Analysis of Time Dependent Data and PRA

D. Mandelli†, D. Maljovec, A. Alfonsi, C. Smith, C. Rabiti

Idaho National Laboratory, 2525 North Fremont Street, Idaho Falls (ID)

† *Corresponding author: diego.mandelli@inl.gov*

INTRODUCTION

In the past decades, several numerical simulation codes have been employed to simulate accident dynamics (e.g., RELAP5-3D [1], MELCOR [2], MAAP [3]).

In order to evaluate impact of uncertainties into accident dynamics several stochastic methodologies have been coupled with these codes. These stochastic methods range from classical Monte-Carlo and Latin Hypercube sampling to stochastic polynomial methods.

Similar approaches have been introduced into the risk and safety community where stochastic methods (such as RAVEN [4], ADAPT [5], MCDET [6], ADS [7]) have been coupled with safety analysis codes in order to evaluate the safety impact of timing and sequencing of events. These approaches are usually called Dynamic PRA or simulation-based PRA methods.

These uncertainties and safety methods usually generate a large number of simulation runs (database may be on the order of gigabytes and higher). The scope of this paper is to present a broad overview of methods and algorithms that can be used to analyze and extract information from large data sets containing time dependent data. By "extracting information" we mean the following: construct input-output correlations, find communalities and identify outliers.

Some of the algorithms presented here have been developed or are under development within the RAVEN [4] statistical framework.

DATA SET FORMAT

We will indicate with Ξ the original data set which contain N time series H_n: $\Xi = \{H_1, \dots, H_n, \dots, H_N\}$. To preserve generality of this paper, we can assume that each history H_n contains three components:

$$H_n = \{\boldsymbol{\theta}_n, \boldsymbol{\Delta}_n, \boldsymbol{\Gamma}_n\} \quad (1)$$

These components are the following:

- Continuous data $\boldsymbol{\theta}_n$: this data contains the temporal evolution of each scenario, i.e., the time evolution of the M state variables x_m $(m = 1, \dots, M)$ (e.g., pressure and temperature at a specific computational node). All these state variables change in time t (where t ranges from 0 to t_n[1]):

$$\boldsymbol{\theta}_n = \{x_1^n, \dots, x_M^n\} \quad (2)$$

  where each x_m is a an array of values having length T_n. Hence $\boldsymbol{\theta}_n$ can be viewed as a $M \times T_n$ matrix.
- Discrete data $\boldsymbol{\Delta}_n$: which contains timing of events. Note that a generic event E_i^n can occur:
  - At a time instant t_i: in this case the event can be defined as (E_i^n, t_i), or,
  - Over a time interval $[t_i^\alpha, t_i^\omega]$: in this case the event can be defined as $(E_i^n, [t_i^\alpha, t_i^\omega])$
- Set $\boldsymbol{\Gamma}_n$ of boundary conditions BC_t^n $(t = 1, \dots, T)$ and initial conditions $IC_s^n (s = 1, \dots, S)$.

DATA PRE-PROCESSING

This section focuses on the continuous part $\boldsymbol{\theta}_n$ of the data set Ξ. Depending on the applications, the data set may need to be pre-processed. The most common pre-processing is the Z-normalization procedure: each element x_m of $\boldsymbol{\theta}_n$ is transformed x'_m:

$$x'_m = \frac{x_m - mean(x_m)}{stdDev(x_m)} \quad (3)$$

where $\boldsymbol{mean(x_m)}$ and $\boldsymbol{stdDev(x_m)}$ represent the mean and the standard deviation of $\boldsymbol{x_m}$.

This transformation provides an equal importance to every $\boldsymbol{x_m}$ and it compensates for amplitude offset and scaling effects when distance between time series is computed[2].

In case the time-series are affected by noise, it might be worth to smooth the time series so that the noise is filtered out and the series information is maintained.

