Probability Paths and the Structure of Predictions over Time

Zhiyuan (Jerry) Lin
Stanford University

Hao Sheng
Stanford University

Sharad Goel
Stanford University

Abstract

In settings ranging from weather forecasts to political prognostications to financial projections, probability estimates of future binary outcomes often evolve over time. For example, the estimated likelihood of rain on a specific day changes by the hour as new information becomes available. Given a collection of such *probability paths*, we introduce a Bayesian framework—which we call the Gaussian latent information model, or GLIM—for modeling the structure of dynamic predictions over time. Suppose, for example, that the likelihood of rain in a week is 50%, and consider two hypothetical scenarios. In the first, one expects the forecast is equally likely to become either 25% or 75% tomorrow; in the second, one expects the forecast to stay constant for the next several days. A time-sensitive decision-maker might select a course of action immediately in the latter scenario, but may postpone their decision in the former, knowing that new information is imminent. We model these trajectories by assuming predictions update according to a latent process of information flow, which is inferred from historical data. In contrast to general methods for time series analysis, this approach preserves the martingale structure of probability paths. We show that GLIM outperforms two popular baseline methods, producing predictions that are both better calibrated and better capture the volatility of forecasts. By elucidating the dynamic structure of predictions over time, we hope to help individuals make more informed choices.

1 Introduction

Probabilistic predictions of future binary outcomes are ubiquitous in real-world statistical and machine learning problems, ranging from election modeling (Erikson and Wlezien, 2012; Shirani-Mehr et al., 2018) to weather forecasting (Esteves et al., 2019) to assessing mortality risk (Teres et al., 1987; Malmberg et al., 1997; Martinez-Alario et al., 1999; Yan et al., 2020). Often such probabilistic predictions are not static, but rather evolve over time as more information becomes available. For instance, the estimated likelihood a candidate wins an election updates with every new poll that is conducted, and the estimated chance of rain on a given future day changes by the hour as new meteorological data become available.

In many of these domains, a large body of work aims to provide accurate, real-time forecasts that account for the very latest information. However, significantly less attention has been paid to understanding how the predictions themselves evolve over time. If there's a 35% chance of rain one week from now, what can we say about tomorrow's prediction of rain on that same day? In expectation, tomorrow's prediction must be the same as today's—if it were not, we would be better off updating today's prediction to match our expectation of tomorrow's forecast. But aside from satisfying this martingale property, the forecast trajectories can vary widely from one setting to the next.

Consider, for example, the collection of dynamic weather forecasts presented in Figure 1. The plot shows the evolution of 7-day precipitation predictions for a set of Australian cities (Williams, 2011; Young and Young, 2018), where we have disaggregated the predictions by season (summer or winter) and restricted to those instances for which the 7-day forecast starts at approximately 35%. As is visually apparent, the mean prediction on any later day is still approximately 35%, as expected. But the summer forecasts are considerable more volatile than those in the winter. In the winter months (June–August), the prediction is unlikely to change substantially until immediately before the target date. In contrast, in the summer months

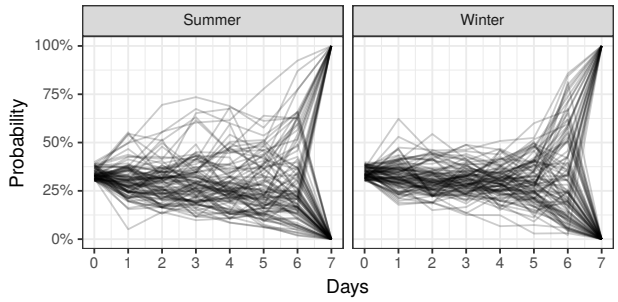


Figure 1: Dynamic forecasts of rainfall across several Australian cities during the summer and winter, starting one week before the target date. All forecasts initially indicate an approximately 35% chance of rain. But the forecast trajectories—which we call probability paths—have higher volatility in the summer than in the winter.

(December–February), the prediction is likely to oscillate considerably—either up or down—with each passing day.

These patterns have immediate consequences for time-sensitive decision makers. As a simple example, suppose one is planning a weekend picnic, with a preference both for sunshine and for sending out invitations as soon as possible. If one knows that a weather forecast is unlikely to change in the immediate future, one might opt to simply send out the invitations and hope for the best. Alternatively, knowing that more information is imminent—as reflected by forecasts that are likely to soon change—one might choose to wait another day before deciding to hold the party.

In this paper, we develop a Bayesian time-series model to investigate—and predict—the structure of dynamic probabilistic forecasts: the Gaussian latent information model, or GLIM. In contrast to existing techniques for time-series modeling, our approach is tailored to the specific properties of evolving probabilistic forecasts, which we call probability paths. Most importantly, our approach satisfies the martingale property described above, with the expected value of future predictions along the path equal to the current prediction. Our model is trained directly on historical paths, and learns the heterogeneous, covariate-dependent structure of forecasts. As a result, it is rich enough to predict the type of structure illustrated in Figure 1. We show that this approach often outperforms existing state-of-the-art methods for modeling time series, ostensibly because past general-purpose methods do not leverage the idiosyncratic properties of probability paths. As real-time forecasting becomes increasingly popular, we hope this work helps researchers and decision makers investigate and act on these dynamic predictions.

2 Related Work

Dynamic probability forecasts—what we call probability paths—have attracted attention in a wide range of fields, particularly finance (Harvey and Shephard, 1993), economics (Diebold et al., 1993; Kitsul and Wright, 2013), and, more recently, robotics (Park et al., 2010). Typically, though, these probability paths are modeled using methods for general time-series analysis, and there is a dearth of work that aims to account for their specific statistical properties.

Our approach builds on the time-series literature by drawing on two broad techniques: first, like stochastic volatility models, our approach captures covariate-dependent volatility over time; and, second, we incorporate latent variables, as in the state-space modeling approach. Finally, our work has implications for the decision-theoretic literature on optimal stopping. We briefly discuss these connections below.

Stochastic volatility models. The earliest approaches to modeling volatility typically assume a recurrent dependency structure: ARCH (Engle, 1982) assumes the variance of the current error term—or innovation—to be a function of the error terms in previous time periods; GARCH (Bollerslev, 1986) includes previous values of the process into the function as well; and GARCHX (Francq and Thieu, 2019; Engle and Patton, 2007) extends the list to cross-sectional covariates. Augmented ARCH (Bera et al., 1992) brought in a more complicated temporal structure by allowing the conditional variance to depend on cross-products of the lagged innovations.

