Sparse Bayesian ARX models with flexible noise distributions

Johan Dahlin, Adrian Wills and Brett Ninness*

July 6, 2018

Abstract

This paper considers the problem of estimating linear dynamic system models when the observations are corrupted by random disturbances with nonstandard distributions. The paper is particularly motivated by applications where sensor imperfections involve significant contribution of outliers or *wrap-around* issues resulting in multi-modal distributions such as commonly encountered in robotics applications. As will be illustrated, these nonstandard measurement errors can dramatically compromise the effectiveness of standard estimation methods, while a computational Bayesian approach developed here is demonstrated to be equally effective as standard methods in standard measurement noise scenarios, but dramatically more effective in nonstandard measurement noise distribution scenarios.

Keywords: Bayesian inference, Hamiltonian Monte Carlo, Gaussian mixture models.

*E-mail adresses to authors: `firstname.lastname@newcastle.edu.au`. JD and AW are with the School of Engineering, The University of Newcastle, Australia. BN is with the Faculty of Engineering and Built Environment, The University of Newcastle, Australia. This work was supported by the Australian Research Council Discovery Project DP140104350.

1 Introduction

The distribution of the noise is an important assumption in modelling dynamical systems. This as the accuracy of the parameter estimates of the model obtained by system identification methods [Ljung, 1999] can suffer if the noise model is wrong or does not capture the main characteristics of the noise. This is possibly a common problem as outliers and other un-modelled mechanisms naturally occur in many real world situations. In this paper, we consider and illustrate the potentially dramatic effect of noise distribution mis-modelling and develop a method which estimates the noise model internally without detriment to system parameter estimates.

In particular, we consider a Bayesian approach [Robert, 2007, Gelman et al., 2013] to dynamic system estimation. We illustrate this approach using the very commonly employed auto-regressive exogenous input (ARX) model

$$A(q)y_t = B(q)u_t + e_t, \tag{1}$$

where $y_t \in \mathbb{R}$ and $u_t \in \mathbb{R}$ denotes the output and input of the system at time t respectively and the system polynomials are given by

$$A(q) = 1 + \sum_{k=1}^{n_a} a_k q^{-k}, \quad B(q) = \sum_{k=1}^{n_b} b_k q^{-k}, \tag{2}$$

where $a_{1:n_a} \triangleq \{a_k\}_{k=1}^{n_a}$ and $b_{1:n_b}$ denote the unknown system polynomial coefficients. The shift operator is denoted by q, i.e., $q^{-k}y_t = y_{t-k}$. Here, it is assumed that the system orders n_a and n_b are unknown parameters to be estimated from data.

To be able to handle a wide range of noise distributions, we assume that the noise e_t can be modelled using a Gaussian mixture model (GMM) parameterised by

$$p(e_t) = \sum_{k=1}^{n_e} w_k \, \mathcal{N}(e_t; \mu_k, \sigma_k^2), \quad \sum_{k=1}^{n_e} w_k = 1, \tag{3}$$

where w_k, μ_k and σ_k denote the weight, mean and standard deviation of the mixture component k, respectively. The GMM can successfully capture the behaviour of many types of commonly encountered noise distributions including multi-model, skewed and long/heavy-tailed cases.

The main contributions of this paper are the following:

(i) developing a novel Bayesian ARX (BARX) model.

(ii) tailoring an efficient sampling scheme for inference.

The main features of BARX is that:

(a) it includes sparsity promoting priors to automatically determine the required model orders n_a and n_b from data and;

(b) the inclusion of a GMM to model the noise in a semi-parametric manner, which adapts its flexibility automatically to promote a sparse solution.

The inference is carried out using Hamiltonian Monte Carlo (HMC; Neal, 2010, Betancourt, 2017), which is an efficient Markov chain Monte Carlo (MCMC) scheme for estimating high dimensional posterior distributions.

We offer three numerical illustrations to study the BARX model using synthetic and real-world data. These establish that the approach can perform well in different scenarios and that its data-driven nature is able to correctly estimate ARX model parameters, model orders and noise distributions from data. Moreover, we observe that BARX can provide superior one-step-ahead predictors in comparison with other Bayesian approaches.

There is much related previous work to the present paper. These include work within signal processing and system identification to allow for handling outliers within ARX models, see e.g., Christmas and Everson [2011], Dahlin et al. [2012] and Troughton and Godsill [1998]. However, the BARX model also allows for skewed and multi-modal noise, which is more general than just handling outliers.

Finite and infinite mixtures of Gaussian have a long history in statistics and machine learning, see e.g., Frühwirth-Schnatter [2006] and Escobar and West [1995]. For example, Malsiner-Walli et al. [2016] consider similar problems but makes use of Gibbs sampling and do not consider time series models, only regression. Moreover, Baldacchino et al. [2017] considers a mixture of Student's t distributions for a class of time series models (not ARX) and use variational Bayes to estimate the posterior.

