Conference Paper Title*

*Note: Sub-titles are not captured for https://ieeexplore.ieee.org and should not be used

Long Tran
Applied Artificial Intelligence Institute
Deakin University
Geelong, Australia
s224930257@deakin.edu.au

Truyen Tran
Applied Artificial Intelligence Institute
Deakin University
Geelong, Australia
truyen.tran@deakin.edu.au

Phuoc Nguyen
Applied Artificial Intelligence Institute
Deakin University
Geelong, Australia
phuoc.nguyen@deakin.edu.au

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

To be written.

II. Related Work

To be written. [1]

III. Methodology

We assume the stochastic differential equation (SDE) to be time-homogeneous linear additive noise and defined in $\mathbb{R}^d$, its form is as follows:

$$d\mathbf{X}_t = \mathbf{A}\mathbf{X}_t dt + \mathbf{G} d\mathbf{W}_t, \tag{1}$$

where:

- $\mathbf{A} \in \mathbb{R}^{d\times d}$ is the drift matrix.
- $\mathbf{G} \in \mathbb{R}^{d\times m}$ is the diffusion matrix; its *instantaneous* covariance is $\mathbf{H} := \mathbf{G}\mathbf{G}^\top \succ 0$ ($\mathbf{H} \in \mathbb{R}^{d\times d}$).
- $\mathbf{W}_t$ is a *m*-dimensional standard Wiener process.

We observe *N* independent trajectories $\{\mathbf{X}^{(j)}_{i\Delta t}\}_{i=0}^{T}$ on an equally-spaced grid $t_i = i\Delta t$ (step-size $\Delta t >$), where *T* is the total time steps and the increments are denoted as $\Delta\mathbf{X}_i^{(j)} = \mathbf{X}_{i+1}^{(j)} - \mathbf{X}_i^{(j)}$ $(j = 0, 1, ..., N)$.

As mentioned earlier, our problem is the missing temporal order, which means the steps presented in the trajectories are not in the correct order with respect to time. To deal with this problem, we propose our score-based iterative method. For each iteration, there will be two parts presented in section A and B below, respectively.

Identify applicable funding agency here. If none, delete this.

A. Estimate SDE parameters from Maximum Likelihood

For small Δt the exact transition can be replaced by the first-order scheme

$$\Delta\mathbf{X}_i^{(j)} = \mathbf{A}\mathbf{X}_i^{(j)}\Delta t + \boldsymbol{\varepsilon}_i^{(j)},\ \boldsymbol{\varepsilon}_i^{(j)} \sim \mathcal{N}(\mathbf{0},\, \mathbf{H}\Delta t). \tag{2}$$

Since each increment is Gaussian, we can write down the probability density of one increment as follows:

$$\begin{aligned} p(\Delta\mathbf{X}_i^{(j)} \mid \mathbf{X}_i^{(j)}; \mathbf{A}, \mathbf{H}) &= \frac{1}{(2\pi)^{d/2}|\mathbf{H}\Delta t|^{1/2}}\cdot \\ \exp\Big(-\frac{1}{2}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]^\top (\mathbf{H}\Delta t)^{-1}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]\Big) \\ &= (2\pi\Delta t)^{-d/2}|\mathbf{H}|^{-1/2}\cdot \\ \exp\Big(-\frac{1}{2\Delta t}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]^\top \mathbf{H}^{-1}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]\Big). \end{aligned} \tag{3}$$

Assuming independence between increments and trajectories, the joint likelihood of all increments is the product of each conditional likelihood:

$$\text{Likelihood}(\mathbf{A}, \mathbf{H}) = \prod_{j=1}^{N}\prod_{i=0}^{T-1} p(\Delta\mathbf{X}_i^{(j)} \mid \mathbf{X}_i^{(j)}; \mathbf{A}, \mathbf{H}).$$