Instead of recursively updating the error terms, one can explicitly model their joint distribution. For example, for each prediction made at time s about the forecast at time t, the Martingale Model of Forecast Evolution (MMFE) (Heath and Jackson, 1994) models the covariance of the predicted change, simultaneously for all (s, t) pairs. More recently, the Gaussian Copula Process Volatility (GCPV) model (Wilson and Ghahramani, 2010) was proposed to facilitate covariate-dependent innovation covariance.

It is worth noting that most of these methods are primarily designed to model a single time-series, rather than to model the multiple, independent time series in our motivating examples. Further, while some of these approaches may still be applied to model probability paths, they do not fully leverage their distinctive characteristics, such as their martingale property, or their finite time horizon (meaning the predictions converge to 0 or 1 at a fixed time T), as discussed below.

State-space modeling. State-space models assume that an unobserved series of vectors determine the evo-

lution of a process. A common class of state-space models is the Hidden Markov Model (HMM) (Cheng and Li, 2011; Hassan and Nath, 2005), which models the latent state transitions directly. Among similar models with hidden states is Kalman filter (Welch et al., 1995), which has been widely used in many applications from modeling water demand (Nasseri et al., 2011) to estimating crop yield (De Wit and Van Diepen, 2007). These classic approaches typically assume a static noise distribution. Modern deep-learning based recurrent neural networks such as LSTM (Hochreiter and Schmidhuber, 1997) can also be viewed as complicated state-space models. With much higher model capacity, it is believed to approximate the posterior even better with random regularization (Srivastava et al., 2014; Gal and Ghahramani, 2015; Kingma et al., 2015) or weights distribution (Blundell et al., 2015). All of these techniques are powerful methods for modeling general time-series data, but none are tailored to the idiosyncratic properties of probability paths.

Optimal stopping. Finally, we note that our work connects to the extensive literature on optimal stopping. Given an accurate estimate of how predictions evolve over time, many time-dependent decision problems can be viewed as optimal stopping problems (Chow and Robbins, 1963; Ferguson, 2004; Jacka, 1991), a subclass of Markov decision process (MDP) problems in which the decision-maker aims to find the optimal time of taking a particular action to maximize utility. Previous literature (Tsitsiklis and Van Roy, 1999; Bingham and Peskir, 2006; Dumitrescu et al., 2016) has shown that such problems can be solved efficiently via (approximate) dynamic programming or its variants. Modern reinforcement learning has started tackling the optimal stopping problem without explicitly modeling the underlying probability distributions (Becker et al., 2019). However, in contrast to our setting, these approaches require specifying a utility function to maximize. In our case, the goal is to model the probability paths themselves, rather than to optimize for a specific decision.

3 Modeling Probability Paths

Suppose we are interested in a binary outcome $Y_T \in \{0, 1\}$ that will either occur or not at some fixed future time T and the current time is $t = 0$. For example, we may be interested in whether it will rain $T = 7$ days from today. At time 0, we observe an initial probabilistic point estimate Y_0 about the final outcome (e.g., from a static weather forecast model), as well as a collection of covariates X (e.g., current and recent meteorological measurements). Our goal, then, is to accurately estimate how the predictions $\{Y\}_{t=0}^{T}$ of the event in question will evolve over time. To do so, we assume access to a dataset of past probability paths, annotated with the same set of covariates X considered above.

3.1 Gaussian latent information model

To model a probability path $\{Y\}_{t=0}^{T}$, we introduce a corresponding sequence of latent variables $\{Z\}_{t=1}^{T}$, where the scalar Z_t intuitively represents the amount of information received between time $t-1$ and t. Receiving positive information (i.e., positive Z_t) makes the ultimate outcome Y_T more likely, while receiving negative information makes the ultimate outcome less likely.

Now, we assume the latent variables $\{Z\}_{t=1}^{T}$ follow a multivariate Gaussian distribution with mean zero and $T \times T$ covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}(X, \theta)$, where the covariance matrix can depend on both the observed covariates X and a parameter θ. Finally, we model $\{Y\}_{t=0}^{T}$ by assuming predictions can be expressed in terms of the following conditional probabilities:

$$Y_t = \Pr\left(\gamma + \sum_{i=1}^{T} Z_i \geq 0 \mid Z_1, \ldots, Z_t\right), \qquad (1)$$

where γ is a fixed constant.

Eq. (1) is closely related to the latent variable formulation of logistic regression models, but where we additionally condition on the latent variables. This expression has several attractive properties for modeling probability paths. First, Y_t is naturally constrained to lie between 0 and 1. Second, Y_T is guaranteed to be either 0 or 1, since once we condition on $Z_1, \ldots, Z_T$, there is no randomness left in the expression. Finally, and most importantly, Y_t satisfies the martingale property, as shown below.

Proposition 1. *For* $\{Y\}_{t=0}^{T}$ *satisfying Eq.* (1)*, we have*

$$\mathbb{E}[Y_{t+1} \mid Y_0, \ldots, Y_t] = Y_t$$

for $0 \leq t < T$.

Proof. The proof follows by repeatedly applying the law of iterated expectations. In particular,

$$\begin{aligned}
&\mathbb{E}[Y_{t+1} \mid Y_0, \ldots, Y_t] \\
&\quad = \mathbb{E}[\mathbb{E}[Y_{t+1} \mid Z_1, \ldots, Z_t] \mid Y_0, \ldots, Y_t] \\
&\quad = \mathbb{E}[\mathbb{E}[\mathbb{E}[Y_T \mid Z_1, \ldots, Z_{t+1}] \mid Z_1, \ldots, Z_t] \mid Y_0, \ldots, Y_t] \\
&\quad = \mathbb{E}[\mathbb{E}[Y_T \mid Z_1, \ldots, Z_t] \mid Y_0, \ldots, Y_t] \\
&\quad = \mathbb{E}[Y_t \mid Y_0, \ldots, Y_t] \\
&\quad = Y_t.
\end{aligned}$$

□

Given a collection of probability paths, our goal is to infer the parameter θ, as we describe next.

3.2 Model inference

We take a Bayesian approach to fitting our Gaussian latent information model (GLIM) defined above. The main difficulty is deriving a tractable expression for the corresponding likelihood function. To do so, we first introduce some notation.

Given any $0 < t < T$, we divide the covariance matrix $\mathbf{\Sigma}$ into four quadrants as follows:

$$\mathbf{\Sigma} = \begin{bmatrix} \overbrace{\mathbf{\Sigma}^t_{11}}^{t} & \overbrace{\mathbf{\Sigma}^t_{12}}^{T-t} \\ \mathbf{\Sigma}^t_{21} & \mathbf{\Sigma}^t_{22} \end{bmatrix} \begin{matrix} \} \\ \} \end{matrix} \begin{matrix} t \\ T-t \end{matrix}$$

For $t = 0$, we further define $\mathbf{\Sigma}^0_{22} = \mathbf{\Sigma}$. For an arbitrary matrix M, we use the notation $\mathbf{M}_{(i,j)}$ to denote the (i, j) entry in the matrix. Similarly, for an arbitrary vector $\mathbf{v}$, we use the notation $\mathbf{v}_{(i)}$ to denote the i-th entry in the vector.