2 Sparse Bayesian modelling

This paper employs a Bayesian approach to the estimation of the model orders, system parameters and noise mixture model coefficients in the model structure (1)-(3). Delivering this rests on two key aspects - the selection of priors on all components to be estimated and provision of a means to combine these with the data likelihood to provide posterior estimates. In this section we will address prior distribution selection.

2.1 System polynomial coefficients

The estimation of the system polynomials is basically a linear regression problem. In the Bayesian setting, this requires us to choose a prior distribution for the coefficients. The typical choice is a uniform distribution or a Gaussian distribution, which leads to a closed-form expression for the posterior if the likelihood is Gaussian. These priors are known as conjugate priors in the Bayesian literature, see e.g., Bishop [2006] and Murphy [2012]. Furthermore, the use of a Gaussian prior is equivalent to L_2-regularised least squares.

In this work, we assume a different prior that induces sparsity - unnecessary coefficients in an over-parametrized model are shrunk towards zero. This allows us to select a maximum model order for the system polynomials and the sparsity promoting mechanism will decide the appropriate model order from the data.

The optimal sparsity promoting prior should have a large probability mass at zero and long tails to accommodate for possible large system polynomial coefficients. A prior with these qualities is given by

$$[a_{1:n_a} b_{1:n_b}] \sim \mathcal{N}(0, \sigma_f^2), \quad \sigma_f \sim \mathcal{C}_+(0,1), \tag{4}$$

where $\mathcal{C}_+(\mu, \gamma)$ denotes the Cauchy distribution restricted to be positive with location $\mu \in \mathbb{R}$ and scale $\gamma > 0$.

This choice corresponds to a so-called *horseshoe prior*, which exhibits the above desired properties since the Cauchy distribution puts a large fraction of its probability mass around its mode while also possessing infinite variance. However, it has tails that decay geometrically and therefore coefficients will be shrunk towards zero if there is no evidence in the data saying otherwise.

The horseshoe prior has been shown to perform better (in the MSE sense) in regression problems than e.g., a Laplacian or Gaussian priors, see e.g., Polson and Scott [2010] and Armagan et al. [2011]. These benchmarks and promising results in a pilot study (not presented here) are the main reasons for this choice of prior distribution.

2.2 Noise distribution

The noise distribution is given by the GMM in (3) with unknown order, weights, means and variances. This is a powerful and versatile model class that covers many different interesting distributions as illustrated in Figure 1.

Here, the green histograms present the data simulated from four different distributions (orange lines): a uniform distribution, a GMM, a skewed Gaussian and a heavy-tailed Student's t. The solid green line is the corresponding kernel density estimates formed from these histograms. The GMM distribution components are shown in shaded grey with their summation as a solid grey line.

Note that in all these cases the GMM can successfully model all of these distributions using only five components. As a consequence, the model structure (1)-(3) will be able to accommodate all these types of noise distributions without prejudicing the quality of the estimates of the system polynomials (2).

A popular approach for Bayesian inference in mixture models is to assume a non-parameteric prior known as the Dirichlet process (DP; Gelman et al., 2013, Escobar and West, 1995) on the mixture