DATA REPRESENTATION

One of the most fundamental modeling choices regarding time dependent data is how each time series is numerically represented. Reference [] provides a broad analysis of the many representation methods which are here summarized:

- *Real-valued*: the original format of the time series is maintained
- *Polynomial*: the time series is approximated by a polynomial function (e.g., Chebyshev) up to a fixed degree and the vector of coefficients are retained as representatives for the time series

[1] This allows us to maintain generality by having time series with different time lengths

[2] This is in particular relevant when x_m have different scales (e.g., temperatures in the [500,2200] F interval while pressures are in the [0,16 10^6] Pa)

- *Discrete Fourier*: similar to the polynomial representation, the time series is approximated by a Fourier series and the series coefficients are retained as representatives for the time series
- *Singular Value Decomposition* (SVD): this method performs an Eigen-value and Eigen-vector decomposition of $\boldsymbol{\theta}_n$ and selects a reduced set of Eigen-vectors. Each time series H_n is represented by the coefficients associated to each Eigen-vector
- *Symbolic*: this method performs a symbolic conversion of the continuous data $\boldsymbol{\theta}_n$. This is accomplished by quantizing the time and state variables x_m and by associating to each quantized element a symbol (see Fig.1)

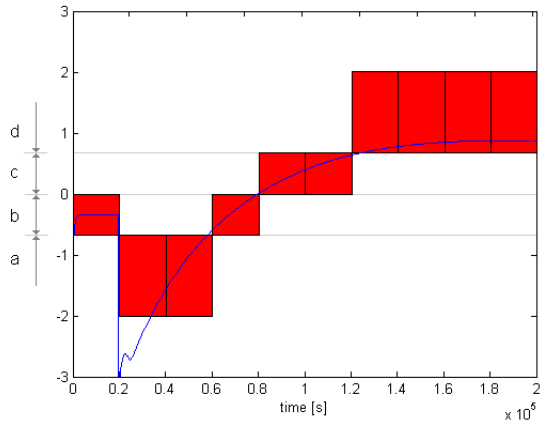

Fig. 1. Example of symbolic representation of a time series (blue line) into a sequence of symbols through a discretization process in both time and amplitude (red blocks) []. The resulting sequence of symbols is: *baabccdddd*.

MEASURING SIMILARITY

The second important modeling choice when dealing with time series regards the type of similarity metric also knows as distance. Similarly to the theory behind distances in Euclidean space, a distance metric $d(S,Q)$ measures the "similarity" between two generic objects S and Q. The only requirement behind $d(S,T)$ is that it has to obey the following rules:

$$\begin{cases} d(S,S) = 0 \\ d(S,Q) = d(Q,S) \\ d(S,Q) = 0 \text{ iff } S = Q \\ d(S,Q) \leq d(S,K) + d(K,Q) \end{cases} \quad (4)$$

When dealing with time series, the following two metrics are the most commonly used:

- Euclidean distance
- Dynamic Time Warping (DTW) distance

Both these distances are described in the next two subsections for the univariate case, i.e., two time series Q and S where their continuous part has $M = 1$. The more generic case, i.e., multivariate case, can be easily expanded from what is shown below.

Euclidean distance

Given two univariate time series S and Q having identical length (i.e., $T_S = T_Q$) the Euclidean distance $d_2(S,Q)$ is defined as:

$$d_2(S,Q) = \sqrt{\sum_{t=0}^{T_S} \left(x_1^S(t) - x_1^T(t)\right)^2} \quad (5)$$

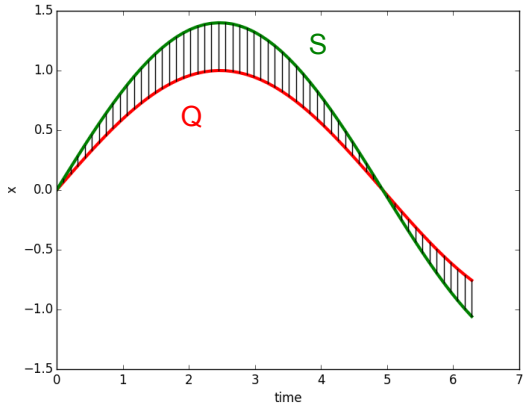

Fig. 2. Euclidean distance metric for two time series S and Q. Each black segment represents: $x_1^S(t) - x_1^T(t)$.

DTW Distance

This distance can be viewed as a natural extension of the Euclidean distance applied to time series. Given two univariate time series S and Q having length T_S and T_Q respectively[3]. The distance value $d_{DTW}(S,Q)$ is calculated by following these two steps:

1. Create a matrix $D = [d_{i,j}]$ having dimensionality $T_S \times T_Q$ where each element of D is calculated as follows: $d_{i,j} = (x_1^S[i] - x_1^Q[j])^2$ for $i = 1, \dots, T_S$ and $j = 1, \dots, T_Q$.
2. Search a continuous path $w_k|_1^K$ in the matrix D that, starting from $(i,j) = (0,0)$, it ends at $(i,j) = (T_S, T_Q)$ and it minimizes the sum of all the K elements $w_k = \left(d_{i,j}\right)_k$ of this path:

$$d_{DTW}(S,Q) = \min\left(\sum_{k=1}^{K} w_k\right) \quad (6)$$

This metric has the advantage of capturing similarities between time series that are shifted in time.

