Approximation of Time Series Data using MCs: A Study of State Features and Transition Probabilities

Stephen Wormald; stephen.wormald@ufl.edu

***Abstract*—Autonomous systems frequently use probabilistic models, reinforcement learning, and machine learning techniques built upon Markov Chains (MC) and Markov Decision Processes (MDP) to model problem spaces, perform forecasting, and learn policies for task completion. However, explicit structures of Markov models can be challenging to define, and models hand-crafted by experts become challenging to maintain. Related approaches use high-dimensional and continuous state spaces and neural networks to define MDPs for reinforcement learning applications. However, the resulting models can require increased data size and training time. Furthermore, high-dimensional models are often inexplicable due to complex mathematical abstraction and are challenging to visualize. This paper proposes a method to construct MCs by partitioning a state-space using Gaussian mixture models to generate symbolic representations of time-series datasets. The algorithm is demonstrated on an Electrocardiogram (ECG) dataset, and is extended to the problem of forecasting the next symbolic state using a MC representing learned states and state transition times.**[1]

I. Background

In modern Machine Learning (ML), Neural Networks (NN), Deep NNs (DNN), Generative Adversarial NNs (GAN), Convolutional NNs (CNN), and transformers have proven useful across various application spaces. While capable, these models can be challenging to explain and approve for safety [1]. The ML formulation of Reinforcement Learning (RL) is built upon the more basic idea of maximizing reward in a Markov Decision Processes (MDP) via use of the bellman equations. Though RL often uses neural networks and loses explainability, simpler Markov Models are often more explainable due to the reduced dimensionality of the model when the state definition and transitions are well understood.

However, systems built upon Markov models are constrained by the model structure and can lack adaptivity when encountering anomalous states or out-of-distribution observations. For example, expert systems depending on graphical models like MDPs encountered challenges with brittleness because describing the full set of states and transition rules depend on domain-knowledge which can be inconsistent, incomplete, and challenging to maintain [2]. Furthermore, it can be non-trivial to know how observations map to expert-described states, or whether the model contains a state for new observations. To address this problem, methods are emerging to detect and account for out-of-distribution observations [3]. This paper demonstrates an algorithm for constructing reduced-order representations of time-series data that may be used to forecast the next abstract state in the learned MC.

[1]This paper was written as a final project report and is not peer reviewed.

Various methods exist to derive MCs from time series data (Fig. 1 shows notional data). These methods are based on wavelet-based partitioning [6], information theoretic approaches [5], and phase-space partitioning [8] to compress time-series data into a symbolic representation with reduced-order. Generally, after partitioning a state space, a symbol is assigned to each state region. A trajectory through the state space can be converted into a symbolic representation by adding one of these symbols to a symbol sequence whenever the trajectory transitions to a new partition. This paper mimics this methodology using Gaussian Mixture Models (GMM) to find K clusters of trajectory types, effectively partitioning a larger state space into K regions, before generating a representative MC that was used to forecast the next state. Section II describes the problem statement before the algorithm and project approach are discussed in Section III. Section III describes results from MC construction and application to forecasting the next state.

II. Problem statement

The problem statement is to develop an algorithm for constructing MCs that act as reduced-order and symbolic representations of time-series data, and then to use the generated MC to forecast the next symbolic state. This problem was self-selected as a project topic for a Safe Autonomous Systems class taught at the University of Florida. Note the proposed problem statement changed from implementing several published algorithms to developing an algorithm due to time constraints (20hr/week). After developing the algorithm (MC_{luster}), MC_{luster} was used to generate MCs with probabilistic state transitions. The generated MC was then used to forecast the next N states to determine how long the next state could be predicted accurately without observing the true state. MC_{luster} was tested on single-dimensional time series data from an electrocardiogram (ECG) dataset, though in principle the approach extends to multidimensional or multi-channel time series data.

III. Project Approach

The problem approach comprised of four key tasks. These tasks consist of understanding existing algorithms, creating and verifying the MC_{luster} algorithm, using MC_{luster} to generate MC chains for the ECG dataset, and using generated MCs to perform forecasting for the next sybmol states.

