The Journal of Open Source Software

StateSpaceDynamics.jl: A Julia package for probabilistic state space models (SSMs)

Ryan Senne [1,2], **Zachary Loschinskey** [1], **Carson Loughridge** [1], **James Fourie** [1], **and Brian D. DePasquale** [1,2]

1 Department of Biomedical Engineering, Boston University **2** Graduate Program for Neuroscience, Boston University

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

State-space models (SSMs) are powerful tools for modeling time series data that naturally arise across a variety of domains, including neuroscience, finance, and engineering. The unifying principle of these models is that they assume that an observation sequence, $Y_1, Y_2, ..., Y_T$, is generated through an underlying hidden latent sequence, $X_1, X_2, ..., X_T$. This general framework encompasses two of the most popular models for time series analysis: the Hidden Markov Model (HMM) and the (Gaussian) Linear Dynamical System (LDS, i.e., the Kalman filter). Thus, SSMs provide a probabilistic framework for describing the temporal evolution of many phenomena, and their generality naturally leads to widely applicable use cases. We introduce `StateSpaceDynamics.jl`, an open source, modular package designed to be fast, readable, and self contained for the express purpose of fitting a plurality of SSMs, easily in Julia. (Senne et al., 2024)

Statement of need

Advancements in systems neuroscience have enabled the collection of massive, multivariate, and complex time series datasets, where simultaneous recordings from hundreds to thousands of neurons are increasingly common. Interpreting these high-dimensional recordings presents a significant challenge. Recent modeling approaches suggest that neural activities can be effectively characterized by a set of latent factors evolving over a low-rank manifold. Consequently, there is a growing need for models that combine dimensionality reduction with temporal dynamics, for which state-space models (SSMs) provide a natural framework.

While advanced SSM implementations exist in Python—such as the SSM package (S. Linderman, 2022) and Dynamax (Chang et al., 2024), the Julia programming language lacks an equivalent library that meets the needs of modern neuroscientists. Existing Julia offerings, like StateSpaceModels.jl (Bodin, Guilherme, 2024), are limited to Gaussian observation models and rely on analytical calculation of the marginal log-likelihood function. This fundamental limitation precludes the analysis of non-Gaussian observations, which are common in neuroscience where spike counts often follow Poisson or other discrete distributions. Furthermore, the requirement for analytical computation of the marginal likelihood integral:

$$p(y_{1:T}|\theta) = \int_{x_{1:T}} p(y_{1:T}|x_{1:T}, \theta)p(x_{1:T}|\theta)dx_{1:T} \tag{1}$$

restricts the development of learning algorithms for non-conjugate observation models. To address these limitations, we have developed StateSpaceDynamics.jl, which provides a flexible framework for fitting a variety of SSMs, including non-Gaussian observation models, while maintaining computational efficiency.

Package design

To address these limitations, we have developed `StateSpaceDynamics.jl`, which employs a previously advocated approach of directly maximizing the complete-data log-likelihood with respect to the hidden state path for Linear Dynamical System models (Paninski et al., 2010). By leveraging the block-tridiagonal structure of the Hessian matrix, this method allows for the exact computation of the Kalman smoother in O(T) time (Paninski et al., 2010). Furthermore, it facilitates the generalization of the Rauch–Tung–Striebel (RTS) smoother to accommodate other observation noise models (e.g., Poisson and Bernoulli), requiring only the computation of the gradient and Hessian of the new model to obtain an *exact* maximum a posteriori (MAP) path (Macke et al., 2011).

Furthermore, given the analytical Hessian matrices are available, once can make use of this to perform an approximate EM algorithm by generating an LaPlace approxiamtion of the posterior distirubtion of the latent states. One can easily make use of fast inversion algorithms of the negative Hessian (i.e., Fisher Information Matrix), which are block-triadiagonal (Rybicki & Hummer, 1990). From here one can compute the approximate second moments of the posterior distribution i.e., Cov(X_t, X_t) and Cov(X_t, X_{t-1}), and use the analytical updates of the canonical LDS (Bishop, 2006; Paninski et al., 2010). This approach becomes exact EM in the case of Gaussian observations.

