State Tracking in the Presence of Heavy-tailed Observations

Yaman Kindap[a]
Department of Computer Engineering, Bogazici University, Bebek, Istanbul, Turkey

Keywords: Signal Processing, Bayesian Models, Stochastic Methods, Classification, Clustering.

Abstract: In this paper, we define a state-space model with discrete latent states and a multivariate heavy-tailed observation density for applications in tracking the state of a system with observations including extreme deviations from the median. We use a Gaussian distribution with an unknown variance parameter which has a Gamma distribution prior depending on the state of the system to model the observation density. The key contribution of the paper is the theoretical formulation of such a state-space model which makes use of scale mixtures of Gaussians to yield an exact inference method. We derive the framework for estimation of the states and how to estimate the parameters of the model. We demonstrate the performance of the model on synthetically generated data sets.

1 INTRODUCTION

Modelling sequential data and making inference on these systems is an essential area of research in various fields of science and engineering. Sequential data may arise from dynamical systems in the form of a time series or a single dimensional spatial process where the position of an observation is critical such as in biological sequences. While the evolution of a dynamical system is often described by a deterministic function, some systems are inherently stochastic or the complexity of some systems may require statistical arguments in which case the model of the system evolution is stochastic.

While randomness caused by measurement noise is involved in most systems of interest, the evolution of the dynamical system may be stationary which simply means that some statistical properties of the system are constant with respect to time. Such systems can be modelled by evolution functions that are independent of time. On the other hand, non-stationary dynamical systems present the additional problem that the evolution mechanism of the system also changes in time. In such systems, one solution is to cluster the observations in terms of similar characteristic behaviour in order to model the different dynamics of the system separately. One extensively studied model that is used for this problem is the hidden Markov model with a mixture of Gaussian observation likelihoods (GHMM) (Rabiner, 1990). Because of its mathematical simplicity and its ability to estimate any density, the GHMM can be used in various different applications.

However, for systems that display extreme deviations from its median characteristic behavior for each state, the GHMM is limited because of its assumption that observations follow a conditional Gaussian distribution. In this work, we show how to expand the GHMM in a concise mathematical way by introducing an additional latent scale parameter into the GHMM which enables the dynamical modeling of heavy-tailed observations. Algorithms for inference of latent variables and learning the parameters of the proposed model are presented along with their theoretical formulations.

The use of scale mixture of Gaussian distributions for the purpose of modelling heavy-tailed observations is presented in (Cemgil et al., 2007). Recently, the extension of hidden Markov models to include heavy-tailed observations are studied, especially in the case of Student's t-distribution (Chatzis et al., 2009). It has been shown that the Student's t-distribution HMM (SHMM) is able to identify more persistent states in time series data (Bulla, 2011). The use of SHMMs in classification tasks is presented in (Zhang et al., 2013).

Hidden Markov models used in the literature with t-distribution as the emission distribution (SHMM) use the well-known derivation of t-distribution as a scale mixture of the Gaussian distribution where the variance of the Gaussian distribution is treated as a

[a] https://orcid.org/0000-0002-9269-039X

Kindap, Y.
State Tracking in the Presence of Heavy-tailed Observations.
DOI: 10.5220/0010150601350142
In *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, pages 135-142
ISBN: 978-989-758-486-2

random variable. While this formulation allows the expectation-maximization algorithm to be tractable, the main simplicity of its derivation comes from the fact that the resulting distribution has an exponential family form, which there is no reference to in the HMM literature. In this work we explicitly show this derivation. Furthermore, rather than introducing the variance of the Gaussian distribution into the model as a latent variable, previous works only use this formulation for mathematical convenience. In this work, we will treat the variance of the Gaussian distribution as an additional latent variable in our model which allows us to have a dynamically scaled model at each time step.

2 GAUSSIAN-GAMMA HIDDEN MARKOV MODEL

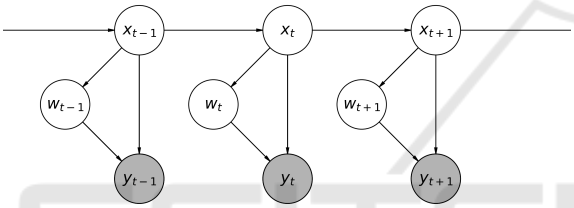

Figure 1: Probabilistic graphical model of GGHMM.