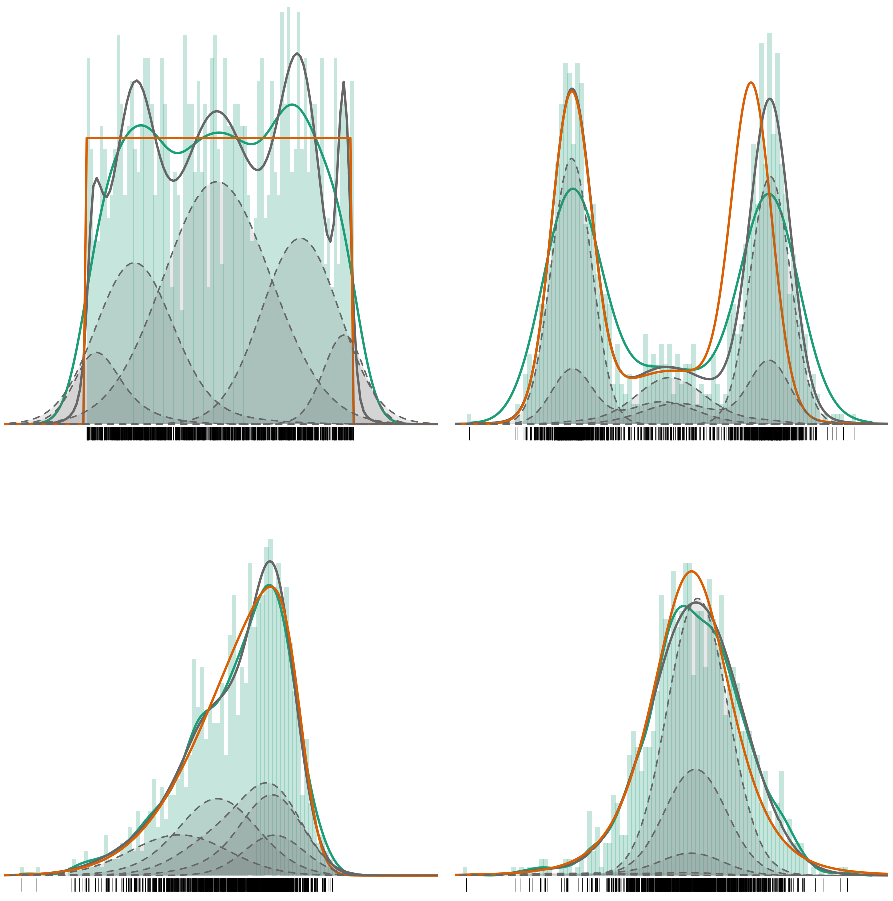

Figure 1: Four examples of GMMs (orange lines) for capturing the behaviour of the noise: uniform noise (top left), multi-modal noise (top right), skewed noise (bottom left) and heavy-tailed noise (bottom right). The histogram of the generated data is presented together with the kernel density estimate (green lines) and the GMM approximation (gray lines). The mixture components in the GMM are presented as gray lines and areas.

weights. The benefit of a DP is that it automatically determines the number of components required to describe the data and is flexible in the sense the this number increases with the number of observations.

However the drawback is that inference in such models can be challenging, especially using common MCMC methods as realisations from the associated Markov chain can exhibit poor mixing which results in a large computational burden. Furthermore, the fact that the expected number of components grows with the amount of data is not desirable in our setting as the aim is to find the *correct* number of components of the noise distribution.

Instead, we make use of recent progress in introducing sparsity in over-parameterised Bayesian finite mixture models. This allows for more efficient inference and avoids the numerical challenges connected with infinite mixture models. To achieve this, as in (4) we assume a sparsity inducing prior for the component means

$$\mu_{1:n_e} \sim \mathcal{N}(0, \sigma_\mu^2), \quad \sigma_\mu \sim \mathcal{C}_+(0,1),$$

Moreover, we follow the recommendations by Gelman et al. [2013] and also select a half-Cauchy distribution as the prior for $\sigma_{1:n_e}$,

$$\sigma_{1:n_e} \sim \mathcal{C}_+(0,5),$$

which corresponds to a weakly informative prior on $\sigma_{1:n_e}$.

Finally, we assume a Dirichlet prior for the mixture weights

$$w_{1:n_e} \sim \mathcal{D}(e_0, \dots, e_0),$$

where n_e denotes the maximum number of components that could be present in the mixture. The Dirichlet distribution is a standard choice in Bayesian mixture models as it is a distribution over the simplex, which means that the sum of $w_{1:n_e}$ is one at all times. Note that, the Dirichlet distribution is the finite dimensional equivalent to the Dirichlet process. Furthermore, it is also the appropriate conjugate prior for the the multinomial distribution, which allows for closed-form expressions for the posterior of the mixture weights.

The Dirichlet distribution can be used to introduce sparsity in the mixture by setting e_0 to a small number. This results in only a few weights receiving the majority of the probability mass and therefore creating a mixture with a few active components if the data does not strongly support more active components.

A popular choice introduced by Ishwaran and Zarepour [2002] is to select $e_0 = 0.1 n_e^{-1}$ as the value of the hyperprior. This empties superfluous components of the mixture and mimics the behaviour of the DP with concentration parameter 0.1 asymptotically [Rousseau and Mengersen, 2011]. However, empirical studies by e.g., Miller and Harrison [2013] have suggested that this approach typically over-estimate the number of components. Another option is to use a hyper-prior for the concentration parameter as proposed by Malsiner-Walli et al. [2016],

$$e_0 \sim \mathcal{G}(\alpha_w, n_e \alpha_w), \tag{5}$$

which is a Gamma distribution with mean n_e^{-1} and variance $(\alpha_w n_e^2)^{-1}$. Hence a large value of α_w results in a prior that is concentrated around n_e^{-1} with a small variance. If n_e is large, this results in that only a few components are a priori given a large mixture weight. This intuition is validated empirically by Malsiner-Walli et al. [2016] which argue that this approach with $\alpha_w = 10$ usually recovers a mixture with the correct number of components. We have also seen evidence supporting this in our preliminary work (not presented here). Note that the consistency results obtained by Rousseau and Mengersen [2011] also holds in this case and this is the reason for choice of (5) as the prior structure for the mixture weights in this paper.