Lastly, `StateSpaceDynamics.jl` provides implementations od discrete state space models i.e., Hidden Markov Models, and the ability to fit these models using the EM algorithm. While this is not the primary development target of the package, these models are necessary for the development of hierarchichal models, e.g., the switching LDS (SLDS) and the recurrent switching LDS (rSLDS) (S. W. Linderman et al., 2016; Murphy, 1998). However, the recent development of HiddenMarkovModels.jl, may make this feature redundant, and our future work my entail directly interfacing with this package (Dalle, 2024). Nonetheless, we provide a suite of HMM models popular in neuroscience including the classic Gaussian HMM and GLM-HMMs.

By providing these features, `StateSpaceDynamics.jl` fills a critical gap in the Julia ecosystem, offering modern computational neuroscientists the tools necessary to model complex neural data with state-space models that incorporate both dimensionality reduction and temporal dynamics. # Example

Model architecture

Consider the Poisson Linear Dynamical System (PLDS):

$$X_t \sim \mathcal{N}(AX_{t-1}, Q) \tag{2}$$
$$Y_t \sim \text{Poisson}(f(CX_t + b)\Delta t) \tag{3}$$
$$X_0 \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{4}$$

Availability

`StateSpaceDynamics.jl` is publicly available under the GNU license at https://github.com/depasquale-lab/StateSpaceDynamics.jl.

Conclusion

Author contributions

RS was the primary developer of StateSpaceDynamics.jl, implementing the core algorithms, designing the package architecture, and writing the manuscript. ZL (Zachary Loschinskey), CL (Carson Loughridge), and JF (James Fourie) contributed to package development, including implementation of key features, testing, and documentation. BDD (Brian D. DePasquale) conceived the project, provided theoretical guidance and technical oversight throughout development, secured funding, and supervised the work. All authors reviewed and approved the final manuscript.

Acknowledgements

This work was supported by the Biomedical Engineering Department at Boston University.

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bodin, Guilherme, S., Raphael. (2024). *StateSpaceModels.jl*. https://github.com/LAMPSPUC/StateSpaceModels.jl

Chang, P., Harper-Donnelly, G., Kara, A., Li, X., Linderman, S., & Murphy, K. (2024). *Dynamax: State space models library in JAX* [Computer software]. Github.

Dalle, G. (2024). *HiddenMarkovModels.jl*. https://github.com/gdalle/HiddenMarkovModels.jl

Linderman, S. (2022). *SSM: Bayesian learning and inference for state space models*. https://github.com/lindermanlab/ssm

Linderman, S. W., Miller, A. C., Adams, R. P., Blei, D. M., Paninski, L., & Johnson, M. J. (2016). Recurrent switching linear dynamical systems. *arXiv [Stat.ML]*.

Macke, J. H., Buesing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V., & Sahani, M. (2011). *Empirical models of spiking in neural populations*. *24*. https://proceedings.neurips.cc/paper_files/paper/2011/file/7143d7fbadfa4693b9eec507d9d37443-Paper.pdf

Murphy, K. P. (1998). *Switching kalman filters* [Technical report].

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rahnama Rad, K., Vidne, M., Vogelstein, J., & Wu, W. (2010). A new look at state-space models for neural data. *J. Comput. Neurosci.*, *29*, 107–126.

Rybicki, G. B., & Hummer, D. G. (1990). *Fast solution for the diagonal elements of the inverse of a tridiagonal matrix*.

Senne, R., Loughridge, C., Loschinsky, Z., & DePasquale, B. (2024). *StateSpaceDynamics.jl: A julia package for probabilistic state space models*. https://github.com/depasquale-lab/StateSpaceDynamics.jl. https://github.com/depasquale-lab/StateSpaceDynamics.jl