The Gaussian-Gamma hidden Markov model (GGHMM) is a particular class of the state-space model shown in Figure 1 where the variables $x_{1:T}$ are discrete-valued while $w_{1:T}$ and $y_{1:T}$ are continuous-valued. Additionally, observed variables are shaded for emphasis. The model assumes that an unobserved first-order Markov process generates observations from a mixture distribution $p(y_t|x_t,w_t)$. Hence, the HMM can be interpreted as an extension of mixture models where sequential information is also encoded (Bishop, 2006).

Let's assume that we have a N-state system. The state at time t, defined as x_t, has an evolution process which is assumed to be a discrete state, discrete time first order Markov process. Thus, we can represent this process as a categorical distribution where the parameters of the distribution depend on the value assumed by the previous state x_{t-1}.

$$f(x_t|x_{t-1},A) = \prod_{i=1}^{N}\prod_{j=1}^{N} a_{ij}^{[x_{t-1}=i][x_t=j]} \tag{1}$$

where a_{ij} is the i^{th} row and j^{th} column of a transition matrix A. The notation $[P]$ is the Iverson bracket where $[P]$ equals 1 if P is true and 0 otherwise. The initial probabilities of states at time 1 is parameterized as a vector π, where the density is defined as $p(x_1|\pi)$.

An observation at time t, defined as y_t, is assumed to be conditionally independent of previous observations $y_{1:t-1}$ given the current state x_t and has a k-dimensional Gaussian distribution with mean μ and an unknown variance. We treat the variance of the observation process as a random variable and therefore define a scaling random variable w_t that has a Gamma distribution with parameters $\alpha = \beta = \frac{\nu}{2}$. Similar to the state x_t of the system, the scaling variable w_t is a latent variable. Furthermore, we assume that the value of parameters of our model depends on the current state of the system. The corresponding model can be shown as:

$$\begin{aligned} w_t|x_t,\nu &\sim \text{Gamma}\left(\frac{\nu}{2},\frac{\nu}{2}\right) \\ &= \frac{\frac{\nu}{2}^{(\frac{\nu}{2})}}{\Gamma(\frac{\nu}{2})} w_t^{\frac{\nu}{2}-1}\exp\left(-\left(\frac{\nu}{2}\right)w_t\right) \end{aligned} \tag{2}$$

$$d(y_t,\mu) = (y_t-\mu)^T\Sigma^{-1}(y_t-\mu) \tag{3}$$

$$\begin{aligned} y_t|x_t,w_t,\mu,\Sigma &\sim \text{Gaussian}\left(\mu,\frac{\Sigma}{w_t}\right) \\ &= \frac{\sqrt{w_t^k}}{\sqrt{(2\pi)^k|\Sigma|}}\exp\left(-\frac{1}{2}w_t d(y_t,\mu)\right) \end{aligned} \tag{4}$$

where Σ is considered to be a constant covariance parameter of the observation density encoding the correlations between different observations and $d(y_t,\mu)$ is a distance measure. The scale of the covariance parameter is adjusted according to w_t. When $x_t = i$ the state dependent parameters are shown with the subscript i as μ_i, Σ_i and ν_i. This parameterization is equivalent to setting a conjugate prior on the variance of the Gaussian distribution. Such a formulation of the observation variable y_t has the convenient property that when we integrate over the latent scale variable w_t of the Gaussian distribution it becomes a generalized multivariate t-distribution which has a density shown in equation 5 where C is the normalization constant shown in equation 6. Furthermore, the Gaussian-Gamma distribution density define an exponential family which enable efficient inference.

$$p(y_t|x_t,\mu,\Sigma,\nu) = C\left(1+\frac{1}{\nu}d(y_t,\mu)\right)^{-\frac{\nu+k}{2}} \tag{5}$$

$$C = \frac{\Gamma(\frac{\nu+k}{2})}{\Gamma(\frac{\nu}{2})\nu^{(k/2)}\pi^{(k/2)}|\Sigma|^{(1/2)}} \tag{6}$$

Using the model assumptions we have defined, the joint probability distribution of the variables $y_{1:T}$, $w_{1:T}$ and $x_{1:T}$ is defined as:

$$p(y_{1:T}, w_{1:T}, x_{1:T}|\theta) = p(x_1|\pi) \prod_{t=2}^{T} p(x_t|x_{t-1}, A) \prod_{t=1}^{T} p(y_t|w_t, x_t, \mu, \Sigma) p(w_t|x_t, \nu) \quad (7)$$

where θ is the set of parameters of the model.

