Summary Markov Models for Event Sequences

Debarun Bhattacharjya[1*], **Saurabh Sihag**[2], **Oktie Hassanzadeh**[1] and **Liza Bialik**[3]
[1]IBM Research
[2]University of Pennsylvania
[3]University of Massachusetts, Amherst
{debarunb,hassanzadeh}@us.ibm.com, sihags@pennmedicine.upenn.edu, rbialik@cs.umass.edu

Abstract

Datasets involving sequences of different types of events without meaningful time stamps are prevalent in many applications, for instance when extracted from textual corpora. We propose a family of models for such event sequences – summary Markov models – where the probability of observing an event type depends only on a summary of historical occurrences of its influencing set of event types. This Markov model family is motivated by Granger causal models for time series, with the important distinction that only one event can occur in a position in an event sequence. We show that a unique minimal influencing set exists for any set of event types of interest and choice of summary function, formulate two novel models from the general family that represent specific sequence dynamics, and propose a greedy search algorithm for learning them from event sequence data. We conduct an experimental investigation comparing the proposed models with relevant baselines, and illustrate their knowledge acquisition and discovery capabilities through case studies involving sequences from text.

1 Introduction

Numerous applications require interpretable models for capturing dynamics in *multivariate event sequences*, i.e. sequences of different types of events *without time stamps*. The classic k^{th}-order Markov chain captures such dynamics, where the probability of observing a particular event type depends on the preceding k positions. Choosing a large k could however result in a blowup of the state space and over-fitting, while a small k ignores potentially important older events. In this paper, we wish to learn the specific influencers of a particular event type of interest in an event sequence, i.e. the types of events that most affect its probability of occurrence. We are particularly interested in learning such potential influencers for knowledge discovery in the *low-data* or *noisy data* regimes.

Consider the following illustrative narrative, which is a sequence of events involving a common protagonist [Chambers and Jurafsky, 2008]: [‘person visits bank’, ‘person visits restaurant’, ‘person makes phone call’, ‘person eats meal’]. Given enough data, understanding the effect of prior events may reveal that ‘person eats meal’ is more likely to occur if ‘person visits restaurant’ has occurred previously, and does not depend on ‘person visits bank’ or ‘person makes phone call’. As another example, consider a patient with numerous treatments who recovers from a chronic condition partially due to a much older therapy event. In both the examples, it is desirable to discover *which* event types affect observations of interest, where the impacts may be from older occurrences.

In this paper, we formalize a general family of models for multivariate event sequences – *summary Markov models* (SuMMs) – where the probability of occurrence of a select set of event types at any position depends on a summary of historical occurrences of a subset of all the event types. Our intent is to learn this subset as well as the quantitative nature of the influence, with the help of parent search techniques that are popular in graphical models such as Bayesian networks [Pearl, 1988]. SuMMs identify the impact of prior (possibly older) occurrences of only the key event types, i.e. those that determine the probability of observing any particular event type (or subset of types) of interest at any sequence position – we refer to these as the *influencing set*. In the aforementioned examples, SuMMs should ideally be able to identify that ‘person visits restaurant’ is an influencer for ‘person eats meal’ in the daily schedule narrative example, and that ‘therapy’ is an influencer for ‘recovery’ in the healthcare example. The emphasis here is primarily on providing interpretable insights about the dynamics of events of interest, by inferring statistical/causal relations between event types from event sequences.

SuMMs generalize many well known Markov models, such as k^{th}-order and variable order Markov chains. However, since they identify a potentially much smaller influencing set, they are able to manage the state space size while accounting for information from the past. In that regard, they also generalize models for prediction in sequences that are typically regression-based and potentially enforce sparsity on event types. To complement prior literature, we propose two specific models within the SuMM family that use novel mappings to summarize historical occurrences for event sequences.

SuMMs are applicable when events have a temporal order but no meaningful time stamps, for instance, they are suitable when: 1) it is natural to model an event sequence in discrete time – such as top-story news events in a regular news feed;

*Contact Author. Distribution Statement “A" (Approved for Public Release, Distribution Unlimited).

or 2) time-stamps for events are either too noisy or irrelevant – such as for events recorded by an unreliable operator/machine; or 3) when time stamps are unavailable or difficult to obtain – such as events extracted from text through NLP techniques.

Contributions. In this paper, we: 1) formalize a general Markov model family for event sequences where a set of event types are affected by an influencing set, proving that a unique minimal influencing set exists for general event sequence dynamics; 2) formulate two instances from the family that are suitable for many real-world sequences; 3) propose a greedy algorithm for learning the proposed models including the minimal influencing set; 4) conduct experiments around probabilistic predictions for specific event types of interest, comparing with some relevant baselines; and 5) investigate knowledge discovery benefits of being able to identify influencers of individual event types through two case studies.