3 Hamiltonian Monte Carlo Computation of Posteriors

The parameter vector for the BARX model is given by

$$\theta = \{a_{1:n_a}, b_{1:n_b}, w_{1:n_e}, \mu_{1:n_e}, \sigma_{1:n_e}, \sigma_\mu, e_0\}, \tag{6}$$

which needs to be estimated from the input-output data. Having specified the priors for these parameters, we now turn to the question of how to combine these priors with the likelihood of the observed data to deliver the required Bayesian posterior estimates.

We approach this via a computational Bayesian approach wherein a Markov chain is constructed with an invariant density which will converge to being equal to the posterior of interest. That is, we construct a random number generator that generates realisations from the posterior, which can be used to obtain estimates of e.g, the posterior mean and its uncertainty.

This can be done using a Metropolis-Hastings (MH; Robert and Casella, 2004) algorithm which operates by sampling from some *proposal* Markov chain governed by

$$\theta' \sim q(\theta' | \theta_{k-1}), \tag{7}$$

where $q(\cdot)$ denotes some Markov kernel. The candidate process that generates θ' is then modulated by setting $\theta_k \leftarrow \theta'$ with probability

$$\alpha(\theta', \theta_{k-1}) = \frac{\pi(\theta')}{\pi(\theta_{k-1})} \frac{q(\theta_{k-1}|\theta')}{q(\theta'|\theta_{k-1})}, \tag{8}$$

otherwise the candidate is rejected and $\theta_k \leftarrow \theta_{k-1}$. Here, $\pi(\cdot)$ denotes the *target distribution* which in our case is which is the parameter posterior

$$\pi(\theta) = p(\theta|y_{1:T}) \propto p(\theta)p(y_{1:T}|\theta). \tag{9}$$

MH is very simple algorithm, but unfortunately its effectiveness is very much dependent on how well the proposal density $q(\cdot)$ is *tuned* to the the target density $\pi(\cdot)$. If the two are not similar then the acceptance probability α either rejects a great majority of candidates θ' that propose any significant movement, or accepts a high proportion of tiny movements, and in both cases generates highly correlated realisations with very poor sample average convergence rates to underlying true estimates.

HMC [Neal, 2010, Betancourt, 2017] methods address this problem by replacing the target density $\pi(\theta)$ with an extended target given by

$$H(\theta, p) = -\log \pi(\theta) + \frac{1}{2} p^\top M^{-1} p, \tag{10}$$

which is known as the Hamiltonian in physics. Here, we introduce an auxiliary *momentum* variable p to facilitate computation, which later can be removed by marginalisation. A standard choice is that p is a zero-mean Gaussian with the so-called mass matrix M as its covariance matrix.

Samples from the posterior can be obtained by simulating the dynamical system described by (10). That is, we explore level sets of the posterior where the Hamiltonian is constant, which enables almost uncorrelated sampling from it compared with the small local moves usually employed within MH. The possibility of larger steps using the Hamiltonian dynamics is what decreases the correlation in the samples, which is beneficial for the performance of the sampling. The simulation is carried out by solving

$$\frac{\partial \theta}{\partial t} = \frac{\partial H(\theta, p)}{\partial p} = M^{-1} p, \tag{11a}$$
$$\frac{\partial p}{\partial t} = -\frac{\partial H(\theta, p)}{\partial \theta} = \nabla_\theta \log \pi(\theta), \tag{11b}$$

over a user chosen time period. Note that the gradient of the log-posterior enters into the time derivative of the momentum. Hence, it will help to guide the Markov process to areas of high posterior probability.

Unfortunately, it is not possible to solve (11) in closed-form for this particular problem. Instead, it is possible to make use of numerical integration methods to simulate the simulation. However, not all methods are applicable for this, see Neal [2010] for details, but leap-frog integrators can be employed to numerically simulate (11). The resulting move from θ_{k-1} to θ' is accepted with probability

$$\alpha(\theta_k|\theta_{k-1}) = \exp\Big(- H(\theta', p') + H(\theta_{k-1}, p_{k-1})\Big). \tag{12}$$

This basically accepts candidates if the Hamiltonian does not decrease significantly over the simulation. Hence it corrects for the error introduced by the numerical integration.

The implementation of HMC requires the user to determine the mass matrix M as well as the step-length and number of steps in the leap-frog algorithm. In this paper, we make use of STAN [Stan Development Team, 2017] for implementing HMC and this software includes an adaptive method known as NUTS [Hoffman and Gelman, 2014] to set these parameters automatically.

4 Numerical illustrations

We provide three different numerical illustrations to investigate the properties and the performance of the proposed BARX model using both synthetic and real-world data. In each of the illustrations, the *same* model structure and algorithmic settings are used and the *only* difference is the input-output data supplied to the algorithm. All implementation details are summarised in Appendix A and the source code is available via GitHub, see Section 5.