3 LATENT VARIABLE INFERENCE

Our main goal is to infer the posterior probability distribution of the state of the system at each time t. Assuming that our model sufficiently represents the system and the true parameters of the model are known, the posterior probability of x_t can be calculated without knowing the posterior distribution of w_t since we will be marginalizing it out. However, for a complete understanding of the model and it's requirement in parameter estimation, we will also be concerned with the inference of the posterior distribution of w_t.

We formulate the filtering and smoothing problems for the estimation of latent variables in our model using a recursive Bayesian framework. In the GGHMM the filtered density includes w_t and shown as $p(x_t, w_t|y_{1:t}, \theta)$. Similarly the smoothed density is defined as $p(x_t, w_t|y_{1:T}, \theta)$.

3.1 Filtering

Let's assume that the posterior distribution of the state x_{t-1} at time $t-1$ is known. Using the state evolution equation $f(x_t|x_{t-1}, A)$ we can obtain a prior distribution on the state x_t.

$$p(x_t|y_{1:t-1}, \theta) = \sum_{x_{t-1}} f(x_t|x_{t-1}, A) p(x_{t-1}|y_{1:t-1}, \theta) \quad (8)$$

The joint posterior distribution on the latent variables x_t and w_t is shown in equation 9.

$$p(x_t, w_t|y_{1:t}, \theta) = \frac{1}{p(y_t|y_{1:t-1}, \theta)} * p(y_t|w_t, x_t, \mu, \Sigma) p(w_t|x_t, \nu) p(x_t|y_{1:t-1}, \theta) \quad (9)$$

Notice that the denominator $p(y_t|y_{1:t-1}, \theta)$ is obtained by marginalizing x_t and w_t out from the nominator in equation 9. At this point, we can obtain the posterior distribution of x_t by using the convenient property of integrating out w_t in order to obtain the density of t-distribution.

$$p(x_t|y_{1:t}, \theta) = \frac{p(y_t|x_t, \mu, \Sigma, \nu) p(x_t|y_{1:t-1}, \theta)}{p(y_t|y_{1:t-1}, \theta)} \quad (10)$$

where the density $p(y_t|x_t, \mu, \Sigma, \nu)$ is the generalized t-distribution shown in equation 5. Notice that using the chain rule, the posterior distribution of the latent variables can be factorized as shown in equation 11.

$$p(x_t, w_t|y_{1:t}, \theta) = p(x_t|y_{1:t}, \theta) p(w_t|x_t, y_{1:t}, \theta) \quad (11)$$

Using the factorization shown in equation 11, the filtered posterior distribution of w_t can be obtained using equation 12.

$$p(w_t|x_t, y_{1:t}, \theta) = \frac{1}{p(x_t, y_t|y_{1:t-1}, \theta)} * p(y_t|w_t, x_t, \mu, \Sigma) p(w_t|x_t, \nu) p(x_t|y_{1:t-1}, \theta) \quad (12)$$

where the joint density $p(x_t, y_t|y_{1:t-1}, \theta)$ is equivalent to the nominator of equation 10. With some rearrangement of parameters, the joint density $p(w_t|x_t, y_{1:t}, \theta)$ actually defines an exponential family in equation 12. This property enables us to have some very efficient algorithms.

$$p(w_t|x_t, y_{1:t}, \theta) = \frac{w_t^{\frac{\nu_i - 2 + k}{2}} \exp\left(-\frac{1}{2} d(y_t, \mu) w_t \right)}{\Gamma(\frac{\nu_i + k}{2}) \left(\frac{1}{2} d(y_t, \mu) + \frac{\nu_i}{2} \right)^{-\frac{\nu_i + k}{2}}} \quad (13)$$

3.1.1 Moments of the Latent Scaling Variable

Since we will be needing to calculate some functions of w_t in the parameter learning section, let's analyze the posterior distribution of w_t shown in equation 13. Firstly, notice that the only information we have about w_t is through the observation of y_t and the estimate of x_t. Any observation other than y_t does not have any influence over our estimates of w_t. While we expect that consecutive w_t values do not vary significantly in real data sets, the GGHMM model does not take this dependence into account in order to simplify the mathematical formulation of the model. This assumption mathematically means that we cannot improve our estimate of the posterior distribution by using a

smoothing procedure and $p(w_t|x_t = i, y_{1:t}, \theta)$ is equal to $p(w_t|x_t = i, y_{1:T}, \theta)$. We will comment on this limitation of our model in section 6.