2 Related Work

Event Sequences in Data Mining. There is a large body of related work on mining patterns in sequences, with a wide range of applications related to explanation or prediction of events [Mannila *et al.*, 1997; Weiss and Hirsh, 1998; Rudin *et al.*, 2012; Letham *et al.*, 2013], recommendation systems [Quadrana *et al.*, 2018], specification mining [Lemieux *et al.*, 2015] and declarative process mining in information systems [Di Ciccio *et al.*, 2018]. Much of the work in data mining related areas has focused on the problem of efficiently discovering sub-sequences that occur frequently in the input data and meet certain criteria related to the application. Our proposed family of models is related to finding *episodes*, i.e. frequently occurring patterns, but goes further by introducing the influencing set notion and performing a graph search that leverages summary statistics from the frequent patterns.

Markov & Related Models. These form a broad class of statistical models for prediction algorithms for discrete sequences. Variable order Markov models extend concepts of Markov chains by incorporating dynamic dependence on history by incorporating both high- and small-order dependencies in the data [Begleiter *et al.*, 2004]. Algorithms for learning variable order Markov models borrow tools from data compression algorithms such as context tree weighting [Willems *et al.*, 1995]. A related class of models is that of hidden Markov models (HMM), which capture latent complexities of sequential datasets [Rabiner, 1989] and are a special case of dynamic Bayes nets for modeling discrete-time temporal data [Dean and Kanazawa, 1989; Murphy, 2002].

Other Graphical Models. Other related work includes discrete-time Granger time series [Granger, 1969; Eichler, 1999] and continuous-time graphical event models for multivariate event streams [Didelez, 2008; Gunawardana and Meek, 2016; Bhattacharjya *et al.*, 2018], which also include event time stamps. We refer to the latter data type as event streams rather than event sequences; note that the time stamps are crucial in this literature as they enable a temporal point process representation [Aalen *et al.*, 2008]. Granger causal models for time series data typically consider continuous-valued measurements and therefore involve regression. A crucial distinction between SuMMs and Granger-causal models is that only one event can occur in a position in a sequence, which affects the dynamics. Chain event graphs are another related representation that model discrete processes exhibiting strong asymmetric dependence structures [Collazo *et al.*, 2018].

3 Model Formulation

3.1 Preliminaries

An **event sequence dataset** involves sequences of events of different types. Formally, $\mathbf{D}$ is a multiset $\{\mathbf{D_k}\}_{k=1}^{K}$, where $\mathbf{D_k} = [l_i]_{i=1}^{N_k}$ and event label (or type) l_i at index i in the sequence is from a known label set (or alphabet), $l_i \in \mathscr{L}$, such that $|\mathscr{L}| = M$. There are K sequences of events in the dataset with $N = \sum_{k=1}^{K} N_k$ events. We are interested in how historical occurrences of some event types impact others.

Definition 1. *The **history** at position i in an event sequence is $h_i = \{(j, l_j)\}_{j=1}^{i-1}$. The **history restricted to label set** $\mathbf{Z} \subset \mathscr{L}$ at position i only includes prior occurrences of labels from $\mathbf{Z}$; it is denoted $h_i^{\mathbf{Z}} = \{(j, l_j) : j < i, l_j \in \mathbf{Z}\}$. We remove subscript i when referring to a generic position.*

Example 1. For event sequence $[A, A, C, B]$, history $h_4 = \{(1, A), (2, A), (3, C)\}$. When restricted to $\mathbf{Z} = \{A, B\}$, $h_4^{\mathbf{Z}} = \{(1, A), (2, A)\}$. ($B$ at position 4 is excluded.)

Note that we explicitly retain indices of relevant labels in history for modeling flexibility. For instance, one may wish to ignore older prior events and only consider the most recent k positions, or conversely to ignore the most recent positions so as to model delay in the dynamics.

Definition 2. *A sequence **summary function** $s(\cdot)$ for label set $\mathbf{Z}$ maps any restricted history $h^{\mathbf{Z}}$ at any sequence position to some **summary state** $s_{\mathbf{Z}}$ in a discrete range $\Sigma_{\mathbf{Z}}$. History h is **consistent** with state $s_{\mathbf{Z}}$ if the summary function applied to h restricted to $\mathbf{Z}$ results in $s_{\mathbf{Z}}$, i.e. $s(h^{\mathbf{Z}}) = s_{\mathbf{Z}}$.*

A summary function is intended to summarize any possible history in a sequence into a (relatively) smaller number of states, which enables learning from limited data. The following example illustrates the ability to determine consistency between summary states for different label sets.

Example 2. Consider a summary function $s(\cdot)$ that results in binary instantiations over a label set that specify whether a label occurred at least once in the restricted history. (In Section 3.4, we consider this function for a proposed model). For $\mathbf{Z} = \{A, B, C\}$ and history h_4 from Example 1, the summary state is $\{a, \bar{b}, c\}$, which specifies that A and C occur in history but B does not. This history is also consistent with summary $\{a, c\}$ over label set $\{A, C\}$ but not with $\{a, b\}$ over $\{A, B\}$.