Now, using this notation, Theorem 1 presents a recursive formula for computing the (log) probability density function for our Gaussian latent information model with parameter θ. That expression can in turn be used to infer the parameters of the model from the observed data via maximum likelihood estimation or, alternatively, fully Bayesian inference.

Theorem 1. *Consider a probability path $y_0, \ldots, y_T$, with associated covariate vector X. For parameter θ and covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}(X, \theta)$, suppose Y_t is given by the GLIM model, defined in Eq.* (1)*. Then there is a unique γ such that $Y_0 = y_0$:*

$$\gamma = \Phi^{-1}(y_0)\sqrt{\sum_{i,j} \mathbf{\Sigma}_{(i,j)}}, \tag{2}$$

where Φ is the CDF of the standard normal distribution. Further, the log probability density function f of the path $y_1, \ldots, y_T$ under the model is given by

$$\begin{aligned} &\log f(y_1, \ldots, y_T; \mathbf{\Sigma}, \gamma) \\ &= -\sum_{t=1}^{T-1} \left(\log \tilde{\sigma}_t + \frac{(\Phi^{-1}(y_t) - \tilde{\mu}_t)^2}{2\tilde{\sigma}_t^2} - \frac{(\Phi^{-1}(y_t))^2}{2} \right) \\ &\quad + y_T \cdot \log(y_{T-1}) + (1 - y_T) \cdot \log(1 - y_{T-1}), \end{aligned}$$

where $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ are computed according to the following four-step procedure:

1. *For $1 \le t < T$, calculate $\mathbf{\Sigma}^t$ and $\mathbf{a}^t$ according to the following expressions (and set $\mathbf{\Sigma}^0 = \mathbf{\Sigma}$):*

$$\begin{aligned} \mathbf{\Sigma}^t &= \mathbf{\Sigma}^t_{22} - \mathbf{\Sigma}^t_{21}(\mathbf{\Sigma}^t_{11})^{-1}\mathbf{\Sigma}^t_{12} \\ \mathbf{a}^t &= \mathbf{1}^T \mathbf{\Sigma}^t_{21}(\mathbf{\Sigma}^t_{11})^{-1} \end{aligned}$$

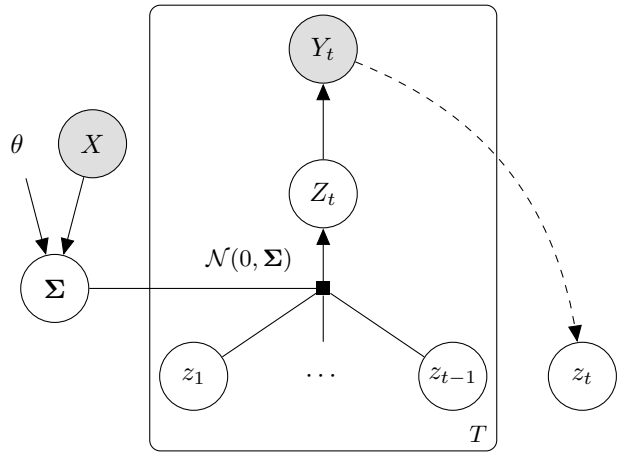


Figure 2: The directed graphical model under consideration of GLIM. Solid lines denote the generative model. Dashed lines denote the identification of z_t from realized values of Y_t; this induction also requires knowledge of Σ and $z_1, \cdots, z_{t-1}$, but those arrows are omitted in the diagram for simplicity. The parameters θ of the covariance matrix Σ are learned from the data.

2. *For $1 \le t < T$, iteratively compute z_t:*

$$z_t = \frac{\sqrt{\sum_{i,j} \mathbf{\Sigma}^t_{(i,j)}}\Phi^{-1}(y_t) - \sum_{i=1}^{t-1}(1 + \mathbf{a}^t_{(i)})z_i - \gamma}{1 + \mathbf{a}^t_{(t)}}$$

3. *For $1 \le t < T$, calculate μ^t (and set $\mu^0 = \mathbf{0}$):*

$$\mu^t = \mathbf{\Sigma}^t_{21}(\mathbf{\Sigma}^t_{11})^{-1}[z_1, \cdots, z_t]^T$$

4. *Finally, for $1 \le t < T$, calculate $\tilde{\mu}_t$ and $\tilde{\sigma}_t$*

$$\begin{aligned} \tilde{\mu}_t &= \frac{\gamma + \sum_{i=1}^{t-1}(1 + \mathbf{a}^t_{(i)})z_i + (1 + \mathbf{a}^t_{(t)})\mu^{t-1}_{(1)}}{\sqrt{\sum_{i,j} \mathbf{\Sigma}^t_{(i,j)}}} \\ \tilde{\sigma}_t &= \frac{\sqrt{\mathbf{\Sigma}^{t-1}_{(1,1)}}(1 + \mathbf{a}^t_{(t)})}{\sqrt{\sum_{i,j} \mathbf{\Sigma}^t_{(i,j)}}}. \end{aligned}$$

A proof of Theorem 1 is given in the Appendix, and, in Figure 2, we provide a graphical representation of the model and inference process. The key observation is that given realized values of $Y_1, \ldots, Y_t$, together with the parameters of the underlying data-generating process, one can compute the implied values of the latent variables $Z_1, \ldots, Z_t$. Using change of variables, one can then compute the probability density function itself.

Much of the notational complexity in Theorem 1 stems from the fact that the latent variables can have a non-trivial correlation structure. This flexibility allows our model to better capture the patterns of real-world

data. However, in some cases, it is sufficient to assume the latent variables are independent, which in turn considerably simplifies our expression of the density.

Corollary 1. *Suppose the latent variables* $\{Z\}_{t=1}^T$ *are independent, with a diagonal covariance matrix* $\mathbf{\Sigma}$ *having diagonal entries* $\sigma_1^2, \sigma_2^2, \cdots, \sigma_T^2$. *Then the expressions for* $\tilde{\mu}_t$ *and* $\tilde{\sigma}_t$ *in Theorem 1 have the following simplified form:*

$$\tilde{\mu}_t = \Phi^{-1}(y_{t-1}) \sqrt{\frac{\sum_{i=t}^T \sigma_i^2}{\sum_{i=t+1}^T \sigma_i^2}},$$

$$\tilde{\sigma}_t = \frac{\sigma_t}{\sqrt{\sum_{i=t}^T \sigma_i^2}}.$$

A proof of Corollary 1 is given in the Appendix.