Task 1 – Identify and Understand Algorithms that Generate MCs: Several key algorithms acted as inspiration

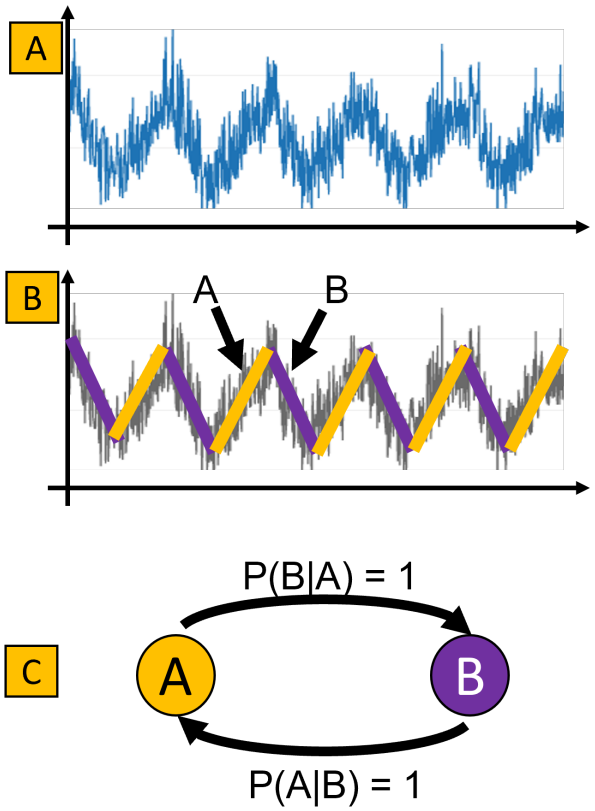

Fig. 1. Depiction showing the reduced order representation of time-series data using a MC. (a) Representative time series data. (B) Time series data with two predominant features, increasing and decreasing data trends, represented by a two-symbol alphabet of the set a,b. (C) Reduced order MC representing the state transitions and the transition probabilities.

for the MC_{luster} algorithm, including: (1) Symbolic False Nearest Neighbors (SFNN), which is a phase-space partitioning algorithm for continuous data [4]; (2) Wavelet-based partitioning, which is built on the wavelet transform [6]; (3) The ϵ-machine, which was designed to approximate the motion of non-linear dynamical systems [11]; (4) the D-Markov machine which focuses on the characterization of anomaly detection in chaotic systems [5]. The D-Markov machine was reviewed in more detail and acted as the main inspiration when developing MC_{luster}. The D-Markov machine algorithm seeks to partition a state space into cells that produce causal transitions between cells, where each cell is assigned a symbol in an symbol alphabet **A**. Mathematically, given a state sequence **S**, composed of historical states ($\overleftarrow{S}$) and future states ($\overrightarrow{S}$), an agglomerate or abstract state **R** is considered prescient (or effectively causal) if:

$$H[\overrightarrow{S}^L|\mathbf{R}] = H[\overrightarrow{S}^L|\overleftarrow{S}] \forall L \in \mathbb{N} \quad (1)$$

Where L determines the length of state sequences being considered. Intuitively, this formulation means that an **R** is a good effective state (representing a state space partition) when the prediction of of the future state sequence ($\overrightarrow{S}^L$) made using **R** is the same as predicting ($\overrightarrow{S}^L$) when knowing the full state history ($\overleftarrow{S}$).

Practically, this algorithm is implemented as a sub-tree merging problem for many generated state histories $\overleftarrow{S}$. The system dynamics are observed and used to generate a tree data structure before performing the sub-tree merging which essentially groups like states so that the agglomerate states are maximally causal.

Task 2 – Develop and test the MC_{luster} algorithm for MCs construction: The MC_{luster} algorithm was designed to use Gaussian mixture models in order to identify N number of states that best represent different kinds of patterns in time series data. The feature space was generated by taking a single times series data stream and generating sequences of length L, which generates a number or set of observations in an L dimensional space. Clusters in this space represent signal patterns that stand out in the time series data set. Similar to the papers read and described in Task 1, each cluster corresponds to a state space partition, and is assigned a unique symbol When in order to generate a MC representing the time series data set. Each point in the time series is assigned a symbol based upon the cluster it is assigned to. The resulting sequence may be compressed to only include the current state and number of repetitions of that state before transitioning to a different state. This method was tested for a simple time series data set representing a triangle wave with noise. Results from this testing are described in Section IV. Note the number of clusters and the segment length L are both hyperparameters that need to be optimized. Figure 2 shows a graphical illustration of the approach.

