Summary Markov Models for Event Sequences

Authored by: Debarun Bhattacharjya, Saurabh Sihag, Oktie Hassanzadeh and Liza Bialike

Abstract

- There are many datasets without meaningful time stamps that are prevalent in many real world applications
- Proposed a family of models for such unlabeled (time stamps) event sequences
    - Summary Markov Models - probability of observing an event type depends only on a summary of historical occurrences of its influencing set of event types.
- Show that a unique minimal influencing set exists for any set of event types
- Propose a greedy search algorithm for learning them from event sequence data

Introduction and Problem Statement

- **Problem Statement**:
  - Many applications involve event sequences without meaningful timestamps, making it difficult to analyze causality.
  - Traditional Markov models struggle with finding an optimal order for considering past events: too high leads to overfitting, too low ignores important older events.
- **Core Concept**:
  - Summary Markov Models (SuMMs) introduce a summarization function that retains only the most relevant past events, improving efficiency and interpretability.

Related Work

- **Data Mining & Event Sequence Analysis**:
  - Pattern mining, sequential rule mining.
- **Markov Models & Extensions**:
  - Higher-order Markov models, Hidden Markov Models (HMMs), Bayesian networks.
- **Graphical Models & Causal Inference**:
  - Granger causality, Chain Event Graphs (CEGs).
- **Key Distinction**:
  - SuMMs operate without timestamps, identifying key influencers efficiently.

Model Formulations

Traditional Markov Model

A **kth-order Markov model** assumes that the probability of an event E sub t depends on the previous k-th events:
$P(E_t|E_{t-1}, ..., E_{t-k})$

However, increasing k leads to **exponential growth in parameters**, making the model difficult to train with limited data.

SuMMs using Summary Function

Instead of keeping the entire past sequence, SuMMs summarize past events using a function s: $P(E_t|S_t)$

where: $P(E_t|s(E_{t-1}, ..., E_{t-k}))$

This function s extracts only the necessary past influences rather than storing the full event history.

Derivation of Two SuMM Variants

Binary Summary Markov Model (BSuMM)

The simplest summary function is just tracking whether an event occurred within a past window: $s(H_t) = (I_{e_1}, I_{e_2}, \ldots, I_{e_m})$

Where

$$\begin{cases} I_{e_i} = 1, & \text{if event } e_i \text{ occurred} \\ 0, & \text{otherwise} \end{cases}$$

Ordinal Summary Markov Model (OSuMM)

Instead of just tracking presence, we also encode the order in which past events occurred:

$$s(H_t) = \text{ranked sequence of past events}$$

This keeps more information but requires more data for training.

Learning the Best Influencing Set

Instead of considering all past events, we search for a minimal set of events that still provides strong predictive power.

This is done using a greedy search algorithm that minimizes the Bayesian Information Criterion (BIC):

$$BIC = -2 \log L + k \log N$$

Where:

- L is the likelihood of the observed data given the model,
- k is the number of parameters, and
- N is the size of the dataset.

A smaller BIC means a better balance between accuracy and simplicity

Why SuMMs Work

1. Avoids Overfitting → Only keeps important past events.
2. Efficient → Uses greedy search instead of brute-force checking.
3. Generalizable → Works even with small datasets.

Learning SuMMs

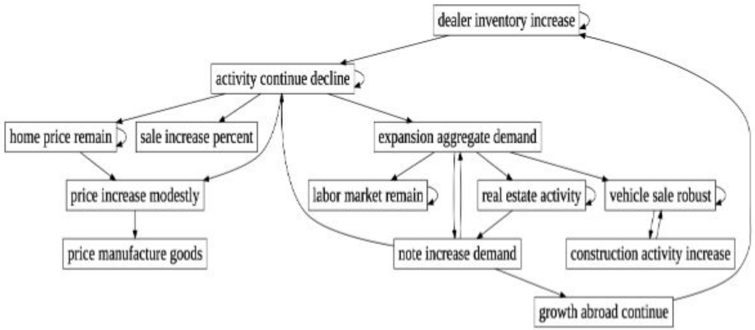
- **Efficient Learning**:
    - Greedy search algorithm identifies the optimal influencing set.
    - Uses model selection criteria to prevent overfitting.
- **Key Advantages**:
    - More interpretable than deep learning methods.
    - Computationally efficient for large datasets.

Experiments

- **Synthetic Data Tests**:
  - SuMMs correctly identify influencing sets in controlled experiments.
- **Real-World Benchmarks**:
  - Compared against Markov Chains, Logistic Regression, and LSTMs.
  - Performs better in limited-data scenarios.

Case Studies

1. **IED Bombing Events (Wikipedia Timelines)**
   - Identified key sequences leading up to incidents.
2. **FOMC Economic Trends (Beige Books)**
   - Detected causal relationships in economic indicators.



Conclusions

- **Key Takeaways**:
  - SuMMs offer an interpretable framework for event sequence analysis.
  - Provides causal insights and enhances predictive accuracy.
- **Future Directions**:
  - Extend summary functions with count-based and frequency-based methods.
  - Improve handling of noisy datasets.

Possible Further Research + Optimizations

Learning Summary Functions via Deep Learning

- **Current Limitation**: SuMMs use predefined summary functions, meaning there is no adaptive way to learn what aspects of past history should be summarized.
- **Proposed Optimization**:
    - Use Transformer-based architectures to dynamically learn which past events are most predictive.
    - Instead of manually specifying the influencing set, a neural network learns which events to keep and which to ignore.

Possible Further Research + Optimizations

SuMMs for Online and Streaming Data

- **Current Limitation**: SuMMs are designed for static datasets and do not naturally adapt to continuously evolving event streams.
- **Proposed Optimization**:
  - Implement online learning algorithms that allow SuMMs to update dynamically as new events arrive.
  - Develop streaming SuMMs that detect concept drift and adjust summary functions in response to changes in event distributions.

Possible Further Research + Optimizations

Multi-Scale Event Modeling for Variable Granularity

- **Current Limitation**: The existing SuMM framework assumes that all event sequences operate at a single level of granularity, but some influences happen over short timescales, while others unfold over longer durations.
- **Proposed Optimization**:
    - Multi-Scale SuMMs (MS-SuMMs): Maintain separate SuMMs for different timescales, such as a short-term SuMM for immediate influences and a long-term SuMM for historical patterns.
    - Introduce a hierarchical structure where different models capture dependencies at different levels of time granularity.