Dichotomic Pattern Mining with Applications to Intent Prediction from Semi-Structured Clickstream Datasets

Xin Wang and Serdar Kadıoğlu

AI Center of Excellence, Fidelity Investments, Boston, USA
{firsname.lastname}@fmr.com

Abstract

We introduce a pattern mining framework that operates on semi-structured datasets and exploits the dichotomy between outcomes. Our approach takes advantage of constraint reasoning to find sequential patterns that occur frequently and exhibit desired properties. This allows the creation of novel pattern embeddings that are useful for knowledge extraction and predictive modeling. Finally, we present an application on customer intent prediction from digital clickstream data. Overall, we show that pattern embeddings play an integrator role between semi-structured data and machine learning models, improve the performance of the downstream task and retain interpretability.

1 Introduction

Intent prediction is an integral part of designing digital experiences that are geared toward user needs. Successful applications of intent prediction boost the performance of machine learning algorithms in various domains, including recommendation systems in e-commerce, virtual agents in retail, and conversational AI in the enterprise.

Intent prediction is a specific learning task as part of Knowledge Discovery. Knowledge extraction from various data sources has gained attraction in recent years as we become digitally connected more than ever before. Recent work in this area expanded our capabilities from processing structured data such as traditional databases to unstructured data such as text, image, and video.

In this paper, we consider digital clickstream as a source of *semi-structured data*. On the one hand, the clickstream data provides *unstructured text*, such as web pages. On the other hand, it yields sequential information where visits can be viewed as *structured event streams* representing customer journeys. Given the clickstream behavior of a set of users, we are interested in two specific questions ranging from population-level to individual-level information extraction. At the population level, we are interested in finding the most frequent clickstream patterns across all users subject to a set of properties of interest. At the individual level, we are interested in downstream tasks such as intent prediction. Finally, an overarching theme over both levels is the interpretability of the results.

SEQUENCE DATABASE ⟨(item, price, timestamp)⟩
⟨ (A, 5, 1), (A, 5, 1), (B, 3, 2), (A, 8, 3), (D, 2, 3)⟩
⟨(C, 1, 3), (B, 3, 8), (A, 3, 9)⟩
⟨(C, 4, 2), (A, 5, 5), (C, 2, 5), (D, 1, 7)⟩

Table 1: Example sequence database with three sequences.

Our contributions show that i) constrained-based sequential pattern mining is effective in extracting knowledge from semi-structured data and ii) serves as an integration technology to enable downstream applications while retaining interpretability. The main idea behind our approach is first to capture the population characteristics and extract succinct representations from large volumes of digital clickstream activity. Then, the patterns found become consumable in machine learning models. Overall, our generic framework alleviates manual feature engineering, automates feature generation, and improves the performance of downstream tasks.

To demonstrate our approach, we explore a public clickstream dataset with positive and negative intents for product purchases based on shopping activity. We apply our framework to find the most frequent patterns in digital activity and then leverage them in machine learning models for intent prediction. Finally, we show how to extract high-level signals from patterns of interest.

2 Mining Clickstream Datasets

Sequential Pattern Mining (SPM) is relevant for customer intent prediction, especially for sequential digital activity. Applications of SPM include customer purchases, call patterns and digital clickstream (Requena et al. 2020).

In SPM, we are given a set of sequences that is referred to as *sequence database*. As shown in the example in Table 1, each sequence is an ordered set of *items*. Each item might be associated with a set of *attributes* to capture item properties, e.g., price, timestamp. A *pattern* is a subsequence that occurs in at least one sequence in the database maintaining the original ordering of items. The number of sequences that contain a pattern defines the *frequency*. Given a sequence database, SPM aims to find patterns that occur more than a certain frequency threshold. Here, we find three frequent patterns: [A, D] in rows 1 and 3, [B, A] in rows 1 and 2 and [C, A] in rows 2 and 3, each occurring in two sequences.

