patternly: Anomaly Discovery in Dynamical Systems

Drew Vlasnik and Ishanu Chattopadhyay*
University of Chicago, Chicago, IL

I. Motivation: What is an anomaly?

Anomalies are outliers, exceptions, or aberrations in data that do not conform to some well defined notion of normal behavior [2]. Exhaustive enumeration of normal behaviors, to then isolate deviations as anomalous patterns, is generally ineffective on account of the following:

1) Enumerating all normal behaviors is typically infeasible. In addition, normality is often application dependent, requiring extensive subject matter expertise for characterization. Enumerating anomalous behaviors is even more problematic.
2) Normal behaviors often evolve, and a current notion of normal might not be sufficiently representative in future.
3) Labeled data to train anomaly classifiers is often absent.
4) Measurement noise might confound anomalies, or make normal behavior seem anomalous.
5) In adversarial scenarios, anomalies might be intentionally crafted to manifest normal characteristics.

Anomaly detection algorithms have investigated diverse strategies to address these concerns to varying degrees of success [11], [9], [2].

A. Anomaly Discovery in Dynamical Systems

In this study, we investigate *anomaly discovery* in dynamical systems, where templates of normal or anomalous dynamical behaviors are not known a priori. Thus, 1) we necessarily do not know a priori how many of acceptable or normal behaviors exist or what they look like, or if they evolve with time, and 2) we also do not have any prior knowledge of what constitutes an anomaly in the system of interest. In such a scenario, conceptualizing a well-founded notion of an anomaly is itself non-trivial.

Here, we consider sequential data streams as our observations, which are generated by a dynamical system whose rules of evolution or operation are not directly observable. Consistent with its intuitive definition, we conceptualize an anomaly as a dynamical pattern emerging within a observation sequence that is *uncommon, unexpected, and/or is poorly explained as having simply arisen by chance*. Thus, in our framework, anomalies arise due to a non-random deviation in the underlying dynamical system structure or its parameters. Allowing for the possibility that the system of interest might switch between different dynamical regimes, we intend to discover these patterns of switching, and then characterize when an anomalous switch transpires. In such a setting, the problem is complicated by the need to recognize distinct regimes, back out the models of dynamical operation in these regimes, and detect when an unexpected switch occurs; one that is not explainable as a likely noise artifact.

B. Relation to Anomaly Discovery in Tabular Data

In tabular data $i.e.$ where we have samples specified by feature-values, the isolation forest [10] is an effective approach to unsupervised anomaly detection, which directly characterizes anomalies without first profiling normal behavior. The isolation forest identifies anomalies via binary trees, exploiting the fact that in a recursive partitioning of the feature space, anomalies on account of their assumed rarity, will typically be isolated using smaller number of partitions compared to normal samples. The algorithm is time and memory efficient, and is thus applicable in high dimensional problems that have a large number of irrelevant attributes, and in situations where training set does not contain any anomalies.

However, the isolation forest algorithm does not automatically transfer to a dynamical system setting, $e.g$, where our objects of interest are not samples specified by features, but are time series of possibly unequal observation lengths. The notion of an anomaly is this setting is also more intrinsic to system dynamics; while we also assume that anomalies are rare, the key defining feature here is a non-random deviation in the underlying dynamical rules, which are generally not directly observable, or known a priori.

C. Key Insight for Patternly

The key insight in the patternly approach is that dynamical behavior can be modeled efficiently with stochastic generators represented as a special class of probabilistic finite state automata (PFSA) [6], [4]. Representation of dynamics via a PFSA is applicable to both data streams that have a finite alphabet, or continuous-values, with the latter case requiring quantization of the values over a finite alphabet. Importantly, the special class of models that our PFSA framework refers to has efficient inference algorithms, and other important mathematical properies that suggests that a large class fo ergodic stationary finite-valued processes can be exactly represnted in this framework [3]. While for continuous-valued data astreams, quantization invariably loses information, discovering anomalies via deviation of models from adequately quantized data is still a viable approach: unexpected deviations in the quantized representaion is sufficient but not necessary to detect an emergent anomaly in the original data.

