Recognition of Sequence of Activities - The Log-Loss Sequence Model

Eldor Shir & Roi Zilberzwaig

Advisor: Dr. Izack Cohen

October 2021

1 Introduction

Process mining includes several techniques related to the fields of data science and process management to support process analysis based on event logs - details of the process performed and time stamps for each action, i.e. start time and end time. A process consists of a sequence of actions that maintain a relationship between them. For example, an action that precedes another action or an action that can be performed in parallel with another action. Also, there are processes where the order of actions, or some of them, is significant and there are processes where the order does not matter. For example, making coffee can be prepared in some ways and forms, some will pour water first, and only then will put coffee and some will put coffee, and only then they will pour water. In any case, pouring the water and coffee can only be done after the coffee maker has taken out a cup which is an action that must be done before pouring water to make coffee.

In our project, we will focus on video data, as today cameras are located in every corner of the world. Improving the analysis of this information may lead to a great contribution in a variety of fields.

The purpose of the project is to classify the videos using conformance checking. Conformance checking relates events in the event log to activities in the process model and compares both. The goal is to find commonalities and discrepancies between the modeled behavior and the observed behavior.

1.1 Background

In recent years there has been a lot of interest in video analysis. This is due to a variety of reasons. The main reasons include the increasing use of cameras everywhere (shops, streets, private homes, etc.) and the significant impact that will be created when we can analyze data of this type. Given the probability matrix of a video, we will strive to identify and categorize the process shown in

the video. This is done by correctly labeling the various frames based on the given probability matrix.

One of the most common solutions today for dealing with this type of problem is to create labels using a softmax operation (our predicates) and examine the distance to a given labeling sequence - Ground Truth (GT). There are several metrics for calculating the distance between different labeling sequences.

One of the major challenges in this type of problem is the different lengths of videos representing the same processes. For example, a given action can be long and performed slowly, while it can be very short if performed quickly.

One of the proposed solutions to deal with this problem is to condense the probability matrix by inserting vectors of zeros at the end of the matrix. In this way, we can compare the length of the probability matrix and the given labeling sequence of the action (GT).

1.2 The SF-NET system

As part of the project we used the SF-NET system [1]. This system is weakly-supervised, which means that the video is tagged by watching a single time - without having to watch it again. This significantly reduces CPU utilization and runtime, which can be meaningful when it comes to a massive amount of data. In addition, there are three major innovations in this system:

1. Create a probability for a particular tag for each of the frames sampled in the video.

2. Use of an innovative algorithm to distinguish between background frames and actions.

3. Using an algorithm in an attempt to create certainty in labeling while raising the probability of labeling given as an action (rather than as a background).

Unlike our goal in the project, the system is not intended to identify the process (complex activity), but to identify the individual frames sampled from the video. We used this system as will described later.

1.3 Process mining

Process mining is a computerized method for analyzing and understanding complex business processes operating in organizations. The need for process mining stems from the growing recognition by management of the importance of quality business processes for the success of the organization, while at the same time they often find it difficult to identify and evaluate how these processes actually work.

In many organizations, computerized systems used to carry out business processes contain event logs (log files) that chronologically document the activities performed. These logs manage for each instance, i.e. a private case of a business

process, information that includes, among other things, the details of the events performed by the users in its framework.

In our project, we used videos that do not include data event logs. We use the SF-NET system that helps us to obtain a probability matrix - represents all the sampled frames. Using the matrix we create a log file, while by manipulating the probability matrix we aim to arrive at a more accurate event log.

Process mining techniques contain three main categories (which can be used together):

1. Process Discovery - The purpose of this algorithm is to learn how the actual process is performed in the various instances and to identify those that deviate from its normal model. In the first stage, the algorithm identifies the different variants of the process, i.e. the different paths in which it was actually performed. The analysis of the log files makes it possible to identify those that violate the rules of the process. The most "interesting" log files for the visitor are those that are low frequency.

2. Conformance checking - a comparison between events in the event log as sampled from the process (videos in our project) and the actions in the process model (GT). The goal is to find common ground alongside discrepancies between the behavior in the model and the behavior observed (in the video).

3. Performance Analysis - This technique is used when there is an a priori model. The model is expanded with additional performance information such as processing times, cycle times, standby times, etc. So the goal is not to test compatibility but to improve the performance of the existing model relative to the performance of certain processes.

As will be explained in detail in the Method chapter, in the project we focused on the Conformance checking technique.

There are several approaches to finding a match between a viewed event log and a given model (GT). In order to meet the challenge of the changing lengths of the videos, we used the project in the Alignment technique [2, 3]. The idea in this technique is that the algorithm performs an exhaustive search in order to find the optimal "alignment" between the observed event log (the prediction based on the softmax matrix) and the process model (given GT). In this way, it is guaranteed to select the "closest" model to the event log (and thus actually characterize the process).