3.2 Event Sequence Dynamics

A sequential process over labels in $\mathscr{L}$ where the global dynamics in the multivariate event sequence are conditionally homogeneous given the history can be captured by a parameterization using probabilities $\Theta = \{\Theta_X : X \in \mathscr{L}\}$, $\Theta_X = \{\theta_{x|h}\}$ s.t. $\sum_{X \in \mathscr{L}} \theta_{x|h} = 1$ for all possible histories h, where $\theta_{x|h}$ is the probability of event label X occurring at any position in the sequence given history h. While considering the dynamics of a subset of the event labels $\mathbf{X} \subseteq \mathscr{L}$, we introduce a corresponding random variable denoted $\mathscr{X}$ which has a state for each label in $\mathbf{X}$ and a single state for when the label belongs to $\mathscr{L} \setminus \mathbf{X}$, if the set $\mathscr{L} \setminus \mathbf{X}$ is not empty. Thus, when $\mathbf{X}$ is a strict subset of all labels ($\mathbf{X} \subset \mathscr{L}$), there are $|\mathbf{X}| + 1$ states of $\mathscr{X}$; we denote these as x. We use $\tilde{\Theta}_{\mathbf{X}}$ to denote the sum of probabilities over label set $\mathbf{X} \subseteq \mathscr{L}$, i.e. $\tilde{\Theta}_{\mathbf{X}} = \{\tilde{\theta}_{x|h}\}$ where $\tilde{\theta}_{x|h} = \sum_{X \in \mathbf{X}} \theta_{x|h}$.

Definition 3. *Label sets $\mathbf{U}$ and $\mathbf{V} = \mathscr{L} \setminus \mathbf{U}$ are **influencing** and **non-influencing** sets for event labels $\mathbf{X}$ under summary function $s(\cdot)$ if for all $s_{\mathbf{U}} \in \Sigma_{\mathbf{U}}$, $\tilde{\theta}_{x|h} = \tilde{\theta}_{x|h'}$ for all h, h' consistent with $s_{\mathbf{U}}$. $\mathbf{U}$ is **minimal** if the condition cannot be satisfied after removing any label in $\mathbf{U}$.*

As per the definition above, historical occurrences of non-influencers of $\mathbf{X}$ under summary function $s(\cdot)$ do not affect the probability of observing labels from $\mathbf{X}$ at any sequence position. (Please see Appendix A[1] for proofs.)

Theorem 4. *There is a unique minimal influencing set for any set of labels $\mathbf{X} \subseteq \mathscr{L}$ and summary function $s(\cdot)$ for any conditionally homogeneous event sequence dynamics Θ.*

It is natural to pose the question of whether it is possible to formulate a directed (potentially cyclic) graphical representation for global dynamics in event sequences, similar to Granger causal related graphs [Eichler, 1999; Didelez, 2008].

Definition 5. *An event sequence parameterization Θ is said to **simplify** according to a directed graph $\mathscr{G}$ over labels $\mathscr{L}$ and summary function $s(\cdot)$ if the parents of each node $X \in \mathscr{G}$ are influencing sets for their children individually, under $s(\cdot)$.*

Theorem 6. *For any directed graph $\mathscr{G}$ over labels $\mathscr{L}$ and a summary function $s(\cdot)$, there exists some event sequence parameterization Θ that simplifies according to $\mathscr{G}$ and $s(\cdot)$.*

Simplification is analogous to factorization in Bayes nets [Pearl, 1988]; it is used to formalize the situation where event sequence dynamics satisfy the parameterization constraints implied by some underlying graph. Note that any parameterization Θ simplifies according to the fully connected directed graph including self loops (which indicate self-influence), just like any distribution over random variables factorizes according to a fully connected Bayes net.

3.3 Summary Markov Models (SuMMs)

In many applications, one is interested in modeling the dynamics of an individual label (or small set of labels) of significance to the modeler, with the intent of finding the minimal influencing set. This is of practical importance particularly for knowledge discovery using event sequence datasets. Theorem 4 highlights that a unique minimal influencing set exists for any label set $\mathbf{X} \in \mathscr{L}$ for any summary function $s(\cdot)$.

We therefore propose a family of models that relates occurrences of prior event labels belonging to a subset $\mathbf{U}$ to the random variable $\mathscr{X}$ corresponding to label set $\mathbf{X} \in \mathscr{L}$. The idea is to enable summarizing any restricted history with respect to these influencing labels $\mathbf{U}$ for random variable $\mathscr{X}$, making it conditionally independent of other histories given the summary at any position. Thus, $P(x|h_i) = P(x|h_i^{\mathbf{U}}) = P(x|s_{\mathbf{U}})$ for any state x of the random variable $\mathscr{X}$ and for any position i in an event sequence, where $s_{\mathbf{U}}$ is the summary state. A general formulation follows:

[1] Appendices are in the arXiv version of the paper.