Algorithm 1: Dichotomic Pattern Mining

```
**In:** Sequence database $\mathcal{SD}$
**In:** Binary label for sequences $\mathcal{Y}$
**In:** Minimum frequency threshold $\theta$
**In:** Pattern constraints $C_{type}(\cdot)$
**Out:** Frequent pattern sets $\mathcal{P}$

**Step 1.** Dichotomic split over the dataset
$Pos \leftarrow \{SD_i \mid Y_i = \top\}$
$Neg \leftarrow \{SD_i \mid Y_i = \bot\}$

**Step 2.** Apply constraint-based frequent pattern mining
$Pos_{frequent} \leftarrow CSPM(Pos,\ C_{type}(\cdot),\ \mathcal{L})$
$Neg_{frequent} \leftarrow CSPM(Neg,\ C_{type}(\cdot),\ \mathcal{L})$

**Step 3.** Find unique patterns and their union
$Pos_{unique} \leftarrow Pos_{frequent} \setminus Neg_{frequent}$
$Neg_{unique} \leftarrow Neg_{frequent} \setminus Pos_{frequent}$
$PN_{union} \leftarrow Pos_{frequent} \cap Neg_{frequent}$

**Step 4.** Return frequent patterns for downstream tasks
$\mathcal{P} \leftarrow \{Pos_{unique}, Neg_{unique}, PN_{union}\}$
return $\mathcal{P}$
```

In practice, finding the entire set of frequent patterns in a sequence database is not the ultimate goal. The number of patterns is typically too large and may not provide significant insights. It is thus important to search for patterns that are not only frequent but also capture specific properties of the application. This has motivated research in Constraint-based SPM (CSPM) (Chen et al. 2008). The goal of CSPM is to incorporate constraint reasoning into sequential pattern mining to find smaller subsets of interesting patterns.

As an example, let us consider online retail clickstream analysis. We might not be interested in all frequent browsing patterns. For instance, the pattern $\langle login, logout\rangle$ is likely to be frequent but offers little value. Instead, we seek recurring clickstream patterns with unique properties, e.g., frequent patterns from sessions where users spend at least a minimum amount of time on a particular set of items with a specific price range. Such constraints help reduce the search space for the mining task and help discover patterns that are more effective in knowledge discovery than arbitrary patterns.

3 Dichotomic Pattern Mining

We now describe our dichotomic pattern mining approach that operates over sequence databases augmented with binary labels denoting positive and negative outcomes. In our application, we use intent prediction as the outcome.

Algorithm 1 presents our generic approach. The algorithm receives a sequence database, $\mathcal{SD}$, containing N sequences $\{S_1, S_2, \ldots, S_N\}$. Each sequence represents a customer's behaviors in time order, for example, the digital clicks in one session. Sequences are associated with binary labels, $\mathcal{Y}$, indicating the outcome of the sequence to be positive or negative, e.g., purchase or non-purchase. As in our example in Table 1, the items in each sequence are associated with a set of attributes $\mathbb{A} = \{\mathcal{A}, \ldots, \mathcal{A}_{|\mathbb{A}|}\}$. There is a set of functions $C_{type}(\cdot)$ imposed on attributes with a certain type of operation. For example, $C_{avg}(\mathcal{A}_{price}) \geq 20$ requires a pattern to have minimum average price 20. Similarly, there is a minimum threshold θ as frequency lower bound.

SYMBOL	EVENT
1	Page view
2	Detail (see product page)
3	Add (add product to cart)
4	Remove (remove product from cart)
5	Purchase
6	Click (click on result after search)

Table 2: The symbols used to depict clickstream events.

Our algorithm is conceptually straightforward and exploits the dichotomy between outcomes. At a high level, we first split the sequences into positive and negative sets. We then apply CSPM on each group separately subject to minimum frequency while satisfying pattern constraints. Notice that frequent patterns found might overlap. Therefore, we perform a set difference operation in each direction. This allows us to distinguish between recurring patterns that *uniquely* identify the positive and negative populations.

The output of Algorithm 1 is a set of frequent patterns, $\mathcal{P}$, that provides insights into how the sequential behavior varies between populations. Thus, our algorithm serves as an integration block between pattern mining algorithms, CSPM in this paper , and the learning task, intent prediction in this paper. Using $\mathcal{P}$, we learn new representations for sequences. To create a feature vector for each sequence, we encode them using patterns. A typical approach is one-hot encoding to indicate the existence of patterns. Overall, this approach yields an automated feature extraction process that is generic and independent of the subsequent machine learning models applied to pattern embeddings.

