Applied Artificial Intelligence Institute (A2I2)
Deakin University
March 23, 2025

Problem Overview

Long Van Tran

s224930257@deakin.edu.au

1 Problem Setting

In a conventional setting, we have a time-series dataset $\{\mathbf{X}^{(m)}\}_{m=1}^{M}$,

$$\mathbf{X}^{(m)} = \left[\mathbf{X}_{t_1}^{(m)}, \mathbf{X}_{t_2}^{(m)}, \ldots, \mathbf{X}_{t_k}^{(m)}\right],$$

where $\mathbf{X}_{t_i}^{(m)}$ can be vectors ($\mathbf{X}_{t_i}^{(m)} \in \mathbf{R}^d$) or matrices ($\mathbf{X}_{t_i}^{(m)} \in \mathbf{R}^{t \times d}$), depending on the unit of time being considered here is a single time step or a segment of t steps. Here we consider the latter matrix case. Each of these trajectories describe patient records across k time points/segments (e.g., patient measurements throughout k hours).

We assume that the data is governed by a time-homogeneous linear additive noise stochastic differential equation (SDE), which has the form:

$$\mathrm{d}\mathbf{X}_t = \mathbf{A}\mathbf{X}_t \mathrm{d}t \; + \; \mathbf{G}\mathrm{d}\mathbf{W}_t,$$

with unknown drift-diffusion parameters $\mathbf{A} \in \mathbf{R}^{d \times d}$ and $\mathbf{G} \in \mathbf{R}^{d \times m}$. $\mathbf{W}$ is an m-dimensional Brownian motion.

However, due to various practical reasons (such as data anonymization, incomplete data logs, etc.), the temporal ordering (i.e. the time dimension) might be lost or the time values at which measurements are taken might be noisy (measurement errors). Thus, we want to find solutions for these complex real-world challenges.

1.1 Data without Temporal Order

Our first data complication is measurements taken without any temporal order. Usually, in a time-series dataset we would know the process states through a sequence of time steps. But when this "hidden" time dimension is lost, we face another challenge of sorting the data in the correct temporal order before using it to infer the underlying SDE's parameters.

1.2 Noisy Data Measurements

Other than missing temporal order completely, another complication that can happen to our data is noisy measurements, whereas our data is either measured incorrectly, or the time at which measurements are taken are logged incorrectly. In other words, either the data values $\mathbf{X}_{t_i}^{(m)}$ or their corresponding time points t_i are presented with independent, randomly distributed noise.

For both these data settings, our aim is to estimate the parameters $\mathbf{A}$ and $\mathbf{G}$ while enforcing a Directed Acyclic Graph (DAG) constraint on the adjacency matrix $\mathbf{A}$ (thus avoiding "cyclic" order).

2 Proposed Method

For each of the two mentioned data complications, we proposed a solution outlined as follow:

2.1 Data without Temporal Order

This solution consists of two parts, whereas part two is a two-step iterative scheme where we estimate the parameters of the SDE.

2.1.1 Sorting Data by Variance

- Empirically evaluate the variance of each observed segment of all our trajectories.
- Sort segments in ascending order of variance, with the assumption that we are dealing with a diverging SDE and that a diverging SDE usually has increasing variance over (hidden) time. This can be done in a similar manner for converging SDEs.

2.1.2 Iterative Two-Step Scheme

First, we randomly initialize the parameters $\mathbf{A}^{(0)}$ and $\mathbf{G}^{(0)}$. Then, we follow a two-step iterative scheme as follows:

(a) **Update Previously-Sorted Segments**
Keeping $\mathbf{A}^{(n)}$ and $\mathbf{G}^{(n)}$ at iteration n fixed, rearrange the previously

sorted segments for each trajectory to maximize the log-likelihood minus a DAG penalty Ω_{DAG}:

$$\{\widetilde{\mathbf{X}}^{(m)}\} = \arg\max_{\{\widetilde{\mathbf{X}}^{(m)}\}} \sum_{m=1}^{M} \ln p\big(\widetilde{\mathbf{X}}^{(m)} \mid \mathbf{A}^{(n)}, \mathbf{G}^{(n)}\big) \; - \; \Omega_{\text{DAG}}\big(\{\widetilde{\mathbf{X}}^{(m)}\}\big).$$

A well-known continuous DAG-penalty comes from the NOTEARS approach by Zheng et al., 2018 [3]:

$$h(\mathbf{A}) = \text{trace}(e^{\mathbf{A}\circ\mathbf{A}}) - d,$$

$$\Omega_{\text{DAG}}\big(\{\widetilde{\mathbf{X}}^{(m)}\}\big) = \alpha(h(\mathbf{A})),$$

where d is the number of variables (i.e. the size of the square matrix $\mathbf{A}$) and α is a regularization hyper-parameter. It is proved in [3] that $h(\mathbf{A}) = 0$ if and only if $\mathbf{A}$, our adjacency matrix, corresponds to a DAG (no directed cycles). If cycles exist, $h(\mathbf{A}) > 0$. Hence, adding this term as a penalty encourages $\mathbf{A}$ to remain acyclic.