4.1 Synthetic data with Gaussian noise

We generate a data set with $T = 1,000$ observations using the system polynomial coefficients

$$a_{1:2} = \{-1.5, 0.7\}, \quad b_{1:3} = \{0, 1, 0.5\},$$

using a pseudo-binary input signal and adding standard Gaussian noise to the output signal. We partition the output signal into an estimation and a validation set, using 2/3 and 1/3 of the data, respectively.

The BARX estimate model is compared with the results of the `arx` command in MATLAB combined with a cross-validation scheme to find the model order. Regarding the latter, we partition the estimation data into two sets of equal length and make use of the first part for estimating models where n_a and n_b varies between 1 and 5. The model order is selected as the one that minimises the squared prediction error computed on the second part. This approach selects $n_a = 3$ and $n_b = 5$.

Figure 2 summarises the results obtained from the experiment. In the top plot, the validation data set (dots) is presented together with the one-step-ahead predictors obtained via the BARX model (orange) and `arx` (purple). We note that the predictors are virtually the same for most time steps. Using the

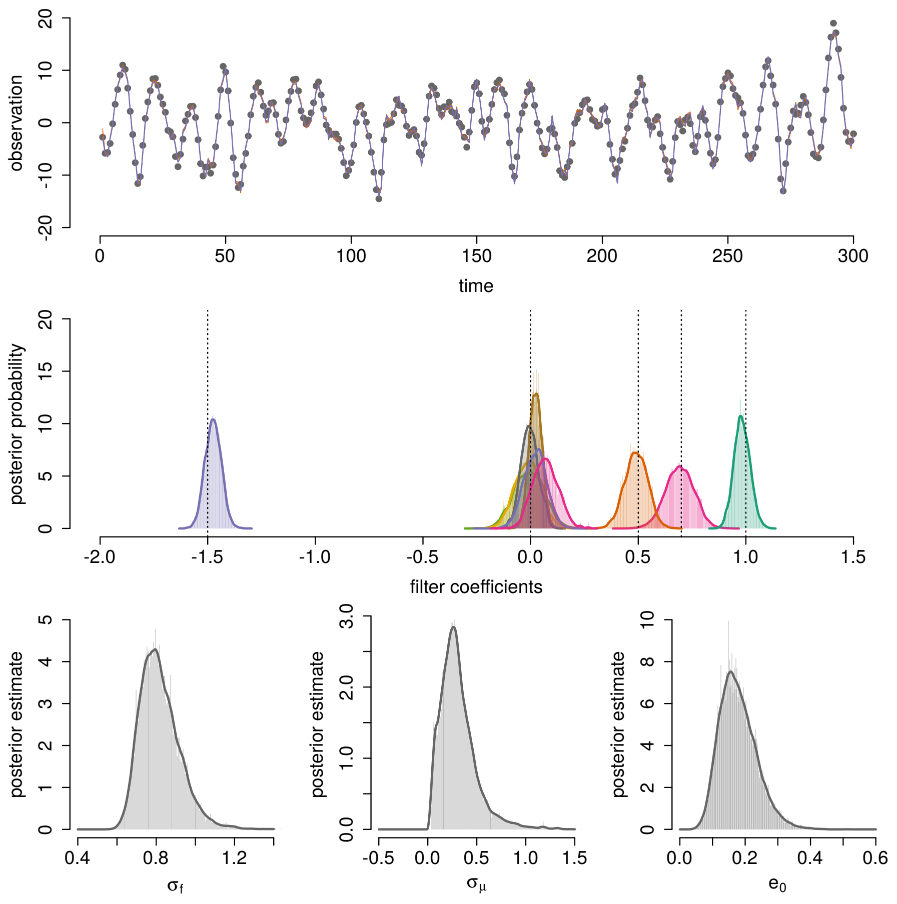

Figure 2: Top: the one-step-ahead predictors for `arx` (purple) and BARX (orange) models versus validation data (dots) in a model with unimodal Gaussian noise. The estimated posteriors of the system polynomial coefficients (middle) and priors (bottom) are also presented. Dotted vertical lines indicate the true system coefficients.

predictors, the model fit is computed on validation data by

$$\mathsf{MF} = 100\left(1 - \frac{\sum_{t=1}^{T_v}(y_t - \widehat{y}_t)^2}{\sum_{t=1}^{T_v}(y_t - \bar{y})^2}\right),$$

where $\widehat{y}_t$ and $\bar{y}$ denote the one-step-ahead predictor and the sample mean of the output signal, respectively. Note that the mean of the predictive distribution from BARX is used for this computation. We obtain the model fit 96.6% and 96.5% for `arx` and BARX, respectively. This indicates that BARX can replicate the result of `arx` for models with unimodal Gaussian noise and thereby offers a good sanity. Here, `arx` runs in a few seconds and the estimation of BARX takes a few minutes.