Task 3 – Use the MC_{luster} algorithm to build MCs representing the ECG dataset: The algorithm was used to construct a MC representing the ECG data set from reference. For this work, the number of clusters was assumed to be K=3, though this is a potential area of improvement. After building a symbol sequence that represents the transitions between different states, two different types of MCs were constructed. The first MC represents only state transitions by counting the number of times one state transitions into the next states. This kind of MC represents a reduced-order transition between states, but does not lend itself to forecasting in time. A second MC type was created with states representing the time spent inside of each class (see Fig. 3). In this transient MC formulation, each class corresponds to T states, where T is the maximum number of time increments a class ID remained unchanged. The probability of transitioning from one state to the next state was calculated by dividing the number of state transitions between two states (between S_0 and S_1, or S_{01}) by the total number of state transitions out of a the prior state ($S_{0,total}$). Section IV describes results from this approach.

Task 4 – Perform forecasting of the next state in a MC to evaluate the MC predictive capability: The transient MC generated in Task 3 was used to perform forecasting of the

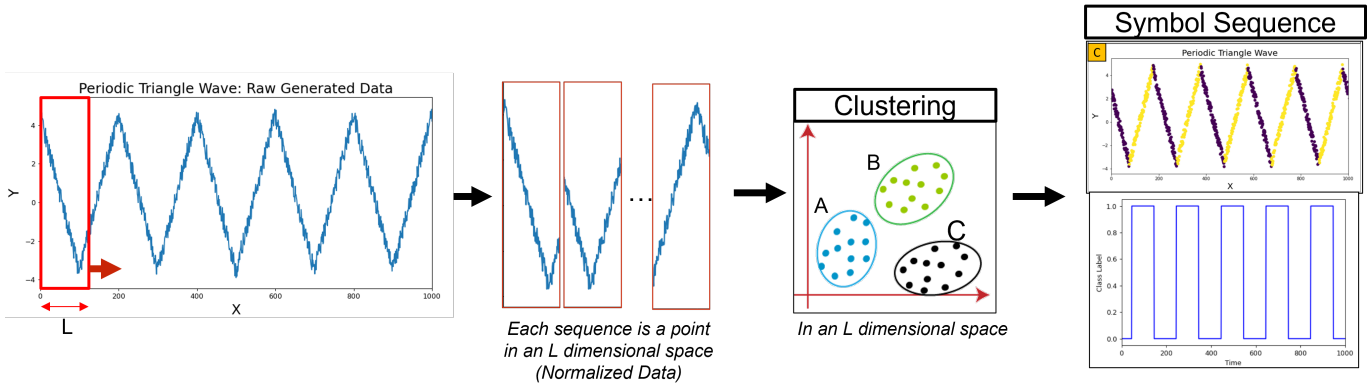

Fig. 2. Approach for generating data from time series data set by partitioning it into sections of length L, where each data section is represents a single point in an L dimensional space where clustering may be used to identify patterns. This approach was tested for a triangle wave with noise showing that the upwards and downward slopes are classified as two distinct states. Gaussian mixture models were used to identify two states in this test. The rightmost figure shows points classified according to class type, where the found state is plotted with respect to time in the bottom figure on the rightmost side.

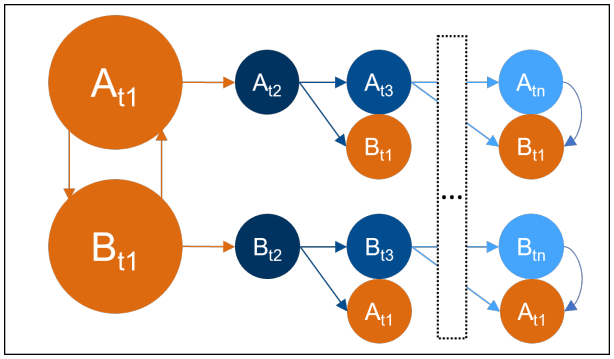

Fig. 3. Representative time dependent MC with two states (A and B), assuming that the end state will eventually transition to an alternate state. For this work, a length of n = 800 was selected as the longest time without state a transition was 800 time steps. The 2-state may be generalized to any number of states.

next ECG state in a probabilistic manner. Using the transient MC, the transition probability to other states was generated as a function of time elapsed inside of a state. This process creates, transition probability curves which were generated for each state. These transition probability curves were used to perform forecasting.