II. Background & Prior Work

To obtain a general solution to the anomaly discovery problem in dynamical systems, we need a modeling framework for observational data streams, that infers a model of the underlying dynamical system. We carry out this inference within the framework of probabilistic finite automata, which have been shown to be expressive enough to model a large class of ergodic stationary stochastic processes, evolving over a finite alphabet of values [3], [5], [8], [6], while admitting efficient inference algorithms [4].

A. Probabilistic Finite State Automata

Definition 1 (PFSA)**.** A probabilistic finite-state automaton G is a quadruple $(Q, \Sigma, \delta, \widetilde{\pi})$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \rightarrow Q$ called transition map, and $\widetilde{\pi} : Q \times \Sigma \rightarrow [0, 1]$ specifies observation probabilities, with $\forall q \in Q, \sum_{\sigma \in \Sigma} \widetilde{\pi}(q, \sigma) = 1$.

We use lower case Greeks (e.g. σ or τ) for symbols in Σ and lower case Latins (e.g. x or y) to denote sequence of symbols, with the empty sequence denoted by λ. The length of a sequence x is denoted by $|x|$. The set of sequences of length d is denoted by Σ^d.

*Corresponding Author E-mail: ishanu@uchicago.edu

The directed graph (with possible loops) with vertices in Q and edges specified by δ is called the graph of the PFSA and, unless stated otherwise, assumed to be strongly connected [1].

Definition 2 (Observation and Transition Matrices)**.** Given a PFSA $(Q, \Sigma, \delta, \widetilde{\pi})$, the observation matrix $\widetilde{\Pi}_G$ is the $|Q| \times |\Sigma|$ matrix with the (q, σ)-entry given by $\widetilde{\pi}(q, \sigma)$, and the transition matrix Π_G is the $|Q| \times |Q|$ matrix with the (q, q')-entry, written as $\pi(q, q')$, given by

$$\pi(q, q') = \sum_{\sigma : \delta(q,\sigma) = q'} \widetilde{\pi}(q, \sigma). \tag{1}$$

Both Π_G and $\widetilde{\Pi}_G$ are row-stochastic, *i.e.* non-negative with rows of sum 1. Since the graph of a PFSA is strongly connected, there is a unique probability vector $\mathbf{p}_G$ that satisfies $\mathbf{p}_G^T \Pi_G = \mathbf{p}_G^T$, and is called the stationary distribution of G [14].

Definition 3 (Γ-Expression)**.** δ and $\widetilde{\pi}$ may be encoded by a set of $|Q| \times |Q|$ matrices $\mathbf{\Gamma} = \{\Gamma_\sigma | \sigma \in \Sigma\}$, where

$$\Gamma_\sigma \big|_{q,q'} = \begin{cases} \widetilde{\pi}(q, \sigma) & \text{if } \delta(q, \sigma) = q', \\ 0 & \text{if otherwise.} \end{cases} \tag{2}$$

We extend the definition to $\Sigma^\star$ by $\Gamma_x = \prod_{i=1}^n \Gamma_{\sigma_i}$ for $x = \sigma_1 \dots \sigma_n$ with $\Gamma_\lambda = I$, where I is the identity matrix.

Definition 4 (Sequence-Induced Distributions)**.** For a PFSA $G = (Q, \Sigma, \delta, \widetilde{\pi})$, the distribution on Q induced by a sequence x is given by $\mathbf{p}_G^T(x) = \left[\!\left[\mathbf{p}_G^T \Gamma_x \right]\!\right]$, where $[\![\mathbf{v}]\!] = \mathbf{v} / \|\mathbf{v}\|_1$.