Taking logarithms, we have the log-likelihood function $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}(\mathbf{A}, \mathbf{H}) &= \log(\text{Likelihood}(\mathbf{A}, \mathbf{H})) \\ &= \sum_{j=1}^{N}\sum_{i=0}^{T-1} \log p(\Delta\mathbf{X}_i^{(j)} \mid \mathbf{X}_i^{(j)}; \mathbf{A}, \mathbf{H}). \end{aligned} \tag{4}$$

Substituting explicitly from (4):

$$\begin{aligned} \mathcal{L}(\mathbf{A}, \mathbf{H}) = \sum_{j=1}^{N}\sum_{i=0}^{T-1}\Bigg[&-\frac{d}{2}\log(2\pi\Delta t) - \frac{1}{2}\log|\mathbf{H}| \\ &- \frac{1}{2\Delta t}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]^\top \mathbf{H}^{-1}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]\Bigg] \\ = &-\frac{dNT}{2}\log(2\pi\Delta t) - \frac{NT}{2}\log|\mathbf{H}| \\ &-\frac{1}{2\Delta t}\sum_{j=1}^{N}\sum_{i=0}^{T-1}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]^\top \mathbf{H}^{-1}[\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t]. \end{aligned} \tag{5}$$

Then, given the current trajectories (either the original unordered trajectories if it is the first iteration, or the reordered trajectories from the previous iteration), we can update the SDE parameters via maximum likelihood estimation.

1) Fixing $\mathbf{H}$ *and estimate* $\mathbf{A}$*:*
First, we define the residuals:

$$\mathbf{R}_i^{(j)}(\mathbf{A}) = \Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t$$

The log-likelihood, as a function of $\mathbf{A}$ since $\mathbf{H}$ is fixed, is:

$$\mathcal{L}(\mathbf{A}) \propto -\frac{1}{2\Delta t}\sum_{j=1}^{N}\sum_{i=0}^{T-1}\mathbf{R}_i^{(j)}(\mathbf{A})^\top\mathbf{H}^{-1}\mathbf{R}_i^{(j)}(\mathbf{A})$$

Then, we set the gradient of $\mathcal{L}(\mathbf{A})$ to zero to find $\hat{\mathbf{A}}$:

$$\nabla_{\mathbf{A}}\mathcal{L}(\mathbf{A}) = 0 \Rightarrow \sum_{i,j}\mathbf{H}^{-1}\Big(\Delta\mathbf{X}_i^{(j)} - \mathbf{A}\mathbf{X}_i^{(j)}\Delta t\Big)\mathbf{X}_i^{(j)\top} = 0$$

Since $\mathbf{H}^{-1}$ is invertible, this can be simplified to:

$$\sum_{i,j}\Delta\mathbf{X}_i^{(j)}\mathbf{X}_i^{(j)\top} = \mathbf{A}\Delta t\sum_{i,j}\mathbf{X}_i^{(j)}\mathbf{X}_i^{(j)\top}$$

Provided the Gram matrix $\sum_{i,j}\mathbf{X}_i^{(j)}\mathbf{X}_i^{(j)\top}$ is invertible, the explicit MLE solution for $\mathbf{A}$ is:

$$\hat{\mathbf{A}} = \frac{1}{\Delta t}\Big(\sum_{i,j}\Delta\mathbf{X}_i^{(j)}\mathbf{X}_i^{(j)\top}\Big)\Big(\sum_{i,j}\mathbf{X}_i^{(j)}\mathbf{X}_i^{(j)\top}\Big)^{-1} \quad (6)$$

2) Fixing $\mathbf{A}$ *and estimate* $\mathbf{H}$*:*
Given the estimated $\hat{\mathbf{A}}$, we now begin to estimate $\mathbf{H}$ by substituting $\hat{\mathbf{A}}$ into the residuals. The covariance estimator $\hat{\mathbf{H}}$ is the covariance of these residuals:

$$\hat{\mathbf{H}} = \frac{1}{NT\Delta t}\sum_{i,j}\Big(\Delta\mathbf{X}_i^{(j)} - \hat{\mathbf{A}}\mathbf{X}_i^{(j)}\Delta t\Big)\cdot \Big(\Delta\mathbf{X}_i^{(j)} - \hat{\mathbf{A}}\mathbf{X}_i^{(j)}\Delta t\Big)^\top \quad (7)$$