Given a current state, (defined by the class ID and the number of elapsed time steps) the transition probabilities to the next states were identified from the transition probability curves corresponding to that state. A random number was generated and was used to determine the next state with a frequency according to the state transition probabilities leading out of the current state. This approach was repeated one hundred times for M time steps into the future to predict a single symbolic sequence. The predicted symbol at each time step was predicted as the class with the highest probability of occurring. Note M is defined as a variable so the total prediction accuracy can be recorded as a function of the distance of time predicted into the future. The results from state forecasting are described in Section IV.

IV. Results and Discussion

This section details the project results. Result correspond to the four tasks described Section III.

Results from testing the algorithm:

In the initial software formulation for this project, the D-Markov algorithm was not implemented exactly due to time constraints. Specifically, the sub-tree merging algorithm was replaced with a clustering approach which mimics the idea of merging similar states with similar transitions.

The simplified algorithm is explained with reference to Figure 4, where a periodic triangle wave with noise was generated for testing and illustration purposes. The full time series dataset is first used to generate a set of state sequences of length L. Essentially, a sliding window of length L is shifted across the initial time series dataset, where the state sequence associated with each shift is added to the set state histories ($\overleftarrow{S}$). If the initial time series sequence is 10000 samples long, and L = 100, then the number of state histories in $\overleftarrow{S}$ would be 9900. The shaded regions in Figure 4.a represent two state histories. In other words, this process generates 9900 points in an L-dimensional space, where clustering algorithms may be used to find groups of similar state histories. In this work, the Gaussian Mixture Model (GMM) clustering algorithm was used as this algorithm considers the covariance of the data. The K-Means clustering algorithm was tested, but gave worse results. Figure 4.b shows results from clustering when L = 50 and the number of Gaussian models (K) was 2, where each subplot shows the set of state sequences associated with either Class 0 or Class 1 (plotted with high transparency to show general trends). Note that the set of state histories in Class 0 generally represents the set of downward sloping lines, whereas Class 1 generally represents the set of upward sloping lines. The fit GMM may be used to generate class IDs for state histories in the initial dataset. Figure 4.c colors points from Figure 4.a with the class ID. Note how the upward and downward lines are

cleanly distinguished from each other, where the color or class ID represents the symbol from the symbol alphabet **B** ID of a given state history. When expressed as a MC, this state space partitioning produces a 2 state MC with transition probabilities of 1 between each state.

Results from MC generation for ECG data:

The approach used for the noisy triangle wave was repeated for the single-dimensional electrocardiogram data (ECG) where K=3, as seen in Figure 5. Note each state history was normalized prior to using the GMM clustering algorithm because the magnitude of response changes significantly through the time series. The clusters identified are better understood through Figure 5.b. In this subplot, the state histories associated with Class 2 capture the heartbeat pattern, allowing these sections of the time series to be classified as seen in Figure 5.c. Note Class 1 and 0 from Figure 5.b capture the stationary regions between heartbeats, as represented by the largely random distribution of state histories in Figure 5.b.

Note there is a relationship between L, the optimal number of GMM clusters K, the periodicity of trends in time series data, and the quality of MC construction. Furthermore, a single value of L is used in the MC construction, though there is likely benefit in using a variety of L per cluster. These variables and relationships are not studied in the scope of this project.

Time series forecasting can be approximated in the symbol sequence space after constructing a MC from time series data. As mentioned previously, the transition between each Class ID represents the transition between individual states, which can be used to generate a representative MC. Specifically, the number of transitions from one state to every other state can be counted and normalized by the total number of transitions to represent the individual state transition probabilities. Figure 6 shows a MC constructed in this manner.

However, this MC does not represent the time spent inside of a single state. The time inside of a single state is important when determining time series forecast. The transient MC was also created as described in the project approach, and the transition probabilities for each class ID to each other class ID is depicted in Fig. 7. Note how for each state, the probability of transitioning to the same class ID (not changing state) is high for the initial 200-400 states. After this time, the probability of an alternate class ID becomes higher. Note that due to the stochastic nature of the transition probabilities, the transition probability after approximately 350 states was forced to transition to the most likely state that is not the current state. This problem could be avoided if more samples were collected, or if better state transition probabilities were estimated.

