the following facts:

- Addition of a CF's exponents corresponds to addition of independent random variables (processes);
- The CF of a drifted Brownian motion $W_t = -at + \sigma B_t$ is $\exp\{-t(i(a\cdot\xi) - \frac{1}{2}\|\sigma\xi\|^2)\}$;
- The CF of a compound Poisson process with jump parameter m (i.e. a distribution) and rate parameter λ (i.e. a Lévy process) is $\exp\{-t\int_{\mathbb{R}^d}\lambda(1-e^{i\xi\cdot z})m(dz)\}$.
- The CF of a compensated compound Poisson process with jump parameter m (i.e. a distribution) and rate parameter λ (i.e. a Lévy process and a martingale) is $\exp\{-t\int_{\mathbb{R}^d}\lambda(1-e^{i\xi\cdot z}+i(\xi\cdot z))m(dz)\}$.

Let $I_0 = \{z : \|z\| \geq 1\}$ and $I_k = \{z : \|z\| \in [2^{-k-1}, 2^{-k})\}$ for $k \geq 1$. Rewrite (15) as

$$\begin{aligned} t\Psi(\xi) = t(i(a\cdot\xi) + \frac{1}{2}\|\sigma\xi\|^2) + t\int_{I_0} m(I_0)(1-e^{i\xi\cdot z})\frac{m(dz)}{m(I_0)} \\ + \sum_{k=1}^{\infty} t\int_{I_k} m(I_k)(1-e^{i\xi\cdot z}+i(\xi\cdot z))\frac{m(dz)}{m(I_k)}, \end{aligned} \tag{16}$$

where the right-hand side corresponds to three independent processes: a drifted Brownian motion, a compound Poisson process, and a series of independent compensated compound Poisson processes. Under the integrability condition on m, the last part can be shown to converge using the martingale theory. Moving the drift term (cumulation of the compensated terms; deterministic) in the last part of (16) into the Brownian motion, we decompose X_t into two independent processes, a drifted Brownian motion and a pure-jump process (with countably many jumps). This construction is the celebrated Lévy-Itô construction.

To end this section, we give a useful interpretation of compound Poisson processes. A compound Poisson process Z_t with jump parameter m and rate parameter λ is defined as

$$Z_t = \sum_{i=1}^{N_t} M_i,$$

where $M_i \overset{iid}{\sim} m(dx)$ are independent of $N_t \sim \text{Poisson}(\lambda)$. Note $\{M_i\}_{i\in[N_t]}$ follows a PPP with intensity measure $m(dx)$, for fixed t, we may also consider Z_t as the integration of x against the Poisson random measure on $[0,t] \times \mathbb{R}^d$.

B Gamma processes

A Gamma process X_t (with shape parameter β and rate parameter λ) is a Lévy Process with

$$X_t \sim \Gamma(\beta t, \lambda) \qquad\qquad t > 0.$$

It can be checked that

$$\mathbb{E}[e^{i\xi X_t}] = \left(1 - \frac{i\xi}{\lambda}\right)^{\beta t} = \exp\left\{-t\int_{\mathbb{R}} (1 - e^{-i\xi x})\frac{\beta e^{-\lambda x}}{x}\mathbf{1}_{(0,\infty)} dx\right\},$$

i.e., the Lévy measure m is

$$m(dx) = \frac{\beta e^{-\lambda x}}{x}\mathbf{1}_{(0,\infty)} dx.$$

For convenience, we set $\beta = \lambda = 1$. From m one can deduce the following properties of X_t:

- By the Lévy-Itô construction, X_t is a pure-jump process (i.e. no Brownian part), i.e., X_t has countably infinitely many jumps in $(0,t)$;
- We can associate a sample path of X_t (up to time t) with the following measure:

  $$w_t = \sum_{0<s\le t} \Delta X_s \delta_s \qquad\qquad \Delta X_s = X_s - \lim_{r\to s^-} X_r, \tag{17}$$

  where the summation is well-defined since there are at most countably many s that $\Delta_s > 0$. w_t is finite thanks to

  $$\int_{\mathbb{R}} (1 \wedge x) m(dx) < \infty;$$

- The jump sizes of X_t, $\{\Delta X_s\}_{s\in\text{supp}(w_t)} \subset (0,\infty)$, follows a PPP with intensity measure m; see the last paragraph in Section A.