The middle and bottom plots in Figure 2 present the estimates of the system coefficients and the estimates of the prior distributions for the BARX model, respectively. Note that the sparsity of the horseshoe prior in the system coefficients has shrunk most of system coefficient posteriors towards zero. At least four coefficients do not overlap zero and therefore remain statistically significant, which corresponds well to the model from which the data was generated.

From the posteriors of the priors, we can see that the prior for the system coefficients can accommodate larger deviations from zero than the prior for the mixture means. This is reasonable as some of the system coefficients are quite far from zero. Finally, we note that the prior on the mixture weights indicates a sparse behaviour as it is quite small and therefore promotes a few mixture components to have large weights with the rest being small.

4.2 Synthetic data with Gaussian mixture noise

We continue with a more interesting example where the output is corrupted with noise from a GMM with a multi-modal behaviour. This might occur in practice due to unmodelled system behaviour of sensor imperfections. We generate a data set with $T = 1000$ observations from an ARX system (1) with system coefficients (2)

$$a_{2:3} = \{-0.25, 0.2\}, \quad b_{1:3} = \{0, 1, 0.5\},$$

and being driven by an exogenous input u_t being an i.i.d. zero mean and unit variance Gaussian process. The observations y_t involve a noise sequence e_t which are i.i.d. realisations from a GMM (3) given by

$$p(e_t) = 0.4 \cdot \mathcal{N}(e_t : 7, 1) + 0.6 \cdot \mathcal{N}(e_t : 0, 1).$$

This means that the mean of the noise switches between 0 and 7 due to some unknown underlying process. The model fit for this data set obtained by `arx` using the same approach as in Section 4.1 is -0.22%.

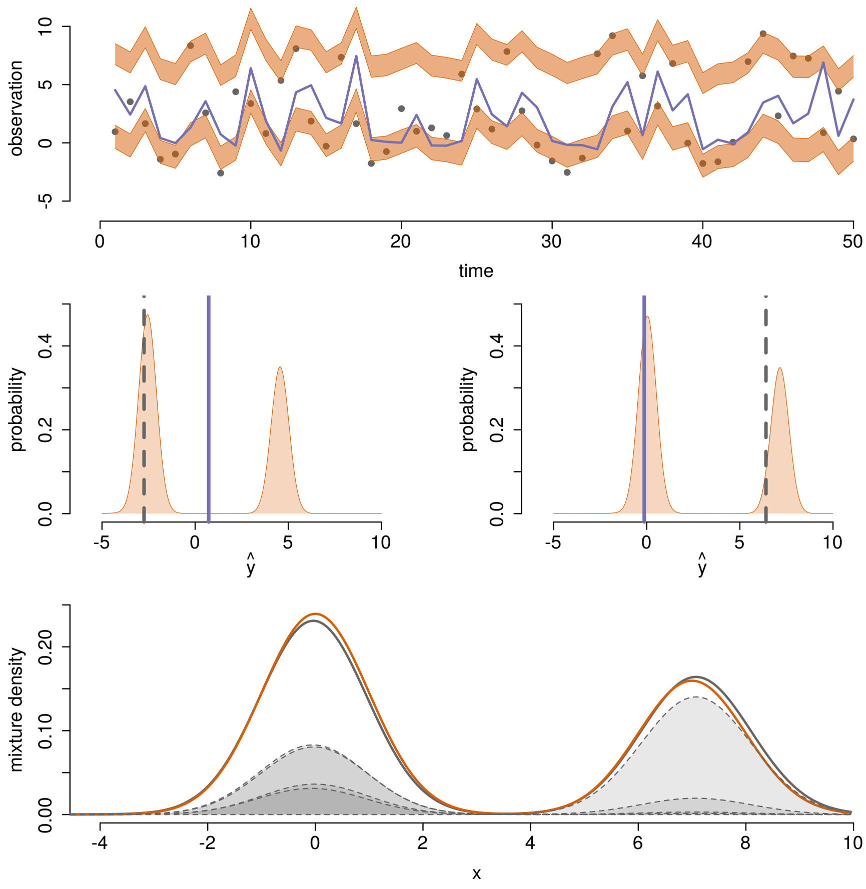

Figure 3: BARX with Gaussian mixture noise. Top: validation data (grey dots) with HPD of the one-step-head prediction distribution from BARX (orange) and the corresponding prediction from `arx` (purple). Middle: The one-step-head predictions for `arx` (solid line) and BARX (orange) at two time-steps with the true observations (dashed line). Bottom: the estimated noise distribution (grey solid line) together with the true distribution (orange) and the GMM components (grey areas). See the main text for details and a discussion.