Let's now calculate the moments of the sufficient statistics of w_t defined by equation 13. For exponential families, we define the functions $t(x)$, $\eta(\theta)$ and $a(\eta)$. We have 2-dimensional sufficient statistics shown as:

$$\begin{aligned} t_1(w_t) &= -w_t \\ t_2(w_t) &= \log(w_t) \end{aligned} \tag{14}$$

The parameters $\eta(\theta)$ corresponding to these sufficient statistics are:

$$\begin{aligned} \eta_1(\theta) &= \Big[\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big] \\ \eta_2(\theta) &= \Big(\frac{\nu_i - 2 + k}{2}\Big) \end{aligned} \tag{15}$$

Most importantly the log-normalizer function $a(\eta)$ is defined as:

$$\begin{aligned} a(\eta) &= \log\left[\Big(\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big)^{-(\frac{\nu_i+k}{2})}\Gamma(\frac{\nu_i+k}{2})\right] \\ &= \log\left[\eta_1^{-(\eta_2+1)}\Gamma(\eta_2+1)\right] \end{aligned} \tag{16}$$

Now, using the log-normalizer function, we can calculate the first moments of the sufficient statistics $-w_t$ and $\log(w_t)$.

$$\begin{aligned} \frac{\partial}{\partial\eta_1}a(\eta) &= \frac{\Gamma(\eta_2+1)(-1)(\eta_2+1)\eta_1^{-(\eta_2+2)}}{\eta_1^{-(\eta_2+1)}\Gamma(\eta_2+1)} \\ &= (-1)(\eta_2+1)\eta_1^{-1} \\ &= E(-w_t) \end{aligned} \tag{17}$$

We can transform the parameters η_1 and η_2 into their θ forms in order to obtain an expression for the expectation of w_t.

$$\begin{aligned} \frac{\partial}{\partial\eta_1}a(\eta) &= (-1)\Big(\frac{\nu_i+k}{2}\Big)\Big[\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big]^{-1} \\ &= E(-w_t) \end{aligned} \tag{18}$$

Since the expectation operator scales linearly when multiplied by a constant factor, we find that:

$$E(w_t) = \Big(\frac{\nu_i+k}{2}\Big)\Big[\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big]^{-1} \tag{19}$$

Similarly, let's find the first moment of $\log(w_t)$:

$$\frac{\partial}{\partial\eta_2}a(\eta) = \psi_0(\eta_1) - \ln(\eta_1) \tag{20}$$

where the function $\psi_0(z)$ is the digamma function which describes the derivative of the Gamma function $\Gamma(z)$ through the identity:

$$\Gamma'(z) = \Gamma(z)\psi_0(z) \tag{21}$$

We can transform the parameters η_1 and η_2 into their θ forms in order to obtain an expression for the expectation of $\log(w_t)$.

$$\begin{aligned} \frac{\partial}{\partial\eta_2}a(\eta) &= \psi_0(\eta_1) - \ln(\eta_1) \\ &= \psi_0\Big(\frac{\nu_i+k}{2}\Big) - \ln\Big(\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big) \end{aligned} \tag{22}$$

$$E\Big(log(w_t)\Big) = \psi_0\Big(\frac{\nu_i+k}{2}\Big) - \ln\Big(\frac{1}{2}d(y_t,\mu) + \frac{\nu_i}{2}\Big) \tag{23}$$

3.2 Smoothing

In this section, our goal is to obtain a smoothed estimate of the latent variables $x_{1:T}$ which utilizes the full information available at time T. As we have mentioned, the smoothed posterior density of w_t is equal to the filtered posterior since we do not have any sequential dependence between consecutive w_t values. The smoothed estimate we aim to calculate in this section is defined as $p(x_t|y_{1:T}, \theta)$.

Notice that similar to our considerations in an HMM, we can use the filtered estimates $p(x_t = i, y_{1:t}|\theta)$ together with a density $p(y_{t+1:T}|x_t = i, \theta)$ in order to calculate the joint probability density of the model:

$$p(x_t, y_{1:T}|\theta) = p(x_t, y_{1:t}|\theta)p(y_{t+1:T}|x_t, \theta) \tag{24}$$

which can be normalized by using the incomplete data likelihood $p(y_{1:T}|\theta)$ in order to obtain the smoothed estimate $p(x_t|y_{1:T}, \theta)$.

