Prediction of Batch Processes Runtime
Applying Dynamic Time Warping and Survival Analysis

Paolo Graniero[1] and Marco Gärtler[2]

[1] Università La Sapienza, Rome, Italy
paolograniero8@gmail.com
[2] ABB AG, Corporate Research Germany, Ladenburg, Germany
marco.gaertler@de.abb.com

Abstract. Batch runs corresponding to the same recipe usually have different duration. The data collected by the sensors that equip batch production lines reflects this fact: time series with different lengths and unsynchronized events. Dynamic Time Warping (DTW) is an algorithm successfully used, in batch monitoring too, to synchronize and map to a standard time axis two series, an action called alignment. The online alignment of running batches, although interesting, gives no information on the remaining time frame of the batch, such as its total runtime, or time-to-end. We notice that this problem is similar to the one addressed by Survival Analysis (SA), a statistical technique of standard use in clinical studies to model time-to-event data. Machine Learning (ML) algorithms adapted to survival data exist, with increased predictive performance with respect to classical formulations. We apply a SA-ML-based system to the problem of predicting the time-to-end of a running batch, and show a new application of DTW. The information returned by open-ended DTW can be used to select relevant data samples for the SA-ML system, without negatively affecting the predictive performance and decreasing the computational cost with respect to the same SA-ML system that uses all the data available. We tested the system on a real-world dataset coming from a chemical plant.

Keywords: Batch Process Monitoring · Dynamic Time Warping · Time-to-end Prediction · Survival Analysis

1 Introduction

Batch production is a common manufacturing technique for chemical, pharmaceutical and food industries where a given product is realized in a stage-wise manner following a given formula or recipe. Such recipe defines rigorously the sequence of steps that each batch run follows. Nonetheless, it is quite common that distinct batch runs for the same recipe differ in several aspects, some of them time-related. For example, total duration or duration of the individual subphases can vary and, even in synchronized phases, significant events or alarms can happen at different relative positions in time. The magnitude of such variations requires advanced process monitoring techniques to ensure the consistency

J. Beyerer et al. (Hrsg.), *Machine Learning for Cyber Physical Systems*, Technologien für die intelligente Automation 13,
https://doi.org/10.1007/978-3-662-62746-4_6

and the quality of the product, to improve the safety levels of the plant, and to better understand and control the process [16]. This monitoring is, in turn, beneficial to the process operations, planning and can lead to overall improvements. A particular branch of process monitoring is statistical process monitoring: these techniques rely on mathematical tools that help to identify and control variations in the production process, analyzing data coming from the reactors. This data, collected from a large number of sensors, comes with some peculiar characteristics as a consequence of the temporal variability stated above, affecting the analysis performed. Many established statistical techniques for process monitoring, such as Multiway Principal Component Analysis (MPCA), require that the input data have equal length, that for a batch process is its duration. As stated above, even in case of the same duration, significant events could be asynchronous, leading to problems during the analysis, such as comparing different but synchronous events. These issues are not unique to batch data: Dynamic Time Warping (DTW), originated in signal processing [19], is a widely successful and often adopted technique to homogenize data with a different time frame to a standard duration and synchronization of characteristics. An adoption to batch data is given, for example, in [9].

There exist different versions of DTW. In batch monitoring, standard DTW is used to align two completed batches, mainly during offline analysis when completed batches are available. The open-ended version is useful in online scenarios, where a running, and therefore incomplete, batch is aligned to a prefix of a completed batch. As such, a comparison between the running batch and a historical one is straightforward. Unfortunately, such alignment does not give any information on the remaining time frame; for example, the time left before completion of the batch, the focus of our work. This specific issue can be seen as a time-to-event modeling problem, addressed by techniques such as Survival Analysis (SA). SA is a standard adopted technique in clinical studies to model the time until the occurrence of an event of interest, usually linked to the course of an illness. Many machine learning (ML) algorithms, like the ones described in [6], have been adapted to handle survival data, resulting in improved predictive performance over more standard techniques.

In this work, we propose a system combining DTW and SA that aims at predicting the total duration of a running batch given historical information on the process. We first check the feasibility of using SA in the context of batch monitoring. Then we investigate if by applying DTW, it is possible to obtain any improvement regarding the predictive performance or the computational cost. As described in Section 5, we apply DTW, not as a time normalization technique or to compute a distance measure, but as a data-selection tool: we use the mapping information contained in the warping path, one of the algorithm's output, to select only a fraction of the data available. This data is used to train a SA-ML-based algorithm that, followed by a standard regression model, returns an estimate of the time-to-end of a running batch. We show that the same SA model trained only on the fraction of data selected via DTW performs as good as the same model trained on all the data available, sometimes even better. A significant improvement is observable in the computational cost of the approach.