Definition 5 (Stochastic process generated by PFSA)**.** Let $G = (Q, \Sigma, \delta, \widetilde{\pi})$ be a PFSA, the Σ-valued stochastic process $\{X_t\}_{t \in \Sigma}$ generated by G satisfies that X_1 follows the distribution $\mathbf{p}_G^T \widetilde{\Pi}_G$ and X_{t+1} follows the distribution $\mathbf{p}_G \left(X_1 \cdots X_t\right)^T \widetilde{\Pi}_G$ for $t \in \mathbb{N}$.

We denote the probability an PFSA G producing a sequence x by $p_G(x)$. We can verify that $p_G(x) = \left\| \mathbf{p}_G^T \Gamma_x \right\|_1$.

B. Process KL Divergence Measures

Detailed proofs of the propositions in this section are available in cited work [3]. The notion of process KL divergence generalizes the analogous, well-studied notion for probability distributions [7].

Definition 6 (Entropy rate and Process KL divergence)**.** The entropy rate $\mathcal{H}(G)$ of a PFSA G is the entropy rate of the process generated by G [7]. Similarly, the KL divergence $\mathcal{D}_{\mathrm{KL}}\left(G \,\|\, G'\right)$ of a PFSA G' from the PFSA G is the KL divergence of the process generated by the G' from that of G [13]:

$$\mathcal{H}(G) = -\lim_{d \to \infty} \frac{1}{d} \sum_{x \in \Sigma^d} p_G(x) \log p_G(x) \tag{3}$$

$$\mathcal{D}_{\mathrm{KL}}\left(G \,\middle\|\, G'\right) = \lim_{d \to \infty} \frac{1}{d} \sum_{x \in \Sigma^d} p_G(x) \log \frac{p_G(x)}{p_{G'}(x)}, \tag{4}$$

whenever the limits exist.

Lemma 1. *For any PFSA G, H, KL divergence satisfies:*

$$\mathcal{D}_{KL}\left(G \,\|\, H\right) \geqq 0 \tag{5}$$

$$\mathcal{D}_{KL}\left(G \,\|\, H\right) = 0 \text{ iff } G = H \tag{6}$$

where we interpret equality of PFSA G, H as

$$\forall x \in \Sigma^\star, p_G(x) = p_H(x) \Rightarrow G = H \tag{7}$$

Proof: Follows from the standard argument for non-negativity of KL divergence for probability distributions [7]. ∎

Algorithm 1: PFSA Log-likelihood

Data: A PFSA $G = (Q, \Sigma, \delta, \widetilde{\pi})$ and a sequence x of length n.
Result: Log-likelihood of G generating x

1. Get the stationary distribution $\mathbf{p}_G$ as the left eigenvector of Π_G of eigenvalue 1;
2. Let $\mathbf{p}$ be the current distribution on states, and initialize it with $\mathbf{p}_G$;
3. Let L be the log-likelihood of G generating x and initialize it with 0;
4. **for** *each symbol σ in x* **do**
5. Get the current distribution on symbols $\phi = \mathbf{p}_G^T \widetilde{\Pi}_G$;
6. Update $L = L - \log \phi(\sigma)$;
7. Let $\mathbf{p}_{\text{new}}$ be the new distribution on states, and initialize all its entries with 0;
8. **for** *each state $q \in Q$* **do**
9. Let the next the state $q_{\text{new}} = \delta(q, \sigma)$;
10. Let $\mathbf{p}_{\text{new}}(q_{\text{new}}) = \mathbf{p}_{\text{new}}(q_{\text{new}}) + \mathbf{p}(q) \widetilde{\pi}(q, \sigma)$;
11. Update $\mathbf{p}$ with $\mathbf{p}_{\text{new}} / \|\mathbf{p}_{\text{new}}\|_1$;
12. Let $L = L/n$;
13. **return** L;

Definition 7 (Log-likelihood)**.** The log-likelihood [7] of a PFSA G generating $x \in \Sigma^d$ is given by

$$L(x, G) = -\frac{1}{d} \log p_G(x). \tag{8}$$

Theorem 1 (Convergence of Log-likelihood)**.** *Let G and H be two irreducible PFSA, and let $x \in \Sigma^d$ be a sequence generated by G. Then we have*

$$L(x, H) \to \mathcal{H}(G) + \mathcal{D}_{KL}\left(G \,\|\, H\right), \tag{9}$$

in probability as $d \to \infty$.