2 Method

2.1 Motivation

In most if not all machine learning models there is a reduction in the dimensions, usually with the help of using the argmax function, which leads to loss of

information. As part of our project, we will try to minimize the loss of information by adding another layer using the conformance checking technique from the field of process mining. Thus, our framework focuses on the stage before turning a matrix of probabilities into a vector.

2.2 Problem Definition

We start with a ground truth list GTs comprising T labeled videos of R processes $GTs = \{GT_i | i = 1, 2, .., T\}$, and a probability matrix of a video $PM \in \mathbb{R}^{MxN_j}$ where M is the number of action labels and N_j is the number of frames in the video. Each vector $GT_i \in \mathbb{R}^{1xN_i}$ contains action label per frame as N_i is the number of labeled frames in GT_i. Our goal is to classify the process appearing in the video which is represented by the PM, i.e. our output (prediction) Y held $Y \in \{0, 1, ..., R-1\}$.

For this purpose we suggest two models - the Sequence Model (SM) and the Log-Loss Sequence Model (LLSM).

2.3 Sequence Model

The basic model, serves as a basis for comparison with the LLSM, is divided into 3 parts, as presented in Figure 1:

2.3.1 Argmax on PM

Given Probability Matrix (PM) we using an argmax operation on it to obtain a vector $argmax(PM) \in \mathbb{N}^{N_j}$ that includes the prediction tags of the model per frame $j \in N_j$.This, in order to make a comparison of the GTi vector.

2.3.2 LCS Algorithm

Given GT, $argmax(PM)$ from previous part and 3 scores as hyper parameters - MatchScore, MismatchScore and GapScore which represent the score awarded for a match, mismatch and adding a token respectively, we perform the Longest Common Subsequence (LCS) algorithm [4] in order to compare $argmax(PM)$ with each GTi as described below:

$$Score_{i,j} = \begin{cases} 0 & \text{i=0 or j=0} \\ max(XY_{i,j}, TokenX_{i,j}, TokenY_{i,j}) & otherwise. \end{cases} \quad (1)$$

where, we define $XY_{i,j}$, $TokenX_{i,j}$ and $TokenY_{i,j}$:

$$XY_{i,j} := \begin{cases} Score_{i-1.j-1} + MatchScore & X_i = Y_j \\ Score_{i-1.j-1} + MismatchScore & X_i \neq Y_j \end{cases}$$

$$TokenX_{i,j} := Score_{i,j-1} + GapScore$$

$$TokenY_{i,j} := Score_{i-1,j} + GapScore$$

The $Score_{N_i,N_j}$ is the conformance between the GTi and the PM.

2.3.3 Argmax among all scores

For each comparison between $argmax(PM)$ and GTi we received a score. At this point we will select our prediction by selecting the appropriate GTi that received the highest score.

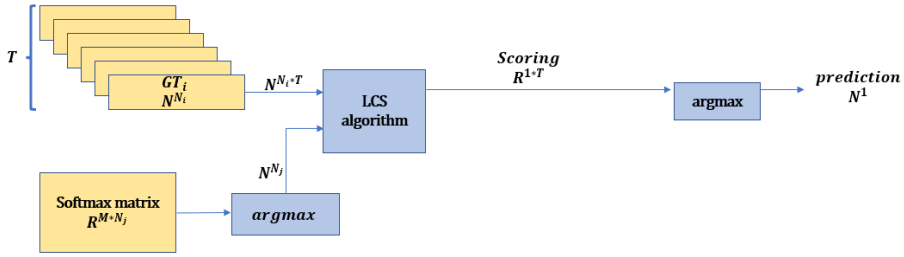

Figure 1: Scheme of the $SequenceModel$: (i) Perform an argmax operation on the PM. (ii) Use the LCS algorithm as alignment to deal with the challenge of the different lengths of vectors. (iii) Perform an argmax operation among all the score results we received in to get the prediction.

2.4 Log-Loss Sequence Model

The improved model includes 5 parts, as shown in Figure 2:

2.4.1 Argmax on PM

Given PM we using an argmax operation on it to obtain a vector $argmax(PM) \in \mathbb{N}^{N_j}$ that includes the prediction tags of the model per frame $j \in N_j$.This, in order to make a comparison of the GTi vector.