The rest of the paper is organized as follows. Section 2 and 3 expose some useful facts about Dynamic Time Warping and Survival analysis, respectively. Section 4 briefly describes the batch data used in evaluating the system proposed, and some data preprocessing realized. Section 5 describes in detail the proposed system, while Section 6 shows an evaluation of the results obtained. Finally, Section 7 concludes the paper.

2 Dynamic Time Warping

Dynamic Time Warping (DTW) is the name of a class of algorithms used to compare two series of values with possibly different length; the main applications regard time series. The idea behind this technique is to stretch and compress the two series to make one resemble the other as much as possible. This warping accounts for non-linear fluctuations of the time axis.

Once warped, the two series have the same length, and similar patterns are aligned. These are the time-normalization and event synchronization effects, respectively.

The main objects of interest are two: the warping path and the DTW distance. The warping path is a mapping between the time indexes of the two series. We can interpret multiple correspondences between one index on a series and multiple ones on the other as the stretches and compressions mentioned above. The warping path is the optimal mapping that minimizes the distance between the warped series: this minimal distance is the so-called DTW distance.

Usually, when using DTW, we identify a reference series and a query one. The usual set-up is to select only one reference Y to which we align several queries $X^i,\ i \in \{1, \ldots, I\}$.

The standard version of DTW aligns the two series in their entirety, mapping the end-points of the two series to each other. Formally, if

$$Y = (y_1, \ldots, y_N) \qquad X = (x_1, \ldots, x_M)$$

then the standard alignment satisfies the following condition for the end-points

$$x_1 \to y_1 \qquad x_M \to y_N$$

On the other hand, the open-ended version aligns a query to a reference's prefix:

$$x_1 \to y_1 \qquad x_M \to y_n,\ 1 \leq n \leq N \tag{1}$$

In this work, we mainly use the open-ended version, that is suited for online applications, being able to align an incomplete batch to a reference one.

DTW has been applied for time series classification, clustering, in various domains. References to such applications can be found in [5]. In these applications, the quantity of interest is mainly the DTW distance.

First applications of DTW to batch process monitoring can be found in [17,9]. The emphasis of these works is on the alignment of batch data, but only offline (completed batches). Online applications of DTW are present in the literature, for example in [4], mainly in conjunction with more advanced monitoring techniques as MPCA. In this work we use open-ended DTW as a data-selection tool: we use the index n in Equation 1 to select the points $x^i_{N_i}$ of the historical batches that were mapped to the same point. We suppose that these points are the ones containing most of the relevant information to the time-to-end prediction at the given time on the running batch.

3 Survival Analysis

Survival Analysis (SA) is a sub-field of statistics where the goal is to analyze and model the data where the outcome is the time until the occurrence of an event of interest [20]. The main feature of SA is the ability to deal with censored data, that is when the event could be unobserved in the time-frame considered in an experiment. Standard approaches to SA allow to model the time to the event of interest and obtain an estimate of it. The statistical modeling is the focus of classical approaches, while their predictive performance is limited. More recent developments focus on machine learning approaches to SA [6], adapting loss functions of known algorithms to the specific problem of SA, time until the event of interest. For survival models that do not rely on Cox's proportional hazards assumption [3], the predictions are risk scores of arbitrary scale and not the actual time-to-event. If samples are ordered according to their predicted risk score (in ascending order), one obtains the sequence of events, as predicted by the model [2]. In this work, we use one of these algorithms, Gradient Boosting Survival Analysis [6,1]; we apply a standard regression algorithm to convert the predicted risk to an actual time-to-event estimate.

4 Data

The data used to test the proposed system comes from a chemical batch production line. The data consists of 383 batches, spanning three years (2015-2017) of measurements, with a duration between 146 and 960 minutes. We represent each batch as a multivariate time series: each dimension is a process variable (PV), with a standard sampling rate of 1 value per minute for every batch and every PV. The PVs come from different sensors that equip the production line: they can represent engineering variables, control variables, or state variable. The exact nature of the PVs is confidential.

Data normalization. The application of DTW requires the data to be normalized to avoid artifacts due to different scales of the values. When applying DTW, we have to select a batch as reference, to which we align every other batch. To have a coherent online normalization, we decided to use a Min-Max scaling approach, and as the normalizing interval, we chose the range of values of each PV in the reference batch. This choice makes the normalization of the data coming from the running batch straightforward.