4 Customer Intent Prediction

We apply our algorithm for customer intent prediction from click sequences of online shoppers. In the following, we describe the data, the constraint-based pattern mining, feature generation, and prediction models. We then present numeric results and study feature importance to drive insights and explanations from auto-generated features.

4.1 Clickstream Dataset

The dataset contains rich clickstream behavior on online users browsing a popular fashion e-commerce website (Requena et al. 2020). It consists of 203,084 shoppers' click sequences. There are 8,329 sequences with at least one purchase, while 194,755 sequences lead to no purchase. The sequences are composed of symbolized events as shown in Table 2 with length L between the range $5 \leq L \leq 155$. Sequences leading to purchase are labeled as positive (+1); otherwise, labeled as negative (0), resulting in a binary intent classification problem.

MODEL	FEATURES SPACE	PRECISION(%)	RECALL(%)	F1(%)	AUC(%)
LightGBM	Seq2Pat Patterns	44.70 (± 1.92)	63.15 (± 4.65)	52.20 (± 0.65)	94.98 (± 0.15)
Shallow_NN	Seq2Pat Patterns	44.40 (± 2.18)	64.11 (± 4.57)	52.31 (± 0.54)	95.00 (± 0.17)
LSTM	Clickstream	**54.96** (± 1.77)	69.53 (± 4.31)	61.28 (± 0.95)	96.41 (± 0.15)
LSTM_Seq2Pat	Clickstream + Seq2Pat Patterns	54.35 (± 2.40)	**73.64** (± 4.70)	**62.39** (± 0.81)	**96.76** (± 0.12)

Table 3: Comparison of averaged intent classification performance by different methods over 10 random Train-Test splits.

4.2 Constrained-based SPM

In Algorithm 1 at Step 2, we need a data mining approach to extract frequent patterns. For that purpose, we utilize Seq2Pat (Wang et al. 2022) to find sequential patterns that occur frequently. Seq2Pat supports constraint-based reasoning to specify desired properties. It uses the state-of-the-art multi-valued decision diagram representation of sequences (Hosseininasab, van Hoeve, and Ciré).

Next we declare our constraint model, $C_{type}(\cdot)$, to specify patterns of interest. For each event, we have two attributes: the sequential order in a sequence, $\mathcal{A}_{order}$, and the dwell time on a page, $\mathcal{A}_{time}$. We enforce two constraints to seek interesting patterns. First, we require the maximum length of a pattern to be 10. Additionally, we seek page views where customers spend at least 20 secs on average. More precisely, we set $C_{span}(\mathcal{A}_{order}) \leq 10$ and $C_{avg}(\mathcal{A}_{time}) \geq 20_{(sec)}$. We set the minimum frequency threshold θ as the 30% of the total number of sequences.

With this constraint model, Seq2Pat finds 457 frequent patterns in purchase sequences, $Pos_{frequent}$, and 236 frequent patterns from the non-purchase sequences, $Neg_{frequent}$, with some overlap between the two groups.

4.3 Feature Generation

When the sets of patterns from purchaser and non-purchaser are compared, we find 244 unique purchaser patterns, Pos_{unique}, and 23 unique non-purchaser patterns, Neg_{unique}. The groups share 213 patterns in common. In combination, we have 480 unique patterns PN_{union}. We generate the feature space via one-hot encoding. For each sequence, we create a 480-dimensional feature vector with a binary indicator to denote the existence of a pattern.

4.4 Intent Modeling

To study the the behaviour of auto-generated features we develop four different models to predict customer intent:

1. LightGBM over the Seq2Pat patterns.
2. Shallow_NN shallow neural network using one hidden layer over the Seq2Pat patterns.
3. LSTM Long short-term memory network from (Requena et al. 2020) that uses input sequences as-is. LSTM applies one hidden layer on the output of the last layer followed by a fully connected layer to make intent prediction.
4. LSTM_Seq2Pat The LSTM model boosted with pattern embeddings. LSTM_Seq2Pat uses the same architecture with LSTM, the only difference being Seq2Pat based features are concatenated to the output of LSTM and are used together as input of the hidden layer.