Given the expression for the likelihood derived in Theorem 1, there are multiple ways to carry out model inference. For example, one could sweep over the parameter space to maximize the likelihood of the observed probability paths under the model. Here we instead apply a Bayesian approach, putting a weakly informative prior on θ, and then approximating its posterior distribution via Hamiltonian Monte Carlo (HMC), as implemented in `Stan` (Carpenter et al., 2017).[1] Without further constraints, θ is not fully identified by the data, since multiplying all of the latent variables in Eq. (1) by a positive constant does not affect the sign of the relevant expression. Thus, in our applications below, we constrain the scale of the latent variables by requiring $\text{Var}(Z_1) = 1$.

From a fitted GLIM model, it is straightforward to draw a probability path from the posterior distribution over paths. We do so in three steps. First, we draw a value of $\hat{\theta}$ from the inferred posterior. Next, we draw the vector of latent variables $(z_1, \ldots, z_T)$ from the multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}(X, \hat{\theta}))$. Finally, we compute values of $y_1, \ldots, y_T$ according to Eq. (1). As shown in the Appendix, the value of Eq. (1) can be computed analytically, yielding:

$$y_t = \Phi\left(\frac{\gamma + \sum_{i=1}^{t-1} z_i + z_t + \sum_i \mu_{(i)}^t}{\sqrt{\sum_{i,\,j} \mathbf{\Sigma}_{(i,j)}^t}}\right) \qquad (3)$$

for $1 \le t < T$, where γ, μ^t, and $\mathbf{\Sigma}^t$ are defined as in Theorem 1. For $t = T$, we have $Y_T = 1$ if and only if $\gamma + \sum_{i=1}^T z_i \ge 0$.

4 Experiments

We now explore the efficacy of GLIM through several experiments on synthetic and real-world datasets. In all of our experiments, we use a covariance matrix $\mathbf{\Sigma}(X, \theta)$ with autoregressive structure and heteroskedastic variance. Specifically, we set

$$\mathbf{\Sigma}_{(i,j)} = \sigma_i \sigma_j \rho^{(n-|i-j|)},$$

where

$$\sigma_t^2 = \exp(\beta^T X(t-1)).$$

In this setup, the covariance matrix $\mathbf{\Sigma}$ is parameterized by ρ and β. The first parameter, ρ, controls the correlation between latent variables, with $\rho = 0$ corresponding to independence. The second parameter, β, controls how the latent information evolves over time. We put a weakly informative $\mathcal{N}(0, I)$ prior on β (where I is the identity matrix). On ρ we similarly put a weakly informative prior while constraining the parameter to lie in the interval $[-1, 1]$. Specifically, we set $\rho = \tanh(r)$, where $r \sim \mathcal{N}(0, 1)$.

As we demonstrate, this relatively simple structure often works well in practice. However, other parameterization are also possible, and, in particular, deep kernel learning (Wilson et al., 2016) could be applied to facilitate more complex structure.

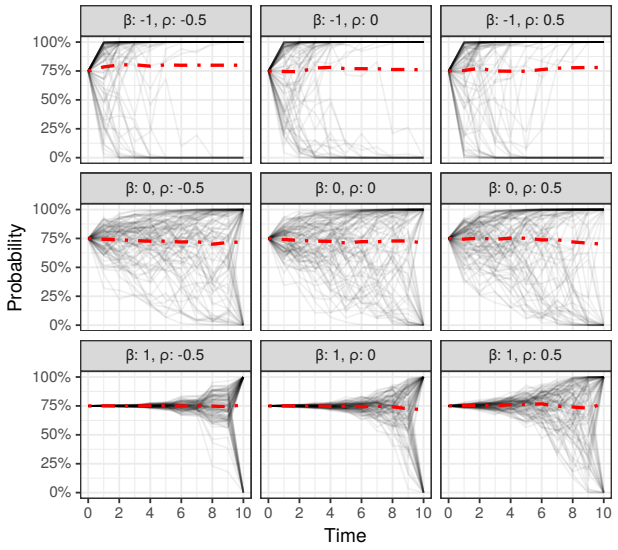
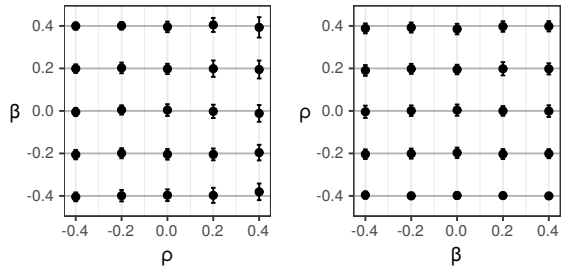


Figure 3: The distribution of probability paths for different covariance structures of the latent variables, parameterized by β and ρ. The grey lines show 100 sample probability paths, and the red lines show the path sample means over time.

4.1 Synthetic data

We start by examining the expressiveness of GLIM by generating probability paths under the model with different parameter settings. In particular, we consider

[1] Our code is available online, though we have not provided a link in order to preserve anonymity.



(a) Fitted β (b) Fitted ρ

Figure 4: The distribution of inferred parameters across synthetic datasets, for different choices of ρ and β. Each point shows the mean inferred parameter across 50 datasets, as well as the standard deviation of estimates. Figure 4a shows inferred values for β (vertical axis) for different values of ρ (horizontal axis). Figure 4b shows inferred values for ρ (vertical axis) for different values of β (horizontal axis).

paths of length $T = 10$, with a constant covariate $X = 1$, and initial value $y_0 = 0.75$. Figure 3 shows the path distribution for several different settings of ρ and β. As is visually apparent from the figure, the model can produce paths exhibiting a wide range of structures while maintaining the martingale property. For example, one can generate paths exhibiting variance near the end of the time period ($\beta = 1$), or near the beginning ($\beta = -1$).

Under quite general regularity conditions, Bayesian posterior means yield consistent parameter estimates (Miller, 2018), yet their finite-sample properties are not always as nice. Here, we explore the efficacy of GLIM to recover estimates in a limited data setting.

In our simulated setting, we consider a time horizon of $T = 5$ steps, with probability paths associated with a single binary covariate X. We tested $5 \times 5 = 25$ different pairs of β and ρ values, ranging from -0.4 to 0.4. For each (β, ρ) pair, we generated 50 synthetic datasets, with each dataset comprised of 500 probability paths, and half of of the paths having $X = 0$ and the other having $X = 1$. On each synthetic dataset, we fit a GLIM model to compute the posterior means $\hat{\beta}$ and $\hat{\rho}$ of the model parameters.