This poor performance is due to the obvious violation of the Gaussian noise assumption, but it can be challenging to validate these assumptions in practice. To compare, we fit the BARX model to the same data to see how its data-driven properties handle the bi-modal noise distribution.

Figure 3 presents the results from obtained by BARX. In the top part, we see the high posterior density (HPD) intervals of the one-step-ahead prediction distribution obtained from BARX (orange) together the `arx` predictor (purple) and the validation data (dots). Note that the predictor from `arx` tries to capture the bi-modality by adjusting the mean but this results in poor performance as no data is found in the gap between the two modes.

In the middle, we present the one-step-ahead prediction distributions (orange) from BARX together from the same prediction from `arx` (purple solid line) and the true observation (grey solid line). The `arx` predictor struggles in these to cases and is far away from the true observation which is covered by the distribution from BARX. In the lower part, we present the true noise distribution (orange line) together with the BARX estimate (gray line). BARX is able to provide a good estimate of the noise distribution in this case.

As BARX is a Bayesian approach, the one-step-ahead predictor is actually a distribution which is able to capture the multi-modal nature of the data generating model. This highlights the benefit of working with distributions of quantities instead of point-estimates, which are the standard approach in likelihood-based System Identification methods. Hence, a distribution of the predictor could be useful within e.g., stochastic MPC.

4.3 Real-world EEG data

Finally, we revisit a real-world EEG data set analysed in Dahlin et al. [2012], where the authors assume an ARX model with Student's t-distributed noise. We apply the same procedure as before to estimate the BARX model. There is no input signal in this data set.

Figure 4 presents the results obtained by using BARX. The one-step-ahead predictors seem to perform well and this is reflected in a high model fit of 92.3% on validation data for the BARX model, which is a substantial increase from the model fit of 85.6% obtained in Dahlin et al. [2012].

Moreover, we note that the estimated noise of the data set is clearly not Gaussian, as the behaviour of the tails are very different. An interesting finding from this experiment is that only $\{a_1, a_2, a_4, a_5, a_6, a_7\}$ seems to be non-zero and contribute to the predictor.

5 Conclusions

The numerical illustrations indicate proposed BARX model can be useful in problems where non-standard measurement noise corruptions cause traditional system estimation approaches to fail by a significant

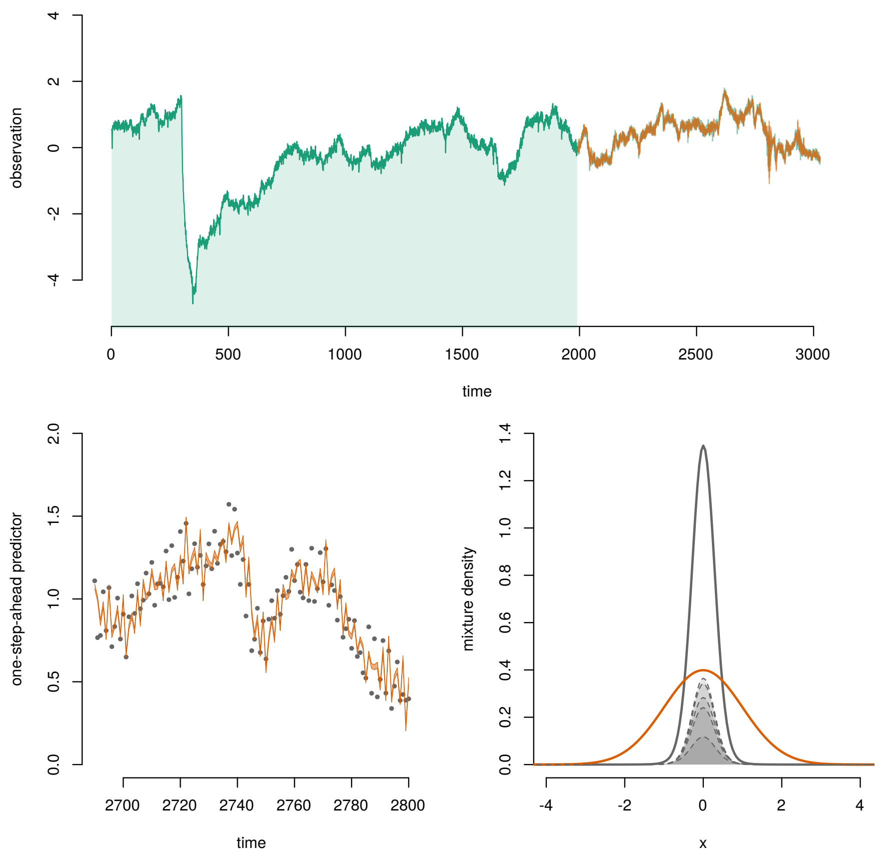

Figure 4: BARX model estimated on real-world EEG data. Top: the estimation data (shaded green) and validation data set (unshaded green) are presented together with the one-step-ahead predictor (orange). Bottom left: zoom-in of top plot with validation data indicated by dots. Bottom right: the estimate of the noise distribution with the best Gaussian approximation (orange) to the estimation data.

margin. BARX performs equivalently to standard methods when the measurement noise is unimodal and Gaussian, which makes it an attractive option when the noise distribution is known as possibly multi-modal.