Notice that simpler models such as LightGBM and Shallow_NN cannot operate on semi-structured clickstream data since they cannot accommodate recurrent sequential relationships. Contrarily, more sophisticated architectures, such as LSTM can work directly with the input. For the former, our approach allows simple models to work with sequence data. For the latter, our approach augments advanced models by incorporating pattern embeddings into the feature space.

4.5 Model Training

We use 80% of the data as the train set and 20% as the test set and repeat this split 10 times for robustness. We compare the average results for each model based on Precision, Recall, F1 score, and the area under the ROC curve, aka AUC.

Hyper-parameter Tuning: We apply 3-fold cross-validation for hyper-parameter tuning in the first train-test split. We apply grid search on the number of iterations [400, 600, 800, 1000] for LightGBM, number of nodes in the hidden layer [32, 64, 128, 256, 512] for Shallow_NN and LSTM models, number of LSTM units [32, 64, 128]. We use 10% of train set as a validation set to determine if training meets early stop condition. When the loss on validation set stops decreasing steadily, training is terminated. The validation set is used to determine a decision boundary on the predictions for the highest F1 score. The final parameters are 400 iterations for LightGBM, 64 nodes for shallow_NN, 32 LSTM units in LSTM models with 64 and 128 nodes in hidden layers.

4.6 Prediction Performance

Table 3 presents the average results that compare the performance of the models. For feature space, we either use the patterns found by Seq2Pat, the original clickstream events in the raw data, or their combination. Using auto-generated Seq2Pat features, LightGBM and Shallow_NN models achieve a performance that closely match the results given in the reference work (Requena et al. 2020). The difference is, models in (Requena et al. 2020) use hand-crafted features, while we automate the feature generation process here. When a more sophisticated model such as LSTM is used, it outperforms LightGBM and Shallow_NN. When the LSTM model is combined with Seq2Pat, LSTM_Seq2Pat yields substantial increase in Recall, and consequently, the highest F1 score. LSTM_Seq2Pat is also superior to others in terms of AUC. We conclude that the features extracted automatically via Seq2Pat boost ML models in the downstream task for shopper intent prediction.

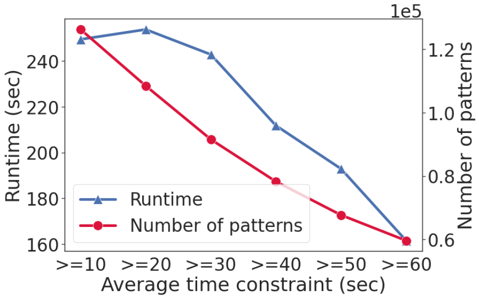

Figure 1: The runtime (y-axis-left) and the number of patterns found (y-axis-right) with varying constraints (x-axis).

4.7 Runtime Performance

We report runtime performance of pattern mining on a machine with Linux RHEL7 OS, 16-core 2.2GHz CPU, and 64 GB of RAM. We apply Seq2Pat on the positive set with 8,329 clickstream sequences. We impose the same types of constraint as described in Section 4.2 while we vary the constraint on the minimum average time spent on pages. To stress test the runtime, we set the minimum frequency $\theta = 2$ which returns almost all the feasible patterns. Figure 1 shows the runtime in seconds (y-axis-left) and the number of patterns found (y-axis-right) as the average constraint increases (x-axis). As the constraint becomes harder to satisfy, the number of patterns goes down as expected. The runtime for the hardest case is ~250 seconds while we observe speed-up as constraint reasoning becomes more effective.

4.8 Feature Importance

Finally, we study feature importance to drive high-level insights and explanations from auto-generated Seq2Pat features. We examine the Shapley value (Lundberg et al. 2020) of features from the LightGBM model.

