Sequence Synopsis:
Optimize Visual Summary of Temporal Event Data

IEEE VAST 2017 (TVCG)

Yuanzhe Chen
Hong Kong University of Science and Technology

Panpan Xu, Liu Ren
Bosch Research North America
Palo Alto, CA
Event Sequences
Use Case: Human Activities Analysis

wake up → breakfast → start work
Event Sequences

Use Case: Website Click Streams Analysis

log in → browse products → checkout

Understand customer behavior
Adjust UI design & improve customer experience
Event Sequences
Use Case: Car Faults Analysis

- Car modules like ECUs (electronic control units) / sensors emit fault signals like DTCs (diagnostics trouble codes) during operation.
- Fault data is archived for most car brands.
Event Sequences
Use Case: Car Faults Analysis

What are the typical development paths of faults? (Identify sequential patterns)

Do cars matched to the same pattern come from the same country? (correlation analysis)

Insights support predictive diagnostics (i.e. identify faults likely to happen in the future).
Better driving experience & warranty cost saving.
Visualize Event Sequences

Plotting Raw Data

259 sequences & 2500 events in total

Difficult to identify sequential patterns
Visualizing Event Sequences
Aggregation and Interaction

EventFlow
Monroe et. al. 2013

Outflow
Wongsuphasawat and Gotz, 2015

Provide succinct overview of sequences
Not robust to noisy data
Visualizing Event Sequences

Visual Summary through Sequential Pattern Mining / Clustering

- Sequence Clustering
  - Robust to noisy data
  - Interpretation of clusters: How to characterize each sequence cluster

- Sequential Pattern Mining
  - Interpretable algorithmic parameters and results
  - Large number of patterns: Need to be pruned based on heuristics

We need to have an interpretable, noise tolerant, principled approach for event sequence summarization.

Visual cluster exploration, Wei et. al. 2012

Frequence, Perer and Wang, 2014
Peekquence, Kwon et. al. 2016
Patterns&Sequences, Liu et. al. 2016
Our Approach – Sequence Synopsis

Overview

- Two-part representation of event sequences as lossless compression of the data
- Optimal pattern set selection for visual summary based on the Minimum Description Length (MDL) principle
  - Optimization algorithm
- Speedup with locality sensitive hashing
Two-Part Representation of Event Sequences

Representative pattern summarizes multiple sequences.
Our Approach – Sequence Synopsis
Two-Part Representation of Event Sequences

Corrections - event insertions (edits) recover the original sequences from the pattern.

Representative pattern summarizes multiple sequences.

Use sequential patterns for visual summary. Model information loss with the required edits (corrections).
Our Approach – Sequence Synopsis

Two-Part Representation of Event Sequences

Event deletion is another possible type of edit.

Representative pattern summarizes multiple sequences.

Different types of edits allow different variations from the pattern. Enable noise tolerant & robust pattern matching.
What can be considered as a good set of patterns to summarize a collection of event sequences?
The Minimum Description Length (MDL) Principle

- The best model (or hypothesis) of a data set should minimize its **total description length**:

  \[ L = L(M) + L(D|M) \]

  - **Model description length**
  - **Data description length** with the help of the model

- Widely used **information-theoretic** criteria for model selection
- Introduced by Jorma Rissanen in 1978
- Formalizes "**Occam’s Razor**"
Our Approach – Sequence Synopsis
Description Length of Event Sequences

\[ L = L(M) + L(D|M) \]

\[ L(P, f) = \sum_{P \in P} \text{len}(P) + \alpha \sum_{S \in S} \| \text{edits}(S, f(S)) \| + \lambda \| P \| \]

Trade-off between reducing visual complexity & minimizing information loss.
Optimize Description Length for the Best Set of Patterns

- Basic Idea: **iteratively find & merge** two groups of sequences with maximum description length reduction

- How to calculate description length reduction?
  - Find **representative sequence** for the merged group
  - Calculate the **minimum number of edits** (insertion, deletion, swapping event positions) needed to transform the representative sequence to the individual sequence in the merged group
    - Assuming insertion & deletion are allowed. Longest common subsequence (LCS) algorithm can be applied to calculate min #edits
  - **Sum up** the description length
Our Approach – Sequence Synopsis

Optimize Description Length for Best Set of Patterns

- Basic Idea: **iteratively find & merge** two groups of sequences with maximum description length reduction
Our Approach – Sequence Synopsis
Optimize Description Length for Best Set of Patterns