Assuming we know $p(y_{t+1:T}|x_t, \theta)$, we can find a joint distribution $p(y_{t:T}|x_{t-1}, \theta)$ by initially using the emission density $p(y_t|x_t, \mu, \Sigma, \nu)$ of the multivariate t-distribution in order to find the likelihood of $y_{t:T}$ for each state x_t. We call this operation the update step since it involves the addition of an observation y_t to a known joint probability distribution.

$$p(y_{t:T}|x_t, \theta) = p(y_t|x_t, \mu, \Sigma, \nu)p(y_{t+1:T}|x_t, \theta) \tag{25}$$

The next step is to estimate the likelihood of $y_{t:T}$ given that we only know state x_{t-1}. This estimate can be obtained by a marginalization procedure and is called postdiction since it involves the estimation of the likelihood of a state with information from the future.

$$p(y_{t:T}|x_{t-1},\theta) = \sum_{x_t} p(y_{t:T}|x_t,\theta) f(x_t|x_{t-1},A) \quad (26)$$

4 PARAMETER LEARNING

We formulate a maximum likelihood estimation procedure for our model based on the expectation-maximization algorithm. The well known Baum-Welch algorithm is derived for the hidden Markov model which has the same state transition structure as our model. Thus, we build upon this algorithm to derive update equations for the model parameters μ, Σ and ν, and note that π and A are same the same as in the Baum-Welch algorithm (Bilmes, 2000). The maximum likelihood estimation problem can be stated as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \quad \log p(y_{1:T}|\theta) \quad (27)$$

The expectation maximization algorithm is a method of maximizing equation 27 in an iterative manner. Using Jensen's inequality we can find a lower bound to the incomplete data likelihood $p(y_{1:T}|\theta)$ and maximize this lower bound at each iteration. The lower bound turns out to be in the form of an expectation over the posterior distribution of the latent variables $x_{1:T}$ and $w_{1:T}$, $p(x_{1:T}, w_{1:T}|y_{1:T},\theta)$, which is shown in equation 28.

$$\log p(y_{1:T}|\theta) \geq E\left[\log \frac{p(y_{1:T},x_{1:T},w_{1:T}|\theta)}{p(x_{1:T},w_{1:T}|y_{1:T},\theta)}\right] \quad (28)$$

Let's define the estimated parameters at each iteration with a superscript k, shown as $\theta^{(k)}$. For computational efficiency, the posterior distribution over the latent variables are calculated with the set of parameters $\theta^{(k)}$. The details of this derivation is similar to the considerations in the Baum-Welch algorithm which are shown in detail in (Bilmes, 2000). In summary, it is based on the chain rule of probability applied to the posterior distribution of the latent variables.

In order to simplify the update equations we define the posterior probability distribution of x_t, $p(x_t = i|y_{1:T},\theta^{(k)})$, as $\gamma_t^{(k)}(i)$, which is consistent with the literature. The update equations for μ_i and Σ_i are shown below:

$$\mu_i^{(k+1)} = \frac{\sum_{t=1}^{T} \gamma_t^{(k)}(i) \langle w_{t,i} \rangle y_t}{\sum_{t=1}^{T} \gamma_t^{(k)}(i) \langle w_{t,i} \rangle} \quad (29)$$

$$\Sigma_i^{(k+1)} = \frac{\sum_{t=1}^{T} \gamma_t^{(k)}(i) \langle w_{t,i} \rangle (y_t - \mu_i^{(k+1)})(y_t - \mu_i^{(k+1)})^T}{\sum_{t=1}^{T} \gamma_t^{(k)}(i) \langle w_{t,i} \rangle} \quad (30)$$

where we denote the posterior expected value of $w_{t,i}$ as $\langle w_{t,i} \rangle$ for brevity. Notice that we have shown the calculation of this expectation is relatively simple because of the exponential family form of the posterior distribution of $w_{t,i}$ shown in 3.1.1. Therefore, equations 29 and 30 have similar analogs in Baum-Welch update equations, and their calculation is straightforward.