2.4.2 LCS Algorithm

In the LLSM we extended the algorithm and used a direction function. The purpose of the direction function is to bring us the optimal way, with the highest score, to create alignment between the GTi and the $argmax(PM)$. The direction function return an arrow symbol of the path that will maximize the score.
more formally:

$$Direction_{i,j} = \begin{cases} \nwarrow, & XY_{i,j} = max(XY_{i,j}, TokenX_{i,j}, TokenY_{i,j}) \\ \leftarrow, & TokenX_{i,j} = max(XY_{i,j}, TokenX_{i,j}, TokenY_{i,j}) \\ & \text{and } TokenX_{i,j} \neq XY_{i,j} \\ \uparrow, & TokenY_{i,j} = max(XY_{i,j}, TokenX_{i,j}, TokenY_{i,j}) \\ & \text{and } TokenY_{i,j} \neq XY_{i,j} \\ & \text{and } TokenY_{i,j} \neq TokenX_{i,j} \end{cases} \quad (2)$$

$XY_{i,j}$, $TokenX_{i,j}$ and $TokenY_{i,j}$ are as defined in the SM.

Using this function we extend X and Y to $\hat{X}$ and $\hat{Y}$ respectively when held $\hat{X}, \hat{Y} \in \mathbb{N}^{N_k}$. The arrow returned by $Direction_{i,j}$ are interpreted as follows:

- Diagonal arrow ($\nwarrow$): X_i and Y_j will receive the same frame index in the extended sequences - $\hat{X}$ and $\hat{Y}$.
- Left arrow ($\leftarrow$): Y_j and a token will receive the same frame index in $\hat{Y}$ and $\hat{X}$ respectively.
- Up arrow ($\uparrow$): X_i and a token will receive the same frame index in $\hat{X}$ and $\hat{Y}$ respectively.

Regarding X as the $argmax(PM)$ and Y as the GTi we obtain the traces for the best score founded for this two sequences.

2.4.3 PM Extension

In this part, we extend the PM according to the extended $argmax(PM)$ such that on every frame that have a token on the extended $argmax(PM)$ we will add a column to the PM to get the extended PM - $P\hat{M} \in \mathbb{N}^{MxN_k}$.

2.4.4 Log-Loss Calculation

Having extended PM and GTi to same length, we are summing the logs of the PM which are selected according to the GTi.
formally:

$$LogLoss(PM, GTi) = -\sum_{l=0}^{N_k-1} Log(PM_{l,GTi_l})$$

2.4.5 Argmin among all log-loss calculations

Comparing all log-loss calculations, we choose the process of the GTi with the lowest log-loss as our prediction.

$$\arg\min_{t \in T} \{LogLoss(PM, GT_t)\}$$

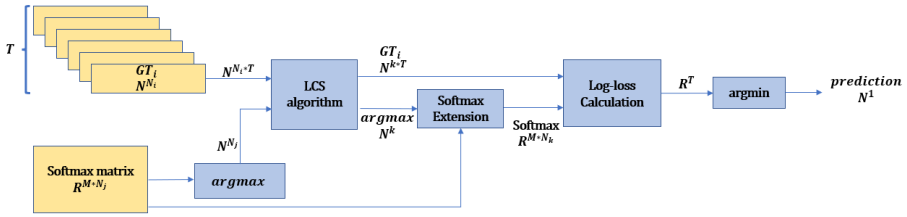

Figure 2: Scheme of the log loss sequence model: (i) Perform an argmax operation on the PM as we performed in the SM. (ii) Use LCS algorithm as alignment as in SM with an addition of the direction function in order to extend $argmax(PM)$ and GTi to the same length. (iii) Extend the PM according to the extended $argmax(PM)$. (iv) Calculate the log-loss. (v) Get our prediction by performing an argmin operation among all log-losses.

3 Experiment

3.1 Dataset

As part of the project, we worked on a GTEA dataset [5] that includes 28 videos lasting about half a minute to a minute. The videos are divided into quartets, i.e. 7 quartets, with each group representing a different process. This dataset focuses on various kitchen operations, such as making tea, making a sandwich with cheese, etc. Each video representing a process that is performed by four different people.

It should be noted that the GTEA dataset includes Ground Truth (GT) for each of the videos that contain appropriate tags for each of the sampled frames. This is very significant for our work which focuses on the conformance checking technique to make the comparison to the predictions of the model.

First, We performed an extensive pre-processing on the GTEA dataset, including examining the lengths of the videos, representing a graph for a video, and turning the numerical tags into action names.

There are a total of 8 possible tags: stir, scoop, pour, take, close, put, open and background. We will notice that the background tag was set only in case the system failed to characterize the frame as one of the other tags.

3.2 Implementation Details

In our work, we used the SF-NET system to receive the PM. For the LCS algorithm we based on the dynamic implementation which its time and memory complexity is $\Theta(M * N)$, where M and N are the length of the sequences. Since we compare the PM with T ground truths, the total complexity is $\Theta(T * M * N)$, which is both models complexity since the LCS Algorithm is our bottleneck. In addition for the LCS algorithm, we set three hyper-parameters as mentioned in section 2.3.2: MatchScore, MismatchScore and TokenScore to 3, -2 and -2 respectively in the SM and 10, -1 and -5 respectively in the LLSM. These hyper-

parameters achieved the highest accuracy out of a variety of permutations, when only assuming non-negative number to MatchScore. For accuracy testing, since we worked with an inadequate dataset we compared each process against all processes but itself.