Next, we denote the log-likelihood of PFSA H generating a sequence x of length d which is actually generated by PFSA G as $L\left(x \overset{d}{\leftarrow} G, H\right)$. We show that the probability that sequences x, y generated by distinct processes cannot be distinguished by a random set of PFSA vanishes with enough data.

Theorem 2 (Approximate Metric)**.** *Let X and Y be two distinct PFSA in the sense of Eq.* (7)*, and x, y be of length at least d generated respectively by X, Y. If $\mathcal{G}$ is a randomly chosen set of k PFSA, then $Pr(\theta_{\mathcal{G}}(x, y) = 0) \to 0$, as $d, k \to \infty$.*

1) Implementation Issues & Complexity: The algorithm for evaluating the log-likelihood of a PFSA generating a given sequence is given in Alg. 1. It is immediate that the time complexity of log-likelihood evaluation is $O\left(d \times |Q|\right) + A$ with d is the input length and $|Q|$ is the number of states in the PFSA being considered, and A is the complexity of computing the stationary eigenvector in step 1. We note that the complexity for likelihood scoring of HMMs with the forward algorithm has time complexity $O\left(d \times |Q|^2\right)$, where Q is the number of the hidden states [12]. Nothwithstanding asymptotic time complexities, Alg. 1 is clearly significantly simpler to the dynamic programmimg involved in the forward algorithm of HMM likelihood scoring.

2) SLD with fixed base sets: As fixed base sets,, we use $\mathcal{G}$ composed the four simple PFSA shown. While better results may be obtained by random set of base models, using a fixed set yields sufficiently good performance when compared with the state of art. In contrast to using a fixed set of base models, we can also infer good base models in a classification problem, by selecting as the

base models the class-specific PFSA inferred from the training set.

3) SLD with continuous data: Since PFSA model sequences on finite alphabet, continuous-valued input should first be quantized to discrete ones. The simplest approach of discretization is to choose $k-1$ cut-off points $p_1 < p_2 < \cdots < p_{k-1}$ and replace a value $< p_1$ by 0, in $[p_i, p_{i+1})$ by i, and $\geq p_{k-1}$ by k. We call the set of cut-off points a *partition*. In our implementation, we use the entropy maximization principle to obtain bins in which data points are evenly distributed. If there are clear trends in the data stream, we carry out partitioning after detrending. Inference of the two base models is carried our using the algorithm GenESeSS [4].

C. Inference of Probabilistic Automata

For PFSA inference we use GenESeSS [4], outlined in Algorithm 2.