C Sparse tensor/hypergraph processes

The sparse hypergraph model considered in this paper is a superposition of R independent sparse tensor process introduced in (Tillinghast and Zhe, 2021). Without loss of generality, we assume $R = 1$; the general case can be analyzed similarly. In this case, the sampled entries in the sparse tensor model are obtained as follows: Given $K \geq 2$ and time α, we

- Use K i.i.d. Gamma processes, $X_\alpha^{(1)}, \cdots, X_\alpha^{(K)}$, to generate discrete measures $W_1^\alpha, \cdots, W_K^\alpha$ as in (17). Here we change the notations to be consistent with the ones in the manuscript, with the index r omitted;

- Construct a product measure $M = \prod_{k=1}^{K} W_k^\alpha$ on $[0, \alpha]^K$;
- Take M as the intensity measure to construct a Poisson random measure T. In particular, one can take

$$T = \{Y_i\}_{i=1}^{|T|} \overset{\text{i.i.d.}}{\sim} \frac{M}{M([0,\alpha]^K)} \qquad |T| \sim \text{Poisson}(M([0,\alpha]^K)) \perp\!\!\!\perp Y_i, \tag{18}$$

  where $X \perp\!\!\!\perp Y$ means that X and Y are independent. The support of T corresponds to sampled entries in the sparse tensor, and the marginals of the support of T corresponds to the size of the sparse tensor in the respective dimension.

To get an explicit rate of convergence of sparsity as $\alpha \to \infty$, we need to estimate the cardinality of the support of T, N^α, as well as of the corresponding marginals, which are denoted by $D_1^\alpha, \cdots, D_K^\alpha$.

Lemma 2. *Fix $K \geq 2$. For a sparse tensor process defined as above, for all sufficiently large α, there exists an absolute constant $C > 0$ such that, with probability at least $1 - (C\alpha)^{-K}$,*

$$\frac{e^{-1.03(2K)^{1/K} K (\log \alpha)^{1/K}}}{2K \log \alpha} \cdot \left[\frac{1.82}{(K-1)\log(1.01\alpha)}\right]^K \leq \frac{N^\alpha}{\prod_{k=1}^K D_k^\alpha} \leq \left[\frac{2.11}{(K-1)\log(0.99\alpha)}\right]^K .$$

Proof. **Analysis of N^α.** We begin by deriving an upper bound for N^α. Note $N^\alpha \leq |T|$, where the $|T|$ is defined in (18). Conditioned on M,

$$|T| \sim \text{Poisson}(\gamma_\alpha) \qquad \gamma_\alpha = M([0,\alpha]^K) = \prod_{k=1}^K W_k^\alpha([0,\alpha]).$$

By a Poisson tail estimate (Vershynin, 2018, Exercise 2.3.6), we have

$$\mathbb{P}\left(0.99\gamma_\alpha \leq |T| \leq 1.01\gamma_\alpha\right) \geq 1 - 2e^{-c_1\gamma_\alpha}, \tag{19}$$

where $c_1 > 0$ is an absolute constant. For each $k \in [K]$,

$$W_k^\alpha([0,\alpha]) \overset{(17)}{=} \sum_{0<s\leq\alpha} \Delta X_s^{(k)} = X_\alpha^{(k)} \sim \Gamma(\alpha, 1).$$

Without loss of generality, we assume $\alpha \in \mathbb{N}$ (otherwise consider $\lceil \alpha \rceil$ and $\lfloor \alpha \rfloor$). Then, we can write $X_\alpha^{(k)}$ as a sum of i.i.d. exponentials with unit rate:

$$X_\alpha^{(k)} \overset{\mathcal{D}}{=} \sum_{i=1}^{\alpha} G_i \qquad G_i \overset{\text{i.i.d.}}{\sim} \text{Exp}(1),$$

where $\text{Exp}(1)$ is the exponential random variable with unit rate. An application of Bernstein's inequality yields

$$\mathbb{P}\left(0.99\alpha \leq W_k^\alpha([0,\alpha]) \leq 1.01\alpha\right) \geq 1 - 2e^{-c_2\alpha},$$

where $c_2 > 0$ is an absolute constant (depending only on β and λ both of which are equal to 1). Taking a union bound over k yields

$$\mathbb{P}\left(0.99\alpha \leq \min_k W_k^\alpha([0,\alpha]) \leq \max_k W_k^\alpha([0,\alpha]) \leq 1.01\alpha\right) \geq 1 - 2Ke^{-c_2\alpha}. \tag{20}$$