[3] Note that here he we have relaxed the requirement: $T_S = T_Q$

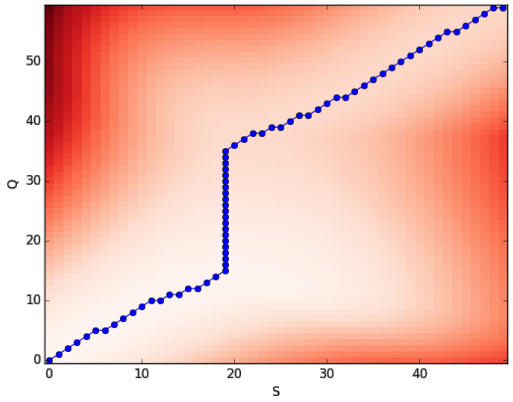

Fig. 3. Colored plot of the distance matrix D for two time series S and Q plotted in Fig. 4. Blue line represents the warp path $w_k|_1^K$.

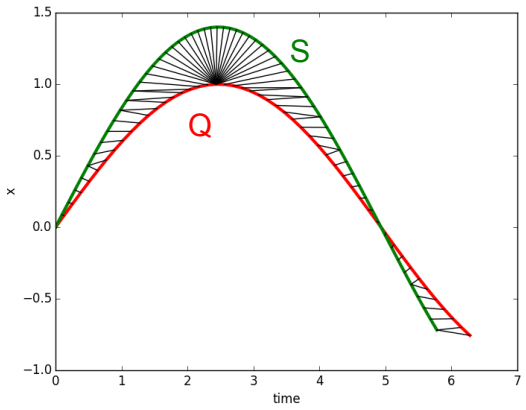

Fig. 4. DTW distance metric for two time series S and Q. Each black segment represents an elements $w_k = (d_{i,j})_k$ of the warp path shown in Fig. 3.

DATA MINING TECHNIQUES

For the scope of this article we focused on two applications: data searching and clustering. While we believe clustering offers the best tools to "extract information" from data (see first section of this paper), time series searching algorithms allow the user to match time series coming from different data sets.

Data Searching

Data searching algorithms are an important class of data analysis tools that can be very useful to compare and analyze similarities between two time series data sets (e.g., for code validation). In our experience, the two most reliable methods are the following: K-Nearest Neighbors (KNN) [] and Kd-Tree [].

Clustering

From a mathematical viewpoint, the concept of clustering [] is that we aim is to find a partition $\boldsymbol{C} = \{C_1, \dots, C_l, \dots, C_L\}$ of the set of N scenarios $\Xi = \{H_1, \dots, H_n, \dots, H_N\}$ where each scenario H_n is represented as shown in (1). Each C_l $(l = 1, \dots, L)$ is called a cluster. The partition $\boldsymbol{C}$ of $\boldsymbol{X}$ is the following one:

$$\begin{cases} C_l \neq \emptyset \\ \bigcup_{l=1}^{L} C_l = \Xi \end{cases} \tag{7}$$

Even though the number of clustering algorithms available in the literature is large, usually the most commonly used ones when applied to time series are the following: Hierarchical [], K-Means [] and Mean-shift [].

Hierarchical algorithms build a hierarchical tree from the individual points (leaves) by progressively merging them into clusters until all points are inside a single cluster (root). Clustering algorithms such as K-Means and Mean-Shift, on the other hand, seek a single partition of the data sets instead of a nested sequence of partitions obtained by hierarchical methodologies.

Approach 1

The first approach we followed is to perform clustering time series using classical clustering algorithms (e.g., K-Means, Mean-Shift and hierarchical) not directly on the time series but on the pre-processed data. This can be accomplished when one of the above-mentioned representations is chosen: polynomial, Fourier, SVD. Each time series is represented as a multi-dimensional vector where each dimension of the vector represents the coefficient of a specific base: polynomial, sin/cos, and Eigen-vector decomposition respectively.