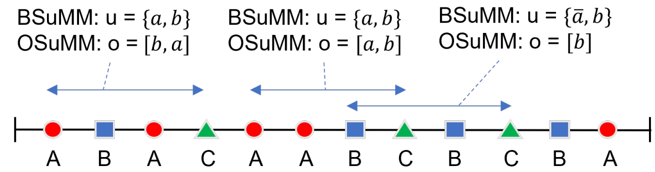

Figure 1: Illustrative event sequence over labels $\{A, B, C\}$ where $X = C$ is of interest. Also shown are instantiations $\mathbf{u}$ for BSuMM, $\mathbf{o}$ for OSuMM at all C occurrences, for influencing set $\mathbf{U} = \{A, B\}$ and look-back of 3 positions.

Definition 7. *A **summary Markov model** (SuMM) for event label set $\mathbf{X} \subseteq \mathscr{L}$ (and corresponding random variable $\mathscr{X}$) includes a summary function $s(\cdot)$, a set of influencing labels $\mathbf{U}$ and probability parameters $\theta_{x|s_{\mathbf{U}}}$ for each state of $\mathscr{X}$ and each summary state $s_{\mathbf{U}} \in \Sigma_{\mathbf{U}}$.*

In practice, if the summary function in a SuMM has range $\Sigma_{\mathbf{U}}$ that is too large, it is challenging to learn a model without enough instances to generalize over. This is the case for instance with k^{th}-order Markov chains for large k.

Remark 8. *k^{th}-order Markov chains and variable order Markov chain models such as context-tree weighting (CTW) are special cases of summary Markov models.*

SuMMs are a broad class of models for sequential dynamics around a subset of event labels, whose occurrences depend on a summary of historical occurrences of relevant event types. The nature of the mapping from the relevant historical past $h^{\mathbf{U}}$ to $\Sigma_{\mathbf{U}}$ is what distinguishes various models within the broad family, which is what we expound upon next.

3.4 Two Specific SuMMs

We propose two practical models within the broader SuMMs family, suitable for studying dynamics of individual event labels in a variety of real-world datasets that involve event sequences. In the first model, the probability of observing a state of random variable $\mathscr{X}$ corresponding to labels $\mathbf{X}$ depends on whether one of its influencing event labels have happened or not, within some label-specific and potentially user-provided historical look-back positions. Formally:

Definition 9. *A binary summary Markov model (BSuMM) for event label set $\mathbf{X} \subseteq \mathscr{L}$ (and corresponding random variable $\mathscr{X}$) includes a set of influencing labels $\mathbf{U}$, a set of look-back positions for each influencing label $\kappa = \{k_Z : Z \in \mathbf{U}\}$, and probabilities $\theta_{x|\mathbf{u}}$ for each state of $\mathscr{X}$ and each binary vector instantiation $\mathbf{u}$ of $\mathbf{U}$.*

BSuMMs are similar to Bayesian networks with binary variable parents in that there is a parameter for every state x and every possible parental configuration $\mathbf{u}$ from the $2^{|\mathbf{U}|}$ possibilities. BSuMMs are simple yet suitable for a wide range of event sequence data, as we will demonstrate later.

In a BSuMM, only the presence or absence of a parent in the relevant history has an effect, regardless of the order. For the second model, we allow for different parameters for different orders; before formalizing the model however, we introduce some notation, modifying definitions from prior work on historical orders in continuous-time models for time-stamped event streams [Bhattacharjya *et al.*, 2020].

Definition 10. *A masking function $\phi(\cdot)$ takes event sequence $s = \{(j, l_j)\}$ as input and returns a sub-sequence without label repetition, $s' = \{(k, l_k) \in s : l_k \neq l_m$ for $k \neq m\}$.*

Since labels may recur in a sequence, a masking function $\phi(\cdot)$ reduces the number of possible orders to manageable levels. In this paper, we apply a $\phi(\cdot)$ that favors more recent occurrences, consistent with other Markov models. Specifically, we only retain the last occurrence of an event label in an input sequence but we note that other choices are possible.

Definition 11. *An order instantiation $\mathbf{o}$ for label set $\mathbf{Z}$ is a permutation of a subset of $\mathbf{Z}$. The order instantiation at index i in an event sequence over k preceding positions is determined by applying masking function $\phi(\cdot)$ to events restricted to labels $\mathbf{Z}$ occurring in positions $[\max(i-k, 0), i)$.*

Definition 12. *An ordinal summary Markov model (OSuMM) for event label set $\mathbf{X} \subseteq \mathscr{L}$ (and corresponding random variable $\mathscr{X}$) and masking function $\phi(\cdot)$ includes a set of influencing labels $\mathbf{U}$, a single look-back position κ for all influencing labels, and probabilities $\theta_{x|\mathbf{o}}$ for each state of $\mathscr{X}$ and each order instantiation $\mathbf{o}$ of $\mathbf{U}$.*

Example 3. Figure 1 shows an illustrative event sequence with labels $\{A, B, C\}$. It also highlights the instantiations $\mathbf{u}$ for BSuMM and $\mathbf{o}$ for OSuMM that would be applicable for each occurrence of label C with respect to label set $\{A, B\}$ using a look-back of 3 positions. The effect of the masking function for OSuMM can be seen at the occurrence of label C at position 4, where the order instantiation is $[b, a]$ because only the more recent A occurrence at position 3 is retained. For C's occurrence at position 10, the order instantiation is $[b]$ since C is not a parent of itself.