We plot the results of this exercise in Figure 4. For each parameter choice, we plot the mean and standard deviation of $\hat{\beta}$ and $\hat{\rho}$ across the 50 synthetic datasets. For all choices of β and ρ, the posterior means are tightly clustered around the true values, indicating our inference is generally working well, even with short paths and relatively small datasets. Estimates are somewhat more dispersed when the correlation across latent variables is higher (e.g., when $\rho = 0.4$), ostensibly because we shrink the effective number of sample points in this case, creating a more challenging inference problem. Nevertheless, these results suggest GLIM is often able to effectively recover model parameters.

4.2 Real-world data

Experiment setup

Finally, we apply GLIM to model two real-world prediction problems with evolving probability paths: (1) the minute-by-minute win probability of professional basketball teams, updated as the game unfolds; and (2) forecasts of rainfall in Australian cities, updated daily as new information becomes available.

In both prediction tasks, we compare GLIM against two baselines: (1) A set of linear regression (*LR*) models $\{m_t\}$ that predict the future estimated probability at time t; and (2) a Bayesian Seq2Seq LSTM (*BLSTM*) model. We evaluate GLIM's performance against these two baselines in terms of *realized volatility* and *calibration* of each model's predicted paths, described below.

Volatility is a commonly used metric in finance to quantify the level of variation in prices over a given period of time (Bollerslev and Mikkelsen, 1996; Koopman et al., 2005). In our setting, better models should be able to produce paths having similar variance to those that are actually observed in the data. We specifically use realized volatility as described in Koopman et al. (2005), and measure the mean squared error (MSE) between the realized volatilities of the paths observed in the data and the predicted paths produced by each model. We follow Koopman et al. (2005) to compute realized volatility at time t: first, we compute the squared difference in probability between each consecutive time step; then we sum these squared differences for the five consecutive time steps starting at t. Given a collection of posterior path samples, we estimate the volatility at time t for each path as above, and then compute the squared difference between the realized volatility and the average volatility of the sample paths. Finally, volatility MSE at t is the average of these squared differences, averaged over all instances in our dataset. The exact equations for calculating the realized volatility and the corresponding volatility MSE are provided in the Appendix.

We further evaluate model performance in terms of calibration. For each time t, with $0 < t \leq T$, we first compute a model's average prediction at time t for each event (e.g., for each basketball game). We then compute the squared difference between this average and the initial path probability y_0. Finally, we average that error over all events, yielding the calibration MSE at t. Because GLIM is designed to satisfy the martingale

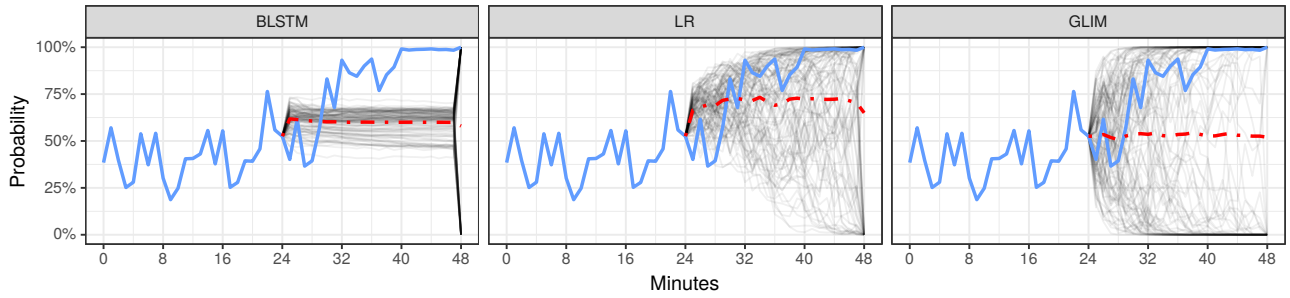
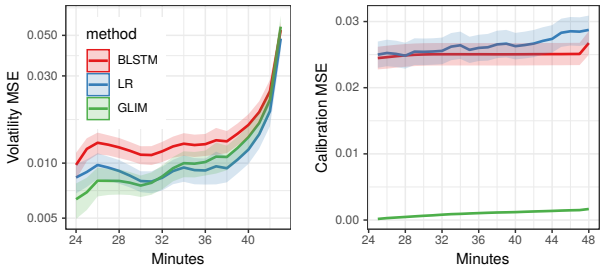


Figure 5: A comparison of real and predicted probability paths for a single basketball game, for GLIM and for our two baseline models (BLSTM and LR). The blue line corresponds to the real path, the grey lines show 100 samples drawn from the distribution of paths predicted by each model, and the red lines show the mean predictions over time.



(a) Volatility MSE (b) Calibration MSE

Figure 6: Model performance for basketball games, as measured by volatility MSE and calibration MSE. The vertical axis for volatility MSE is on the log scale.

property, it is theoretically guaranteed to have perfect calibration. However, the general time-series baselines we compare against offer no such guarantees.

Basketball predictions

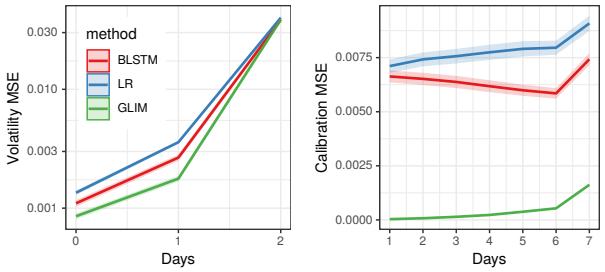
Suppose the second half of a basketball game is about to start. Based on all the available information, the probability that the home team wins is 60%. How is the game likely to evolve? Do we expect a game-deciding event to occur in the next few minutes? Can we safely run to the bathroom and not miss the action? More specifically, do we expect the 60% prediction to remain stable until the final minutes of the game, or, alternatively, do we expect it to quickly veer toward 0 or 100%?

Here we analyze regular-season NBA basketball games for the 2017–2018 and 2018–2019 seasons, training our model on the first season, and evaluating our predictions on the second.[2] To do so, we first created minute-by-minute probability paths for every 48-minute game.

[2]Our basketball data were collected from `www.nba.com` with a third-party API client (Patel, 2018).

For every minute t (for $t \in \{0, \cdots, 47\}$), we trained a separate random forest model to predict the final game outcome based on information available at that point, including: the current home-team and away-team score, the maximum score margin up until that point in the game, and the win rates for the home and away teams for the previous season. On the resulting probability paths, we fit our GLIM model—as well as our two baseline methods—to model predictions for the second half of the game. All three methods use the same information available at half-time, including the score, each team's performance in the previous season, and summary statistics for the probability path in the first half of the game (e.g., the minimum, maximum, average, and standard deviation of predictions).