Hence, BARX is an automated data-driven approach that adapts its complexity to capture the dynamic behaviour observed in the data. This is similar to many modern non-parameteric Bayesian models employed within e.g., machine learning, see Ghahramani [2015]. The main benefit with this kind of model is that its complexity is allowed to scale with the amount of available data. It is therefore able to capture more and more complicated system behaviours as more data is accumulated.

Future work includes the generalisation to more flexible model structures such as Box-Jenkins transfer function models. Moreover, it would also be interesting to integrate this inference approach within MPC to create controllers that can handle e.g., multi-modal or heavy-tailed noise.

The source code and data used in this paper are available from GitHub `https://github.com/compops/barx-sysid2018/` and via Docker (see `README.md`).

Acknowledgements

The EEG data was kindly provided by Eline Borch Petersen and Thomas Lunner at Eriksholm Research Centre, Oticon A/S, Denmark. The authors would like to thank Johan Wågberg for comments on an earlier version of this paper.

References

A. Armagan, M. Clyde, and D. B. Dunson. Generalized beta mixtures of Gaussians. In *Proceedings of the 2011 Conference on Neural Information Processing Systems (NIPS)*, Granada, SP, December 2011.

T. Baldacchino, K. Worden, and J. Rowson. Robust nonlinear system identification: Bayesian mixture of experts using the t-distribution. *Mechanical Systems and Signal Processing*, 85:977–992, 2017.

M. Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *Pre-print*, 2017. arXiv:1701.02434.

C. M. Bishop. *Pattern recognition and machine learning.* Springer Verlag, New York, USA, 2006.

J. Christmas and R. Everson. Robust autoregression: Student-t innovations using variational Bayes. *IEEE Transactions on Signal Processing*, 59(1):48–57, 2011.

J. Dahlin, F. Lindsten, T. B. Schön, and A. Wills. Hierarchical Bayesian ARX models for robust inference.

In *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, Brussels, Belgium, July 2012.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

S. Frühwirth-Schnatter. *Finite mixture and Markov switching models*. Springer Verlag, 2006.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 3 edition, 2013.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

M. D. Hoffman and A. Gelman. The No-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

H. Ishwaran and M. Zarepour. Dirichlet prior sieves in finite normal mixtures. *Statistica Sinica*, 12(3): 941–963, 2002.

L. Ljung. *System identification: theory for the user*. Prentice Hall, 1999.

G. Malsiner-Walli, S. Frühwirth-Schnatter, and B. Grün. Model-based clustering based on sparse finite Gaussian mixtures. *Statistics and computing*, 26(1-2):303–324, 2016.

J. W. Miller and M. T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Proceedings of the 2013 Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, USA, December 2013.

K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Press, 2010.

N. G. Polson and J. G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

C. P. Robert. *The Bayesian choice*. Springer Verlag, 2007.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2 edition, 2004.

J. Rousseau and K. Mengersen. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):689–710, 2011.

Stan Development Team. Stan: A C++ library for probability and sampling, version 2.14.0, 2017. URL `http://mc-stan.org/`.

P. T. Troughton and S. J. Godsill. A reversible jump sampler for autoregressive time series. In *Proceedings of the 23rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, USA, May 1998.

A Implementation details

The implementation details are almost the same for all three illustrations in Section 4. In the first two experiments, we make use of a maximum order of 5 for the system polynomials, i.e., $n_a = n_b = 5$ and use $n_e = 5$ components for the noise distribution mixture. In the third experiment, we set $n_a = 10$ and $n_b = 0$ (as no input is present). These values have not been calibrated in any way and were just selected arbitrarily. The hyperpriors are selected as in Section 2.

The HMC sampler is implemented using the STAN software [Stan Development Team, 2017] and NUTS [Hoffman and Gelman, 2014] to adaptive the settings of the algorithm. We run the sampler for $30,000$ iterations discarding the first $15,000$ iterations as burn-in.

The initialisation of the system coefficients and the means of the mixture components greatly influences the accuracy of the estimates. We therefore initialise the former randomly over the interval $(-1, 1)$ and the latter on an equally spaced grid over the range of the output data. All mixture weights are initialised as $1/n_e$ and the remaining parameters are initialised as 1.