B. Update previously-sorted trajectories

After estimating the SDE parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$ based on the current ordering of trajectories (or the initial unordered data), the next step is to re-evaluate and potentially reorder the sequence of observed states $\{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_T\}$. The goal is to find the permutation $\pi = (\pi_0, \pi_1, \ldots, \pi_T)$ of the original indices $\{0, 1, \ldots, T\}$ that maximizes the likelihood of the observed sequence under the estimated SDE dynamics.

For an Itô-diffusion with constant diffusion matrix $\mathbf{G}$,

$$d\mathbf{X}_t = b(\mathbf{X}_t)dt + \mathbf{G}d\mathbf{W}_t,$$

the **time-reversed** process is again a diffusion whose drift $\bar{b}$ satisfies the time-reversal (cite Nelson-Haussmann-Pardoux) identity:

$$\bar{b}(\mathbf{X}_t) = b(\mathbf{X}_t) - \mathbf{G}\mathbf{G}^\top\nabla_x \log p_t(x) \quad (8)$$

with the same diffusion coefficient $\mathbf{G}$. Equation (8) says the gap between forward and backward drift equals the scaled score (gradient of the log-density). Thus, if we have snapshots at two instants t and s (unknown order) and a short time-lag $\Delta t = |s - t|$, the direction that makes (8) fit the data best is (with high probability) the correct temporal order.

The marginal distribution at time t of the linear Gaussian SDE is Gaussian, expressed as:

$$p_t(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t),$$

where the mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$ satisfy the following ODEs:

$$\frac{d}{dt}\boldsymbol{\mu}_t = \mathbf{A}\boldsymbol{\mu}_t,$$
$$\frac{d}{dt}\boldsymbol{\Sigma}_t = \mathbf{A}\boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_t\mathbf{A}^\top + \mathbf{H}.$$

From these Gaussian approximations, the *score function* at timestep t is:

$$\nabla_{\mathbf{x}}\log p_t(\mathbf{x}) = -\boldsymbol{\Sigma}_t^{-1}(\mathbf{x} - \boldsymbol{\mu}_t).$$

Practically, given data, we estimate:

$$\hat{\boldsymbol{\mu}}_t = \frac{1}{N}\sum_{j=1}^{N}\mathbf{X}_t^{(j)}, \quad (9)$$

$$\hat{\boldsymbol{\Sigma}}_t = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{X}_t^{(j)} - \hat{\boldsymbol{\mu}}_t)(\mathbf{X}_t^{(j)} - \hat{\boldsymbol{\mu}}_t)^\top. \quad (10)$$

Thus, the empirical Gaussian score, to be used for (8), becomes:

$$\nabla_{\mathbf{x}}\log\hat{p}_t(\mathbf{x}) = -\hat{\boldsymbol{\Sigma}}_t^{-1}(\mathbf{x} - \hat{\boldsymbol{\mu}}_t). \quad (11)$$

Equation (11) gives us the score function at time t. Using this score function, we can compute the scores at two time points t, s and the corresponding estimated errors for the estimated drift between the two time points. The empirical drift $\hat{\mathbf{b}}$ is computed as:

$$\hat{\mathbf{b}} = \mathbb{E}[\mathbf{X}_s - \mathbf{X}_t]/\Delta t. \quad (12)$$

The error between the empirical drift and the drift predicted by the score function scaled by $\mathbf{H}$ is:

$$\text{Error}(\mathbf{X}_t) = ||\hat{\mathbf{b}} - \mathbf{H}\cdot\nabla_{\mathbf{x}}\log\hat{p}_t(\mathbf{x})||^2. \quad (13)$$

Putting all the pieces together, we can compute the score function at time t using (11) and the empirical drift using (12). The error between the empirical drift and the drift predicted by the score function scaled by $\mathbf{H}$ is computed using (13). Then we can compare the errors at two time steps, whichever step has the smaller error is therefore assumed to be more likely to fit into the **time-reversed** process. Algorithm 1 below shows in detail how we leverage this idea to reorder the trajectories point-by-point.