Results from time series forecasting:

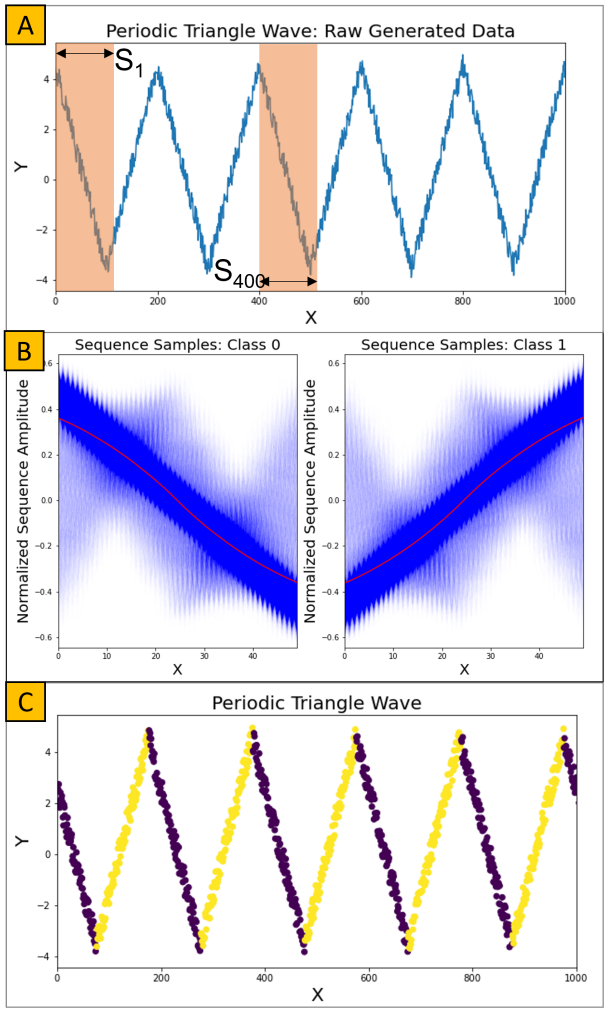

Fig. 4. Process of finding symbolic representations of state space trajectories applied to synthetic triangle wave with noise. (A) The initial time series dataset is used to generate a set of state histories of length L. These state histories act as points in an L-dimensional where GMM clustering is performed. (2) Clustered state histories corresponding to Class 0 and Class 1, where each class represents a symbolic state. (3) Points in the original time series dataset are colored according to their class value, where purple represents Class 0, and yellow represents Class 1.

Time series forecasting was completed using the transient state transition probabilities in Fig. 7. Starting with some initial state, the next state could be computed probabilistically by generating a random number, and then generating the next state with a frequency in proportion to the probability of each next state. The next predicted state would then be set as the current state. This process would be repeated M times into the future. After M steps, the predicted state was set to equal the real state, which simulates the act of observing the real state in an environment. In other words, changing M and calculating the prediction accuracy shows how far into the future predictions can be made well before the prediction accuracy becomes poor.

When generating symbol trajectories, it is important that the state sequence statistics match the original sequence

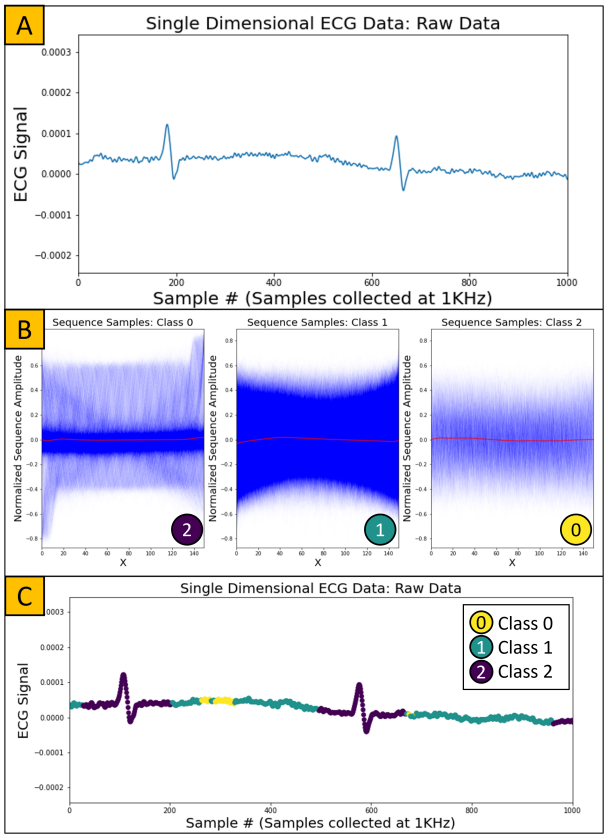

Fig. 5. Process of finding symbolic representations of state space trajectories applied to synthetic fetal ECG data. (A) The initial time series dataset is used to generate a set of state histories of length L. These state histories act as points in an L-dimensional where GMM clustering is performed. (2) Clustered state histories corresponding to Class 0, Class 1, and Class 2, where each class represents a symbolic state. (3) Points in the original time series dataset are colored according to their class value, showing how the heartbeat can be segmented as Class 0.