Figure 2 shows the top-20 features with highest impact. Our observations match previous findings in (Requena et al. 2020). The pattern $\langle 3, 1, 1 \rangle$ provides the most predictive information, given that the symbol (3) stands for adding a product. Repeated page views as in $\langle 1, 1, 1, 1, 1, 1, 1 \rangle$, or specific product views, $\langle 2, 1, 1, 1 \rangle$ are indicative of purchase intent, whereas web exploration visiting many products, $\langle 1, 1, 2, 1, 2 \rangle$, are more negatively correlated to a purchase. Interestingly, searching actions $\langle 6 \rangle$ have minimum impact on buying, raising questions about the quality of the search and ranking systems. Our frequent patterns also yield new insights not covered in the existing hand-crafted analysis. Most notably, we discover that removing a product but then remaining in the session for more views, $\langle 4, 1, 1 \rangle$ is an important feature, positively correlated with a purchase. This scenario, where customers have specific product needs, hints at missed business opportunity to create incentives such as prompting virtual chat or personalized promotions.

5 Conclusion

Pattern mining is an essential part of data analytics and knowledge discovery from sequential databases. It is a powerful tool, especially when combined with constraint reasoning to specify desired properties. In this paper, we presented

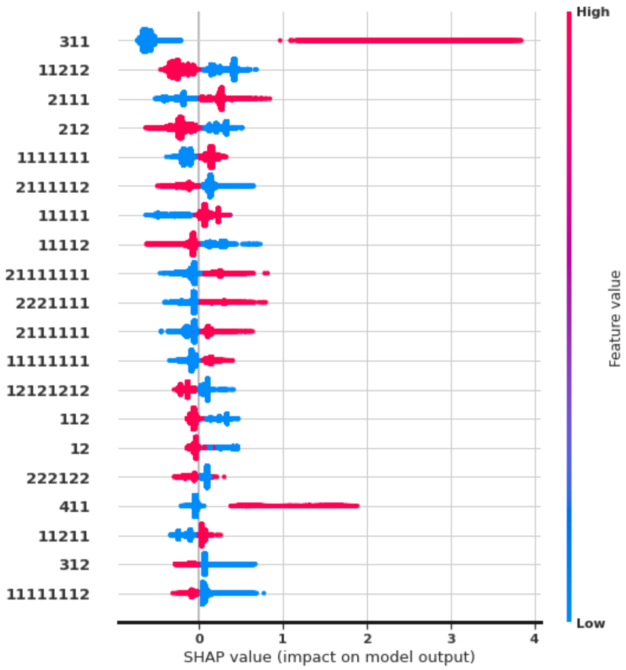

Figure 2: SHAP values of auto-generated Seq2Pat features. Top-20 features ranked in descending importance. Color indicates high (in red) or low (in blue) feature value. Horizontal location indicates the correlation of the feature value to a high or low model prediction.

a simple procedure for Dichotomic Pattern Mining that operates over semi-structured clickstream datasets. The approach learns new representations of pattern embeddings. This representation enables simple models, which cannot handle sequential data by default, to predict from sequences. Moreover, it boosts the performance of more complex models with feature augmentation. Experiments on customer intent prediction from fashion e-commerce demonstrate that our approach is an effective integrator between automated feature generation and downstream tasks. Finally, as shown in our feature importance analysis, the representations we learn from pattern embeddings remain interpretable.

References

Chen, E.; Cao, H.; Li, Q.; and Qian, T. 2008. Efficient Strategies for Tough Aggregate Constraint-Based Sequential Pattern Mining. *Information Sciences*, 178: 1498–1518.

Hosseininasab, A.; van Hoeve, W.; and Ciré, A. ???? Constraint-Based Sequential Pattern Mining with Decision Diagrams.

Lundberg, S. M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J. M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; and Lee, S.-I. 2020. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature Machine Intelligence*, 2: 56—67.

Requena, B.; Cassani, G.; Tagliabue, J.; Greco, C.; and Lacasa, L. 2020. Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Scientific Reports*, 2020.

Wang, X.; Hosseininasab, A.; Pablo, C.; Serdar, K.; and van Hoeve, W.-J. 2022. Seq2Pat: Sequence-to-Pattern Generation for Constraint-based Sequential Pattern Mining. In *To appear in AAAI-IAAI'22*.