Reference batch and significant features. For every set of batches considered during the experiments, we had to select one batch as reference, considered the typical one. Following domain expert's advice, we chose the batch with median duration. Since constant variables have no warping information, we disregarded PVs with a constant trend in the reference, removing them from the rest of the batches too.

5 Proposed System

The system proposed in this work, which we call SA+DTW-system, is schematically represented in Figure 1. It has two phases: offline and online.

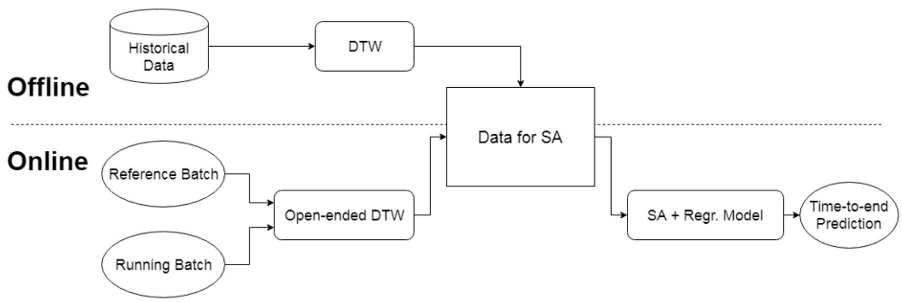

Fig. 1: Overview of the proposed system

During the offline phase, the historical data about previous batches is aligned to the selected reference batch using open-ended DTW on every prefix of the batches, storing the information about DTW distance and alignment. The resulting dataset contains the following information:

- The current time index on the running batch
- The index n of Equation 1. It's the end point of the reference's prefix to which the running batch has been aligned
- The DTW distance between the query batch and the reference's prefix
- The value of every PV of the query batch at the given time
- The time-to-end of the batch. This quantity is the target variable of the model

During the online phase, the running batch is aligned to the same reference as in the offline phase via open-ended DTW. From this alignment, we get two pieces of information: the index n of the mapped prefix on the reference and the DTW distance. n is used to filter the dataset obtained offline: only the entries mapped to the same n are selected and kept for the next step. This step consists of training a SA-ML-based model to learn the risk score based on the features mentioned above. The model is then able to assign a risk score to the running batch. This risk score is then converted to an actual time-to-end estimate by a regression model (Random Forest) trained on the risk scores assigned to the filtered dataset.

Table 1: SA-system stats

Year	2015	2016	2017
Training dataset size [rows × columns]	39431×28	77989×34	79833×34
Training time [minutes]	34	126	134
Single prediction time [seconds]	~ 0.6	~ 1.1	~ 1.1

Table 2: SA+DTW-system stats

Year	2015	2016	2017
Average dataset size [rows × columns]	149×28	180×34	131×34
Single prediction time [seconds]	< 1	< 1	< 1

Software used We used the Python [18] programming language to develop the whole system. The python packages we used are: Numpy [11], Scikit-learn [12], SciPy [8], Matplotlib [7], Pandas [10], Scikit-survival [14,15,13]

6 Results

The data set had distinct characteristics in each year. Thus we split the overall data set into three parts and considered them on their own. For each year we considered approximately the first 2/3 of the batches as historical data, and the remaining 1/3 of the batches was used to test the system. In particular, we have for the three years considered the following historical/test batches: 80/40, 101/50, 75/37.

The focus of this work is on the improvements that can be obtained selecting the data via DTW, in particular before applying SA. We compared three methods to this end. The first method uses the average duration of the batches longer than the running batch as an estimate for the total duration of the running batch. This method uses only historical information, and it is considered as a minimum performance to be reached by any system to be considered useful. The second method, the SA-system, uses all the historical data at once as training data for a SA-ML based algorithm, followed by a regression model (Random Forest) for the conversion of the risk score to time-to-end estimate. All the predictions obtained by this method come from the same model. The third method is the system proposed, the SA+DTW-system described in Section 5. Figure 2 shows the results for the 2017 data. The results of the other two years show no significant differences from this one.