In Figure 5, we show an illustrative example of a single game, modeled by all three methods. The identical blue curve in all three panels is the ground-truth probability path of the game. Starting at $t = 25$ (i.e., at the start of the second half), each panel displays the distribution of probability paths predicted by each method, with the dashed red line indicating the mean prediction under each model over time. In this example, BLSTM (shown in the left panel) produces probability paths that have far too little volatility until the very end of the game. Such conservative behavior is potentially due to its internal strong Gaussian prior as described in Gal and Ghahramani (2015). In contrast, LR (shown in the center panel) produces paths that appear to appropriately fan out over time. Those paths, however, skew towards the positive outcome, failing to satisfy the martingale property, as indicated by the red line. Finally, GLIM (shown in the right panel) generates predicted paths that are both calibrated (as is theoretically guaranteed) and that appear to appropriately capture the path volatility. In particular, the predicted paths concentrate around 0% and 100% at minute $t = 32$, inline with the actual probability path in this case.



(a) Volatility MSE (b) Calibration MSE

Figure 7: Model performance for rain forecasts, with volatility MSE plotted on the log scale.

Moving beyond this single example, Figure 6 shows the volatility MSE and calibration MSE computed over our complete test set for each model. As suggested by the example above, GLIM has lower volatility MSE than the BLSTM model, and performs comparably with the LR model throughout the prediction period. Further, while GLIM has nearly perfect estimated calibration, both BLSTM and LR have non-negligible calibration MSE.[3]

Weather predictions

Finally, we consider the problem of modeling dynamic forecasts of precipitation. Whereas basketball may ostensibly be well-modeled as a biased random walk, weather outcomes arguably have much more complex temporal dynamics. We specifically use a dataset of Australian rainfall observations (Williams, 2011; Young and Young, 2018), and construct daily predictions starting seven days in advance of the target date. In this case, our predictions were based on a variety of meteorological features, including air pressure, wind speed, location, and month of year. We randomly sampled 10,000 target dates in the dataset prior to 2014 for training our models, and randomly sampled 10,000 target dates in or after 2014 for testing. GLIM and our baseline models were then fit to the training paths, using the same features as above.

We show the results of our evaluation in Figure 7. In contrast to the basketball predictions, BLSTM outperforms LR in this case, having lower volatility MSE and lower calibration MSE. GLIM, however, outperforms both BLSTM and LR on both of these metrics.

[3]The estimated calibration MSE for GLIM is not identically zero because our estimate is based on a finite number of sample paths.

5 Discussion

In this paper, we formally investigated the structure of probabilistic predictions that evolve over time. In doing so, we introduced the Gaussian latent information model (GLIM), a Bayesian framework to model these probability paths. In contrast to many general-purpose methods for time-series analysis, GLIM can naturally incorporate covariates, is able to produce multi-step predictions without explicitly modeling changes in covariates over time, and, perhaps most importantly, preserves the martingale structure of probability paths. We investigated GLIM's efficacy on both synthetic and real-world datasets, helping us understand the dynamic structure of predictions over time.

Aside from being an intriguing problem in and of itself, understanding the structure of evolving probability paths can aid time-sensitive decision makers choose an appropriate action. Examples range from helping ordinary individuals plan weather-contingent events, to physicians treating patients with rapidly changing symptoms (Wassenaar et al., 2015), to prosecutors making judgements based on updating information (Lin et al., 2019).

In theory, many such decision problems can be formulated as a Markov decision process (MDP), and accordingly can be solved via approximate dynamic programming or reinforcement learning algorithms. However, in order to do so, one typically needs the decision maker either to explicitly specify their utility function, or to provide enough data on past decisions to enable inverse reinforcement learning (Arora and Doshi, 2018). Additionally, for many high-stakes decisions in the real world, people are often reluctant to delegate their decisions to an automated agent (Pomerol, 1997). In contrast, by directly estimating and displaying the probability paths themselves, it is possible to disentangle the decision from the information on which it is based. For example, one could imagine a weather forecaster presenting the types of plots we have produced, aiding individual decisions without directly prescribing a specific course of action—or even needing to know any individual's decision problem or utility function.

With the increasing availability of real-time data across domains, rapidly evolving forecasts are also likely to become more common. GLIM offers one promising route for explicitly modeling—and, in turn, predicting—the trajectory of forecasts. Looking forward, we hope our work sparks further interest in uncovering and leveraging the subtle structure of such dynamic predictions.

References

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.

Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20:74, 2019.

Anil K Bera, Matthew L Higgins, and Sangkyu Lee. Interaction between autocorrelation and conditional heteroscedasticity: A random-coefficient approach. *Journal of Business & Economic Statistics*, 10(2): 133–142, 1992.

Nick H Bingham and Goran Peskir. Optimal stopping and dynamic programming. *Encyclopedia of Quantitative Risk Analysis and Assessment*, 1:1236–1243, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3): 307–327, 1986.

Tim Bollerslev and Hans Ole Mikkelsen. Modeling and pricing long memory in stock market volatility. *Journal of econometrics*, 73(1):151–184, 1996.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Yi-Chung Cheng and Sheng-Tun Li. Fuzzy time series forecasting with a probabilistic smoothing hidden markov model. *IEEE Transactions on Fuzzy Systems*, 20(2):291–304, 2011.

Yuan Shih Chow and Herbert Robbins. On optimal stopping rules. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(1):33–49, 1963.

AJW de De Wit and CA Van Diepen. Crop model data assimilation with the ensemble kalman filter for improving regional crop yield forecasts. *Agricultural and Forest Meteorology*, 146(1-2):38–56, 2007.

Francis X Diebold, Joon-Haeng Lee, and Gretchen C Weinbach. Regime switching with time-varying transition probabilities. *Advanced Texts in Econometrics*, 1993.

Roxana Dumitrescu, Marie-Claire Quenez, and Agnès Sulem. A weak dynamic programming principle for combined optimal stopping/stochastic control with ε^f-expectations. *SIAM Journal on Control and Optimization*, 54(4):2090–2115, 2016.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

Robert F Engle and Andrew J Patton. What good is a volatility model? In *Forecasting volatility in the financial markets*, pages 47–63. Elsevier, 2007.

Robert S Erikson and Christopher Wlezien. Markets vs. polls as election predictors: An historical assessment. *Electoral Studies*, 31(3):532–539, 2012.

João Trevizoli Esteves, Glauco de Souza Rolim, and Antonio Sergio Ferraudo. Rainfall prediction methodology with binary multilayer perceptron neural networks. *Climate Dynamics*, 52(3-4):2319–2331, 2019.

T Ferguson. Optimal stopping and applications. mathematics department ucla, 2004.

Christian Francq and Le Quyen Thieu. Qml inference for volatility models with covariates. *Econometric Theory*, 35(1):37–72, 2019. doi: 10.1017/S0266466617000512.

Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

Andrew C Harvey and Neil Shephard. 10 structural time series models. *Handbook of Statistics*, 1993.

Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using hidden markov model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 192–196. IEEE, 2005.

David C Heath and Peter L Jackson. Modeling the evolution of demand forecasts ith application to safety stock analysis in production/distribution systems. *IIE transactions*, 26(3):17–30, 1994.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

SD 1 Jacka. Optimal stopping and the american put. *Mathematical Finance*, 1(2):1–14, 1991.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pages 2575–2583, 2015.

Yuriy Kitsul and Jonathan H Wright. The economics of options-implied inflation probability density functions. *Journal of Financial Economics*, 110(3):696–711, 2013.

Siem Jan Koopman, Borus Jungbacker, and Eugenie Hol. Forecasting daily variability of the s&p 100 stock index using historical, realised and implied volatility measurements. *Journal of Empirical Finance*, 12(3): 445–475, 2005.

Zhiyuan Lin, Alex Chohlas-Wood, and Sharad Goel. Guiding prosecutorial decisions with an interpretable statistical model. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 469–476, 2019.

Klas Malmberg, Lars Rydén, Anders Hamsten, Johan Herlitz, Anders Waldenström, and Hans Wedel. Mortality prediction in diabetic patients with myocardial infarction: experiences from the digami study. *Cardiovascular research*, 34(1):248–253, 1997.

J Martinez-Alario, ID Tuesta, E Plasencia, M Santana, and ML Mora. Mortality prediction in cardiac surgery patients: comparative performance of parsonnet and general severity systems. *Circulation*, 99 (18):2378–2382, 1999.

Jeffrey W Miller. A detailed treatment of doob's theorem. *arXiv preprint arXiv:1801.03122*, 2018.

Mohsen Nasseri, Ali Moeini, and Massoud Tabesh. Forecasting monthly urban water demand using extended kalman filter and genetic programming. *Expert Systems with Applications*, 38(6):7387–7395, 2011.

Wooram Park, Yunfeng Wang, and Gregory S Chirikjian. The path-of-probability algorithm for steering and feedback control of flexible needles. *The International journal of robotics research*, 29(7):813–830, 2010.

Swar Patel. An api client package to access the apis for nba.com. `https://github.com/swar/nba_api`, 2018. Accessed: 2020-10-1.

Jean-Charles Pomerol. Artificial intelligence and human decision making. *European Journal of Operational Research*, 99(1):3–25, 1997.

Houshmand Shirani-Mehr, David Rothschild, Sharad Goel, and Andrew Gelman. Disentangling bias and variance in election polls. *Journal of the American Statistical Association*, 113(522):607–614, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

Daniel Teres, Stanley Lemeshow, JILL SPITZ Avrunin, and HARRIS Pastides. Validation of the mortality prediction model for icu patients. *Critical care medicine*, 15(3):208–213, 1987.

John N Tsitsiklis and Benjamin Van Roy. Optimal stopping of markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44(10):1840–1851, 1999.

A Wassenaar, MHWA van den Boogaard, Theo van Achterberg, AJC Slooter, MA Kuiper, ME Hoogendoorn, KS Simons, E Maseda, N Pinto, C Jones, et al. Multinational development and validation of an early prediction model for delirium in icu patients. *Intensive care medicine*, 41(6):1048–1056, 2015.

Greg Welch, Gary Bishop, et al. An introduction to the kalman filter, 1995.

Graham Williams. *Data mining with Rattle and R: The art of excavating data for knowledge discovery*. Springer Science & Business Media, 2011.

Andrew G Wilson and Zoubin Ghahramani. Copula processes. In *Advances in Neural Information Processing Systems*, pages 2460–2468, 2010.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378, 2016.

Li Yan, Hai-Tao Zhang, Jorge Goncalves, Yang Xiao, Maolin Wang, Yuqi Guo, Chuan Sun, Xiuchuan Tang, Liang Jing, Mingyang Zhang, et al. An interpretable mortality prediction model for covid-19 patients. *Nature Machine Intelligence*, pages 1–6, 2020.

Joe Young and Adam Young. Kaggle: Rain in australia, 2018. `https://www.kaggle.com/jsphyg/weather-dataset-rattle-package`.

Probability Paths and the Structure of Predictions over Time: Supplementary Information

1 Proofs

1.1 Proof of Theorem 1

Proof. Initially, by definition of Y_0, the observed y_0 can be expressed as

$$y_0 = \Phi\left(\frac{\gamma + \mathbb{E}[\sum_{i=1}^{T} Z_i]}{\sqrt{\mathrm{Var}(\sum_{i=1}^{T} Z_i)}}\right) = \Phi\left(\frac{\gamma}{\sqrt{\sum_{i,j} \mathbf{\Sigma}_{(i,j)}}}\right)$$

and γ can be uniquely identified as

$$\gamma = \Phi^{-1}(y_0)\sqrt{\sum_{i,j} \mathbf{\Sigma}_{(i,j)}}.$$

Similarly, definition of Y_t gives

$$Y_t = 1 - \Phi\left(\frac{-\gamma - \sum_{i=1}^{t-1} Z_i - Z_t - \bar{\mu}_t}{\bar{\sigma}_t}\right).$$

By symmetry, we have

$$Y_t = \Phi\left(\frac{\gamma + \sum_{i=1}^{t-1} Z_i + Z_t + \bar{\mu}_t}{\bar{\sigma}_t}\right)$$

$$\Phi^{-1}(Y_t) = \frac{\gamma + \sum_{i=1}^{t-1} Z_i + Z_t + \bar{\mu}_t}{\bar{\sigma}_t}, \tag{1}$$

where $\Phi(\cdot)$ is the standard normal cumulative density function (CDF); $\bar{\mu}_t$ and $\bar{\sigma}_t$ are respectively the mean and standard deviation of the Gaussian variable $\sum_{i=t+1}^{T} Z_i$ conditioned on previous latent information variable values.

Now we show how to obtain the defining parameters $\bar{\mu}_t$ and $\bar{\sigma}_t$ for the conditional distribution of $(\sum_{i=t+1}^{T} Z_i \mid Z_1 = z_1, \ldots, Z_t = z_t)$.