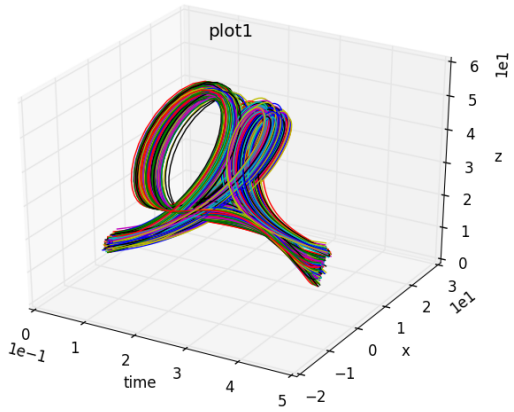

Fig. 5. Plot of a 1000 time series data set in a 2-dimensional space (plus time).

Approach 2

The second approach we followed is to reconstruct the major clustering algorithms available in the literature (K-Means, Mean-Shift and Hierarchical) so that they can natively perform data analysis on the time series data set.

The major challenge in this approach is the need to define an operator that given a sub set of time series it generates a distance-based average time series. This average value can be challenging to obtain especially if DTW distance is used.

An example of a time series application is shown in Fig.5: a data set that contained the time evolution of 1000 series has been generated by randomly changing (through a Monte-Carlo sampling) the initial condition.

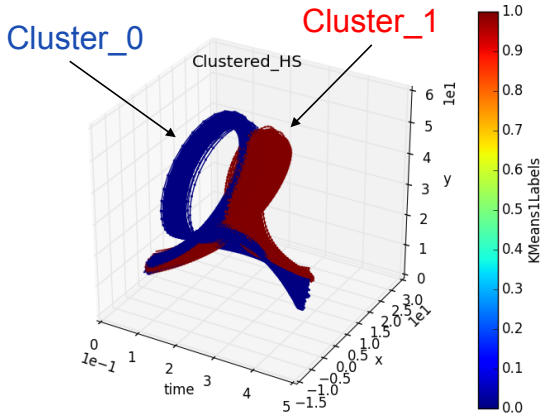

Fig. 6. Plot of the two clusters obtained from the data set shown in Fig.5.

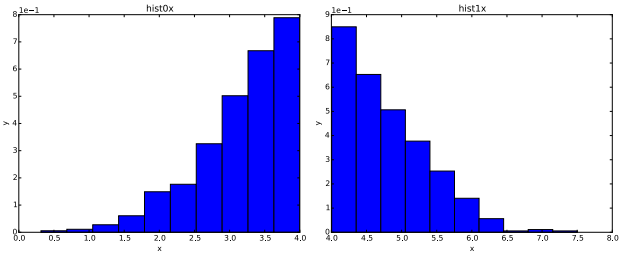

Fig. 7. Histograms of the sampled values for Cluster_0 and Cluster_1 (shown in Fig. 6) that created them and were captured by the clustering algorithm.

CONCLUSIONS

In this paper we have presented an overview of methods that can be employed to analyze time dependent data. We cover all main aspects of a typical analyze ranging from data pre-processing, metric choice, data searching and clustering. These algorithms have been developed or are under current development within the RAVEN statistical framework.

REFERENCES

1. RELAP5-3D CODE DEVELOPMENT TEAM, RELAP5-3D Code Manual. 2005.
2. R. O. GAUNTT, MELCOR Computer Code Manual, Version 1.8.5, Vol. 2, Rev. 2. Sandia National Laboratories, NUREG/CR-6119.
3. MAAP5 - Modular Accident Analysis Program for LWR Power Plants. EPRI, Palo Alto, CA: 2013.
4. A. ALFONSI, C. RABITI, D. MANDELLI, J. COGLIATI, R. KINOSHITA, AND A. NAVIGLIO, "RAVEN and Dynamic Probabilistic Risk Assessment: Software Overview," in *Proceedings of European Safety and Reliability Conference* (*ESREL*), Wroklaw (Poland), 2014.
5. B. RUTT, U. CATALYUREK, A. HAKOBYAN, K. METZROTH, T. ALDEMIR, R. DENNING, S. DUNAGAN, AND D. KUNSMAN, "Distributed dynamic event tree generation for reliability and risk assessment," in *Challenges of Large Applications in Distributed Environments*, pp. 61-70, IEEE, 2006.
6. E. HOFER, M. KLOOS, B. KRZYKACZ-HAUSMANN, J. PESCHKE, AND M. WOLTERECK, "An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties," *Reliability Engineering and System Safety* **77**, pp. 229-238, 2002.
7. K. S. Hsueh and A. Mosleh, "The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants," *Reliability Engineering and System Safety* **52**, pp. 297-314, 1996.