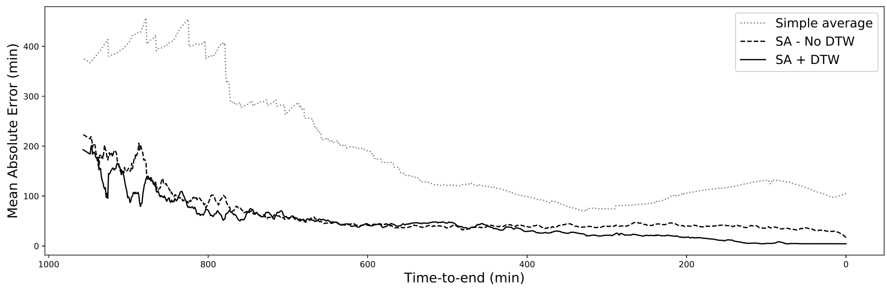

Fig. 2: Comparison of the performance (mean absolute error) of the three models tested (2017 data). The SA+DTW-system shows a comparable performance to the SA-system

7 Conclusion

The results from last section show two main facts. Firstly, Survival Analysis is suited to be used in batch process monitoring. The information contained in the process variables can be used to model the time-to-end of a batch process, at least for this type of process, with a sufficiently good performance, as judged by some domain experts to which the results were shown. Secondly, the mapping between running and reference batch performed by Dynamic Time Warping is an effective tool to select relevant data from the historical one. The consistent performance between the SA and the SA+DTW system can be interpreted as a good action of the DTW mapping in reducing the noise contained in the data: the relevant information to model the time-to-end is retained when cutting out, on average, more than 99% of the samples available.

These results represent only a first approach to this problem. We think that many improvements could be obtained with careful tuning of the system. Firstly, we used non-optimized parameters for the SA-ML algorithm: the definition of a global performance metric could help in choosing optimal parameters for a given process. Then, the DTW mapping has been performed using the standard version of DTW, but more sophisticated approaches are already present in literature, such as shape-DTW [21]. We think that the performance on the data-selection side could improve given more stable alignment since the version of the algorithm used is susceptible to the noise in the process variables. We think that taking into account this noise is an effective way to stabilize the online alignment and possibly remove the need to aggregate successive predictions of the system to stabilize them.

References

1. Gradient boosting survival analysis. `https://scikit-survival.readthedocs.io/en/latest/generated/sksurv.ensemble.GradientBoostingSurvivalAnalysis.html`, accessed: 2019-06-14

2. Understanding predictions in survival anlysis. https://scikit-survival.readthedocs.io/en/latest/understanding_predictions.html, accessed: 2019-06-14
3. Cox, D.R.: Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) **34**(2), 187–202 (1972)
4. Gao, X.: On-line monitoring of batch process with multiway pca/ica. In: Principal Component Analysis. IntechOpen (2012)
5. Giorgino, T., et al.: Computing and visualizing dynamic time warping alignments in r: the dtw package. Journal of statistical Software **31**(7), 1–24 (2009)
6. Hothorn, T., Bühlmann, P., Dudoit, S., Molinaro, A., Van Der Laan, M.J.: Survival ensembles. Biostatistics **7**(3), 355–373 (2005)
7. Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in science & engineering **9**(3), 90–95 (2007)
8. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–), http://www.scipy.org/, [Online; accessed: 2019-06-14]
9. Kassidas, A., Taylor, P.A., MacGregor, J.F.: Off-line diagnosis of deterministic faults in continuous dynamic multivariable processes using speech recognition methods. Journal of Process Control **8**(5), 381 – 393 (1998). https://doi.org/10.1016/S0959-1524(98)00025-0, aDCHEM '97 IFAC Symposium: Advanced Control of Chemical Processes
10. McKinney, W., et al.: Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. vol. 445, pp. 51–56. Austin, TX (2010)
11. Oliphant, T.E.: A guide to NumPy, vol. 1. Trelgol Publishing USA (2006)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
13. Pölsterl, S., Gupta, P., Wang, L., Conjeti, S., Katouzian, A., Navab, N.: Heterogeneous ensembles for predicting survival of metastatic, castrate-resistant prostate cancer patients. F1000Research **5** (2016)
14. Pölsterl, S., Navab, N., Katouzian, A.: Fast training of support vector machines for survival analysis. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 243–259. Springer International Publishing, Cham (2015)
15. Pölsterl, S., Navab, N., Katouzian, A.: An efficient training algorithm for kernel survival support vector machines. arXiv preprint arXiv:1611.07054 (2016)
16. Ramaker, H., van Sprang, E.: Statistical batch process monitoring (01 2004)
17. Ramaker, H.J., van Sprang, E.N., Westerhuis, J.A., Smilde, A.K.: Dynamic time warping of spectroscopic batch data. Analytica Chimica Acta **498**(1), 133 – 153 (2003). https://doi.org/10.1016/j.aca.2003.08.045
18. van Rossum, G.: Python tutorial. Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam (May 1995)
19. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing **26**(1), 43–49 (1978)
20. Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: A survey. arXiv preprint arXiv:1708.04649 (2017)
21. Zhao, J., Itti, L.: Shapedtw: shape dynamic time warping. Pattern Recognition **74**, 171–184 (2018)