Combining (19) and (20) via a union bound yields that, with probability at least $1 - 2e^{-c_1(0.99\alpha)^K} - 2Ke^{-c_2\alpha}$,

$$N^\alpha \leq |T| \leq 1.01^{K+1}\alpha^K \qquad |T| \geq 0.99^{K+1}\alpha^K. \tag{21}$$

A lower bound on N^α requires more refined analysis. Let

$$a = \frac{1.01}{0.99^{\frac{K+1}{K}}}(2K\log\alpha)^{1/K} \leq 1.03(2K\log\alpha)^{1/K} \qquad K \geq 2. \tag{22}$$

For all sufficiently large α, $1 \le a \to \infty$, and

$$\alpha \cdot m([a,\infty)) \le \alpha \cdot m([1,\infty)) = \alpha \int_1^\infty \frac{e^{-x}}{x} dx \le \alpha \int_1^\infty e^{-x} dx \le \alpha \tag{23}$$

$$\begin{aligned} \alpha \cdot m([a,\infty)) &\ge \alpha \cdot m([a,1.01a)) = \alpha \int_a^{1.01a} \frac{e^{-x}}{x} dx \\ &\ge \alpha \int_a^{1.01a} \frac{e^{-x}}{1.01a} dx = \frac{\alpha e^{-a}\left(1-e^{-0.01a}\right)}{1.01a} \ge \frac{0.99\alpha e^{-a}}{a} \ge \sqrt{\alpha}. \end{aligned} \tag{24}$$

In this case, a similar Poisson tail estimate as before yields

$$\begin{aligned} &\mathbb{P}\left(0.99\alpha m([a,\infty)) \le \min_k \#\{s \le \alpha : \Delta X_s^{(k)} \in [a,\infty)\} \le \max_k \#\{s \le \alpha : \Delta X_s^{(k)} \in [a,\infty)\} \le 1.01\alpha m([a,\infty))\right) \\ &\overset{(24)}{\ge} 1 - 2Ke^{-c_1\sqrt{\alpha}}. \end{aligned} \tag{25}$$

Conditioning M and $|T|$ on the intersection of the events in (20), (21) and (25), we write

$$N^\alpha = \sum_{s=(s_1,\cdots,s_K)\in \text{supp}(M)} \mathbf{1}_{s\in T} \ge \sum_{s\in\mathcal{S}} \mathbf{1}_{s\in T}, \tag{26}$$

where

$$\mathcal{S} = \left\{ s = (s_1,\cdots,s_K) \in \text{supp}(M) : \Delta X_{s_k}^{(k)} \ge a,\ \forall k \in [K] \right\}.$$

Each term in the summand in the right-hand side of (26) is a Bernoulli random variable with parameter

$$\begin{aligned} p_s := 1 - \left(1 - \frac{\prod_{k=1}^K \Delta X_{s_k}^{(k)}}{M([0,\alpha]^K)}\right)^{|T|} &\overset{(20),(21)}{\ge} 1 - \left[1 - \left(\frac{a}{1.01\alpha}\right)^K\right]^{0.99^{K+1}\alpha^K} \\ &= 1 - \left(1 - \frac{2K\log\alpha}{0.99^{K+1}\alpha^K}\right)^{0.99^{K+1}\alpha^K} \ge 1 - \alpha^{-2K}, \end{aligned} \tag{27}$$

where the last step used the fact that $(1-\frac{1}{x})^x \le e^{-1}$ for $x > 1$. In particular, for $s \in \mathcal{S}$,

$$\mathbb{P}\left(\mathbf{1}_{s\in T} = 0\right) = 1 - p_s \overset{(27)}{\le} \alpha^{-2K}. \tag{28}$$