Although an OSuMM is more expressive than a BSuMM with the same influencing set $\mathbf{U}$, since it allows for a parameter for every order instantiation rather than a configuration relying on the presence or absence of an influencing label in the relevant history, it comes at the price of learnability for datasets of limited size; there are $\sum_{i=0}^{|\mathbf{U}|} \frac{|\mathbf{U}|!}{i!}$ order instantiations for set $\mathbf{U}$ in general. Therefore if $|\mathbf{U}| = 3$, BSuMM and OSuMM would involve $2^3 = 8$ and $\sum_{i=0}^{3} \frac{3!}{i!} = 16$ parameters respectively.

4 Learning Summary Markov Models

The primary purpose in learning SuMMs, differentiating it from prevalent Markov models, is to identify the influencing set of labels of interest $\mathbf{X}$ from event sequence data. Algorithm 1 presents an approach that applies to the two specific proposed models as well as others within the family. The 'InfluencerSearch' procedure is a greedy forward and backward search, which is efficient and popular in graphical models [Chickering, 2002]; it requires a sub-procedure that can compute a model's score on a dataset when set $\mathbf{U}$ is known.

Algorithm 1 Greedy Score-based Search

```
procedure INFLUENCERSEARCH(Label set X ⊆ 𝓛, data D, model: BSuMM/ OSuMM, look-back(s) κ, masking fn φ(·) (for OSuMM), prior param. α, penalty γ)
    Pa(X) ← ∅ ; S* ← −Inf
    for each label E in 𝓛 \ Pa(X) do          ▷ Forward search
        Pa(X)' ← {Pa(X) ∪ E}
        Compute S(Pa(X)') ('ComputeScore')
        if S(Pa(X)') > S* then
            S* ← S(Pa(X)'); Pa(X) ← Pa(X)'
    for each label E in Pa(X) do              ▷ Backward search
        Pa(X)' = {Pa(X) \ E}
        Compute S(Pa(X)') ('ComputeScore')
        if S(Pa(X)') > S* then
            S* ← S(pa(X)'); Pa(X) ← Pa(X)'
    Return Pa(X), {θ̂_{x|s_Pa(X)}}

procedure COMPUTESCORE(Label set X, influencers U, data D, model: BSuMM/ OSuMM, look-back(s) κ, masking fn φ(·) (for OSuMM), prior param. α, penalty γ)
    Compute summary statistics N(x; s) and N(s)
      • For BSuMM, N(x, u) requires κ
      • For OSuMM, N(x, o) requires κ and φ(·)
    Compute Bayesian parameter estimates using α
    Compute log likelihood at these estimates and score as computed in eqs (1) and (2) using γ
      return {θ̂_{x|s}} and score S(U)
```

The 'ComputeScore' procedure in Algorithm 1 estimates conditional probability parameters $\{\hat{\theta}_{x|\mathbf{s}}\}$ and ultimately the model score. We rely on the following straightforward log likelihood computation for random variable $\mathscr{X}$ for a SuMM on an event dataset using *summary statistics*:

$$LL_{\mathscr{X}} = \sum_{x} \sum_{\mathbf{s}} \big(N(x; \mathbf{s}) log(\theta_{x|\mathbf{s}}) \big), \tag{1}$$

where $N(x, \mathbf{s})$ counts the number of times in the dataset where the random variable $\mathscr{X}$ was observed to be in state x and the historical summary in that position was $\mathbf{s}$, based on look-back(s) κ that are treated as hyper-parameters in this work. For simplicity, note that all equations are written for a single event sequence but they extend easily to multiple sequences. Also, the dependence of the summary $\mathbf{s}$ on the influencing set $\mathbf{U}$ is hidden throughout for the sake of notational convenience.

To avoid learning zero probabilities so that a model generalizes to a test dataset, we take a Bayesian approach for estimating probability parameters from equation (1). For a Dirichlet prior with parameters $\alpha_{x|\mathbf{s}}$, a Bayesian estimate for probability parameters is computed from summary statistics as $\hat{\theta}_{x|\mathbf{s}} = \frac{\alpha_{x|\mathbf{s}} + N(x;\mathbf{s})}{\alpha_{\mathbf{s}} + N(\mathbf{s})}$, where $N(x, \mathbf{s})$ is as described earlier, $N(\mathbf{s}) = \sum_{\mathbf{x}} N(x, \mathbf{s})$ and $\alpha_{\mathbf{s}} = \sum_{\mathbf{x}} \alpha_{x|\mathbf{s}}$. For experiments, we use a single parameter α as hyper-parameter, assuming that $\alpha_{x|\mathbf{s}} = \alpha \; \forall \; x, \mathbf{s}$.