Algorithm 1 combines parameter estimation and trajectory reordering in an iterative loop. In each iteration, the SDE parameters (drift A and diffusion H) are first updated using the current order of the data. Then, for each adjacent pair of time steps, the algorithm:

- Computes the empirical score (the gradient of the log-density estimate) at the two time points.

Algorithm 1 Score-based iterative method for estimating SDE parameters and reordering trajectories.

Input: Un-ordered data $\mathbf{X}$, number of iterations $\mathbf{K}$, number of timesteps T

Output: Re-ordered data $\tilde{\mathbf{X}}$ and estimated parameters $\mathbf{A}$, $\mathbf{H}$

for $i = 1$ to K **do**
  $\hat{\mathbf{A}}$, $\hat{\mathbf{H}} \leftarrow$ SDE-Parameters-Solving (from (6) and (7))
  $end \leftarrow T - 1$
  while $end \neq 0$ **do**
    for $t = 0, ..., end - 1$ **do**
      $s \leftarrow t + 1$
      Score$(\mathbf{X}_t) = \nabla_{\mathbf{x}} \log \hat{p}_t(\mathbf{x})$
      Score$(\mathbf{X}_s) = \nabla_{\mathbf{x}} \log \hat{p}_s(\mathbf{x})$
      $\hat{\mathbf{b}} = \mathbb{E}[\mathbf{X}_s - \mathbf{X}_t]/\Delta t$ (Empirical Drift)
      Error$(\mathbf{X}_t) = ||\hat{\mathbf{b}} - \mathbf{H} \cdot \text{Score}(\mathbf{X}_t)||^2$
      Error$(\mathbf{X}_s) = ||\hat{\mathbf{b}} - \mathbf{H} \cdot \text{Score}(\mathbf{X}_s)||^2$
      if Error$(\mathbf{X}_t) <$ Error$(\mathbf{X}_s)$ **then**
        Swap $\mathbf{X}_t$ and $\mathbf{X}_s$, since $s > t$ is more likely
      end if
    end for
  end while
  Check-convergence $\rightarrow$ True: Stop Algorithm
end for
return $\tilde{\mathbf{X}}$, $\hat{\mathbf{A}}$, and $\hat{\mathbf{H}}$

- Calculates an empirical drift estimate from the change in state over the time interval.
- Quantifies the discrepancy between the empirical drift and the drift predicted by the score function scaled by H.
- Compares the computed errors; if the error for the earlier time point exceeds that for the later point, a swap is performed, indicating that the observed order is likely reversed.

This process is repeated in a similar manner to the popular sorting algorithm named Bubble Sort, until the ordering converges, ensuring that the temporal direction in the data aligns optimally with the underlying SDE dynamics.

IV. Experiments

In this section, we write down all the implementation details of our proposed method and different experiments, including the main setting, such that the readers can reproduce our results. We also introduce the datasets and baseline method we used in our experiments and the evaluation metrics. Our code is available on Github[1].

A. Datasets and Experiments

1) Main setting: Our main setting uses the data following the time-homogeneous linear additive noise SDE form mentioned in (1), which is generated using Euler-Maruyama discretization:

$$\mathbf{X}_{i+1}^{(j)} = \mathbf{X}_i + \mathbf{A}\mathbf{X}_i^{(j)}\Delta t + \mathbf{G}d\mathbf{W}_i.$$

The dataset typically has shape $(num\_trajectories, num\_timesteps, d)$, where d is the dimension of each data point. From this, we then randomize the data along the time-step dimension to obtain the data with lost temporal order as input. This input data will then be processed using our proposed method to obtain the temporally sorted data and the estimated parameters $\mathbf{A}, \mathbf{H}$.