Since

$$\frac{0.95^K (\alpha e^{-1.03(2K)^{1/K}(\log\alpha)^{1/K}})^K}{2K\log\alpha} \overset{(22)}{\le} 0.99^{2K}\left(\frac{\alpha e^{-a}}{a}\right)^K \overset{(24),(25)}{\le} |\mathcal{S}| \overset{(23),(25)}{\le} 1.01^K\alpha^K, \tag{29}$$

taking a union bound over $s \in \mathcal{S}$ yields that, with probability at least

$$\begin{aligned} &1 - 2e^{-c_1(0.99\alpha)^K} - 2Ke^{-c_2\alpha} - 2Ke^{-c_3\alpha} - 2Ke^{-c_1\sqrt{\alpha}} - \sum_{s\in\mathcal{S}}(1-p_s) \\ \overset{(28),(29)}{\ge}\ &1 - 2e^{-c_1(0.99\alpha)^K} - 2Ke^{-c_2\alpha} - 2Ke^{-c_3\alpha} - 2Ke^{-c_1\sqrt{\alpha}} - 1.01^K\alpha^{-K}, \end{aligned}$$

the following holds:

$$N^\alpha \ge |\mathcal{S}| \ge \frac{0.95^K(\alpha e^{-1.03(2K)^{1/K}(\log\alpha)^{1/K}})^K}{2K\log\alpha}. \tag{30}$$

Analysis of D_k^α. Analysis of D_k^α is easy owing to an observation in (Tillinghast and Zhe, 2021): For $k \in [K]$, conditioned on W_ℓ^α, $\ell \ne k$,

$$D_k^\alpha \sim \text{Poisson}\left(\alpha\psi\left(\gamma_\alpha^{(-k)}\right)\right),$$

where

$$\gamma_\alpha^{(-k)} = \prod_{\ell \neq k} W_k^\alpha([0,\alpha]) \qquad\qquad \psi(\alpha) = \int_{\mathbb{R}} (1 - e^{-\alpha x}) m(dx).$$

By a similar Poisson tail estimate as before,

$$\mathbb{P}\left(0.99\alpha\psi(\gamma_\alpha^{(-k)}) \leq D_k^\alpha \leq 1.01\alpha\psi(\gamma_\alpha^{(-k)})\right) \geq 1 - 2e^{-c_1\alpha\psi(\gamma_\alpha^{(-k)})}. \tag{31}$$

It is easy to check via L'Hôpital's rule that

$$\lim_{\alpha\to\infty} \frac{\psi(\alpha)}{\log\alpha} = \lim_{\alpha\to\infty} \alpha \frac{d}{d\alpha}\psi(\alpha) = \lim_{\alpha\to\infty} \alpha \int_{\mathbb{R}} x e^{-\alpha x}(1 - e^{-\alpha x}) m(dx) = \lim_{\alpha\to\infty} \frac{\alpha^2}{(\alpha+1)(2\alpha+1)} = \frac{1}{2}.$$

Hence, for all sufficiently large α,

$$0.49 \log\alpha \leq \psi(\alpha) \leq 0.51 \log\alpha. \tag{32}$$

Thus, conditioned on the event $0.99\alpha \leq \min_{\ell\neq k} W_\ell^\alpha([0,\alpha]) \leq \max_{\ell\neq k} W_\ell^\alpha([0,\alpha]) \leq 1.01\alpha$ (which holds with probability at least as the lower bound in (20)), (31) and (32) together implies that, for all sufficiently large α,

$$\mathbb{P}\left(0.48(K-1)\alpha\log(0.99\alpha) \leq D_k^\alpha \leq 0.52(K-1)\alpha\log(1.01\alpha)\right) \geq 1 - 2e^{-c_1(0.99\alpha)^K}.$$

Taking a union bound over k yields that, for all sufficiently large α, with probability at least $1 - 2Ke^{-c_1(0.99\alpha)^K}$,

$$0.48(K-1)\alpha\log(0.99\alpha) \leq \min_k D_k^\alpha \leq \max_k D_k^\alpha \leq 0.52(K-1)\alpha\log(1.01\alpha). \tag{33}$$

Combining (21), (30), (33) and renaming the constants yields the desired result. □