- Basic Idea: **iteratively find & merge** two groups of sequences with maximum description length reduction

![Diagram of sequence synopsis]

Try to merge each pair of sequences/patterns

**Calculate description length reduction**
Our Approach – Sequence Synopsis
Optimize Description Length for Best Set of Patterns

- Basic Idea: iteratively find & merge two groups of sequences with maximum description length reduction

![Diagram showing sequence optimization process]

Try to merge each pair of sequences/patterns
Our Approach – Sequence Synopsis

Optimize Description Length for Best Set of Patterns

- Basic Idea: **iteratively find & merge** two groups of sequences with maximum description length reduction

---

**Collapse:**

<table>
<thead>
<tr>
<th>S1</th>
<th>C</th>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>S3</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>S4</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>S6</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
</tbody>
</table>

-4

**Collapse:**

<table>
<thead>
<tr>
<th>C</th>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>+A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
</tbody>
</table>

*Merge the pair with maximum description length reduction*
Optimize Description Length for Best Set of Patterns

Basic Idea: **iteratively find & merge** two groups of sequences with maximum description length reduction

![Diagram showing iterative process of finding and merging sequences to optimize description length.](image)

Need to perform pairwise comparison at each iteration.
Our Approach – Sequence Synopsis
Algorithm Speedup through Locality Sensitive Hashing (LSH)

- **Bottleneck of the approach:** find best pair of event sequence groups to merge
- **Locality sensitive hashing:** algorithm for fast approximate neighbor search
Our Approach – Sequence Synopsis
Algorithm Speedup through Locality Sensitive Hashing (LSH)

- **Bottleneck of the approach:** find best pair of event sequence groups to merge
- **Locality sensitive hashing:** algorithm for fast approximate neighbor search

Simplified similarity measure with set relation
Our Approach – Sequence Synopsis
Algorithm Speedup through Locality Sensitive Hashing (LSH)

- **Bottleneck of the approach**: find best pair of event sequence groups to merge
- **Locality sensitive hashing**: algorithm for fast approximate neighbor search
Our Approach – Sequence Synopsis
Algorithm Speedup through Locality Sensitive Hashing (LSH)

- **Bottleneck of the approach:** find best pair of event sequence groups to merge
- **Locality sensitive hashing:** algorithm for fast approximate neighbor search

20x ~ 50x speed gain
Our Approach – Sequence Synopsis
Algorithm Speedup through Locality Sensitive Hashing (LSH)

- **Bottleneck of the approach**: find best pair of event sequence groups to merge
- **Locality sensitive hashing**: algorithm for fast approximate neighbor search
Our Approach – Sequence Synopsis

Advantages

- **Simultaneous** event sequence clustering and pattern extraction
- **Soft constraints** on pattern matching, therefore robust to noisy data
- **Generalizability**: possibility to include different sequence editing operations (e.g. event insertion, deletion, swapping positions)
SYSTEM
System
Visual Design

Original Data

Patterns

Corrections

Visual Design

(P1, \{S1, S2, S3\})
(P2, \{S4, S5, S6\})

#Additional events
#Matched sequences
#Missing events
System Architecture

Event sequences → Filtering data → Computing MDL representation → Interactive visualization

Interactive filtering

Level-of-detail exploration

Query raw data & attribute info
Supportive Views, UI, Case Study – Vehicle Fault Analysis
Case Study – Application Log Analysis

- D. Fisher. Agavue event data sample
- ~2000 user sessions
- Interaction log of using a data visualization application
Case Study – Application Log Analysis

- D. Fisher. Agavue event data sample
- ~2000 user sessions
- Interaction log of using a data visualization application
EVALUATION & SUMMARY
Evaluation & Summary

Comparative Experiment

- Vehicle Fault Sequence
- 259 cars & 2500 events

EventFlow
Monroe et. al. 2013

Our method
Evaluation & Summary

Contributions

- A new application domain of event sequence visualization
- A generic **two-part representation** of event sequences that:
  - **Quantifies visual complexity & information loss** in visual summaries
  - Combined with the **MDL principle**, defines an optimal set of patterns for summary
- An efficient algorithm to optimize visual summary using LSH
- A visual analytics system that supports interactive analysis of **real-world** event sequences from **different application domains**
Evaluation & Summary

Future Work

- Revise model representation to discover multiple patterns in a single sequence
- Towards quantifiable visual designs by applying the MDL principle to different types of data: graph/networks, time series ...
THANK YOU!

Q&A