Unfortunately, the update equation for ν_i does not have a simple form similar to equations 29 and 30 because of the presence of the Gamma function $\Gamma(z)$ in the gamma distribution $p(w_t|x_t = i,\nu)$. We have to introduce the digamma function $\psi_0(z)$ which is defined as the logarithmic derivative of the gamma function. Let's define the posterior distribution of w_t, $p(w_t|x_t = i, y_{1:T}, \theta^{(k)})$, as $\kappa_t^{(k)}$. Taking the derivative of the expectation shown in equation 28 with respect to ν_i and setting it equal to zero yields:

$$0 = \log\left(\frac{\nu_i}{2}\right) + 1 - \psi_0\left(\frac{\nu_i}{2}\right) + K \quad (31)$$

where K denotes a constant with respect to ν_i and can be shown as:

$$K = \frac{\sum_{t=1}^{T} \gamma_t^{(k)}(i) \int_0^\infty dw_t \, \kappa_t^{(k)} \left(\log(w_{t,i}) - w_{t,i}\right)}{\sum_{t=1}^{T} \gamma_t^{(k)}(i)} \quad (32)$$

Notice that the nominator of the equation above includes the expected value of $\log(w_t)$ and w_t which we previously calculated in section 3.1.1.

In order to find a value $\nu_i^{(k+1)}$, we propose to find the root of the equation 31 by using the bisection (binary search) method. We prefer this method over other root finding algorithms such as the Newton–Raphson method because it enables us to constrain the space of possible ν_i values to be in $(1,\infty)$. This is important for the robustness of the parameter estimation process since the student's t-distribution is only defined for $\nu > 0$ and its mean is defined for $\nu > 1$. When the mean of the emission density is not defined, all other parameters diverge. An alternative method of approximating the value of $\nu_i^{(k+1)}$ is shown in (Shoham, 2002).

5 EXPERIMENTS

In this section, we test the performance of our proposed Gaussian-Gamma hidden Markov model with synthetically generated data since that is the only case we can know the actual realizations of the latent variables. Problems involving real data in the context where we assume the presence of latent variables generally requires the construction of objective functions which heavily influence the performance of such a model and is out of scope of this work.

We present the theoretical capabilities of our model by evaluating its performance on state identification for synthetically generated multivariate heavy-tailed time series data and compare it to a Gaussian hidden Markov model. The synthetic data generation process of a multivariate heavy-tailed time series involves a state-space model with an identical structure to the GGHMM. The parameters of the model are randomly generated in each realization of the experiment with a 2-dimensional latent Markovian state space. Our aim is to understand the effects of maximum training epochs used for each realization and dimensionality of the observation space on the state identification task using smoothed estimates of the latent states.

5.1 Evaluation Criteria

There are various evaluation criteria for classification and clustering tasks such as accuracy, precision and recall. However, we need to consider the underlying decision making problem in order to select an appropriate metric. While our main goal is to identify the state at each time t, since we assume that states are persistent in a dynamical system, the difficulty of this task arise from state change-points.

Let's discuss the potential effectiveness of each evaluation metric in order to understand their significance. Accuracy, defined in equation **??**, measures the correctness of all state identifications. This measure does not reflect the persistent nature of our dynamical system since correctly guessing that there wasn't any change in the system is not as informative as detecting a change.

$$\text{Accuracy} = \frac{\sum \text{True Positive} + \sum \text{True Negative}}{\sum \text{Total Population}} \quad (33)$$

Instead of focusing on identifying every state, let's direct our attention to how well we can identify a specific state. This change in our focus leads us to the precision and recall metrics which are defined in equations 34 and 35, respectively. Precision is a measure of the correctness of our positively identified states and recall is a measure of how correct we were actually able to identify a specific state in reality. These two metrics are a better measure of the performance of our models considering that states of a system are not equally informative such as in the case of identifying malignant and benign biological activities. Thus, we select the state with higher overall standard deviations as the relevant state to be identified, since we assume that critical states have higher uncertainty. We report the corresponding precision and recall values for each realization of the experiment.

$$\text{Precision} = \frac{\sum \text{True Positive}}{\sum \text{True Positive} + \sum \text{False Positive}} \quad (34)$$

$$\text{Recall} = \frac{\sum \text{True Positive}}{\sum \text{Condition Positive}} \quad (35)$$

Since we do not want our models to be dependent on a specific data set, we generate multiple different realization of the experiments with randomly generated data sets and compare the models based on an equally weighted average performance on these different realizations. The results report the mean and standard deviation in each evaluation criteria.