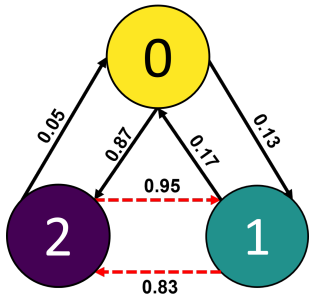

Fig. 6. Markov chain with state transition probabilities listed on edges for the time series GMM compression represented in Figure 5. Red dashed arrows represent the state transitions which maximize transition probability.

distribution. Fig. 10 shows a distribution of how long a trajectory stayed in a given class ID before transitioning to a different class ID (for both the actual and simulated data). The distributions are similar, except for where the class type duration is greater than 175 for the simulated data. This feature arises because state transition was forced after approximately 200 time steps come on whereas in the actual data some state transitions occurred after this duration. Fig. 8 superimposes the real estate trajectory, the distribution of simulated state trajectories, and the average state prediction. In this figure, predictions are made up to 10 time steps into the future (M=10). Note how the average state prediction in yellow typically matches the real prediction, even though the simulated trajectories in blue do not always match.

The accuracy of prediction may be calculated as the number true predictions divided by the total number of predictions. The prediction accuracy was calculated as a function of M, and is shown in Figure 10. Note that the prediction accuracy can achieve up to 80% when M $<$50. Note that these results are achieved using a standard transient MC that is not optimized to maximize the effective accuracy. Further work could improve these results by altering the state clustering to improve accurate forecasting.

V. Conclusion

Salient conclusions include:

- States captured through the GMM approach represent meaningful patterns of interest from time series data, such as heartbeats in ECG data. These results indicate that features captured by the GMM approach have the potential to be human explainable, which is valuable for explainable machine learning methods.
- While MCs can be constructed from time series data, these resulting MCs cannot always be used to perform accurate forecasting. Learned MCs can be used to generated distributions of feasible trajectories, though the results are often constrained to the symbol space rather than the original trajectory space. Transient MCs help perform more accurate forecasting in the symbol space.

Salient next steps include:

- *Vary window length*: Clusters could be generated with a variety of window lengths, effectively changing the filter length used when findings states in the time series data. A kind of hierarchical clustering could be incorporated to identify states with different window lengths L.
- *Extend to multidimensional data*: The current project focused on the use of single dimensional data. However, the method could easily be extended to multidimensional data by changing the clustering space. Data sets could include video data, wristwatch wearable data, or any dataset with multiple communication channels.
- *Analyze cluster features*: To aid explainability, the segments extracted from the time series data set could be analyze to determine feature uniqueness. For example, principal component analysis could be used to identify how different each state is from one another period. The clustering algorithm could be adapted in order to maximize the difference between state features.
- *Dynamic MC creation in the presence of new observations*: The clustering algorithm could be extended to optimize the number of clusters in response to new data. This approach would seek to capture situations