In this main setting, we test our method many times to demonstrate how changes in time step size Δt, total time period T, data dimension d, number of trajectories, and drift-diffusion parameter initialization individually affect its performance.

2) Noisy Measurement Setting: Many previous work **cite here** also explore the problem of estimating parameters when given noisy data. Thus, we also generate data that are corrupted by Gaussian noise at each time step:

$$\mathbf{Y}_t = \mathbf{X}_t + \boldsymbol{\epsilon}_t, \ \ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}),$$

where $\mathbf{R} = \sigma_\epsilon^2 \mathbf{I}$.

Because both the state and the noise are Gaussian, the observed data at a single time-slice are still Gaussian, so we can still use our proposed Algorithm 1, with some small modifications from (9) and (10):

$$\hat{\mathbf{S}}_t = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{Y}_t^{(j)} - \hat{\boldsymbol{\mu}}_t)(\mathbf{Y}_t^{(j)} - \hat{\boldsymbol{\mu}}_t)^\top,$$
$$\hat{\boldsymbol{\Sigma}}_t = \hat{\mathbf{S}}_t - \mathbf{R},$$

where $\hat{\mathbf{S}}_t$ is the sample covariance of $\mathbf{Y}_t$ instead of $\mathbf{X}_t$, we use the same mean $\hat{\boldsymbol{\mu}}_t$ since the measurement noise $\boldsymbol{\epsilon}_t$ is zero-mean.

3) Real Datasets and Evaluation: Ornstein-Uhlenbeck processes are a popular example of a time-homogeneous linear additive noise SDE, and are used in many real-world examples **cite here**. Thus, we test our algorithm on some real datasets as follows:

- US Treasury yields – FRED "DGS10"[2]: Daily 10-year constant-maturity Treasury yield (plus dozens of other terms) since 1962.
- Single-particle Brownian trajectories – Figshare "Dynamic and asymmetric colloidal molecules"[3]: This dataset contains time-stamped (x, y, z, t) coordinates of fluorescent colloidal particles acquired with 3-D confocal microscopy. Translational motion of each bead in an optical trap is an OU process with additive (thermal) diffusion.
- UCI "Individual Household Electric Power Consumption"[4]: Minute-level active/reactive power, current and voltage for one French household over 47 months (about 2 million time steps). A multivariate OU process can capture short-time mean-reversion of load and voltage

[1] Github

[2] FRED

[3] Molecules

[4] Power

around diurnal trends. Furthermore, additive Gaussian measurement error is explicit in the documentation.

Generally, ground-truth parameters are absent for real-world datasets. So we cannot simply use Mean Absolute Error (MAE) or Mean Squared Error (MSE) as evaluation metrics like in the case of synthetic data. Instead, we use some different evaluation techniques as follow:

- Out-of-sample negative log–likelihood (NLL), or MAE:
  1) Split the time series into train/test blocks.
  2) Apply our model to the train part, and use it as a generative model to predict the next step with the transition formula (2).
  3) Then compute NLL/MAE on the test block. Lower NLL/MAE often suggest better fit.
- Coverage of prediction intervals:
  1) Using the fitted parameters, generate 95% predictive ellipsoids.
  2) Count how often the next data point falls inside.
  3) Over many steps the data should hit $\approx 95\%$. Under-coverage can suggest diffusion underestimated and over-coverage means over-estimated model.
- Posterior predictive checks: Answer the question "If my fitted SDE were the truth, would I see time series that look like my data?".
  1) Simulate many synthetic datasets from the fitted parameters.
  2) Compute relevant summary statistics (variance, hitting times, etc.) for both synthetic and real data.
  3) Then visualise the real statistic inside the simulated distribution (histogram / box-plot). Large discrepancies suggest estimator (or model class) inadequate.

B. Implementation Details

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

[1] V. Guan, J. Janssen, H. Rahmani, A. Warren, S. Zhang, E. Robeva, and G. Schiebinger, "Identifying drift, diffusion, and causal structure from temporal snapshots," 2024. [Online]. Available: https://arxiv.org/abs/2410.22729