3.3 Results

We examined both models using an accuracy index. In SM we got an accuracy of 56.25% while in LLSM we got an accuracy of 62.5%. It seems that the improved model has succeeded in bringing about a better result although not noticeably. Below, we suggest various options for improving the accuracy of our model.

Table 1: Description of model results

Video name	Sequence model results	Log-loss sequence model results
S1_Peanut_C1	True	True
S2_Peanut_C1	True	True
S3_Peanut_C1	True	False
S4_Peanut_C1	True	False
S1_Hotdog_C1	True	False
S2_Hotdog_C1	True	True
S3_Hotdog_C1	False	True
S4_Hotdog_C1	True	True
S1_Pealate_C1	True	True
S2_Pealate_C1	False	True
S3_Pealate_C1	True	True
S4_Pealate_C1	False	True
S1_Cheese_C1	False	False
S2_Cheese_C1	False	True
S3_Cheese_C1	False	False
S4_Cheese_C1	False	False
Summary	56.25%	62.5%

4 Summary

During our project, we presented two models based on the Conformance checking technique. A basic model, known as the "sequence model" and an advanced model is known as the "log-loss sequence model". In the advanced model, we were able to achieve better results by about 6% (56.25% compared to 62.5% accuracy).

4.0.1 Conclusions

We understand that several factors may affect the model and lead to better results:

1. Enlarging the dataset - In the project, we used part of the GTEA dataset containing 28 videos. Because we focused on different videos of making

sandwiches, we worked on a total of 16 videos. We assume that if we expand the dataset, we can get different modeling of the same process and thus improve the results. The dataset we worked on contained videos of processes and GTs, which is hard to find.

2. Attempt to work around the use of argmax - Throughout the project, we tried to think about how to replace the operation of argmax that is usually performed on the probability matrix. This, in order not to lose information (in the transition between the matrix to a vector of labels). In practice, in both models, we used the argmax operation. We estimate that if we succeed in finding an alternative way of working on the matrix of probabilities, there may be better results.

During the work, we were exposed to the field of process mining in general, and the conformance checking technique in particular. We understand that if there is significant progress in the field of video processing, it will revolutionize a wide variety of fields.

4.0.2 Further Work

Our work is the infrastructure and basis for further extensive work in the field. In doing so, we thought of some directions that should be examined below and see how they will affect the results (from a starting point that these may bring about significant improvement):

1. Differentiation of the costs of inserting the tokens:
   (a) Create a distinction between inserting a token when a parallel action is classified as background, and inserting a token in a situation where the corresponding activity is not the background. This thinking assumes that inserting a token on a background action is less significant (and therefore will have a lower cost), since it may be an intermediate operation (i.e., an unidentified action performed between different actions).
   (b) Providing a low cost for token insertion in case there were previously, near the same point, a sequence of identical actions. This assumption is based on the fact that there are videos of different lengths that represent the same processes. Therefore, it may be the same process except that in one video the action is done quickly, while in the other video the action is performed slowly.

2. Changing the log-loss calculation - In our work we did not calculate the loss cost at all when we entered a token. Creating a cost may improve the forecasting process. Such a case is not trivial because in this case we have to think what is the appropriate cost for such a case. This is in contrast to standard labeling where the cost of loss is calculated according to the probability we have obtained.

3. Giving a relative score according to the length of the videos - because there are videos of different lengths, they have the potential, on the one hand, to rake in more points and on the other hand to "punish" more. It is possible that the creation of a normalization of the score between all the videos may also affect the forecasting of the processes.

5 Code

Our code is based on the article code "SF-Net: Single-Frame Supervision for Temporal Action Localization ". We expanded the code by using the LCS algorithm to create the alignment. We matched the algorithm so that it could get two vectors of different lengths and return the optimal alignment. This, depending on the score we bring to each selected step (match, mismatch or insert a token). In addition, we added log-loss calculations. The link to our code is at the following link: https://github.com/RoiZilberzwaig/The-Log-Loss-Sequence-Model.

References

[1] Linchao Zhu Fan Ma, Shengxin Zha Yi Yang, Matt Feiszli Gourab Kundu, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. -, 2020. 2

[2] Wil van der Aalst. Process mining : data science in action. -, 2016. 3

[3] Carmona and Solti abd Weidlich van Dongen. Conformance Checking - Relating Processes and Models. Springer, 2018. 3

[4] Cormen, Rivest Leiserson, and Stein. Introduction to Algorithms, 3rd Edition. Mit Press, 2009. 4

[5] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Georgia tech egocentric activity datasets. http://cbs.ic.gatech.edu/fpv, 2017. 7