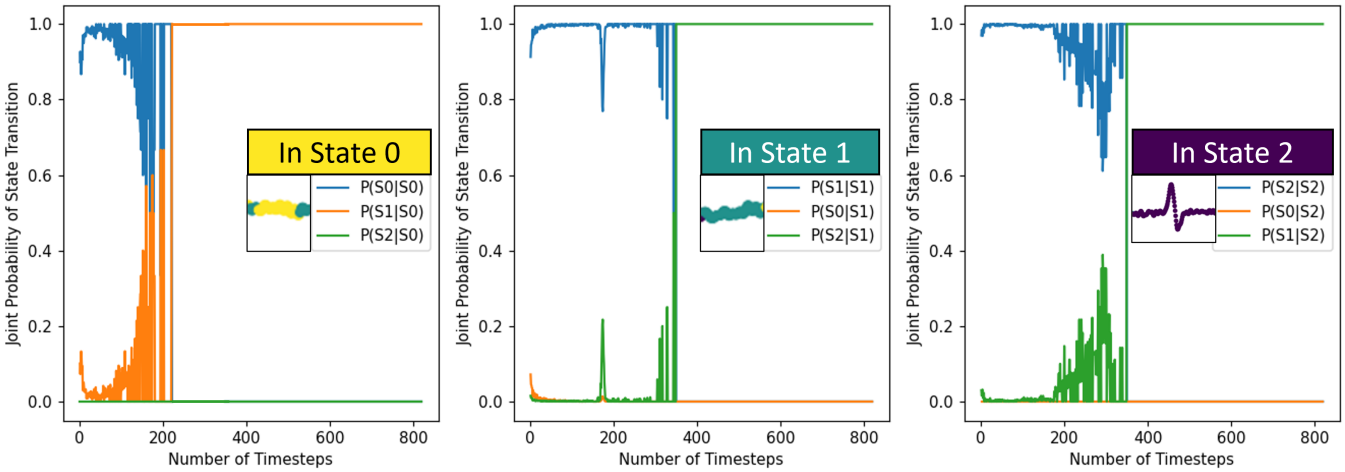

Fig. 7. State transition probabilities calculated for each state. The state transition was calculated by dividing the total number of transitions to the next state by the total number of transitions out of the current state. The total number of states was determined by a transient MC of length 800, as represented in Fig. 3. Note each figure corresponds to a single state, as described pictorially in the legend using clipped portions of Fig. 5.

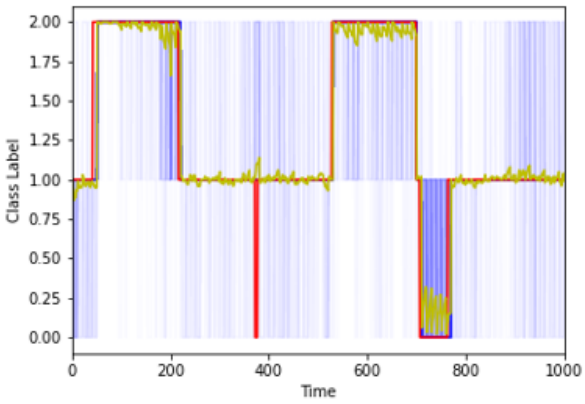

Fig. 8. Time series prediction of the state when the time between observations was 10. In other words, the next state was predicted blindly for 10 time steps before setting the predicted state to the true state, replicating an observation. The red line represents the true state, the blue lines represent simulated states representing many superimposed trajectories, and the yellow state represents the average state prediction.

where out of distribution observations are incorporated by connecting new states to an existing MC. Alternately, new states could be defined in response to different error types to aid in the prediction of the next state.

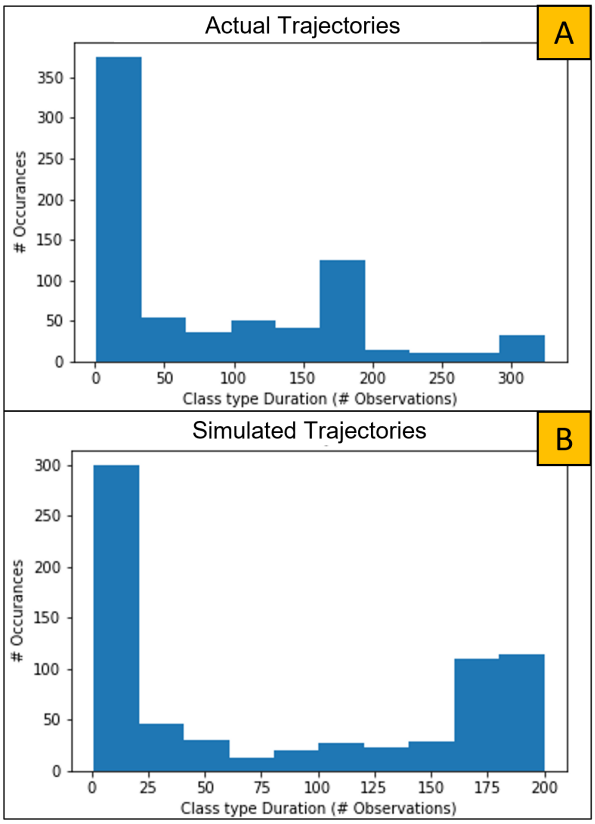

Fig. 9. Statistics showing how quickly the class changed through time for the ECG data (Fig. 5) modeled with a transient MC (Fig. 8). The class type duration represents the number of consecutive time steps that the class remained unchanged. The statistics from the truth data (a) and from the simulated data (b) are similar, though there is an increased number of sample lengths between length 175 and 200 for the simulated data as the simulated trajectories were forced to transition to the next state after 200 time steps.