5.2 Performance Statistics of Experiments

The performance of both models highly depend on the random data generation mechanism. Particularly, having a heavy-tailed distribution only matters in the case where the theoretical probability distribution of each state intersects. This fact is actually quite intuitive when we consider the one-dimensional observation space case. If the two states have mutually exclusive probability distributions it would be easy to distinguish between the two. The problem gets progressively more difficult as their sample space intersect. Thus, we constrain the mean behavior of both states to be close in terms of some dispersion metric. This corresponds to a unique problem where the regimes of the process have very similar average behavior while the covariance structure and the extreme events define their difference. In order to reflect the persistence of each state while still keeping the structure of data relatively random, the diagonal terms in the transition matrix are constrained to be $a_{ii} \geq 0.8$. We explore the performance of each model according to this case and a particular realization of the synthetically generated data is shown in Figure 2 where different states are shaded.

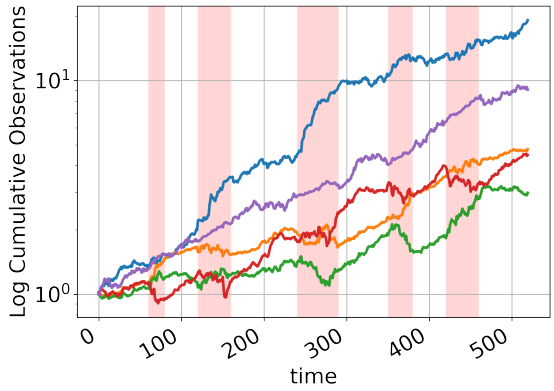

Figure 2: Example realization of experiments.

Table 1: Performance statistics of equally trained models.

	GGHMM		GHMM	
	Mean	Std	Mean	Std
Accuracy	0.849	0.150	0.907	0.145
Precision	0.782	0.126	0.831	0.171
Recall	0.727	0.131	0.911	0.207

Initially, we have recorded the results of 100 different realizations of the experiment where both of the models are trained for a maximum of 20 epochs. The observation space is 3-dimensional and each realization has a length of 500. The results are presented in Table 1. We see that the GHMM has a significantly better performance in terms of mean accuracy and standard deviation in accuracy throughout the realizations. The same result carries over to their performance in terms of precision and recall. Thus, under the current conditions the GHMM is a better choice.

We arguably need to train the GGHMM model with more maximum training epochs in order have comparable performance statistics because of its increased complexity. It is likely that the parameters did not converge to a set which maximizes the likelihood of the incomplete data set in 20 epochs. The convergence properties of the EM algorithm for the case of Gaussian emissions are studied in (Xu and Jordan, 1996). Since the GHMM is relatively simpler when compared to the GGHMM, we conclude that under constrained compute power the GHMM displays better results.

Table 2: Performance statistics of unconstrained training epoch realizations.

	GGHMM		GHMM	
	Mean	Std	Mean	Std
Accuracy	0.957	0.063	0.901	0.139
Precision	0.953	0.099	0.868	0.241
Recall	0.948	0.116	0.896	0.218

Let's investigate the effects of maximum training epochs in the performance statistics of both models. We have recorded the results of 100 different realization of the experiment where the GGHMM is trained until its results converge according to a threshold difference between each iteration and the GHMM is trained for a maximum of 20 epochs. The results are shown in Table 2. Since we left the maximum training epochs for the GHMM unchanged, we expect that the mean and standard deviation in performance measures are not significantly different from the first set of 100 realizations. Results show that we can make a significant improvement for the GGHMM by increasing the maximum training epochs. The GGHMM has a better performance in every metric compared to both the GHMM for these realizations and the GGHMM for the previous 100 realizations. Furthermore, convergence in the GGHMM is reached in 80 epochs on average.

However, considering that a single epoch of training in GGHMM is significantly longer than a single epoch in GHMM because of the increased complexity of the calculations, both models may have their advantages. For most problems, the training time of the model is another important selection metric because of time constraints. While the GGHMM is able to display better performance, it is at the cost of significantly more training time. The assessment of how this fact affects the choice between the two models is dependent on the problem and the priorities of the researcher.

Table 3: Performance statistics in 5-dimensional observation space.