Algorithm 2: GenESeSS

Data: A sequence x over alphabet Σ, $0 < \varepsilon < 1$
Result: State set Q, transition map δ, and transition probability $\widetilde{\pi}$
/* **Step One: Approximate ε-synchronizing sequence** */
Let $L = \left\lceil \log_{|\Sigma|} 1/\varepsilon \right\rceil$;
Calculate the **derivative heap** $\mathcal{D}_\varepsilon^x$ equaling $\left\{ \hat{\phi}_y^x \ : \ y \text{ is a sub-sequence of } x \text{ with } |y| \leq L \right\}$;
Let C be the convex hull of D_ε^x;
Select x_0 with $\hat{\phi}_{x_0}^x$ being a vertex of C and has the highest frequency in x;
/* **Step Two: Identify transition structure** */
Initialize $Q = \{q_0\}$;
Associate to q_0 the **sequence identifier** $x_{q_0}^{\text{id}} = x_0$ and the probability vector $d_{q_0} = \hat{\phi}_{x_0}^x$;
Let $\widetilde{Q}$ be the set of states that are just added and initialize it to be Q;
while $\widetilde{Q} \neq \emptyset$ **do**
  Let $Q_{\text{new}} = \emptyset$ be the set of new states;
  for $(q, \sigma) \in \widetilde{Q} \times \Sigma$ **do**
    Let $x = x_q^{\text{id}}$ and $d = \hat{\phi}_{x\sigma}^x$;
    if $\left\| d - d_{q'} \right\|_\infty < \varepsilon$ *for some* $q' \in Q$ **then**
      Let $\delta(q, \sigma) = q'$;
    else
      Let $Q_{\text{new}} = Q_{\text{new}} \cup \{q_{\text{new}}\}$ and $Q = Q \cup \{q_{\text{new}}\}$;
      Associate to q_{new} the sequence identifier $x_{q_{\text{new}}}^{\text{id}} = x\sigma$ and the probability vector $d_{q_{\text{new}}} = d$;
      Let $\delta(q, \sigma) = q_{\text{new}}$;
  Let $\widetilde{Q} = Q_{\text{new}}$;
Take a strongly connected subgraph of the labeled directed graph defined by Q and δ, and denote the vertex set of the subgraph again by Q;
/* **Step Three: Identify transition probability** */
Initialize counter $N[q, \sigma]$ for each pair $(q, \sigma) \in Q \times \Sigma$;
Choose a random starting state $q \in Q$;
for $\sigma \in x$ **do**
  Let $N[q, \sigma] = N[q, \sigma] + 1$;
  Let $q = \delta(q, \sigma)$;
Let $\widetilde{\pi}(q) = \left[\!\left[(N[q, \sigma])_{\sigma \in \Sigma} \right]\!\right]$;
return Q, δ, $\widetilde{\pi}$;

III. Notion of Anomaly Discovery

References

[1] J. Bondy and U. Murty, *Graph theory (2008)*, Grad. Texts in Math, (2008).

[2] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR), 41 (2009), pp. 1–58.

[3] I. Chattopadhyay, Y. Huang, and J. Evans, *Deep learning without neural networks: Fractal-nets for rare event modeling*, (2020).

[4] I. Chattopadhyay and H. Lipson, *Abductive learning of quantized stochastic processes with probabilistic finite automata*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371 (2013), p. 20110543.

[5] ———, *Data smashing: uncovering lurking order in data*, Journal of The Royal Society Interface, 11 (2014), p. 20140826.

[6] I. Chattopadhyay and A. Ray, *Structural transformations of probabilistic finite state machines*, International Journal of Control, 81 (2008), pp. 820–835.

[7] T. M. Cover and J. A. Thomas, *Elements of information theory*, John Wiley & Sons, 2012.

[8] J. P. Crutchfield, *The calculi of emergence: computation, dynamics and induction*, Physica D: Nonlinear Phenomena, 75 (1994), pp. 11–54.

[9] V. Hodge and J. Austin, *A survey of outlier detection methodologies*, Artificial intelligence review, 22 (2004), pp. 85–126.

[10] F. T. Liu, K. M. Ting, and Z.-H. Zhou, *Isolation forest*, in 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[11] M. Markou and S. Singh, *Novelty detection: a review—part 1: statistical approaches*, Signal processing, 83 (2003), pp. 2481–2497.

[12] L. R. Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE, 77 (1989), pp. 257–286.

[13] M. Vidyasagar, *Bounds on the kullback-leibler divergence rate between hidden markov models*, in 2007 46th IEEE Conference on Decision and Control, IEEE, 2007, pp. 6160–6165.

[14] M. Vidyasagar, *Hidden markov processes: Theory and applications to biology*, vol. 44, Princeton University Press, 2014.