At time t, given the realization of the latent information variables $z_1, \ldots, z_t$, the remaining ones will follow a multivariate Gaussian distribution:

$$(Z_{t+1}, \ldots, Z_T \mid Z_1 = z_1, \ldots, Z_t = z_t) \sim \mathcal{N}(\mu^t, \mathbf{\Sigma}^t) \tag{2}$$

with

$$\mu^t = \mathbf{\Sigma}^t_{21} \left(\mathbf{\Sigma}^t_{11}\right)^{-1} [z_1, \ldots, z_t]^T$$
$$\mathbf{\Sigma}^t = \mathbf{\Sigma}^t_{22} - \mathbf{\Sigma}^t_{21} \left(\mathbf{\Sigma}^t_{11}\right)^{-1} \mathbf{\Sigma}^t_{12}.$$

The terms μ^t and $\mathbf{\Sigma}^t$ are simply the conditional mean and variance of the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ when conditioned on the first t latent information variables.

Then the conditional sum $(\sum_{i=t+1}^{T} Z_i \mid Z_1 = z_1, \ldots, Z_t = z_t)$ will follow a Gaussian distribution $\mathcal{N}(\bar{\mu}_t, \bar{\sigma}_t^2)$. Let $\mathbf{a}^t$ be $\mathbf{1}^T \boldsymbol{\Sigma}_{21}^t \left(\boldsymbol{\Sigma}_{11}^t\right)^{-1}$, we have mean $\bar{\mu}_t$ and variance $\bar{\sigma}_t^2$ being

$$\begin{aligned}
\bar{\mu}_t &= \mathbf{1}^T \mu^t \\
&= \mathbf{1}^T \boldsymbol{\Sigma}_{21}^t \left(\boldsymbol{\Sigma}_{11}^t\right)^{-1} [z_1, \ldots, z_t]^T \\
&= \mathbf{a}^t [z_1, \ldots, z_t]^T = \sum_{i=1}^{t} \mathbf{a}_{(i)}^t z_i \\
\bar{\sigma}_t^2 &= \sum_{i,\, j} \boldsymbol{\Sigma}_{(i,j)}^t,
\end{aligned}$$

where $\boldsymbol{\Sigma}_{(i,j)}^t$ is the element at the i-th row, j-th column in $\boldsymbol{\Sigma}^t$.

Once we have observed y_t and identified $z_1, \ldots, z_{t-1}$, by substituting and rearranging Eq. (1), we can uniquely identify Z_t:

$$Z_t = z_t = \frac{\bar{\sigma}_t \Phi^{-1}(y_t) - \sum_{i=1}^{t-1} (1 + \mathbf{a}_{(i)}^t) z_i - \gamma}{1 + \mathbf{a}_{(t)}^t}.$$

Therefore, given $\boldsymbol{\Sigma}$ and γ, once we have observed $y_1, \ldots, y_t$, we can uniquely identify $z_1, \ldots, z_t$. As a result, $(\Phi^{-1}(Y_t) \mid Y_{t-1} = y_{t-1}, \ldots, Y_1 = y_1)$ is equivalent of $(\Phi^{-1}(Y_t) \mid Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1)$.

Now we are going to find the conditional distribution for $(\Phi^{-1}(Y_t) \mid Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1)$.

As shown in Eq. (2), when conditioning on the first t latent information variables Zs, the remaining ones follow distribution $\mathcal{N}(\mu^t, \boldsymbol{\Sigma}^t)$. Then when conditioned on $(Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1)$, the mean and variance of the conditional marginal distribution of Z_t are simply the first elements in μ^{t-1} and $\boldsymbol{\Sigma}^{t-1}$, namely:

$$(Z_t \mid Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1) \sim \mathcal{N}(\mu_{(1)}^{t-1}, \boldsymbol{\Sigma}_{(1,1)}^{t-1}).$$

Substituting $\bar{\mu}_t$, $\bar{\sigma}_t^2$, and replacing the conditioned $Z_{t-1}, \ldots, Z_1$ with $z_{t-1}, \ldots, z_1$ in Eq. (1), we can obtain the conditional distribution of $\Phi^{-1}(Y_t)$, which is a linear transformation of $(Z_t \mid Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1)$:

$$(\Phi^{-1}(Y_t) \mid Z_{t-1} = z_{t-1}, \ldots, Z_1 = z_1) \sim \mathcal{N}(\tilde{\mu}_t, \tilde{\sigma}_t^2),$$

with

$$\begin{aligned}
\tilde{\mu}_t &= \frac{\gamma + \sum_{i=1}^{t-1} (1 + \mathbf{a}_{(i)}^t) z_i + (1 + \mathbf{a}_{(t)}^t) \mu_{(1)}^{t-1}}{\bar{\sigma}_t} \\
\tilde{\sigma}_t &= \frac{\sqrt{\boldsymbol{\Sigma}_{(1,1)}^{t-1}} (1 + \mathbf{a}_{(t)}^t)}{\bar{\sigma}_t}.
\end{aligned}$$

Finally, by applying change-of-variable trick, we are able to write out the conditional likelihood $P(Y_t = y_t \mid y_{t-1}, \ldots y_1; \boldsymbol{\Sigma}, \gamma)$ in terms of $\Phi^{-1}(y_t)$ for $0 < t < T$ as

$$\begin{aligned}
&P(Y_t = y_t \mid y_{t-1}, \ldots y_1; \boldsymbol{\Sigma}, \gamma) \\
&\quad = P(Y_t = y_t \mid z_{t-1}, \ldots z_1; \boldsymbol{\Sigma}, \gamma) \\
&\quad = P(\Phi^{-1}(Y_t) = \Phi^{-1}(y_t) \mid z_{t-1}, \ldots z_1; \boldsymbol{\Sigma}, \gamma) \times \left| \frac{\partial \Phi^{-1}(y)}{\partial y} \bigg|_{y = y_t} \right| \\
&\quad = \frac{\varphi(\Phi^{-1}(y_t); \tilde{\mu}_t, \tilde{\sigma}_t^2)}{\varphi(\Phi^{-1}(y_t))},
\end{aligned} \tag{3}$$

where $\varphi(\cdot; \tilde{\mu}_t, \tilde{\sigma}_t^2)$ is the PDF of a normal distribution with mean $\tilde{\mu}_t$ and variance $\tilde{\sigma}_t^2$ and $\varphi(\cdot)$ is the standard normal PDF. When $t = T$, we have $P(Y_T = y_T \mid y_{T-1}, \ldots y_1; \boldsymbol{\Sigma}, \gamma) = P(Y_T = y_T \mid y_{T-1}) = y_{T-1}^{y_T}(1 - y_{T-1})^{(1-y_T)}$ as it follows a Bernoulli distribution with $p = y_{T-1}$ by definition. Multiplying $P(Y_t = y_t \mid y_{t-1}, \ldots y_1; \boldsymbol{\Sigma}, \gamma)$ for all $0 < t \leq T$ gives the likelihood of the path $y_1, \ldots, y_T$. With expansion of normal PDFs, log transformation, and some rearrangement of constants, we have the expression for log PDF of $y_1, \ldots, y_T$ as presented in Theorem 1. □