	GGHMM		GHMM	
	Mean	Std	Mean	Std
Accuracy	0.957	0.073	0.867	0.156
Precision	0.941	0.100	0.893	0.179
Recall	0.961	0.057	0.880	0.227

Next, we investigate the effects of the dimensionality of the observation space. Thus, we have recorded the results of 100 different realizations of the experiment with a 2-dimensional latent Markovian state-space, a 5-dimensional observation space and each realization has a length of 500. As we have established in the previous 200 realizations of the experiment, a better performance is recorded for the GGHMM when we let the model train until convergence. Since the average epochs for GGHMM was found to be 80 for convergence, we use this value as an upper bound to constrain the experiment. Therefore, we train the GHMM and GGHMM for a maximum of 20 and 80 training epochs, respectively. The

results are shown in Table 3. Notice that we cannot show any improvement in the performance for the GHMM while the performance of the GGHMM is similarly good compared to the 3 dimensional case. Thus, we conclude that the dimensions of the observation space does not have a significant effect on the performance for these realizations.

Overall, we are able to show that the GGHMM performs better than the GHMM under specific circumstances. While both models have their advantages, there may be some benefit in using a GGHMM for the identification of states of a system with extereme deviations over the GHMM. On the other hand, the increased training time for the GGHMM also needs to be considered when deploying such a model in production.

6 CONCLUSION

In this work, we have introduced an extension to the hidden Markov model in order to identify the states of a non-stationary dynamical system with observations that have heavy-tailed distributions. Such systems pose a significant challenge to researchers in computational fields and even incremental advancements may be highly lucrative.

Our proposed model, the Gaussian-Gamma hidden Markov model, can be considered as a variant of the hidden Markov model where we increase the complexity of the model in order to accommodate our prior knowledge on heavy-tailed distributions. The increased complexity of our model can be efficiently handled by formulating the observation density as an exponential family. Our model can potentially be used to model various non-stationary dynamic systems with multiple regimes and heavy-tailed distributions and is capable of representing heteroscedastic processes within each state and is highly flexible. Furthermore, the model is mostly analytically tractable aside from requiring an auxiliary root finding algorithm. This allows it to be trained relatively quickly compared to the state-of-the-art deep neural networks and requires much less data in order to be trained.

We have shown the application of GGHMM in state identification for synthetically generated data. Results show that the GGHMM has a performance comparable with the GHMM in the state identification task. In terms of improvements to the GGHMM, one obvious way of improvement can come from explicitly modelling the dependencies between the sequential latent scale variables. However, since this would introduce intractable calculations we have left it as a future research.

Another source of improvement may be to also learn the number latent states in the system. Such models use the hierarchical Dirichlet process as their latent representation of the state transition system (Teh et al., 2006). For an unknown number of states, the intuition that the states are persistent can be modelled using the work in (Fox et al., 2007). For more efficient learning in such models, practical considerations are presented in (Ulker et al., 2011).

REFERENCES

Bilmes, J. (2000). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical Report ICSI-TR-97-021, University of Berkeley*, 4.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bulla, J. (2011). Hidden markov models with t components. increased persistence and other aspects. *Quantitative Finance*, 11(3):459–475.

Cemgil, A. T., Fevotte, C., and Godsill, S. J. (2007). Variational and stochastic inference for bayesian source separation. *Digital Signal Processing*, 17(5):891 – 913. Special Issue on Bayesian Source Separation.

Chatzis, S., Kosmopoulos, D., and Varvarigou, T. (2009). Robust sequential data modeling using an outlier tolerant hidden markov model. *IEEE transactions on pattern analysis and machine intelligence*, 31:1657–69.

Fox, E., Sudderth, E., Jordan, M., and Willsky (2007). The sticky hdp-hmm: Bayesian nonparametric hidden markov models with persistent states. Technical Report 2, MIT Laboratory for Information & Decision Systems, Cambridge, MA 02139.

Rabiner, L. R. (1990). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, page 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Shoham, S. (2002). Robust clustering by deterministic agglomeration em of mixtures of multivariate t-distributions. *Pattern Recognition*, 35:1127–1142.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Ulker, Y., Gunsel, B., and Cemgil, A. T. (2011). Annealed smc samplers for nonparametric bayesian mixture models. *IEEE Signal Processing Letters*, 18(1):3–6.

Xu, L. and Jordan, M. I. (1996). On convergence properties of the em algorithm for gaussian mixtures. *Neural Comput.*, 8(1):129–151.

Zhang, H., Jonathan Wu, Q. M., and Nguyen, T. M. (2013). Modified student's t-hidden markov model for pattern recognition and classification. *IET Signal Processing*, 7(3):219–227.