We use Bayesian information criterion (BIC) as the score for $\mathscr{X}$ for a SuMM on a dataset:

$$Score_{\mathscr{X}} = LL^*_{\mathscr{X}} - \gamma |P| \frac{\log(N)}{2}, \tag{2}$$

K:	10	50	100	500	1000
$F1$:	0.23	0.59	0.69	0.93	1

Table 1: Mean F1 scores for minimal influencing set discovery in a BSuMM synthetic dataset, as a function of the # of event sequences (K). Monte Carlo error is $\sim 0.003 - 0.02$.

where γ is a penalty on complexity, generally set at a default value of 1 unless otherwise specified, $|P|$ is the number of free model parameters and $LL^*_{\mathcal{X}}$ is the log likelihood from equation (1), computed at the probability parameter estimates. Recall that N is the total number of events in the dataset. The BIC score penalizes models that are overly complex.

Theorem 13. *If the dynamics of $\mathbf{X} \in \mathscr{L}$ are governed by a BSuMM/OSuMM with look-back(s) κ, then Algorithm 1 asymptotically returns the minimal influencing set.*

Algorithm 1 can accommodate any SuMM as long as summary statistics can be computed from data. The way in which it is deployed for BSuMM vs. OSuMM differs in how the summary is specified: $\mathbf{s}$ is represented by parent configuration $\mathbf{u}$ in BSuMMs and order instantiation $\mathbf{o}$ in OSuMMs. The number of independent parameters $|P|$ can be obtained by multiplying $|\mathscr{X}| - 1$ by the summary domain size $|\Sigma_{\mathbf{U}}|$.

Theorem 14. *The time complexity of Alg. 1 is $O(M^2 N)$ where M and N are the number of labels and events, respectively.*

5 Experiments

The following experiments assess our two proposed SuMMs as well as our learning approach. Unlike many event sequence datasets in the literature such as books and musical sequences [Rudin *et al.*, 2012], our work is motivated by real-world events and event sequences extracted from text.

5.1 Influencing Set Discovery

We conduct an experiment using synthetic data to verify learning capabilities of Algorithm 1. A simple event sequence BSuMM dynamics over 5 event labels is considered, where the single label of interest has 2 other labels as its minimal influencing set. Table 1 displays mean F1 scores, comparing the estimated and ground truth influencers, over multiple generated sequence datasets as a function of the number of sequences (K) generated. The increasing trend shows asymptotic convergence. Details about the ground truth sequence dynamics and data generation are provided in Appendix B.1.

5.2 Probabilistic Prediction

In this experiment, we gauge how the proposed SuMMs compare with some baselines on a (probabilistic) prediction task.

Task & Metric(s). We are concerned with the dynamics of individual labels and therefore conduct an evaluation around probabilistic prediction. For any event label X, all our models can ascertain the probability of observing the label at any position in the sequence, given the history. We choose *negative log loss* as the evaluation metric, a popular metric for probabilistic prediction [Bishop, 2006]. For our binary prediction, a model's negative log loss is identical to its *log likelihood*.

Datasets. We consider the following structured datasets, some derived from time-stamped event datasets where the time stamp is ignored (assumed to be missing or erroneous).

- **Diabetes** [Frank and Asuncion, 2010]: Events for diabetic patients around meal ingestion, exercise, insulin intake and blood glucose level transitions b/w low, medium and high.
- **LinkedIn** [Xu *et al.*, 2017]: Employment related events such as joining a new role for 1000 LinkedIn users.
- **Stack Overflow** [Grant and Betts, 2013]: Events for engagement of 1000 users (chosen from [Du *et al.*, 2016]) around receipt of badges in a question answering website.

Besides these structured datasets, we also experiment with event sequences extracted from (unstructured) textual corpora so as to test on noisy event sequence datasets. Appendix B.2 provides further details about curation of these corpora and how they were processed into event sequences:

- **Beige Books**: Sequences of topics extracted from documents published by the United States Federal Reserve Board on events reflecting economic conditions in the United States.
- **Timelines**: Sequences of event-related Wikidata [Vrandecic and Krötzsch, 2014] concepts extracted from the timeline sections of event-related Wikipedia articles.

Baselines & Setup. We use two types of similar interpretable parametric models as well as a neural model as baselines: k^{th} order Markov chains (MC) for varying k, logistic regression (LR) with a varying look-back of k positions for obtaining features, and a simple LSTM [Hochreiter and Schmidhuber, 1997]. For experiments, each dataset is split into train/dev/test sets (70/15/15)%, removing labels that are not common across all three splits. Further information about training all models is in Appendix B.3.

Results. Table 2 shows the negative log loss (identical to log likelihood) on test sets, after averaging over event labels of interest. For the structured datasets and Beigebooks, all labels are of interest; for Timelines, we chose 15 newsworthy events (ex: 'disease outbreak', 'military raid', 'riot', 'war', etc.). Please see Appendix B.5 for a more comprehensive comparison with the MC and LR baselines.