(b) **Update SDE Parameters**
With the newly completed trajectories fixed, re-estimate the SDE parameters:

$$(\mathbf{A}^{(n+1)}, \mathbf{G}^{(n+1)}) = \arg\max_{\mathbf{A},\mathbf{G}} \sum_{m=1}^{M} \ln p\big(\widetilde{\mathbf{X}}^{(m)} \mid \mathbf{A}^{(n)}, \mathbf{G}^{(n)}\big).$$

This step can be done using the parameter estimation framework named APPEX introduced by Guan et al., 2024 [1].

2.2 Noisy Data Measurements

Based on the theory of Gaussian Process and the work of Mchutchon et al. [2], we propose using the noisy input data/time steps to directly learn the underlying SDE's parameters.

2.2.1 Using a Gaussian Process Surrogate for the Drift

We want to estimate the drift function $f(x)$ of the SDE (in this case the matrix $\mathbf{A}$). But because we only observe noisy inputs $Y_t = X_t + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \Sigma_x)$, fitting a standard GP to the pairs (Y_t, Y_t') would lead to biased estimates due to the input noise. This uncertainty in Y_t affects the

estimate of $f(x)$ non-uniformly — especially in regions where $f(x)$ changes rapidly.

To account for this, we:

- Treat the Euler–Maruyama-discretized form as a supervised learning problem, whereas:

$$Y_{t+1} \approx Y_t + AY_t dt + \text{noise}$$

- Fit a GP to the pairs $\{(Y_t, Y_{t+1})\}$. This GP will provide a posterior mean function $\bar{f}(x) \approx \mathbf{A}x$, which allows us to compute the gradient $\nabla \bar{f}(x)$ analytically. Note that in GPs, derivatives are also GPs and easy to compute.

2.2.2 Applying Noisy-Input GP (NIGP) Correction

NIGP use the first-order Taylor expansion of $f(x + \epsilon_x)$ around the mean input x:

$$f(x + \epsilon_x) \approx f(x) + \nabla \bar{f}(x)^T \epsilon_x$$

Then, the variance in the output due to input noise is approximately:

$$\text{Var}[x + \epsilon_x] \approx \nabla \bar{f}(x)^T \Sigma_x \nabla \bar{f}(x) \approx \nabla \bar{f}(y)^T \Sigma_x \nabla \bar{f}(y)$$

Thus, the GP posterior is corrected with this heteroscedastic variance term instead of assuming constant noise variance Σ_y:

$$\text{Var}(y) = \Sigma_y + \nabla \bar{f}(y)^T \Sigma_x \nabla \bar{f}(y)$$

This gives us a modified GP model that model input noise as heteroscedastic output noise — larger variance in high-gradient regions, tighter confidence in flatter regions.

2.2.3 Constructing Marginal Likelihood

The negative log likelihood (NLL) of the dataset under the GP with this structured noise becomes:

$$\log p(Y_{t+1}|Y_t) = \frac{1}{2} \sum_t [(Y_{t+1} - \phi Y_t)^T (\Sigma_y + \phi^T \Sigma_x \phi)^{-1} (Y_{t+1} - \phi Y_t) + \log|\Sigma_y + \phi^T \Sigma_x \phi|],$$

where Σ_x is the input noise covariance to be inferred. Maximizing the above NLL with respect to:

- $\phi = I + Adt$ gives $\mathbf{A}$
- $\Sigma_y = GG^T dt$ gives $\mathbf{G}$

Our iterative procedure should be:

1. Initialize $\mathbf{A}, \mathbf{A}, \Sigma_x$
2. Use GP with the input-dependent noise $\text{Var}(Y)$ above
3. Minimize the NLL with respect to parameters via gradients (using L-BFGS, Adam, ...)
4. Re-estimate gradients based on updated GP predictions and iterate

3 Notes

Some notes to keep in mind:

- Proving convergence of the iterative scheme.
- Leveraging the identifiability conditions presented in [1].

References

[1] Vincent Guan, Joseph Janssen, Hossein Rahmani, Andrew Warren, Stephen Zhang, Elina Robeva, Geoffrey Schiebinger, Identifying drift, diffusion, and causal structure from temporal snapshots, 2024, available from: `https://arxiv.org/abs/2410.22729`.

[2] Andrew Mchutchon, Carl Rasmussen, Gaussian process training with input noise, in J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Volume 24. Curran Associates, Inc., 2011, available from: `https://proceedings.neurips.cc/paper_files/paper/2011/file/a8e864d04c95572d1aece099af852d0a-Paper.pdf`.

[3] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, Eric P. Xing, Dags with no tears: Continuous optimization for structure learning, 2018, available from: `https://arxiv.org/abs/1803.01422`.