References

[1] Rojat, T., Puget, R., Filliat, D., D'iaz-Rodr'iguez, N., Gelin, R., &; Ser, J. (2021). Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey. ArXiv.

[2] Gill, T. G. (1995). Early expert systems: Where are they now? MIS Quarterly, 19(1), 51. urlhttps://doi.org/10.2307/249711

[3] Yuan, L., Park, H. S., amp; Lejeune, E. (2022). Towards out of distribution generalization for problems in Mechanics. Computer Methods in Applied Mechanics and Engineering, 400, 115569. https://doi.org/10.1016/j.cma.2022.115569

[4] Kennel, M. B. (2003). Estimating good discrete partitions from observed data: Symbolic false nearest neighbors. AIP Conference Proceedings. urlhttps://doi.org/10.1063/1.1612266

[5] Ray, A. (2004). Symbolic dynamic analysis of complex systems for anomaly detection. Signal Processing, 84(7), 1115–1130. https://doi.org/10.1016/j.sigpro.2004.03.011

[6] Rajagopalan, V., &; Ray, A. (2006). Symbolic time series analysis via

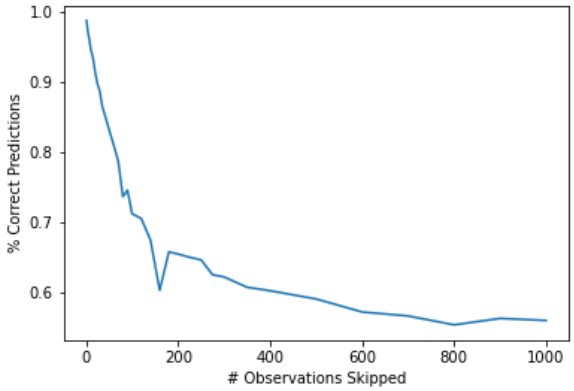

Fig. 10. Forecasting accuracy as the time between observations increases. The time between observations represents a system that operates for some time interval without detecting its current state. The predicted state is set to the true state after each observation. The dip in accuracy when the number of skipped observations is 180 likely relates to the increased likelihood of state transition when the time since previous state transition is near 180 - 200 time steps.

wavelet-based partitioning. Signal Processing, 86(11), 3309–3320. https://doi.org/10.1016/j.sigpro.2006.01.014

[7] Jha, D. K., Virani, N., Reimann, J., Srivastav, A., &; Ray, A. (2018). Symbolic analysis-based reduced order Markov modeling of Time Series Data. Signal Processing, 149, 68–81. https://doi.org/10.1016/j.sigpro.2018.03.004

[8] Mukherjee, K., amp; Ray, A. (2014). State splitting and merging in probabilistic finite state automata for signal representation and analysis. Signal Processing, 104, 105–119. https://doi.org/10.1016/j.sigpro.2014.03.045

[9] Lin, J., Keogh, E., Wei, L., amp; Lonardi, S. (2007). Experiencing sax: A novel symbolic representation of Time Series. Data Mining and Knowledge Discovery, 15(2), 107–144. https://doi.org/10.1007/s10618-007-0064-z

[10] Murphy, K. P. (2021). Machine learning: A probabilistic perspective. MIT Press.

[11] Crutchfield, J. P., &; Young, K. (1989). Inferring statistical complexity. Physical Review Letters, 63(2), 105–108. https://doi.org/10.1103/physrevlett.63.105

[12] Tuyen, L. (2018). A Higher order Markov model for time series forecasting. International Journal of Applied Mathematics and Statistics.

[13] Wilinski, A. (2019). Time Series Modeling and forecasting based on a Markov chain with changing transition matrices. Expert Systems with Applications, 133, 163–172. https://doi.org/10.1016/j.eswa.2019.04.067

[14] Amin, M. R., Wickramasuriya, D., &; Faghih, R. T. (2022, May 26). A wearable exam stress dataset for predicting cognitive performance in real-world settings. A Wearable Exam Stress Dataset for Predicting Cognitive Performance in Real-World Settings v1.0.0. Retrieved September 20, 2022, from https://www.physionet.org/content/wearable-exam-stress/1.0.0/