BSuMM and OSuMM perform better than the other parametric baselines on 4 of the 5 datasets, with BSuMM faring well except on Diabetes. LR is a strong baseline for the structured datasets because some of these are more suitably modeled by accounting for repetition of previous events. LinkedIn is a prime example with several repeated labels for job role changes in the same company. We expected the LSTM to perform better but it may be restricted by the dataset sizes here, which are actually already quite large for the sorts of datasets of interest in this work. Indeed, SuMMs are intended to be applicable to even smaller datasets like those considered in the next subsection, in contrast to neural models which suffer from a lack of interpretability and whose performance generally depends on the amount of training data. LSTM performs best on Timelines, perhaps because this dataset has many more event labels compared to the others and an LSTM might be able to leverage historical information from these more effectively than other models.

The empirical evaluation shows that SuMMs are compara-

Dataset	1-MC	2-MC	3-MC	3-LR	5-LR	BSuMM	OSuMM	LSTM
Beige Books	-60.91	-40.37	-37.66	-36.85	-36.15	**-36.11**	-38.07	-63.65
Diabetes	-513.01	-488.42	-473.96	-506.05	-497.92	-497.90	**-432.89**	-595.57
LinkedIn	-110.58	-112.55	-119.55	**-92.23**	-93.37	-114.52	-115.63	-135.92
Stack Overflow	-1278.96	-1283.66	-1435.12	-1277.84	-1263.54	**-1242.59**	-1246.64	-1246.45
Timelines	-154.47	-611.78	-1343.52	-160.84	-184.05	**-141.42**	-142.18	**-135.98**

Table 2: Negative log loss (log likelihood), averaged over labels of interest, for various models computed on the test sets. k^{th} order Markov chains (MC) range from $k = 1$ to 3, and logistic regression (LR) is shown for look-back of $k = 3$, $k = 5$.

ble and often more suitable than the closest baselines, showing flexibility in fitting a variety of event sequence datasets with the additional benefit that they can identify influencers.

5.3 Case Studies of Influencing Set Identification

SuMMs are more beneficial for knowledge discovery than prediction, since they explicitly indicate influencers of event labels. Investigations on two case studies follow, with details about learning parameters relegated to Appendix B.4.

Complex Bombing Events. We surveyed 15 Wikipedia articles about planned IED attacks, where each article describes a bombing attempt. We manually curated a single sequence of event labels from each article using a small alphabet. Each sequence is an instance of a 'complex event', comprised of simpler primitive events that are represented as event labels in our model. The dataset statistics are as follows: # of labels $M = 12$, # of sequences $K = 15$, # of events $N = 60$.

Table 3 shows results from learning BSuMM, indicating selected labels of interest, their influencers and selected parameters. We use compact parameter notation, e.g. $\theta_{p|\bar{r}}$ denotes the prob. of bombing material 'purchase' given that 'radicalization' was not observed. We make a few observations:

- For many labels, we find reasonable single parent labels that make occurrence more likely, for instance, 'injury' $\longrightarrow$ 'denouncement' and 'radicalization' $\longrightarrow$ 'purchase'.
- OR relationships are discovered, e.g. 'sentencing' can occur either when bomb fails to detonate, presumably when the culprit is caught, or when an investigation occurs.

FOMC Economic Conditions. We process a subset of the Beige Books dataset mentioned previously. Here we are interested in temporal economic condition trends, therefore we only retain the shorter FOMC statements (which contain fewer topics) and construct a single temporal sequence of topics from all statements, resulting in a dataset with the following statistics: # of labels $M = 13$ (out of the 15 original topics), # of sequences $K = 1$, # of events $N = 676$.

Figure 2 shows a graph learned using a BSuMM that is learned for each topic individually, and the learned influencing set is shown as the topic's parents. The graph could help reveal insights about relations, e.g. (i) 'activity continue decline' and 'expansion aggregate demand' are the most influential economic conditions, with the former inhibiting economic progress – although the direction of the influence can only be identified from studying model parameters; (ii) some economic conditions are mutually influencing, such as 'vehicle sale robust' and 'construction activity increase', and 'expansion aggregate demand' and 'note increase demand' – in the latter case, the conditions are mutually amplifying. Graphical

Label (X)	Influencers ($\mathbf{U}$)	Parameters (θ)
denouncement	injury	$\theta_{d\|i} = 0.19$ $\theta_{d\|\bar{i}} = 0.002$
purchase	radicalization	$\theta_{p\|r} = 0.12$ $\theta_{p\|\bar{r}} = 0.002$
sentencing	failure-bomb, investigation	$\theta_{s\|\bar{f},\bar{i}} = 0.06$ $\theta_{s\|f,\bar{i}} = 0.95$ $\theta_{s\|\bar{f},i} = 0.5$

Table 3: Select BSuMM results on small IED dataset.

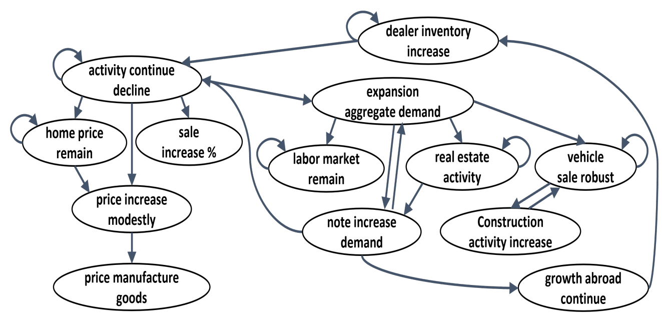

Figure 2: Visualization of BSuMM graph over FOMC topics.

representations like these from SuMMs could in general be leveraged to assist analysts with knowledge discovery.

We emphasize that a learned SuMM depends on the chosen event sequence dataset. In these case studies, all influencers are learned in the context of small domain-specific corpora. This should always be kept in mind during the analysis so as to deploy the models effectively in practice.

6 Conclusions

We have proposed summary Markov models for dynamics in a sequential event dataset, including two specific models that leverage distinct summary mappings to identify the influence of a set of event labels. Experiments on structured datasets as well as event sequences extracted from text illustrate robustness of our models in comparison with prior approaches. The main advantage of the proposed models is that they discover influencing sets in addition to predictive performance comparable with baselines. The scope of summary Markov models could be expanded by incorporating other approaches to summary mapping such as counting-based models, and adapting parameter sharing ideas from variable order Markov models to allow for expressive models with a reasonable size for the summary range. Handling noisy event sequence datasets with many irrelevant event labels poses a challenge for future work.

Acknowledgments

We thank the anonymous reviewers for helpful feedback. This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-C-0206. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing official views or policies of the Department of Defense or the U.S. Government.

References

[Aalen *et al.*, 2008] O. O. Aalen, O. Borgan, and H. K. Gjessing. *Survival and Event History Analysis: A Process Point of View*. Springer Science & Business Media, New York, NY, USA, 2008.

[Alghamdi and Alfalqi, 2015] R. Alghamdi and K. Alfalqi. A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications*, 6(1), 2015.

[Barker *et al.*, 2021] K. Barker, P. Awasthy, J. Ni, and R. Florian. IBM MNLP IE at CASE 2021 task 2: NLI reranking for zero-shot text classification. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 193–202, 2021.

[Begleiter *et al.*, 2004] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004.

[Bhattacharjya *et al.*, 2018] D. Bhattacharjya, D. Subramanian, and T. Gao. Proximal graphical event models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8147–8156, 2018.

[Bhattacharjya *et al.*, 2020] D. Bhattacharjya, T. Gao, and D. Subramanian. Order-dependent event models for agent interactions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1977–1983, 2020.

[Bishop, 2006] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Chambers and Jurafsky, 2008] N. Chambers and D. Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 94305, pages 789–797, 2008.

[Chickering, 2002] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[Collazo *et al.*, 2018] R. A. Collazo, C. Goergen, and J. Q. Smith. *Chain Event Graphs*. CRC Press, 2018.

[Dean and Kanazawa, 1989] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.

[Di Ciccio *et al.*, 2018] C. Di Ciccio, F. M. Maggi, M. Montali, and J. Mendling. On the relevance of a business constraint to an event log. *Information Systems*, 78:144–161, 2018.

[Didelez, 2008] V. Didelez. Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society, Ser. B*, 70(1):245–264, 2008.

[Du *et al.*, 2016] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1555–1564, 2016.

[Eichler, 1999] M. Eichler. *Graphical Models in Time Series Analysis*. PhD thesis, University of Heidelberg, Germany, 1999.

[Frank and Asuncion, 2010] A. Frank and A. Asuncion. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2010.

[Granger, 1969] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37:424–438, 1969.

[Grant and Betts, 2013] S. Grant and B. Betts. Encouraging user behaviour with achievements: An empirical study. In *Proceedings of the IEEE Working Conference on Mining Software Repositories (MSR)*, pages 65–68, 2013.

[Gunawardana and Meek, 2016] A. Gunawardana and C. Meek. Universal models of multivariate temporal point processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 556–563, 2016.

[Hochreiter and Schmidhuber, 1997] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Lemieux *et al.*, 2015] C. Lemieux, D. Park, and I. Beschastnikh. General LTL specification mining. In *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 81–92, 2015.

[Letham *et al.*, 2013] B. Letham, C. Rudin, and D. Madigan. Sequential event prediction. *Machine Learning*, 93(2-3):357–380, 2013.

[Mannila *et al.*, 1997] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.

[Murphy, 2002] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California Berkeley, USA, 2002.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[Quadrana *et al.*, 2018] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-aware recommender systems. *ACM Computer Surveys*, 51(4):66:1–66:36, 2018.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Rudin *et al.*, 2012] C. Rudin, B. Letham, A. Salleb-Aouissi, E. Kogan, and D. Madigan. Sequential event prediction with association rules. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, pages 615–634, 2012.

[Vrandecic and Krötzsch, 2014] D. Vrandecic and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of ACM*, 57(10):78–85, 2014.

[Weiss and Hirsh, 1998] G. M. Weiss and H. Hirsh. Learning to predict rare events in event sequences. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 359–363, 1998.

[Willems *et al.*, 1995] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995.

[Xu *et al.*, 2017] H. Xu, D. Luo, and H. Zha. Learning Hawkes processes from short doubly-censored event sequences. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3831–3840, 2017.